

Proceedings of

THE THIRD INTERNATIONAL WORKSHOP OF THE IFIP
WG 5.7 SPECIAL INTEREST GROUP ON
“ADVANCED TECHNIQUES IN PRODUCTION
PLANNING & CONTROL”

24-25 February 2000, Florence, Italy

Edited by

Mario Tucci, Marco Garetti

Assisted by

Gianni Bettini

Organized by

Università degli Studi di Firenze Dept. of Energy Engineering “Sergio Stecco”
Industrial Plants and Technologies, Firenze, Italia

Politecnico di Milano Dept. of Management
Industrial Engineering and Economics, Milano, Italia

Sponsored by

International Federation for Information Processing
Working Group 5.7 on Integrated Production Management
Special Interest Group on Advanced Techniques in Production Planning & Control

FIRENZE UNIVERSITY PRESS

2002

Proceedings of The Third International Workshop of the IFIP WG5.7 Special Interest Group on “Advanced Techniques in Production Planning & Control”: 24-25 February 2000, Florence, Italy / edited by Mario Tucci, Marco Garetti; assisted by Gianni Bettini. – Firenze : Firenze University Press, 2002.
Modalità di accesso della versione elettronica: <http://epress.unifi.it/>

ISBN 88-8453-042-3

658.503 (ed. 20)

1. Aziende industriali – Produzione – Applicazioni dell'informatica

I. Titolo II. Tucci, Mario

Print on demand is available

© 2002 Firenze University Press

Firenze University Press
Borgo Albizi, 28, 50122 Firenze, Italy
<http://www.unifi.it/e-press>

Grafica e layout
Fulvio Guatelli
consorzioeditoriale@libero.it

TABLE OF CONTENTS

<i>List of contributors</i>	v
<i>Preface</i>	ix
MARIA GRAZIA GNONI, RAFFAELLO IAVAGNILIO, GIORGIO MOSSA, GIOVANNI MUMMOLO <i>Solving a lot sizing and scheduling problem by hybrid modelling</i>	1
PAUL VALCKENAERS, PATRICK PEETERS, HENDRIK VAN BRUSSEL, TAPIO HEIKKILÄ, MARTIN KOLLINGBAUM <i>Pheromone based emergent manufacturing control by multi-agent system</i>	17
JAN-WILHELM BREITHAUPT <i>Advanced backlog-and utilisation control based on flexible capacities</i>	29
AGOSTINO G. BRUZZONE, PIETRO GIRIBONE, ROBERTO MOSCA, ROBERTO REVETRIA <i>Applied neural networks for improving production capabilities</i>	45
SERGIO CAVALIERI, VITTORIO CESAROTTI <i>A multi-agent model for coordinated supply chain planning</i>	51
P. VAN BAELE, M. RIJCKAERT <i>Specific knowledge of the job shop scheduling problem incorporated in local search, how good is it?</i>	71
HANS-HERMANN WIENDAHL, ENGELBERT WESTKÄMPER <i>Manufacturing in turbulent markets effects on production planning and control</i>	81
GIANCARLO MACCARINI, GIOVANNI VALENTINI, LUCIO ZAVANELLA <i>Improving the simulation of manufacturing systems: the implementation of hybrid simulation and fuzzy logic</i>	95

Proceedings of ATPPC 2000

MARIO TUCCI, MARIO RAPACCINI, EMANUELE CHELI, GIANNI BETTINI <i>Towards integrated simulators. First step: please pass the data</i>	107
J.M. VAN DE MORTEL-FRONCZAK, J.E. ROODA <i>Agent-based control of a lithoshop – a simulation study</i>	125
ALISTAIR R. CLARK <i>A local search approach to lot sequencing and sizing</i>	145
FERNANDO LOPES, NUNO MAMEDE, HELDER COELHO <i>Negotiation in a multi-agent supply chain system</i>	153

LIST OF CONTRIBUTORS

- BETTINI GIANNI, Department of Energy Engineering “Sergio Stecco”, Sezione Impianti e Tecnologie Industriali, Università degli Studi di Firenze, Via C. Lombroso 6/17 – Firenze, Italy. Tel. +39 (055) 4796 708, fax +39 (055) 4224137, <http://siti.de.unifi.it>.
- BREITHAUP JTAN-WILHELM, Lufthansa Technik Logistik GmbH, Hamburg Branch HAM UH/B, Weg beim Jaeger 193, – 22335 Hamburg, Germany. Tel. +49 40 5070 4506, E-mail: jan-wilhelm.breithaupt@ltd.dhl.de
- BRUZZONE AGOSTINO G., Dept. University of Genoa, Via Opera Pia 15, – 16145 Genova, Italy. Tel. +39353 28 83, Fax +39 010 317 750.
- CAVALIERI SERGIO, Dipartimento di Economia e Produzione, Politecnico di Milano, Piazza Leonardo da Vinci, 3 – 20133 Milano, Italy. Tel. +39.02.2399.2729, E-mail: sergio.cavaliere@polimi.it.
- CESAROTTI VITTORIO, Dipartimento di Ingegneria Meccanica, Università di Roma Tor Vergata, Via di Tor Vergata, 110, – 00133 Rome. Tel. +39.06.7259.7178, E-mail: cesarotti@uniroma2.it.
- CHELI EMANUELE, Department of Energy Engineering “Sergio Stecco”, Sezione Impianti e Tecnologie Industriali, Università degli Studi di Firenze, Via C. Lombroso 6/17 – Firenze, Italy. Tel. +39 (055) 4796 708, fax +39 (055) 4224137, <http://siti.de.unifi.it>.
- CLARK ALISTAR R., Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, – Bristol, BS16 1QY, England. Tel. +44 (0) 117 344 3134, E-mail: alistair.clark@uwe.ac.uk.
- COELHO HELDER, Faculdade de Ciências, Campo Grande, 1700, – Lisboa, Portugal. E-mail: hcoelho@di.fc.ul.pt.
- GIRIBONE PIETRO, DIP University of Genoa, Via Opera Pia 15, – 16145 Genova, Italy. Tel. +39 353 28 83, Fax +39 010 317 750, <http://st.itim.unige.it>
- GNONI MARIA GRAZIA, Politecnico di Bari, Dept. of Mechanical and Industrial Engineering. Viale Japigia 182, – 70126 Bari, Italy. Tel. +39 080 5962758, fax +39 080 5962788
- HEIKKILÄ TAPIO, Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B, – 3001 Heverlee (Leuven), Belgium. <http://www.mech.kuleuven.ac.be/pma/pma.html>
- IAVAGNILIO RAFFAELLO, Politecnico di Bari, Dept. of Mechanical and Industrial Engineering, Viale Japigia 182 – 70126 Bari, Ital. Tel. +39 080 5962758, fax +39 080 5962788

- KOLLINGBAUM MARTIN, Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B, – 3001 Heverlee (Leuven), Belgium. <http://www.mech.kuleuven.ac.be/pma/pma.html>
- LOPES FERNANDO, INETI, Estrada do Paço do Lumiar, 1699, – Lisboa Codex, Portugal. flopes@dms.ineti.pt
- MACCARINI GIANCARLO, Università degli Studi di Brescia – Facoltà di Ingegneria, Via Branze, 38, – 25123 Brescia (Italy). Tel. +39.030.37151.
- MAMEDE NUNO, IST, Avenida Rovisco Pais, 1049-001, – Lisboa, Portugal. E-mail: Nuno.Mamede@acm.org
- MOSCA ROBERTO, DIP University of Genoa, Via Opera Pia 15, – 16145 Genova, Italy. Tel. +39 353 28 83, Fax +39 010 317 750, <http://st.itim.unige.it>
- MOSSA GIORGIO, Politecnico di Bari, Dept. of Mechanical and Industrial Engineering, Viale Japigia 182 – 70126 Bari, Italy. Tel. +39 080 5962758, fax +39 080 5962788
- MUMMOLO GIOVANNI, Politecnico di Bari, Dept. of Mechanical and Industrial Engineering, Viale Japigia 182, – 70126 Bari, Italy. Tel. +39 080 5962758, fax +39 080 5962788, E-mail: mummolo@poliba.it
- NOVAIS Q., INETI, Estrada do Paço do Lumiar, 1699, – Lisboa Codex, Portugal, E-mail: anovais@dms.ineti.pt
- PEETERS PATRICK, Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B, – 3001 Heverlee (Leuven), Belgium, <http://www.mech.kuleuven.ac.be/pma/pma.html>
- RAPACCINI MARIO, Department of Energy Engineering “Sergio Stecco”, Sezione Impianti e Tecnologie Industriali, Università degli Studi di Firenze, Via C. Lombroso 6/17 – Firenze, Italy. Tel. +39 (055) 4796 708 - fax +39 (055) 4224137, <http://siti.de.unifi.it>
- REVENTRIA ROBERTO, DIP University of Genoa, Via Opera Pia 15, – 16145 Genova, Italy. Tel. +39 353 28 83, Fax +39 010 317 750, E-mail: reventria@itim.unige.it, <http://st.itim.unige.it>
- RIJCKAERT M., K.U. Leuven - Chemical Engineering Department, De Croylaan 46, – 3001 Heverlee, Belgium. E-mail: marcel.rijckaert@cit.kuleuven.ac.be
- ROODA J.E., Eindhoven University of Technology, Department of Mechanical Engineering, P.O. Box 513 – 5600 MB Eindhoven, the Netherlands. Tel. +31 40 247 4553, E-mail: j.e.rooda@tue.nl
- TUCCI MARIO, Department of Energy Engineering “Sergio Stecco”, Sezione Impianti e Tecnologie Industriali, Università degli Studi di Firenze, Via C. Lombroso 6/17 – Firenze, Italy. Tel. +39 (055) 4796 708, fax +39 (055) 4224137, E-mail: mario.tucci@siti.de.unifi.it <http://siti.de.unifi.it>
- VALCKENAERS PAUL, Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B, – 3001 Heverlee (Leuven), Belgium. E-mail: Paul.Valckenaers@mech.kuleuven.ac.be; <http://www.mech.kuleuven.ac.be/pma/pma.html>
- VALENTINI GIOVANNI, Università degli Studi di Brescia, Facoltà di Ingegneria, Via Branze, 38, – 25123 Brescia (Italy). Tel. +39.030.37151.
- VAN BAELE P., K.U. Leuven, Chemical Engineering Department, De Croylaan 46, –

List of contributors

- 3001 Heverlee (Belgium), E-mail: patrick.vanbael@cit.kuleuven.ac.be
- VAN BRUSSEL HENDRIK, Katholieke Universiteit Leuven, Dept. of Mech. Engineering, Division P.M.A., Celestijnenlaan 300-B, – 3001 Heverlee (Leuven), Belgium, <http://www.mech.kuleuven.ac.be/pma/pma.html>
- VAN DE MORTEL-FRONCZAK J.M., Eindhoven University of Technology Department of Mechanical Engineering, P.O. Box 513, – 5600 MB Eindhoven, The Netherlands. Tel. +31 40 247 5697, E-mail: j.m.v.d.mortel@tue.nl
- WESTKÄMPER ENGELBERT, Institute of Industrial Manufacturing and Management (IFF), University of Stuttgart and Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), Nobelstrasse 12, – 70569 Stuttgart, Germany. Tel. +49 (0)711 970-1100, E-mail: wke@ipa.fhg.de
- WIENDAHL HANS-HERMANN, Institute of Industrial Manufacturing and Management (IFF), University of Stuttgart and Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), Nobelstrasse 12, – 70569 Stuttgart, Germany. Tel. +49 (0)711 970-1968, E-mail: hhw@ipa.fhg.de
- ZAVANELLA LUCIO, Università degli Studi di Brescia – Facoltà di Ingegneria, Via Branze 38, – 25123 Brescia (Italy). Tel. +39.030.37151, E-mail: zavanell@bsing.ing.unibs.it

PREFACE

This book is the result of the third International Workshop of the IFIP 5.7 Special Interest Group (SIG) on “Advanced Techniques in Production Planning & Control”. The two previous editions of the International Workshop have been held in Ascona - Switzerland (ATPPC1997), Hannover - Germany (ATPPC 1999).

Scope of the SIG activity is to deal with the wide variety of new and computer-based techniques that has become available to the scientific and industrial world in the past few years: formal modeling techniques, artificial neural networks, autonomous agent theory, genetic algorithms, chaos theory, fuzzy logic, simulated annealing, tabu search, etc. So, in addition to the exploitation of manufacturing paradigms (i.e. lean production and agile manufacturing just to refer some of the last ones), production management and manufacturing strategy can be helped in dealing with manufacturing systems, that are becoming every day more complex and difficult to manage, through the use of such advanced techniques.

Growing attention has been addressed by scientific researchers to the development of new applications of such techniques, as demonstrated by the large and still increasing number of papers, technical reports, survey papers and conference sessions on this topic.

In such a lively context, false expectations and easy enthusiasm are to be avoided, not to repeat what already happened in the past when other computer-based techniques or instruments became available to the scientific community (i.e. expert systems), therefore some clarifications have to be done:

(i) none of the industrial applications of these techniques seems to be able to create in the near future a “revolutionary” change in the production planning and control methods;

(ii) notwithstanding this last consideration, these new tools, provided they are properly applied, are already able to produce appreciable results which can reasonably improve with the advance of the scientific research;

(iii) the growing complexity of PP&C problems seems however to enhance the use of these techniques, especially in relation to their possibility, which anyway has still to be investigated and understood, to solve traditional problems in a new way with respect to classical techniques.

Purpose of the Special Interest Group on *Advanced Techniques in Production Planning & Control* is to address the above issues in order to increase industrial awareness of advanced modeling techniques, to improve the understanding of the effectiveness of each technique in solving specific problems within the domain of production management; and to find new approaches to the solution of traditional PP&C problems thanks to new potentialities offered by these advanced modeling techniques. For the first time, beside the general session, a special session on Autonomous Agents was planned, to recognize the relevance of such emerging modeling technique.

We hope the papers presented at this workshop may represent a significant contribution in this direction and may serve the purpose of the industrial and scientific community.

The workshop was organized by the Department of Energy Engineering “Sergio Stecco”, of Florence University, and Department of Management, Industrial Engineering and Economics of Politecnico di Milano.

Mario Tucci
Marco Garetti

MARIA GRAZIA GNONI
RAFFAELLO IAVAGNILIO
GIORGIO MOSSA
GIOVANNI MUMMOLO

*Solving a lot sizing and scheduling problem by
hybrid modelling*

Dept. of Mechanical and Industrial Engineering
Politecnico di Bari, Italy

Abstract — The authors propose an analytic-simulation hybrid model (HM) to solve a lot sizing and scheduling problem in a multi-product/dynamic demand/single machine environment. Problem complexity is increased due to sequence dependent and relevant setup times as well as to stochastic variability of both process and setup times. Resource availability is also considered in evaluating capacity at each period of the planning horizon. The analytic model consists of a mixed integer linear programming model obtained by improving a model available in literature; it interacts with a simulation model in order to meet a production plan that allows minimizing an economic objective function. The approach tries to overcome traditional limits of both analytic and simulation models as each of them fails in jointly capturing system complexity and searching for optimal solutions. HM proposed is applied to a case study. It concerns with production of braking systems components for automotive industry. Results obtained are compared with those that could have been obtained if only the analytic model adopted in HM was used. Comparison outlines capabilities of HM in facing problem complexity as it is able to evaluate stochastic dependency among manufacturing variables; such a dependency is neglected by analytic models. Moreover, the iterative procedure adopted in HM reveals an effective tool in searching for a good production planning avoiding expensive and low effective “trial and error” procedures required by simulation to meet the same goal when a relevant number of decision manufacturing variables occurs in a production planning problem in cases of full scale industrial cases.

Keywords — production planning, hybrid modelling, lot-sizing and scheduling.

Introduction

Changeovers in flow shops manufacturing systems represent a major obstacle in pursuing high resource utilization and low inventory costs [Allahverdi et al. 1999]. In order to reduce setup costs, the number of change overs should be kept as low as possible and customer demand has to be matched by pooling orders in lots. However, the lower the number of changeovers is, the greater the holding costs are, due to high inventories. The problem is known as Lot Sizing and Scheduling Problem (LSSP). The problem is usually solved searching for lot sizes and scheduling in the planning horizon in order to meet a trade-off between setup and inventory costs. Further economic evaluations may deal with fixed production costs and backorders costs.

Traditional approaches to capacity planning in material requirement planning systems are based on three main phases aiming at reaching a feasible production plan instead of an optimal or a near optimal one. In the first phase, sizes of lots are evaluated for each product, level by level in a gozinto diagram. In this phase, lot sizes generally exceed resource capacities in some periods of the planning horizon.

Therefore, in the second phase some lots are shifted in order to satisfy resource capacity constraints at each period. Once again, the solution could be not feasible since shifts of lots can cause violations of precedence relationships. In the third phase, sequences of lots are modified to make precedence relationships satisfied. Such a stepwise procedure generally leads to a rough, though feasible, production plan since mutual dependency between lot sizing and scheduling problems is neglected and no optimisation criterion is adopted.

Research on these issues provides a wide spectrum of solutions. Solutions apply to different manufacturing problems ranging from very simple cases, referring to single level production, no capacity constraints, sequence independent setup times, stationary demands in an infinite planning horizon, to complex cases which consider multilevel, capacity constrained, sequence dependent setup times, dynamic demands in a finite planning horizon; more accurate models assume process times and sequence dependent setup times as stochastic variables. According to hypotheses adopted, several problem formulations can be defined which range from the economic lot scheduling problem (ELSP) to capacitated lot sizing problem (CLSP), discrete lot sizing and scheduling problem (DLSP), continuous setup lot sizing problem (CSLP) up to general lot sizing and scheduling problem (GLSP). Several studies have been carried out on these issues. Some significant studies are in [Aucamp 1987; Smith-Daniels & Ritzman 1988; Gopalakrishnan et al. 1995]. Excellent recent surveys are provided in [Drexel & Kimms 1997; Allahverdi et al. 1999; Meyr 2000]. Computational complexity of LSSPs can be NP-hard and destroys any hope to find optimal solutions; problem complexity justifies the relevant number of heuristics adopted [e.g. Arosio & Sianesi 1993; Ozdamar & Birbil 1998; Goncalves & Leachman 1998; Ouenniche et al. 1999].

Solutions of LSSPs are often based on analytic and simulation models. The former include mathematical programming and numerical optimisations. The latter are coded by more and more powerful simulation softwares, often user and object oriented. In the scientific literature, analytic and simulation models are generally considered as mutually exclusive methods. Analytic models search for solutions evaluating optimal values of decision variables according to a given technical-economic objective. However, solutions are generally limited in their fields of applications because of restricting hypotheses. On the other hand, simulation models are capable of accurate descriptions of system behaviours but reveals as not adequate in problem optimisation.

Integration of analytic and simulation models leads to hybrid models (HMs) which represent a challenging option as they allow to capture best capabilities of both types of models. The first taxonomy of HMs is in [Sargent & Shantikumar 1983]. Scientific research on hybrid modelling deals with design and operation of manufacturing systems. In [Mahadevan & Narendran 1994] a HM is proposed to design an AGV-based material handling system in FMS environment; a loading problem and an AGV fleet sizing problem are jointly solved in [Mummolo & Ranaldo 1996; Mummolo et al. 1999]. The design of inspection stations at assembly lines by hybrid modelling is performed in [Shin et al. 1995] while a decision support system based on the same approach is proposed in [Starr 1991]. A remark on capability of

hybrids models in solving efficiently production planning and control problems is in [Garetti & Taisch 1999]. A hybrid heuristic is proposed in [Ozdamar & Birbil, 1998] for a CLSP. Recently [Byrne & Bakir 1999], a HM is proposed to solve a Multi-Period Multi-Product (MPMP) LSSP. The problem is solved in case of multi-machine, deterministic dynamic demands. Setup times are assumed as sequence independent and deterministically known; process times are assumed as certain and no unavailability is considered in estimating resource capacity.

In this paper, the authors propose a HM for a MPMP, single machine LSSP in case of deterministic dynamic demand, stochastic variability of both process and sequence dependent large setup times; stochastic variability of failure and repair times are considered to evaluate resource capacity at each period of the planning horizon. The model is built up to tackle complexity of a case study; this concerns the production planning of braking systems components for automotive industry.

The case study

The manufacturing system investigated consists of three flow lines. Each line produces a family of similar components of braking systems for automotive industry: braking controllers (P1), by-passes valves (P2), and hydraulic actuators (P3). The process diagram is in Fig. 1.

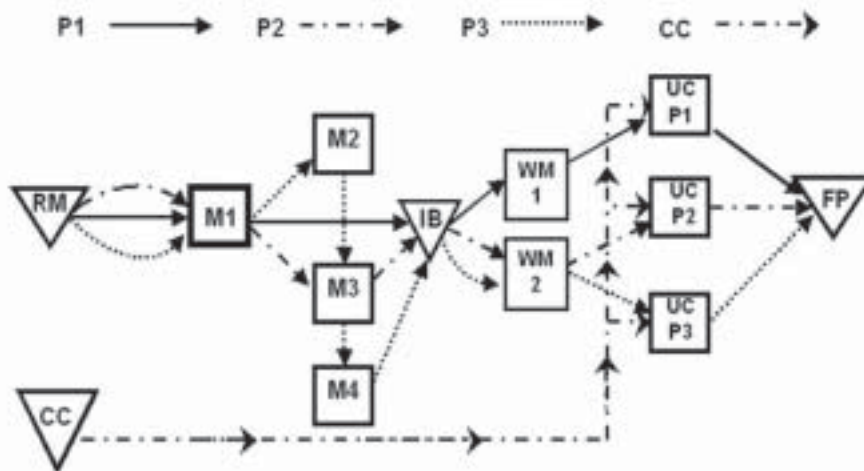


Fig. 1 – Process diagram of the case study.

The machining centres (M1, M2, M3, and M4) carry out machining operations (operations 10, 20, 30, and 40) on row parts according to routings outlined in table 1 (*next page*).

Product	Oper. 10	Oper. 20	Oper. 30	Oper. 40
P1	M1			
P2	M1	M3		
P3	M1	M2	M3	M4

Table 1 – Routings of products.

Parts wait at an inter-operational buffer (IB) and then are sent to two washing machines (WM) before reaching three U-shape assembly cells (UC-P1, UC-P2, UC-P3), one for each part family. Finishing and complementary components (CC) come from outside the system. Further inter-operational buffers, not depicted in figure 1, are located between each couple of machining centres and before assembly cells.

Manufacturing data outline the sharing of M1 among part families. Changeovers of M1 require sequence dependent and significant setup times between each couple of part families; setup times reduce M1 capacity. Setup times between parts of the same family are negligible. M1 is a critical resource of the manufacturing system as system efficiency significantly depends on production planning of such a critical resource. However, lot sizing and scheduling of M1 is a complex task due to uncertainty of both sequence dependent setup times and process times. Work data sheets and field interviews of operators suggest setup times shown in table 2; a uniform variability of setup times between P1 and P3, as well as between P2 and P3 is considered. Uncertainty on setup times of changeovers P1-P3 and P2-P3 is due to technical complexity requiring a setup crew performing changeovers; setup crew is not always available since it performs changeovers and maintenance also for other manufacturing resources in the facility. On the contrary, changeovers between P1 and P2 require less technical efforts and are performed by regular work force available on site.

From / To	P1	P2	P3
P1	-	1350	(2250 - 3150)
P2	1350	-	(2250 - 3150)
P3	(1350-2700)	(2250 - 3150)	-

Table 2 – Sequence dependent setup times [min/setup].

Problem complexity is further increased by stochastic variability of process times of M1. Probability density functions (pdf_s) of process times of M1 are defined in table 3.

Location parameter	Scale parameter	Shape parameter	Mean
0.00	1.00	0.71	1.06
0.21	1.70	1.41	2.01
1.00	1.72	1.26	2.77

Table 3 – Lognormal parameters for P1, P2 and P3 process time [min].

Observed process times data fit well (a chi-square test was adopted at 0.05 level of significance) a lognormal pdf for each part family.

Capacity of M1 is also reduced by failures and repairs. Times between failures follow a Gamma pdf with expected value of 6033 [min], shape parameter 1.41 and scale parameter 4277 [min]; times to repairs follow a negative exponential distribution with expected value of 261 [min]. Finally, backlogging is not allowed, that is unmet demands in a period can not be transferred to a next period.

The LSSP concerned is complex also due to a dynamic, though deterministic, demand (components are produced for the after-market). Nominal M1 capacity per period is given. Demands of products and nominal M1 capacity over the planning horizon are in table 4.

Period #	Demand [unit/period]			M1 capacity [h/period]
	P1	P2	P3	
1	716	0	0	255
2	5827	483	402	300
3	5504	711	856	300
4	3688	935	1590	375
5	6918	1022	1080	300
6	5735	1605	710	300
7	5686	277	1000	300
8	7174	1089	2131	225
9	6242	535	1353	300
10	5245	1697	1066	375
11	7858	565	3450	300
12	6550	2378	350	300
Total [unit]	67143	11297	13988	
Mean [u/period]	5595	941	1165	
Std. Dev. [u/period]	1867	674	920	
Std. Dev./Mean	0.334	0.716	0.790	

Table 4 – Product demands and nominal M1 capacity over the planning horizon.

Demand of P2 and P3 varies over the planning horizon more dynamically than P1 demand as shown by variation coefficient, $R = \text{standard deviation} / \text{mean}$, of product demands. Cost data are in table 5.

Cost figures	Product		
	P1	P2	P3
Setup Cost [€/h]	19.88	19.88	19.88
Holding Cost [€/unit period]	0.0098	0.0181	0.0433
Fixed cost [€/period]	458.82	458.82	458.82

Table 5 – Costs data.

Problem complexity is captured by an adequate LSSP concerned. This is the rationale for adopting the hybrid model described in the following section.

The hybrid model

The basic idea underlying the proposed hybrid model is to couple the capability of an analytic model in reaching an optimal solution of the LSSP concerned with the capability of a simulation model in describing manufacturing system evolution, in case of stochastic variability of mutual dependent manufacturing variables.

The analytic model adopted is a mixed-integer linear programming (MILP) model introduced in [Gopalakrishnan et al. 1995] subject to some modifications proposed in this paper (Modified Analytic Model-MAM) both to integrate the model with simulation and to make it able to face with problem complexity of the case study concerned. Details on MAM and modifications introduced are in the appendix of this paper. Other models could have been adopted without reduce generality in the hybrid approach proposed. The model adopted is able to find out optimal lot sizing and sequencing in a MPMP, capacity constrained, single machine problem aiming at minimizing the sum of setup, holding, and fixed costs. Large time periods justify setup carryovers since multiple products can be produced in a single period. MILP solution provides a partial lot sequence establishing the first and the last lot of a sequence at each period. However, the case study we are faced with is characterised by a mix of three products; therefore, partial sequences coincide with complete sequences of lots. Main modifications introduced in MAM allow considering setup times and resource availability at each period: setup times as well as failures and repairs times reduce M1 capacity. A simulation model (SM) provides MAM with unit setup times, number of changeovers, and resource availability at each period. SM is coded by ARENA® (rel. 3.0). MAM and SM interact by information flows according to the scheme shown in Fig. 2.

The solving procedure is iterative. The procedure starts (iteration $r=0$) by assuming in MAM an approximate sequence independent average setup times as setup number is unknown “a priori”. Further starting assumptions are nominal M1 capacities, $\{Ct\}^{(0)}$,

over the planning horizon which are not affected by failure and repair times. On the basis of initial assumptions, MAM provides SM with a first set ($r=1$) of lot sizes and sequences. On the basis of such a production plan, the manufacturing system behaviour is simulated considering stochastic variability of process, failures/repairs, and sequence dependent setup times. Technical performances obtained, $\{TP\}^{(1)}$ (e.g. lot sizes and sequences, inventory levels, number of changeovers, unit setup times, resource availability at each period) allow evaluating an objective function ($OF^{(1)}$), defined in the case study concerned as the sum of setup, holding, and fixed costs. The number of changeovers, setup times, and resource availability, evaluated at each period by SM, are more accurate estimates which can be provided to MAM for a new iteration: they represent new constants of the objective function and constraints of the MILP defined by MAM. Again, MILP solution obtained (lot sizes, sequences) is simulated and a new OF value is obtained. The iterative procedure aims at searching for a production plan which takes a minimum OF value among the r solutions, one for each iteration.

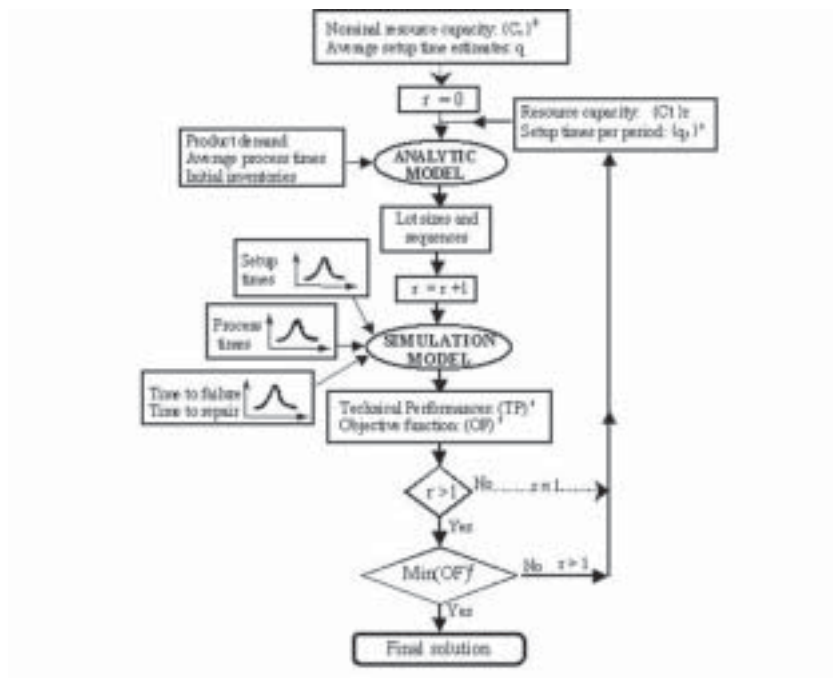


Fig. 2 – The hybrid model.

It should be stressed how iterative procedure meets convergence. If at a given iteration the corresponding production plan gives prevalence of setup costs over holding costs, then in the successive iteration the new production plan will increase lot size, allowing a reduction in the number and costs of changeovers, and causing

an increase of inventory levels and holding costs. A similar line of reasoning applies if the opposite situation (prevalence of holding costs over setup costs) occurs.

In case of significant backorders estimated by SM, OF should be increased by correspondent costs which could affect trade-off between setup and holding costs. However, the logic of iterative procedure does not change. Fixed costs could affect final solution; however, in cases tested in this paper they do not change significantly from one iteration to another one as the total number of part families assigned to periods of planning horizon is almost constant at each iteration.

It should make evident the mutual stochastic dependency between setup times and failures times when setup times are significant (in the case study developed in this paper, setup times range from about 10% to 30% of resource capacity). Failures can not occur during changeovers. Therefore, nominal probability distribution of times between failures is affected by the number of changeovers. In turn, stochastic dependency of failure times from setup times affects M1 capacity. The combined effect of setup times and failure times affects M1 capacity. Computational complexity can be tackled by hybrid modelling.

Finally, in case MAM does not provide a feasible solution at a given iteration, a new set of increased nominal resource capacity could be adopted as suggested in [Bakir & Byrne 1999]; costs of increased capacity should be introduced in OF. This is a way of using hybrid modelling also in designing manufacturing systems.

Analysis of results

Results obtained by HM proposed are in tables 6÷8. They are compared with results that could have been obtained by the analytic model (AM) proposed in [Gopalakrishnan et al. 1995] in order to assess how more realistic hypotheses on stochastic variability of manufacturing variables, introduced in HM and neglected in AM, affect technical and economic performances of production plans compared. Solutions provided by HM and AM are compared in table 6. Both solutions are feasible in meeting product demands; however, they differ both in lot sizes and in lot sequences. Standard deviation / mean ratios evaluated for lots size distribution of products outline a more dynamic distribution of lots over the planning horizon in HM solution than in AM solution.

A major cause of differences in the compared solutions is a different distribution of unit setup times [h/setup] over the planning horizon adopted. Unit setup time in AM is estimated as sequence independent, since the number of changeovers is unknown "a priori". On the contrary, sequence dependent setup times are evaluated for each period by HM. In figure 3, unit setup times (UST) is shown over the planning horizon with reference to both AM and HM. It is quite evident how the hypothesis of constant and sequence independent unit setup times assumed in AM does not meet more realistic dynamic trend and stochastic variability as evaluated by HM. At each period, mean value and 95% confidence interval estimate for mean calculated by HM allows estimating statistical reliability of setup time per period.

Solving a lot sizing and scheduling problem

Period	Lot size [unit/period]						Sequence	
	P1		P2		P3		AM	HM
	AM	HM	AM	HM	AM	HM		
1	716	716	0	0	0	0	P1	P1
2	5827	5827	483	483	402	402	P1-P2-P3	P1-P3-P2
3	7674	6482	1194	1194	856	856	P3-P2-P1	P2-P3-P1
4	9357	11178	1436	974	1590	1590	P1-P2-P3	P1-P2-P3
5	0	0	3151	3151	2993	1790	P3-P2	P3-P2
6	12295	17045	0	0	0	0	P1	P1
7	0	0	5033	0	1646	4484	P3-P2	P3
8	5379	0	0	5495	1635	0	P3-P1	P2
9	11487	11487	0	0	0	0	P1	P1
10	0	0	0	0	4866	4866	P3	P3
11	14408	14408	0	0	0	0	P1	P1
12	0	0	0	0	0	0	-	-
Mean [u/period]	5595	5595	941	941	1166	1166		
Std Dev. [u/period]	5434	6437	1605	1708	1512	1761		
StdDev/Mean	0.971	1.150	1.705	1.814	1.297	1.511		

Table 6 – Comparison of lot sizes and sequences as evaluated by HM and AM.

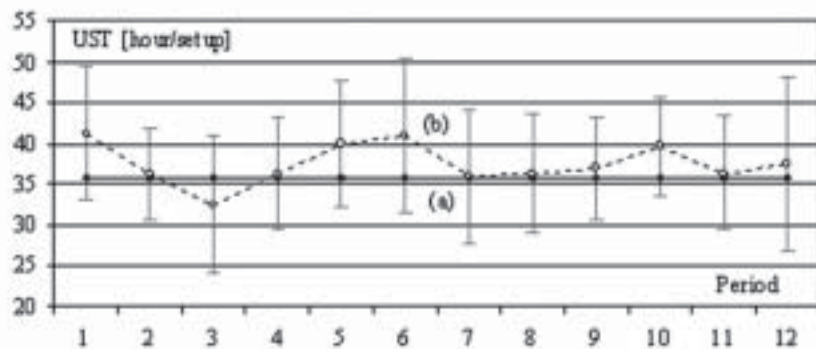


Fig. 3 – Unit setup time vs. period. (a): AM unit setup time; (b): HM mean setup time and 95% confidence interval for mean.

In table 7 (next page) a comparison of average setup times and total inventories, as well as setup and holding costs, provided by HM and AM solutions is summarized. The same table provides results obtained simulating AM solution, S(AM), that is results of the first iteration.

Solution provided by AM reveals optimistic in evaluating technical and economic performances as it neglects the effect on system performances of both stochastic variability of manufacturing variables and sequence dependency of setup times. In particular, average setup time per period evaluated by HM is of about 8.8% greater than the one provided to AM. Since setup cost per period is linear in

setup time per period, the same reduction is evaluated comparing HM and AM setup costs. HM solution evaluates significantly increase in total inventory (46.9%) and holding cost (53.1%) in comparison to the same estimations provided by AM. Since holding costs depend on total production cost of each product, being this different for each product, linearity between inventory levels and holding costs does not occur. The increase in setup costs and holding costs allows evaluating a more realistic value of the objective function by HM with reference to AM which underestimates the objective function of about 14%. Differences in fixed costs of the compared solutions are negligible. Backorders are not allowed in AM while are evaluated as negligible by HM; therefore, also differences in backorders costs are not significant.

Solutions	AM	S(AM)	HM	Diff. AM/AM (%)	Diff. S(AM)/AM (%)	Diff. HM/AM (%)
Average setup time [hour/period]	38.8	44.7	42.2	15.2	8.8	-5.6
Tot. Inventory [unit]	69958	102053	102790	45.9	46.9	0.7
Setup cost [€/year]	9268	10675	10081	15.2	8.8	-5.6
Holding cost [€/year]	1265	1939	1937	53.3	53.1	-0.1
Obj. function [€/year]	10533	12614	12018	19.8	14.1	-4.7

Table 7 – Technical-economic performances comparison: SM vs. AM, HM vs. AM, HM vs, S(AM).

As far as AM vs. S(AM) the comparison is concerned, results reveal once more how much unrealistic is AM solution since its analytical results would estimate an objective function lower than about 20% than the one calculated by results obtained by simulating AM solution.

Finally, comparing HM vs. S(AM) solutions, while no significant differences occur in inventory level and holding costs, a reduction of 5.6% in setup times and costs as well as a reduction of 4.7% in objective function is allowed by adopting HM solution. Such an economic advantage, though moderate in the case study concerned, is the effect of iterative procedure adopted in the hybrid model. In Fig. 4 the shape of the objective function vs. iteration number is depicted; the 3rd solution is considered for HM in the comparisons provided.

Computational complexity: MILP solved by HM is characterized by 360 decision variables and 756 constraints. Solutions of HM has been obtained by the iterative procedure described in “The hybrid model” section. An increase in the computational time from about 50 minutes to solve MILP by AM (a LINDO® spreadsheet solver on a PIII 500 MHz was used) to about 70 min/iteration required by HM (LINDO® spreadsheet solver plus an ARENA® module were adopted on the same PC) has been observed. The shape of the objective function vs. iteration number depends on problem data and on initial solution adopted in the iterative procedure. HM reached optimal solution within 3-4 iterations on cases tested by authors.

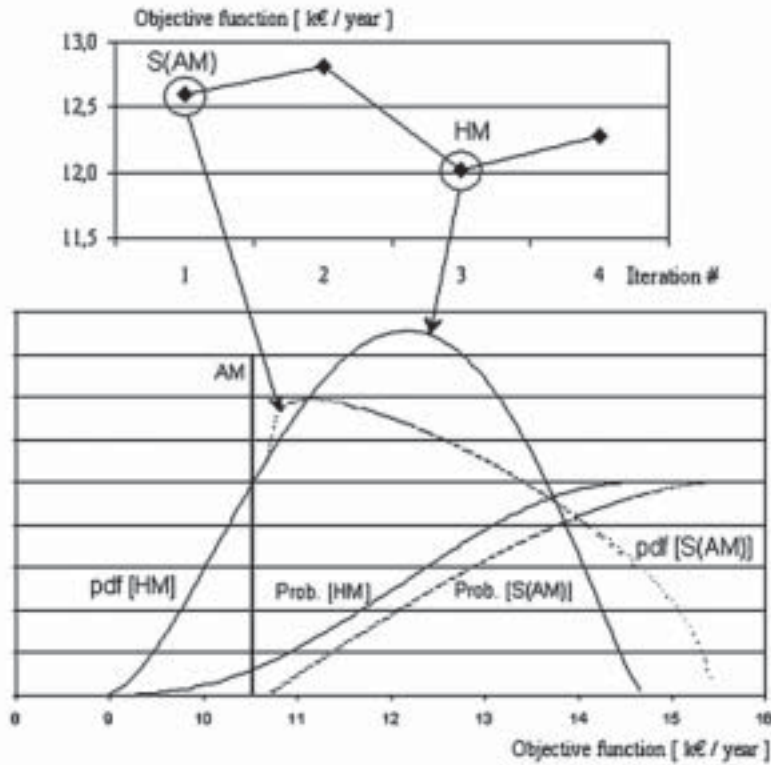


Fig. 4 – Objective function comparison: HM vs. S(AM) and AM.

Conclusions

Traditional approaches, analytic or simulation models, are limited in facing with lot sizing and scheduling problems when optimality is searched for in manufacturing environments characterised by dynamic demand, uncertainty in process and setup times, significant sequence dependent setup times, resource capacity affected by reliability performances. In such environments, simulation is ineffective as optimisation tool while analytic models are approximate as they neglect complex relationships between uncertain manufacturing variables and dynamic demands.

In this paper, the authors propose an analytic-simulation hybrid (HM) model to overcome the above-mentioned limits. Scientific research on this issue shows an increasing interest on integration of different modelling techniques as witnessed by their relevant number of applications. On one hand, this is due to peculiarities of integration in capturing capabilities of different techniques; on the other hand, logic interactions and integration between models can be implemented in interactive and integrated software modules by using wide adopted, low cost, commercial products which are able to face with no negligible problem sizes.

The hybrid model is applied to a case study concerning production of braking system components for automotive industry. After describing the production system, a critical manufacturing resource is identified and a capacity constrained, MPMP, single machine, sequence dependent setup times LSSP formulated. It is a MILP aiming at evaluating a production plan of minimum setup, fixed, and holding costs. Further dimensions of complexity of the case study consist of dynamic demands, stochastic variability of process, and setup times. Reliability performances of manufacturing system are also considered. Results obtained have been compared with results provided by an analytic model (AM) available in literature. Main limiting hypotheses of AM consist in sequence independent setup times, deterministic process and setup times, and nominal resource capacities. The hypotheses are relaxed in HM proposed in this paper.

Solution provided by AM reveals too optimistic in evaluating both technical and economic performances. Differences obtained in solutions compared outline the need for an in deep approach in solving LSSP when the problem one is faced with shows at the same time a dynamic demand and uncertainty in manufacturing variables. Classical analytic models reveal as inadequate as they manage average values of decision variables and neglect local in time effects caused by stochastic variability of manufacturing variables, especially in case of dynamic demand.

The use of HM as a resource design capacity is an interesting field of further investigations. Moreover, hybrid model capability encourages further research in searching for model formulations in case of multi-machine, multi-site manufacturing systems with dynamic demands. In such production environments the number of uncertain manufacturing variables increases as well as the need for synchronism among production sites. In this situation a still higher usefulness and pertinence of hybrid modelling is expected.

Appendix – The Modified Analytic Model (MAM) adopted in the Hybrid Model

Major modifications introduced in MAM with respect to AM [Gopalakrishnan et al. 1995] are in setup times and resource capacity. In MAM average setup times are calculated period-by-period on the basis of their sequence dependency and stochastic variability as well as on the basis of stochastic variability of process times. On the contrary, in AM unit setup times is assumed as constant in the planning horizon, as well as sequence and process times independent. Moreover, failures and repairs times of manufacturing resource considered in MAM allow evaluating reduction in resource availability; once again, such an evaluation is neglected in AM.

The MAM belongs to the class of single machine, multi-item, CLSP model.

The objective function is defined as:

$$\text{MIN } \Phi = B \sum_t N_t \times q_t^{(r)} + \sum_i \sum_t F_t \times Y_{i,t} + \sum_i \sum_t H_{i,t} \times I_{i,t} \quad (1)$$

where:

Solving a lot sizing and scheduling problem

B = average setup cost per hour [€/hour];

$I_{i,t}$ = inventory of item i at the end of period t ;

N_t = total number of setups in period t ;

F_t = fixed charge incurred whenever an item is produced in period t ;

$H_{i,t}$ = cost of holding a unit of item i in period t ;

$(q_i)^r$ = average setup times at period t , provided by SM at the r -th iteration.

The first set of constraints models inventory balance, capacity availability and the relationship between production lot size for each item and the associated binary variable (Y_{it}), as follows:

$$I_{i,t-1} + X_{i,t} - I_{i,t} = D_{i,t}, \quad \forall i, t \quad (2)$$

$$\sum_i b_i \times X_{i,t} + \sum_i (q_i)^r \times N_t \leq (A)^r \cdot C_t, \quad \forall i, t \quad (3)$$

$$X_{i,t} \leq M_{i,t} \times Y_{i,t}, \quad \forall i, t \quad (4)$$

where:

$X_{i,t}$ = the lot size quantity of item i in period t ;

$D_{i,t}$ = the demand for item i in period t ;

b_i = the capacity consumed per unit of production of item i [hours/unit];

C_t = resource capacity in period t [hours];

$(A)^r$ = resource availability provided by SM at the r -th iteration,;

$M_{i,t} = \min \{ \sum_{k=t}^T d_{i,k}, C_t \}$.

$\sum_{k=t}^T d_{i,k}$ is the cumulative demand for item i from period t to the end of the planning horizon, T .

Constraints (5) to (16) explained below refer the number of setups and the partial sequencing of products in a period. Constraints (17) to (20) model the non-negativity requirements.

$$N_t = \sum_i Y_{i,t} + \sum_i S_{i,t} + \sum_i V_{i,t} + \sum_i O_{i,t} - 1, \quad \forall t \quad (5)$$

$$S_{i,t} \quad \gamma_{i,t-1} - \alpha_{i,t}, \quad \forall i, t \quad (6)$$

$$V_{i,t} \quad \beta_{i,t} - \gamma_{i,t}, \quad \forall i, t \quad (7)$$

$$O_{i,t} \quad \gamma_{i,t} - \gamma_{i,t-1} - \omega_t, \quad \forall i, t \quad (8)$$

$$Y_{i,t} \leq \omega_t, \quad \forall i, t \quad (9)$$

$$\sum_i Y_{i,t} - 1 \leq (P - 1) \delta_t, \quad \forall t \quad (10)$$

$$\omega_t \leq \sum_i \alpha_{i,t} \leq 1, \quad \forall t \quad (11)$$

$$\omega_t \leq \sum_i \beta_{i,t} \leq 1, \quad \forall t \quad (12)$$

$$\beta_{i,t} \leq Y_{i,t}, \quad \forall i, t \quad (13)$$

$$\alpha_{i,t} \leq Y_{i,t}, \quad \forall i, t \quad (14)$$

$$\alpha_{i,t} + \beta_{i,t} \leq 2 - \delta_t, \quad \forall i, t \quad (15)$$

$$\sum_i \gamma_{i,t} = 1, \quad \forall t \quad (16)$$

$$X_{i,t}, I_{i,t}, S_{i,t}, V_{i,t}, O_{i,t} \geq 0, \quad \forall i, t \quad (17)$$

$$0 \leq \delta_t \leq 1, \quad \forall t \quad (18)$$

$$N_t, \omega_t \geq 0, \quad \forall t \quad (19)$$

$$Y_{i,t}, \alpha_{i,t}, \beta_{i,t}, \gamma_{i,t} \in \{0, 1\}, \quad \forall i, t \quad (20)$$

where:

P = Number of products

$\delta_t = \begin{cases} 1 & \text{if exactly one product is produced in period } t \\ 0 & \text{otherwise} \end{cases}$

Binary variables

$Y_{i,t} = \begin{cases} 1 & \text{if product } i \text{ is produced in period } t \\ 0 & \text{otherwise} \end{cases}$

$\alpha_{i,t} = \begin{cases} 1 & \text{if product } i \text{ is produced first in period } t \\ 0 & \text{otherwise} \end{cases}$

$\beta_{i,t} = \begin{cases} 1 & \text{if product } i \text{ is produced last in period } t \\ 0 & \text{otherwise} \end{cases}$

$\gamma_{i,t} = \begin{cases} 1 & \text{if the machine is setup for product } i \text{ at the end of period } t \\ 0 & \text{otherwise} \end{cases}$

$S_{i,t} = \begin{cases} 1 & \text{if } \gamma_{i,t-1} = 1 \text{ and } \alpha_{i,t} = 0 \\ 0 & \text{otherwise} \end{cases}$

$$V_{i,t} = \begin{cases} 1 & \text{if } \gamma_{i,t} = 0 \text{ and } \beta_{i,t} = 1 \\ 0 & \text{otherwise} \end{cases}$$
$$O_{i,t} = \begin{cases} 1 & \text{if idle period } t \text{ is used to setup product } i \\ 0 & \text{otherwise} \end{cases}$$
$$\omega_t = \begin{cases} 1 & \text{if at least one product is produced in period } t \\ 0 & \text{otherwise} \end{cases}$$

References

- Allahverdi A. Gupta J.N.D. & Aldowaisan T. (1999) *A review of scheduling research involving setup considerations*. Omega, The International Journal of Management Science, Vol. 27, 219-239.
- Arosio M. & Sianesi A. *A heuristic algorithm for master production schedule generation generation with finite capacity and sequence dependent setups*. International Journal of Production Research, 31 (3), 531-553.
- Aucamp D.C. (1987) *A lot-sizing policy for production planning with application in MRP*. International Journal of Production Research, 25 (8), 1099-1108.
- Byrne M.D. & Bakir M.A. (1999) *Production planning using a hybrid simulation-analytical approach*. Int. J. Production Economic, Vol. 59, 305-311.
- Drexel A. & Kimms A. (1997) *Lot sizing and scheduling-Survey and Extensions*, European Journal of Operational Research. Vol. 99, 221-235.
- Garetti M. Taisch M. (1999) *Advanced computing techniques in production planning and control*. Intern. Journal of Production Planning and Control, Special Issue on Advanced computing techniques in production planning and control, 10 (3).
- Goncalves J. F. & Leachman, R.C. (1998) *A hybrid heuristic and linear programming approach to multi-product machine scheduling*. European Journal of Operational Research., Vol. 110, 548-563.
- Gopalakrishnan M., Miller D.M. & Schmidt C.P. (1995) *A framework for modelling setup carryover in the capacitated lot sizing problem*. International Journal of Production Research, 33 (7), 1973-1988.
- Meyr H. (2000) *Simultaneous lotsizing and scheduling by combining local search with dual reoptimization*. European Journal of Operational Research., Vol. 120, 311-326.
- Mahadevan B. & Narendran T. T. (1994) *A hybrid modeling approach to the design of an AGV-based material handling system for an FMS*. International Journal of Production Research., 32 (9), 2015-2030.
- Mummolo G. & Ranaldo S. (1996) *Modellizzazione "ibrida" di sistemi ad automazione flessibile*. Memorie XXIII Convegno ANIMP, Venezia, 117-138.

- Mummolo G., Ranaldo S. & Iavagnilio R. (1999) *Integrating Analytic and Simulation Models for FMS Loading and AGV Fleet Sizing*. 3rd Int. Conference on Engineering Design and Automation, Vancouver, August 1-4, 813-824.
- Ozdamar L. & Birbil S.I. (1998) *Hybrid heuristic for the capacitated lot sizing and loading problem with setup times and overtime decisions*. European Journal of Operational Research, Vol. 110, 525-547.
- Ouenniche J., Boctor F. F. & Martel A. (1999) *The impact of sequencing decisions on multi-item lot sizing and scheduling in flow shops*. International Journal of Production Research, 37 (10), 2253-2270.
- Shanthikumar J. G. & Sargent R. G. (1983) *A Unifying View of Simulation/Analytic Models and Modeling*. Operations Research, 31 (6) 1030-1052.
- Shin W.S., Hart S. M. & Lee H. F. (1995) *Strategic allocation of inspection stations for a flow assembly line: a hybrid procedure*. IIE Transactions, 27 (6), 707-715.
- Smith-Daniels V.L. & Ritzman L.P. (1988) *A model for lot sizing and sequencing in process industry*. International Journal of Production Research, 26 (4), 647-674.
- Starr P. J. (1991) *Integration of simulation and analytical submodels for supporting manufacturing decisions*. International Journal of Production Research, 29 (9), 1733-1746.

PAUL VALCKENAERS
PATRICK PEETERS
HENDRIK VAN BRUSSEL
TAPIO HEIKKILÄ
MARTIN KOLLINGBAUM

*Pheromone based emergent manufacturing
control by multi-agent system*

Dept. of Mech. Engineering, Division P. M. A.
Katholieke Universiteit Leuven, Belgium

Abstract — Changes and disturbances on the shop floor require rapidly responding and easy-to-implement-and-modify control systems. Re-configurable systems and disturbance handling can be guaranteed by using multi-agent technology. However, no sound co-operation mechanism has already been developed to overcome all algorithmic shop floor control problems. This paper presents a shop floor control system for flexible flow shops, analogous to natural multi-agent systems like ants. The pheromone concept, its advantages, the integration in the shop floor control system, and first test results are discussed.

Keywords — Emergent behaviour, manufacturing control.

Introduction

This paper discusses agent-based manufacturing control where the control system design is inspired by pheromone-based interaction in ant colonies. The manuscript describes how the coordination mechanism from such a successful natural system is mapped onto the domain of manufacturing control.

The second section of this paper describes the proposed control system. It discusses the pheromone concept as it is used in natural ant systems and discusses its advantages. It also describes how this pheromone concept is translated for application in shop floor control systems. The third section addresses the results of the using of this control system on (simulation models of) two car painting shops.

Control system

This section describes the control system. It is divided into three subsections. The first subsection describes the different agent types that act in a shop floor control system. The second subsection describes how the pheromone concept is integrated in the control system. And the third subsection describes the control algorithm in more detail. The introduction to the pheromone concept and the (dis)advantages of this concept are described in the second subsection.

Agent types in manufacturing control

The different agent types, acting in a manufacturing system, are described by the PROSA reference architecture [Wyns 1999; Van Brussel et al. 1998]. PROSA describes four types of agents: Product, resource, order, and staff agents. These

agents only have a limited, specific, knowledge space and have to co-operate to achieve their goals. Their knowledge space is limited to self-monitoring, self-control, self-reflection, and to knowing and observing of their immediate neighbourhood.

The specific types of co-operation between the agent types each solve a specific problem of the manufacturing control. As the remainder of the paper only takes the manufacturing control viewpoint, only order agents and resource agents are of importance. The next paragraphs shortly introduce the order agent, the resource agent and the co-operation between them.

(i) An *order* agent is connected to each workpiece (or a group of workpieces). Its knowledge space is limited to the order information and the state of the workpiece: e.g. due date, resource on which the workpiece is proc-processed, order state model, ... But the order agent does not know anything about the other orders.

(ii) A *resource* agent is connected to each resource (e.g. machine) in the system. Its knowledge space is limited to self-monitoring and self-control. It also has a list of the resources in its direct neighbourhood. For instance, it knows the state of the resource, the load of the resource, to which order is the resource assigned, to which resource its output x is connected...

(iii) The resource agents and the order agents exchange process execution knowledge and co-operate together to complete the workpieces. Remark that although their decisions influence the global performance of the system, they only are exposed to a limited part of the overall system.

Pheromone based control

This subsection first introduces the pheromone concept and discusses its (dis)-advantages. Next, the integration of the pheromone concept in the control system is addressed.

General concept of pheromone-inspired control

The proposed emergent shop floor control system is based on the use of pheromones; such mechanism is used by several natural multi-agent systems [Dorigio et al. 1996]. A pheromone is a chemical substance that is dropped by a member of a natural species in the environment as guidance for other members.

For instance, ants use pheromones to optimize foraging. When an ant is returning from a food source, it leaves a pheromone trail in the environment, indicating the direction of the food source. When another ant, also looking for food, approaches this pheromone trail, it is attracted by the smell of the pheromone; this ant will be induced by its instinct to follow the trail because the probability to find some food in that direction seems to be higher.

When more and more ants discover the food source, the strength of the pheromone trail between the food source and the nest increases. As the strength increases, the attraction range of the trail increases also, and even more ants get attracted to the trail. When the food source is running out, no new pheromones are dropped in the environment. The pheromones that made up the trail evaporate

te (over time) and avoid that ants still get attracted to the trail. To explore new food sources, the ants sometimes take another direction than indicated by the trail. The result of this concept is a self-reinforcing guided search process to optimised solutions.

(Dis)advantages

The advantages of the pheromone concept are threefold:

- The simplicity of the co-ordination mechanism;
- The automatic guidance to optimised solutions;
- The ability to handle dynamic situations.

A simple co-ordination mechanism. The communication protocol is extremely simple. The ants do not communicate directly to each other. They do not need references to each other. But, the communication goes locally via the environment. The environment and the pheromones decouple the different ants. As a consequence, the ants only have to know how to put information on and how to get information from this environment. The environment itself becomes part of the systems knowledge base, and its complexity need not be replicated (including updating) in the brains of the ants.

Automatic guidance to optimised solutions. The depositing of global information in the environment (e.g. where to find the food) and the sense-act-reinforce actions triggered by this information guide the system to optimised solutions. In addition, exploration by the ants, who randomise their actions, prevents that the system gets stuck in a local optimum.

Ability to handle dynamic situations. The system easily reconfigures at run-time. Ants can be added and removed without affecting the coordination. Even changes in the environment will not break the coordination. The decoupling of the ants, the emergent behaviour, exploration, evaporation and feedback make that the system can adapt to changing conditions. Especially, the evaporation of the pheromones enables the system to forget information that is no longer valid.

The main *disadvantages* of the pheromone concept are the time delays and the need for tuning.

The *time delay* between locally acquiring new information and spreading this new information through the environment –overriding some older invalid information– can cause a transient time period in which the solution becomes significantly sub-optimal. This delay will be especially high when evaporation is solely responsible for forgetting invalid information (in contrast with situations where the smells of more recently deposited pheromones overwhelm the older invalid ones).

Tuning is another important aspect. The solution and the time to get to the solution depends heavily on the amount of ants that are used, the evaporation constant, the influence range of the pheromones, feedback factors...

This pheromone-based concept has a lot of potential. It has already proven to be able to solve different types of problems [Dorigio 1998], including scheduling [Snyers 1998; Federspiel 1999] and load balancing in telecommunications networks [Snyers 1998]. It is especially useful when the problem can be described as a shortest path problem in a graph.

Nonetheless, it is worthwhile to keep in mind that the ants' coordination design relies on a number of assumptions, which are not necessarily true in manufacturing control (e.g. the system assumes that losing a couple of ants is no problem). Deeper understanding of why pheromone-based control performs well in certain areas is required for the successful application in manufacturing control: *Modern aircraft do not flap their wings like the birds.*

Pheromones in a shop floor control system

This section describes how the pheromone concept is translated for shop floor control purposes. Fig. 1 shows the mapping of the different objects of the ant system onto the objects of the control system.

In the ant system, the ant plays four roles as indicated in Fig. 1. First, it is a *problem solver*: the ant has to find the food and bring it to the nest. Second, it is an *information observer*: it observes the existence of 'food'-pheromones in the environment. Third, it is also an *information creator*: after the ant found the food, it initialises the spreading of the 'food'-pheromone. And fourth, it is also an information spreader: while the ant is returning to the nest, it continuously drops the 'food'-pheromone in the environment.

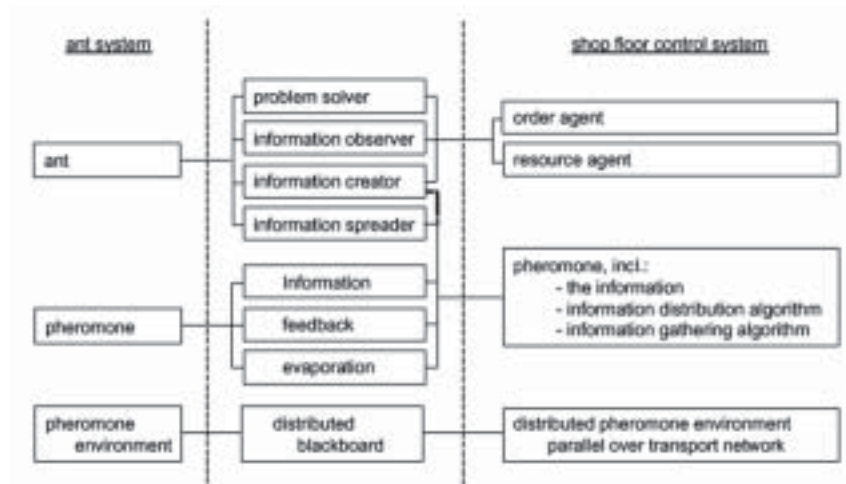


Fig. 1 – Mapping of the natural system's objects on the shop floor control system's objects.

The decision taking entities in the shop floor control system –the order and the resource agents– take up the first three roles of the ant, i.e. the problem solver, the information observer and the information creator. They are the problem solvers because they are responsible to finish the workpieces. To make globally more performant decisions, they also observe and take up additional information out of the environment. Remark that other (order and resource) agents created this

additional information. And finally, dependent on the impact of the agents' decision, they create positive or negative feedback to stimulate or prevent other agents to incorporate the same information in their problem solving process.

The role of information spreader is integrated within the information itself and constitutes a new object, called the (control system's) *pheromone object*. The role of information spreader could not be performed also by the order and resource agent as in the ant system. This is due to the difference in propagation direction of the agent and of the information that the agent has sent out. In the ant system, both propagation directions correspond to the walking direction of the ant itself. In shop floor systems, this is different. The resources are very often static, while information has to be propagated. And the order agents trace their workpiece downstream, while very often information, including the feedback, has to be propagated upstream.

Like in the ant system's pheromones, no constraints are put on the richness/complexity of the information object. However, the information object must be able to incorporate and represent (positive/negative) feedback on its information value. And, the information object must evaporate when its value becomes outdated.

A new feature, the *information gathering algorithm*, is integrated in the (control system's) pheromone object. This algorithm describes how the information has to be modified during its propagation. This object enriches the traditional static view of information with the intelligence to modify itself according to the situation. This algorithm —the change of information— can also trigger the creation of a new pheromone, and as such acts as an information creator. The information-creating object defines both the information and information gathering algorithm.

The *information distribution algorithm* describes the information spreading. To avoid that the pheromone is propagated all over the environment, stop criteria have to be built in. For example, these stop criteria can be related to the type of resource at which the pheromone has arrived.

The environment in which the (control system's) pheromones propagate is illustrated in Fig. 2 (*next page*). It is modelled as a separate network of local pheromone environments in parallel over the physical transport network – the transport network should be interpreted as the connected network of all resources. This includes the resources that only move workpieces, as well as the resources that process the workpieces. This pheromone environment can be interpreted as a distributed blackboard consisting of connected separate local blackboards. A local pheromone environment is connected to each resource body and to each input and output port of the resource. These additional local pheromone environments at the input and output ports of the resources are necessary, because very often the value of the propagated information depends on the input or output via which the pheromone was propagated. In each local pheromone environment, pheromones can be stored. A stored pheromone is only accessible/observable via the local pheromone environment in which it was stored. The connections of the resources in a flexible flow shop make this construction straightforward.

Within this construction, all decision-taking agents have access to the pheromone environment. All resource agents have (only) access to their resource's local phero-

mone environments. And the order agents have (only) access to the local pheromone environments of the resource on which their workpiece is processed. When the order and resource agents have to negotiate and take a decision, they can enrich their local knowledge with the available information in the resource's local pheromone environments.

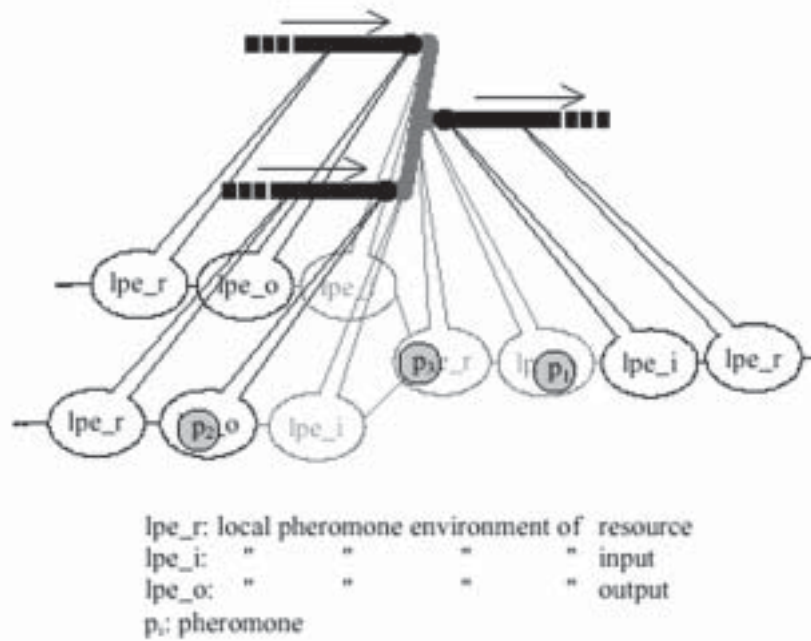


Fig. 2 – The shop floor control system's distributed.

The pheromone environment can easily be extended if more decision taking agents have to be integrated in the shop floor control system, e.g. a central high level planning system, the maintenance department, the design department...

Pheromone based control algorithm

The pheromone based shop floor control algorithm is designed in a bottom-up way as a distributed 'vertical layered architecture with one pass control' [Wooldridge 1999]. Both aspects, the bottom-up design approach and the 'vertical layered architecture with one pass control', are expressed in Fig. 3 and explained in the next paragraphs.

The bottom-up design approach starts from the individual abilities of the system's transport and processing resources and adds constraints to their individual behaviours to increase the overall system's performance. Each constraint is modelled in a separate layer and added on the previous one. The constraints are categorised under the

following three types: the hard constraints, the optimising constraints and the meta-constraints:

- The *hard constraints* keep the system in a feasible state. For example, never transport a workpiece in a direction in which it cannot be finished or can cause deadlock. As illustrated in fig. 3, feasibility layers model these hard constraints.
- The optimising constraints improve the overall system's performance and do not take the feasibility aspect into account. The optimising layers model these optimising constraints. For example, as illustrated in fig. 3, in a system of which the performance is measured by throughput, the OPT constraints [Goldratt & Cox 1992] can be modelled. The due date, for ex-ample, is another constraint.
- The meta-constraints put constraints on the parameters that are used in the optimising layers. For example the drum-buffer-robe parameters from the previously mentioned OPT example can be tuned b these layers. As illustrated in Fig. 3, the tuning layers model these meta-constraints.

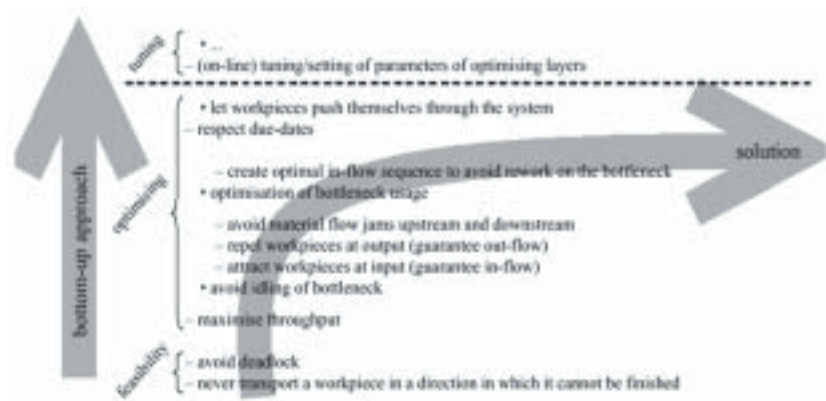


Fig. 3 – Bottom-up design approach of the pheromone based control system's vertical layered architecture with one pass control.

The execution of an algorithm that has been built according to a '*vertical layered architecture with one pass control*', starts at the bottom layer and works its way up to the top layer, reducing the search space in each layer. The top layer, finally, provides the solution. In the proposed pheromone based control system, only the feasibility layers reduce the search space, while the optimising layers act in group as the top layer, providing the solution. Only the feasibility layers reduce the search space, because these layers model the hard constraints. These have to be followed; otherwise the system can get blocked in an unfeasible state. The optimising layers only improve the performance of the whole system and do not act as real constraints but rather state facts over performance-related matters. The optimising layers are sub-layers –sub-algorithms– of the algorithm's top layer. The outcome of this top layer and as such the final solution of the algorithm is a choice that is made based on the facts stated by these sub-algorithms.

The testbed

A flexible flow shop of industrial scale is used to evaluate the new control system. The shop has the following characteristics:

- Throughput is the major performance measure in the shop. Note that makespan of a limited set of production orders is not a relevant measure for throughput in a system that keeps on producing indefinitely.
- Only one product type is produced. However, the characteristics (colour, model...) of the product instances affect the product flow.
- Multiple processing machines can be selected for each processing step. The subsets of these machines that can really be used depend on the product instance's characteristics.
- Due to the uncertainty of the processing result, the next processing step can only be determined after the processing itself. This limits the predictability of the system. And it also causes feedback loops in the product flow.
- Due to the transport flexibility, several routes can be taken to go from one processing unit to another. However, not all routes are possible.
- The order of magnitude of work-in-process is twice as high as the number of processing machines.
- The cycle time of the system is of the order of minutes.

The next sections describe this testbed set-up, the control and the results.

Set up

Two different emulation models of industrial car painting shops have been (and still are) used to evaluate the proposed control system. The first model is an emulation of a DaimlerChrysler paint shop (Mascada 1997). Fig. 4 illustrates the size of this model. It shows one of the floors of the shop. The second model does not correspond to a real shop, but to a virtual paint shop with the same capacity as DaimlerChrysler's. This second emulation has been built with the capabilities of the new control system in mind. It is used to show the real advantages of the new control system.

Both emulation models are very detailed. They imitate the plant down to the level of PLCs that decide at each crossing to where to send a car. Also the dependency between the painting result and the trio "colour, number in batch, car type", which determines the throughput, has been taken into account.

The performance of paint shops is measured by throughput. This corresponds to the throughput at the painting lines. The yield of the painting lines, which also determines the re-flow and as such the throughput, depends on the trio: colour, number in batch and car type. This yield can only be influenced by the control system via the second parameter, the number in batch. The relation between the yield and the position in the batch is as follow: for the first cars in a batch, the yield increases as the position in the batch increases. However, after a few positions, the effect is gone. To create large batches –to improve the yield– most of the plants have a large sorting buffer in front of their painting lines.

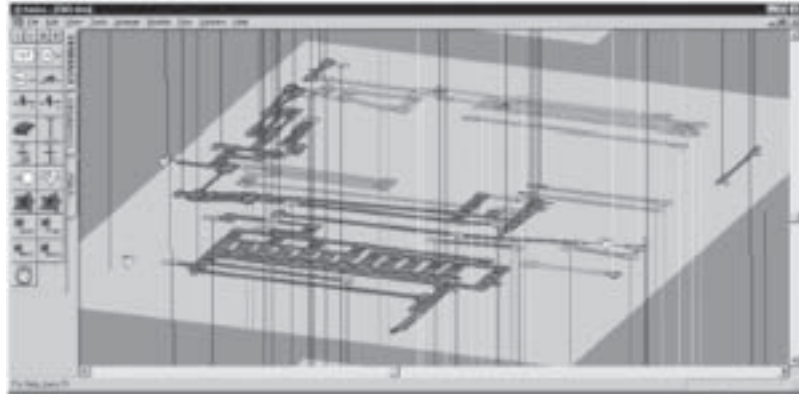


Fig. 4 – View of one floor of the emulation model of the DaimlerChrysler paint shop.

Other throughput losses are caused by colour breakdowns on the painting lines and blockages of crossings upstream or downstream the bottleneck. Colour breakdowns cause a lot of repair and rework on the painting lines. Blocked crossings upstream or downstream cause idle time on the painting lines.

The implementations of the emulation models and the connection to the control system follow the methodology described by [Peeters et al. 1999]. The control system has been implemented in JAVA, JDK-1.2. and runs on a WinNT4.0 PIII-400Mhz PC. The emulation models have been implemented in ARENA-RealTime (Arena) and run also on a WinNT4.0 PIII-400Mhz PC. The control system and the emulation model are connected via one TCP/IP socket.

Implemented Control Layers

At the moment, the main focus is on the control layers that are used at the crossings: which layers/algorithms are needed and which information has to be propagated. The control layers that are needed at the other decision points, e.g. the sorting buffer or the painting lines, have been kept as simple as possible. So far, the control system behaves like a push system: cars are pushed in the system, all resources (also the crossings) process the cars more or less directly according to the First-In-First-Out rule, and the sorting buffer decides autonomously when to push the cars back further down the system.

This control system has been coupled to both emulation models. The remainder of this section lists the already implemented control layers.

The implemented control layers for the crossings include two feasibility layers and six optimising layers. All layers are listed in the remainder of this section.

The following two feasibility layers keep the system's state feasible:

- 'Never transport a work piece in a direction in which it cannot be finished' to avoid at the crossings that cars are switched to outputs after which they cannot be finished anymore.

- ‘Avoid deadlock.’ Because there exist loops in the transport system, deadlock avoidance measures are required.

Six optimising layers have been implemented to avoid idle time and rework on the painting lines and, as a secondary aspect, to finish the car before its due date:

- ‘Avoid blockage of this crossing’ to avoid idling of the upstream or downstream painting lines.

- ‘Increase throughput at crossing’ to avoid that this crossing becomes the bottleneck instead of processing stations. As this is only important when the crossing tends to become the bottleneck, *the weight factor of this rule will vary according to the load of the crossing.*

- ‘Attraction to sorting buffer’ to detour the cars that are heading for the painting lines via the sorting buffer. Although the sorting buffer is the main batch building mechanism, this processing step cannot be put in the process plan because not all car flows that are heading for the painting lines can physically pass the sorting buffer. As the ‘crossing’ agent does not have any lay-out information in its local knowledge space, the sorting buffer has to send out a pheromone downstream, indicating that the sorting buffer can be reached via this way.

- ‘Create batches at outputs’ to create, at the crossing’s outputs, colour batches of cars that are heading for a painting line. This layer set preference for the crossing’s outputs of which the last car that was switched via this output and that was heading for a painting line had to be painted in the same colour. *The increase of batch size represents the self-reinforcing behaviour,* because more cars that have to be painted in that colour will be stimulated to take the same output.

- ‘Avoid loops’ to reduce the possibility that a car is switched to an output that can result, due to the topology of the transport system, in a loop. It prevents that the available slack till the due date is reduced or exceeded without any reason.

- ‘Avoid already chosen outputs’. This layer has the same functionality as the previous one. However, the previous layer only takes static look-a-head information into account, while this layer takes dynamic historical information into account.

Test Results

The common interface, the layers and the emergence in the control system make the system easy to implement and modify or maintain. Only after a 5 minutes re-tuning of the control system, the same implementation of the control system ran on the second simulation model, which has a completely different topology. No re-implementation of any algorithm was necessary, although the layouts of the shops are different. This shows that in a limited amount of time, a lot of functionality can be implemented. You get it all for free, due to the emergence in the control system. If a library of algorithms and information structures exists, covering different shop floor characteristics and performance measures, the implementation of the control system would only be a matter of tuning. The current implementation is too immature to generate significant performance figures. Current and future work is focusing on designing and implementing a high-performance control in which system-specific tuning will be compared with generic control designs.

Conclusions

Pheromone based control has the potential to solve the control problem for flexible flow shops. The main advantages of the pheromone concept are: (i) a simple coordination mechanism, (ii) the automatic 'guidance' to the 'optimised' solution, and (iii) the ability to handle dynamic situations. However the concept has also some disadvantages: (i) time delays, and (ii) tuning.

The control system is tested on two simulation models of car painting shops. Preliminary testing showed that the same implementation of the control algorithm could be used for both simulation models. Only some small retuning was necessary to improve the system's performance again..

Acknowledgement

This paper presents research results obtained through work sponsored by the European Community (ESPRIT LTR Project Mascada, ESPRIT Working Group IMS-WG). All Mascada partners –Katholieke Universiteit Leuven, AISystems, Daimler-Benz, University of Cambridge and VTT Automation– contributed to this work and assume the scientific responsibility.

References

- Arena. Systems Modeling Corporation, <http://www.sm.com>
- Dorigio M., Maniezzo V. & Colorni A. (1996) The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26 (1), 29-41.
- Dorigio M. (1998) ANT COLONY OPTIMIZATION. <http://iridia.ulb.ac.be/~mdorigio/ACO/ACO.html>
- Federspiel F. (1999) Complexity Solutions for Manufacturing and Logistics. In: *Proceedings of the 7 th Annual Santa Fe Chaos in Manufacturing Conference*, Santa Fe.
- Goldratt E.M. & Cox Jeff (1992) *The goal: a process of ongoing improvement*. North River Press Croton-on- Hudson (N.Y.), ISBN 0-88427-061-0.
- Mascada (1997) *Manufacturing Control Systems Capable of Managing Production Change and Disturbances – An Application of Autonomous Cooperating Agents*. Esprit LTR Project 22728, <http://www.mech.kuleuven.ac.be/pma/project/mascada/welcome.html>
- Peeters P., Van Brussel H., Valckenaers P. & Wyns J. (1999) Methodology to integrate simulation in the development phase of shop floor systems. In: *Flexible Automation and Intelligent Manufacturing 1999 (FAIM99)*, Ashayeri J., Sullivan W.G., M.M. Ahmed (Eds), 761-771, Begell House Inc, ISBN 1-56700-133-5.
- Snyers D. (1998) *Cooperative Agents for Dynamic Routing in Communication Networks and Job Shop Scheduling*.

- Van Brussel H., Wyns J., Valckenaers P., Bongaerts L., & Peeters P. (1998) Reference Architecture for Holonic Manufacturing Systems. PROSA. Computers In Industry, special issue on intelligent manufacturing systems, 37 (3), 255-276.
- Wooldridge M. (1999) Layered Architectures for Intelligent Agents. In: Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, G. Weiss (Ed.), 61-65, MIT Press, ISBN 0-262-23203-0.
- Wyns J. (1999) Reference architecture for Holonic Manufacturing Systems-the key to support evolution and reconfiguration. Ph.D. Dissertation, K.U.Leuven, PMA Division.

Lufthansa Technik Logistik GmbH

Abstract — An advanced backlog and utilisation controller for automatic production control will be presented in this paper. The controller adjusts the capacity of individual work systems in order to eliminate the backlog as soon as possible under consideration of the mean smoothen input rate of the system. Because of that, the quality of the backlog control process can be improved significantly. In case of an under-load situation, the controller reduces the capacity in order to guarantee the planned utilisation regarding the required performance. Within a case study logistical rationalisation potentials opened up by this controller concept are shown impressively. The objective is to improve automatic production control (APC) with defined control and reference variables based on the logistical objectives.

Keywords — backlog control, automatic production control, production logistics, flexible capacities, feedback control.

Introduction

The permanent changes regarding the competitive conditions compel companies to intensify their hunt for innovative and dominating competitive strategies. Well-known strategies as *cost-and quality leadership*, *product differentiation* and *service orientation* do not appeal to the customers anymore due to a balance between the competitors. Contrary, strategies which focus on flexible production, high delivery capability, large product variety, enhanced innovation rate and agility are mainly based on the “forth dimension of competition”, the resource *time* [Stalk 1988; Warnecke 1993; Oetinger 1995; Milberg 1997]. Mason-Jones, Naylor and Towill formulated graphically in 1999 that ‘getting the right product, at the right price, (*and*) at the right time to the customer is not only the lynch pin to competitive success but also the key to survival’, regardless whether there existing uncertainties in the demand. Agile manufacturing –as a response to that– has been defined as the capability to react effectively to changes in the marketplace in a cost effective way, simultaneously prospering from uncertainty [Helo 1999].

The increasing import of the production resource time requires a change in production: A change from static to dynamic systems [Warnecke 1993]. Forrester has described the effects of changing customer demands on the production rate of companies within a supply chain as depicted in Fig. 1 (*next page*).

Due to uncertain demand forecasts and delays the dynamics in supply chains are magnified. This effect is commonly termed as the ‘Bullwhip effect’ [Forrester 1961]. Various approaches try to optimise the supply chain as a whole in order to balance the dynamic effects by means of more certain upstream demands. However, uncertainty is impossible to remove from supply chains completely and with that also from each link of the chain [Mason-Jones et al. 1999]. New approaches for production planning and control are still necessary to manage the dynamics within

the company. A new method under the catchword *automatic production control* (APC) which is able to react agile to changing customer demands has been presented in recent publications (Wiendahl & Breithaupt 1998; Breithaupt 1999; Wiendahl HP & Breithaupt JW 2000). Within this paper an advanced backlog and utilisation control

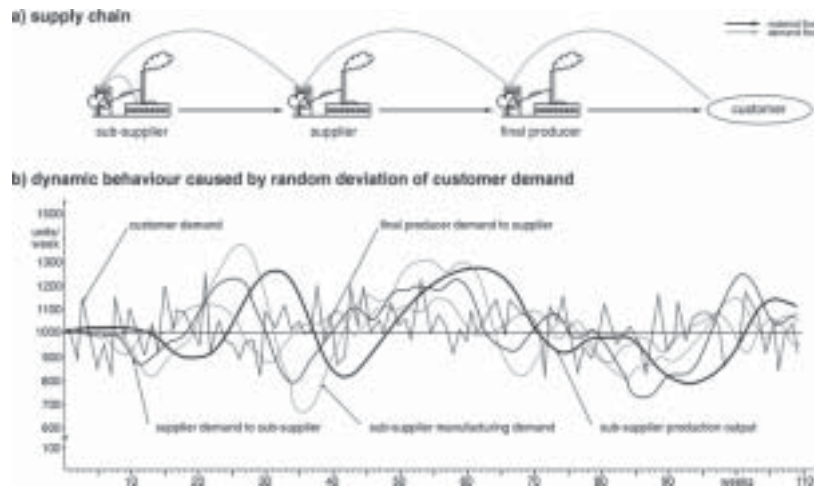


Fig. 1 – Dynamic behaviour within supply chains.

including a detailed description of it's functionality shall be pointed out. The concept is based on the control theoretical model described in the publications mentioned above.

The continuous job shop model

A first step towards automatic production control has to be made by developing a continuous job shop model, because control theory offers much more methods for continuous than for discrete models. In 1996 Petermann has published a simple continuous model for a single work system based on the funnel model and the logistic operating curve. Since then, an extended model for modelling several work systems connected via the material flow has been developed. This model forms the basis of the controller concept described in the following. A detailed derivation of the model can be found in [Wiendahl & Breithaupt 1998].

The input and output variables as well as a simple control loop of the continuous model are shown in Fig. 2 (next page). The flow of incoming orders measured in number of orders per unit of time is converted into the work-content-related input rate by multiplication with the mean order time. The input rate will be integrated over a time interval into the cumulated input of the work system. The same can be done for the output side of the work system. Analogous, the mean performance is

also integrated over a time interval into the output of the system. Similar to the input rate, the output of the work system is converted into the output rate by dividing the mean performance through the mean order time. These transformations are required because the material flow between two work systems is measured in number of orders per unit of time, whereas within the work system the work content is of interest. In reality the order time of several orders processed by a specific work system differ. Therefore, the transformation leads to practicable results only over time periods long enough to enable the different order times to balance. This can be mentioned on the planning level.

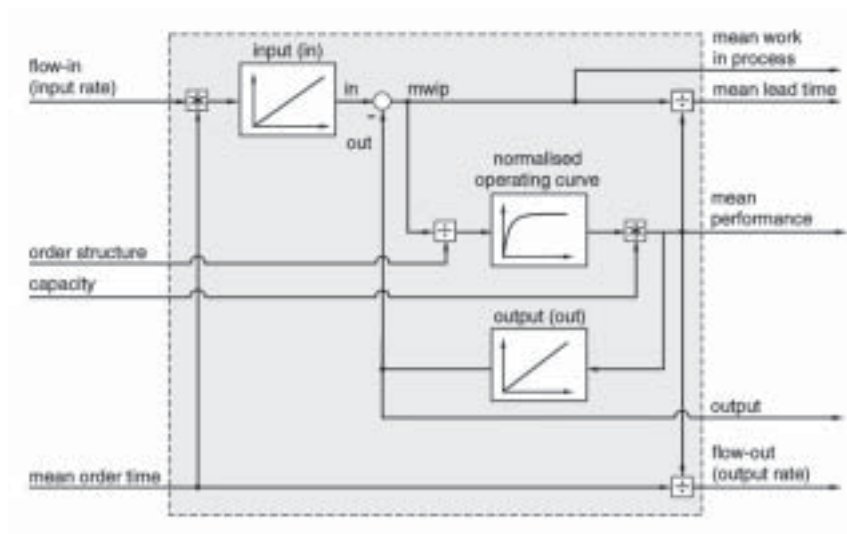


Fig. 2 – Continuous model of a single work system.

The mean work-in-process can be determined by subtracting input and output of the work system. Several former tries to develop a control theory based production control failed, because the transmission function between the input and output variables was missing. This gap can be filled by means of the logistic operating curve, which forms the dependencies between work-in-process (input variable) and the performance (output variable) of the work system. Detailed derivations of this curve are published in [Nyhuis & Wiendahl 1999]. In order to be independent to varying capacities the normalised version of the curve has been implemented where the performance is substituted by the utilisation of the system.

After the definition of the basic work system model which is principally suitable for connecting several work systems via the material flow, the connection itself is still missing. Therefore, different types of connections has been examined and integrated into a job shop model based on control theory [Wiendahl & Breithaupt 1998]. Most of them are dealing with transition probabilities to distribute the output of upstream work systems to the following downstream systems. The n:n-connection

is the most universal one because every work system can theoretically deliver to each other. If there is no transition between two work systems the transition probability is set to zero. Even transitions from a single work system to itself are possible. Therefore, this type of connection shall be integrated into the job shop model.

As mentioned above, the definition of the transition probabilities is essential for the generation of the job shop model. There are two ways to determine transition probabilities. For plant layout tasks and master production scheduling the determination is based on the prospected production schedule and related routing plans. Alternatively, for the medium range planning feedback data from the job shop is suitable. Both ways follows the same procedure in principle (Fig. 3). Starting from

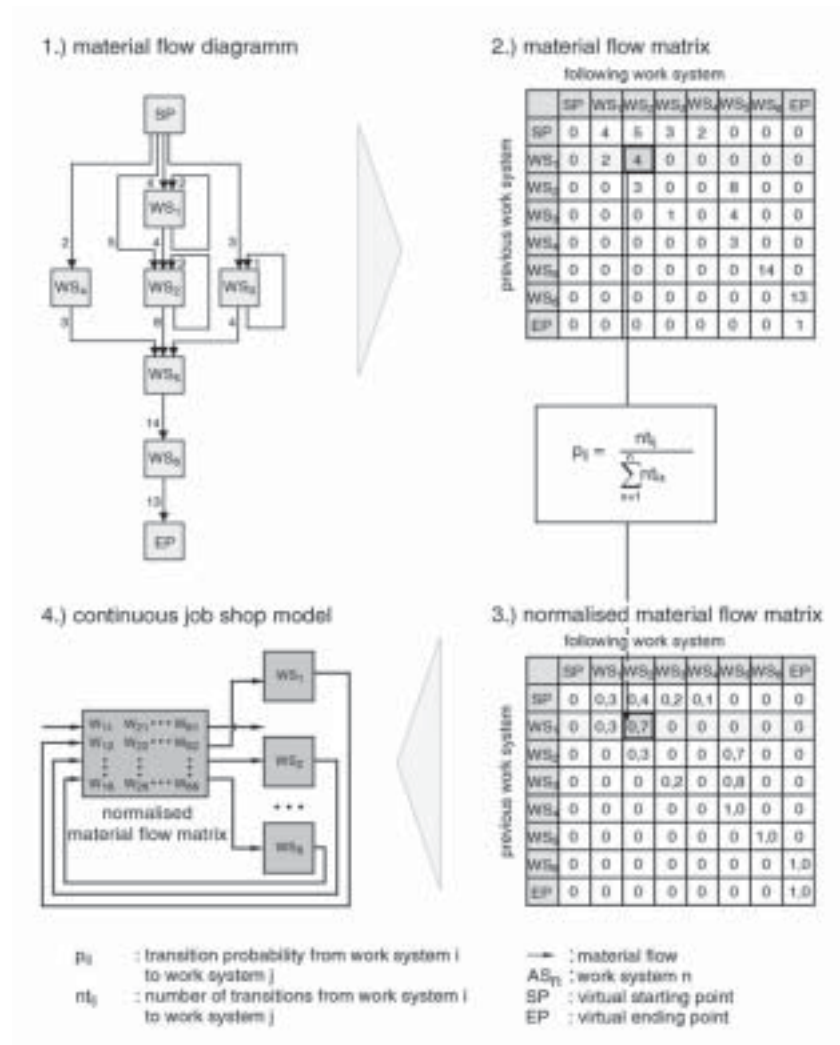


Fig. 3 – From material flow diagram to continuous job shop model.

a material flow diagram a common material flow matrix can be determined. The matrix states how many orders have been transported from each work system to each other within the viewed job shop during a certain period of time. The virtual starting and ending points are defined to describe the flow of incoming and outgoing orders to the viewed job shop.

For the calculation of the transition probabilities not the absolute number of orders flowing from a previous work system to a following is of interest, but rather the percentage relation of the system's output as mentioned above. Therefore, a normalisation of the material flow in relation to the summarised output of each work system is imperative. The normalised material flow matrix serves as the transition probability matrix within the job shop model.

An advanced, distributed backlog-and utilisation controller

The controller concept

On the basis of the model presented an advanced distributed *backlog-and utilisation-controller* has been modelled. Fig. 4 shows the concept of the controller.

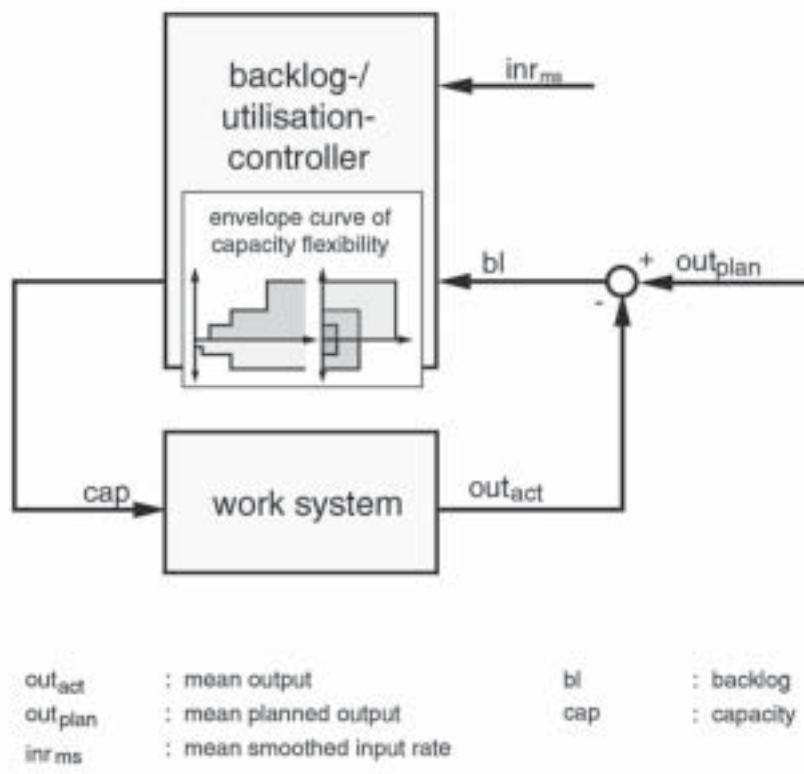


Fig. 4 – Concept of an automatic backlog-and utilisation controller.

The concept is similar to the backlog controller presented in [Wiendahl & Breithaupt 1998; Breithaupt 1999; Wiendahl & Breithaupt 2000]. The main difference exists in the consideration of the *mean smoothen input rate* of the work system to be controlled and in the implementation of utilisation control.

The capacity is used here as a correcting variable of the system. The planned output is the reference variable. The difference between the actual and the planned cumulated output results in the above mentioned backlog. The controller adjusts the required capacity of the work system to reduce the backlog to zero as fast as possible under consideration of the prospective input of the system. Because in reality it is impossible to adjust the capacity immediately, a reaction time between the request for capacity and the following allocation has to be introduced. In an existing job shop the system's capacity can normally be increased or decreased in different sized steps. For each step a specific reaction time is required. The fastness in that an increase or a decrease can be realised and the amount of changed capacity is a measure for the system's capacity flexibility. The different steps to increase or decrease the system's capacity under consideration of their individual reaction time are depicted in the so-called envelope curve of capacity flexibility (Fig. 4). A detailed description of the envelope curve can be found in [Breithaupt 1999, 2000; Wiendahl & Breithaupt 2000].

The procedure of capacity planning

The procedure of capacity planning shall be described in the following. At the beginning of each planning period (planning time) a new capacity planning is carried out (Fig. 5). Therefore a *data collection* is necessary which is based on the output data of the previous period and the input data of the actual period. At first, all the relevant parameters will be determined and the actual capacity profiles of the work systems be extracted. The *actual backlog* and the *actual work-in-process* are well-known parameters and do not need any further explanation. The *mean smoothen input rate*, measured in hours of work content per shop calendar day (scd), can be determined based on the input rate of the controlled work system with the aid of a control theoretical proportional block with first order delay. The *ideal capacity* is the required capacity to obtain the planned utilisation on the basis of the actual system's performance (Eq. 1):

$$CAP_{ideal} = \frac{PER_m}{UT_{plan}} \quad (1)$$

with:

CAP_{ideal} = ideal capacity [h/scd]

PER_m = mean performance [h/scd]

UT_{plan} = planned utilisation [-]

This parameter is important for utilisation control only. The definition of a *wip-limit* serves to prevent the backlog controller from allocating additional capacity due to an existing backlog even if there is no work to be processed. If the work systems work-in-process falls short of this limit the backlog control procedure will be switched off. This is imperative in the case of upstream bottleneck systems. Then, planned work can not be finished at the viewed work system because the work is still waiting for processing at the bottleneck systems and therefore a backlog is arising even there are no orders waiting. In this case the standard backlog controller would now allocate additional capacity in order to eliminate the backlog and, therefore, losses in utilisation occurred immediately.

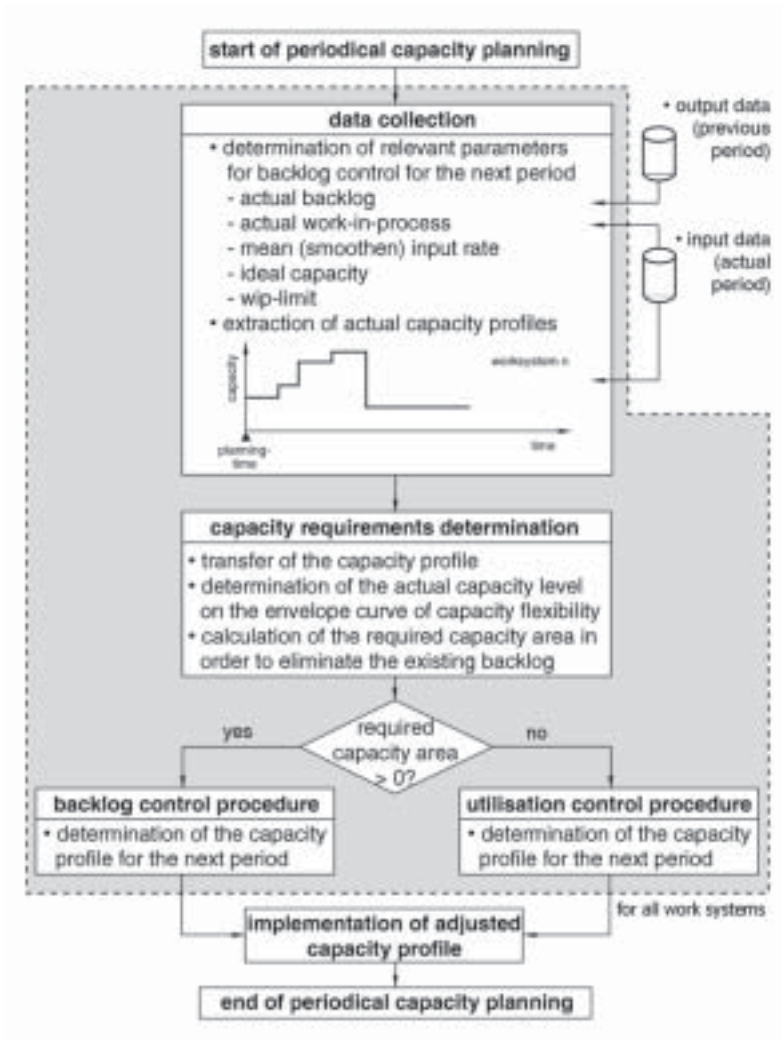


Fig. 5 – Procedure of capacity planning.

After the data collection the *capacity requirements determination* is carried out. At first, the capacity profiles of the individual work systems are transferred into the planning routine. Afterwards, the actual capacity level on the envelope curve of capacity flexibility is determined. Finally, the required capacity area to eliminate the existing backlog is calculated. In case of a positive area which means that additional capacity is required the *backlog control procedure* is carried out. Otherwise, a capacity reduction is necessary in order to avoid losses in utilisation. This is the task of the *utilisation control procedure*.

Both, *backlog control* and *utilisation control procedure* result in an adjusted capacity profile for the following periods which is implemented in the input data of the next simulation period.

The procedure of backlog control

Figs 6 & 7 depict the backlog control procedure mentioned above. The procedure starts with a determination of the state of capacity adjustment. Three different states have to be distinguished: *standard (planned) capacity*, *additional capacity*, or *reduced capacity* installed.

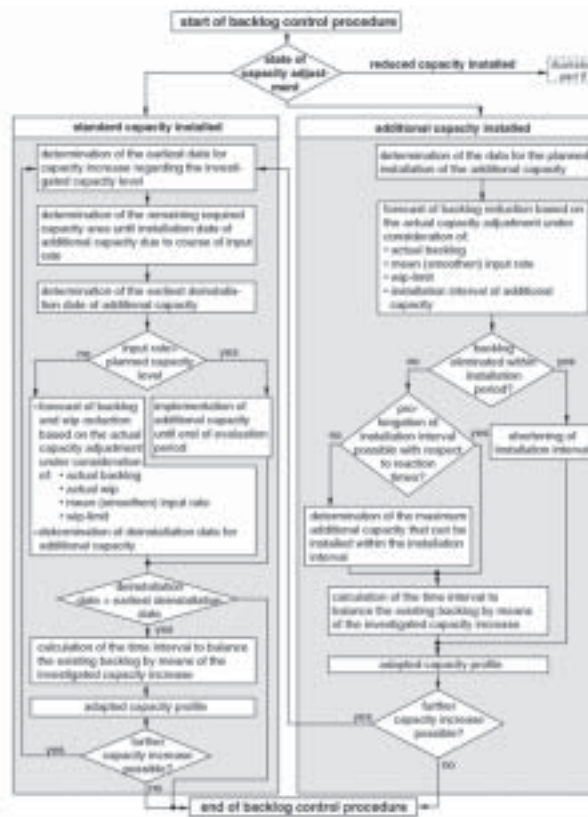


Fig. 6 – Procedure of backlog control (part I).

So far as the standard capacity is still installed the earliest date for installation of the first level of capacity increase has to be determined (Fig. 6, left side). Secondly, the prospectively required capacity area at installation date of the additional capacity is calculated under consideration of the mean smoothen input rate. This is of importance because the difference between the actual capacity and the input rate influences the course of the backlog significantly. Furthermore, the earliest date of de-installation of the examined capacity level is calculated on the basis of its *minimum installation time*. In a following step a check is made whether the input rate will still exceed the planned capacity level. If so, the additional capacity has to be allocated until the end of the evaluation period in order to minimise the unavoidable increase of backlog. On the other hand, the expected date for elimination of the backlog as well as the date for falling short of the wip-limit are determined under consideration of the *actual backlog*, the *actual wip*, the *mean smoothen input rate*, and the *wip-limit*. The date for the de-installation of the additional capacity depends on the earliest date of both mentioned above. So far as the prospective de-installation date takes place before the date of the earliest possible date of de-installation, the backlog control procedure will be closed immediately without adapting the capacity profile. Otherwise, the additional capacity will be planned which results in the adapted capacity profile. If there are further levels on the envelope curve available the whole procedure will be repeated.

The second possible state of capacity adjustment is an already realised capacity increase (Fig. 6, right side). In this case, the planned date for returning to standard capacity will be determined firstly. Furthermore, the prospective course of the backlog will be calculated-similar to the procedure regarding the standard capacity-on the basis of the actual capacity adjustment. With it, the *actual backlog*, the *actual wip*, the *mean smoothen input rate*, the *wip-limit*, and additionally the *installation interval* will be considered. By means of this calculation one can determine whether the backlog can be eliminated successfully within the installation interval. If so, the planned date for returning to standard capacity will be brought forward on the prospective date of complete elimination of the backlog. The procedure continues with the preparation of an adjusted capacity profile. In the case, that a complete reduction of the backlog can not be realised within the installation interval, a check is made whether a prolongation of the installation interval is possible with respect to the reaction time of the viewed capacity level. If a prolongation is not possible the highest capacity level will be selected that can be installed directly after the end of the actual installation period. Regardless whether a prolongation has been taken place or not the procedure will continue with a determination of the course of the backlog on the basis of the present steps of capacity adjustment. The result of the procedure is the adapted capacity profile. So far as there are further steps of capacity increase possible, the procedure will be continued analogous to the case of standard capacity installed, but on the next higher level on the envelope curve.

An already reduced capacity is the third possible state of capacity adjustment (Fig. 7). This constellation appears if the utilisation controller has caused a capacity reduction at a former planning date due to an under-load situation at the work

system. The task of the sub-procedure depicted in Fig. 7 is to raise capacity to the standard level as soon as possible in order to continue with the standard procedure to increase capacity (Fig. 6, left side). For that purpose, the planned date for de-installation

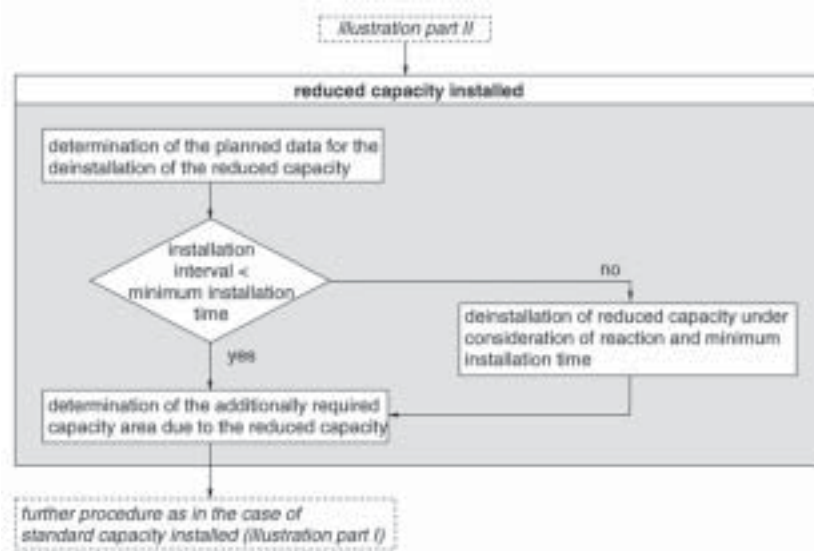


Fig. 7 – Procedure of backlog control (part II).

of the reduced capacity will be determined. If the remaining time interval until de-installation falls short of the minimum installation time of the examined capacity level, a premature return to standard capacity is not realisable. Otherwise, the de-installation date will be postponed to the soonest possible date according to the minimum installation time of the capacity level. As mentioned above, the procedure will be continued as depicted in Fig. 6 on the left side (standard capacity installed).

Fig. 8 depicts the effect of the backlog control procedure in the throughput diagram. The figure shows the cumulated planned output and the actual, respectively, prospected cumulated output of the viewed work system within the evaluation period. The vertical distance between these curve is defined as the backlog. The mean system's performance results from the gradient of the output curve. At planning time a significant backlog is evident. The main task of the backlog controller is now to eliminate the backlog as soon as possible by means of flexible capacities. The backlog controller previously published by the author [Wiendahl & Breithaupt 1998; Breithaupt 1999] requests exactly that amount of capacity to reduce the existing backlog to zero. This is useful only if daily capacity and loading are well synchronised in general and the backlog has been caused by a single disturbance e.g. a rush order or capacity breakdown.

However, if there exists a temporary deviation between mean planned performance and the mean performance of the work system, as depicted in Fig. 8, it is not

sufficient enough to consider the system's backlog only. Rather, it is necessary to bear in mind the trend which led to the actual situation. Therefore, the mean smoothen input rate is taken into account. This seems to be astonishing in the first place, because this is a mingling of input and output parameters. But a work system can be driven in a stable condition only, if input and output are synchronised. Otherwise work-in-process and throughput time will increase (input rate > output rate) or decrease (input rate < output rate). Therefore, the mean smoothen input rate has to be considered also as a reference variable in order to estimate the work system's capacity requirements (and as result of that also to estimate the course of the output).

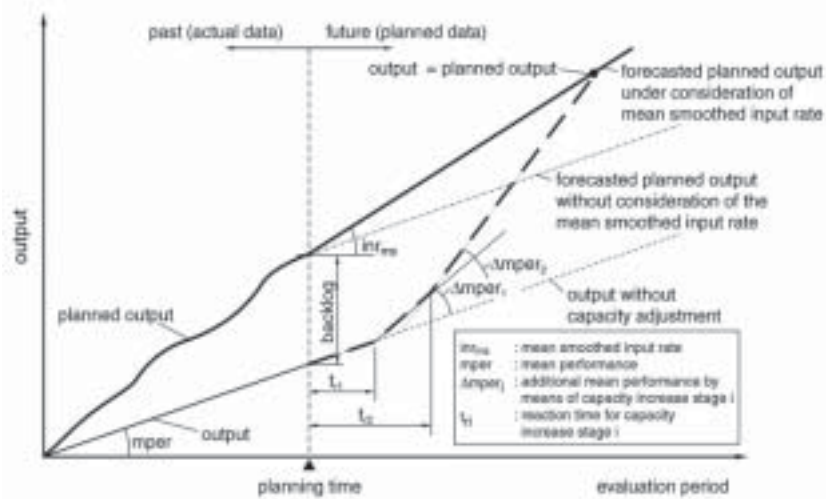


Fig. 8 – Procedure of the backlog-controller in the throughput diagram.

Then, the planned output will be extrapolated in the future based on actual backlog and mean smoothen input rate (Fig. 8, solid bold line). The input rate has been preferred to the mean smoothen planned output because it describes the system's loading situation more currently. It is directly responsible for the momentary and future loading situation. However, the planned output only depicts, which amount of work should have been finished until a certain date. There is no proof whether this work is already available at the work system or still waiting at an upstream system due to a bottleneck situation. Therefore, the question comes into the fore to what extend the input rate influences the existing backlog within the actual planning period.

To determine the necessary capacity to eliminate the actual backlog the forecasted planned output will be calculated (only for the controller decisions at planning time) under consideration of the mean smoothen input rate as depicted in Fig. 8. Then, the backlog controller requests that amount of additional capacity which is

necessary to let the forecasted system's output to catch up the forecasted planned output as soon as possible in order to eliminate the backlog.

In the example shown in Fig 8. two steps of capacity increase are initiated, which take place after expiration of their individual reaction times t_1 and t_2 . Because of that, the mean performance of the system increases and with it the course of the output snaps off upwards. The control process is done, when the backlog has been eliminated (based on the side constraints at planning time). The backlog control procedure takes place at the beginning of every planning period.

From the control theoretical point of view the backlog controller is defined as a three-step action controller. The capacity acts as control variable, not only with two steps of capacity adjustment (additional/reduced capacity) but also with various steps having a certain reaction time. This is the main difference to classic three-step action controllers. The controller behaviour is that complex, that conventional methods of control theory are not suitable for this task. Therefore, software-based logical decision procedures have been implemented (Figs 6 and 7) to carry out the capacity adjustments.

Technical implementations of three-step action controllers distinguish themselves through a so-called switching difference or hysteresis. This switching difference appears, because normally the controller switches on the control signal at a different value of the control variable than it switches the signal off. Usually, the control variable has already reached a value a little bit above the reference one when the control signal is switched on and little below when it is switched off. The sum of both deviations is defined as the controller's switching difference. Mostly, this difference is desirable or necessary in order to prevent continuous controllers from frequent switching. However, there is no switching difference implemented in backlog controller at the moment. It is unnecessary, because there is no permanent control. Moreover, a new control process can be established only at the beginning of every planning period. Due to this inertia a too frequent switching of the controller can be prevented. Nevertheless, from the technical point of view it is easy to implement switching difference into the control algorithm if necessary. This is useful only if the length of the planning period is very short, so that the controller behave almost like a continuous one.

The procedure of utilisation control

In the case of an existing negative backlog at planning time, the utilisation control procedure is launched (Fig. 9). Therefore, the state of capacity adjustment will be determined, analogous to the backlog control procedure. Two states have to be distinguished: firstly, the *standard* or a *reduced capacity* respectively are installed, or secondly, an *additional capacity* is adjusted. In the first case the soonest possible date for further capacity reduction is determined regarding the examined capacity level. So far as the reduced capacity falls short of the *ideal capacity* (see Eq. 1) the procedure will be abandoned, because otherwise not enough capacity to process the system's load will be allocated. In the other case, the reduced capacity will be planned until end of evaluation period. This results in an adapted capacity profile. If there are

further steps of capacity reduction possible, the procedure will be repeated; if not, terminated.

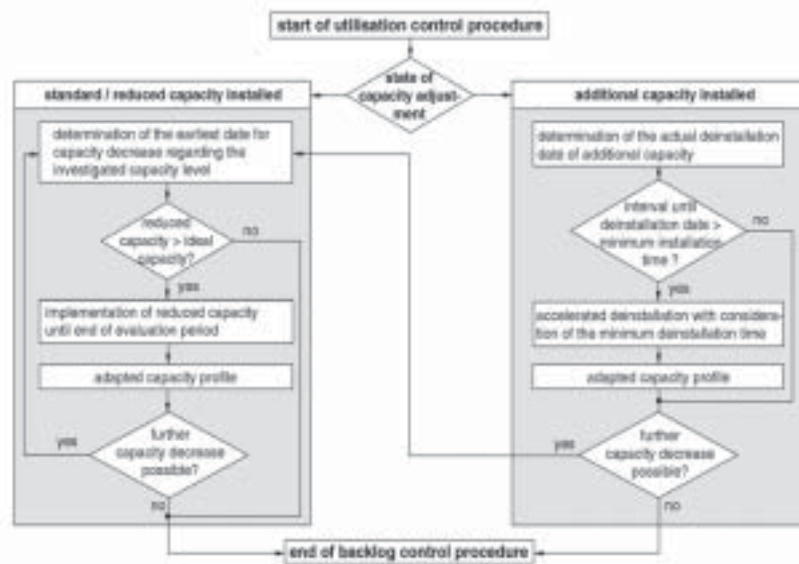


Fig. 9 – Procedure of utilisation controller.

In case of an additional capacity already installed the de-installation date will be determined firstly. If the de-installation date is located within the time interval determined by the minimum de-installation time, no further activities for capacity reduction are carried out at present. Otherwise, de-installation will be accelerated according to the minimum de-installation time. This results in an adapted capacity profile as well. Regardless whether the de-installation is advanced or not a check is made whether a further capacity decrease is possible. If so, the procedure will be continued analogous to the case of standard/reduced capacity installed, but on the next lower level on the envelope curve.

A case study

The logistical rationalisation potentials by means of automatic backlog control has been assessed in a job shop production of an automotive industry supplier. The examination is based on real data from a production area consisting of 16 work systems. The automotive components are produced within a batch and serial production with lot sizes between five and 5000 pieces. The work systems' capacity flexibility is very high and standardised compared with the work systems individual mean daily capacity. With a reaction time of 5 days a capacity increase of 100% can be allocated.

Fig. 9 depicts the courses of wip, backlog and daily capacity for the uncontrolled and backlog controlled case for the work system ‘Special Drilling Machines’. Whereas wip and backlog are increasing constantly in the uncontrolled case, a significant backlog and wip reduction can be realised by means of automatic production control. The backlog is completely eliminated at scd 21. The mean backlog can be reduced by over 80% from 177.7 h to 33.4 h, the wip by 56.9% from 165.1 h to 71.1 h.

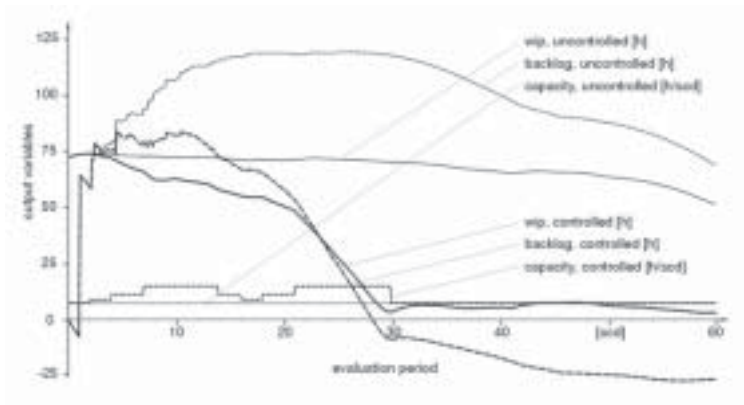


Fig. 10 – Courses of output variables of the work system “Special Drilling Machines” in case of a controlled and uncontrolled system.

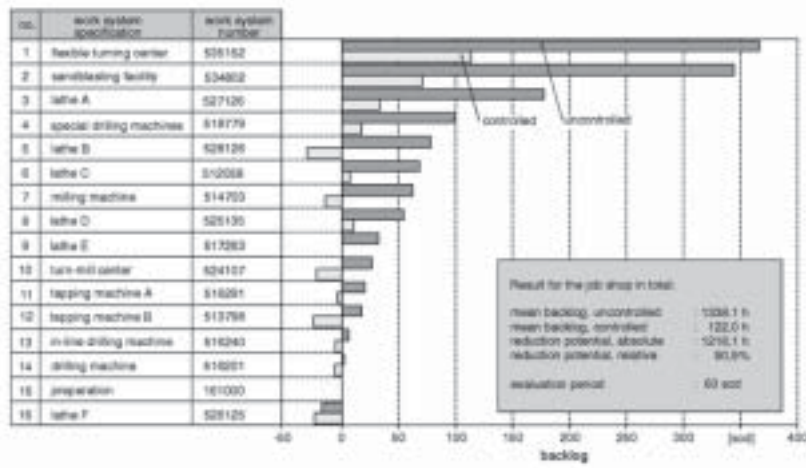


Fig. 11 – Potential analysis for backlog reduction within a job shop of an automotive components supplier by means of Automatic Production Control.

Fig. 10 shows the logistical rationalisation potentials regarding the backlog for all work systems assigned to the above mentioned job shop. The backlog can be

sustainably reduced at all work systems, in total by 90.9% from 1338.1 h to 122.0 h. A more detailed description of this study can be found in (Breithaupt 2000).

Conclusions

With in this paper an advanced backlog-and utilisation controller for production planning based on methods of control theory has been pointed out. The controller allocates the required capacity to guarantee an excellent due date performance without a risk of losses in utilisation. The advantage of this controller lies in the consideration of the mean smoothen input rate. By means of the controller not only disturbances (e.g. caused by rush orders) but also failures in the determination of planning parameters such as *mean daily capacity* can be balanced.

The procedure of the controller is based on logical decisions and can be easily implemented also in production simulation and planning systems without a complex control theory based background. Furthermore this method allows an efficient control of highly flexible capacities, which has been shown impressively by the case study. The method of Automatic Production Control (APC) will be now implemented at Lufthansa Technik Logistik, an logistical service provider for spare part distribution for the aviation industry.

References

- Breithaupt J.W. (1999) *Mastering Systems Dynamics by means of Automatic Production Control*. In: Wiendahl H.P. & Breithaupt J.W. (eds). Proceedings of the 2nd International workshop on Advanced Techniques in Production Planning & Control, 11-12 February 1999, Hanover, Germany.
- Breithaupt J.W. (2000) *Rückstandsorientierte Produktionsregelung von Fertigungsbe-reichen-Grundlagen und Anwendung*. Fortschritt-Berichte VDI, Reihe 2, Nr. 571, VDI Düsseldorf (in German).
- Forrester J.W. (1961) *Industrial Dynamics*. Wiley & Sons: New York-London.
- Mason-Jones R., Naylor B. & Towill D.R. (1999) *Lean, agile or leagile? Matching your supply chain to the marketplace*. Proceedings of the 15th ICPR conference, Limerick, Ireland, 593-596.
- Helo P.T. (1999) *Dynamic modelling of agility in supply chains*. Proceedings of the 15th ICPR conference, Limerick, Ireland, 593-596.
- Milberg J. (1997) *Produktion-Eine treibende Kraft für unsere Volkswirtschaft*. In: Reinhart G. Mit Schwung zum Aufschwung-Information-Inspiration-Innovation, mi Verlag Moderne Industrie: Landsberg/Lech, 19-39 (in German).
- Nyhuis P. & Wiendahl H.P. (1999) *Logistische Kennlinien* Springer. Berlin. (in German).
- Oetinger B.V. (1995) *Von der dritten zur vierten Dimension des Wettbewerbs: Zeit*. In: Oetinger B.V. (ed), Das Boston Consulting Group, Strategie-Buch, Econ, Düsseldorf-Wien-New York-Moskau, 529-534 (in German).

- Petermann D. (1996) *Modellbasierte Produktionsregelung*. Fortschritt-Berichte VDI, Reihe 20, Nr. 193, VDI Düsseldorf (in German).
- Stalk G. (1988) *Time – The next source of competitive advantage*. Harvard Business Review, 4.
- Wiendahl H.P. (1995) *Load-orientated manufacturing control*. Springer: New York.
- Wiendahl H.P. & Breithaupt J.W. (1998) *Automatic Production Control*. In: Drexl A & Kimms A (Eds), *Beyond Manufacturing Resource Planning (MRP II)*. Springer: Berlin-New York-London-Hong Kong, 335-356.
- Wiendahl H.P. & Breithaupt J.W. (2000) *Automatic Production Control applying control theory*. International Journal of Production Economics 63, 33-46.

AGOSTINO G. BRUZZONE
PIETRO GIRIBONE
ROBERTO MOSCA
ROBERTO REVETRIA

*Applied neural networks for improving
production capabilities*

Dept. University of Genoa
Genova, Italy

Abstract — The paper proposes an innovative approach for estimating production capabilities by using Artificial Neural Networks and Simulation; the methods is applied for testing on a real case study involving a large production facility involving very different machines and equipment.

Introduction

The production capability in modern facilities is heavily affected by maintenance and ancillary operations; today the possibility to pre plan the different activities on the shop floor could shut down the impact of such not-productive activities; in order to be effective from this point of view it becomes very critical to be able to properly predict the workload and to estimate times and machinery availability.

Considering the importance of such activities in order to keep low the defects and maintain high the customer satisfaction it is critical to integrate such planning models with production management systems. The authors propose in this paper the development of a modelling approach based on statistical analysis integrated with neural networks and simulation models for planning production and maintenance.

ANN for production capability estimation and maintenance evaluation

The methodology use analysis of variance in order to create a learning database for a set of neural networks self-training that identify the key parameters for the production planning system based that is integrated with a check unit and cost estimator using discrete-event simulation.

The ANOVA module collects the data directly from the field and classify the machine behaviour in order to control the production process; this module is based on two different kind of control.

The first control system checks the resources working on similar jobs in order to improve the technical performance indexes, the second control identify significant difference in the production process versus time.

The neural networks are self-trained on the statistical database organized by the ANOVA module and provide a time series estimation for predicting the optimal time for interrupting the process for resetting and maintain tools and equipment and carrying out ancillary operations.

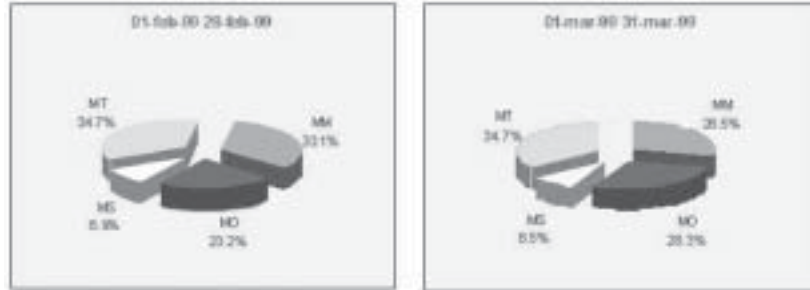


Fig. 1 – Maintenance on the Press Workshop.

The neural networks are based on Boltzman machine architecture that has been proved for time series analysis in this application area, the interface is shown in Fig. 2.

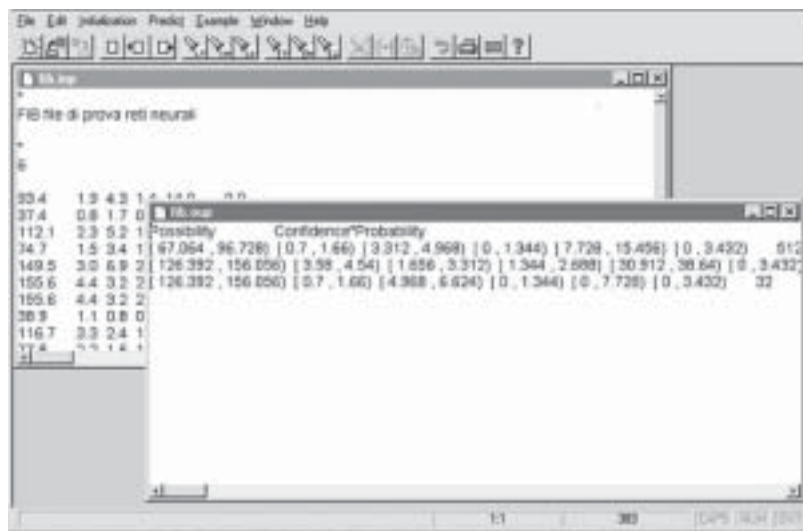


Fig. 2 – Neural Network Predictor's Results.

The predictions are reorganized by the production planning systems that check the overall performances by simulation both from technical and economical point of view.

This general approach summarized guarantees robustness and high efficiency in real production facilities; in effect the system has been implemented on a real industrial case related to a plastic molding production facility.

The authors presents in this report the experimental results obtained with this methodology on the real case studies compared with the traditional operation ma-

nagement; the implementation approach used for this specific case study will be briefly described by the authors in order to exploit the potential of such methods.

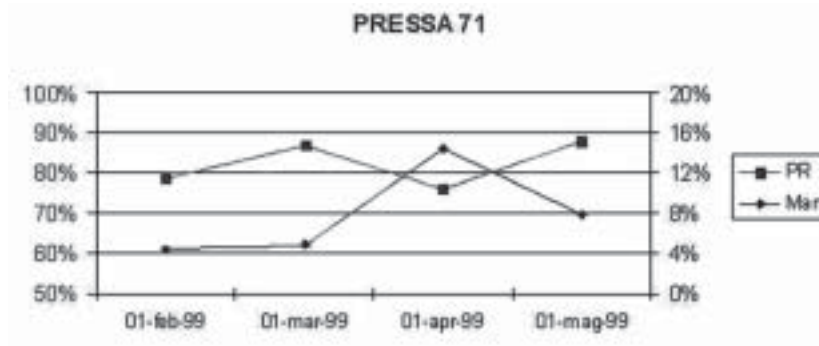


Fig. 3 – Productivity – Maintenance Time Evolution.

Particularly in the last figures are presented some of the results obtained by the implemented methodology: the ANOVA techniques guarantee a consistent Dbase used to train an test the Neural Network, in this way the obtained parameters (see Fig. 4)

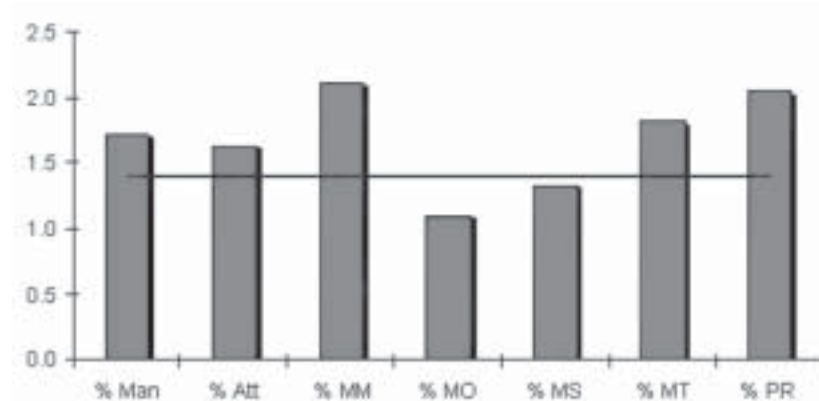


Fig. 4 – ANOVA Results on Maintenance Kind.

such as maintenance time percentage, maintenance kinds can be used as input for the Simul8^e simulation model in order to verify if the planned goals are still achievable.

Conclusions

The experimental results obtained applying this innovative techniques were very satisfactory in application to a quite complex industrial case study.

The definition of methodologies such that proposed in this paper, could improve the impact of AI (Artificial Techniques) on industry, by the definition of operative procedure for their application and use; obviously the results can't be generalized, however in similar case study it could be possible to accept hypotheses of fast convergence and effectiveness of this approach.

In the future this research could be integrated with production scheduler and planning system by an hierarchical approach that guarantees quick and consistent response time at operation level and general overview of strategic issues; the application of such method will be extend in the near future to other industrial case studies such as automotive production facilities.

References

- Anderson J. & Rosenfeld E. (1988) *Neurocomputing-Foundation of Research*. MIT Press: Cambridge MA.
- Bruzzone A.G. & Kerckhoffs (1996) *Simulation in Industry*. Genoa, October, Vol. I & II, ISBN 1-56555-099-4.
- Bruzzone A.G., Giribone P., Revetria R., Solinas F. & Schena F. (1998) *Artificial Neural Networks as a Support for the Forecasts in the Maintenance Planning*. Proceedings of Neurap 98, Marseilles, March 11-13.
- Bruzzone A.G. & Giribone P. (1998) *Simulating Assembly Processes in Automotive Support Industries for Production and Design Improvement*. Proceedings of XV Simulator International, Boston, April 4-9.
- Bruzzone A.G. & Giribone P. (1998) *Decision-Support Systems and Simulation for Logistics: Moving Forward for a Distributed, Real-Time, Interactive Simulation Environment*. Proceedings of the Annual Simulation Symposium IEEE, Boston, April 4-9.
- Hassoun M.H. (1995) *Fundamentals of Artificial Neural Networks*. MIT Press: Cambridge MA.
- Hillis D.W. (1989) *The Connection Machine*. MIT Press: Cambridge MA.
- Lippman R.P. (1987) *An Introduction to Computing with Neural Networks*. IEEE ASSP, April.
- Mehrotra K., Mohan C.K. & Ranka S. (1997) *Elements of ANN*. MIT Press: Cambridge MA.
- Montgomery D.C. (1976) *Design and Analysis of Experiments*. Wiley and Sons: New York.
- Mosca R., Giribone P. & Bruzzone A.G. (1993) *Application of Neural Networks to On-Line Simulations of Plant Processes*. Proceedings WNN93, San Francisco, November 7-10.
- Mosca R., Giribone P., Bruzzone A.G., Orsoni A. & Sadowski S. (1997) *Evaluation and Analysis by Simulation of a Production Line Model Built with Back-Propagation Neural Networks*. International Journal of Modeling and Simulation, 17 (2), 72-77.

- Mosca R. & Bruzzone A.G. (1997) *Simulation as a Support for Customer Satisfaction-Oriented Planning*. Proceedings of Simulators International XIV, SMC'97, Atlanta GA, April 6-10.
- Mosca R., Giribone P. & Bruzzone A.G. (1993) *Optimum Area Search Techniques Applied to Studies Relative to Plant Problems performed by means of Simulation*. Proceedings Simtec93, San Francisco, November 8-10.
- Padgett Mary Lou & Roppel T.A. (1992) *Neural Networks and Simulation: Modeling for Applications*. Simulation, 58 (5).
- Weigend A.S. & Gershenfeld N.A. (1994) *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley: Reading MA.

SERGIO CAVALIERI¹
VITTORIO CESAROTTI²

*A multi-agent model for coordinated supply
chain planning*

¹ Dipartimento di Economia e Produzione
Politecnico di Milano, Italy

² Dipartimento di Ingegneria Meccanica
Università di Roma Tor Vergata, Italy

Abstract — In this paper the authors propose a multi-agent model for the introduction of coordinated policies in supply and replenishment of inventories within an integrated supply chain. The aim of this work is to show the advantages that the use of a distributed approach can provide in implementing vertical and lateral coordination among the elements of a non centralized supply chain. The paper, after introducing the main concepts related to the adoption of coordination mechanisms within supply chains, the main issues of agent oriented technology and a review of the related literature, presents the structure of the model and the main agents' static and dynamic behaviours. Then, the paper describes the design and structuring of specific coordinated models, the negotiation mechanisms introduced and the parameterisation of the models. Finally, the implementation of the models to a case study has been simulated and the results analysed.

Keywords — multi-agent systems, supply chain, negotiation, coordination, simulation.

Introduction

According to the principles of supply chain management, modern companies attempt to achieve high-volume and broad-mix production using minimal inventories throughout the logistic chain, without neglecting at the same time shorter response times.

As a result, most of the integrative research (from a supply chain context) in the literature has been historically based on an inventory management perspective. In fact, the term "Supply Chain" first appeared in the literature as an inventory management approach, and, more precisely, with the proposition of "multi-level" or "multi-echelon" inventory control systems [Vollman 1992; Ganeshan 1997], aiming at modelling and managing the concurrent presence and interaction of multiple tiers within a logistic chain.

However, as [Anupindi 1999a] and [Swaminathan 1997] highlight, the majority of these works assume centralized control of the supply network, thus overlooking the possibility of decentralized decision making. Moreover, most of the models under the "inventory theoretic" paradigm are very restrictive in nature: they restrict themselves to certain well known forms of demand or lead time or both.

These design hypothesis appear quite unrealistic and highly limiting the applicability of these models. In fact, the majority of distribution systems are characterized by the presence of autonomous or semi-autonomous entities, each pursuing its own objectives and taking its decisions upon availability of local

knowledge and information. These entities are highly interdependent when it comes to improving performance of the whole supply chain in terms of on-time delivery, quality assurance and cost minimization.

Hence, coordination strategies need to be defined in order to manage the interaction among heterogeneous decision-makers, thus ensuring a proper balance between local objectives and constraints and the global performance of the entire supply chain.

Of course, the absence of any integrated information system hinders the proposition of a valuable and effective coordination strategy. Most of the earlier attempts in supply chain management, due to limitations in available technology and to the complexity of the problems, did not enable an integrated approach of problem solving. In the recent years, the advent of strong development of powerful hardware systems and complex software techniques has encouraged the development of multi-agent systems with suitable inter-agent communication and coordination protocols.

In this paper, a review of the most consolidated coordination mechanisms employed for supply chain modelling is proposed. Then, the potentialities of agent technology in implementing effectively the coordination strategies will be introduced.

In the second part, a multi-agent model for coordinated supply chain planning will be described and the first results of experimental campaigns will be reported and discussed. The model has been applied to a real two-level distribution system, constituted by a supplier and a geographically distributed network of retailers.

Coordination mechanisms

A network is coordinated when a single decision maker optimises the network with the union of information that the various decision makers have. Lack of coordination occurs due to the existence of multiple decision makers in the network who may have different information and incentives. Even under information symmetry (all parties are equally informed) the performance of the network may be sub-optimal, since each decision maker optimises a private objective function and local optima need not to be globally optimal [Anupindi 98].

Considering a multilevel distribution system (made up of a supplier and a network of geographically distributed retailers), two different mechanisms of coordination can be adopted:

Vertical coordination-it can be in turn declined in:

(i) *Hierarchical relationship*-where the downstream tier has merely only informative role, delegating all the decisions to the higher tier. This is the typical strategy that is subsumed in continuous replenishment strategies, where there is a centralized control by the supplier of all the information as well as of the material flows of the whole downstream logistic chain.

(ii) *Competitive relationship*-characterized by decisions assumed locally by the single decision maker, without any previous agreement or negotiation. This practice, widely adopted, is implemented by the *installation stock reordering* techniques [Svoronos 88].

(iii) *Cooperative relationship*-considers possible relaxation of local constraints and objectives in favour of a better global performance of the whole system [Kjenstad 1998].

Lateral coordination-generally cooperative, declinable in:

(I) *Full cooperation*-where the utility functions of the single entities are fully identical to the global objectives of the whole system (see, as an example, the work of Axsater e Zhang [Axsater 1999]).

(II) *Antagonistic cooperation*-which implies the adoption of negotiation mechanisms (mainly based on the contract-net protocol) between peer-level decision makers (see the work of Anupindi [Anupindi 1999a]).

As depicted in Fig. 1, while vertical coordination rules the dynamics of interaction between upstream entities and their downstream tiers, lateral coordination protocols support interaction between peer-level entities. A typical example of lateral forms of

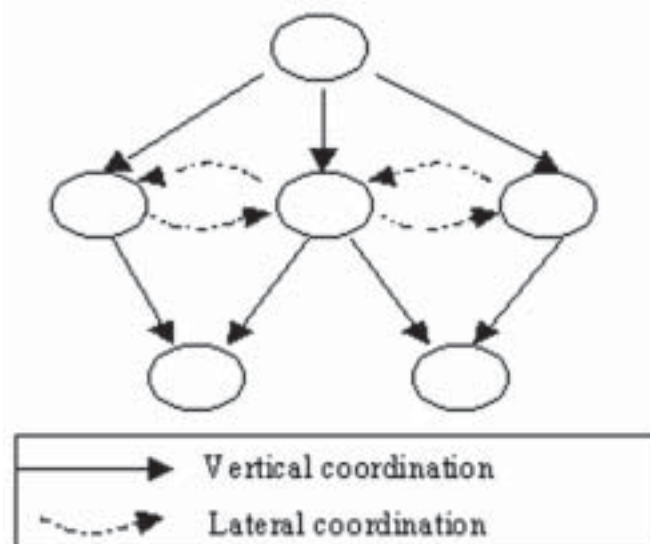


Fig. 1 – Vertical and lateral coordination in a supply chain context.

coordination is the possibility to face local stock-outs by using transshipments of overstocks of the same item residing in depots of peer-level entities within the same distribution network.

The rising of the agent technology

The need of proper coordination mechanisms for ensuring a consistent behaviour across a supply chain makes evident how information and knowledge sharing play a decisive role within the highly distributed and strongly interrelated decision-making process between suppliers and distributors (or wholesalers).

In this sense, the absence of any integrated information system hinders the proposition of a valuable and effective coordination strategy.

As a result, in more evolved industrial sectors, as food distribution and microelectronics industry, major attention is given towards the design and development of *Interorganizational Information Systems (IOIS)*. These systems can be viewed as an integrated data-processing and/or data communication system shared by two or more organizations.

At current time, different typologies of systems are commercially available differing each other mainly for their own level of complexity and sophistication. They span from the EDI (*Electronic Data Interchange*) systems or PoS (*Point of Sale*) applications for data communication or process monitoring, to DSSs (*Decision Support Systems*), mainly used for providing computational support whenever complex and partially structured decisions need to be made [Ram98]. The DSSs are mainly systems based on relational databases, linear programming algorithms or expert systems, which, in response to specific queries, support the operator in his/her normal decision making activities.

One main lack of these systems is that they normally operate on local scale in a stand-alone manner. In fact, they are mainly developed according to an antagonistic “pipeline” view of the relationship between supplier and consumer. They merely lack of any effective coordination mechanisms.

One possible improvement, able to comprehend the main integration pushes, is given by the possibility to develop distributed decision support systems based on agent technology. On the concept and properties of agents, several definitions have been provided. In particular, considering the development of a distributed DSS, two main properties of an agent are the possibility to operate in *autonomous* manner, which means having own responsibilities and tasks to accomplish according to local objectives, and the *social ability*, considered as the capacity to decide own strategies consistently and in a coordinated way with the outer environment (normally made up of other agents).

Thus, in order to drive the whole supply chain to an optimal state, inter-agent communication and coordination is necessary. Such a multi-agent system, that needs explicit coordination rules between agents and where the overall system itself has an objective, is called agent organization. When viewed globally, this agent system can be viewed as a distributed decision support system: each agent has the responsibility of a specific task, but the organization as a whole accomplishes the objective of optimal integrated supply chain management.

Literature review

An application example of adoption of multi-agent systems in a supply chain context is given by the approach proposed by Hinkkanen et al. [Hinkkanen 1997]. In their model, after having identified the main business processes within a logistic chain, each process/task is associated to a specific agent. These agents develop, in turn of their human decision making counterparts, all the operating activities of a

logistic network, by coordinating with each other for all the choices pertaining to replenishment of warehouses, production and transportation planning. Coordination is ensured by using a simultaneous auction, where, unlike usual auction, resource allocation is not performed after resource agent selects the winning bids among those submitted by the requesting agents. The former rather makes all bids public to the requesting agents who, in their turn, can adjust their bids. The reason why the resources are not allocated at once is because a requesting agent may need at the same time different resources in order to function, and only access to all resources results in feasibility.

The authors maintain that the main strength point of this approach resides on its highly distributed nature, which ensures the DSS high flexibility, modularity and scalability (since it is possible to plug new agents in different moments, thus allowing a progressive and not abrupt substitution of the human decision makers with their automated counterparts).

Sadeh et al. [Sadeh 1999] introduce MASCOT, an architecture that aims at providing a framework for coordinated development and manipulation of planning and scheduling solutions at multiple levels of abstraction across the supply chain. By using the MASCOT framework, Kjenstad [Kjenstad 1998] compares several lateral coordination policies under different sets of business assumptions, ranging from simple scheduling policies based on historical lead time data to finite capacity lateral coordination. Also Brun and Portioli [Brun 1999] compare in their work the performances of a distributed coordination mechanism based on a market-like auction framework with those of an implicit and a centralized coordination model.

In the model proposed by Anupindi et al. [Anupindi 1999], each agent is an independent profit-maximizing entity. It is faced with two different types of decisions with respect to the interaction with other agents:

(i) the first type, regarding the decision of how much inventory a given agent decides to buy on its own, can be made unilaterally, without the need to reach an agreement with the other agents;

(ii) for the second type of decisions, such as on a level of common pool inventory or on the shipping and allocation decisions, the agents somehow must reach a certain level of consensus, i.e. must act in a non-unilateral fashion; they need to agree on the way to allocate, by using transshipment mechanisms, eventual excess stocks to satisfy unmet demand in the whole distribution system and to allocate the surplus profits resulting from the sharing of stocks.

This time separation suggests a solution concept in which agents act cooperatively *ex post*, even though their *ex ante* actions are determined in a competitive fashion. Such an hybrid approach is termed by the authors as “cooperative strategy”.

Finally, Swaminathan et al. [Swaminathan 1997] highlight the high level of difficulty in developing a set of generic processes which can capture the dynamics of a supply chain across a wide spectrum. They propose a multi-agent based framework for developing customized supply chain models from a library of software components. These components capture generic supply chain processes and concepts, thereby promoting modular construction and reuse of models for a wide range of applications.

Considering previous experience gained by the authors with multi-agent systems [Cavalieri 1998] and triggered by a real industrial test case, in this paper major efforts have been addressed to the mutual comparison of different coordination protocols with the common aim of improving the efficiency and efficacy of a distribution chain. As it will be clear in the next part of the paper, starting from a base reference model, lateral and vertical coordination mechanisms have been progressively added and their main effects simulated, thus allowing an objective and evident mutual evaluation and comparison.

The structure of the model

Unlike the model proposed in [Hinkkanen 1997], the modelling activity in the present research is particularly focused on the distribution part of a logistic chain.

The model is articulated in more levels and assumes a tree-like form, representative of a logistic chain which spans over a wide geographical territory. The model comprehends a production agent which is charged of the fabrication of the product and the management of a centralized warehouse, wholesaler agents (distributing items of different brands), which cover the whole market, and final consumers, which are representative of the final demand.

The agents

The description of each class of agents (consumer, wholesaler, supplier) outlines both static and dynamic characteristics of the various supply chain entities.

Final user

An user is defined by the following set of characteristics at a given time instant:

S_i -set of attributes that characterise its state; in particular, possible states of the final user can be:

- *satisfied*-when it receives the requested product in the quantities ordered within its maximum allowable waiting interval time;
- *unsatisfied*-when it doesn't get the requested product within the allowable waiting interval; in the model, we have assumed that each final user can order only one type of product at time and doesn't accept fewer quantities than that initially ordered;
- *waiting*-when it submits the order to the wholesaler, but it is said to wait, because of current unavailability of the goods, with a forecast delivery time within the maximum allowable waiting interval time;
- *searching*-when, in presence of a stock-out of the requested product by a wholesaler, it searches the same product by another wholesaler; only one possibility of search is given within the model; if the user succeeds in his search then its state is registered as satisfied;
- *replacing*-when, in presence of a stock-out of the requested product, it purchases a substitutive product (this state can be considered a sub-state of the unsatisfied condition).

o_j -outgoing message to a specific wholesaler j .

i_j -incoming message by wholesaler j .

$M_i(o_j)$ -defines the message semantics for the outgoing message o_j : it specifies the type of product requested, the quantities ordered and the maximum time interval the user is willing to wait.

$M_i(i_j)$ -defines the message semantics for the incoming message i_j : it specifies the type of product requested, the quantities ordered and the availability to promise within the maximum time interval.

As a result, the dynamic behaviour of the final user is strictly dependent on the content of the response message received from the wholesaler. In particular, among the different possible states, also the *replace* and *search* functions are considered, thus making the model more realistic. Normally, in fact, the final user is given the possibility to search for another wholesaler or to replace the desired product with a substitutive one of a different brand.

Wholesaler

Also the wholesaler is characterized by a set of characteristics at a given time instant:

S_i -set of possible states; in particular, two states are given:

(i) *user's order processing*-when a new order is received;

(ii) *request to central warehouse*-when request for replenishment of the local stocks is formulated and transmitted to the supplier.

In addition to the messages received and transmitted to the final user (described previously), we consider also:

o_j -outgoing message to supplier j .

$M_i(o_j)$ -defines the message semantics for the outgoing message o_j : it specifies the type of product requested and the quantities ordered.

Each wholesaler agent manages its own inventory and keeps track of the different destinations of all the goods being stocked, ordered by the final user or requested, but not yet arrived, to the supplier. The replenishment of the local inventory is ensured by a ROL(Re-Order Level) strategy.

Supplier

The production plant is supposed to serve the wholesalers by producing and shipping different typologies of products. Thus, the main decision making activities carried out by the supplier are the decisions on how to allocate the finite capacity of the plant on alternative productions and how to treat competing requests of the same product submitted by different wholesalers.

Accordingly, the possible states S_i of the supplier are:

(i) *production*-during the normal production shifts;

(ii) *order processing*-at the end of the day, by gathering and processing all the requests submitted by the wholesalers.

In particular, regarding the latter state, if the amount ordered can be processed then the item is shipped to the wholesaler (which will receive it after a fixed number

of days according to the required transportation lead times), otherwise the request is rejected.

In addition, every week a forecast of the future product demands is carried out considering the historical data of the previous twelve weeks. This forecast is then employed as input data for deciding how to allocate in the next period (i.e. the days of the next week) the finite capacity of the plant between the different typologies of products.

The basic model

In order to experiment the features of the model, we have realized a basic reference model on the basis of which we have then implemented different policies.

The basic model is structured in two levels of supply. The final customers buy from a territorial network of wholesalers. The wholesalers replenish their stocks from the warehouse of the supplier (there is only one supplier, with one centralized warehouse for all the network). In case the quantity available in the supplier's warehouse is not sufficient, the wholesalers orders are fulfilled with a first-in-first-out policy.

There is no kind of coordination in the supply chain and no kind of information sharing.

The coordinated model

The basic model described above is based on simple, non-coordinated dynamics that represent many cases one can find in the distribution sector.

In order to evaluate the impact that different kinds of coordination can have on the performances of a supply chain, we have implemented, by means of the multi-agent structure described in § 5, three models of coordination:

- (i) vertical coordination, based on a negotiation system between wholesale agents and plant;
- (ii) lateral coordination, based on a negotiated transshipment model;
- (iii) information symmetry, based on a complete transparency of information between levels.

The models foresee the presence of a monitor agent which has at the same time a triggering function for the negotiation process and a control function for the transaction of information and materials.

Vertical coordination

The vertical coordination model is based on a market-like negotiation system able to model both cooperative and competitive relationships (Fig. 2).

The negotiation is based on the contract-net multi-stadium protocol [Smith 1980] working with a virtual objective pricing system.

The model is scalable since it is possible to use the same negotiation system to any number of levels of the supply chain (with the opportunity of specifying parameters and methods different for each level) and extremely flexible allowing the

implementation of different coordination relationships just modifying the evaluation procedures at the above level of the supply chain (as it will be more clear in the following case study simulation).



Fig. 2 - Interaction between wholesaler and supplier.

The negotiation process is based on a pricing mechanism with an offer, a first selection, a counter-offer and a final selection of the best offers.

During its supplying phase, each wholesaler agent assigns an offer price to each batch of orders he requires from the central warehouse.

This price is formulated on the basis of equations [1] as a function of the weight of each order (Q_i, P_i) in relation to the specific objectives of the wholesaler.

$$p_0 = \frac{q_0}{q_{max}} \cdot k; p_i = \frac{q_i}{q_{med}} \cdot \frac{t_{std}}{\Delta t_{ATP}}; P_j = \frac{q_0 \cdot p_0 + \sum q_i \cdot p_i}{q_0 + \sum q_i} \quad [1]$$

where:

q_0 = necessary quantity for the replenishment of the fixed level of stock,

q_{max} = maximum quantity of goods in stock (for hypothesis known and fixed),

k = proportionality factor between p_0 and p_i ,

- p_0 = priced assigned by a wholesaler for the supply of a quantity q_0 ,
 q_i = quantity of a generic order reserved by a final customer,
 q_{med} = average quantity of the products in an order of final customers,
 t_{std} = standard lead time of the system,
 t_{ATP} = residual lead time to the delivery date promised to the final customer,
 P_j = priced assigned by the wholesaler for the supply of a global quantity $Q_j = q_0 + \sum q_i$.

That means:

- (i) for the stock replenishment orders (q_0, p_0) in relation to the current stock level (therefore to the risk of going out of stock);
 (ii) for the reserved orders (q_i, p_i) in relation to the weight of each customer that has reserved the product and to the due date promised to the customer (according to an *available to promise-ATP* logic).

The warehouse, once received orders Q_j and offers P_j expressed according to the wholesalers' interest, starts an evaluation process aiming to the satisfaction of the supplier's interest, that is the intent of using the available stocks in a way that can maximize the global income. For this reason, the warehouse evaluates each offer on the basis of a factor weighing the importance of the wholesaler (G_j) (since it is interested in advantaging the distribution channels with the highest selling capacity) and on the basis of a factor weighing the importance of every single order (O_j), for a maximization of the total sales:

$$P'_j = P_j \cdot G_j \cdot O_j \quad [2]$$

Then, the warehouse verifies the capacity of satisfying all the orders received. If this check is positive, it distributes the goods to the wholesalers, otherwise it splits the orders in two groups on the basis of their importance [2]. The first group has the goods directly delivered, while the second must face a new negotiation starting a competition based on a progressive price rise.

Lateral coordination

The lateral coordination model, able to model both full and antagonistic cooperation) is represented in the Fig. 3.

As for the vertical coordination model, the process is controlled by the monitor. A modification in the monitor's method of formulation/evaluation of counter-offers can allow the switch between full and antagonistic coordination systems. While a modification in the monitor's trigger can allow changes in the integration between lateral and vertical coordination mechanisms.

In the model we have implemented for the case study an antagonistic lateral coordination; it is activated only when the stock-out risk for the wholesaler rises (that is when a wholesaler is not able to satisfy an order within the due date promised

to the final customer). This kind of lateral coordination does not implement multi-stage negotiation.

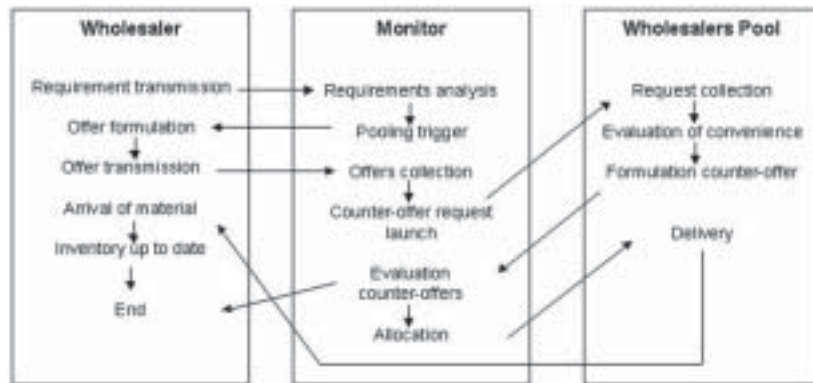


Fig. 3 – Lateral coordination between wholesalers.

The wholesaler, when the final date for the order's fulfillment is approaching, starts looking for the missing goods among its competitors, by means of the launch of an offer at a virtual price expressed with the equations [1]. Once the offer has been launched by the wholesaler, the monitor agent collects it and sends out a message to all the other wholesalers collecting their availability and their counter-offers. Then, the monitor agent evaluates the counter-offers and if it finds one or more compatible with the original offer, it accepts on behalf of the wholesaler the most interesting one (that is the lowest counter-offer) starting the transshipment procedure.

The pooling counter-offer mechanism, due to its antagonistic nature, is implemented in order to satisfy the counter-offering wholesaler's interest; that is pricing the counter-offer on the basis of the available stock level (supposing that each wholesaler tends to have the highest rotation of stocks as possible) by means of a function $f(q_F^j)$, and in no case going under a minimum price linked to the price the goods had been bought at (see equation [3]).

$$p_i = \min \{f(q_F^j); p_0\} \quad [3]$$

Information symmetry

Both vertical and lateral coordination systems seen in the above paragraphs modify the wholesalers' supply processes. They do not modify the processes that regulate the transaction between the wholesaler and the final customer. However, this is the most critical transaction, where the whole chain generates its revenue and there are some critical elements that could be improved by means of a reinforcement of the information symmetry of the supply chain.

Within the model that has been developed for the case study, we have tried to face this criticality by enhancing the information symmetry of the supply chain.

The actions have been the following: from one hand, we have given to the wholesaler the possibility of knowing at every moment the current level of stocks in the supplier's warehouse (and therefore to give to his customers more reliable due dates). From the other hand, we have given to the supplier the possibility of knowing the sale forecasts and the orders of each wholesaler.

In fact, usually the real market demand arrives to the central warehouse (and therefore to the manufacturing plant) filtered by the supply policies of the wholesalers. These modify essentially the demand curves for the plant (since it is impossible to distinguish the orders for stock replenishment from those for customer demand) causing a loss in service level for the whole chain.

A first result we expect from this common sharing of information is an improvement in the production scheduling together with a reduction of the stock level in the warehouse without any change in the service level.

In the implemented model each wholesaler sends every week its sale forecasts to the plant agent. The forecasts are stored in a vector and used by the plant as the time series on which the production planning for the four following weeks is based on.

This process is not so different from the one implemented in the basic model. However, the possibility of basing the planning on real demand forecasts and not on a replenishment demand forecast is expected to give good results in terms of effectiveness of the planning process. In this way the overall performance of the network is expected to improve.

This approach is quite different from the approaches known in literature or in the industrial practice. Usually some cases in which the wholesaler has visibility over the centralized warehouse are found, but it is very uncommon, especially in decentralized supply chains (that are supply chains not owned and controlled by one centralized decision making entity), to find wholesalers that share information of sales with the suppliers.

However, it has been shown by direct interviews that wholesalers are ready to share this kind of information as far as this does not involve any kind of control by the supplier and if they have something in change from the supplier (in this case information about stocks in the central warehouse).

Moreover, one must consider that with the great improvement in communication tools and protocols, any kind of data transfer and sharing is by far much simpler to implement and does not need any kind of structure (and fixed cost) to become effective.

The simulation environment

The model described above has been applied to a case study consisting in a company producing electromechanical components (perfectly replaceable by equivalent components produced by the competitors) distributed by means of a network of wholesalers (not owned by the company nor dedicated to the distribution of the company's products) spread all over the national territory.

The model has been implemented with an *object-oriented* technology with a software application especially developed in Java language, chosen among the available object-oriented languages, for its multi-platform capabilities. This structure has allowed a very simple modelization of many different configurations of the supply chain and of its main processes in order to evaluate and compare different solutions for the distribution system.

Moreover, in order to facilitate the simulation and the analysis of the results, we have added to the Java engine a relational database for the collection and the comparison of the simulation results and for a step-by-step control of the simulation runs.

The two systems (the Java simulation engine and the relational database) have been linked with a ODBC (Object Database Connectivity) technology by means of a JDBC-ODBC (JDBC: Java Database Connectivity) bridge.

Model parameterization

In the model we have chosen as the general structure for the negotiation system a vertical coordination with the possibility of integration with an antagonistic lateral coordination to be activated only in case of imminent stock-out of goods.

The analysed models are scalable and flexible by means of the possibility of modifying (by simple inputs at the simulation run) the following structural parameters:

- (i) Number and characteristics (volume of sales) of the wholesalers;
- (ii) Number and characteristics of the products in the system (the number m of goods, the set z of wholesalers in charge of their distribution, the demand pattern for each good);
- (iii) Number of customers expected to enter the system based on a stochastic distribution (normal, Poisson or exponential) of which is known the average and the standard deviation (not considering the seasonal effects);
- (iv) Seasonal effect in the demand curve (supposed periodic with a parametric period) with the definition of the period and the width of variability;
- (v) Average quantity of goods required by each customer (distributed according to a Pareto distribution among the different products) and the stochastic variation of this quantity (again according to a normal, Poisson or exponential distribution);
- (vi) Dimension of the warehouses of the supplier and of the wholesalers (in this way authors can fix the maximum limit of grounded goods);
- (vii) Safety stocks for each warehouse (the supplier's and the wholesalers') and for each good, that is based on the variance of the demand and on which is based the dimension of the maximum quantity and the reorder level for each warehouse;
- (viii) Reorder level that is used when, at the end of each day, each warehouse checks its stock to verify the need of a replenishment order;
- (ix) Maximum quantity that is the fixed level of stock for each good to be replenished when the stock goes under the reorder level;
- (x) Replace factor, that is the probability of finding a competitors equivalent product in the wholesalers' stock once that the customer has not found the suppliers' product;

(xi) Search factor, that is the number of customers that look for a different wholesaler (belonging to the same supply chain) after not having found the required good in a wholesaler;

(xii) Tuning parameters, useful for the system's negotiation process setup, since they regulate the rules and the processes of each negotiation such as the relative importance of the wholesalers, of the customers and of the orders.

The length of the simulation run has been determined by the MSPE (Mean Square Pure Error) theory. The result has been an ideal simulation run of 1800 days (corresponding to 5 years).

Metrics for the results analysis

The metrics chosen for the analysis of the system's performance in its different configurations are of two different kinds:

(a) global metrics,

(b) local metrics.

In the first case, we have observed parameters which reflect the results of the whole system (supplier, wholesalers and customers), with the objective of analysing the supply chain as a whole and its ability in satisfying the markets needs.

On the contrary, in the second case we have observed parameters which reflect the results of each element of the system in order to evaluate the profitability of each element and therefore its interest to stay within the supply chain (since we are working under the hypothesis that the supply chain is not centralized).

Global metrics

In this group, we have collected the following metrics:

(i) Satisfied, unsatisfied, waiting, twice unsatisfied (after the reservation), etc., that are all the different conditions in which a customer may be found while he is in the system or once he has left the system.

(ii) Sold and required, that allow the analysis of the global sales of the systems relatively to the demand and split for each different product;

(iii) Number of customers waiting in general for the delivery of an order or for each single product;

(iv) Average wait time for the satisfied customers;

(v) Percentage of customers immediately satisfied (with no return after reservation) globally or for each product;

(vi) The average global level of stocks in the warehouse.

Local metrics

In this group, we have analysed the same metrics seen above but at the detail of the wholesaler, with the addition of the average value of products in stock (not included, since irrelevant, at a global level). Analysing these metrics we have distinguished between the different categories of wholesalers, mainly on the basis of their dimension and of their sales.

Analysis of the results

We have simulated the model described above in its four main configurations (basic model, vertical coordination, lateral coordination and information symmetry) analysing the impact the configuration changes have on the main performance indexes. The performance indexes are parameters resulting from the elaboration of the metrics seen in the previous paragraph.

In detail, we have chosen five main performance indexes that are the following:

(i) percentage of final customers satisfied relatively to the total number of customers that entered the system (*sat*);

(ii) percentage of customers satisfied in ready delivery (without coming back after reservation) relatively to the total number of satisfied customers (*rdel*);

(iii) percentage of sold goods relatively to the total amount of goods required (*sales*);

(iv) average level of stocks in the warehouses calculated as the ratio between average inventory over the average daily demand (relatively to product and to the wholesaler) (*stk*);

(v) average wait time for the customers (*wait*).

The parameters chosen in the simulation model are 20 wholesalers distributed over a continuous range of importance (sales), 3 categories of products representing A, B and C classes of the Pareto distribution, average quantity per order of the final customer 100, with a normal distribution with a variance of 30, 200 new customers per day with a Poisson distribution, season periodicity of 6 months with a maximum variation of 25%, reorder level set to four times the safety stock and maximum quantity set to eight times the safety stock, competitors' service level of 85% (for the replace function) and a 15% of search degree.

Analysis of simulation runs

Here we analyse the simulation results. We will analyse separately the four different models studying the effect of search and replace functions on each model.

Basic model

The basic model shows an average behaviour over almost all the performance indexes (*see table 1 in the next page*). The search function has almost no influence on the performances while the replace function causes a strong decrease (to the lowest level) of the satisfied customers and of the total sales, with an increase of stocks (the replace function also improves the number of ready deliveries and the waiting time, but this is trivial and adds nothing to the global performances).

Vertical coordination model

The vertical coordination model (*see table 2 in the next page*), compared to the basic model, shows a sensible improvement both in global sales (+1.5%) and in percentage of satisfied customers (+1.0%). On the other hand, there are no major changes in stocks level and in waiting time and there is a worsening in ready delivery.

<i>Model</i>	<i>Sat</i>	<i>Rdel</i>	<i>sales</i>	<i>stk</i>	<i>wait</i>
Basic	93.0%	87.1%	93.2%	2.42	1.54
+search	93.1%	87.2%	93.3%	2.44	1.53
+repl.	91.8%	98.6%	92.1%	2.92	1.39
Worst	91.8%	78.9%	92.1%	3.09	2.29
Best	96.1%	98.8%	96.4%	2.22	1.30

Table 1 – Simulation results of the basic model (worst and best are the worst and best performance registered throughout the whole simulation campaigns).

<i>Model</i>	<i>sat</i>	<i>rdel</i>	<i>sales</i>	<i>stk</i>	<i>wait</i>
Ver.C.	94.0%	81.2%	94.7%	2.34	1.56
+search	93.7%	81.1%	94.5%	2.32	2.04
+repl.	91.8%	98.3%	92.1%	2.95	1.51
Worst	91.8%	78.9%	92.1%	3.09	2.29
Best	96.1%	98.8%	96.4%	2.22	1.30

Table – Simulation results of the model with vertical coordination (worst and best are the worst and best performance registered throughout the whole simulation campaigns).

<i>Model</i>	<i>sat</i>	<i>rdel</i>	<i>sales</i>	<i>stk</i>	<i>wait</i>
Lat.C	94.7%	79.6%	95.2%	2.28	1.57
+search	94.3%	78.9%	95.0%	2.22	2.29
+repl.	92.0%	98.2%	92.2%	2.92	1.55
Worst	91.8%	78.9%	92.1%	3.09	2.29
Best	96.1%	98.8%	96.4%	2.22	1.30

Table 3 – Simulation results of the model with lateral coordination (worst and best are the worst and best performance registered throughout the whole simulation campaigns).

<i>Model</i>	<i>sat</i>	<i>rdel</i>	<i>sales</i>	<i>stk</i>	<i>wait</i>
Inf.Sym	96.1%	81.8%	96.4%	2.54	1.74
+search	96.0%	81.5%	96.3%	2.53	2.06
+repl.	95.0%	98.8%	95.0%	3.09	1.30
Worst	91.8%	78.9%	92.1%	3.09	2.29
Best	96.1%	98.8%	96.4%	2.22	1.30

Table 4 – Simulation results of the model with information symmetry (worst and best are the worst and best performance registered throughout the whole simulation campaigns).

The search function influences only slightly the system's performances and, surprisingly, with a minimal decrease, that shows a negative interaction of the two factors (the wholesaler negotiating for its orders and the customer looking at different wholesalers for the goods he needs). It is interesting to notice that the introduction

of the replace function cancels completely all the improvements the vertical coordination had brought compared to the basic model.

Lateral coordination model

When a transshipment mechanism is activated within the supply chain, the most interesting result is a further increase of the global sales and of satisfied customers compared to the vertical coordination model (Table 3). However, there is also a further decrease of the percentage of customers satisfied with ready delivery down to the lowest percentages of all cases (and this might be very critical in case of a very competitive market, where customers are not available to wait). Once again the search function does not cause any major change in the system's performances, while the replace function causes a loss in all performance indexes apart from the ready delivery that (obviously) increases.

Information symmetry model

The effect of increasing visibility upwards (visibility of the wholesalers on the supplier's stocks) and downwards (visibility of the supplier on real demand forecasts) causes an excellent increase on sales and percentage of satisfied customers, both to the highest performances measured (see Table 4). The ready delivery percentage is quite good although it is still lower than the one observed in the basic model, while the inventory level and the average waiting time are grown slightly worst.

Again, the search function does not influence the performances apart from an increase in waiting time, while the replace function has the same effect but with lower intensity than what has been observed in the previous cases.

In general one can notice a gradual improvement trend from the basic model to the negotiation based vertical coordination model, to the mixed vertical-lateral coordination model (with transshipment in case of stock-out risk) to a high symmetric visibility model. As it was supposed, the introduction of a replace behaviour in the market decreases the performances measuring profitability. On the contrary, it is quite surprising to notice that a search behaviour has no positive effect on performance. One of the possible explanations behind this effect might be that the search behaviour in customers causes an additional indirect competition between wholesalers belonging to the same supply chain that disturbs the system's dynamics.

Conclusions and remarks

Before concluding, it is important to remark the triple use, common to many multi-agent applications, that can be done of the developed model.

In fact, in first place the model has been developed as a decision support tool for the planning and the management of transactions within a coordinated and integrated supply chain. Therefore the model is able, once the data regarding demand, sales and reservations is inserted, to elaborate in real time supply and replenishment decisions based on vertical and lateral coordination that can assure the best global results in respect of the independence of each element of the network.

At the same time, the model can be used as an off-line simulator with the objective of evaluating the effectiveness of different coordination policies on the system's and on its elements' performances, in order to choose the best policy to implement in a specific supply chain.

A third use of the model is the one most closely related to the intrinsic nature of autonomous agents. The agents working within the model can become, in future applications, independent entities that can perform specific tasks such as supply, replenishment, or placement of orders negotiating with other agents belonging to different elements of the supply chain. This can be realized in a supply chain fully linked in an informative network where the agents can live, move and act exchanging information and executing tasks according to how they have been programmed.

The model described in this paper is known under experimentation in its simulation mode. However, the chances of making it a decision support tool are quite realistic on a short term, while for a use of its agents as autonomous agents on a network there still is some work and some time to go (also for the low level of information integration the actual supply chains implement).

In conclusion, the model has shown very good flexibility and scalability, it has easily adapted to different policies and different structures of supply chain, and it has succeeded in implementing typically hierarchical approaches of vertical and lateral coordination in a non-hierarchical environment such as a non centralized supply chain.

This paper is the result of a joint work of the authors. In particular, S. Cavalieri has produced paragraphs n. 1, 2, 3 4 and 5.1; V. Cesarotti has produced paragraphs n. 5.2, 6, 7 8 and 9.

References

- ARPA Knowledge Sharing Initiative (1993) *Specification of the KQML agent-communication language*. ARPA Knowledge Sharing Initiative, External Interfaces Working Group, working paper.
- Anupindi R. & Bassok Y. (1994) *Centralization of Stocks: Retailers vs. Manufacturer*. Management Science, 45 (2).
- Anupindi R. & Bassok Y. (1988) *Supply Contracts with Quantity Commitments and Stochastic Demand*. In: Tayur S., Magazine M. & Ganeshan R. (eds). Quantitative Models for Supply Chain Management. Kluwer Academic Publishers.
- Anupindi R., Bassok Y. & Zemel E. (1999a) *A General Framework for the Study of Decentralized Distribution Systems*. Technical Report. Northwestern University.
- [Anu99b] Anupindi R., Bassok Y. & Zemel E. (1999b) *Study of Decentralised Distribution Systems: Part II – Applications*. Technical Report. Northwestern University.
- Axsäter S. & Zhang W. (1999) *A joint replenishment policy for multi-echelon inventory control*. International Journal of Production Economics 59, 243-250.

- Barbuceanu M. & Teigen R. (1998) *System Integration through Agent Coordination*. In: Handbook on Architectures of Information Systems. (Bernus,P., Mertins, K., Schmidt, G. (Ed.)), 797-826.
- Brun A. & Portioli A. (1999) *Effective Supply-Chain Co-ordination: an investigation*. In: Proc. of the Second Int. Workshop on IMS, Leuven, 411-420.
- Cavaliere S. (1998) *Studio dell'impiego di architetture multiagente nello scheduling e controllo di sistemi produttivi*. Tesi di dottorato in Ingegneria Gestionale, Milano.
- Chen F. & Zheng Y.S. (1997) *One-warehouse Multiretailer systems with centralised stock information*. Operations Research. Vol. 45, 275-287.
- Cohen M., Kleindorfer P.R. & H.L.Lee (1986) *Optimal Stocking Policies for Low Usage Items in Multi-Echelon Inventory Systems*. Naval Research Logistics Qt. Vol.33, 17-38.
- Das, C. (1975) *Supply and Redistribution Rules for Two-Location Inventory Systems: One Period Analysis*. Management Science, Vol. 21, 765-776.
- Dawson, J.A. (1993) *Issues and trends in European Retailing and their impact on Logistics*. Proceedings of the ILDM National Conference, June.
- Dempsey P. (1999) *New Trends in Rapid Response Manufacturing Logistics*. In: Quick Response in the Supply Chain (Ed. Hadjiconstatinou). Springer-Verlag Berlin, 97-129
- Fernie J. (1999) *Quick Response in Retail Distribution: an International Perspective*. In: Quick Response in the Supply Chain (Ed. Hadjiconstatinou). Springer-Verlag Berlin, 173-187.
- Fox M.S. & Gruninger M. (1998). *Enterprise Modelling*. AI Magazine, AAAI Press, Fall 1998, 109-121.
- Ganeshan R. & Harrison T.P. (1997) *An Introduction to Supply Chain Management. Working Paper*. Department of Management Science and Information Systems, Penn. State University.USA.
- Genesereth M.R. & Fikes R.E. *Knowledge Interchange Format, Version 3.0*. Reference Manual. *Technical report Logic-92-1*. Computer Science Dept., Stanford University.
- Gross, D. (1963) *Centralized Inventory Control in Multilocation Supply Systems*. In: Multistage Inventory Models and Techniques. (H.E. Scarf, D. M. Gilford and M.W. Shelly (Eds.)). Stanford University Press, Stanford California.
- Hadjiconstatinou E. (1999) *Quick Response in the Supply Chain*. Springer-Verlag Berlin.
- Handfield R.B. & Nichols E.L. (1998) *Introduction to Supply Chain Management*. Prentice Hall, Upper Saddle River, New Jersey.
- Hinkkanen A., Kalakota R., Saengcharoenrat P., Stallaert J. & Whinston A.B. (1997) *Distributed Decision Support Systems for Real Time Supply Chain Management using Agent Technologies*. In: Readings in Electronic Commerce (ed. R. Kalakota and A.B. Whinston), Addison Wesley, 275-291.
- Hoadley B. & Heyman D.P. (1977) *A Two Echelon Inventory Model with Purchases Dispositions, Shipments, Return and Transshipments*. Naval Research Logistics Qt..V.24, 1-19.

- Karmarkar, U. (1977) *The One Period, N Location Distribution Problem*. Naval Research Logistics Quarterly. Vol. 24, 559-575.
- Karmarkar U. (1979) *Convex/Stochastics Programming and Multilocation Inventory Problems*. Naval Research Logistics Quarterly. Vol. 26, 1-19.
- Karmarkar U. (1981) *The Multiperiod Multilocation Inventory Problem*. Operations Research Vol. 29, 215-228.
- Kjenstad D. (1998) *Co-ordinated Supply Chain Scheduling*. PhD Thesis. Norwegian University of Science and Technology, Trondheim, Norway.
- Lee H.L. (1987) *A Multi-Echelon Inventory Model for Repairable Items with Emergency Lateral Transshipments*. Management Science, 33 (10), 1302-1315.
- Martin, A.J. (1995) *Distribution resource planning: the gateway to true quick response and continual replenishment*. John Wiley & Sons, New York.
- Narus J.A. & Anderson J.C. (1996) *Rethinking Distribution: Adaptive Channels*. Harvard Business Review, 112-120. July-August 1996.
- Ramakrishnan S. & Srihari K. (1998) *Supply Chain Management – An Overview*. Technical Report. IIEC State University of New York.
- Robinson L.W. (1990) *Optimal and Approximate Policies in Multiperiod, Multilocation Inventory Models with Transshipments*. Operations Research. 38 (2), 278-295.
- Sadeh N.M., Hildum D.W., Kjenstad D. & Tseng A. (1999) *MASCOT: An Agent-Based Architecture for Coordinated Mixed-Initiative Supply Chain Planning and Scheduling*. In: Proc. of the Third Int. Conference on Autonomous Agents, Seattle WA.
- Silman M. (1999) *A Quick Response Model and its Applicability in the UK*. Quick Response in the Supply Chain (Ed. Hadjiconstatinou) Springer-Verlag Berlin, 7-10.
- Smith R.G. (1980) *The contract net protocol: high-level communication and control in a distributed problem solver*. IEEE Transactions on computers, V.C29, No.12, 1104-1113.
- Svoronos A. & Zipkin P. (1988) *Estimating the Performance of Multi-Level Inventory Systems*. Operations Research, Vol. 36, 57-72.
- Swaminathan J.; Smith S. & Sadeh-Konieczpol N. (1997) *Modeling Supply Chain Dynamics: A Multiagent Approach*. Decision Sciences, April.
- Vollman T. E. Berry W.L. & Whybark D.C. *Manufacturing Planning and Control Systems*. Irwin, Homewood, IL.

P. VAN BAEL
M. RIJCKAERT

*Specific knowledge of the job shop scheduling problem
incorporated in local search, how good is it?*

K.U. Leuven-Chemical Engineering Department

Abstract— In general, it is known that scheduling algorithms need to use specific knowledge to perform competitive. In solving the job shop scheduling problem with a simulated annealing algorithm, we show that incorporation of specific knowledge into neighborhood structures is not always effective. On the other hand, an improvement can be realized if one incorporates repair strategies to restore infeasible schedules. The results conducted from well known JSSP benchmarks indicate how a good balance between effectiveness and efficiency can be found.

Local search

Since the last decade, local search techniques have been developed and successfully applied to solve optimisation problems of different kinds. The idea of a local search method is that within the neighborhood of a good solution a better solution can be found. Minor changes to local decisions, which constitute the current solution, might improve the current solution. In other words, the current solution contains knowledge that can be used for a new solution, which is probably better than creating a new solution randomly.

A local search method is general in the sense that it can be used if one can define

- (i) a configuration of a solution,
- (ii) a cost function to distinguish solutions
- (iii) a neighborhood structure that defines a transition between solutions to guide the search

The configuration of a solution is in fact the way the real problem is modelled. It defines the search space, which is mapping of the solution space of the problem to be solved. The configuration of this solution can generally be represented as a set of decision variables where the solution is formed by a sequential value assignment of the variables.

The local search method explores the search space by starting with a solution x . Once an assignment took place, a cost function can be evaluated. A neighborhood structure $N(x)$ defines a set of neighbor solutions, where every solution $x' \in N(x)$ is evaluated through the same cost function. The neighborhood structure usually implies small variations on the solution x , hence a local search or neighborhood search. One of the solutions x' becomes the new solution x so a move made from x to x' . Continuing making moves explores the search space. The exploration continues until a termination criterion is met. Roughly speaking, a local search method starts with an initial solution and then continuously tries to improve by searching neighborhoods as is illustrated in Fig. 1. Consequently, the neighborhood structure is the key to the success of the local search method.

Neighborhood structures define the size and content of the neighborhood set of a solution. It should be defined in a way that the better solutions around the current solution are subject to be selected for the next solution. If the neighborhood structure

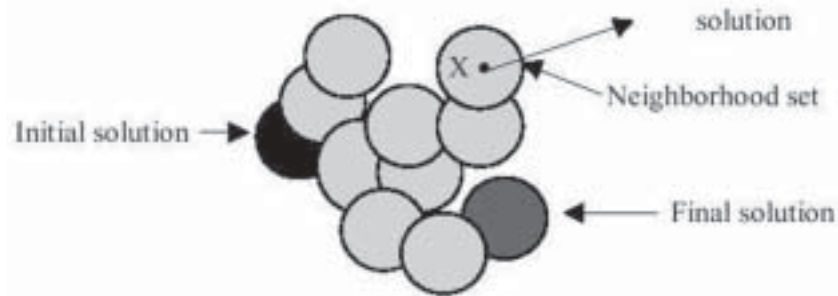


Fig. 1 – Moving from an initial solution via neighbors to the final solution.

would generate the total search space, evaluating all neighbors is similar to a total enumeration of the search space. Clearly, this is to be avoided for large combinatorial problems where search spaces are larger than billions of solutions. Therefore, the subset of solutions within the neighborhood must be smaller. However, this causes local optimality whenever a solution is superior to every solution within its neighborhood set. This means that the neighborhood structure defines the difficulty of exploring the search space itself because it creates the number and distribution of local optima. To transcend local optima, non-improving moves can be allowed, which is the basic idea of modern local search techniques.

Every variant of local search is either a variant of the moves or a variant of the neighborhood structure. The variants of moves is determined by the chosen local search method while the variants of neighborhood structure depend on the configuration of the problem. The success of the different NS's cannot yet be justified by any theory, it can only be measured by experimentation and analyses of the results. Nevertheless, a few considerations can be helpful in constructing a good neighborhood structure:

- *Correlation*: A neighbor solution should be highly correlated with its original solution to ensure a local search is performed.
- *Feasibility*: Neighborhood structures should construct feasible solutions. If possible, the search should be restricted to the domain of feasibility in order to avoid either expensive repair procedures or either inefficient search.
- *Improvement*: The neighborhood structure should be emphasizing selection of good knowledge to improve the chance of a good move. Specific knowledge may be incorporated into the neighborhood structure to obtain this.
- *Size*: The neighborhood structures should create an optimal average size of a neighborhood set. A small number may results in an ineffective convergence speed as well as being trapped in early stages of the search space. To the opposite, a large

number may lead to an inefficient convergence speed as well as getting trapped in very good local optima from which escaping can become nearly impossible.

- *Connectivity*: It should be guaranteed that there is a finite sequence of moves leading from a random solution to a global optimal one. Otherwise, promising areas of the search space may be excluded from the search space.

- *Topology*: The neighborhood structure causes the possible differences of solutions in the neighborhood set and hereby the differences in makespan. If these differences are large within a set, a very rough search space arises with many local optima, which makes it difficult to locate the global optima. On the contrary, if the differences are small, it would be easier to navigate to a global optima.

It is not evident to fulfill every consideration when constructing a neighborhood structure because either no computable measures exist to verify them or conflicting considerations makes it impossible. Nevertheless, depending on the problem at hand, some should get more attention than others. Two of them are rather easy to realize namely size and improvement. A small size as well as a good improvement should both increase efficiency and effectiveness since the convergence speed is effected by them. In general, it is believed that the smaller the size and the more specific knowledge are included, the better the performance of a local search variant.

A popular algorithm that fits in the class of neighborhood search philosophy is simulated annealing. A problem domain where SA has been applied with moderate success is job shop scheduling problems. After a description of the JSSP problem and the SA method, a comparison will be made of different neighborhood structures. It shows the reason of its moderate success and the relation between size and improvement of neighborhood structures.

Job shop scheduling problem

The job shop scheduling problem (JSSP) is a well known NP-complete combinatorial optimisation problem. It has been used repeatedly as a least discrepancy yet simplified model of real world production planning systems. A scheduling problem assigns activities to shared and limited resources and specifies its start time. The JSSP consists of m resources and n jobs, where every job has m activities each on a different resource. The activities within a job have to be completed in a specific sequence and every activity has to be finished prior to start the next activity. The optimal schedule is the one with the smallest makespan (= the latest completion time of all activities). As an example, a Gantt chart is shown in Fig. 2 (*next page*), where 4 jobs consisting of each 3 activities are assigned over 3 machines.

A schedule has also a *critical path*. It is the path formed by a chain of critical activities going from the start point to the latest completion time of the schedule. Every critical activity on the critical path has a single possible earliest start time and latest stop time and is assigned to the activities as start and stop time. Moreover, every move of an activity within the chain will change the makespan value. The critical path can be subdivided into *critical blocks*. Every critical block contains a

subsequence of the critical path where every activity belongs to the same resource (Fig. 3).

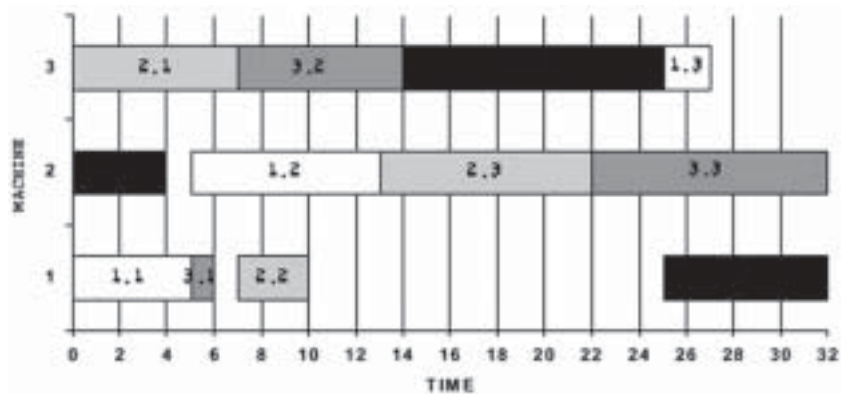


Fig. 2 – An 3x4 JSSP.

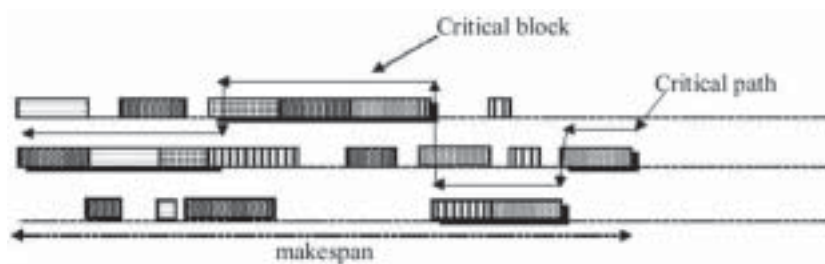


Fig. 3 – A schedule with its critical path consisting of critical blocks.

Applying SA to JSSP

A general simulated annealing algorithm is shown in Fig. 4. The algorithm is run as follows: initially the control parameter ' c_0 ' is very high and an initial solution ' i_{start} ' is generated. An inner loop iterates until equilibrium is reached, symbolized by ' L_k '. Within this inner loop neighborhood solutions are evaluated. Whenever a neighborhood solution is accepted, the current solution is replaced by it. An outer loop recalculates the control parameter ' c_k ', which is decreased according to a cooling schedule and the equilibrium length ' L_k ' at that stage. A stop criterion determines when the algorithm stops. With this procedure and the Metropolis criterion, large deteriorations of the solutions are accepted in the beginning. Gradually the acceptance criterion is tightened and large deteriorations are rejected. Finally, as the control parameter becomes very small, only improvements are accepted which converts the algorithm into a Hill climbing method. The algorithm has many parameters, method parameters as well as problem

parameters. The method parameters explicitly belong to the simulated annealing method. We have the initial control parameter 'c0', the decrement function 'ck = f(ck-1)', the equilibrium calculation 'Lk' at every 'ck', and the acceptance criterion, in this case the metropolis criterion. The problem parameters are problem dependent, namely the neighborhood structure 'NEIGHBOR' and the way a solution is configured, here 'i' and 'j'. The ways these parameters are instantiated depend on the problem to solve.

```

INITIALISE (istart, c0, L0, k, i)
repeat
  for L = 1 to Lk do
    begin
      j = NEIGHBOR (i)
      if (f(j) NOT in MEMORY(T))
        if (f(j) ≤ f(i)) then i = j
        else if ( exp(  $\frac{f(i) - f(j)}{c}$  ) > random[0,1] ) i = j
      if (∀ j : f(j) ≥ f(i)) i = jbest
    end
  k = k+1
  CALC(Lk)
  CALC(ck)
  if (RESTART(kreset) i = jbest & ck = c0)
until stop criterion

```

Fig. 4 – General simulated annealing algorithm.

The modified SA method definitely needs specific knowledge incorporated into its neighborhood structure to make it powerful. The enhancements will be described in following paragraphs. Besides specific knowledge, the enhanced sa-method incorporates a memory 'T' and a reintensification strategy 'RESTART(k_{reset})'. The memory is a cycle detector as mentioned earlier. It stores the last T elements that were selected to become the current solution. If an element in the NS set has a same objective value it is not accepted as the current solution. To improve the efficiency, also an implicit memory is assumed to ensure no element is selected twice out of the NS set and if every element is selected ones, the best neighbor of the current solution is chosen. Reintensification is meant to refocus on previous good search spaces or total new search space. If the best solution so far is not improved after 'kreset' iterations then i.e. the best solution found so far becomes the current solution again and the initial c0 is given to the control parameter at that stage.

Neighborhoods structures

As said earlier, two parameter sets can be distinguished, namely the method and the problem parameters. The method parameters are very general and can be used

to cover a large range of problems. In this research, they are considered as less important. We are going to focus on the problem parameters, namely the neighborhood structures.

The use of neighborhood structures allows SA to explore the search space. Different neighborhood structures are developed through the years to enhance the performance of local search techniques. Mainly a reduction of the size of the NS applied is focused, to avoid that SA should select the non-improving solutions. Hence to increase the probability of selecting an improving solution. Also important is the use of NSs that do not change the configuration into a final infeasible schedule. Although we could insert an extra cost parameter in the objective function, which avoids further selection, we do not use this option since many infeasible solutions can be constructed by simple NSs. Next we will describe the NS's developed that decreased the size of the neighborhood structures but also ensures that feasible final schedules will be constructed. This feasibility will be realized by using NSs, which do not allow making configurations, which contain infeasible precedence constraints. The following NSs can be distinguished:

- N1: Insert: Replace one activity within the sequence
- N2: Swap: Exchange two activities within the sequence
- N3: Swap adjacent activities within the sequence
- N4: Replace an activity within a critical block
- N5: Exchange an activity within a critical block
- N6: Exchange adjacent activities within a critical block
- N7: Insert an activity at the head (tail) of the critical block it belongs to
- N8: Exchange activities within a critical block of which at least one is the head (tail)
- N9: Insert an activity at the head (tail) of the critical block it belongs to (*only first or last two*)

As you can see in the Gantt chart in Figs 2 or 3, the activities are spread out on their respective machine within a period of zero until the makespan of the solution. Improving this solution could be done by deleting one activity within the configuration and insert it at another place. The solution is altered and the original Gantt chart will look different afterwards. The distance of the replacement could be kept small to create small changes. This neighborhood structure is called N1. The neighborhood set contains a total of $(n-1) \cdot 2$ different N1 moves. A second neighborhood structure N2 exchanges two activities within the sequence. It can have $n(n-1)/2$ number of different possibilities. The total amount of different solutions within a neighborhood structure of N1 and N2 depends on the dimension of the problem. A factorial increase in the number of neighborhood solutions occurs with these NSs. The probability to pick the one with the best improvement becomes small. Actually too small, to be selected from the set and to give the SA a good performance. Even if the total set would be evaluated, it would be too time consuming. These NSs give the method an inefficient convergence speed and a local improvement, which is not capable of reaching a global optimum, unless it was initialised close to the global optimum. Therefore, even the use of a tabu search method is not capable of using these neighborhood structures efficiently. They are

to naïve, but problem independent. Hence, non-specific knowledge containing neighborhood structures will be inefficient and ineffective to be used with SA.

Including specific knowledge is a way to improve the effectiveness of the neighborhood structures. It is based on a tighter selection of possible neighbors due to specific knowledge from the JSSP. Next, a range of improvements will demonstrate that if the amount of specific knowledge increases, the size of the neighborhood structure drops drastically.

A first improvement could be exchanging two adjacent activities giving rise to neighborhood structure N3. Adjacent means produced after each other without delay. The fact could indicate a possible bottleneck because they concur for the same period at a machine. The number of adjacent activities depends on the solution. The average number of N3 elements will not be that much. A more important fact is that this number will be affected rather linearly in the size of the jobs instead of exponential in problem size. However, in the worst case, still $n(m-1)$ possible elements will be in the set. Hence the probability to select an N3 element that improves the solution will grow slightly because the size of the neighborhood set is still too big. The change to have a deteriorating or even unchanged improvement is still more likely.

A second improvement is far more specific and based on particular knowledge about the makespan criterion, which is to be optimised. The makespan criterion can only be improved if an exchange occurs of activities belonging to the critical path. The exchanged activities belong to the same critical block. Depending if one uses N1 or N2 to realize the change, the neighborhoods are called N4 and N5 respectively. The neighborhood structures N4 and N5 have a tremendous decrease in size. In addition, the sizes depend on the critical path length, which increases linear in the number of jobs. This reduction can be even further reduced. By experiment, one saw that only if one of the activities, which is replaced, belonging to a critical block is the head or tail then an improvement is possible. Using this information, the neighborhood structures N6 and N7 were born. The decrease in reduction is spectacular. These neighborhood structures are very powerful. They keep the size of the NS so low that the probability to pick the improving solution becomes real. A last neighborhood structure N9 exists which is similar to N6 but now only the first or last two activities within the critical block are exchanged. One can see that these last NSs' are possible using very specific knowledge of the JSSPs critical path.

A last improvement is only possible regarding the efficiency. Whenever a schedule becomes compacted, the probability that a replacement of an activity in the sequence jumps over one of its activities from the same job, becomes real. Normally, these solutions become infeasible due to violated precedence constraints. Moreover, they are neglected for further use. This is inefficient. One could shuffle the conflicting activity to the front or rear as needed with an N1 neighborhood structure, to restore the infeasibility. In this way it is not inefficient. Moreover, it breaks down barriers, which makes the search space smoother. This enhancement can be used on every previously mentioned neighborhood structure.

To compare the power of the neighborhood structures, a simple experiment was set up. A described simulated annealing algorithm was used to solve the 10x10 mt10 JSSP and the 20x10 la27 JSSP, two tough JSSP problems. The algorithm used the following method parameters: $N_{iterations} = 20000$, $L_k = 10$; control schedule = linear; $c_0 = 30$; i_{start} = random. The basic algorithm is enhanced with memory features. A short-term memory and a long-term memory as distinguished. The short-term memory remembers NS elements visited while exploring the neighbors of the current solution. This avoids being trapped at very strong local optima, where escaping is impossible with the current control parameter value. If every N_s element is tried and escaping was not possible, the best solution is accepted as the next current solution. The long-term memory is needed to avoid going back to the previous strong local optima and is called a cycle detector. The cycle detector here forbids an objective value to appear within the next 10 iterations of its appearance. Several neighborhood structures were applied, namely N_1 , N_4 , N_6 , N_9 and each once with and once without the repair strategy N_1 . Every different scenario, in total 16, was run 40 times. Table 1 & 2 summarizes the results where average, best and worst makespan over 40 runs and average number of neighborhood elements is listed as the time needed to complete one run.

Neighborhood Structure	Without cycle detection			With cycle detection				Size of NS
	Average	Best	Worst	Average	Best	Worst	T*	
N_1	1029	992	1079	1025	941	1066	5	9801
N_1 +repair	964	938	994	966	939	994	7	9801
N_4	967	938	997	960	937	989	29	45
N_4 +repair	954	930	978	947	935	967	31	45
N_6	967	938	990	960	938	986	15	14
N_6 +repair	967	944	988	959	938	988	15	14
N_9	986	939	1130	959	930	1012	9	6
N_9 +repair	978	938	1040	965	938	992	9	6

(*)Time for one run(s) on Pentium II 450 Mhz

Table 1 – Neighborhood structures applied on the 10x10 JSSP.

Neighborhood Structure	Without cycle detection			With cycle detection				Size of NS
	Average	Best	Worst	Average	Best	Worst	T*	
N_1	1479	1440	1514	1483	1453	1523	14	39601
N_1 +repair	1457	1430	1498	1460	1422	1492	18	39601
N_4	1318	1306	1345	1311	1297	1331	150	103
N_4 +repair	1291	1275	1310	1280	1269	1308	179	103
N_6	1327	1285	1388	1316	1296	1356	59	21
N_6 +repair	1321	1299	1349	1315	1287	1329	54	21
N_9	1349	1296	1415	1314	1290	1393	23	6
N_9 +repair	1338	1293	1417	1295	1274	1312	27	6

(*)Time for one run(s) on Pentium II 450 Mhz

Table 2 – Neighborhood structures applied on the 20x10 JSSP.

Conclusion

The results of the experiments listed in table 1, 2 illustrate several facts. First, the sa algorithm has a stochastic nature which results in a cycle effect. A considerable improvement is made when the cycles are reduced. Second, the neighborhood structures are clearly effecting the results. The tables illustrate an incredible decrease in neighborhood size if one incorporates specific knowledge. The reduction is larger for bigger JSSPs. The neighborhood sets without specific knowledge grow exponential while the one with specific knowledge have an almost steady or linear increase. The computation times illustrate that if one wants to move to a neighbor, which improves the current solution, then it takes a much longer time if the neighborhood set is bigger. This increases the effectiveness. The efficiency increases when additional specific knowledge is incorporated. However, a too large reduction of the size decreases its effectiveness again. Probably, because the experimental conclusions are not valid on our configuration of the problem (Note that we do not use disjunctive graph representation). A more disastrous effect occurs involving the worst case scenario. The worst solution tends to increase as high as without incorporation of specific knowledge, which is not tolerable. From the tables, it is also clear that the repair strategy improves the results. Especially the worst cases are improved.

The clear winner is the neighborhood structure N4 with the repair facility. It has a good best and worst makespan within affordable time limits. Unfortunately, the results show clearly that the basic method is not capable of finding optimal schedules. A pure lucky initial solution leading straight to a global optimal is the only change of obtaining it. One can conclude that the smallest neighborhood size is not necessarily the best choice and that the basic simulated annealing method, even enhanced with a cycle reducer and specific knowledge incorporated is not powerful enough.

References

- Van Laarhoven P.J.M. & Aarts E.H.L. (1988) *Simulated Annealing: theory and applications*. Reidel, Dordrecht, The Netherlands.
- Downsland K. (1993) *Simulated annealing*. In: Reeves chapter 2.
- Lundy M &, Mees A. (1986) *Convergence of an annealing algorithm*. Math.Prog. 34, 111-124.
- Reeves C.R. *Modern heuristic techniques for combinatorial problem*. Blackwell Scientific Publications: Oxford.
- Aarts E. & Korst J. (1989) *Simulated Annealing and Boltzmann Machines*. Wiley & Sons Inc. Chichester.
- Grabowski J., Nowicki E. & Zdrzalka S. (1986) *A Block Approach for Single Machine Scheduling with release dates and due dates*. European Journal of Operational Research, 26: 278-285.
- Mute J.F. & Thompson G.L. (eds) (1963) *Industrial Scheduling*. Prentice-Hall, Englewood Cliffs, N.J.

- Rayward-Smith V. J., Osman I.H.; Reeves C.R. & Smith G.D. (1996) *Modern heuristic search methods*. Wiley & Sons: Chichester.
- Vaasens R.J.M.; Aarts E.H.L. & Lenstra J.K. (1996) *Job Shop Scheduling by Local Search*. *INFORMS Journal on Computing* 8, 302-317.
- Yamada T.; Rosen B.E. & Nakano R. (1994) *A simulated Annealing Approach to Job Shop Scheduling using Critical Block Transition Operation*. In: Proc. Of IEEE ICNN (IEEE, Florida, 1994), 4687-4692.
- Yamada T. & Nakano R. (1996) *Job-Scheduling by Simulated Annealing Combined with Deterministic Local Search*. In: *Meta-Heuristics: Theory & Applications*, edited by Osman I. H. & Kelly J. P., Kluwer Academic Publishers, Boston.

HANS-HERMANN WIENDAHL
ENGELBERT WESTKÄMPER

*Manufacturing in turbulent markets effects
on production planning and control*

Institute of Industrial Manufacturing and Management (IFF),
University of Stuttgart, Germany
Fraunhofer Institute for Manufacturing Engineering and Automation (IPA)

Abstract — Strong market fluctuations require an enlarged understanding of PPC and how it is designed, since the conventional PPC vision of a steady flow of orders independent from the external and internal requirements is no longer realistic. The paper introduces an analogy between physical and manufacturing phenomena to classify the logistic system behavior. This forms the theoretical fundament to allow the external und internal logistic positioning of a manufacturing company.

Keywords — manufacturing planning and scheduling (PPC), turbulence, logistic system behaviour.

Introduction

Management publications describe turbulence as a problem of planning reliability, an objective which has become increasingly difficult to achieve. But is it always fair to complain about the turbulence of markets? Or is it possible that the turbulence one perceives also results from the fact that enterprises assume dependability and predictability where there is none [Mintzberg 1994]?

But what does it mean in practice? The exemplary analysis shows a boring mill group as the bottleneck of the parts fabrication segment. This machine group determines the delivery behavior to the assembly segment as an internal customer:

(i) From an order-oriented perspective it is typical that the distribution of lead times leans strongly to the left and is accompanied by a high percentage of express orders (Fig. 1 – *next page*). Accordingly, a mean value can be calculated but represents only a small proportion of order lead times. Since the acceptable schedule tolerance in this case is ± 2 days, only 10% of orders are delivered punctually compared to planning based on mean values.

(ii) From a resource-oriented perspective, the throughput diagram is characterized by big variations in output (Fig. 2 – *next page*). The planned capacity was on average well enough utilized during the reference period. This, however, applied not to shorter intervals since the backlog was steadily increasing to 2000 hrs. Strikingly, the inventory fluctuates because of a poor coordination of input and output.

From a conventional PPC point of view, the above-described situation would be analyzed as a lack of input control, i.e. an insufficient order release. A more constant input control would enable more reliable logistical processes with stable mean values and little variations, thus ideally representing a steady flow of arriving and processed orders.

A turbulent environment calls for a broader interpretation: input and output should meet short-term market demands. The output curve shows that the company

was essentially able to compensate fluctuations, even though with a time lag occurred. The main problem was the logic of the MRP software –being based on mean values– which could not map the changed due dates caused by a strongly varying demand.

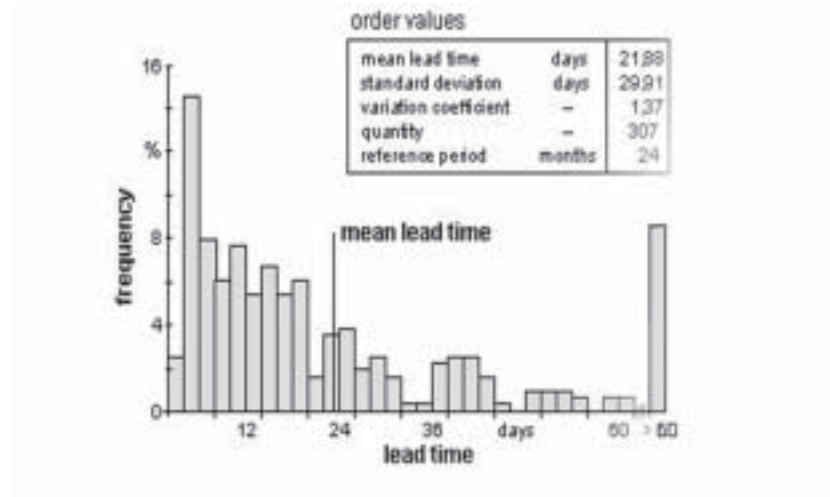


Fig. 1 – Logistic behavior of a boring mill group: order-oriented perspective.

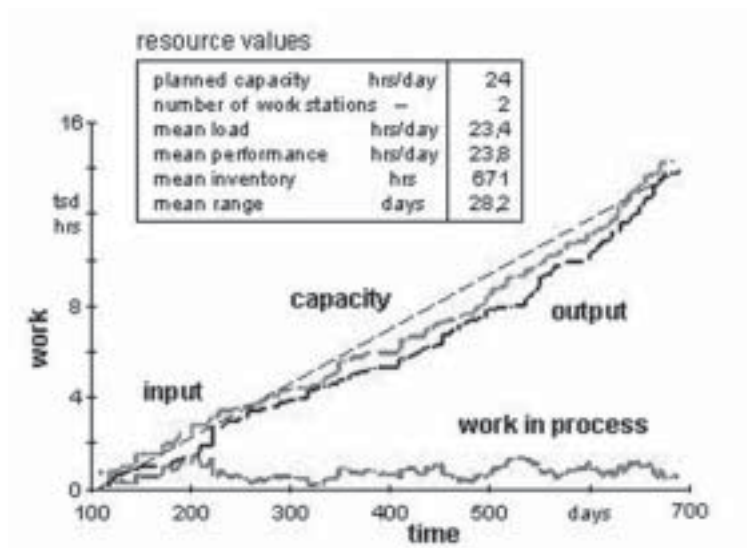


Fig. 2 – Logistic behavior of a boring mill group: resource-oriented perspective.

When the delivery date promised to the internal customer, i.e. the assembly segment, became more and more uncertain, the persons responsible for planning

and control (i.e. the foremen) decided to respond by “manual control“, which was the right thing to do from their point of view.

The foremen actually mastered the market fluctuations using a manual intervention. But this results in a the failure of the conventional PPC approach (i.e. planning based on mean values). This also affected the planning software and led to a vicious circle with poorly accepted PPC system dates. Often such a procedure is the reason why the expenditure for manual order control accounts for up to 90% of the complete expenditure on PPC staff. A better alternative would be to introduce a suitable PPC method.

Regarding PPC, it is obvious that circumstances are ideal if capacity and market demand are in balance and enable a steady flow of orders. Reality, however, is often different and the above-described phenomena cannot be avoided. In order to assess the consequences for PPC, the following two steps are necessary:

- (i) First, the described phenomena have to be closely analyzed using an analogy from physics.
- (ii) Second, the findings must be applied to the design of PPC.

Turbulence and PPC

The Causes of turbulence can be distinguished into external changes of the market and internal changes of the enterprise. Furthermore, turbulence can be perceived in two ways – *subjectively* by those affected and *objectively* by measurement. Planning is used to identify *turbulence germs* such as market fluctuations, whereas control focuses on malfunctions and modifications made by customers (Fig. 3) [Westkämper et al. 2000].



Fig. 3 – Aspects of Turbulence.

The *core PPC task* is to *link items, resources and processes to orders*. The focus of PPC is the internal order progress, because:

(i) Schedule reliability in manufacturing largely depends on whether or not planning will succeed to create realistic, i.e. *feasible production* plans.

(ii) To create realistic production plans, it is important to take the *manufacturing behavior* into consideration, i.e. planning forecasts the logistical behavior.

This requires a closer look on the area of objective, internal turbulence. The aim is to understand the described phenomena between the planned and actual order progress.

Defining turbulence

From the study of physical phenomena knowledge from physics, especially the mechanics of fluids, can be analogous applied to the area of manufacturing. Turbulence in physics means that the macroscopic behavior of a fluid is not equivalent to the microscopic movement of individual particles [Massey 1989].

Accordingly, the *definition* of internal, objective *turbulence in manufacturing* is as follows [Wiendahl et al. 2000]: Turbulence in manufacturing means that the current position of individual orders cannot be derived from the mean order progress, i.e. not from a plan that is based on mean values. Hence conclusions drawn from individual results at the control level are also no longer valid for corresponding mean values. The analogy from physics explains, why:

(i) the strain involved in PPC rises with increasing turbulence.

(ii) turbulence thwarts the classic interaction of planning and control.

Turbulence germs in PPC

In physics, turbulence germs are considered as triggering turbulence. In manufacturing they are systematized according where they originate (external or internal causes) and within what area of PPC (planning or control) they occur [Wiendahl et al. 2000]. The differentiation between market fluctuations and modifications by customers takes place in order release (Fig. 4).

The most obvious triggers of turbulence are input and planning parameters varying over time. Fluctuations in order and production quantities increase turbulence in a company. If parameters are more strongly fluctuating the demands made on planning and control and on manufacturing are growing. However, these considerations do not reveal if a company is able to cope with these fluctuations and to what extent.

Another turbulence germ is the way things are done. Modifications, failures and exceptions are the order of the day in most manufacturing companies. And for specific order situations these might even occur systemic instabilities. Usually its only a small percentage of orders than can be handled according planning. Turbulence increases if the production has to cope with additional modifications required by the customer. Describing the deviations from planning allows to identify to what extent a company has reacted “correctly” of planning.

The third important turbulence germ are modifications of manufacturing goals. In a concrete example, the management realized that inventory costs had increased

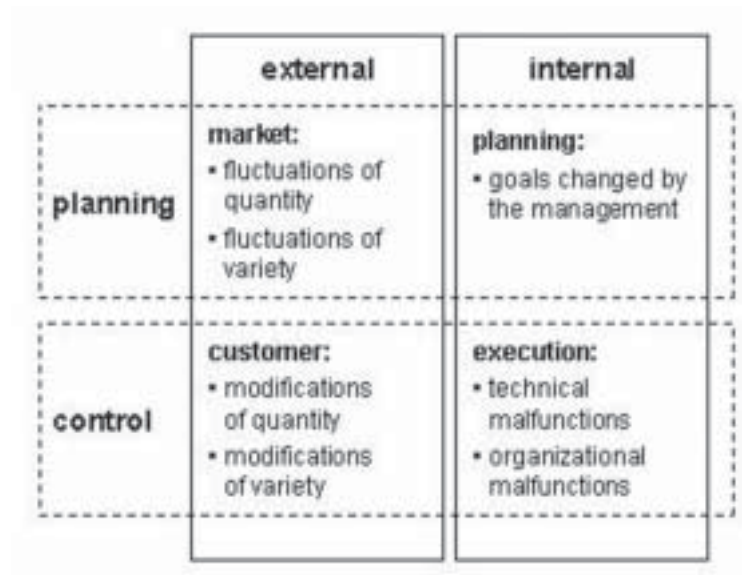


Fig. 4 – Turbulence Germs in PPC.

and decided to change the goal from on-time schedule performance to low inventories. In effect, the annual average output did correspond with the original planning but the semi-annual output quantities differed. Actually, production load during the first six months had been very low (to cut the inventory of finished products), whereas in the second half of the year output nearly had to be doubled. The efforts to accomplish this were immense and, in this case, not caused by market fluctuations.

The appearance of turbulence

It is not only the fact that turbulence germs exist, but the ratio of demand to capability that defines when turbulence occurs. There are three requirements to be distinguished (Fig. 5 – next page):

(i) *Response Requirements*: Do heterogeneous delivery times (external view) require heterogeneous lead times (internal view)?

This applies if the minimum delivery time required by the customer is lower than the mean lead time of production orders (if necessary, consider subcontracting because of speed).

(ii) *Flexibility Requirements*: Does fulfilling customer demands require running the manufacturing system to its capacity limits?

This applies if the market-required fluctuation of quantity and variety exceeds the available capacity within a certain period (if necessary, consider subcontracting because of capacity).

(iii) *Tolerance Requirements*: Do internal aspects require turbulent behavior, i.e. heterogeneous lead times?

This applies if the market-required planning tolerance is lower than the realizable distribution of lead times, e.g. because order priorities differ due to the different delivery times or because the setup is to be optimized at the bottleneck to avoid an investment into a new machine.

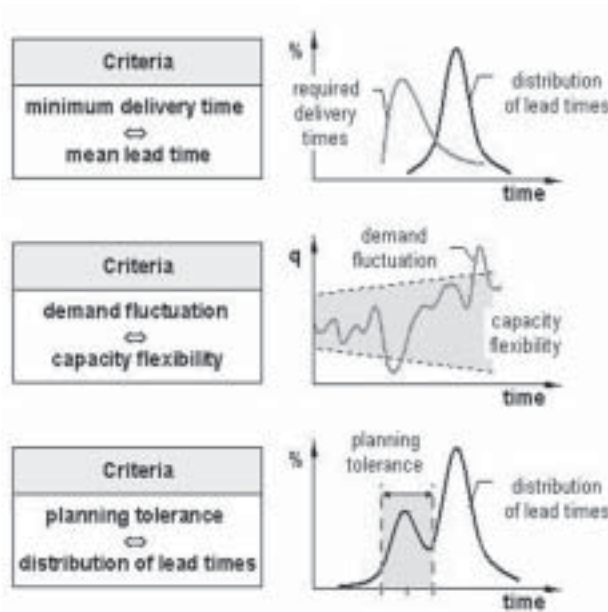


Fig. 5 – Relevant Circumstances for Turbulence.

Experience shows that, in the end, big variations of lead times ?entailing a higher turbulence level? for example through different order priorities (scenario 1) or optimizing setup times (scenario 2) cannot be avoided.

This analysis explains why some authors accentuate on a fundamental difference between planning and control [Valckenaers 1999]. Standard planning algorithms use generalized models which imply an idealized mean flow of orders based on an abstract flow sheet. At the execution level this flow of orders splits up in particular operations and single events. Therefore the control tasks needs order individual models so as to schedule and trace particular orders.





PPC approaches in a turbulent environment

The task of the PPC design is the weighting of the PPC goals. This goal assessment results in the so-called “logistical positioning” based on Produktionskennlinien (logistic operation curves). The latter helps to quantify the operations scheduling dilemma existing among inventories, lead times and utilization. This assessment method is generally accepted today both in theory and in practice. It is used to calculate target inventories, utilization and lead time [Lödding et al. 2000]. In a

stable environment, this one-dimensional evaluation is sufficient to minimize the variation of target values. Under such circumstances the classic mean order progress PPC of MRP logic can be applied.

Classification of logistic behavior

The presented definition of turbulence shows that the principal strategy of minimizing the variation of target values cannot be applied to a turbulent environment. Thus, this variation becomes a PPC design parameter of its own adding to the one-dimensional inventory assessment in a stable environment. Accordingly, answering how much internal turbulence is caused by heterogeneous external demands becomes another task of logistic positioning.

Class of Turbulence	Description
 laminar	Steadily flowing river <ul style="list-style-type: none"> • Low variance of lead times enables «mean order progress PPC».
 slightly turbulent	<ul style="list-style-type: none"> • Production segments are coordinated through mean due dates.
 severely turbulent	Mountain torrent <ul style="list-style-type: none"> • High variance of lead times requires «individual order progress PPC».
 completely turbulent	<ul style="list-style-type: none"> • Production segments are coordinated through individual due dates.






 production area
  turbulence

Fig. 6 – Order-oriented perspective: Turbulence Classes in Manufacturing.

WIP Level	Description
 high	High water level <ul style="list-style-type: none"> • High relative WIP level warrants high utilization of machines. • Low mean speed of orders.
 moderate	Moderate water level <ul style="list-style-type: none"> • Moderate relative WIP level leads to moderate utilization of machines. • Moderate mean speed of orders.
 low	Low water level <ul style="list-style-type: none"> • Low relative WIP level leads to low utilization of machines. • High mean speed of orders.



 relative WIP level
  technical and organizational malfunctions

Fig. 7 – Resource-oriented perspective: Relative WIP-Level.

In accordance with physics, manufacturing distinguishes four different classes of turbulence. This understanding represents an *order-oriented perspective* of manufacturing (Fig. 6):

(i) In case of a low lead time variations, it is possible to coordinate the production segments through mean due dates. Thus, a laminar or slightly turbulent logistic behavior enables a *mean order progress PPC*.

(ii) In case of a high lead time variations, the production segments must be coordinated through individual due dates. Thus, a severely or completely turbulent logistic behavior requires an *individual order progress PPC*.

On the other hand, a *resource-oriented perspective* of manufacturing is needed. The classic philosophy of PPC tried to keep turbulence away from manufacturing. In a turbulent environment, however, it becomes a factor of performance to be able to control the external turbulence germs, i.e. 'market fluctuations' (planning level) and 'modifications by the customer' (control level) through the conscious design of PPC in manufacturing. Thus, the *WIP* (work in process) *level* remains the *central set point of PPC* (Fig. 7):

Firstly, the *WIP level* determines the *speed of orders* and the utilization of machines. The logistic operation curve describes the relationship between the target values [Lödding et al. 2000].

Secondly, it determines the *character of turbulence*, described by the turbulence germs, which has the strongest impact on logistic behavior. The classification depends on the height of the *WIP level*. In case of a stable environment and a moderate relative *WIP level* between 200 and 300%, the flow of orders becomes laminar [Lödding et al. 2000; Wiendahl et al. 2000].

Thirdly, the *WIP level* determines the *interaction between planning and control*: In case of a high or moderate *WIP level*, planning and control are linked by the order flow (macroscopic link). For a high *WIP level*, the order release determines only the input (order flow), for a moderate level it controls both input and output (order flow). If the *WIP level* is low, planning and control are linked by single orders (microscopic link), that means order release determines input and output of individual orders [Wiendahl et al. 2000].

This shows that the scope of the discussion about how inventories can be assessed is too narrow. In a turbulent environment it might be sensible to keep higher inventories in process and in stock to be able to compensate the market-related fluctuations of quantity and variety.

This is an important insight with regard to the PPC target system. So far, the company objectives 'low inventory level' and 'high capacity utilization' of the classic PPC target system seemed to make sense. Since they affected the logistic costs, attaining these objectives contributed to corporate cost reduction. Higher inventories in process and stock, however, help to reduce time-to-market, so that the conventional company objectives have to be reassessed in a turbulent environment.

Turbulence portfolio

The 'variation of lead times' (order-oriented perspective) and the 'WIP level' (resource-oriented perspective) specify the internal logistical positioning in a turbulent

environment. Thus, it suggests itself to use both target values for classifying a production system or its work centers. The target values can be represented in a system of coordinates called turbulence portfolio (Fig. 8).

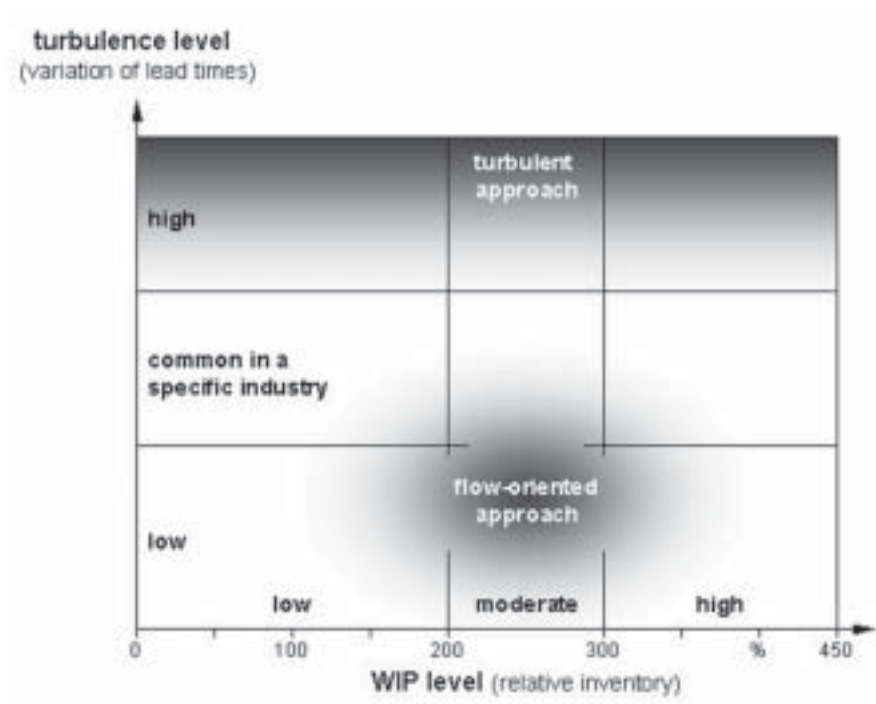


Fig. 8 – Order-oriented perspective: Turbulence Classes in Manufacturing.

From a PPC perspective, the classic solution is the flow-oriented approach based on mean values. Hence it is necessary to determine what *internal conditions* must be fulfilled to realize the flow-oriented approach. From an internal resource-related perspective a *relative WIP* between 200 and 300% enables a laminar flow of orders [Lödding et al. 2000]. From an order-oriented perspective it is possible to check if the approach was actually put into practice. A relevant internal indicator for turbulence is the *variation of lead times*. Additionally, interoperation time (queue time) or operation time can be considered. In accordance with the definition of turbulence three cases are defined: 'low' variation means laminar flow of orders, 'common in specific industry' means slightly turbulent, 'high' variation means severely or completely turbulent.

The turbulence portfolio expands the logistic evaluation of the logistic resources portfolio. The latter compares the features 'lead time percentage' and 'relative WIP level' of individual work centers [Lödding et al. 2000]. Mean values evaluate the logistic positioning of a manufacturing system. This analysis allows to calculate the potential lead time in a stable environment.

Flow-oriented and turbulent approach

The easiest strategy from a planning-related perspective is the *flow-oriented approach*. It tries to afford a stable internal flow of orders despite external market fluctuations (Figs 8 & 9 top). A steadily flowing river means a low variation of lead times and enables a “mean order progress PPC”. Another advantage being the fact that the MRP logic used in current PPC software also assumes a steady order flow.

From a control theory point of view two control loops are required [Wiendahl/Westkämper 2001]:

(i) The WIP controller manipulates the input and controls (observes) the inventory (WIP). The first control loop influences the internal system variables, i.e. inventory and lead time. The WIP Controller links the input (order release) to the output (performance) and realizes a stable internal order flow.

(ii) The backlog controller manipulates the capacity and controls the backlog. The second control loop influences the external system variables, i.e. backlog and schedule deviation. The backlog controller links the output (capacity) to the demand to realize a “manufacturing on demand”.

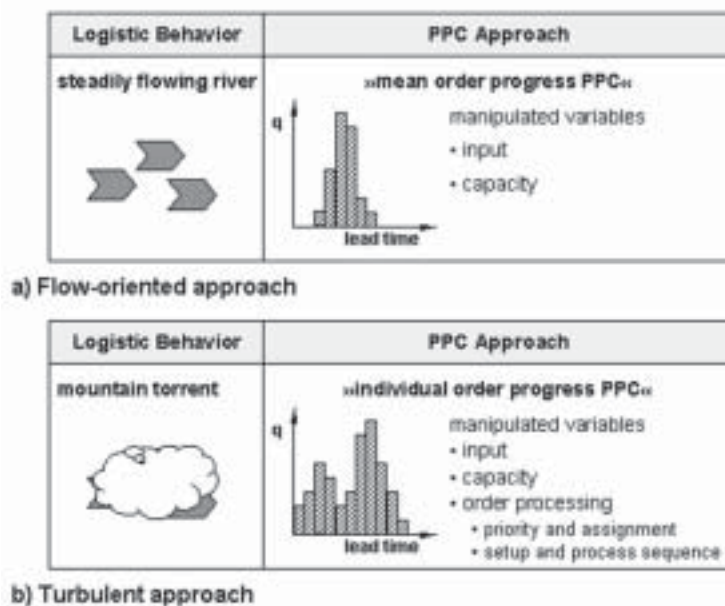


Fig. 9 – Order-oriented perspective: Turbulence Classes in Manufacturing.

Both control loops are linked by the system variable ‘performance’. Analyzing the manipulated and observed variables shows that the resource-oriented view dominates: the aim is to provide sufficient capacity and not a detailed planning of the order sequence for example through order priorities or set-up optimization. Because of this the First-Come-First-Serve Rule should be used for processing the orders.

The *turbulent approach* realizes a heterogeneous internal flow of orders (Figs 8 & 9, bottom). A mountain torrent means a high variation of lead times and requires an “individual order progress PPC”. Thus, it is necessary to take a closer look on the processing of each order. The order-oriented variables order priority and assignment, setup and process sequence should be manipulated. In this approach the order-oriented view dominates: the aim is to plan and afterwards to control (i.e. track and trace) every individual order through each production segment to meet the promised internal or external dead lines. There are two possibilities to realize this approach: On one hand, one can plan every order in detail and give no decisions to those who are responsible for the execution, i.e. the control level. On the other hand it is possible, to focus on the mean values at the planning level. This requires an additional underlying decision level which controls the flow of every individual order.

Basic PPC design

A turbulent environment requires an enlarged positioning concerning the variation of lead times and the height of the WIP level. So additional management decisions concerning PPC design are required.

Chances and risks of inventory

As a result the understanding concerning the inventory level in stock and in process gets a new dimension (Fig. 10):

(i) A *stable environment* enables a simplified understanding of *inventory*. Here it is acceptable to understand the inventory in stock and the work in process as a risk related to *internal goals*: WIP absorbs non-balanced capacities or unstable technical processes.

(ii) A turbulent environment requires an enlarged understanding of inventory. In addition it is necessary to regard the inventory in stock and the work in process as a chance related to *external goals*: WIP enables individual fast deliveries and absorbs demand fluctuations.

	External	Internal
Chances	<ul style="list-style-type: none"> • creates flexibility (enables individual, fast deliveries) 	<ul style="list-style-type: none"> • increases output (especially with changing bottlenecks) • absorbs demand fluctuations regarding quantity and variety
Risks	<ul style="list-style-type: none"> • increases inventory risk (product not saleable because of changing demand) • long mean delivery times and high variation (in case of manufacturing on demand) 	<ul style="list-style-type: none"> • makes planning and control more complex • increases capital tie up • hides organizational and technical malfunctions, non-balanced capacities • late detection of technical product defects

Fig. 10 – Chances and Risks of Inventory.

This shows that the chosen inventory level represents an important design decision within the scope of PPC. It also becomes obvious that the basis of current discussions about the assessment of inventories must be expanded. In a turbulent environment it might be sensible to keep higher inventories in process and in stock in order to reduce the necessary response time to market-related fluctuations of quantity and variety. The factors influencing the chances and risks should be discussed for every specific case.

Design questions

The questions concerning PPC design are arranged in two sections:

1. External logistic positioning: Which logistical performance offers the company to its customer?

- What external requirements are relevant?
- Is the variation of delivery times high or low?
- Does every customer get the same delivery time for the same product, i.e. is there a standard delivery time?

The next step determines how the external logistic positioning can be transferred to the internal aspects:

2. Internal logistical positioning: What is the suitable internal logistical performance of the company?

- What additional internal requirements are relevant?
- Is the variation of lead times high or low?
- Is the WIP level high or low?

These are only basic questions which should be discussed with the management to find the right answers for an individual positioning of the company concerning their logistic performance.

The actual research work focuses on the next step of this design procedure: finding out the right combination of PPC methods. Its goal is to adjust or replace planning methods in day-to-day business. The precondition is the development of a simulation based method test block.

Summary

Strong market fluctuation require to specifically design of the logistic behavior of manufacturing systems which implies that the conventional PPC vision of a steady order flow independent from the external and internal requirements is not realistic. The presented analogy between phenomena of physics and processes in manufacturing systems enables a description of the logistic system behavior. It allows to understand why the classic one-dimensional positioning concerning mean values should be enlarged and take up the variation of target values as visualized in the turbulence portfolio.

This analogy leads to an enlarged understanding of inventory not only as a risk but also as a chance. Thus, it presents a theoretical fundament for the necessary external and internal logistic positioning of a manufacturing company and the next

research steps: a situation-based PPC configuration with the right combination of PPC methods and their suitable parameterization.

Acknowledgements

The research was carried out with the support of the Sonderforschungsbereich (special field of research), *Transformable Business Structures for Multiple-Variant Series Production*, SFB 467, at the University of Stuttgart. It is the objective of special research fields to study interdisciplinary subjects of basic research. These studies are sponsored by the Deutsche Forschungsgemeinschaft DFG (German Research Society).

References

- Lödding H., Nyhuis P. & Wiendahl H.P. (2000) *Durchlaufzeitcontrolling mit dem logistischen Ressourcenportfolio*. In: ZWF 95 (1-2) 46-51.
- Massey, B. S. (1989) *Mechanics of Fluids*. 6. ed. Van Nostrand Reinhold (International): London.
- Mintzberg H. (1994) *That's not „Turbulence“, Chicken Little, It's Really Opportunity*. In: Planning Review, 11/12, 7-9.
- Valckenaers P. & Brussel van H. *On the fundamental difference between manufacturing planning and manufacturing execution*. In: ASI '99; Life Cycle Approaches to Production Systems Management, Control and Supervision: Leuven, Belgium 22.-24.9.99.
- Westkämper E., Pritschow G., Wiendahl H.H., Rempp B. & Schanz M. (2000) *Turbulenz in der PPS – eine Analogie*. In: wt Werkstattstechnik 90 (5), 203-207.
- Wiendahl H.H., Westkämper E., Rempp B. & Pritschow G. *PPC in a turbulent environment-fundamentals and approaches*. In: Sohlenius G. (Ed.): Proceedings of the 33rd CIRP International Seminar of Manufacturing Systems; 5-7 June 2000 Stockholm, Sweden, 320-325.
- Wiendahl H.H., Westkämper E. *Situation – Based Selection of PPC Methods: Fundamentals and Approaches*. In: Chryssolouris G. (Ed.): Proceedings of the 34th CIRP International Seminar of Manufacturing Systems; 16-18 May 2001 Athens, Greece; 241-246.

GIANCARLO MACCARINI
GIOVANNI VALENTINI
LUCIO ZAVANELLA

Improving the simulation of manufacturing systems: The implementation of hybrid simulation and fuzzy logic

Università degli Studi di Brescia
Facoltà di Ingegneria
Brescia, Italy

Abstract — The present study aims to present the methods adopted and the problems met while integrating a simulation model with different tools. The study originated as an improvement of an industrial case study, where a simulation model was developed as a support to production planning and system analysis. The basic idea was to increase the performance and the range of applicability of the simulation model. To this end, it was thought (i) to integrate it with suitable analytical models to speed up calculation times without losing precision and (ii) to introduce fuzzy variables so as to conveniently describe those system parameters affected by a lack of precision in their definition. The paper starts presenting the difficulties met while implementing analytical models, their discussion and the solutions adopted. Successively, a description is given with reference to the fuzzy variables introduced and the results obtained.

Keywords — manufacturing systems, hybrid simulation, fuzzy logic.

The simulation model

The system considered is a Versatile Manufacturing Line, a sort of a “hybrid” from two production typologies, the transfer line and the FMS (see Fig. 1). According to the aims of the present study, this manufacturing system has been chosen without loss of generality, not only for his capacity to realise high production volume while maintaining characteristics of flexibility, but because of its expanding importance in real industrial experience as well. Thus, the considerations reported in the following sections can be extended to every type of manufacturing system.

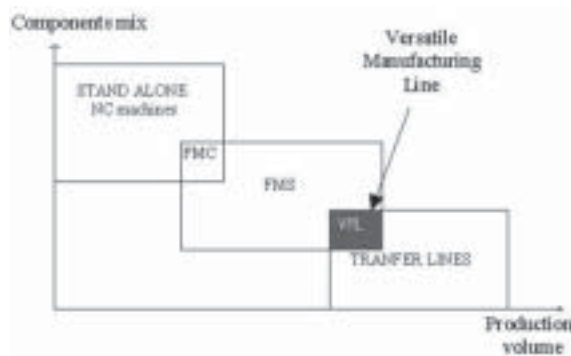


Fig. 1 – Classification of production systems.

In its complete configuration, the studied plant consists of a maximum of 33 working modules disposed in a rectangular layout; all the modules are fed by a central loop transport system (Fig. 2). Only are two different types of CNC working modules present in the system: roughing modules and finishing modules. There are also measuring modules, which perform all the dimensional controls necessary to check the work pieces. One of the peculiarities of this system is the capability of the pallet to store and update, in a proper electronic circuit, the information required to correctly machine the part. In this way, the system has a distributed knowledge and it does not need a supervision computer to control the line.

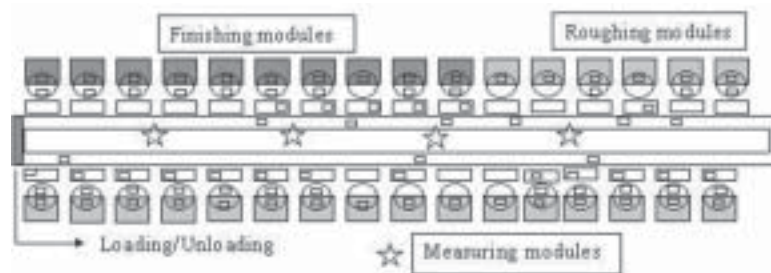


Fig. 2 – Plant layout.

The pallet runs on the central loop transport system and, when it is in front of a module, an exchange of information with the module itself occurs. If, in the tool storage of work machine, there are the tools able to perform the successive required operations and if the buffer before the work machine is free, then the pallet fills the module; otherwise it continues to the next module. When the work piece ends the cycle, its data are updated and it leaves the module to return on the central loop transport system, where it looks for a new machine able to perform the necessary operations. At the end of the work cycle, the pallet goes to the unload station. Since the system is characterised by a very large number of variables, a simulator is useful in solving the following problems:

- how many modules are necessary to achieve the desired productivity?
- how are the single operations to be shared between the modules?

The simulator proposed, developed in ARENA©, is able to:

- design the system, balancing the use of the resources, i.e. the available roughing and finishing modules;
- configure the line so that the various operations are correctly shared between all the modules, with the aim of optimising the line saturation.

Moreover, when the qualitative and quantitative characteristics of the technological cycle of the items change, the specific parameters of the tools can be considered too (tool life, tool reliability, etc.). An appropriate user friendly interface, written in VISUAL BASIC© for EXCEL©, allows the easy input of all the data necessary to describe the system.

This configurator of the simulation program has been used by *Streparava Spa* to study the rocker arms line in different working conditions, with the aim of identifying the parameters affecting the process performance. The simulation model is also described in detail in [Ceretti et al. 1999].

Hybrid simulation

So as to increase the computing performance of the simulation model, it was thought to integrate it with an analytical model describing a part of the manufacturing system. This technique is also known as Hybrid Simulation and it has been studied since the seventies [e.g. J.C. Shanthikumar & R.S. Sargent 1981 & 1983]. According to our purposes, the basic advantage offered by the integration of simulative and analytical techniques lies in the possibility of (i) simplifying the development of the simulation model, saving time in the realisation of the model itself (for skilled staffs) and (ii) speed up the calculation time required by complex programs which must be executed several times to compare different solutions (this topic is also relevant when integrating simulation with other techniques as Genetic Algorithms [Braglia & Zavanella 1999]).

Some researchers suggested a wider utilisation of hybrid modelling [e.g. Ignall & Kolesar 1978], considering it as a tool enabling the efficient implementation of optimisation procedures. Furthermore, this technique allows also to treat the problem from a global point of view (analytical analysis by Linear Programming or Queuing Theory), before facing it at a more detailed level (simulation model): data obtained are passed back to the analytical scheme, thus allowing a recursively optimisation of the problem [Nolan & Sovereign 1972]. In such a way, the analytical model acts as a guide to the experimental campaign carried out by the simulative description of the process.

In the industrial case presented, the complete description of the plant had already been developed via simulation and, consequently, there exists the opportunity to start an experimental campaign leading to (i) the evaluation of the computing benefits associated with hybrid simulation and (ii) the loss of accuracy introduced by the adoption of analytical models in place of a detailed simulation.

The system component selected for analytical modelling was the transport system and the model adopted is derived from Queuing Theory. The basic difficulties met in the implementation of the analytical model were the following.

Selection of the model

This step is not generally critical, but it can hide some traps, e.g. the selection of a model not able to offer a satisfactory description of the real system. The difficulty is also linked to the limited variety of models available in Queuing Theory [Gross & Harris 1985] and to the theoretical hypotheses they introduce (e.g. random selection of items from queues versus the common priority selection). Thus, the selection of the right model is a step which requires the careful analysis of the model to be introduced and the hypothesis linked to the model itself.

Analysis of the transient period

While developing the hybrid simulation, the most interesting difficulty encountered was linked to the different transient periods of the simulation and the analytical model. The question is simply related to the fact that the simulation model may be associated to a duration of the transient period other than the analytical model one. To highlight this issue, it can be thought to the initial period (empty system), where simulation starts adopting some results (derived from the analytical model) for its calculations. Though, analytical data are frequently offered only for the stationary conditions. In other terms, the two transient periods are to be evaluated to identify the instant at which both of them enter the stationary condition and, consequently, data may be collected without loss of accuracy. It is also evident that, while neglecting this fact, “false” information is assumed and wrong conclusions may be drawn.

This fact introduces the problem of evaluating the length of the transient period of both simulative and analytical models. Generally speaking, the transient period analysis is not solved for the largest part of the analytical models, while useful tools are available for simulation techniques. A simple method was proposed by Gross and Harris (1985), and it allows the identification of the beginning of the steady period, even if it does not provide a precise condition related to this event. The basic assumption is that it is possible to arrange a confidence interval based on the average of n simulation runs each of length T . If n is conveniently large (e.g. 20 runs), then:

$$\hat{L} \pm z_{\alpha/2} s_{\hat{L}}$$

where \hat{L} is the average of L (e.g. average number of customers in system) over different replications

$$\hat{L} = \sum_{i=1}^n \frac{L_i}{n};$$

$z_{\alpha/2}$ is the normal distribution standard tables and $(1-\alpha)$ is the confidence level. The value of \hat{L} variance is:

$$s_{\hat{L}}^2 = \frac{1}{n} \frac{\sum_{i=1}^n (L_i - \hat{L})^2}{n-1}$$

The main limit of the procedure described lies in the evaluation of cumulated performances, i.e. depending on the time T . Gordon (1978) presents a criterion based on the variance of the average of the autocorrelated observations. A single but prolonged run is considered: it is subdivided into n intervals each considered as a single replicate. In such a way observations are not statistically independent, as required in Gross and Harris model. A further and rather complex model was proposed by Schruben [in Heidelberg & Welch 1983]. Though, the simplest procedure is the graphical one proposed in Welch (1981): n replications are independently carried out ($n \geq 5$), each one of length m . Then

$$\bar{Y}_i = \sum_{j=1}^n y_{ji}/n$$

is evaluated for $i = 1, 2, \dots, n$ and the new process $\bar{Y}_1, \bar{Y}_2, \dots$ has the same average of the original one, but $(1/n)$ th of its variance, i.e.: $E(\bar{Y}_i) = E(Y_i)$, $\text{var}(\bar{Y}_i) = \text{var}(Y_i)/n$. The new process high-frequency oscillations (low-frequency ones describe the trend of the system) are dumped thanks to the moving average $\bar{Y}_i(w)$, where w is window $w \approx [m/2]$. Values $\bar{Y}_i(w)$ are plotted for $i = 1, 2, \dots, m - w$ and the end of the transient period is identified where $\bar{Y}_1(w), \bar{Y}_2(w), \dots$ seems to stabilise.

The number of observations l to be carried out is calculated by Welch (n, m) graphic method. Figure below shows the width of the interval of confidence while varying the simulation length (MM1 model).



Fig. 3 – Interval width.

Further experiments showed how MM1 analytical behaviour cannot be easily approximated via the transient period analysis described.

The example previously shown described how the basic problem in hybrid simulation may lie in the different time-dependence of its simulative and analytical components. This is especially true when the second class is considered, i.e. where the two components should run in parallel because of the continuous exchange of information. At present, the following approach was adopted to improve the performance of the hybrid model compared to the simulative one: the frequency of “customer” arrivals is calculated by the simulation according to its clock, thus partially “tuning” the effect of the transient period. Actual results confirm a significant improvement, but they still determine a significant difference between hybrid results and complete-simulation ones. First class models adopt the alternative utilisation of simulation and analytical models, thus introducing the possibility of time independence, but in this case calculation times may be excessively prolonged. Finally, third class models appear to be utilisable, as the simulative model is subjected to a more general model.

Fuzzy variables

The study has also involved the introduction of fuzzy variables enabling the description of some parameters of the simulation model. Fuzzy Logic (FL) has been recently used to solve problems in production environment. The advantage of the choice of this logic is generally justified by his flexibility and handiness. FL can be used to define some controlled variables as due date, service times, demand, lead time, etc., that often are vague and imprecise.

Literature quotes three fonts of vagueness:

relationship imprecision, when the process variables are linked each other by using imprecise relations;

data imprecision, when only an imprecise knowledge of parameters is available;

linguistic imprecision, as in the definition of the entity of product demand.

FL is able to deal with expressions of the human language which are difficult to traduce in a scientific algorithm. The common approach to solve these type of problems is the analytic-statistic approach, but this way is often inapplicable because it needs too much information which is not always available. The VLM studied can be described as follows:

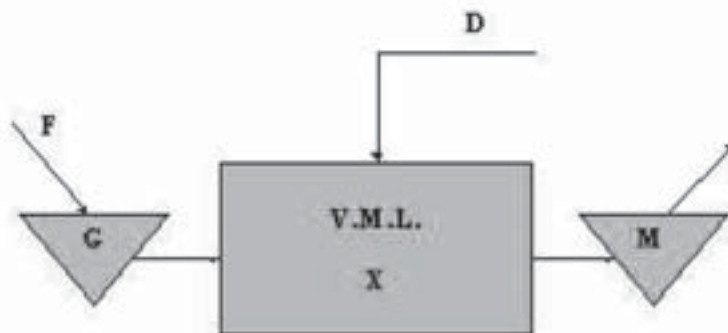


Fig. 4 – The production system scheme.

In Fig. 4, for the period of analysis, it has been indicated:

X = the production of items for a chosen configuration

M = the inventory of finished rocker arms

G = the inventory of raw materials

F = the supply expected

D = the item demand

The main aim of our work is to keep under control the production of this system, using the FL to express the amounts of the demand and the level of the two inventories of finished products M and raw materials G. The fuzzy inputs are the two following:

the level of the inventory M, reduced by the demand in the period of analysis

the level of the inventory G, summed to the supply expected for the same period.

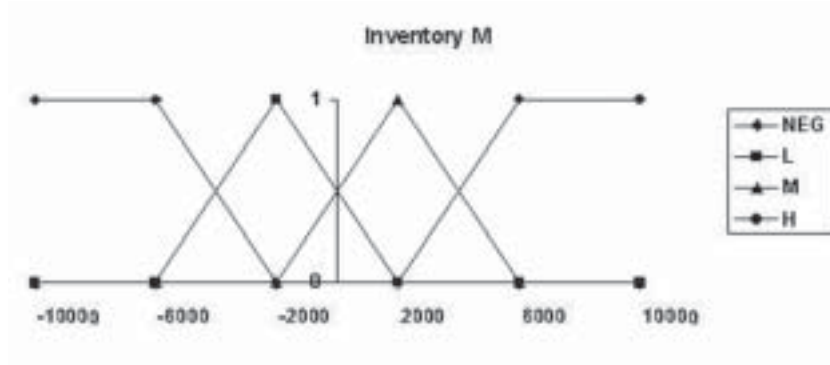


Fig. 5 – Levels of inventory M.



Fig. 6 – Levels of inventory G.

The fuzzy sets proposed have been identified on the basis of the capacity of the inventories, the trend of the stocks, and the maximum range of the demand.

So as to edit the rules of the fuzzy controller system, it is necessary to set a target level for each inventory. These levels can be described using fuzzy sets which indicate the degree of satisfaction of the firm. The target levels are a trade off between the service level and the cost of inventory.

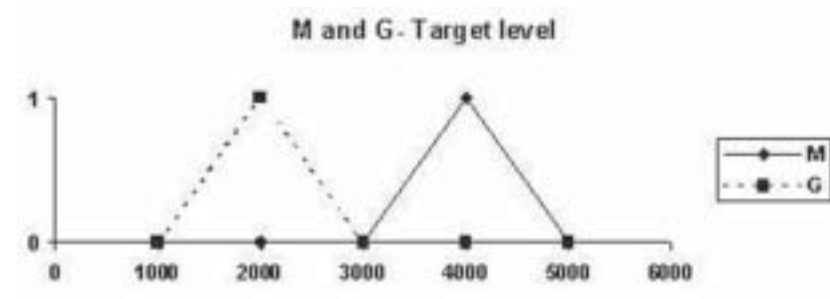


Fig. 7 – Inventory target levels.

The output of our model, also described by fuzzy sets, shows the right configuration of the lay-out, i.e. the number of the machines necessary to meet the demand of the period.

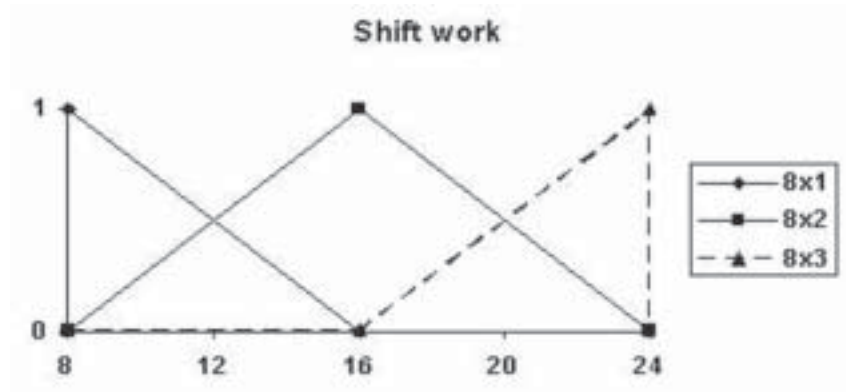


Fig. 8 – The model output.

The rules linking the two inputs with the output configuration are calculated by the software simulating the line dynamics. Thanks to the simulation, the production capacity of every possible (balanced) configuration is obtained over the analysed period.

Applying the IF-THEN decision rules, the following are obtained.

Control Rules:

IF $M_{t-1} - D_t = \text{NEG}$	AND $G_{t-1} + F_{t+LT} = \text{VL}$	THEN	$C = \text{VL}$
IF $M_{t-1} - D_t = \text{NEG}$	AND $G_{t-1} + F_{t+LT} = \text{L}$	THEN	$C = \text{M}$
IF $M_{t-1} - D_t = \text{NEG}$	AND $G_{t-1} + F_{t+LT} = \text{M}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{NEG}$	AND $G_{t-1} + F_{t+LT} = \text{H}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{NEG}$	AND $G_{t-1} + F_{t+LT} = \text{VH}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{L}$	AND $G_{t-1} + F_{t+LT} = \text{VL}$	THEN	$C = \text{VL}$
IF $M_{t-1} - D_t = \text{L}$	AND $G_{t-1} + F_{t+LT} = \text{L}$	THEN	$C = \text{M}$
IF $M_{t-1} - D_t = \text{L}$	AND $G_{t-1} + F_{t+LT} = \text{M}$	THEN	$C = \text{H}$
IF $M_{t-1} - D_t = \text{L}$	AND $G_{t-1} + F_{t+LT} = \text{H}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{L}$	AND $G_{t-1} + F_{t+LT} = \text{VH}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{M}$	AND $G_{t-1} + F_{t+LT} = \text{VL}$	THEN	$C = \text{VL}$
IF $M_{t-1} - D_t = \text{M}$	AND $G_{t-1} + F_{t+LT} = \text{L}$	THEN	$C = \text{L}$
IF $M_{t-1} - D_t = \text{M}$	AND $G_{t-1} + F_{t+LT} = \text{M}$	THEN	$C = \text{M}$
IF $M_{t-1} - D_t = \text{M}$	AND $G_{t-1} + F_{t+LT} = \text{H}$	THEN	$C = \text{H}$
IF $M_{t-1} - D_t = \text{M}$	AND $G_{t-1} + F_{t+LT} = \text{VH}$	THEN	$C = \text{VH}$
IF $M_{t-1} - D_t = \text{H}$	AND $G_{t-1} + F_{t+LT} = \text{VL}$	THEN	$C = \text{VL}$
IF $M_{t-1} - D_t = \text{H}$	AND $G_{t-1} + F_{t+LT} = \text{L}$	THEN	$C = \text{VL}$
IF $M_{t-1} - D_t = \text{H}$	AND $G_{t-1} + F_{t+LT} = \text{M}$	THEN	$C = \text{VL}$

IF $M_{t-1} - D_t = H$ AND $G_{t-1} + F_{t+LT} = H$ THEN $C = L$
 IF $M_{t-1} - D_t = H$ AND $G_{t-1} + F_{t+LT} = VH$ THEN $C = L$

A further fuzzy output was elaborated: the number of the shifts per day, between the three possibilities (8x1; 8x2; 8x3).

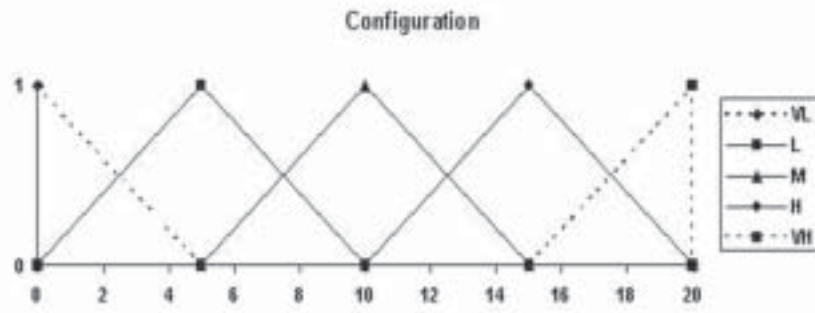


Fig. 9 – Fuzzy variables for shifts.

For this output, the adopted configuration of the line was the ‘VH’ configuration, because of its high productive rate.

The second output rules were the following:

Control Rules:

IF $M_{t-1} - D_t = NEG$	AND $G_{t-1} + F_{t+LT} = VL$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = NEG$	AND $G_{t-1} + F_{t+LT} = L$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = NEG$	AND $G_{t-1} + F_{t+LT} = M$	THEN	$T = 8x2$
IF $M_{t-1} - D_t = NEG$	AND $G_{t-1} + F_{t+LT} = H$	THEN	$T = 8x3$
IF $M_{t-1} - D_t = NEG$	AND $G_{t-1} + F_{t+LT} = VH$	THEN	$T = 8x3$
IF $M_{t-1} - D_t = L$	AND $G_{t-1} + F_{t+LT} = VL$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = L$	AND $G_{t-1} + F_{t+LT} = L$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = L$	AND $G_{t-1} + F_{t+LT} = M$	THEN	$T = 8x2$
IF $M_{t-1} - D_t = L$	AND $G_{t-1} + F_{t+LT} = H$	THEN	$T = 8x2$
IF $M_{t-1} - D_t = L$	AND $G_{t-1} + F_{t+LT} = VH$	THEN	$T = 8x3$
IF $M_{t-1} - D_t = M$	AND $G_{t-1} + F_{t+LT} = VL$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = M$	AND $G_{t-1} + F_{t+LT} = L$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = M$	AND $G_{t-1} + F_{t+LT} = M$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = M$	AND $G_{t-1} + F_{t+LT} = H$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = M$	AND $G_{t-1} + F_{t+LT} = VH$	THEN	$T = 8x2$
IF $M_{t-1} - D_t = H$	AND $G_{t-1} + F_{t+LT} = VL$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = H$	AND $G_{t-1} + F_{t+LT} = L$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = H$	AND $G_{t-1} + F_{t+LT} = M$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = H$	AND $G_{t-1} + F_{t+LT} = H$	THEN	$T = 8x1$
IF $M_{t-1} - D_t = H$	AND $G_{t-1} + F_{t+LT} = VH$	THEN	$T = 8x1$

To develop the model proposed, the application system MATLAB 5.3 was preferred and, in particular, its tool FUZZY LOGIC TOOLBOX.

FL methodology differs from the traditional mathematics to choose a continuum of values (from zero to one) for each fuzzy set, that represent the degree of truth. FL allows the study of a production environment beginning from simple concepts or from vague or non-well-known relations. Starting from an empirical analysis of the variables (inputs and outputs of the model), with the help of the simulator, it is possible to edit the control rules. With the FIS editor of the toolbox of Matlab 5.3, it was obtained a complex description which sets in optimal way the available resources.

Conclusions

The study described is still under development. In the present report, the Authors simply wish to propose some elements for reflecting on the great possibilities of simulation when integrated with other tools also in the field of industrial applications. To this end, the difficulties met while developing a hybrid model were described, with particular reference to the topic of the transient period. Promising (but still limited) results were obtained while utilising fuzzy variables.

Acknowledgements

The present study was supported by MURST funds.

References

- Braglia M. & Zavanella L. (1999) *Experiences and issues in evaluating tool requirements using genetic algorithms*. Int. Jour. of Prod. Planning and Control, 10(4).
- Gross D. & Harris C.M. (1985) *Fundamentals of queueing theory (second edition)*. John Wiley and Sons: New York.
- Pegden C.D. & Rosenshine M. (1982) *Some new results for the M/M/1 queue*. Management Science, 28(7).
- Ceretti E., Maccarini G.C. & Giardini C. (1999) *Study of a decision support tool for the automatic configuration of a modular plant*. Proc. of AMST'99 Conference, Udine (I).
- Heidelberg P. & Welch P.D. (1983) *Simulation run length control in the presence of an initial transient*. Operations Research.
- Pegden C.D. (1984) *Introduction to Siman*. Systems Modelling Corporation.
- Nolan R.L. & Sovereign M.G. (1972) *A recursive optimization and simulation approach to analysis with an application to transportation systems*. Management Science, 18 (12).
- Shanthikumar J.C. & Sargent R.S. (1981) *A hybrid simulation/analytic model of a computerized manufacturing system*. Operational Research.

- Law A. & Kelton D. (1982) *Simulation modelling and analysis*. Mc Graw-Hill: New York.
- Ignall E.J., Kolesar P. & Walker W.E. (1978) *Using simulation to develop and validate analytic models: some case studies*. Operations Research, 26(2).
- Shanthikumar J.C. & Sargent R.S. (1983) *A unifying view of hybrid simulation/analytic models and modelling*. Operations Research.

MARIO TUCCI
MARIO RAPACCINI
EMANUELE CHELI
GIANNI BETTINI

*Towards integrated simulators. First step:
please pass the data*

Dept. of Energy Engineering "Sergio Stecco"
Sezione Impianti e Tecnologie Industriali
Università degli Studi di Firenze, Italy

Abstract — In this paper we present some ideas arising from our past research in the field of automatic simulation model building. Having demonstrated feasibility of such task we had to face the need of collecting huge amount of data deriving from simulation models. We propose here an embryonic architecture to integrate simulators in the most diffused ERP software, where we can find the requested company's data-bank.

Keywords — production planning and control, manufacturing, modelling, simulation, ERP.

Introduction

In order to increase industrial applications of simulation modelling and analysis, especially in the field of manufacturing system, in the recent past interactive-simulators (visual modelling and objects-libraries based) have been developed. This lead to remarkably simplify user-interaction, both in model development and experimental analysis.

In Any case, this evolution does not seem sufficient to achieve the mentioned goal: a lot of qualified and hard job for data collection, model development, verification, validation and analysis is still required.

Since 1997, a feasible solution has been proposed by our research group, based on the past experience in productive systems simulation: using a commercial simulator (System Modeling Arena rel. 2.0), quite good validation of automatic modelling was achieved. This automatization was obtained through appropriate pre-processing of a structured data set.

For this purpose, the essential point was the separation of the physical objects (resources), of the technological objects (products) and of the information flows (process control) of the system itself in three different and separated subsets for system description.

Each of this subsets was then interpreted and a ready-to-run model developed through a library of basic building block (previously precompiled and made available). This was possible only with simple case-studies, because object-orientation was not yet introduced in the used simulator, and the pre-processing could not derive and instantiate new objects to represent some specializations in system logics and data.

The second step was done in 1998, when an Object Oriented Simulator (AESOP Simple++) was tested to substitute ARENA, and the powerful features of this new

modelling paradigm was verified. Then we had an easily-maintainable, reusable, and efficient automatic modelling simulator. To go further, complexity of real-systems' PP&C logics was analysed, and appropriate data structures, aimed to a detailed description of that complexity, were created, coming finally to perform some automatic modelling of sufficient realistic production systems. Then a doubt overwhelmed us.

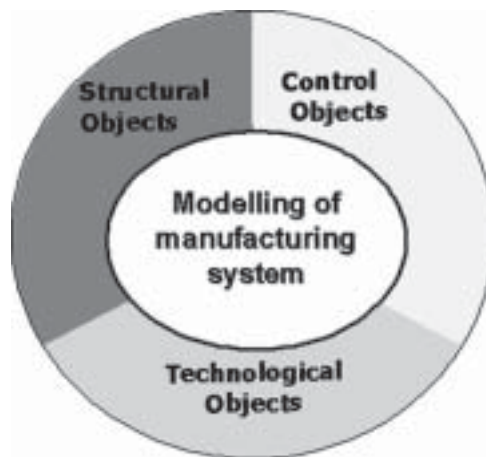


Fig. 1 – logical model.

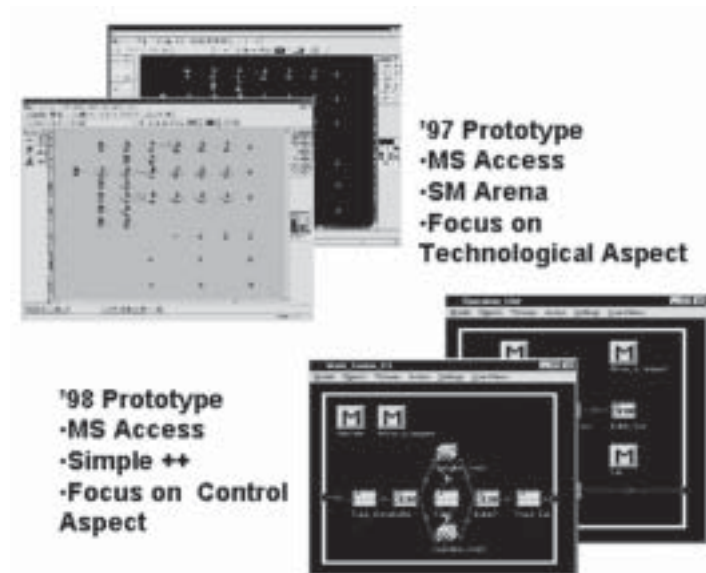


Fig. 2 – Developed prototypes.

The real system is introduced essentially like a large amount of data, under a lot of shapes (designs, tables, documents, etc.) deriving from observation, control, recording. Even if we collected all these data, and it's really a great effort, we also have to maintain them updated in order to preserve the possibility to go on using the simulation models. In the last years many companies found new solutions in IT (Information Technologies), so we look at ERP (Enterprise Resource Planner) as the natural starting point of our research because they represent the up-to-date database of the company.

ERP

The use of informative systems assumed increasing importance as an arm of competitiveness and development for medium-large companies; these systems deal in management of administrative-financial, programming, manufacturing control and part of that technical aspect, they try to integrate all the management processes involving all business areas.

A complete vision of the productive system is undoubtedly the most effective, even if in the past particular models, focused in the analysis of the single business aspects of the productive system, were developed.

The main advantages deriving from the use of an integrated informative system like ERPs (Enterprise Resource Planning) are:

- uniqueness of the data;
- controlled and transparent standard processing;
- fast and focused information distribution;
- control of inter-company processes.

The base for an actual integration is the enterprise information system and mainly its function of database.

ERPs have moved the managerial culture from financial tracing to resource optimising, through activity analysis and planning. This new culture, initially born in great companies, is now extending to small and medium companies. ERPs represent a fundamental step in information system evolution, although the emergent model of Extended Enterprise, more concentrated in external relationship than in the optimisation of the internal procedures, is already watching beyond. In the growing consciousness of this perspective, more that in the Euro introduction and Y2K compliance, we can find the reason of the interest for the new integrated application in full ERP (offered by market leaders) or light ERP, recently offered by various small and medium software-houses.

A modern managerial infrastructure able to quickly accept the third millennium-change of business is a basic factor for all corporate structures.

We decided to study SAP ERP because of the large share of the market gained by this and we present here some of the features of SAP R3 which are very relevant in this perspective of simulation integration with these tools.

SAP stands for Systems, Applications, SAP group, founded in 1972 in Walldorf (Heidelberg, Germany) is today one of the world biggest software producers and

leader in managerial applications. R2 and R3 are the identification codes of SAP main systems: R/2 is developed as a mainframe system and R/3 is developed for a multilevel client/server environment. While many software houses oriented their attention towards some business areas (developing systems in order to support only those specific areas), SAP addressed the overall business, offering a unique system to replace the high number of independent single modules. An unique system composed by depending modules executing specific functions but in order to work with other modules.

The basic architecture of R/3 SAP System is characterized by the following features:

Scalability

Applications can run on computers of various sizes. Using a multi-level client/server architecture makes it possible to separate the application logic, the presentation and the database. In a three level R/3 configuration separate computers are used for the presentation, the application and the database. The database computer can then serve several application servers, and a presentation computer can access different application servers. Other configurations are possible as the following figure shows.

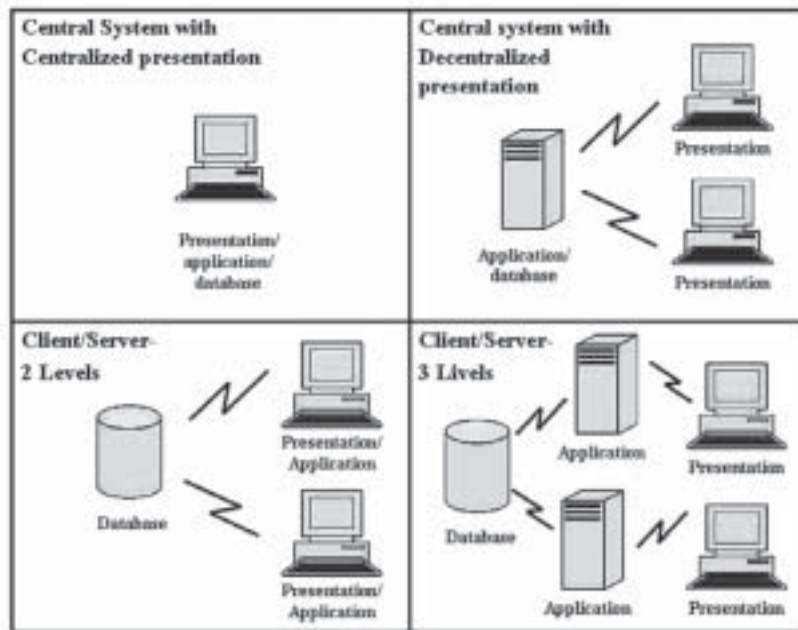


Fig. 3 – Scalability.

Portability

The R/3 System can run on different operating systems (UNIX, MPE/iX, OpenVMS, OS/400, Windows NT); furthermore is compatible with database

systems of various producers (IBM DB2, Informix, Oracle, Software AG, Sybase) and the GUI (Graphical User Interface) can visualize all the output of the principal functions in the majority of front ends presentation systems (Macintosh, OS/2PM, OSF/Motif, Windows).

Openness

Allows applications, data and user interfaces to be integrated through the support of international standards for interfaces (TCP/IP as the network communication protocol, RFC as the open programming interface, CPI-C for program-to-program communication, SQL e ODBC, for access to the relational database, OLE/DDE for the exchange of objects with PC applications, X.400/X.500, MAPI e EDI for external communications).

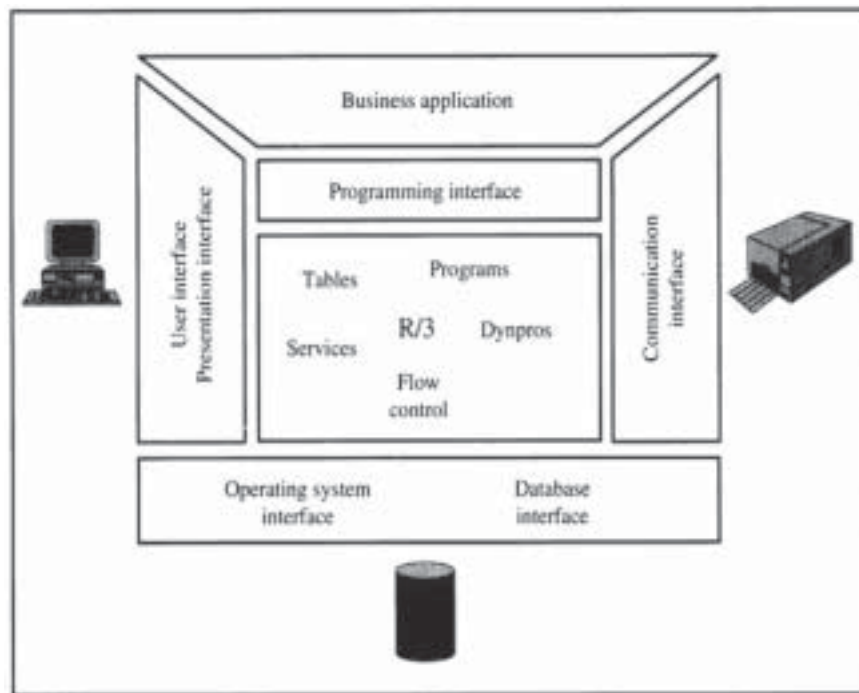


Fig. 4 – Openness.

This is a fundamental property for interfacing the simulation software. R/3 installations include module nucleus, called R/3 BASIS and additional application modules that can be grouped in different areas:

- Accounting
- Human Resources Management
- Logistic.



Fig. 5 – SAP R3 Basis System and Application Modules.

For example Operational Logistic includes the design of the material, information and production flow, from the vendor, through production, down to the consumer. With the implementation of this module we can plan, control and coordinate logistical process across departments boundaries, based on the integration of existing data and functions. The integration of the individual application modules in the R/3 Systems prevents unnecessary and time-consuming multiple entries when processing business events. This integration also ensure that the value-related side of business event is taken into consideration in the quantity-oriented processing steps, and therefore also the accounting requirements

The managerial applications are written in ABAP/4 (Advanced Business Programming Language). The programs of these applications are interpreted from a runtime R/3, installed on every applications server and written in ANSI-C

In addition to standard business modules SAP offers other tools for module development. Some of these are software solutions that are tailored to particular branches of industry ("Industry Solution"), other support different requirements (word processing, archiving system, internal and external communication services, etc.), others provide additional functions for integrating application modules.

The ABAP/4 Development Workbench is the programming environment of the R/3 System and can be uses as a separate product by the customer for developing customer software and carrying out personalisation. The ABAP/4 Development Workbench supports the software development cycle using ABAP/4 language data access, network communication and implementation of user interfaces. With these

tools it is possible to develop software and to customize or expand function modules. The ABAP/4 programming language was developed specifically for business applications and integrate into client/server architecture so it assures that the developed software is compatible with all computers and database systems supported by SAP. The ABAP/4 Development Workbench contains several development tools for prototyping, testing and debugging to support the creation of custom components. These can be combined with standard component in many ways even if it is not recommended to make changes in application logic. In addition to programming possibilities in ABAP/4 the Development Workbench also contains tools for modelling, for defining tables and data structures and for implementing graphical user interfaces.

The ABAP/4 Data Dictionary is a central information source that contains the description of all enterprise's application data, as well as information about relationships between this data and the data's use in programs and screens. The descriptive data of a Data Dictionary are also called "meta data", because they represent data about data. The R/3 Data Dictionary or ABAP/4 Dictionary answers the following core questions for users, developers and end users:

- What data is contained in the enterprise database?
- What characteristics does this data have (name, length, ...)?
- What is the relationship between the data objects?

The relational data model is an important part of R/3 System and its core, it consists of two-dimensional tables, which contain all of the data and relationship. These tables, as well as data fields and database structure, are defined in the R/3 Data Dictionary.

The ABAP/4 Data Dictionary defines, in an effective way, a relational database independently from the database systems of the single vendor and integrates, if necessary, their managerial functions.

Tables are elementary data fields without an internal structure. They are uniquely identified by a primary key and are isolated from each other. Access is possible regardless of the database structure, which ensures that data independence is retained. Access occurs through the specification of a table name, a primary key and a field name and is based on a query language such as SQL. Using foreign keys, it is possible to create links between tables.

A distinction is made between transparent tables, which in the R/3 System, for example, can also be used for other applications, and pooled tables or cluster tables, in which several smaller or internal tables are combined, buffered completely for capacity reasons.

Tables represent the more important data structure and SAP supplies several functions for their management. The ABAP/4 Data Dictionary defines logical tables that can be arranged with Open SQL elements. Moreover internal tables only exist while the program is in phase of execution.

Tables are at first not physically present, but are logically defined in the ABAP/4 Dictionary with the help of meta data. In the dictionary, fields that are used in tables, DYNPROS (Dynamic Programs for Controlling Queries) and applications are described globally.

Another important feature about data change in ABAP/4 Dictionary is to be consider. The architecture of the ABAP/4 Dictionary is completely integrated in the ABAP/4 Development Workbench and the R/3 development environment. When a table is called, it is not presented as a physical table definition (as in conventional database), but is rather regenerated when the ABAP/4 Dictionary is accessed.

Despite having performance-drawbacks, this interpretative incorporation of the ABAP/4 Dictionary into the program flow has the following advantages:

- Tables can be displayed directly in the ABAP/4 editor; application programs can use table structures directly (integrated ABAP/4 Dictionary)
- Change immediately and automatically have an effect on all affected applications (active ABAP/4 Dictionary)

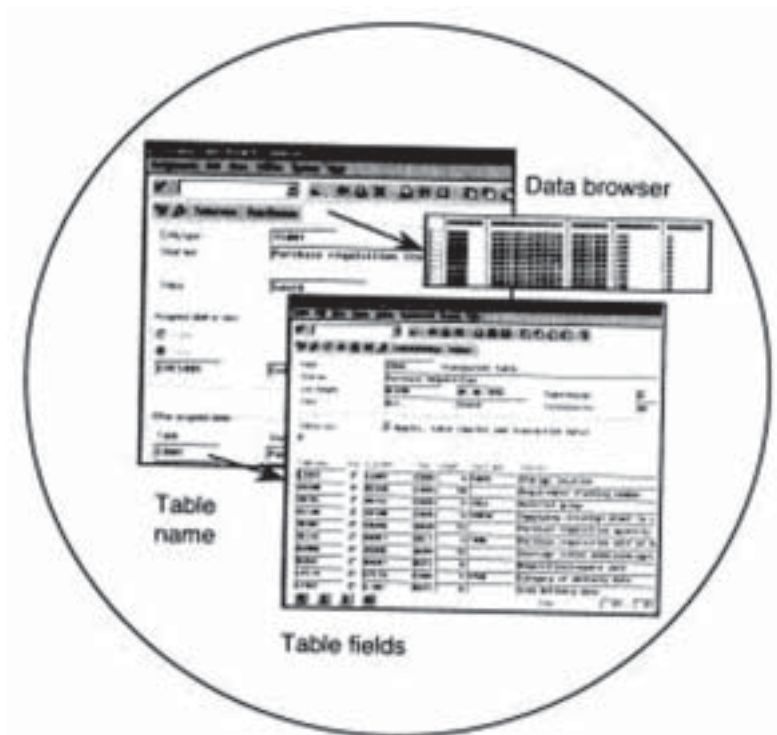


Fig. 6 – ABAP/4 Data Dictionary.

Furthermore, it is possible to define something called views in the ABAP/4 Dictionary.

Views are virtual tables in which user can display application-based data with a specific scope. Only necessary data are displayed; data can be combined; data can be taken from tables in different systems. Fields descriptor can have names that differ

from the original field name. Views support the formatting of data in reports or list screens.

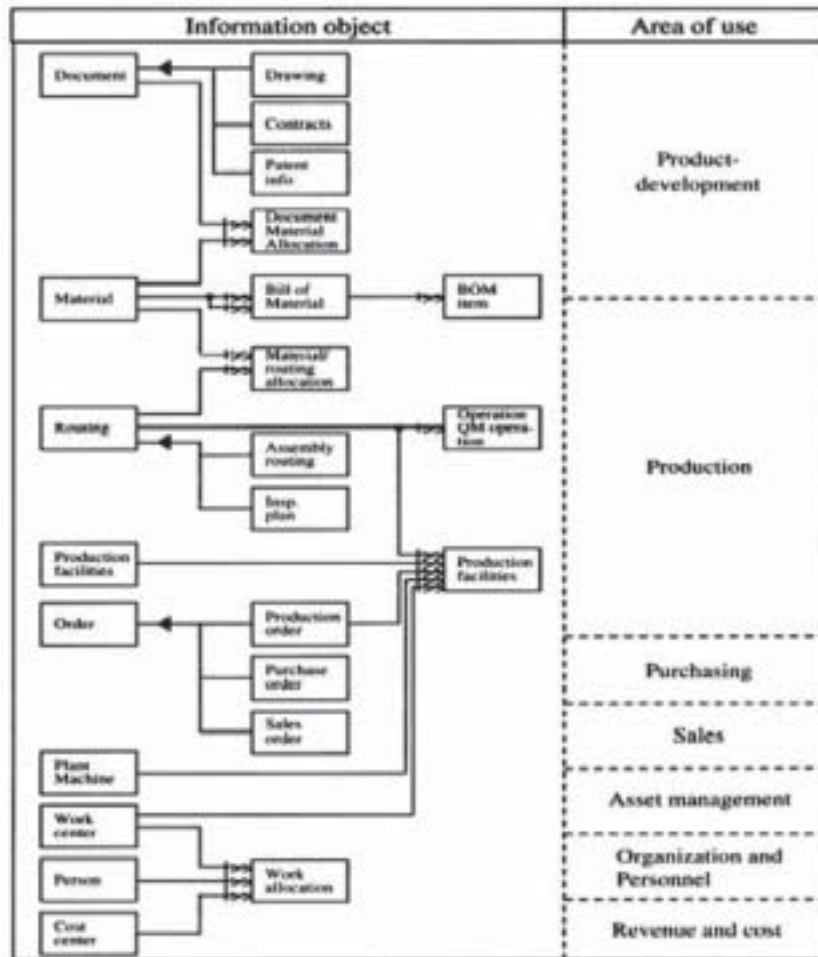


Fig. 7 – data sharing.

The Production Planning and Control module (PP) deals, within this logistics chain, with quantity and time-related planning of products to be manufactured, as well as the control and the production process flow. In addition to the corresponding functionality for master data maintenance, the PP module support all of the quantity and capacity-related steps for planning and control production. This includes both different planning concept, for example MRPII and Kanban, and different types of production, such as production by lot size, make-to-order production (variant production), repetitive manufacturing and process manufacturing.

Interfaces exist between the PP System and the following areas:

- Sales and Distribution (SD),
- Materials Management (MM)
- Controlling (CO)
- Project System (PS)
- Personnel Planning (PD).

All modules are real-time applications, that is, quantities and values are saved directly meaning that all users of the system have access to the same data which is always up-to-date.

The PP components are:

- PP-BD Basic Data
- PP-SOP Sales and Operation Planning
- PP-MP Master Planning
- PP-MRP Material Requirements Planning
- PP-CRP Capacity Planning
- PP-SFC Discrete Orders Processing
- PP-PI Continuous Order Processing (Process Industry)



Fig. 8 – Production Planning Application Module.

The main master data, or basic data, for executing production planning and control includes material, bill of materials, routing, documents, production resources or tools, and work centre. Some of these, for example material, are also used by others modules.

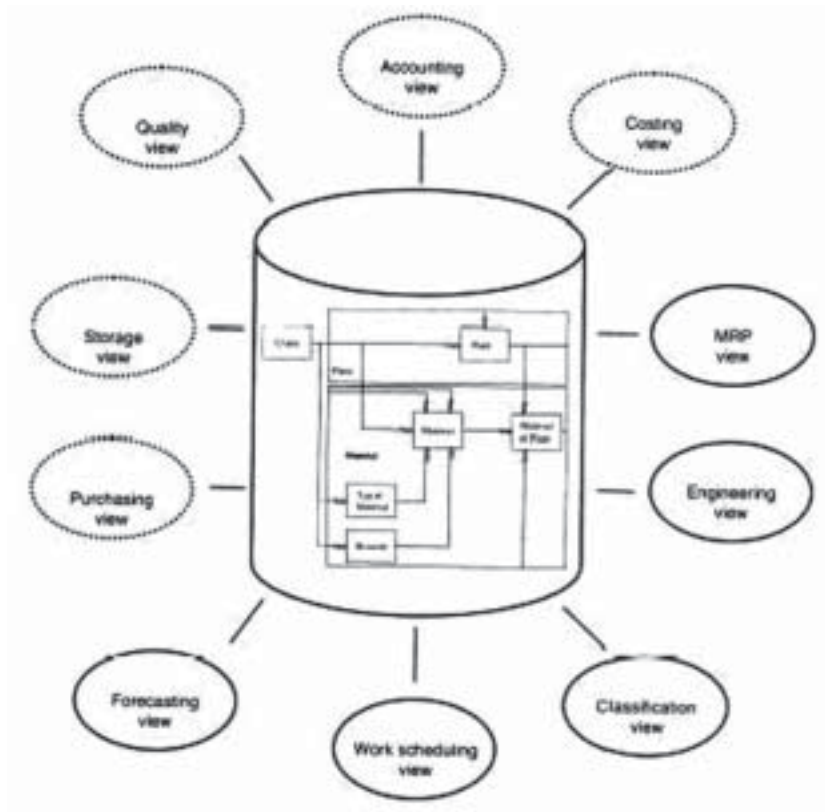


Fig. 9 – Material data and Application Modules.

A material describes the output produced by a manufactory company. This can be finished products, assemblies or components. There can be structural relationships (superordinate and subordinate) between the different materials that, along with their quantity specifications, are stored in the bill of materials. Furthermore in manufacturing company it is necessary to provide the geometrical specifications of a material to production as a model. In R/3 System this information is managed as documents, for example drawing. In addition to the geometrical specifications, a drawing can also contain technological specifications (measurements, cross section views) as well as other notes (references, classification features). A routing is a description of the process flow for the production of a material or for the rendering of a service. Production resources are non-stationary operating facilities that can be

used as a part of production. They are assigned in the plans or orders to those processes for whose execution they are required. In this case, a work centre is an organisational unit in which work can be performed.

A new architecture

The ERP structures are more similar to data set present in traditional managing and accounting tools than to simulation logic. Simulation analysis hasn't yet enough appeal and diffusion to induce IT producers to build simulation-dedicated structures.

Therefore our first step will be to resort an appropriate Data Simulation View.

This means to know in SAP:

- which data are present
- where data are stored
- which data must be added

Obviously in order to realise automatic model building this function is demanded to integrated simulator: we'll call this first function Composition.

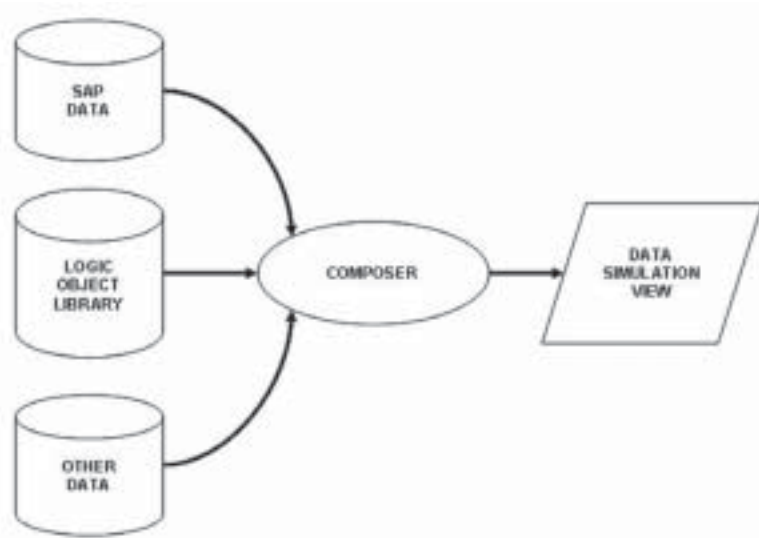


Fig. 10 – Composition.

Composer must extract from SAP useful data and integrate them with data coming from other database and processed data. Requests derive from Logic Object Library, that contains all the needed logic structures.

The second step (Building) consists in organising the Data Simulation View in a Logic Model using again Logic Object Library. This representation follows the O-O paradigm. During this step user must be able to set level of details thus leading to different models.

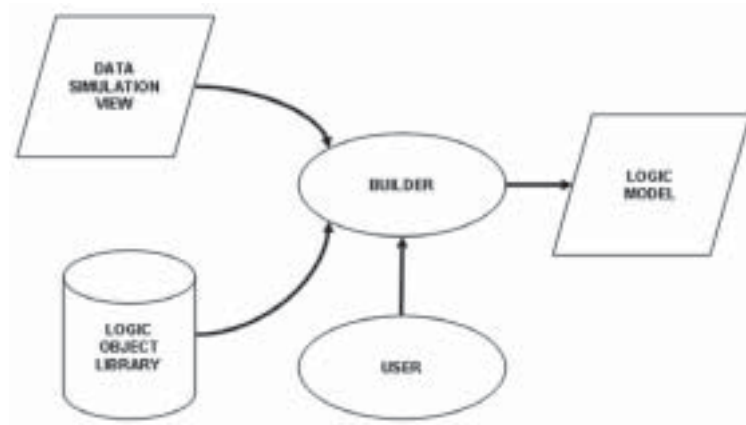


Fig. 11 – Building.

Following step (Translation) achieves the Translation of general O-O Logic Model into a Simulation Model which can be run by a specific simulator.

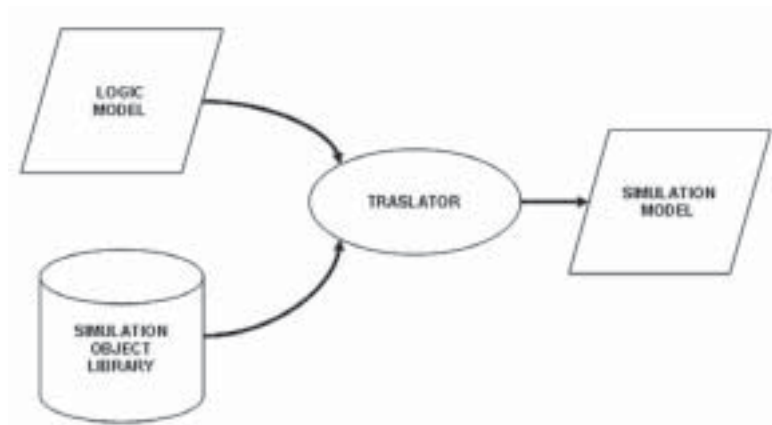


Fig. 12 – Traslatino.

Once Simulation Model has been obtained, user has to design the experimental campaign, according to traditional simulation rules. This means defining time length, data to be collected and so on.

We can think that the first two steps above described, Composition and Building could be developed as internal functions of SAP, using its internal programming language ABAP/4 both to link to external databases and to organize new views.

It can be noticed that the user tasks in the above suggested architecture could be restricted to the integration of data, the definition of all the parameters controlling simulation, and the analysis of collected data.

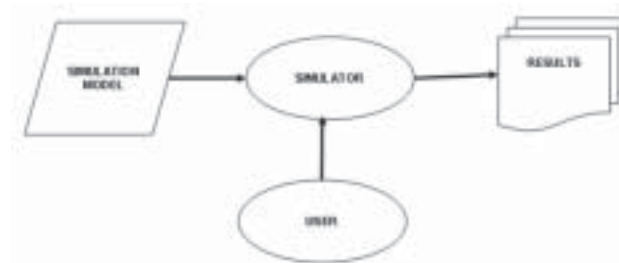


Fig. 13 – Simulation.

A further keypoint in the development of this research, is about the introduction of a differentiated approach within the process of complex-systems analysis. This is mainly due to the following considerations:

- the levels of complexity of real systems are very difficult to afford in a detailed manner
- the managers must try to control this complexity, because this constitutes doubtlessly a successful key within the markets turbulence

To permit an immediate and effective simulation use, two aspects should be investigated:

- system simplification (elimination from the system to be simulated, and therefore from its logical model, of less important elements, workings, controls);
- aggregation-restriction (moving the attention towards objects with more and more combined information, although having in mind the whole system. This attitude could also be not uniform, for instance in some cases would be necessary to consider in detail a particular part of the system).

To extensively apply the simulation as planning and analysing tools, the architecture should permit this diversification of approaches. In our opinion it could be possible but it will be a logical-model structuring and defining problem more than an architectural one.

The object oriented methodology provides an useful help: the decomposition of real systems complexity, based on the objects and relationships identification. It's at the same time a top-down approach (splitting the whole system in smaller elements and classes during the analysis) and bottom-up (organising objects and classes in system model). Thanks to this double property it's applicable either for a vertical architecture (hierarchical/functional) either for an horizontal one (etherarchical). In this sense the method used in the previous works, with their successive specification of the objects, has to be considered valid even if embryonic.

Having different detailed analysis means having different simulation models and so different logical models. The difference must be retrieved in the logical methods and in the descriptive function (builder).

Simplifying means removing from the complete logical model (maximum level of detail) some objects or some of their attributes or methods. It seems to be easy to do for complete objects but not so easy for the single method and attribute.

Maybe that a strong O-O representation could provide a useful model at every sub-classing level. This means that attributes, methods or the exchange messages (defined at the considered level) should be enough to describe the required system simulation complexity. Otherwise a method or an attribute (defined in the sub-classed level) could be relevant for the system simulation. In this second case required attributes and methods should be present (individually or in an aggregate way) as an attribute or a method of the super-class object. Maybe this implies different object classes to reproduce the same object at different levels of details.

The same problem afflicts data structure: different level views must be implemented to interface different simulation levels.

Conclusions and perspectives

Most of data needed to run simulation are already present in standard SAP implementation even if a deep integration is always required in order to achieve a better description of the system, i.e. transportation means, and, generally, all the data affected by stochastic variation (Composition). Nevertheless data must be re-organised according to a scheme which enables a direct conversion to a simulation model (Building).

Increasing interest towards Simulation could push SAP, or other solution providers, to enclose these functions in available standard applications as permitted by ABAP/4Development Workbench.

References

- Blain J. et al. (1999) *Il manuale di SAP R/3*. Jackson Libri, Gruppo editoriale Futura: Milano, Italia.
- Bakalem M., Habci G. & Courtois A. (1995) *PPS: a Contribution for Manufacturing Systems Simulation*. In: Proceedings of the Summer Computer Simulation Conference SCSC'95 Ottawa, Ontario, Canada, July 24-26, 390-395.
- Bakalem M., Habci G. & Courtois A. (1996) *Conceptual Frames for Physical and Control Systems Modeling in Manufacturing Simulation*. In: Proceedings of the 8th European Simulation Symposium, ESS'96, Genoa, Italy, Oct. 24-26, 319-323.
- Bartolotta A & Garetti M. (1995) *Fattibilità ed Obiettivi di un Sistema Integrato per la Progettazione di Sistemi Produttivi*. In: Proceedings of the XXII Convegno Nazionale ANIMP, Oct. 19-21.
- Bartolotta, A. & Garetti M. (1996) *Object-Oriented Representation of Manufacturing Systems: Trends and Perspectives*. In: Proceedings of the International Conference on Advances in Production Management Systems: Kyoto, Japan, Nov. 4-6, 193-198.
- Bartolotta A. & Garetti M. (1997) *The PlantFaber Workbench*. Workshop Il progetto Comunitario PlantFaber: Milan, Italy, Jan. 30.

- Botta G., Guinet A. & Boulle D. (1997) *Object-oriented analysis with structured and integrated specifications and solutions (OASISS) for production system control*. Journal of Intelligent Manufacturing, 8(1), 3-14.
- Cattaneo F., Garetti M., Macchi M., Fidanza F. & Vigliadoro G. (1999) *Principi di progettazione di un simulatore modulare per una industria del settore avionico*. In: Proceedings of the XXVI Convegno Nazionale ANIMP, Oct. 21-22.
- Colsman R., Bas A.O., Escoto R.P., McDonnell L.R., Esteban F.C.L. *Design of simulation model automatically from a given database and its simulation runs*. In Proceedings of 1995 EUROSIM Conference, EUROSIM '95. Elsevier, Amsterdam, Netherlands. 1011-16.
- Corradi E., Bartolotta A. & Garetti M. (1997) *Utilizzo della metodologia object oriented per la descrizione dei sistemi produttivi*. In: Proceedings of the XXIV Convegno Nazionale ANIMP, Oct. 23-25.
- Dangelmaier W., Kuhn A. & Langemann T. *OOPUS – An Approach to Develop Flexible Production Planning and Control Systems*. In: Proceeding of 9th European Simulation Symposium, ESS'97, Passau, Germany, Oct. 19-22, 747-750.
- Dietrich B.L. (1991) *A taxonomy of discrete manufacturing system*. Operation-Research. 39 (6), 886-902.
- Eriksson H.E. & Penker M. (1998) *UML Toolkit*. John Wiley & Sons: New York.
- Garetti M. & Taish M. (1997) *Sistemi di produzione Automatizzati* Edizioni CUSL: Milan, Italy, 1-11, 163-365.
- Greene J.H. (1997) *Production and Inventory Control Handbook* McGraw-Hill: New York, N.Y., USA.
- Graves R.J., Konopka J.M. & Milne J. (1995) *Literature review of material flow control mechanisms*. Production Planning and Control, 6(5), 395-403.
- Halevi (1999) *Integrating process planning and production management*. In: Proceedings of ATPPC Advanced Techniques in Production Planning, Hanover, Germany, 11-12 February 1999, 1-29.
- Keller G. & Teufel T. (1998) *SAP R/3 Process Oriented Implementation*. Addison-Wesley Longman, England.
- Hill D.R.C. *Object-Oriented Analysis and Simulation* Addison-Wesley.
- Law A.M. & Kelton W.D. (1991) *Simulation Modeling and Analysis*. McGraw-Hill: New York, N.Y., USA.
- Martin J. & Odell J.J. (1996) *Object-Oriented Methods*. Prentice-Hall, Upper Saddle River, New Jersey, USA.
- Mize J.H., Bhaskute H.C., Pratt D.B. & Kamath M. (1992) *Modeling of Integrated Manufacturing Systems Using an Object Oriented Approach*. IEEE Transactions, July, 24 (5), 15-26.
- Mondem Y. (1986) *Produzione Just-in-Time*. Petrini Editore: Torino, Italy.
- Pratt D.B., Farrington P.A., Basnet C.B., Bhaskute H.C., Kamath M. & Mize J.H. (1994). *The Separation of Physical, Information and Control Elements for Facilitating Reusability in Simulation Modeling*. Int. Journal in Computer Simulation 4, 327-342.

- Rumbaugh J. et al. (1991) *Object-Oriented Modeling and Desig.* Prentice-Hall, Upper Saddle River: New Jersey, USA.
- Salvendy G. (1992) *Handbook of Industrial Engineering.* John Wiley & Sons: New York, N.Y., USA
- Schotissek P. & Glassner J. (1993) *Exploiting logistic potentials with a simulation-aided test of PPC methods.* System-Analysis and Modeling Simulation, 12 (3-4), 199-216.
- Schroder G. (1889) *Application of simulation for daily production planning and control in shop production.* In: Proceedings of CAD/CAM Robotics and Factories of the Future. 3rd International Conference (CARS and FOF '88), Berlin, Germany, vol. 1, 395-400.
- Tucci M. Rapaccini M. & Bettini G. (1997) *Automatic modeling of manufacturing systems with conventional stochastic discrete events simulation languages.* In: Proceeding of 9th European Simulation Symposium, ESS'97, Passau, Germany, Oct. 19-22, 411-415.
- Tucci M. Rapaccini M. & Bettini G. (1999) *A taxonomy of PPC methodologies: a step towards advanced simulation tools.* Hannover, Germany.
- Kruger M. (1986) *Simulation of a general stochastic network model with office or personal computer.* Neue technik im Buero. 30(2), 48-49.
- Waikar A. M., Sarker B.R. & Lan A.M. (1995) *A comparative study of some priority dispatching rules under different shops loads.* In: Production Planning and Control July-Aug. 6 (4), 301-310.
- Wiendahl H.P. (1995) *Load-Oriented Manufacturing Control.* Springer-Verlag: Berlin, Germany.
- Zeigler B.P. (1984) *Multifaceted modeling and Discrete Event simulation* Academic Press: London.

J.M. VAN DE MORTEL-FRONCZAK
J.E. ROODA

*Agent-based control of a lithoshop
a simulation study*

Eindhoven University of Technology
Dept. of Mechanical Engineering
Eindhoven, the Netherlands

Abstract — Most control systems used in manufacturing have a hierarchical structure. Hierarchical control systems become very complicated and difficult to maintain and modify when the underlying production systems grow in size and complexity. Moreover, they are characterized by a relatively high sensitivity to failures. As opposed to hierarchical systems, heterarchical control systems are flexible, modular, easy to modify, and to some extent fault-tolerant. In this paper, a simulation study is presented of the application of heterarchical control concepts in the production of integrated circuits. Simulation experiments show that the heterarchical control system described enables the lithoshop to perform equally well as the hierarchical control system does. The heterarchical control system can be modified or extended very easily, offers transparency and is very well capable of handling machine and even control component failures.

Keywords — modelling, heterarchical control systems, parallel specifications, simulation.

Introduction

Several control architectures can be used to control manufacturing systems [Dilts et al. 1993]. The centralised control architecture offers the advantage of global optimization, but has low fault tolerance, is difficult to modify or expand, and extension possibilities are limited. The proper hierarchical form is characterised by a pyramidal control structure and is very well capable of controlling complex manufacturing systems. Its most important disadvantages are rigidity and sensitivity to failures. The modified hierarchical form resembles the proper hierarchical form in many ways, but offers a larger degree of autonomy within the structure. Although more autonomy exists in the system, the disadvantages of this form resemble the disadvantages of the proper hierarchical architecture. The heterarchical control architecture is based on the concept of distributed control. Control components (agents) are autonomous and make local decisions based on exchanged information to reach global objectives. To this end, a certain negotiation protocol is used. Global system objectives are realised by means of mutual agreement. In contrast to centralised or hierarchical form, the heterarchical control architecture is flexible, transparent, modular and fault tolerant. To its disadvantages belong local instead of global optimization, communication overhead and the lack of global information.

The purpose of this paper is to present results of a project [van Dongen 1998] concerned with a simulation-based assessment of the performance of a system in which heterarchical control concepts are applied. To this end, the lithography area is studied of MOS4YOU, the latest wafer fabrication facility of Philips Semiconductors situated in Nijmegen, the Netherlands. The lithography area is the

bottleneck of the fab, and therefore, the performance of the entire system can be improved by improving the performance of the lithography area. Production of integrated circuits consists of a sequence of steps (about 300) that have to be performed. Generally, the ICs produced at MOS4YOU consist of 20 layers. Most layers are constructed using the following sequence of steps: deposition of material, patterning (lithography), etching, diffusion and dope implantation or metal deposition. MOS4YOU has the characteristics of a job shop (functional layout). Machines that perform the same operation are placed together in areas. In MOS4YOU, 9 different areas can be identified: lithography, implantation, furnace, dry-etch, wet-process, metallisation, dielectrics, metrology and PCM. Wafers are transported from area to area in an air-sealed box, called a SMIF-pod. An SMIF-pod contains 25 wafers, that all require the same operations. Some areas are visited many times, since ICs are fabricated in layers.

In the lithography area, lithographic patterns are formed on wafers using coating, patterning, and developing processes. The first step is to spin-coat a layer of photo resist. Subsequently, selective exposure of the integrated circuits on the wafers takes place. To this end, a mask—usually referred to as a reticle—is used. After exposure, the resist is developed and the regions that were exposed are removed. After development, several areas of resist remain. These areas protect the substrate regions they cover. Locations from which resist has been removed can be subjected to a variety of additive or subtractive processes. These processes transfer the pattern onto the substrate surface. Several inspection and measurement microscopes are used after this process, to inspect the wafers.

In the simulation study described in this paper, both qualitative and quantitative aspects are explored in order to be able to analyse the advantages as well as the disadvantages of heterarchical control. A good quantitative analysis of the performance of a heterarchical control system is only possible when the outcome of experiments with other control systems is at hand. Then, a comparison of the performances of both control systems can be made. Therefore, the model used here is based on a previous study of the lithography area of MOS4YOU described in [Rulkens 1996].

Models of the lithography area, which are elaborated in [van Dongen 1998] consist of autonomous components cooperating with each other by exchanging information along communication channels, as shown in [van de Mortel-Fronczak et al. 1995]. Exchange of information is modelled by synchronous message passing. Models can be validated by simulation [Naumoski & Alberts 1998] and formally verified [Bos & Kleijn 2000]. Simulation can also be used to assess communication overhead of the specified control system.

The paper is structured as follows. In Section 2, heterarchical control concepts are summarized and the negotiation protocol used is described. Section 3 focuses on the description of the hierarchical model of [Rulkens 1996], and on the heterarchical control system as developed in [van Dongen 1998]. In particular, the model structure and functionality of model components are discussed. The results of simulation are presented in Section 4. In Section 5, concluding remarks are presented.

Heterarchical control

In a heterarchical structure, autonomous control components are represented by agents that communicate with each other. In this paper, two different kinds of agents are described: job agents and workstation agents. Job agents are responsible for managing the job (or an order) from the moment it arrives till all tasks specified in the job have been carried out. To realise each task of the job, resources are necessary. Resource selection is achieved in a negotiation process consisting of several steps and resembling an auction in which bids are placed by the participating components (for instance, parts and machines). In this setting, it is possible to use several micro-economic price negotiation mechanisms [Lin & Solberg 1994]. In this paper, task announcement, bid collection, bid evaluation, task offer submission and task commitment monitoring are the negotiation steps performed by job agents. Resources are represented by workstation agents. Their goal in the negotiation process is to sell processing time to interesting parts. Availability announcement, task announcement monitoring, bid construction, bid submission, task offer collection, task offer evaluation and task offer acceptance are the negotiation steps performed by workstation agents. The communication between agents takes place through the network. Two different negotiation procedures can be followed: job-initiated negotiation, in which job agents take the initiative to negotiate, or workstation-initiated negotiation, in which workstation agents take the initiative to negotiate. In a job-initiated negotiation, a job agent announces that a task is to be processed. Workstation agents react if the workstations they represent are able to perform the task requested. The job agent then chooses a workstation according to a certain criterion. Also workstation agents may choose between different tasks. In a workstation-initiated negotiation, a workstation agent announces that its resource is available. Job agents that need this resource respond and the workstation agent chooses among tasks according to a certain criterion. Job agents also may choose between different workstations.

According to [Lin & Solberg 1994], a lightly loaded system calls for job-initiated negotiations and a busy system calls for workstation-initiated negotiations. This is caused by the negotiation efficiency. Job-initiated negotiations are very efficient in a lightly loaded system, but very inefficient in a busy system. If very few jobs are in the system, the probability that a job is selected by the workstation agent chosen is high, because it hardly has competition from other jobs. So it is very likely that the negotiation process will succeed. If the system is very busy, the workstation agent has the ability to choose among many requests, so the chance that a job will be rejected is high. A job that has been rejected has to be renegotiated, which implies that the negotiation is not performed efficiently. For workstation-initiated negotiations the opposite scenario can be drawn.

In this paper, a negotiation protocol partially derived from [Lin & Solberg 1994] and [Coenen 1995] is used. According to this protocol, job agents initiate the negotiation process by sending task announcements to a selected set of workstation agents. Only workstations that can actually perform the task requested are contacted.

Workstation agents monitor task announcements and reply positively or negatively depending on the state of the resource (idle, down, busy). The job agents that receive the positive replies select the “best” reply to submit bids. The workstation agents then select among bids according to a certain criterion [van Dongen 1998]. Job agents which bids are not accepted send a machine availability request to the same set of workstation agents. When a resource becomes available, the associated workstation agent contacts the job agents that have sent an availability request and the negotiation is restarted. The negotiation protocol is depicted in Fig. 1.



Fig. 1 – Job-initiated negotiation protocol.

In [Coenen 1995], workstation agents do not have the opportunity to choose among bids. This means that handling lots that belong to different priority classes becomes impossible. The protocol depicted in Fig. 1 does enable workstation agents to choose priority lots. In [Lin & Solberg 1994], job agents immediately restart the negotiation upon a bid refusal. This reduces the negotiation efficiency and causes overloading of the network, that might become the bottleneck of the system. In [Stronkhorst 1998], a preliminary request is used to ensure the presence of a vacancy in a machine. Every step in the negotiation is preceded with this request, whether the system is busy or not. In a lightly loaded system, the machine-availability message (in [Stronkhorst 1998]: the preliminary request) is not needed, because the chance that a job is selected is high. In a busy system, the machine-availability message ensures that a request from a job agent is submitted to a workstation agent when the resource is actually available.

The negotiation protocol consists of several phases:

1) *Task announcement*. For every task of the current order, the job agent sends a task announcement to a selected set of workstation agents. Only workstations that can actually perform the task requested are informed.

2) *Task announcement monitoring*. The workstation agent awaits task announcements from job agents and replies positively or negatively depending on the state of the resource (idle, down, busy).

3) *Task evaluation & answering*. All job agents that have sent a task announcement receive an answer. This means that the job agents representing interesting tasks receive a positive reply and that the job agents representing non-interesting tasks receive a negative reply. However, when a machine is down the reply will always be negative. Hence, job agents always receive an answer to a request. This is needed for a job agent to be able to choose the “best” workstation.

4) *Reply collection and evaluation*. The job agent awaits all replies possible and then selects the workstation that has the lowest work-in-progress at machine level. If no positive reply is received (because, for instance, all machines are down), it sends a machine availability request, and awaits a machine availability message to restart the negotiation process.

5) *Task offer submission*. After a workstation has been chosen, the job agent sends a task offer to the associated workstation agent.

6) *Task offer collection & selection*. The workstation agent awaits all task offers and then selects a task according to a certain criterion. All job agents, that have sent a task offer, receive an answer. This means that the job agents representing selected tasks receive a task offer acceptance, and that the job agents representing non-interesting tasks receive a task offer declination. The possibility for workstation agents to choose among tasks enables the system to handle lots that belong to different priority classes and to schedule at machine level. A priority job (sometimes referred to as hot lot) will always be selected prior to “normal” jobs.

7) *Task commitment monitoring*. The job agent will wait for a task offer acceptance message from the chosen workstation agent. If this message arrives, it means that the task will be performed in the associated workstation. The job agent sends

information about the lot to the transporter. If the message arrives that the task offer has been rejected, the job agent sends a machine-availability request to all workstations capable of performing the task requested.

8) *Machine-availability request.* If all replies are negative or when a task offer is rejected, the job agent sends a machine-availability request to all workstation agents whose workstations are capable of performing the task requested. Subsequently, the negotiation is restarted.

9) *Machine-availability monitoring.* If the task offer is rejected, the job agent sends a machine-availability request. When a resource becomes available again, the workstation agent reports this to the job agents that have sent a machine-availability request.

The result of the negotiation is that a task is performed in the workstation. When an operation is completed, the workstation agent reports this to the associated job agent. When all tasks in the order have been performed, the job agent returns the (finished) order and awaits a new order.

Models of the lithography area

In this section, two models of the litho area are globally described. The first equipped with a hierarchical control system [Rulkens 1996] and the second equipped with a heterarchical control system [van Dongen 1998]. System *Li* is formed by subsystems *Pat* and *Ins*, and controlled by C_0 . Controller C_0 decides whether a lot should be scheduled to system *Pat* (litho machines) or to system *Ins* (inspection and measurement machines). System *Li* is depicted in Fig. 2, system *Pat* in Fig. 3, and system *Ins* in Fig. 4.

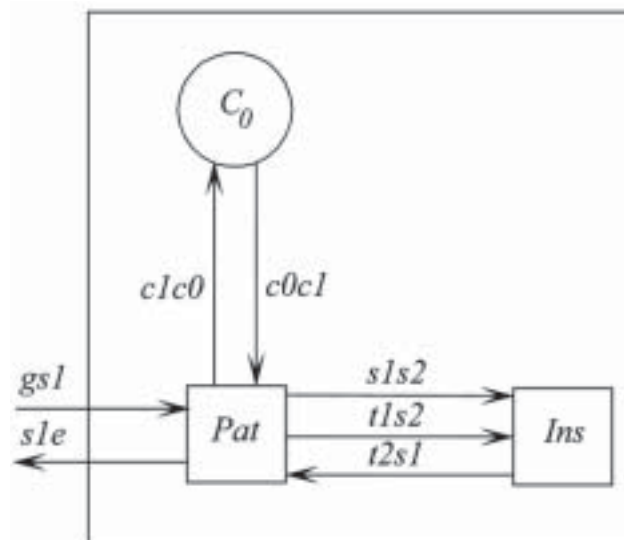


Fig. 2 – System *Li* [Rulkens, 1996].

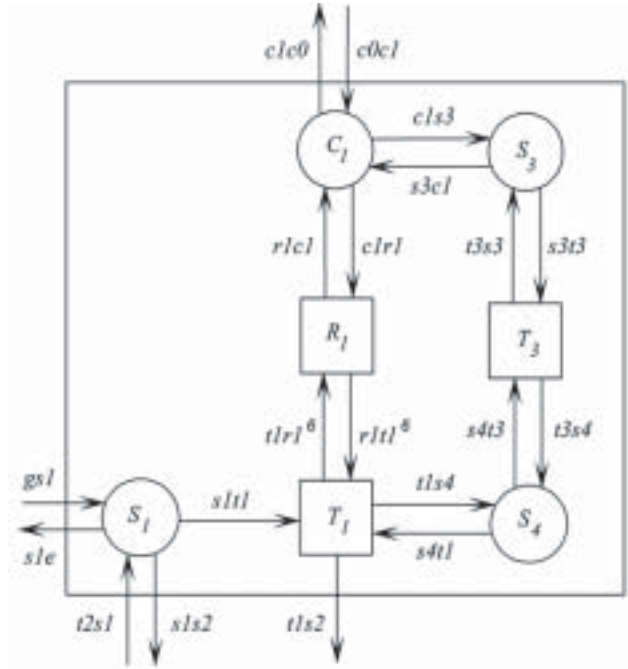


Fig. 3 – System Pat [Rulkens, 1996].

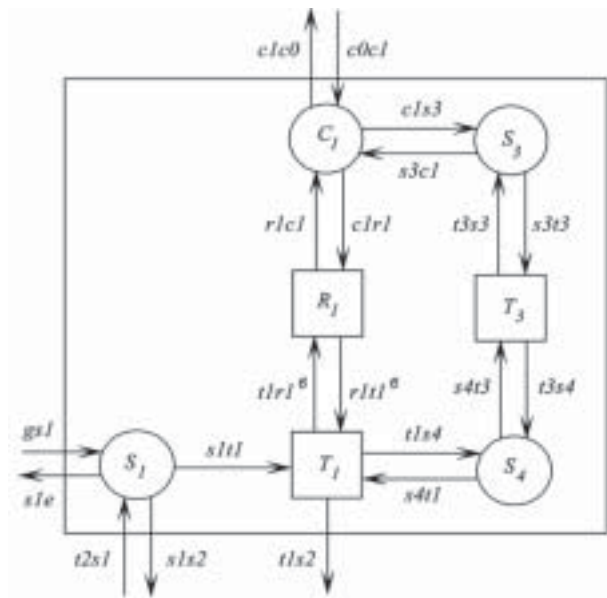


Fig. 4 – System Ins [Rulkens, 1996].

System *Pat* contains models of: stocker S_1 , reticle stocker S_3 (basement) and stocker S_4 (located in the cleanroom), transporter T_1 , reticle transporter T_3 , controller C_1 , and subsystem R_1 in which the litho machines are situated. There are four litho machines for critical lots and two litho machines for non-critical lots. The lots arrive at stocker S_1 , which informs controller C_1 and stores the lots. Controller C_1 is in charge of scheduling the lots to the litho modules. Scheduling a lot, means that controller C_1 sends a dispatch order to stocker S_1 . From stocker S_1 , the lot in question is transported by T_1 to the machine indicated by controller C_1 . On completion of an operation, transporter T_1 transports the lot to system *Ins*. The stepper needs reticles to process a lot. A reticle has to be transported to the cleanroom and this takes a certain time. A restriction is that only one copy of each reticle is available. In the model of [Rulkens 1996], controller C_1 not only evaluates the status of the litho machines (busy, idle or down), but also evaluates the location of the reticle required for the operation. If the reticle is already located at a litho module, then the controller schedules the lot to this machine. In case the required reticle is located in reticle stocker S_3 , the controller orders the reticle to the chosen litho module. It is transported by transporter T_3 from reticle stocker S_3 to stocker S_4 located in the cleanroom. Transporter T_1 performs the last step, which is the transport of the reticle to the indicated litho module. When the reticle is no longer needed in the cleanroom, it is returned to reticle stocker S_3 by transporters T_1 and T_3 .

System *Ins* contains stocker S_2 (the buffer in which lots wait for inspection or measurement), transporter T_2 , controller C_2 , and subsystem R_2 in which the inspection and measurement machines are situated. There is one macro inspection microscope, two visual inspection microscopes, two cd-sem microscopes, one overlay microscope, and two deep UV photo stabilisers. The material flow resembles the one in system *Pat*. Scheduling lots to a specific machine is performed by controller C_2 . System *Pat* and *Ins* can be distinguished by the fact that in system *Ins* the lots return to stocker S_2 on completion of an operation, where controller C_2 then determines the next location of the lot.

To implement a heterarchical control system, a general structure of Fig. 5 is used, which resembles the structures presented in [Veeramani et al. 1993], [Coenen, 1995], and [van de Mortel-Fronczak & Rooda 1997]. The model consists of order distributor O , j job agents J , k workstations $W_{0/1}$ (6 of type 0 and 8 of type 1), k (=14) workstation agents, reticle agent R , 3 transporters T , and communication network N . Reticle agent R is responsible for reticle handling. The components O , J , N , A and R form the actual control system of the model. Order distributor O sends orders to free job agents J through network N . Every job agent is responsible for the realisation of an order. To this end, the negotiation protocol described in previous section is used that is extended with a part related to reticle handling.

A job agent can hold one job at a time. This means that the maximal value of the work-in-progress is determined by the number of job agents. This implies that the number of job agents should be at least as high as the maximal value of the work-in-progress during operation.

The general control framework is as follows. Lots enter the system carrying a process plan (order). Lots are assigned to job agents. Then negotiations between job

agents and workstation agents are performed according to the protocol as depicted in Fig. 1. This negotiation process takes place by using communication network N . All job agents J and all workstation agents A are connected to network N . Subsystem N takes care of collecting messages for and from the components connected to communication network N and passes them to the correct destinations. The negotiation process between agents determines how a lot is scheduled. This means

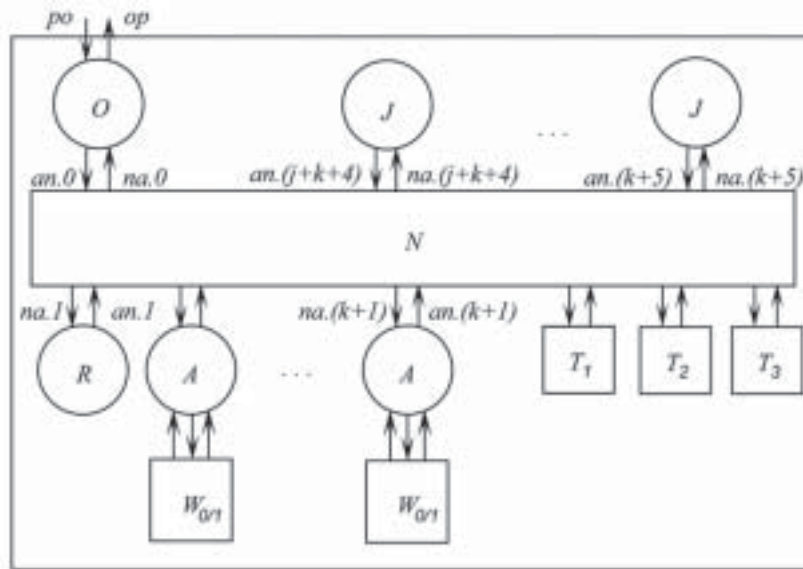


Fig. 5 – Model of the lithography area.

that handling and control are performed by agents. After job agent J has received a task offer acceptance from workstation agent A , the lot is transported by T to the associated workstation. On completion of each operation, workstation agent A sends an appropriate message back to the job agent J . Two transporters for the transport of lots are used, T_1 in system Pat and T_2 in system Ins . Reticle agent R has been introduced to implement and to structure the reticle flow in the model proposed (Fig. 5). The negotiation process for a litho operation is preceded with a reticle location request. This means that the negotiation process starts with job agents consulting reticle agent R about the location of the reticle required (only for the litho operation). In the case that the reticle is located in reticle stocker S_3 , the negotiation process is performed according to the negotiation protocol as depicted in Fig. 1. After a successful negotiation, job agent J orders reticle agent R to send the reticle required on transport (T_3) to stocker S_4 . From the stocker S_4 , transporter T_1 passes it to the machine specified in the message. In the case the reticle is already located at a certain litho machine, job agent J only negotiates with that particular machine. A reticle can also be on transport towards a machine, or towards reticle

buffer S_3 . In [Rulkens 1996], a lot that needs such a reticle cannot be scheduled. In [van Dongen 1998], a reticle being transported towards a machine is considered to be already there (in the negotiation). In the case that the reticle is on transport towards reticle stocker S_3 , it is also considered to be already there (in the negotiation). In this way, time is gained since lots can already be scheduled, when reticles are still on transport.

All processes and systems contained in the model of Fig. 5 are described only informally in the paper. Formal specification of the processes and systems can be found in [van Dongen 1998]. For the purpose of simulation experiments, material flow has not been specified explicitly. Moreover, the following assumptions are made:

- When software components (like agents) go down, no information is lost.
- When machines go down, the lot that is being processed is finished as if nothing had happened.
- The behaviour of the lithography area is studied in a steady state.
- A machine processes, at an average rate, one (or two for the litho module) lot at the same time without interruption (non-pre-emption).
- When a machine turns idle and a lot is available, the lot is processed immediately.
- A processed lot leaves the machine immediately.
- Transport does not fail. All lots and reticles sent on transport will reach their destination without delay.

For every operation on the litho module, a specific reticle is required. Every product for the litho module is uniquely identified by its product name and mask layer [Rulkens 1996]. Almost a half of the products is critical and the rest is non-critical. Every product is identified by the kind of reticle that is required for the operation on the litho module. Since every product is unique, for every product a unique reticle is required. In every order, the reticle (number) required for exposure is specified.

Order distributor O sends an order through network N to a free job agent J , which is responsible for the realisation of orders received. An order consists of several operations that have to be performed on the lot. The negotiation process for job agent J follows the (modified) protocol as described in this section. For every single operation, defined in the order, a negotiation process is started, which consists of the following sequence of steps. A task announcement is sent to those workstation agents A that can actually perform the task requested. This inquiry takes place by using communication network N . For a litho operation the task announcement is preceded by sending a reticle request message to reticle agent R . Reticle agent R gives in response the location of the reticle required. In the case that the reticle is being transported to a certain workstation, it is treated (in the negotiation) as if it were already there. If the reticle is on transport towards the basement, it is considered to be located in the basement (in the negotiation). If the reticle is located in the basement, job agent J sends a reticle send message to reticle agent R (only if the negotiation succeeds). In case the reticle is already located at a certain litho machine, job agent J only negotiates with that particular machine.

After sending task announcements, job agent J awaits the responses, and then selects among the positive replies upon a certain criterion which workstation agent

A is offered a task offer submission. Job agent J will wait for task offer acceptance from the chosen machine. When not a task commitment, but a task offer refusal or no reply at all arrives, job agent J sends an availability request and invokes the machine availability mode. An availability request is also used when after sending task announcements no positive replies arrive. On arrival of a machine availability announcement, the negotiation is restarted. After a successful negotiation process, job agent J sends the information about the lot to either transporter T_1 , which is situated in system Pat or to transporter T_2 , which is situated in system Ins . The choice between transporters depends upon the system (Pat or Ins) in which the operation must take place. System Pat contains the litho machines, and system Ins contains the inspection and measurement machines.

On completion of every task, workstation agent A returns the lot message to transporter T , which delivers it to job agent J . In case all tasks defined in the order have been completed, the order is returned to order distributor O . Since only information flow is treated in this paper, it is not necessary to inform the workstation that performed the previous task about the new destination of the lot (as described in [Coenen 1995]).

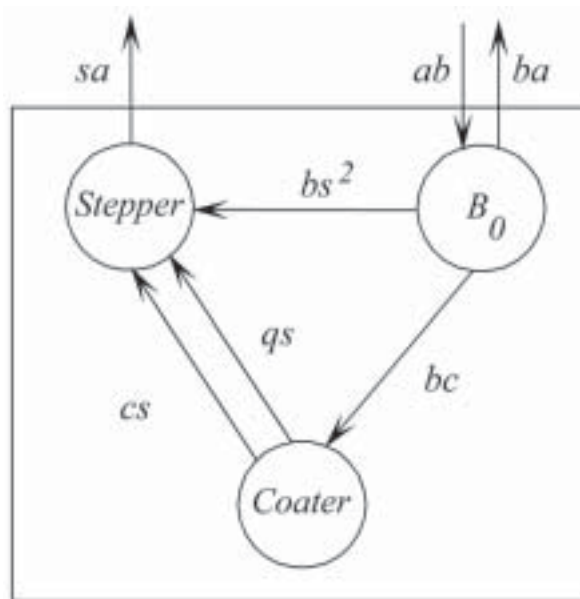


Fig. 6 – Workstation W_0 for the litho module.

For the litho modules and for the inspection and measurement machines and the deep photo stabilisers, two different workstation models W are used. Fig. 6 shows workstation W_0 for the litho module. It consists of machine buffer B_0 , process $Coater$ and process $Stepper$. The coater part of the track is represented by process

Coater and the stepper and developer part of the track is represented by process *Stepper*. Four workstations W_0 are able to process critical lots and two workstations are able to process non-critical lots. Workstation agent *A* negotiates with job agent *J* and sends received messages to machine buffer B_0 . Workstation agent *A* not only receives lot messages, but reticle messages as well. Both kinds of messages are sent to machine buffer B_0 , which sends the lot messages to process *Coater*, and the reticle messages to process *Stepper*. Lots can only leave the coater and enter the stepper if the required reticle is present. Machine buffer B_0 receives information from workstation agent *A* about future operations that have to be performed, and informs the stepper. In case a specific reticle is no longer needed on the stepper, it is sent to workstation agent *A*, which returns it to the basement. The litho module is able to continuously process wafers with a maximum of 2 lots at the same time. Processing the wafers (messages) happens virtually; this means information is simply delayed for a certain period of time and returned to workstation agent *A* on completion.

Workstation W_1 for the inspection and measurement machines and photo stabilisers is depicted in Fig. 7, and consists of machine buffer B_1 and machine *M*. It functions in the same way as workstation W_0 for the litho module. The only difference is that the microscope (or photo stabiliser) is modelled in one process, and that no reticles are needed for operation. One workstation W_1 contains macro inspection microscope, two contain visual inspection microscope, two contain cd-sem microscope, one contains overlay microscope and two contain deep photo stabiliser.

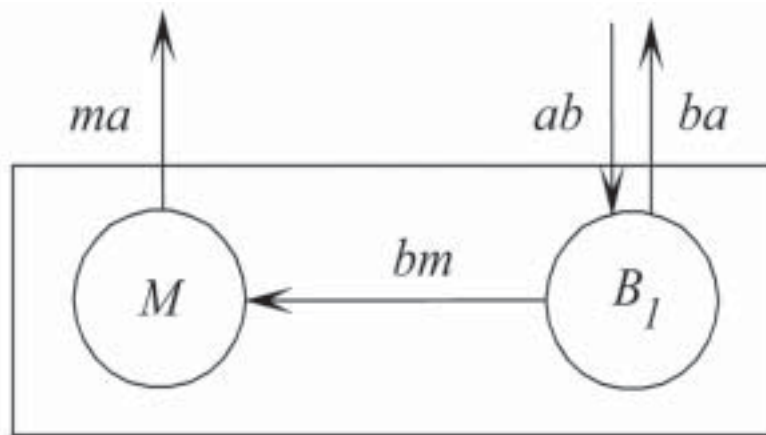


Fig. 7 – Workstation W_1 .

Each machine is represented by a workstation agent *A*, which communicates (according to the protocol described in Section 2 extended with the modifications described in this section) with job agents *J* using communication network *N*. It can receive five different kinds of messages: task announcement or task request, task offer, transport message, availability request, and reticle send message. In case of a

task announcement, the message received is passed to machine buffer *B*. This buffer gives the information whether the workstation is able to perform the task requested (i.e., the machine could be down). If the workstation is able to perform the task, buffer *B* informs workstation agent *A*. This workstation agent *A* passes the information to the associated job agent *J*. In case of a task offer workstation agent *A* awaits other task bids for a very short period of time, and then selects tasks according to a certain criterion. Moreover, workstation *W* has only room for a limited number of lots (two). Job agents *J*, whose tasks are declined, are informed properly. After selecting interesting tasks, workstation agent *A* confirms job agent *J* that the lot can be processed. Job agent *J* sends the message to transporter *T*, which passes it as a transport message with some delay to workstation agent *A*. Subsequently, the message is sent to machine buffer *B*. On completion of an operation, the machine returns the message to workstation agent *A*, which sends it back to transporter *T*. In case of an availability request, workstation agent *A* stores the request. As soon as a machine becomes available again, the associated job agents *J* are informed by sending an availability reply.

In extension of the protocol depicted in Fig. 1, reticle message handling is supplemented to the responsibility of workstation agent *A*. This means receiving reticle messages and passing them to machine buffer *B*, as well as returning the reticle messages and informing reticle agent *R* about it. Furthermore, workstation agent *A* informs the stepper (through machine buffer *B*) about the next product. In this way, the stepper knows if reticles are needed after completion of the current operation.

It should be mentioned that all workstation agents *A* are the same. Although a workstation agent *A* for a measurement machine does not need to handle reticles, the specification of the workstation agent *A* resembles the one for a litho module agent. This enhances simplicity in the specification, because only one kind of workstation agent *A* is needed.

The main function of reticle agent *R* is to store the location of the reticles, and to function as a storage for the reticles (can be considered as the basement). It can receive four different kinds of messages: a reticle location request, a reticle send message, a reticle return announcement message, and a reticle return message. Job agent *J* starts the (litho) negotiation by sending a reticle location request to reticle agent *R*, which gives in response the location of the reticle required. A reticle can be located in three different places: in the basement, at a specific machine, or on transport. Transport can be towards a specific machine, or towards the basement. In [Rulkens 1996] a lot cannot be scheduled, if the reticle required for the operation on the litho module, is on transport. In [van Dongen 1998], a different approach is followed. If job agent *J* inquires about the location of a reticle which is on transport towards a certain location, reticle agent *R* gives in response the identification number of this location. On arrival of a reticle send message, the requested reticle is sent to the indicated machine. If a reticle is no longer needed at a machine, workstation agent *A* informs reticle agent *R* about the return of the reticle and sends the reticle message on transport. If job agent *J* inquires about the location of a reticle which is

on transport towards the basement, reticle agent R gives in response that the reticle is already located in the basement. When job agent J (after a successful negotiation) sends a reticle send message, this message is stored and upon the return of the reticle, it is sent to the machine indicated. The advantage of this approach is clear. Time is gained, because lots can already be scheduled to machines, when the reticles are still on transport.

A scenario can be drawn, in which job agent J sends a reticle location request just after the stepper has returned the reticle indicated to the basement. This means that the reticle, upon arrival in the basement, is transported back to a machine (not necessarily the same machine). All this enhances unnecessary transport, and additional waiting time. However, since the number of different products is reasonably large, the chance that this situation would occur is very small.

In the model of [Rulkens 1996], transporters are used, which transport lots and reticles from one location to the specified location. In order to model transport (-time), transporters T are introduced. Transporter T simply delays messages for a certain period (transport time) and sends them after a certain transport time to the agent specified in the message. Transporter T can hold a certain number of messages (lots), and receives them from either job agent J , workstation agent A or reticle agent R . The capacity of transporter T depends upon the number of operators present.

Network N is modelled as shown in [van de Mortel-Fronczak & Rooda 1997]. Network N contains a certain number of interface components I , that collect messages for and from the components connected to communication network N . Switch S is responsible for passing the messages from the source to the correct destination.

This section describes the model of the lithography area equipped with a heterarchical control system. All machines present in the area have been implemented in the model. Moreover, operators, reticles and transport have been taken into account. The heterarchical control system schedules the lots to the different machines.

Simulation results

In this section, the results of two experiments that have been performed with the model are described. The first experiment has been performed under the same circumstances as model B of [Rulkens 1996]. The second experiment has been performed using the environment of model D of [Rulkens 1996].

In order to validate the model, experiments have to be performed, that are described in [Rulkens 1996]. Only when the results correspond to the results as described in [Rulkens 1996], it is plausible to assume that the fmodel is valid. In model B of [Rulkens 1996], the input consists of only two kinds of orders: a non-critical lot and a critical lot with specific routings. The lots are processed in equal numbers. Moreover, no priority exists and transport time equals zero. Furthermore, no reticles are required for a litho operation, and machines never fail.

The results of simulation are depicted in Figs 8 & 9. The straight lines represent the theoretical values of the cycle time and the output. The values of both the cycle time and the output correspond exactly to the results of model B in [Rulkens 1996].

This means it can be assumed that the model is valid. Moreover, it means that the model equipped with a heterarchical control system is capable of producing the same results (in this situation) as model B of [Rulkens 1996].

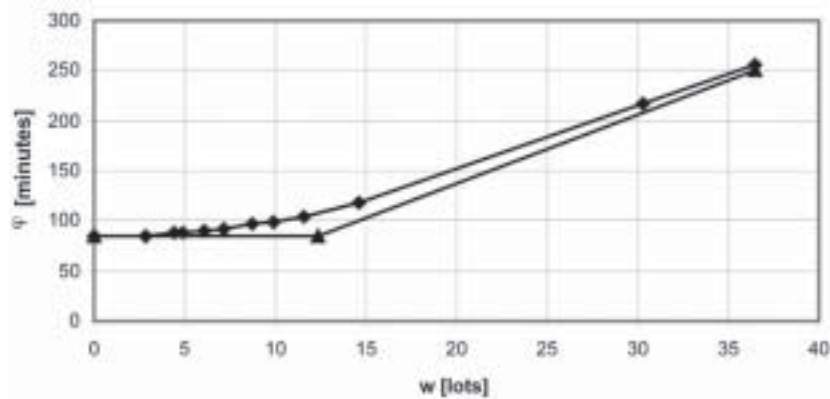


Fig. 8 – Cycle time versus work-in-progress.

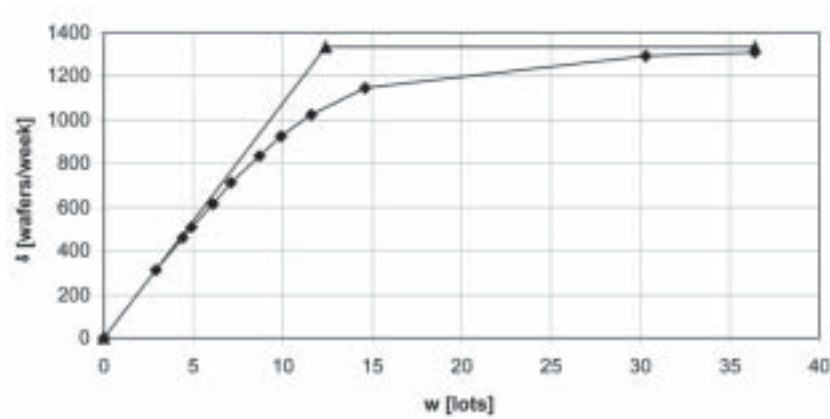


Fig. 9 – Output versus work-in-progress.

In the second part, the same input as used. In the experiment, transport time, reticles, priority and operators are all included. Machines are still considered never to fail. Two experiments have been performed with the model. One with no priority lots, and one with 20 % of the lots having a high priority. The results are depicted in Figs 10 & 11. The straight lines represent the theoretical values. In the situation without hot lots, the cycle time and the output correspond to the expectations. Verification performed in [van Dongen 1998] shows that the model performs as theoretically can be expected. In the situation with hot lots, the normal lots obtain extra delay, the hot lots take precedence and the output is exactly the same as in the situation without hot lots.

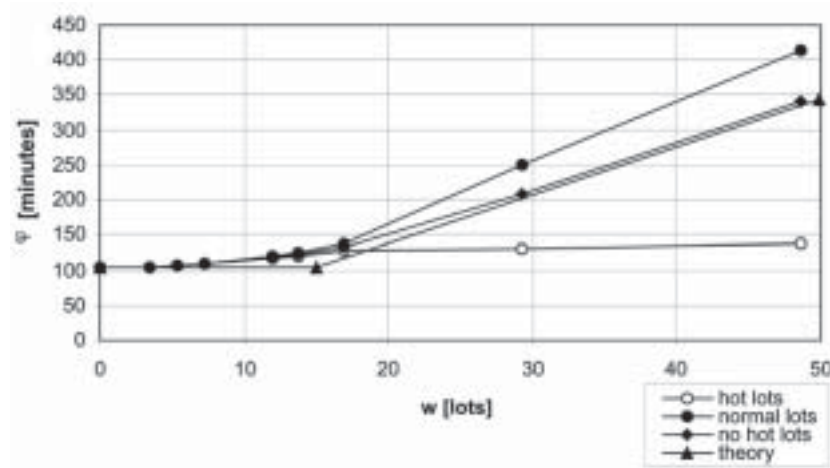


Fig. 10 – Cycle time versus work-in-progress.

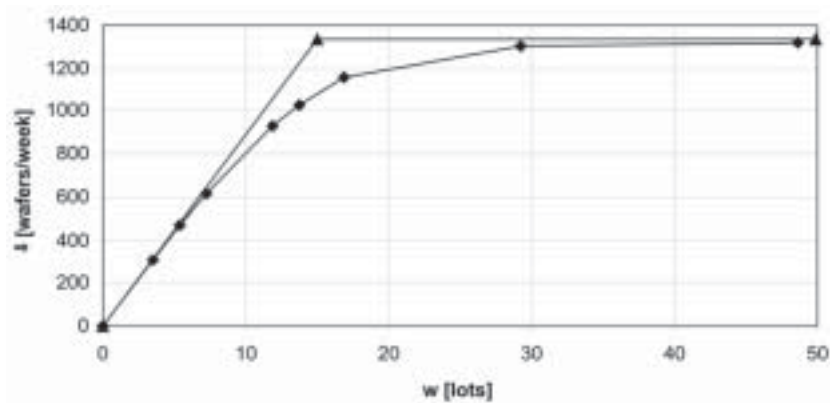


Fig. 11 – Output versus work-in-progress.

Simulation experiments show that the performance of the model equipped with a heterarchical control system is identical to the performance of the model of [Rulken 1996]. Additionally, the heterarchical control system performs as theoretically can be expected. This means that a heterarchical control system is capable of producing the same results as an alternative control system, in a similar environment.

Concluding remarks

According to literature, foreseen or unforeseen addition or removal of components is very easy in a heterarchical environment. Experiments with the model presented in Section 3 support this conclusion. Only minor changes related to agent identification numbers and address functions are necessary.

A heterarchical control system is often referred to as a system that can cope very well with equipment failures. In contrast to the hierarchical system, no difficult algorithms are needed to handle errors such as machine or tools failures. The corresponding workstation agent simply withdraws itself from the negotiation process. In this way, the remaining components of the system continue to operate without interruption caused by the error. This enhances robustness of the system. The heterarchical control system as presented in Section 3 is very well capable of dealing with machine failures. The negotiation continues without being affected by machine failures. However, more job agents J are needed, because jobs remain longer in the system.

In a hierarchical system, the different control layers strongly depend on each other. When a control layer, high in hierarchy, goes down, usually the control layers beneath it come to a halt. This affects robustness of the entire system. A heterarchical system is more capable of handling control components going down. In this paper, it is assumed that no information is lost when control components go down. Job agents are not affected by workstation agents going down. If after sending a task announcement to several workstation agents not every agent replies, the job agent waits a very short period of time. After this waiting period, it chooses among the positive replies. This means that workstation agents may go down without influencing the job agent. Only when the job agent chooses a workstation, and sends a task offer, it awaits a task offer acceptance message. If a workstation agent is down, this message is not sent until the workstation agent becomes active again. The job then has to wait. The chance for this situation to occur is very small, since the time limit in which the situation can happen, is very small. Because job agents wait until all workstation agents respond within the (very small) time limit, the cycle time of the lots increases by this time limit.

Workstation agent A is not affected when job agents go down. If job agents go down when they are not negotiating, workstation agent A does not notice that. The negotiation process simply is continued with other agents. Even when a negotiation between a job agent and a workstation agent is actually taking place, and the job agent goes down, the workstation agent does not notice this. Only when job agent J sends a task offer, and is actually chosen by workstation agent A , the stepper is informed, and the work-in-progress at machine level is increased by 1. This is reported to job agent J , but it is down, so the message is stored until job agent J comes back up. Job agent J does not send a lot message, and (if necessary) does not send a reticle send message to reticle agent R . The result is that one place in the machine is blocked, and this could mean capacity loss. If this happens on a regular basis, preventive measures have to be taken. However, since the time base in which this situation can occur is very small, robustness of the system is hardly affected.

Reticle agent R is responsible for reticle handling. If it goes down, reticles cannot be sent to the stepper, the location of the reticles cannot be passed to the job agents (job agents will wait for response), and the entire negotiation process concerning the litho step comes to a halt. This problem is not really related to the heterarchical control architecture itself. In principle, the negotiation can commence without

consulting reticle agent R , and mutual agreements can be achieved. Products can be sent to workstations, and on completion, returned to their job agents. However, the production step on the stepper requires the presence of the correct reticle. So all negotiation is useless, if the reticle cannot be sent to the correct machine.

Simulation experiments show that the heterarchical control system described enables the lithoshop to perform equally well as the hierarchical control system does. The heterarchical control system can be modified or extended very easily, offers transparency and is very well capable of handling machine and even control component failures. The greatest disadvantage of the control system is the overhead of communication flow.

The hierarchical control system of [Rulkens 1996] is more difficult to modify (i.e. unforeseen machines or conveyors are difficult to implement), and offers less transparency because of the size of the controllers. Machine failures can be dealt with, but control components going down is devastating for the system. Summarising it can be said that the qualitative benefits as described in the literature (flexibility, transparency, modularity and high fault-tolerance as opposed to rigidity, sensitivity to failures and difficulty to modify and/or extend), also apply to the system as described in this paper. The results and analysis show that a heterarchical control system can compete with a hierarchical control system at both quantitative and qualitative level, and that qualitative aspects are much easier implemented in the control system.

The heterarchical control system has been partially derived from [Lin & Solberg 1994] and [Coenen 1995] and resembles the one described in [Stronkhorst 1998]. An important difference in comparison to the agents in [Stronkhorst 1998], is that the agents as described in this report, are capable of handling control component failures. The two most important differences in comparison to [Lin & Solberg 1994], [Coenen 1995], and [Stronkhorst 1998] is handling of reticles.

A disadvantage that is often mentioned in the literature is the overhead of communication. During simulation, the amount of communication increases with the work-in-progress. For high wip levels, the overhead of communication is high (sometimes very high), since many agents have to negotiate with each other to reach a decision (thus the amount of communication is high). In [Stronkhorst 1998], an effort is made to reduce the amount of communication. No job agents, but operation agents are used. For every operation (in this paper 7), an operation agent exists. This agent represents a group of lots, and not a single lot. The group of lots is sorted according to a certain criterion (for instance, FIFO), and the operation agent decides which lot is offered to the workstation agents.

The simulation model can be used to examine the dynamic behaviour of the lithography area. Different priority classes and different process flows can be used. It is possible to analyse the influence of the addition of machines, operators, machine and component failures, and reticle flow on the performance of the system. Scheduling at machine level is possible. In this paper, the FIFO strategy was used. However, other scheduling techniques can easily be implemented in the control system.

Acknowledgements

The authors would like to thank J. van Dongen and V.I. Stronkhorst, the Mechanical Department students of Eindhoven University of Technology, for their valuable contribution to the project. H.J.A. Rulkens and E.J.J. van Campen of Philips Semiconductors are acknowledged for their cooperation.

References

- Bos V. & Kleijn J.J.T. (2000) *Discrete ? : a production systems modelling language*. Technical report. Eindhoven University of Technology, Department of Mathematics and Computing Science. Eindhoven, the Netherlands.
- Coenen F.W.J. (1995) *Een heterarchische besturing voor flexibele productiesystemen*. Master's thesis. Eindhoven University of Technology, Department of Mathematics and Computing Science. Eindhoven, the Netherlands, (In Dutch).
- Dilts D.M., Boyd N.P. & Whorms H.H. (1993) *The evolution of control architectures for automated manufacturing systems*. Journal of Manufacturing Systems 10(1), 79-93.
- Lin G.Y. & Solberg J.J. (1994) *Autonomous control for open manufacturing systems*. In: Computer Control of Flexible Manufacturing Systems, Research and Development (S.B. Joshi and J.S. Smith, Eds.), 169-206. Chapman and Hall: London.
- Naumoski G. & Alberts W. (1998) *A discrete-event simulator for Systems Engineering*. PhD thesis. Eindhoven University of Technology, Department of Mechanical Engineering. Eindhoven, the Netherlands.
- Rulkens H.J.A. (1996) *Analysis and optimization of the lithography area of MOS4YOU*. Final report. Eindhoven University of Technology, Stan Ackermans Institute. Eindhoven, the Netherlands.
- Stronkhorst V.I. (1998) *Heterarchical control of the {MOS4YOU} furnace area*. Eindhoven University of Technology, Department of Mechanical Engineering. Eindhoven, the Netherlands.
- van de Mortel-Fronczak J.M. & Rooda J.E. (1997) *Heterarchical control systems for production cells-a case study*. In: Proceedings of MIM' 97, 243-248. Vienna, Austria.
- van de Mortel-Fronczak J.M., Rooda J.E. & van den Nieuwelaar N.J.M. (1995) *Specification of a flexible manufacturing system using concurrent programming*. The International Journal of Concurrent Engineering, Research & Applications 3(3), 187-194.
- van Dongen J. (1998) *Heterarchical control of a lithoshop*. Master's thesis. Eindhoven University of Technology, Department of Mechanical Engineering. Eindhoven, the Netherlands.
- Veeramani D., Bhargava B. & Barash M.M. (1993) *Information system architecture for heterarchical control of large FMSs*. Computer Integrated Manufacturing Systems 6(2), 76-91.

ALISTAIR R. CLARK

A local search approach to lot sequencing and sizing

Faculty of Computing, Engineering and Mathematical Sciences
University of the West of England
Bristol, England

Abstract — This working paper reports ongoing research into the simultaneous sequencing and sizing of production lots on a set of parallel machines in the presence of sequence-dependent setup times. A flexible mixed integer programming (MIP) model is presented which is optimally solvable only for very small instances. As an alternative, a local search approach is discussed and outlined, making use at each search iteration of an assignment patching algorithm to determine efficient sequences of setups, followed by the dual simplex method to determine optimal lot sizes for that sequence. A solution and varying-sized neighbourhood structure is proposed that will hopefully permit the search to efficiently explore a wide range of solutions and yet eventually home in on a good solution. Computational tests have yet to be carried out.

Keywords — production, lot-sizing, sequencing, heuristics.

Introduction

The ongoing research reported here is concerned with the simultaneous sequencing and sizing of production lots on a set of parallel machines when a sequence-dependent setup time is incurred to change production between lots of different types. Production is organised in lots in order to meet varying periodic demand under conditions of tight capacity. If lot sequencing and sizing is badly planned and managed, then inventory will be larger than necessary and, worse, setups times will consume scarce machine time, causing excessive unmet demand and backorders.

Many companies face the challenges of managing setup times, particularly in industries where capital investments in production capacity are large and product variety is diverse. Simple approaches such as dispatching rules sometimes function well where the sequencing of fixed-sized orders is concerned, but when the sizing of production lots is also part of the planning decision, we are left to confront a very tough problem indeed.

Little research has been carried out into the problem, perhaps because it is so difficult to solve optimally for anything other than very small sized instances. It is NP-hard, so that it is very unlikely that an optimal solution method exists which is efficient for medium sized problems upwards. Consequently, rather than pursue the fruitless goal of trying to develop a fast optimal procedure, this research explores heuristic approaches that permit the identification of good solutions in a reasonable amount of time. To the industrial user, an optimal solution is an invisible yardstick whose value is not known. Furthermore the often dubious quality of production data and frequent updating of demand forecasts means that a supposedly optimal production plan would in fact be optimal for only one sample point among many

millions. A possible approach to overcoming this objection might be a sophisticated stochastic model, but this would have impossible data needs, requiring an knowledge of the probability distributions of the quality of production data and demand forecasts. Instead of following such a hopelessly complex road, a more useful approach would be the development of a lot sizing and sequencing method whose results are generally good, robust and quickly obtainable for industrially sized problems. The purpose of this research is to follow just such a path.

Previous research in this area is rare. A recent survey into lot-sizing [Drexel & Kimms 1997] noted very few papers that studied the problem of lot sequencing as well as sizing. The general problem that interests us includes representation of setup times that are sequence-dependent and permits multiple setups within a planning period. In addition, it does not require that all or none of the production capacity on a machine be utilised within a period. As such, it is related to the capacitated lot-sizing problem (CLSP) as defined in [Drexel & Kimms 1997], although the CLSP does not include sequencing decisions.

Clark and Clark (2000) developed a mixed integer linear programming (MIP) model that allowed an arbitrary number of setups in each period and showed how it could be simplified when applied under a rolling horizon. The resulting model, presented below, was still only practicable for small to medium sized problems, even when solved merely approximately within a branch-and-bound search. It motivated the application of the more flexible approach presented in this research where the sequencing of lots will be tackled through local search rather than mixed integer programming. Once a sequence is defined, the sizing of lots can be quickly optimised via linear programming.

A MIP model for lot sizing and sequencing

Our aim is to satisfy the demand for P products over the T planning periods with minimal backorders or inventory carried over from one period to the next. The products may be manufactured in batches of varying size on any one of M machines in parallel to each other. A changeover from one product to another requires a setup time during which the machine cannot process any products. The machines are not identical and may have different rates of production and setup times. There is no restriction on the number of setups in each planning period, but since it makes no sense to produce a product in more than one batch on the same machine in the same period, there will in practice be at most one setup per product per period and so we can limit the number of setups on machine to P per period.

Suppose we permit up to $N - P$ setups per period on each machine. Then a MIP formulation of the problem is:

Model MIP:
minimise

$$\sum_{i,t} (h_i I_{it}^+ + g_i I_{it}^-) \tag{1}$$

such that

A local search approach

$$I_{i,t-1}^+ - I_{i,t-1}^- + \sum_{m,n} x_{imt}^n - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (2)$$

$$\sum_{j,n} (\sum_i s_{ijm} y_{ijmt}^n + u_{jm}^n x_{jmt}^n) \leq A_{mt} \quad \forall m, t \quad (3)$$

$$y_{ijm1}^1 = 0 \quad \forall i, m, j \neq i_{0m} \quad (4)$$

$$\sum_j y_{ijm1}^1 = 1 \quad \forall m \quad (5)$$

$$\sum_i y_{ijmt}^n = \sum_k y_{jkmt}^{n+1} \quad \forall j, m, t, n = 1, \dots, N-1 \quad (6)$$

$$\sum_i y_{ijm,t-1}^N = \sum_k y_{jkmt}^1 \quad \forall j, m, t = 2, \dots, T \quad (7)$$

$$x_{jmt}^n \leq M_{jmt} \sum_i y_{ijmt}^n \quad \forall j, m, n, t \quad (8)$$

$$y_{ijmt}^n = 0 \text{ or } 1 \quad \forall i, j, m, t \quad (9)$$

$$x_{imt}^n \geq 0 \quad \forall i, j, m, t \quad (10)$$

$$I_{it}^+ \geq 0; I_{it}^- \leq 0 \quad \forall i, t \quad (11)$$

where the decision variables are:

y_{ijmt}^n = with value 1 if the n -th setup on machine m in period t is from product i to product j , and with value 0 otherwise.

x_{imt}^n = Quantity of product i produced between the n -th and $n+1$ -th setups on machine m in period t (it is non-zero only if the n -th setup on machine m is to product i).

I_{it}^+ = Stock of product i at the end of period t .

I_{it}^- = Backlog of product i at the end of period t .

The parameters and data inputs are:

d_{it} = Demand for product i at the end of period t .

A_{mt} = Available time on machine m in period t .

s_{ijm} = Time needed to setup from product i to product j on machine m .

u_{im} = Time needed to produce one unit of product i on machine m .

h_i = Cost of holding one unit of product i from one period to the next.

g_i = Penalty cost of a carrying over a backorder of one unit of product i from one period to the next.

j_{0m} = The product produced on machine m at the end of period 0, i.e. the starting setup configuration on machine m .

and where

$$M_{imt} = \min\{A_{mt}/u_{im}, \sum_{t=1}^T d_{it} + I_{i0}^- - I_{i0}^+\} \quad (12)$$

is an upper bound on x_{imt}^n , since all backlogs and future production of product i might in theory be produced on machine m .

Note that, unlike many formulations in the literature, model MIP allows backlogs. To prohibit them is unrealistic—many companies face occasional or frequent capacity overloads and they often have no immediate choice but to backlog demand. The question of what penalties g_i should be allocated to backlogs can only be answered for each individual context, depending on market conditions and the importance of the customers for particular product. Since the value of such penalties will partly be based on judgement and imprecise information, there is little added value in investing the often huge extra effort needed to solve the model optimally rather than approximately. Only holding and backlog penalty costs are included in the objective function (1) since these are our major concerns. We assume that the total costs of the provision of the production capacity A_{mt} are fixed and do not depend on the production decision variables x_{imt}^n and y_{ijmt}^n . Additional setup and direct production costs are not included in the objective function as they are likely to vary little or be negligible in comparison to the penalty costs of the additional backlogs provoked by the lost machine time that an inefficient sequence of setups would cause.

However, if need be, such costs can be incorporated into the objective function without difficulty by the inclusion of a term such as $\sum_{j,t,n} [\sum SetupCost_{ijmt} y_{ijmt}^n + UnitCost_{jm} x_{jmt}^n]$.

Constraints (2) are the standard equations linking inventory, production and demand while constraints (3) represent the limited availability of capacity. Note that since the machine time capacity parameter A_{mt} is indexed on t , the production periods t in the model may be of different lengths. For example, weekend working may be combined into a single planning period.

Constraints (4) to (7) ensure that a setup on a machine must and may only occur between a single pair of products, possibly both the same product, and that if a certain product is changed to, then it must be changed from in the following setup. The equals sign =, rather than the sign \leq , is necessary in constraints (4) to (7) so that we always know for which product a machine is configured, especially when it is not producing. Thus the combination $y_{ijmt}^n = 1$ and $x_{imt}^n = 0$ must be allowed. Note that constraints (4) to (7) require that for each triple (n, m, t) there is exactly one pair (i, j) for which $y_{ijmt}^n = 1$, i.e., there must be precisely N setups in each period on each machine, even if a setup $y_{ijmt}^n = 1$ is just from a product i to itself. Since a setup time s_{iim} from a product i to itself is zero, the model does not force a machine to have exactly N positive-time setups but rather up to N such setups. The remaining zero-time setups are modelling phantoms and do not exist in reality.

Constraints (8) ensure that there must be a setup if a product is produced on a machine in a period, even if it is just a phantom one from a product to itself. The first setup in a period on a machine must occur at the beginning of the period, but the subsequent $N-1$ setups may occur at any time within the period. If the constraints $y_{iimt}^n = 1, \forall i, m, t$ and $y_{ijmt}^n = 0, \forall i, j, m, t | i \neq j$ are imposed, then model MIP is restricted to just $N-1$ setups per period, but these may occur at any time within the period.

Thus the model is related to (and more general than) the proportional lot-sizing and scheduling problem (PLSP) which allows the single permitted setup in each period to occur at any time within the period, if at all [Drexel & Haase 1995; Drexel & Kimms 1997]. Model MIP does not, however, permit a setup to begin in one period and finish in the next, so it is not totally flexible.

In [Clark & Clark 2000] it was shown that there are limits to the size of problem that model MIP can solve in practical time. A limit on the time spent searching for a solution can be imposed, an approach which is easy to implement in most MIP solvers, but the computational results suggested that for medium to large problems impractical amounts of time will be spent just identifying a feasible solution.

A local search approach to lot sizing and sequencing

An alternative approach is to use a solution method where lot sequencing is solved by local search [Aarts & Lenstra 1997] and lot-sizing by linear programming (LP). This means that a local search solution is uniquely identified by a sequence of a subset of distinct numbers from the set $1, \dots, P$ for each pair (m, t) of machines and periods. The optimal lot-sizes associated with a given setup sequence $y_{ijm}^n \forall i, j, m, n, t$ obeying constraints (4) to (7) are found by solving the following LP:

Model LotSizes:

$$\text{minimize } \sum_{i,t} [h_i I_{it}^+ + g_i I_{it}^-] \quad (13)$$

such that

$$I_{i,t-1}^+ - I_{i,t-1}^- + \sum_{m,n} x_{imn}^n - I_{it}^+ + I_{it}^- = d_{it} \quad \forall i, t \quad (14)$$

$$\sum_i u_{im} x_{imn}^n - \sum_{i,j,n} s_{ijm} y_{ijm}^n \leq A_{mt} \quad \forall m, t \quad (15)$$

where $x_{imn}^n = 0$ if type i must be setup on machine m in period t , and $= 0$ otherwise.

Thus, between successive sequences in a local search, the LP will retain the same objective function, but the right hand sides of only some constraints will change and, in addition, the upper bounds of some x_{imn}^n variables will have to be set to zero or infinity. This means that objective function updating between successive sequences in a local search can be done (quickly we hope) by reoptimising the LP using the dual simplex method.

Furthermore, on each machine there will in practice be at most one setup per product per period and so we can limit the number of setups on a machine to P per period. Thus for a given machine-period pair (m, t) and unordered subset B_{mt} of all P products, we can minimise the value of the total time lost to setups:

$$\sum_{i,j,n} s_{ijm} y_{ijm}^n \quad (16)$$

by an optimal sequencing of the setup times of the products in the subset. This will in turn give the best possible value of the model LotSizes for the given pair (m, t)

and subset B_{mt} . Finding an optimal sequence of products to minimise expression (16) is equivalent to solving an Asymmetric Travelling Salesman Problem (ATSP). This is only viable for very small subsets of products if (16) is to be minimised many times (up to PT) at every local search iteration. However, since we are focusing on finding good (rather than optimal) solutions for model A, a fast ATSP heuristic may be used to minimise (16). The usual symmetric TSP heuristics such as the k -opt [Cook et al. 1998] or the Lin-Kernighan [Lin & Kernighan 1973] tour improvement methods are not necessarily the best heuristics to use for the asymmetric TSP. Instead, a fast procedure that gives good and often near-optimal results is to solve an assignment problem and then use a patching heuristic to convert assignment subtours into a single salesman tour (Karp 1979; Karp & Steele 1985; Frieze & Dyer 1990).

If a machine m is already setup for type i_0 at the end of period $t-1$ and must produce type j_{99} after the first setup of period $t+1$, then the assignment problem to be solved is:

Model AP:

$$\text{minimise } \sum_{i,j \in B_{mt}} s_{ij} z_{ij} \quad (17)$$

such that

$$\sum_j z_{i_0 j} = 1 \quad (18)$$

$$\sum_j z_{ij} = 1 \quad \forall j \in B_{mt} \quad (19)$$

$$\sum_j z_{ij} = 1 \quad \forall i \in B_{mt} \quad (20)$$

$$\sum_j z_{ij_{99}} = 1 \quad (21)$$

where the assignment decision variables are:

$z_{ij} = 1$ if product i is assigned to product j , i.e., if there is a setup from product i to product j ; and 0 otherwise.

Thus at each local search iteration, an assignment patching algorithm is used to determine efficient sequences of setups, followed by the dual simplex method to determine optimal lot sizes for that sequence.

Solving just model AP (without applying the patching heuristic) would be fast and could well provide a good approximate reflection of the optimal value of the ATSP optimal solution that could be used in (15).

Various local search strategies such as Simulated Annealing [Egglese 1990; Dowsland 1993] and Tabu Search [Glover & Laguna 1993] have been proposed to encourage a search not to get stuck in a local optimum, but rather to get very near a global optimal solution, and often involve considerable tuning and effort. However, is such precision appropriate in the messy world of production planning and scheduling where the input data is often imprecise and upsets such as rush orders or

machine failure are common, necessitating frequent replanning? A more useful outcome would be a quickly-obtained solution of reasonable quality, especially for medium to large sized problems where optimal seeking methods would take an impractically long time to converge. What implications does this have for the use of local search and the neighborhood structure?

At one extreme, a small structure would mean a very slow convergence rate, coupled with the risk of getting trapped in a local optimum. At the other extreme, a very wide ranging neighborhood structure would lean in the direction of random sampling among all possible solutions. Random sampling has the advantage that it would avoid entrapment in a local optimum, but is possibly inefficient in identifying good solutions by nature of its randomness. A very wide range of neighbours at the start of a local search might also to some extent avoid the worst local optima, and could then be narrowed later in the search to home in to a good local optimum. The challenge here is to find out how to do this efficiently and map the trade-offs between quality of solution and speed of computation.

In order to start the local search with a large neighbourhood and then gradually reduce it, we propose to generate a neighbour by carrying out a large number of random insertions and deletions of products in the sets $B_{mt} \forall (m,t)$. As the search progresses, we will slowly reduce the number of such insertions and deletions. Specifically, the procedure is:

1. Identify a starting solution $\{B_{mt} \forall (m,t)\}$
2. Let N be a large integer.
3. For $n = 1$ to N do
 - 3.1. Randomly select a pair (m,t) .
 - 3.2. Randomly choose whether the next change is an insertion or a deletion.
 - 3.3. If an insertion, then randomly select $b \notin B_{mt}$ and insert b into B_{mt} .
 - 3.4. If a deletion, then randomly select $b \in B_{mt}$ and delete b from B_{mt} .
4. Solve model AP/ATSP and then model LotSizes.
5. Adopt and record the solution if it is the best so far.
6. Reduce N occasionally. Stop if you wish.
7. Go to step 3.

As there are MT sets B_{mt} , each of maximum size P , all solutions are reachable if the search is started with an initial value of $N = PMT$. A fast rate of reduction of N may well not produce as good a final solution as a slower rate, but we cannot make firm statements about this without further experimental investigation.

At the time of writing, experimental tests are being designed to test the comparative effectiveness of the following procedures:

1. Random Sampling of a set B_{mt} for each pair (m,t) , to serve as a benchmark.
 2. Biased Sampling of a set B_{mt} for each pair (m,t) .
 3. Local Search among all possible solutions $\{B_{mt} \forall (m,t)\}$ using the N random insertions/ deletions neighborhood structure described above, with varying initial values and reduction rates of N .
 4. Other Local Search procedures among all possible solutions $\{B_{mt} \forall (m,t)\}$.
- The results will be reported in the future.

References

- Aarts E.H.L. & Lenstra J.K. (eds) (1997) *Local Search in Optimization*. John Wiley and Sons: New York.
- Clark A.R. & Clark S.J. (July 2000) *Rolling-horizon lot-sizing when setup times are sequence-dependent*. International Journal of Production Research 38(10): 2287-2308.
- Cook W.J., Cunningham W.H., Pulleyblank W. R. & Schrijver A. (1998) *Combinatorial Optimization*. Wiley Interscience.
- Dowland K.A. (1993) *Simulated annealing*. In: C. R. Reeves (ed.), Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific, London, chapter 2, 20-69.
- Drexl A. & Haase K. (1995) *Proportional lotsizing and scheduling*. International Journal of Production Economics 40: 73-87.
- Drexl A. & Kimms A. (1997) *Lot sizing and scheduling-survey and extensions*. European Journal of Operational Research 99: 221-235.
- Eglese R.W. (1990) *Simulated annealing: a tool for operational research*. European Journal of Operational Research 46: 271-281.
- Frieze A.M. & Dyer M.E. (1990) *On a patching algorithm for the random asymmetric travelling salesman problem*. Mathematical Programming 46: 361-378.
- Glover F. & Laguna M. (1993) *Tabu search*. In: C. R. Reeves (ed.), Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific, London, chapter 3, 70-150.
- Karp R.M. (1979) *A patching algorithm for the nonsymmetric traveling-salesman problem*. SIAM Journal on Computing 8(4): 561-573.
- Karp R.M. & Steele J.M. (1985) *Probabilistic analysis of heuristics*. In: E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan and D.B. Shmoys (eds), The Traveling Salesman Problem-A Guided Tour of Combinatorial Optimization, Wiley, Chichester.
- Lin S. & Kernighan B.W. (1973) *An effective heuristic algorithm for the traveling salesman problem*. Operations Research 21: 498-516.

FERNANDO LOPES¹
NUNO MAMEDE²
A. Q. NOVAIS¹
HELDER COELHO⁴

Negotiation in a multi-agent supply chain system

¹INETI

²IST

⁴Faculdade de Ciências
Lisboa, Portugal

Abstract — The integration of isolated supply chain functions into a global system and the coordination of multiple functions across the system are two open problems. In this paper we address these problems by organizing the supply chain as a collection of autonomous agents that are able to coordinate their activities through negotiation. The agents generate plans of action towards the achievement of their goals and, over time, conflicts inevitably occur among them. Conflict resolution is crucial for achieving effective multi-agent coordination. Negotiation is the predominant process for resolving conflicts. This paper presents a prenegotiation model, a generic negotiation mechanism, and a set of negotiation tactics.

Keywords — autonomous agents, conflict of interests, prenegotiation planning, multi-agent negotiation, supply chain management.

Introduction

Autonomous agents are being increasingly used in a wide range of applications [Klusch 1999]. The agents have a high degree of control over their internal state and behavior—they can decide for themselves which goals to adopt, which actions to perform in order to achieve these goals, and when to perform these actions. Furthermore, the agents are situated in complex environments over which they have only partial visibility and influence [Jennings 2000].

Most applications involve or require multiple agents and, over time, conflicts inevitably occur among them, just as they do in human societies. Conflicts are indeed a pervasive aspect of human societies. They are not necessarily bad or good, but they are inevitable [Lewicki et al. 1999].

Negotiation is the predominant process for solving conflicts. This paper presents: (i) a prenegotiation model defining the main activities that every agent must attend to before starting to negotiate, (ii) a negotiation mechanism that handles multi-party, multi-issue and single or repeated rounds, and (iii) a set of negotiations tactics that express the initial attitude of every agent and generate counterproposals either by making or not making concessions.

This paper builds on our previous work [Lopes et al. 1999; Lopes et al. 2000; Lopes et al. 2001a; Lopes et al. 2001b]. More specifically, this paper continues the description of both the negotiation mechanism and the negotiation tactics. Also, we introduce the type of application domains we are interested in, by describing a simplified multi-agent supply chain system.

Our work follows an experimental line. The form of the mechanism, the negotiation tactics, and the assumptions they make, have been guided by our experiences in developing autonomous negotiating agents for the domain of supply chain management. The agents are currently being implemented in Prolog.

The remainder of this paper is organized as follows. Section 2 describes a simplified multi-agent supply chain system. Section 3 presents the components of the mental state and the planning mechanism of every autonomous agent. The concepts presented in this section form a basis for the development of autonomous negotiating agents. Section 4 presents formally the concept of conflict and describes axioms for conflict detection. Section 5 presents a formal prenegotiation model. Section 6 details the negotiation mechanism. Section 7 presents a set of negotiation tactics. Finally, related work and concluding remarks are presented in sections 8 and 9 respectively.

Multi-agent supply chain system

A *supply chain* is a network of facilities that performs the functions of procurement of raw materials from suppliers, transformation of these materials into intermediate goods and finished products, and the distribution of these products to customers [Ganeshan & Harrison 1995]. The *supply chain functions* range from the ordering and receipt of raw materials, to the distribution and delivery of final products, via the scheduling, production, warehousing, and inventory of intermediate goods and final products.

The *integration* of the multiple supply chain functions has received a great deal of attention in the recent years. However, most work addresses only single functions, such as scheduling or production. To date there exist little work that addresses the problem of integrating such isolated functions into a global supply chain.

The *coordination* of the supply chain functions has been another active area of research. Also, most research addresses the coordination of two or more supply chain functions, such as production-distribution and buyer-vendor coordination. Despite the importance of the results obtained, the coordination of multiple supply chain functions is still an open problem [Vidal & Goetschalckx 1997].

We address the integration and coordination problems in this paper by organizing the supply chain as a collection of autonomous agents that are able to coordinate their activities through negotiation.

System Architecture

The architecture of a simplified multi-agent supply chain system is shown in Fig. 1. The system is composed by a set of autonomous agents, each responsible for performing one or more supply chain functions [Fox et al. 1993]. We are currently working on the following agents: logistics agent, scheduler, resource management agent, dispatcher, a number of suppliers, and a number of customers. A brief description of each agent follows.

The *logistics* agent manages the movement of raw materials from the suppliers, the manufacturing of intermediate goods and final products by the enterprise, and

the distribution of the products to the customers. He receives customer orders, deviations in schedules which affects customer orders, and resource demands. He originates production requirements and supplier requests. He also notices the acquisition of resources.

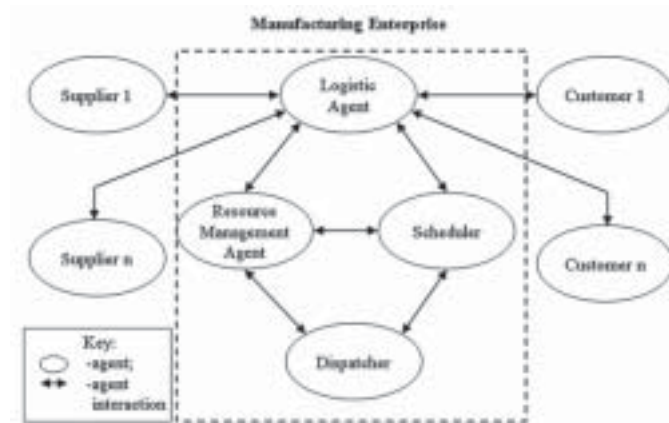


Fig. 1 – Simplified multi-agent supply chain system.

The *scheduler* is responsible for scheduling and rescheduling activities in the manufacturing enterprise. He receives production requests from the logistics agent, resource problems from the resource agent, and deviations of the current schedule from the dispatcher. He originates detailed schedules and sends them to the dispatcher and to the resource management agent. He also communicates the deviations of the current schedule to the logistics agent.

The *resource management* agent is responsible for managing dynamically the availability of resources in order to execute the scheduled activities. He receives the schedule from the scheduler and the consumption of resources from the dispatcher. He also receives information about the acquisition of resources from the logistics agent. He estimates resource demands and identifies resource problems. He transmits resource availability to the dispatcher.

The *dispatcher* is responsible for executing the scheduled activities. This agent controls the real time functions of the factory floor. He receives the schedule and the availability of resources. He notices deviations of the current schedule and the consumption of resources.

The *suppliers* sell raw materials and the *customers* buy finished goods. The suppliers receive orders from the logistics agent and transmit their own alternative orders. The customers send orders to the logistics agent and receive alternative orders.

Multi-Agent Negotiation

The individual agents of the supply chain system must work in a tightly coordinated manner in order to effectively and efficiently achieve their goals.

Coordination is achieved through negotiation between one or more suppliers and the logistics agent, the agents within the manufacturing enterprise, and one or more customers and the logistics agent.

Negotiation between agents in the simplified multi-agent supply chain system and, we believe, a wide range of similar systems, exhibit the following characteristics:

1. *two or more parties*-negotiation may involve two parties (e.g., the logistics agent and a customer) or many parties (e.g., the logistics agent and the scheduler, the resource management agent, etc). Multilateral negotiation consists essentially of a set of mutually influencing bilateral negotiations (e.g., the logistics agent negotiates individually with the scheduler, or with the resource management agent, etc);

2. *multiple issues* – negotiation ranges over a number of interrelated issues (e.g., price, quantity, quality, date, etc);

3. *repeated rounds (encounters)*—more than one bargaining session may occur before reaching an agreement. So, the atmosphere at the end of one session can influence the atmosphere at the next session;

4. *cooperative or non-cooperative negotiation behavior*—negotiation may occur between agents within the same organization (e.g., between the logistics agent and the scheduler) or between inter-organizational agents (e.g., between the logistics agent and a customer). In the former case, negotiation is cooperative in nature. In the latter case, negotiation is purely competitive;

5. *time restrictions*—time is an important factor. The time needed to reach an agreement must be reasonable. Also, the mutually accepted due dates are often important.

Autonomous agents

Let $Agents = \{ag_1, ag_2, \dots\}$ be a set of autonomous agents. This section discusses a single agent $ag_i \in Agents$ from an internal point of view, in terms of his mental state and planning process (see [Lopes et al. 1999; Lopes et al. 2000] for an in-depth discussion).

The agent ag_i has a set $B_i = \{b_{i1}, b_{i2}, \dots\}$ of *beliefs* representing facts about the world and the agent himself. We assume that beliefs are internally consistent, *i.e.*, individual beliefs do not conflict with one another. In addition, we assume that beliefs persist by default over time and are continuously updated to reflect changes in the world.

The agent ag_i has a set $G_i = \{g_{i1}, g_{i2}, \dots\}$ of *goals* representing world states to be achieved. We consider only achievement goals. An achievement goal, denoted by g_{ik} , states that ag_i wants to achieve a world state where g_{ik} holds. We assume that achievement goals are internally consistent.

The agent ag_i has a *library* $PL_i = \{pt_{i11}, pt_{i12}, \dots\}$ of *plan templates* representing simple procedures for achieving goals. A *plan template* $pt_{ikl} \in PL_i$ is a 7-tuple:

$$pt_{ikl} = \langle header_{ikl}, type_{ikl}, preconds_{ikl}, body_{ikl}, constrs_{ikl}, effects_{ikl} \rangle$$

The header is a 2-tuple: $header_{ikl} = \langle pname_{ikl}, pvars_{ikl} \rangle$, where $pname_{ikl}$ is the name of pt_{ikl} and $pvars_{ikl}$ is a set of variables (parameters of pt_{ikl}). In most cases, the header is simply the description of a goal g_{ik} for which pt_{ikl} is a recipe. The $type_{ikl}$ is the type

of pt_{ikl} (composite or primitive). $Preconds_{ikl}$ is a list of conditions that must hold before pt_{ikl} can actually be applied. The $body_{ikl}$ is either a list of subgoals whose achievement constitutes the achievement of g_{ik} or a list of primitive actions (*i.e.*, actions directly executable by ag_i) whose performance constitutes the achievement of g_{ik} . $Constrs_{ikl}$ is a list of constraints (e.g., to impose a temporal order on the members of the body). $Effects_{ikl}$ is a list of statements that hold after pt_{ikl} has been successfully executed.

The library PL_i has composite and primitive plan templates. A *composite plan template* $pt_{ikm} \in PL_i$ is a recipe specifying the decomposition of a goal g_{ik} into a set of subgoals. A *primitive plan template* $pt_{ikn} \in PL_i$ is a recipe specifying a primitive action or a sequence of primitive actions that can achieve a goal g_{ik} .

The agent ag_i is able to generate complex plans from the simpler plan templates stored in the library. A *plan* p_{ik} for achieving a goal $g_{ik} \in G_i$ is a 3-tuple:

$$p_{ik} = \langle PT_{ik}, \leq_h, \leq_t \rangle$$

where $PT_{ik} \subseteq PL_i$ is a list of instantiated plan templates (*i.e.*, plan templates where some or all of the parameters have been instantiated), \leq_h is a binary relation establishing a hierarchy on PT_{ik} ($pt_{ik1} \leq_h pt_{ik2}$, $pt_{ik1}, pt_{ik2} \in PT_{ik}$, means that pt_{ik2} is an immediate successor of pt_{ik1} , *i.e.*, a successor for which no intermediate plan templates are permitted), and \leq_t is another binary relation establishing a temporal order on PT_{ik} ($pt_{ik1} \leq_t pt_{ik2}$ means that pt_{ik1} must be applied before pt_{ik2}).

The plan p_{ik} is represented as a hierarchical and temporally constrained And-tree denoted by $Pstruct_{ik}$. A tree, rather than a linear ordering, is necessary because p_{ik} has both a hierarchical and a temporal dimension. The nodes of the tree are instantiated plan templates. Arcs form a hierarchy between pairs of nodes. Also, arcs represent ordering constraints.

At any instant of time, the agent ag_i has a number of plans for execution. These plans are the plans currently *adopted* by ag_i and are stored in the *intention structure* IS_i :

$$IS_i = [p_{i1}, p_{i2}, \dots, p_{ik}, \dots]$$

where, as stated above, each adopted plan $p_{ik} \in IS_i$ is defined as a 3-tuple: $p_{ik} = \langle PT_{ik}, \leq_h, \leq_t \rangle$. For each plan template $pt_{ikl} \in PT_{ik}$, the header of pt_{ikl} is referred as the description of *intention* int_{ik1} formulated by ag_i . Therefore, an intention is a goal not yet achieved and considered achievable – a goal restricted to the existence of a plan for achieving it. We assume that intentions are internally consistent and consistent with the beliefs.

The agent ag_i has information about the other agents in *Agents*. The information is stored in the *social description* SD_i :

$$SD_i = \{SD_i(ag_1), SD_i(ag_2), \dots, SD_i(ag_n)\}$$

where each entry $SD_i(ag_j) = \langle B_i(ag_j), G_i(ag_j), I_i(ag_j) \rangle$ contains the beliefs, goals and intentions that ag_i believes ag_j has.

Conflict of interests

This section defines formally the concept of conflict of interests and describes axioms for conflict detection (see [Lopes et al. 2000a] for an in-depth discussion).

Potential Conflict of Interests

Let $Ag = \{ag_1, \dots, ag_n\}$, $Ag \subseteq Agents$, be a set of autonomous agents. Let $ag_i \in Ag$ be an agent with intention structure $IS_i = [p_{i1}, \dots, p_{ik}, \dots]$ and social description $SD_i = \{SD_i(ag_1), \dots, SD_i(ag_n)\}$. Let p_{ik} be a plan of ag_i including intention int_{ikm} . Let $A = Ag - \{ag_i\}$ and $PI = \{int_{i11}(ag_1), \dots, int_{inn}(ag_n)\}$ be a set of *possible intentions* of the agents in A , *i.e.*, intentions that ag_i believes these agents have formulated.

Let the intentions in $PI \cup \{int_{ikm}\}$ represent commitments to achieve exclusive world states. In this situation, the intentions are called *incompatible* and represented by $Incomp(int_{ikm}, int_{i11}(ag_1), \dots, int_{inn}(ag_n))$, emphasizing the fact that they cannot be executed together.

A *potential conflict of interests* from the perspective of ag_i and with respect to plan p_{ik} (intention int_{ikm}) is defined formally as follows:

$$PotConf_{ik} = \exists int_{ikm} \in IS_i \wedge \exists int_{i11}(ag_1) \in SD_i(ag_1) \wedge \dots \wedge \exists int_{inn}(ag_n) \in SD_i(ag_n) \wedge Incomp(int_{ikm}, int_{i11}(ag_1), \dots, int_{inn}(ag_n))$$

Potential Conflict Detection

The agents in Ag check regularly their adopted plans in order to detect any potential conflict of interests. Conflict detection is done individually by each agent using pre-specified axioms $Ax_i = \{ax_{i1}, ax_{i2}, \dots\}$. Every axiom $ax_{ik} \in Ax_i$ has the general form:

$$int_{ikm} \& int_{i11}(ag_1) \& \dots \& int_{inn}(ag_n) \& conds \rightarrow false$$

where *conds* is a list of conditions, *false* is a 0-ary predicate symbol, $\&$ is the conjunction operator, and \rightarrow the implication operator.

True Conflict of Interests

Potential conflicts of interests are detected using information (possible intentions) that may be incorrect and have to be validated. Real conflicts of interests are then validated potential conflicts of interests.

Let $P_{Ag} = \{p_{11}, \dots, p_{ik}, \dots, p_{nn}\}$ be a set of plans of the agents in Ag including intentions $I_{Ag} = \{int_{111}, \dots, int_{ikm}, \dots, int_{nnn}\}$, respectively. Let $(IS_1, \dots, IS_i, \dots, IS_n)$ be the intention structures of the agents in Ag . Formally, a *real conflict of interests* amongst agents in Ag with respect to intentions in I_{Ag} is defined as follows:

$$Conf_{Ag} = \exists int_{111} \in IS_1 \wedge \dots \wedge \exists int_{ikm} \in IS_i \wedge \dots \wedge \exists int_{nnn} \in IS_n \wedge Incomp(int_{111}, \dots, int_{ikm}, \dots, int_{nnn})$$

Planning and preparing for negotiation

This section presents a brief description of the main activities that each agent $ag_i \in Ag$ must attend to in order to plan and prepare for negotiation (see our earlier work for an in-depth discussion [Lopes et al. 2001a]).

Negotiation Problem Structure Generation

Conflicts raise negotiation problems. Let B_i and G_i be the sets of beliefs and goals of ag_i , respectively. Let $p_{ik} \in P_{Ag}$ be a plan of ag_i for achieving goal $g_{ik} \in G_i$. Let $int_{ikm} \in I_{Ag}$ be an intention of p_{ik} . Let $I_A = I_{Ag} - \{int_{ikm}\}$. A *negotiation problem* from the perspective of ag_i is a 6-tuple:

$$NP_{ik} = \langle ag_i, B_i, g_{ik}, int_{ikm}, A, I_A \rangle.$$

The problem NP_{ik} has a *structure* $NPstruct_{ik}$ consisting of a hierarchical And-Or tree. The nodes of the tree are plan templates. The header of the root node describes the goal g_{ik} (called *negotiation goal*). Formally, $NPstruct_{ik}$ is a 4-tuple:

$$NPstruct_{ik} = \langle NPT_{ik}, \leq_b, \leq_t, \leq_a \rangle$$

where $NPT_{ik} \subseteq PL_i$ is a list of plan templates, \leq_b and \leq_t have the meaning just specified, and \leq_a is a binary relation establishing alternatives among the plan templates in NPT_{ik} . The structure $NPstruct_{ik}$ defines all the possible solutions of NP_{ik} currently known by ag_i . A *possible solution* is a plan that can achieve g_{ik} .

Issue Identification and Prioritization

The negotiation issues of ag_i are obtained from the leaves of $NPstruct_{ik}$. Let $L_{ik} = [pt_{ika}, pt_{ikb}, \dots]$ be the collection of plan templates constituting the leaves of $NPstruct_{ik}$. The header ($pname_{ikl}$ and $pvars_{ikl}$) of every plan template $pt_{ikl} \in L_{ik}$ is called a fact and denoted by f_{ikl} . Formally, a *fact* f_{ikl} is a 3-tuple: $f_{ikl} = \langle is_{ikl}, v[is_{ikl}], r_{ikl} \rangle$, where is_{ikl} is a *negotiation issue* (corresponding to $pname_{ikl}$), $v[is_{ikl}]$ is a value of is_{ikl} (corresponding to an element of $pvars_{ikl}$), and r_{ikl} is a list of arguments (corresponding to the remaining elements of $pvars_{ikl}$). Typically, r_{ikl} is an empty list (e.g., $fact = \langle price, 50 \rangle$).

Let $F_{ik} = \{f_{ika}, \dots, f_{ikz}\}$ be the set of facts of $NPstruct_{ik}$. The *negotiating agenda* of ag_i is the set of issues $I_{ik} = \{is_{ika}, \dots, is_{ikz}\}$ associated with the facts in F_{ik} (for clarity, we consider that every fact in F_{ik} defines a different issue). The interval of legal values for each issue $is_{ikl} \in I_{ik}$ is represented by

$$D_{ikl} = [min_{ikl}, max_{ikl}].$$

For each issue $is_{ikl} \in I_{ik}$, let w_{ikl} be a real number called *importance weight* that represents its relative importance. Let $W_{ik} = \{w_{ika}, \dots, w_{ikz}\}$ be the set of importance weights of the issues in I_{ik} . The importance weights are normalized, i.e., $\sum_{j=a}^z w_{ikj} = 1$. The *priority* of the issues in I_{ik} is just defined as their relative importance.

Limits and Aspirations Formulation

A *limit* or *reservation value* is a bargainer's ultimate fallback position, the level of benefit beyond which he is unwilling to concede. *Aspiration* is the benefit sought at any particular time. Limit tends to remain constant over time, whereas aspiration declines toward limit [Pruitt 1981].

Limits and aspirations are formulated for each issue at stake in negotiation. The *limit* for issue $is_{ikl} \in I_{ik}$ is represented by lim_{ikl} and the initial *aspiration* by asp^o_{ikl} with $lim_{ikl}, asp^o_{ikl} \in D_{ikl}$

Negotiation Constraints Definition

Negotiation constraints bound the acceptable values for the issues in I_{ik} . *Hard constraints* are linear boundary constraints that specify threshold values for the issues. They cannot be relaxed. *Soft constraints* are linear boundary constraints that specify minimum acceptable values for the issues. They can be relaxed, if necessary. They also can have different degrees of flexibility.

Constraints are defined for each issue $is_{ikl} \in I_{ik}$. The hard constraint hc_{ikl} for is_{ikl} has the form: $hc_{ikl} = (is_{ikl} \geq lim_{ikl}, flex=0)$, where $flex=0$ represents null flexibility (inflexibility). The soft constraint sc_{ikl} for is_{ikl} has the similar form: $sc_{ikl} = (is_{ikl} \geq asp^o_{ikl}, flex=n)$, where $flex=n$, $n \in N$, represents the degree of flexibility of sc_{ikl} .

Negotiation Strategy Selection

The agent ag_i has a library $SL_i = \{str_{i1}, str_{i2}, \dots\}$ of negotiation strategies and a library $TL_i = \{tact_{i1}, tact_{i2}, \dots\}$ of negotiation tactics. *Negotiation strategies* are functions that define the negotiation tactics to be used throughout the negotiation process. *Negotiation tactics* are functions that define the actions or moves to be made at each point of the negotiation process (see section 7).

Strategy selection is an important task and must be carefully planned [Lewicki et al. 1999; Pruitt 1981; Raiffa 1982]. The strategy most suitable for a particular negotiation situation often depends on the situation itself and cannot be specified in advance. As a result, strategy selection is a difficult task. In this paper, we just assume that ag_i selects a strategy $str_{ik} \in SL_i$ that he considers appropriate accordingly to his experience.

The negotiation mechanism

Examination of the literature in the fields of social psychology (e.g., [Pruitt & Rubin 1986; Pruitt 1981]), management science (e.g., [Lewicki et al. 1999; Lewicki & Litterer 1985]), economy and game theory (e.g., [Osborne & Rubinstein 1990; Raiffa 1982]) motivated the development of a generic negotiation mechanism that handles multi-party, multi-issue, and single or repeated rounds.

The mechanism supports the following primary characteristics of negotiation:

- (i) iterative exchange of proposals and counterproposals;
- (ii) communication of negotiation information;
- (iii) dynamic relaxation of negotiation constraints;
- (iv) dynamic discovery of new issues (learning in negotiation).

This section presents a domain-independent description of the negotiation mechanism.

Overview

Fig. 2 shows the negotiation mechanism from the perspective of an agent $ag_i \in Ag$ that generates and communicates a negotiation proposal. Let NP_{ik} represent ag_i 's perspective of the negotiation problem and $NPstruct_{ik}$ be the structure of NP_{ik} .

First, ag_i generates the *initial negotiation proposal set* $INPS_{ik} = \{prop_{ik1}, prop_{ik2}, \dots\}$, i.e., the set of negotiation proposals satisfying the requirements imposed by $NPstruct_{ik}$. Broadly speaking, a *negotiation proposal* $prop_{ikm} \in INPS_{ik}$ is a set of facts (see subsection 6.2). Next, ag_i determines the *initial acceptable proposal set* $IAPS_{ik}$, $IAPS_{ik} \subseteq INPS_{ik}$, i.e., the set of acceptable proposals. An *acceptable proposal* is a negotiation proposal that satisfies both the hard and soft negotiation constraints (see subsection 6.3). Next, ag_i evaluates the acceptable proposals in $IAPS_{ik}$ using an additive scoring function, and selects a particular proposal $prop_{ikm}$ accordingly to his negotiation strategy str_{ik} (see subsection 6.4).

Following this, ag_i communicates the proposal $prop_{ikm}$ to all the agents in A . Each agent $ag_j \in A$ then evaluates $prop_{ikm}$ and either: (i) accepts $prop_{ikm}$, (ii) breaks off negotiation, (iii) rejects $prop_{ikm}$ without making a critique, or (iv) rejects $prop_{ikm}$ and makes a critique. Broadly speaking, a *critique* is a statement of aspirations, priorities of the issues, etc. The tasks performed by each agent ag_j are not shown in Fig. 2.

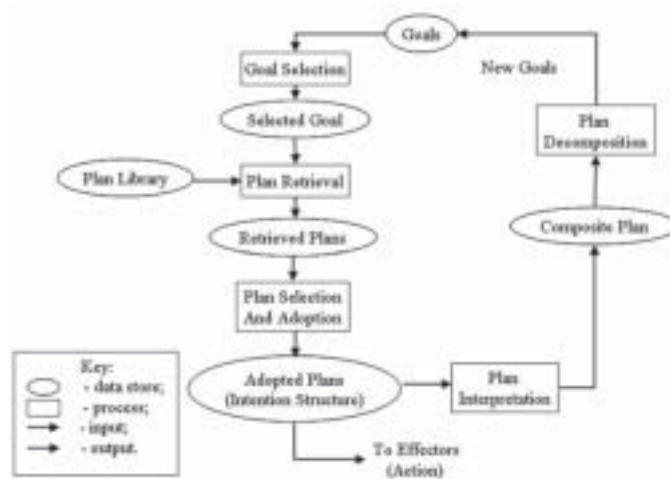


Fig. 2 – Generic negotiation mechanism (perspective of every agent that communicates a proposal).

Next, ag_i processes the responses and checks whether a negotiation agreement was reached. Generally speaking, a *negotiation agreement* is a proposal accepted by

all the agents. So, if the proposal $prop_{ikm}$ is accepted by all agents in A , the negotiation ends. Otherwise, ag_i checks whether any of the agents in A decided to break off negotiation. If at least one agent broke off, the negotiation ends. If not, ag_i determines whether or not to break off negotiation unilaterally. If so, the negotiation ends. Otherwise, ag_i checks whether the negotiation deadline was reached. Again, if so, the negotiation ends.

If the deadline was not reached, ag_i may decide either: (i) to do nothing (inaction), or (ii) to prepare a new proposal $prop_{ikm+1}$. The preparation of $prop_{ikm+1}$ can be done either: (i) by modifying the rejected proposal $prop_{ikm}$ (see section 7), or (ii) by selecting a new proposal from $LAPS_{ik}$. The negotiation strategy str_{ik} of ag_i defines the particular method to use. The new proposal $prop_{ikm+1}$ is then communicated to all agents in A and the tasks just described are repeated.

The decision to do nothing closes one *round* of negotiation. Negotiation proceeds to a new round in which another agent $ag_j \in A$ generates and communicates a counterproposal. Broadly speaking, a *counterproposal* is a proposal made in response to a previous proposal. This is then repeated for all the agents in A .

It is worth pointing out that each agent ag_i in Ag may decide either: (i) to relax the soft constraints, or (ii) to restructure the negotiation problem. This can be done at each point of the negotiation process (fig. 2 shows only constraint relaxation and problem restructuring at the beginning of negotiation). Problem restructuring allows the dynamic addition and remotion of negotiation issues.

Negotiation Proposal Generation

Negotiation proposal generation is a process that takes $NPstruct_{ik}$ as input and generates the set $INPS_{ik}$ of negotiation proposals through an iterative procedure involving three main tasks: (i) problem interpretation, (ii) proposal preparation, and (iii) proposal addition.

Let g_{ik} be the negotiation goal of ag_i . Let $F_{ik} = \{f_{ika} \dots f_{ikz}\}$ be the set of facts of $NPstruct_{ik}$ and $I_{ik} = \{is_{ika} \dots is_{ikz}\}$ be the set of issues associated with the facts in F_{ik} .

Problem interpretation consists of searching $NPstruct_{ik}$ for any possible solution p_{ik} of NP_{ik} and selecting the primitive plan templates of p_{ik} . More specifically, the search starts at the root node of $NPstruct_{ik}$, proceeds towards its leaves, and involves the arbitrary choice of exactly one plan template at each Or node of $NPstruct_{ik}$.

Proposal preparation consists of determining a *negotiation proposal* $prop_{ikm} = \{f_{ika} \dots f_{ikp}\}$, $prop_{ikm} \subseteq F_{ik}$, i.e., a set of facts corresponding to the headers of the primitive plan templates in ppt_{ik} . It is worth noting that the preparation of a proposal $prop_{ikm}$ partitions the set F_{ik} of facts into: (i) subset $prop_{ikm}$, and (ii) subset $pcompl_{ikm} = \{f_{ikp+1} \dots f_{ikz}\}$, called *proposal complement* of $prop_{ikm}$, corresponding to the remaining facts of F_{ik} . The facts in $prop_{ikm}$ are fundamental for achieving the negotiation goal g_{ik} . They are the *inflexible facts* of negotiation, for proposal $prop_{ikm}$. The negotiation issues $Iprop_{ikm} = \{is_{ika} \dots is_{ikp}\}$ associated with these facts are the *inflexible issues*. On the other hand, the facts in $pcompl_{ikm}$ are not important for achieving g_{ik} . They are the *flexible facts* of negotiation, for proposal $prop_{ikm}$.

The issues $I_{compl;ikm} = \{is_{ikp+1}, \dots, is_{ikz}\}$ associated with these facts are the *flexible or bargaining issues*.

Proposal addition consists of adding the negotiation proposal $prop_{ikm}$ to the set $INPS_{ik}$.

Acceptable Proposal Preparation

Acceptable proposal preparation involves two main tasks: (i) feasible proposal formulation, and (ii) acceptable proposal determination. Let $prop_{ikm} = \{f_{ika}, \dots, f_{ikp}\}$ be a negotiation proposal. Let $Iprop_{ikm} = \{is_{ika}, \dots, is_{ikp}\}$ be the set of issues associated with facts in $prop_{ikm}$. Let $HCprop_{ikm} = \{hc_{ika}, \dots, hc_{ikp}\}$ and $SCprop_{ikm} = \{sc_{ika}, \dots, sc_{ikp}\}$ be the sets of hard and soft constraints for issues in $Iprop_{ikm}$ respectively.

Feasible proposal formulation consists of generating the set $IFPS_{ik}$, $IFPS_{ik} \subseteq INPS_{ik}$, of feasible proposals. A negotiation proposal $prop_{ikm} \in INPS_{ik}$ is *feasible* if the issues in $Iprop_{ikm}$ satisfy the set $HCprop_{ikm}$ of hard constraints.

Acceptable proposal determination consists of generating the set $IAPS_{ik}$, $IAPS_{ik} \subseteq IFPS_{ik}$, of acceptable proposals. A feasible proposal $prop_{ikm}$ is *acceptable* if the issues in $Iprop_{ikm}$ satisfy the set $SCprop_{ikm}$ of soft constraints.

Proposal Evaluation and Selection

Proposal evaluation consists of computing a score for each proposal in $IAPS_{ik}$ and ordering the proposals in a descending order of preference.

The score of each proposal $prop_{ikm}$ is computed using an *additive scoring function* [Raiffa 1982]. Let $W_{ik} = \{w_{ika}, \dots, w_{ikp}\}$ be the set of importance weights of the issues in $Iprop_{ikm}$. Let $C_{ikm} = (v[is_{ika}], \dots, v[is_{ikp}])$ be the values of the issues in $Iprop_{ikm}$ (C_{ikm} is called a *contract*). For each issue $is_{ikl} \in Iprop_{ikm}$ defined over the interval $D_{ikl} = [min_{ikl}, max_{ikl}]$, let V_{ikl} be a *component scoring function* that gives the score that ag_i assigns to a value $v[is_{ikl}] \in D_{ikl}$ of is_{ikl} . The score for contract C_{ikm} is given by a function V :

$$V(C_{ikm}) = \sum_{j=1}^p w_{ikj} V_{ikj}(v[is_{ikj}])$$

The proposal $prop_{ikm}$ is identified with contract C_{ikm} and both have the same score.

Proposal selection consists of selecting a particular proposal $prop_{ikm} \in IAPS_{ik}$. The negotiation strategy str_{ik} dictates a specific tactic $tact_{ik} \in TL_i$ to use. The tactic $tact_{ik}$ specifies a particular proposal $prop_{ikm}$.

Negotiation tactics

Negotiation tactics are functions that define the actions or moves to be made at each point of the negotiation process. This section describes a set of tactics from the perspective of each agent $ag_i \in Ag$ (see our earlier work for an in-depth discussion [Lopes et al. 2001b]).

Opening Negotiation Tactics

Opening negotiation tactics are functions that express the initial attitude of ag_i and specify the proposal to submit at the beginning of negotiation. In this paper, we consider the following three tactics:

1. *starting high*—expresses an aggressive opening attitude and specifies the proposal with the highest score;
2. *starting optimistic*—expresses an optimistic opening attitude and specifies a proposal with a score between the highest and the lowest;
3. *starting realistic*—expresses a realistic opening attitude and specifies the proposal with the lowest score.

Let $IAPS_{ik} = \{prop_{ik1}, prop_{ik2}, \dots\}$ be the set of acceptable proposals of ag_i ordered in a descending order of preference ($prop_{ik1}$ is the proposal with the highest score $V_{prop_{ik1}}$).

The tactic starting high is formalized by a function *starting_high* which takes $IAPS_{ik}$ as input and returns $prop_{ik1}$, i.e.,

$$starting_high(IAPS_{ik}) = prop_{ik1} \mid \forall prop_{ikj} \in IAPS_{ik}, V_{prop_{ik1}} \geq V_{prop_{ikj}}$$

The definition of the functions for the tactics starting optimistic and starting realistic is essentially identical to that of *starting_high* and is omitted.

Bargaining Issue Manipulation Tactic

The art of negotiation centers on the willingness to give up something in order to get something else in return. Successful negotiators often exaggerate the importance of what they give up and minimize the importance of what they get in return. Such exaggerations are often used in real world negotiations to extract concessions from the other parties. For instance, bargainers purposely add to the negotiation agenda issues that they do not really care about, in the hope that the other parties will feel strongly about one or more of these issues—strong enough to be willing to make compensating concessions [Raiffa 1982].

Bargaining issue manipulation is a tactic that allows ag_i to act strategically by using the flexible facts (bargaining issue with specific values) to extract concessions from the other parties. Let $prop_{ikm} = \{f_{ik\alpha} \dots f_{ik\beta}\}$ be a negotiation proposal submitted by ag_i and rejected. Let $pcompl_{ikm} = \{f_{ik\beta+1} \dots f_{ikz}\}$ be the complement of $prop_{ikm}$. *Bargaining issue manipulation* allows ag_i to improve $prop_{ikm}$ by adding a flexible fact $f_{ikx} \in pcompl_{ikm}$ to $prop_{ikm}$. More specifically, this tactic is formalized by a function *barg_issue_manip* which maps $prop_{ikm}$ and f_{ikx} into a new proposal $prop_{ikm+1}$ containing f_{ikx} i.e.,

$$barg_issue_manip(prop_{ikm}, f_{ikx}) = prop_{ikm+1} \mid prop_{ikm+1} = prop_{ikm} \cdot f_{ikx}$$

where \cdot stands for concatenation.

Bargaining issue manipulation is a *non-concession* tactic. Indeed, the rejected proposal $prop_{ikm}$ and the new proposal $prop_{ikm+1}$ have very similar scores (hence, ag_i does not make a concession).

Bargaining issue manipulation promotes an image of firmness in the eyes of the other parties and invites concession making from them. Also, this tactic guards against image loss and prevents position loss. *Image loss* is the development in the others parties' eyes of an impression that the bargainer lacks firmness (*i.e.*, is ready to make a substantial concession). Fear of image loss is sometimes called a concern about *face-saving*. *Position loss* is the abandonment of a desirable proposal [Pruitt 1981].

Concession Tactics

Concession tactics are functions that compute new values for each negotiation issue. The tactics model the concessions to be made on every issue at each point of the negotiation process.

Let I_{ik} be the set of negotiation issues. A *concession* on an issue $is_{ikj} \in I_{ik}$ is a change in the value of is_{ikj} that reduces the level of benefit sought. In this paper, we consider the following five tactics:

1. *stalemate*-models a *null* concession on is_{ikj} ;
2. *tough*-models a *small* concession on is_{ikj} ;
3. *moderate*-models a *moderate* concession on is_{ikj} ;
4. *soft*-models a *large* concession on is_{ikj} ;
5. *compromise*-models a *complete* concession on is_{ikj} .

Let $prop_{ikm}$ be a proposal submitted by ag_i and rejected. Let $v[is_{ikj}]_{old}$ be the value of is_{ikj} offered in $prop_{ikm}$. Let lim_{ikl} be the limit for is_{ikj} . Let V_{ikj} be the component scoring function of ag_i for is_{ikj} . Let $v[is_{ikj}]_{new}$ be the new value of is_{ikj} to be offered in a new proposal $prop_{ikm+1}$. The five tactics are formalized by the following expression:

$$v[is_{ikj}]_{new} = v[is_{ikj}]_{old} + (-1)^w F |lim_{ikj} - v[is_{ikj}]_{old}|$$

where $w=0$ if V_{ikj} is monotonically decreasing or $w=1$ if V_{ikj} is monotonically increasing, and $F \in [0, 1]$ is a factor.

The factor F can be simply a constant. The five tactics are then defined by considering different values for F . For instance, the stalemate tactic is defined by setting $F=0$, the tough tactic by $F \in]0, 0.5[$, the moderate tactic by setting $F=0.5$, the soft tactic by $F \in]0.5, 1[$, and the compromise tactic by $F=1$ (or $F = |v[is_{nkj}] - v[is_{ikj}]_{old}| / (lim_{ikj} - v[is_{ikj}]_{old})$), where $v[is_{nkj}]$ is the value proposed by other party ag_n to the issue is_{ikj} .

Alternatively, the factor F can vary throughout the negotiation and be a function of a single criteria [Faratin et al. 1998; Koperczak et al. 1992]. In this paper, we concentrate on a new criteria called *relative total concession*. Let $v[is_{ikj}]_0, v[is_{ikj}]_1, \dots, v[is_{ikj}]_{n-1}, v[is_{ikj}]_n$ be the values of is_{ikj} successively offered by ag_i , with $V_{ikj}(v[is_{ikj}]_{i-1}) \geq V_{ikj}(v[is_{ikj}]_i)$, $1 \leq i \leq n$. Let $C = |v[is_{ikj}]_{i-1} - v[is_{ikj}]_i|$ be a concession made by ag_i on is_{ikj} at a specific point in the negotiation. Let $C_{total} = |v[is_{ikj}]_0 - v[is_{ikj}]_n|$ be the total concession made by ag_i on is_{ikj} . The *relative total concession* is defined by $C_{total} / |lim_{ikj} - v[is_{ikj}]_0|$. We then distinguish the following function for modelling F :

$$F = 1 - \lambda \sum_{i,j} C_{total} | \lim_{i \rightarrow j} v_{[i,j]} | 0$$

where $\lambda \in R^+$.

Related work

Negotiation is a rich, multidisciplinary research area. As a result, we highlight in this section just the negotiation work most related to our own work.

Laasri et al. [1992] present a generic negotiation mechanism. The mechanism is rich, but assumes that agents are inherently cooperative.

Sycara [1991] presents a negotiation mechanism that supports problem restructuring and is based on persuasive argumentation. The mechanism can be employed by non-cooperative agents, but assumes the existence of a centralized mediator.

Faratin et al. [1998] present a multi-party, multi-issue negotiation mechanism. Again, the mechanism is rich, but no consideration was given to integrate it into a unified agent architecture.

Rosenschein and Zlotkin [1994] used game theory to investigate the properties of negotiation mechanisms. Their work has produced significant results, but assumes that agents have complete knowledge.

We are interested in negotiation among self-motivated agents. Our structure for representing negotiation problems allows the direct integration of planning and negotiation into a unified agent architecture. This structure is similar to decision trees [Goodwin & Wright 1991], and goal representation trees [Kersten et al. 1991], but there are important differences. Our approach does not require the quantitative measures typical of decision analysis. In addition, our approach is based on plan templates and plan expansion, and not on production rules and forward and backward chaining. Also, our formula for modeling negotiation tactics is similar to the formulae used by Faratin et al. [1998] and Koperczak et al. [1992]. Again, there are important differences. Our formula assures that the new value of an issue ranges between the reservation value and the previous value of the issue. In addition, our formula models important experimental conclusions about limit, demand, and concession. Finally, the relative total concession criteria is not used by other researchers.

Discussion and future work

This article has introduced a conflict definition, a prenegotiation model, a negotiation mechanism, and a set of negotiation tactics.

There are several features of our work that should be highlighted. First, the prenegotiation task of generating a structure for a negotiation problem acknowledges the role of conflict as a driving force for negotiation. Also, problem structure defines the set of negotiation issues. In addition, the structure of a negotiation problem represents a natural link between the individual and social behavior of agents. Second, the negotiation mechanism is generic and can be used in a wide range of domains.

Finally, the mechanism supports dynamic constraint relaxation and problem restructuring, ensuring a high degree of flexibility. Constraint relaxation and problem restructuring facilitates the remotion of deadlocks and increases the parties' willingness to a compromise.

Our aim for the future is: (i) to extend the negotiation mechanism to consider problem restructuring, and (ii) to validate experimentally a set of negotiation tactics and strategies.

References

- Faratin P., Sierra C. & Jennings N. (1998) *Negotiation Decision Functions for Autonomous Agents*. Journal of Robotics and Autonomous Systems, 24(3-4), 159-182.
- Fox M., Chionglo J., & Barbuceanu M. (1993) *The Integrated Supply Chain management System*. Internal Report, Department of Industrial Engineering, University of Toronto.
- Ganeshan R. & Harrison T. (1995) *An Introduction to Supply Chain Management*. Internal Report, Department of Management Science and Information, Penn State University, USA.
- Goodwin P. & Wright G. (1991) *Decision Analysis for Management Judgement*. John Wiley and Sons.
- Jennings N. (2000) *On Agent-Based Software Engineering*. Artificial Intelligence, 117, 277-296.
- Kersten G., Michalowski W., Szpakowicz S. & Koperczak Z. (1991) *Restructurable Representations of Negotiation*. Management Science, 37(10), 1269-1290.
- Klusch M. (1999) *Intelligent Information Agents*. Springer-Verlag: Berlin.
- Koperczak Z., Matwin S. & Szpakowicz S. (1992) *Modelling Negotiation Strategies with Two Interacting Expert Systems*. Control and Cybernetics, 21(1), 105-130.
- Laasri B., Laasri H., Lander S. & Lesser V. (1992) *A Generic Model for Intelligent Negotiation Agents*. Int. J. Intell. Cooperative Information Systems, 1(1), 291-318.
- Lewicki R., Saunders D. & Minton J. (1999) *Negotiation, Readings, Exercises, and Cases*. McGraw Hill-Irwin: Boston.
- Lewicki R. & Litterer J. (1985) *Negotiation*. Irwin, Homewood, Illinois.
- Lopes F., Mamede N., Coelho H. & Novais A.Q. (1999) *A Negotiation Model for Intentional Agents*. In: Multi-Agent Systems in Production, (P. Kopacek ed.), 211-216, Elsevier Science, Amsterdam.
- Lopes F., Mamede N., Novais A.Q. & Coelho H. (2000) *Towards a Generic Negotiation Model for Intentional Agents*. In: Agent-Based Information Systems, (A. Tjoa, R. Wagner and A. Al-Zobaidie eds), 433-439, IEEE Computer Society Press, CA.
- Lopes F., Mamede N., Novais A.Q. & Coelho H. (2001a) *Conflict Management and Negotiation Among Intentional Agents*. In: Agent-Based Simulation, (C. Urban ed.), 117-124, SCS-Europe.

- Lopes F., Mamede N., Novais A.Q. & Coelho H. (2001b) *Negotiation Tactics for Autonomous Agents*. In: Internet Robots, Systems and Applications, IEEE Computer Society Press (to appear).
- Osborne M. & Rubinstein A. (1990) *Bargaining and Markets*. Academic Press: San Diego, CA.
- Pruitt D. (1981) *Negotiation Behavior*. Academic Press: London.
- Pruitt D. & Rubin J. (1986) *Social Conflict: Escalation, Stalemate and Settlement*. McGraw-Hill: New York, USA.
- Raiffa H. (1982) *The Art and Science of Negotiation*. Harvard University Press: Cambridge.
- Rosenschein J. & Zlotkin G. (1994) *Rules of Encounter*. The MIT Press: Cambridge, USA.
- Sycara K. (1991) *Problem Restructuring in Negotiation*. Management Science, 37 (10), 1248-1268.
- Vidal C. & Goetschalckx M. (1997) *Strategic Production-Distribution Models: A Critical Review with Emphasis on Global Supply Chain Models*. European Journal of Operational Research, 98, 1-18.