Reducing myopic behavior in FMS control : a semi-heterarchical simulation-optimization approach

Gabriel Zambrano Rey

HAL Id: tel-01197082
https://tel.archives-ouvertes.fr/tel-01197082

Submitted on 11 Sep 2015

Thèse de doctorat
Pour obtenir le grade de Docteur de l'Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS

Spécialité Automatique et Génie Informatique

Présentée et soutenue par Gabriel, ZAMBRANO REY.

Le 03/07/2014, à Valenciennes

Ecole doctorale :
Sciences Pour l'Ingénieur (SPI)

Equipe de recherche, Laboratoire :
Laboratoire de Thermique, Ecoulement, Mécanique, Mise en Production (TEMPO)
Equipe : Production, Services, Information (PSI)

Réduction du Comportement Myope dans le contrôle des FMS : Une Approche Semi-Hétérarchique basée sur la Simulation-Optimisation

JURY

Rapporteurs
BENYOUCEF, Lyes. Professeur à Aix-Marseille Université. Marseille, France.
LEITÃO, Paulo. Professeur à l'Instituto Politécnico de Bragança. Bragança, Portugal.

Examinateurs
CARVALHO, Marco. Professeur à Florida Institute of Technology. Melbourne, États Unis.
MONTOYA TORRES, Jairo. Professeur à l' Universidad de La Sabana. Chia, Colombie.

Président du Jury
PIERREVAL, Henri. Professeur à l'Institut Français de Mécanique Avancée, Clermont-Ferrand, France.

Directeur de thèse
TRENTESAUX, Damien. Professeur à l'Université de Valenciennes et du Hainaut Cambrésis.

Co-encadrant
BERGER, Thierry. Maître de conférences, Université de Valenciennes et du Hainaut Cambrésis.

Thesis submitted for the degree of Doctor from Université de VALENCIENNES ET DU HAINAUT-CAMBRESIS

Specialty in Automation and Computer Science

Submitted and Defended by Gabriel, ZAMBRANO REY.

July 3rd 2014, Valenciennes, FRANCE

Doctoral School:
Sciences Pour l'Ingénieur (SPI)

Laboratory and Research Team:
Laboratoire de Thermique, Ecoulement, Mécanique, Mise en Production (TEMPO)
Team: Production, Services, Information (PSI)

Reducing Myopic Behavior in FMS Control: A Semi-Heterarchical Simulation-Optimization Approach

DEFENSE COMMITTEE

Reviewers
BENYOUCEF, Lyes. Professor at Aix-Marseille Université. Marseille, France.
LEITÃO, Paulo. Professor at l'Instituto Politécnico de Bragança. Bragança, Portugal.

Examiners
CARVALHO, Marco. Professor at Florida Institute of Technology. Melbourne, United States.
MONTOYA TORRES, Jairo. Professor at l' Universidad de La Sabana. Chia, Colombia.

President of the Defense Committee
PIERREVAL, Henri. Professor at Institut Français de Mécanique Avancée. Clermont-Ferrand, France.

Dissertation Advisor
TRENTESAUX, Damien. Professor at Université de Valenciennes et du Hainaut Cambrésis.

Dissertation Co-Advisor
BERGER, Thierry. Professor at Université de Valenciennes et du Hainaut Cambrésis.

Acknowledgements

I would like to dedicate these lines to acknowledge many people that made part of this wonderful experience. First and foremost, I would like to express my deepest gratitude to my Ph.D. advisor, Professor Damien Trentesaux. Thanks for giving me this opportunity, for your engagement in leading me and for your constant motivation and interest on the construction of my career. I truthfully appreciate our inspiring discussions and your advices. I also would like to thank Professors Vittal Prabhu and Thierry Berger for their eager participation in this work and for the nourishing discussions that allow me to have a broader view of my research subject. To my research colleagues Abdelghani Bekrar, Cyrille Pach, Nassima Aissani, and Thérèse Bonte, thanks a lot for your valuable participation in the publications we made together; it was very fulfilling working with you.

My sincere gratitude to Professors Benyoucef and Leitão for your thorough review of my dissertation manuscript. Your comments add a significant value to my work and career. As well, to Professors Carvalho, Montoya Torres and Pierreval, my gratitude for taking the time to come to Valenciennes and make part of the dissertation committee. Finally, to my other team colleagues thanks a lot for making these years a memorable time. Certainly, I had the priviledge to work in a wonderful environment, with people that I consider friends.

Agradecimientos

A Dios, quien pone en mi camino retos maravillosos y me acompaña en el día a día. A mi adorada esposa y amado hijo, este logro no hubiera sido posible sin ustedes; este logro es de los tres. Los dos son mi motor interior, mi fuente de energía y amor. A mis padres, gracias por su incansable y constante apoyo; por todo sus esfuerzos, por la motivación y sus oraciones. A mi familia y amigos, gracias por estar ahí, siempre.

Un profundo y sincero agradecimiento a mi país, que a través de COLCIENCIAS, aportó económicamente a la realización de este doctorado. Espero como Colombiano poder retribuir con mi trabajo al desarrollo sostenible y equitativo de Colombia. También, a la Pontificia Universidad Javeriana, gracias por la motivación hacia la constante búsqueda del conocimiento y por su compromiso, proporcionando los medios necesarios para la realización de esta tesis doctoral.

Foreword

After my Master graduation I started working at Javeriana University as a member of the Technology Center for Industrial Automation. Since then, I have been involved in research and teaching activities in the Department of Industrial Engineering. During my first research projects I could realized that I needed a broader scope on how to actually do research. Research in Colombia is becoming a key competitive factor for our economic and social development and thanks to scholarship programs, recently established, there is a real possibility to undertake doctoral studies abroad. The country is changing and I am convinced that, us as Colombians, we have to be part of that change by helping with innovative ideas. This was one of the main reasons that motivated me to look for a Ph.D. program that fulfils that need.

Several reasons made me consider coming to France for my Ph.D. First, during my high school studies in Colombia, I took a lot of interest in the French language and the French culture. Then, during my Master program I met several people, including various professors, who had made their studies in France. Their experiences and the way research is managed in France motivated me to apply for a Ph.D.. Also, politically and economically, both countries have always had close relationships and they will be strengthened by the new trade agreements with the European Union. Research, through different bi-lateral cooperation mechanisms, is then a link necessary to better take advantage of such agreements, allowing us to exchange experiences and work for sustainable economies.

Through these years in Valenciennes, I had the opportunity to not only get in touch with research dynamics in France, but also I have been able to establish a research network. Establishing and continuing collaboration with researchers that I met during these past years would be one of my main objectives to maintain a research dynamic and a broader vision in the context of my research areas. I am excited to go back to Colombia and get in touch with the current situation of my country, hoping to propose research projects with significant impact to the Colombian society.

List of Acronyms

ABS	Agent-based simulation
ACO	Ant-colony optimization
ADACOR	Adaptive holonic control architecture
AI	Artificial intelligence
AIP- PRIMECA	Atelier inter-établissement de productique - Pôle de ressources informatique pour la mécanique
AGV	Automated guided vehicles
AMRF	Automated manufacturing research facility
ANOVA	Analysis of variance
ATC	Arrival-time control
BMS	Bionic or biological manufacturing systems
CIM-OSA	Computer integrated manufacturing - Open system architecture
CLFS	Closed-loop feedback simulation
CNP	Contract-Net protocol
CP	Control problem
CSP	Control sub-problem
CTV	Completion time variance
DAI	Distributed artificial intelligence
DE	Decisional entity
DES	Discrete event simulation
DNA	Deoxyribonucleic acid
DR	Dispatching rule
EGP	Estimated global performance
ELP	Estimated local performance
ERP	Enterprise resource planning
FAL	Flexible assembly line
FAM	First available machine
FJSSP	Flexible job-shop scheduling problem
FHFMS	Fully heterarchical FMS
FMS	Flexible manufacturing system

GA	Genetic algorithm
GDE	Global decisional entity
GDL	Global decisional level
GDV	Global decision variable
GOF	Global objective functions
HFMS	Heterarchical-based FMS
HMS	Holonic manufacturing systems
HSD	Honest significant difference
JIT	Just in time
JSSP	Job-shop scheduling problem
LDE	Local decisional entity
LDL	Local decisional level
LDV	Local decision variable
LO	Local objective
LOF	Local objective functions
LP	Local performance
LS	Local status
MAS	Multi-agent systems
MBR	Myopic behavior reduction
MES	Manufacturing execution systems
MILP	Mixed-Integer Linear Program
mLDE	Machine local decisional entity
MRP	Manufacturing requirements planning
MRP-II	Manufacturing resource planning
MSD	Mean square deviation
MTO	Make-to-order
MTS	Make-to-stock
NP	Non polynomial
OA	Optimization agent
PAC	Production activity control
PD	Product-driven systems
pGDE	Production global decisional entity
PFA	Potential fields approach
PROSA	Product-Resource-Order-Staff architecture
PROSIS	Product-Resource-Order and isoarchic architecture

PSO	Particle swarm optimization
PSI	Production, services and information research team
RFID	Radio frequency identification
RSDMA	Resource scheduling dynamic mediator agent
S/O	Simulation/Optimization
SA	Simulated annealing
SbO	Simulation-based optimization
SD	System dynamics
SHFMS	Semi-heterarchical FMS
SM	Single machine
QLP	Quadratic linear program
TCP/IP	Transmission control protocol/Internet protocol
TD	Transportation device
TEMPO	Laboratory of thermics, fluid flow, mechanics and materials, forming, production
UML	Unified modeling language
vLDE	Transport local decisional entity

Table of Contents

List of Figures

List of Tables

Introduction

1. Background

In the second part of the last century, due to relative market stability and low product diversity, hierarchical manufacturing control architectures were widely accepted and deployed for manufacturing control, since long-term optimality was possible. Nevertheless, this scenario evolved towards more customer-designed products, higher product variety, shorter product life cycles, smaller lot sizes and shorter lead times, requiring more flexibility, reactivity and adaptability of manufacturing control. Flexible manufacturing systems (FMS) tried to respond to such challenges but their efficiency had been constrained by hierarchical control architectures. Therefore, heterarchical FMS control architectures emerged supported by leading-edge-technologies advocating for more decentralization of control decisions. Heterarchical architectures also provide other benefits in terms of low-complex control architecture, local reactivity to internal and external perturbations, and adaptability to production and market changes among others. Unfortunately and despite the benefits claimed by heterarchical FMS control, these architectures have been rarely adopted in industry because of two main reasons. First, it is not easy to predict outcomes due to local behavior and the absence of centralized control. Second, as a consequence of local behavior, it is difficult to guarantee a minimum level of operational performance, yet production objectives can be compromised by critical issues such as deadlocks. These two barriers are mainly due to the myopic behavior showed by local decisional entities in heterarchical FMS control. This myopic view comes from the high degree of autonomy, local goal orientation and locally contained information experienced by decisional entities.

2. Motivation and objectives

Given the opportunities and the benefits that heterarchical FMS control architectures can offer to the development and implementation of agile, highly-adapted and competitive manufacturing systems; the question is how can these architectures provide better global performance without losing the benefits obtained with decentralization? Specifically, since myopic behavior is one of the causes of the lack of global performance guarantee, the main questions are can myopic behavior be reduced or controlled? If so, can this reduction or control be achieved by structural and/or operational features? And what is the impact of those structural changes and additional features to the underlying heterarchical architecture?

So far, heterarchical FMS control architectures that have been proposed by researchers focused on enhancing global performance but not explicitly tackling myopic behavior. As a result, they proved to be more complex, putting at stake important characteristics of heterarchy such as reactivity, fault-tolerance and adaptability, possibly yielding more rigid hierarchies than heterarchy. The scope of this dissertation can be summarized as follows:

Myopic behavior can be reduced by means of simulation-based optimization techniques within a semi-heterarchical architecture, in which different decision-making roles and interaction modes between decisional levels can coexist, aiming for a balance between global performance and reactivity.

This assumption is based on the following foundations:

✓ Semi-heterarchical architectures can merge the benefits of hierarchy in terms of global performance and heterarchy in terms of reactivity, adaptability and fault tolerance

✓ Simulation-based optimization techniques can be used for decision making support within a hierarchical decisional level with the possibility of different decision-making roles, thus different ways of reducing myopic behavior may exist.

✓ Within a semi-heterarchical approach, different types of dynamic behaviors between decisional levels may exist, therefore myopic behavior can be reduced through various instances of the architecture

In this dissertation, the proposed semi-heterarchical architecture is composed of two decisional levels, global and local. The global level aims at explicitly reducing myopic behavior of the local level while it also ensures certain global performance. The proposed configuration for the global level is based on, looking for a balance between myopic behavior reduction (hence performance improvement) and reactivity to disruptions caused by market fluctuations and the stochastic nature of manufacturing processes. Operationally, myopic behavior can be reduced by different decision-making roles adopted by the simulation-based optimization techniques and different interaction modes between the two decisional levels. The following specific requirements were also sought:

✓ The methodological approach should be focused on dealing with myopic behavior from a granular perspective to decouple myopic behavior into several myopic control decisions that can be dealt individually.

✓ The configuration of the global decisional entity should be modular and adaptive to be able to reduce myopic decisions with the same granular perspective. Adaptability should be accomplished on the basis of internal and external perturbations.

✓ Such modularity should also be present in the interaction between global and local decisional entities so different levels of local autonomy can be achieved.

✓ The proposed semi-heterarchical architecture must be generic so it is capable to support different types of local control approaches and/or highly reactive rules at the local level.

✓ The proposed approach must be evaluated on benchmark problems.

3. Outline of the thesis

This dissertation is organized in five chapters as follows:

Chapter 1. It presents important definitions of manufacturing control, flexible manufacturing systems, the existing decision-making algorithms that can be used in heterarchical-based FMS control, and the evolution of control architectures. In addition, myopic behavior is defined and typified in regards to other domains and the few works that treat this issue in the manufacturing context.

Chapter 2. This chapter focuses on reviewing the state-of-the-art in myopic behavior reduction. Several approaches were reviewed and classified according to the type of control architecture, fully heterarchical or semi-heterarchical; and the technique used to reduce myopic behavior, optimization, simulation or simulation-based optimization. In order to better understand this literature review, a general framework for reducing myopic behavior, issued from the analysis of the literature, is proposed to the reader.

Chapter 3. Taking as reference the literature review and the proposed general framework, a semi-heterarchical simulation-based optimization approach to reduce myopic behavior in FMS control is described in this chapter. At first, the general features of the proposed approach are detailed and then the proposed semi-heterarchical FMS (SHFMS) control architecture is described. Since the core of the SHFMS is the decisional entity, the internal structure of global and local decisional entities is detailed as well as the possible hierarchical interaction modes. The control strategy under normal and abnormal conditions is also explained as well as a procedure to generate an instance of the proposed approach.

Chapter 4. This chapter describes an instance of the approach proposed in the precedent chapter. To this end, the FMS control problem, its parameters, assumptions and constraints taken into account are described first. Such information is afterwards used to configure the semi-heterarchical architecture by defining global and local decisional entities, as well as their interaction modes and control strategy.

Chapter 5. With all the constituent components being described in the previous chapter, herein a prototype implementation of the proposed approach into an assembly cell is described. At first, the experimental data is detailed and then the implementation of each decisional entity is described. The evaluation of the proposed approach was executed on the basis of a simulation study and then with hardware-in-the-loop experimentations on the assembly cell. The simulation study was carried out for static and dynamic scenarios taking into consideration two objective functions (single-objective problems) and three different configurations of the global decisional entity. Hardware-in-the-loop experimentations took some problem instances under normal and abnormal conditions to be executed at the assembly cell.

Some conclusions and future work are offered at the end of this dissertation.

Chapter I. Myopic Behavior in Flexible Manufacturing Systems Control

1. Introduction

The objective of this chapter is to define and position myopic behavior as a key research topic in manufacturing control. To this end, some important concepts about manufacturing systems and control are defined in Section I-2. The chapter focuses on flexible manufacturing systems (FMSs) presenting among others some important types of flexibilities (Section I-3). Two dimensions of FMS control are then detailed: the decision-making algorithms and the control architectures, respectively in Sections I-4 and I-5. This is followed by a discussion on current challenges, requirements and issues related to FMS control (Section I-6). Being myopic behavior one of those issues and the topic of this work, myopic behavior in manufacturing control is defined and typified in Section I-7.

2. Manufacturing systems

Manufacturing is the transformation process of raw materials into finished products demanded by the market. In a manufacturing system additional inputs are required, e.g., energy, equipments and facilities, labor, market information, and product design; and inevitably, non-desired outputs such as waste and scrap are also generated (Papadopoulos et al., 2009). An efficient and effective management of a manufacturing system is necessary to attain the company's objectives while optimizing inputs and minimizing non-desired outputs. Performance criteria to assess the efficiency and effectiveness of a manufacturing system can be defined by four factors: flexibility, quality, cost, time and re-configurability (Chryssolouris, 1992; Brennan et al., 2008).

From a systemic point of view, a manufacturing system is composed of three sub-systems: physical, informational and decisional (Blanc et al., 2008)blanc. The physical sub-system (i.e., manufacturing resources such as machines and material-handling systems) is in charge of the transformation of raw materials, labor work and energy into finished products; and other functions such as goods transport and storage. The informational sub-system gathers all the information required to monitor internal and external variables necessary for controlling the system. The informational sub-system also supplies the decisional sub-system

with the required data so the latter can control the entire manufacturing system to meet the objectives (Blanc et al., 2008).

As depicted in Figure I-1, the decisional sub-system has been typically decomposed into three levels: strategic, tactical and operational depending on the long, medium or short term horizon of control decisions executed at each level (Caramia and Dell'Olmo, 2006; Gudehus and Kotzab, 2012).
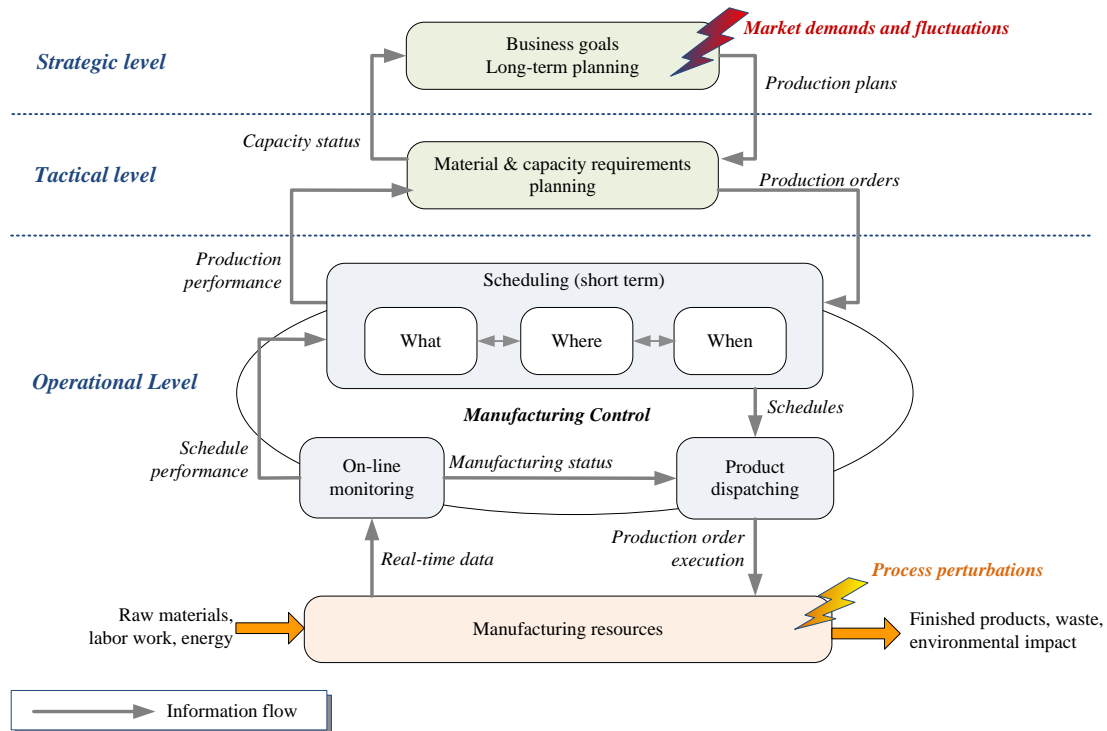
Figure I-1: Typical decisional levels in manufacturing systems (adapted from (Caramia and Dell'Olmo, 2006; Leitão, 2004))

The **strategic level** is the administrative level of the organization and is responsible for developing strategies to ensure the capability of the company to reach its goals and fulfill the requirements of the future. Activities at this level can be classified as business, design and production planning activities. Business activities are related to customer agreements and supplier contracting. Design activities establish which types of products need to be manufactured, types of manufacturing resources needed and the layout to support production (further information on shop floor layouts is available in Appendix A-I). Production planning then issues the production requirements (i.e., quantities, products, due dates, etc) in a master production schedule, using information based on forecasts and market analysis, and the capacity status returned by the tactical level.

At the **tactical level**, the master production schedule is used to calculate medium-term plans, transformed into production orders, either predicted orders or real customer orders

depending on the type of production environment, make-to-stock or make-to-order (production environments and their related performance criteria are detailed in Appendix A-II). On the basis of product composition (known as bill of materials), these plans issue the adequate material requirements and procurement orders to the suppliers. In addition, a rough-cut capacity plan is also generated to check if the manufacturing system can handle the imposed demand. Typically, Material Requirements Planning (MRP), Manufacturing Resource Planning (MRP-II) and/or Enterprise Resource Planning (ERP) carry out decision tasks at the strategic and tactical levels.

On the ***operational level***, the production orders generated at the tactical level are executed based on real time and current conditions of manufacturing resources. The operational level is also known as ***manufacturing control*** (Chryssolouris et al., 1991). The word *"control"* is used since this level is also in charge of monitoring the production order execution and making the necessary decisions to meet the imposed requirements defined at the upper levels, taking the appropriate corrective actions if any perturbation appears.

As shown in Figure I-1, manufacturing control is the main activity at the operational level. However, maintenance (Gao et al., 2006), inventory (Hnaien et al., 2010) and capacity (Cho and Prabhu, 2007) are other control problems, strongly related to manufacturing that are dealt at the operational level. Henceforth, this work focuses on manufacturing control. In the most general sense, manufacturing control is mainly composed of three activities: scheduling, product dispatching and on-line process monitoring (Caramia and Dell'Olmo, 2006; Leitão, 2004). ***Scheduling,*** also termed as ***detailed or short-term scheduling*** due to the short decision horizon, defines the product flow though manufacturing resources, deciding "what" must be manufactured, at what time and during which time periods (i.e., "when"); and by using which manufacturing resources (i.e., "where") (Parunak, 1991). What-type decisions are related to what product should be processed next (i.e. product sequencing), what manufacturing task should be executed next (i.e., task sequencing), what path should be taken (i.e. routing). When-type decisions precise start and finish times for the aforementioned problems, for instance, when a product should be released (i.e. release time), when a product should enter a machine's waiting line (i.e. arrival times). Last, where-type decisions allocate specific resources to accomplish manufacturing tasks, e.g., manufacturing operations, transport, and storage. All those decisions generate several manufacturing control sub-problems (e.g., task and product sequencing, product routing, machine selection, among others) because those decisions must be made respecting managerial and technical constraints such as manufacturing resources' capacity, preemptions, precedence, setup times, among others (Pinedo, 2008; Shafaei and Brunn, 2000).

An usual industrial practice is to generate short-terms schedules and then proceed to **product dispatching** to execute those schedules taking into account current status of manufacturing resources (Caramia and Dell'Olmo, 2006). Last, the **on-line monitoring** revises possible deviations caused by internal errors, process and machine perturbations and reports the schedule performance to the scheduling module so this can react and adapt properly, ensuring the continuing operation of the system.

With the constant emergence of new technologies, higher product variety, smaller lot sizes and shorter lead times, the concept of flexibility in manufacturing has become a key competitive attribute to respond to market, product and process related changes (Sethi and Sethi, 1990; Petkova and van Wezel, 2006). The emergence of flexible manufacturing systems (FMSs) is an important development in this direction.

3. Flexibility in manufacturing systems

Several definitions of flexibility can be found in related literature since there has been no agreement on a particular definition and also on the types of flexibility present in a manufacturing system (Buzacott and Mandelbaum, 2008). So far, based on the review made by Petkova and van Wezel, (2006), 141 different definitions of flexibility have been proposed and flexibility has been classified in 49 different types. In the broadest sense, flexibility can be defined as "*the manufacturing capability to cope with internal and external changes*" (Pyoun and Choi, 1994). Therefore, flexibility of a manufacturing system is dependent upon its components (machines, MHS, etc.), capabilities, interconnections, and the mode of operation and control (Joseph and Sridharan, 2011a).

3.1. Types of flexibility

Manufacturing flexibilities can be classified in three main types: basic, system and aggregate (Sethi and Sethi, 1990). Basic flexibilities, i.e., machine, material handling and operation flexibilities, are related to manufacturing system's components and parts to be produced. System and aggregate flexibilities concern the manufacturing system as a whole. In the context of this dissertation, the flexibilities defined below are considered relevant. Further information of these and other types of flexibilities can be found in (Bordoloi et al., 1999; ElMaraghy, 2006; Naim et al., 2006; Sethi and Sethi, 1990).

Machine flexibility refers to the variety of operations that a machine can perform with reasonable changeover time.

Material-handling flexibility is the ability to offer alternative transfer routes to different product types.

Operation flexibility is a property of the product that allows alternative manufacturing operation sequences to process the same product.

Process flexibility relates to the set of products that can be manufactured without major setup changes. A direct consequence of this flexibility is the absence of constraints on the sequence of operations (i.e., products) on each manufacturing resource (i.e., multiple product release sequences).

Machine-sequence flexibility is its ability to produce a part by alternative machine sequences. In the operations research domain, this type of flexibility is known as routing flexibility. The term machine-sequence flexibility was adopted to avoid any confusion with the transfer routes between machines. The effect of machine-sequence flexibility on FMS performance has been widely recognized (Baykasoğlu and Özbakir, 2008); Joseph and Sridharan, 2011b).

Volume flexibility is the ability to change the volume of output of a manufacturing process (D'Souza and Williams, 2000). In continuously changing markets and consumer behaviors, companies' profitability lies in the extent to which they can adapt their production volume to the demand (Hallgren and Olhager, 2009).

3.2. Flexible manufacturing systems (FMS)

Highly specialized transfer lines dedicated to mass production have evolved towards flexible systems, with heterogeneous components and different layouts. Flexible manufacturing systems (FMS) were introduced in the 1970s aiming for greater responsiveness to market changes, rapid turnaround, high quality, low inventory costs, and low labor costs (Basnet and Mize, 1994; ElMaraghy, 2006). Browne et al., (1984) defined a FMS as "*an integrated, computer controlled complex of automated material handling devices and compute numerically controlled (CNC) machine tools that can simultaneously process medium-sized volumes of a variety of product types*". Generally speaking, a FMS is expected to combine the productivity efficiency of transfer lines and the flexibility of job shops to attain mid-volume, mid-variety needs (Chan and Chan, 2004). Two examples of commercial FMS are shown in Figure I-2[1,2].

On one hand, the positive effects of a FMS are in fact attained if FMS's flexibility is properly utilized by the FMS control system (Anand and Ward, 2004; Baykasoğlu and Özbakir, 2008). On the other hand, internal flexibilities related to components, products and

---

[1] Image from http://www.arnold-gruppe.de/index.php?id=2&L=1 (visited 12/03/2014)

[2] Image from http://kuka.corporate-reports.net/reports/kuka/annual/2010/gb/English/1050/bric-nations-growth-market.html (visited 12/03/2014)

processes together with the number and heterogeneity of FMS components and their interrelationships, impose more complexity into the FMS control system (Chan and Chan, 2004; ElMaraghy et al., 2012; Pereira, 2013). Consequently, flexibility and FMS control are inseparable and obviously major elements necessary for providing a good performance level (Anand and Ward, 2004; Baykasoğlu and Özbakir, 2008).
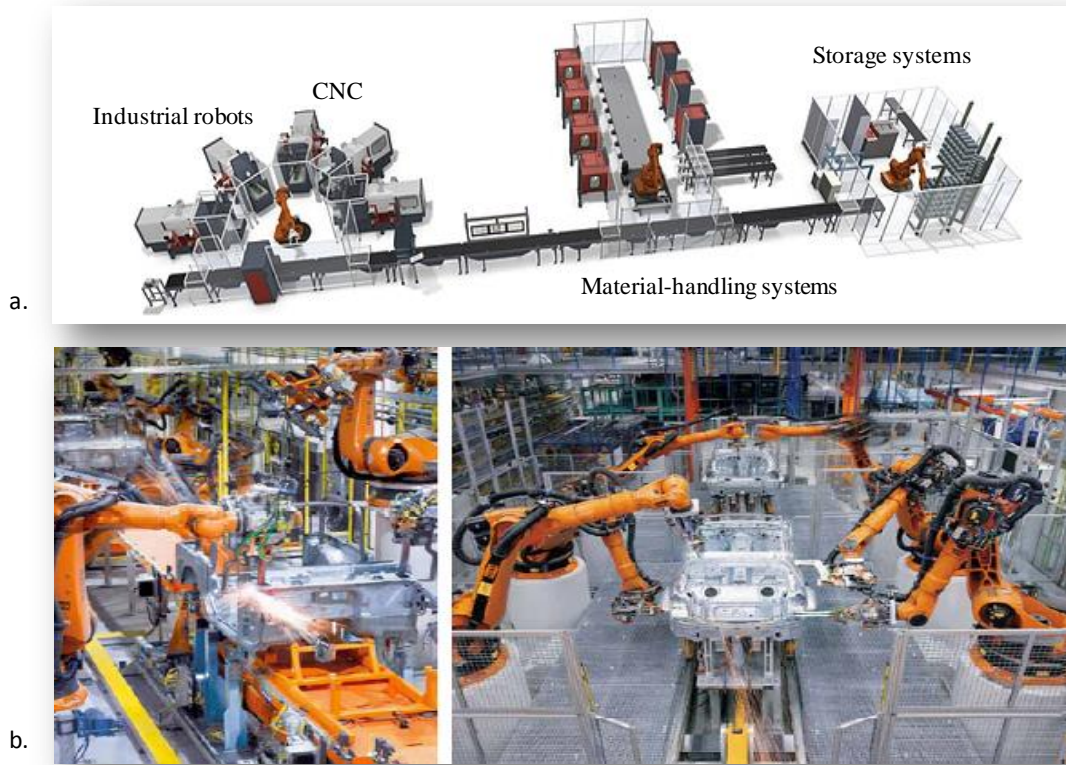
Figure I-2: Examples of an FMS. a) Basic FMS[1], b) FMS in automobile industry[2]

Our interest is focused on FMS control, and to design it, it is necessary to define the decision-making algorithms responsible for making control decisions and specify the control architecture that allocates those decision-making algorithms to one or more control system components. The next section presents a review of decision-making algorithms while Section I-5 takes a closer look on control architectures.

4. Decision-making algorithms

According to definitions of manufacturing control given in Section I-2, FMS control corresponds to operational level decisions dealing with scheduling, dispatching and on-line monitoring of FMS components and production status. For the most part, it is FMS scheduling that is in charge of orchestrating resource utilization, ensuring responsiveness to changes in resource conditions and production demands, and meet production objectives.

Thus, FMS performance strongly depends on scheduling decisions. Therefore, since the FMS scheduling algorithm takes into the decision-making process carry out in the operational level, herein we consider scheduling algorithms as decision-making algorithms. This section focuses on those decision-making algorithms that have been used for FMS control.

From an Operations Research perspective, scheduling in FMS is a more complex version of the classical flexible job-shop scheduling problem, which is known to be NP-hard (Conway et al., 2003). Therefore, it is frequently observed in literature that the FMS scheduling problem is addressed with hard assumptions (e.g., neglecting transport times, unlimited buffer capacity), constraint relaxations, and regular and single performance criteria (Shnits et al., 2004; Shirazi et al., 2011). Since there are different types of algorithms, there has been extensive studies on scheduling algorithms, each one proposing their own classification depending on different features, such as algorithm complexity, optimality degree and type of environment, i.e., deterministic or stochastic (Baker, 1998; Spano et al., 1993; Vieira et al., 2003). Herein, scheduling algorithms are classified in four basic approaches depending on the quality of solutions and algorithm complexity: optimal and near-optimal, artificial intelligence, heuristics and dispatching rules (denoted from a) to d) in Figure I-3).
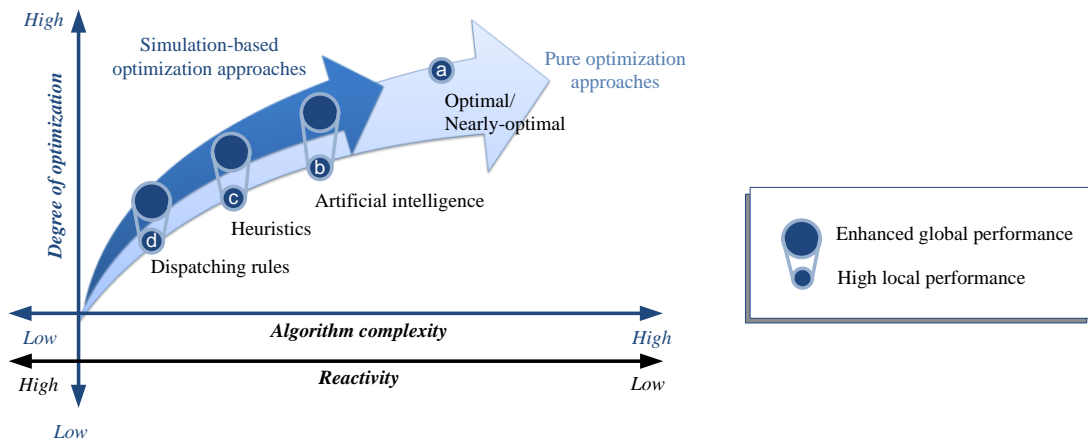
Figure I-3: Scheduling algorithms

The ***algorithm complexity*** refers to the amount of information and the computational effort the algorithm requires to analyze the continuous expansion of possible combinations of FMS control states with time, either combinatorial or periodic (Chryssolouris et al., 2013). Hence, the algorithm complexity determines the ***reactivity*** of the decision-making algorithm, meaning its capacity to start a decision making process for searching alternative solutions, evaluating them and selecting one at each decisional point (e.g., due to a perturbation) as shown in Figure I-4. Thus, there is a high risk of global performance loss if decision making

is not undertaken or the algorithm's time to provide a solution is significantly high compared to system dynamics. Therefore, the ***overall performance*** of a decision-making algorithm can be defined as the balance between the quality of the solution (i.e., alternative solution analysis) and the algorithm's reactivity.

As mentioned in Section I-3.2, FMS are characterized by a high level of complexity; hence analytical methods searching for optimal solutions are solvable only for small problems under simplified assumptions and can rarely provide good reactivity (Lee and Cheng, 1996). Conversely, low-complex scheduling algorithms applying heuristics and dispatching rules have been proposed to reduce such complexity by assessing less amounts of information, tackling control decisions for short-time periods and simultaneously improving the control system's reactivity. In the middle (Figure I-3), artificial intelligence techniques have demonstrated their success because of their ability to find good (not necessarily optimum) solutions in reasonable time periods.

In complex cases such as FMS scheduling, coupling simulation with optimization techniques can be a suitable alternative to improve the algorithm's performance under different conditions (Iassinovski et al., 2003). More on the role of simulation in FMS control is given in Section 4.4.
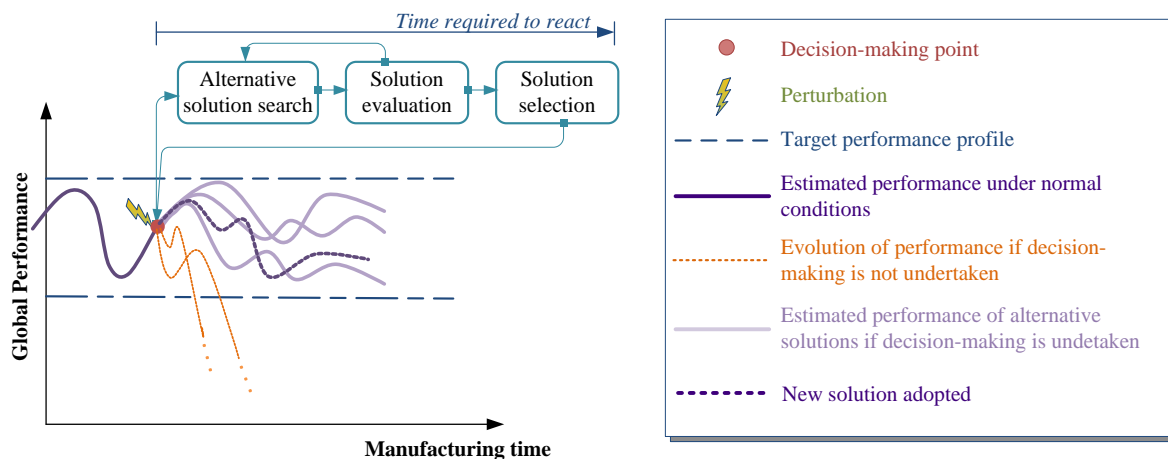
Figure I-4: Reactivity and global performance

4.1. Optimal and nearly optimal approaches

These methods are issued from the Operations Research domain. ***Optimal methods*** are based on mathematical programming, such as linear programming algorithms, dynamic programming and branch and bound methods (Pinedo, 2008). Since using optimal methods to solve realistic problems is computationally intensive, ***nearly optimal methods*** try to reduce the algorithm complexity by relaxing certain problem requirements and constraints, with for

example Lagrangian relaxation techniques (Baker, 1998). For instance, the FMS problem can be modeled as the travelling salesman problem or the flexible job shop problem (Toth, 2000; Rossi and Dini, 2007). Some examples implementing integer linear programs in FMS control are reported by Pach (2013) and Gou et al., (1994).

4.2. Artificial Intelligence (AI) approaches

Contrary to optimal algorithms, AI-based algorithms try to mimic the human brain or biological systems in an attempt to find good solutions (possibly the optimal) to high complex problems, with less calculation effort thus favoring reactivity (Brooks, 1995). Meta-heuristics such as simulated annealing (SA), Tabu search (TS), genetic algorithms (GA), ant colony optimization (ACO) and particle-swarm optimization (PSO) are some examples. For instance, AI-based approaches dealing with the FMS problem divide the scheduling problem into various sub-problems (i.e., machine allocation, route and task selection), for which different AI-based algorithms can be used, either sequentially or integrated (Roshanaei et al., 2013). In the cases of Kang et al., (2007) and Asadzadeh and Zamanifar, (2010) ACO and parallelized GAs are used within a multi-agent architecture to deal with job-shop-based problems.

Artificial intelligent approaches have evolved towards distributed approaches (distributed artificial intelligence, DAI) to avoid that a single entity carries out with global information and centralized algorithms necessitating global models. The DAI field has significantly contributed to manufacturing control with modeling approaches based on biology, physics and social organizations (a brief description of some well-known modeling approaches is presented in Appendix B). In DAI approaches, decisional entities are typified based on the manufacturing control components (e.g., machines, products, orders, workcells) and they create schedules progressively using negotiation protocols, such as bidding (Lima et al., 2006; Wei et al., 2007), market-like mechanisms (Lin and Solberg, 1992; Wu and Weng, 2005) or product-driven techniques (Zbib et al., 2012). More examples are reported in literature reviews conducted by Shen et al. (2006a) and Shen et al. (2006b).

4.3. Heuristics and dispatching rules approaches

Heuristics and dispatching rules are probably the simplest techniques and by far the most commonly implemented techniques in industry (Baker, 1998). Their success lies in that feasible solutions can be generated in a reasonable amount of time, even for highly complex problems. Though they are highly reactive, their performance cannot be proven to be within an acceptable range or evolve towards optimality. Forward/backward heuristics, beam search and greedy are some of the mostly used heuristics. For example, Wang et al., (2008)

proposed a filtered-beam-search-based heuristic inside a cell coordination agent to control FMSs.

In turn, dispatching rules are the most simple rules, since they are temporally local and do not try to predict the future, but make decisions based on current and local information. Dispatching combined with priority rules for resource allocation is the best known heuristic for scheduling (Reaidy et al., 2006). Hybrid dispatching rules, either sequential or integrated, can also be implemented to improve the system response to various types of conditions (Sels et al., 2012).

4.4. Simulation-based optimization

(Shannon, 1998) defined simulation as *"the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system"*. A simulation model can be conceived as a set of algorithms, instructions, rules or equations that represent the system (Zeigler et al., 2000). The valuable contribution of simulation for manufacturing systems modeling and analysis relies in several advantages (Habchi and Berchet, 2003; Shannon, 1998). For instance, a simulation model can be constructed from a system that already exists or a system that is in the designing stage. More, realistic models are possible, especially for complex and stochastic systems (e.g., FMSs) for which analytical methods are perceived unhelpful or extremely difficult. Furthermore, several options and alternatives can be considered for evaluating different conditions and situations that otherwise would not be possible.

Simulation of manufacturing systems is performed using one of three simulation methods: Discrete Event Simulation (DES), System Dynamics (SD), and Agent-Based Modeling and Simulation (ABS[3]) (Seleim et al., 2012). Although these simulation techniques have been mainly used for manufacturing systems design, with the first models dating back to the 1960s (Law and McComas, 1987); an increasing number of research is being focused on simulation for manufacturing systems operation applications, especially for complex manufacturing systems such as FMSs. Within the operation of manufacturing systems, simulation has been used to address different problems regarding operations planning and scheduling, maintenance operations planning, real-time control and operating policies (Negahban and Smith, 2014).

The features provided by the new generation of simulation software facilitate the integration of the simulation models with production control systems. Mainly used as a

---

[3] Herein, the term ABS (Agent-based Simulation) is used as a generic term, which includes ABM (Agent-based Modeling).

descriptive technique, simulation has been applied to evaluate schedules through a series of computer-based experiments, estimate performance measures of re-scheduling strategies, assess the impact of a re-scheduling interval or test what-if scenarios and use such information for rescheduling (Pfeiffer et al., 2008). Schedules can be generated using traditional approaches such as dispatching rules and heuristics or artificial intelligence techniques (Li et al., 2000; Chan and Chan, 2004; Negahban and Smith, 2014). The main drawback of this technique is to know how to drive the experiments since multiple simulation replications with possible multiple parameter combinations require abundant computing capacity (Hong and Nelson, 2009). Optimization techniques such as gradient, stochastic, response surface and heuristic-based methods have been proposed to orchestrate simulation experiments, aiming to improve the process by only evaluating a smaller percentage of alternatives, without sacrificing performance (Paris and Pierreval, 2001; Law and McComas, 2002; Habchi and Berchet, 2003). Due to these reasons, simulation-based optimization techniques have attracted increasing attention for many researchers in areas such as supply chain (Ding et al., 2005), vehicle control (Montoya-Torres et al., 2005), manufacturing systems design (Truong and Azadivar, 2003), among others.

5. Control architectures

The control architecture defines the blueprint for the design and construction of FMS control (Smith et al., 1996). Depending on the structure, the control architecture allocates control responsibilities on one or more decisional entities, determines the inter-relationships between them and establishes the coordination mechanisms for the execution of control decisions. According to the definition proposed by Trentesaux (2009), a *decisional entity* (DE) is a generic term referring to any kind of autonomous unit able to communicate, to make decisions and to act within a manufacturing scenario.

Four basic manufacturing control architectures have been identified (Dilts et al., 1991; Trentesaux, 2009): centralized, fully hierarchical, fully heterarchical and semi-heterarchical (Figure I-5). The centralized control is characterized by a single DE, i.e., centralized controller, which controls the entire system (Vieira and Veiga, 2009; Dilts et al., 1991). Aside from the fact that centralized control allows global optimization since the controller has an entire view of the system, there are hardly any other benefits of this control architecture (Duffie, 2008; Vieira and Veiga, 2009). In the following, the other control architectures are briefly described.
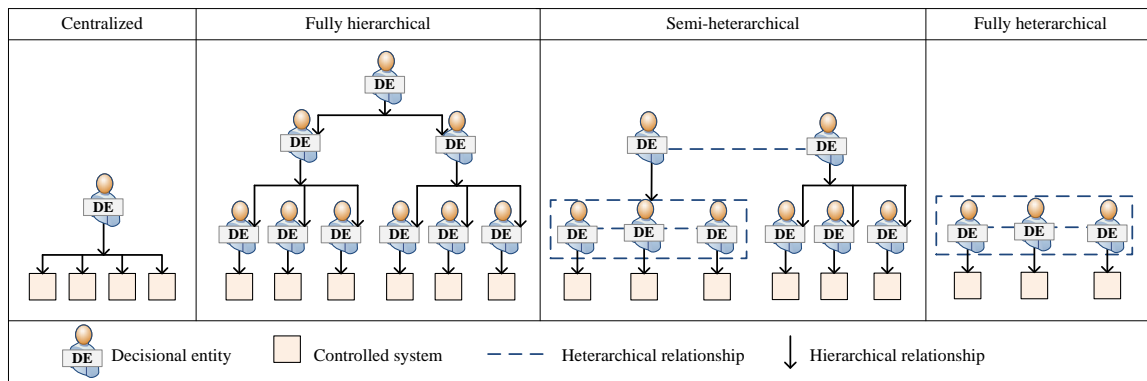
Figure I-5: Manufacturing control architectures

5.1. Fully hierarchical control architectures

These architectures divide the control system into various control levels to reduce the complexity of centralized approaches. In a hierarchical structure, interactions strictly follow a top-down flow of commands, establishing a rigid master-slave relationship between decision-making levels. Hence, decisions at one level constraint the decisions at the sub-sequent lower level, limiting the responsibilities and authority at each level (Smith et al., 1996). Typically, at the top of the hierarchy there is a single decisional entity that is in charge of tracking the overall system efficiency along the entire planning horizon. Planning horizons become shorter as the level goes down, shifting level functions from planning to execution. Feedback and status report from subordinate levels are sent to upper levels, integrating data and feeding aggregate data bases at each level (Dilts et al., 1991; Parunak, 1991). Compared to the centralized structure, hierarchical architectures allow an incremental and gradual implementation of control, which results in reduced software development time. More, the division of control into several levels not only allows integrating adaptive behaviors, but also limits the complexity, resulting in faster response times (Dilts et al., 1991).

Since the AMRF architecture was proposed in 1981 (Jones and McLean, 1986), other architectures such as MSI (Senehi et al., 1994), CIM-OSA (Kosanke et al., 1999) and PAC (Maglica, 1997) have been proposed, focusing more on the operational decisional level, instead of the complete enterprise system. But it is probably the concept of computer integrated manufacturing (CIM) that benefits more from hierarchical control architectures, since most of CIM implementations display a hierarchical structure (Bongaerts et al., 2000; Nagalingam and Lin, 2008).

In spite of these benefits, the rigidity of hierarchical control imposes some major disadvantages. As pointed out by Trentesaux (2009) sufficient long-term and global optimization is only achieved when deterministic behaviors are assumed. And, due to the

16

rigid structure, entities can hardly be reactive and take the initiative to deal with perturbations. Instead, perturbations must be propagated to the upper planning levels, generating lags and disfavoring the system's agility (Van Brussel et al., 1999; Monostori et al., 2006). As a consequence, introducing fault tolerance into hierarchical structures without significantly increasing system complexity is a difficult and expensive task (Duffie, 1990). In addition, hierarchical architectures are difficult to modify and maintain, making any structural changes (e.g., adding or withdraw machines, controllers) difficult while the system is operating (Van Brussel et al., 1999).

5.2. Fully heterarchical control architectures

Since the 1980s, several researches have expressed their concern about the rigidity of fully hierarchical architectures. Hatvany (1985) pointed out the need for a new type of control architecture taking advantage on distributed computing and inspired on open system architectures, such as communication networks. Heterarchical control architectures were thus proposed based on fully decentralized control, retention of a minimal amount of global information (eliminating the need of global databases), and cooperation among loosely coupled, autonomous, communicating decisional entities (Hatvany, 1985; Duffie, 1990; Duffie, 2008). Fault tolerance is achieved by the decomposition of the system into quasi-independent decisional entities, resulting in high local autonomy and avoiding master-slave relationships. Time critical responses are handled locally and should be independent of other time critical responses from other entities. However, decisional entities are encouraged to cooperate with each other, but following the principle of least commitment (Duffie, 1996). The adaptability of heterarchical architectures is ensured by independent modes of operation of decisional entities and their equal rights to access resources.

Regardless its numerous benefits, fully heterarchical architectures have been rarely implemented due to several issues (Prabhu, 2003; Trentesaux, 2009). Leitão (2009) classified these issues in two groups: development- and conceptual-related aspects. Development-related restrictions arise from the lack of design methodologies and standards defining explicitly the structure of decisional entities, the cooperation methods, communication and interoperability protocols. Advances in commercial platforms and industrial controllers are also necessary to handle real-size industrial applications (Mařík and McFarlane, 2005). Conceptual issues are consequence of the high local autonomy. Since there is no central control element, these systems can be highly unpredictable and non-expected emergent behaviors can appeared, including chaotic behavior (Hogg and Huberman, 1991; Thomas et al., 2012). In addition to that, incomplete information make difficult to ensure that local decisions are globally coherent, thus it is hard to guarantee a minimum level of performance (Duffie and Prabhu, 1994). Hence, global optimization is not possible, and conversely, local

responses to perturbations can induce still larger perturbations in the system (Van Brussel et al., 1999).

5.3. Semi-heterarchical control architectures

Semi-heterarchical control (also called modified hierarchical by Dilts et al. (1991) tries to combine the advantages of hierarchical and heterarchical architectures, while avoiding their respective drawbacks. Though semi-heterarchy implies multi-level relationships, the difference with purely hierarchical structures lies in the low-level autonomy. In this particular architecture, the notion of upper-level suggests more flexible and adaptive entity relationships, avoiding rigid structures. Thus, the upper-level decisional entity[4] (or entities) becomes an assistant, capable of processing more information than low-level entities, handling all sorts of interaction issues and conflicts at the lower level, and solving the duality of local and global performance (Dilts et al., 1991). Consequently, the overall efficiency of the system is managed by the upper level since it has a better view of the entire system, while robust operation and reactivity to perturbations is provided at the lower level (Monostori et al., 2006; Rahimifard, 2004). Two important points about semi-heterarchies have been also highlighted by Bongaerts et al. (2000). The first refers to the fact that hierarchy into heterarchy helps to predict the behavior of the control system. This is an important issue for industrial adoption of heterarchical-based approaches (i.e., fully heterarchical or semi-heterarchical). The second benefit of hierarchy in heterarchy is that it eases the migration from current industrial fully hierarchical applications to more decentralized control approaches.

The main shortcomings of this type of architecture are inherited from hierarchical relationships, the specification of autonomy for low-level entities and the coordination between levels. So far, most of semi-heterarchical approaches have proposed structural modifications that tend more towards hierarchy than heterarchy (Leitão, 2009). Although the basic heterarchical structure is not lost at all, the disadvantages of hierarchical dependencies overshadow the heterarchical principles.

6. Challenges, requirements and issues of FMS control

Manufacturing control, including FMS control, is facing several ***challenges*** issued from rapid technological innovations, globalized and customer-driven (i.e., mass customization) markets. Increasingly rapid technological leaps not only concern products but also manufacturing systems themselves. Rapid changes in manufacturing technology require manufacturing resources and control systems to be easily upgradeable, so new technologies

---

[4] Decisional entities at this level are usually refer as supervisors, mediators or coordinators depending on their functionalities.

and new functions can be readily integrated (Pereira and Carro, 2007). Manufacturing enterprises are intensively deploying computer, communication and information technologies in order to face constant changes pulled by globalization and mass customization (Morel et al., 2003).

Indeed, globalization allows manufacturing companies to increase markets opportunities; however, globalization also leads to new issues related to the conception and management of supply chains. Supply chains are evolving towards extended and virtual enterprises aiming more dynamic and interoperable organizations. Such interoperability require full integration of heterogeneous software and hardware systems within at all levels of the enterprise and across the supply chain (Shen et al., 2006a). In the era of interconnection, we are moving towards the "Internet/Web of Things domain" were millions of devices can provide, request and obtain information available on the network (Guinard and Trifa, 2009; Atzori et al., 2010). Nowadays, there is a weak coupling between enterprise resource planning (ERP), with the manufacturing execution system (MES) and control systems, even weaker or inexistent at the supply chain level (Pawlewski et al., 2009; Brintrup, 2010).

Mass customization results in rapidly changing customer requirements, accelerated innovation and shorter product lifecycles. Nowadays, customers significantly influence manufacturing processes through the imposition of personal specifications and exigency for products with higher quality (Christo and Cardeira, 2007). As a consequence, manufacturing control has to deal with lower sized batches, even one-of-a-kind products, and smaller delivery times. Therefore, to accommodate such variations in product quantity, quality and specification types, manufacturing control *requires* to be flexible, scalable, and easily reconfigurable (McFarlane and Bussmann, 2003). In addition, control systems need to be reactive and adapt rapidly to external changes, fault tolerant to detect and gracefully recover from system failures and minimize their consequences; as well as modular and easy to interoperate to manage efficiently recent manufacturing technologies and legacy systems (Colombo and Karnouskos, 2009; Chituc and Restivo, 2009). In addition, new managerial philosophies such as Just-in-time production (JIT) impose additional requirements to eliminate or reduce several sources of waste (more insight on JIT production is imparted in Appendix A-III).

Nevertheless, in the pursuit of those requirements there are still some *issues* to take into consideration such as high investment costs, the need of new interdisciplinary engineering and design methodologies, the guarantee of near-optimal or satisfactory performances with sufficient reactivity to face perturbations (Trentesaux, 2009; Leitão, 2009). Precisely, in the search for reactivity, adaptability and fault tolerance, it is likely to fall into low complex approaches that only evaluate few alternative solutions to favor rapid responses to production

changes, internal or external. Such myopic approaches would, most likely, yield short-term and partial gains (e.g., particular to certain clients, orders, products, etc). However, more often than not, such gains may lead to significant performance losses in the long-term and for the entire production. This myopic behavior has been identified as an issue in various areas related to manufacturing such as enterprise management and planning (Miller, 2002), marketing (Johnston, 2009), and manufacturing control (Trentesaux, 2009). The following section focuses on this myopic behavior, proposing a definition and typology of such behavior, along with its implication in FMS control.

7. Myopic behavior

Decision making is an everyday practice in human and productive organizations, aimed at setting-up goals and accomplishing them. The analogy of this term with that in the medical domain is the limited *"visibility"* of decision makers, i.e., human or artificial entities, to gather and assess information in the current and the long term, necessary to drive the system or organization towards the global objective (Trentesaux, 2009).

With the evolution of control architectures towards non-centralization, i.e., heterarchy, myopic behavior becomes a more evident issue because each decisional entity has a narrower visibility of the entire system and it is only focused on its local objective. In addition, the fact that there is no central entity that guides them towards a global objective does not ensure that those local decisions are aligned with the global objective. Though myopic behavior seems an interesting attribute to keep with fast pace events and achieve short-delay responses, it is also a critical issue that affects the guarantee of a minimum operational performance (Duffie and Prabhu, 1994; Trentesaux, 2009).

Aside from some authors that have identified this issue and very few works on the subject, myopic behavior has not been formally studied in heterarchical-based FMS control (HFMS), i.e., FMS control with heterarchical relationships, fully heterarchical or semi-heterarchical (Zambrano Rey et al., 2011a). Different works and collaboration within the Production, Services and Information (PSI) Team of TEMPO Laboratory have allowed us to define and typify this behavior (Pach and Zambrano Rey, 2011; Pach, 2013). To this end, some concepts and characterizing elements were first withdrawn from other domains and then transferred in the HFMS control context. This section then follows this logic.

7.1. Concepts of myopic behavior in other domains

Myopic behavior has been studied in several domains such as economics, organizational behavior, finances, marketing, inventory theory and robotics. Table I-1 summarizes various types of myopic behaviors, their causes, the characteristics displayed by decision-makers,

possible consequences of myopic decisions and context. The conclusions of those works reveal that, in all cases, myopic decisions yield in performance losses, overemphasizing short-term and local efficiency instead of maximizing long-term share-holder performance.

Table I-1: Concepts of myopic behavior

Domain	Type of myopic behavior	Causes	Characteristics	Consequences	Context
Economic (Thaler et al., 1997)	Myopic loss aversion	External pressures Urgent decisions	Frequent estimation of outcomes Short-term decisions Underestimation of sequences of investments	Rejection of investment programs that would have been accepted otherwise	Temporal
Organizational behavior (Reb and Connolly, 2009)	Myopic regret avoidance	Self-concern over collective benefits	Aversion to feedback Protection of the decision-makers in the short term	Reduced task learning in the long term	Social and temporal
Marketing (Johnston, 2009)	Marketing Myopia (extensions of capability and boundary myopia)	Unawareness of competences Isolation	No idea of what customers really want Tight collaborations Lack of diversity	Failure in creating new value propositions	Social
Knowledge inventory (Miller, 2002)	Managerial myopia	Cognitive limitations	Sequential attention to goals Limited, single-period foresight Lack of awareness of alternatives Considering investment decisions singularly rather than evaluating them as part of the overall portfolio	Missing optimization opportunities and eliminates consideration of interactions	Temporal and social
Robotics (Mataric, 1992)	Myopia	Ignorant coexistence Unreachable information	Selfish behavior Local minima Oscillatory behavior	Loss efficiency to reach global objectives and longer response times	Social and temporal
Robotics (Bajracharya et al., 2009)	Myopic path planning	Restricting perceptual look-ahead distance	Short-range sensing Short-range learning	Restriction of the maximum safe driving speed	Social

7.2. Myopic behavior in HFMS control

In HFMS control, several characteristics reported in Table I-1 are manifested due to the local goal orientation, locally-contained information, and high degree of autonomy of loosely-linked decisional entities (Trentesaux, 2009; Leitão et al., 2010). For instance, bidding, market-like and product-driven approaches use frequent estimation of outcomes and decisional entities display selfish and competing behavior (Zbib et al., 2012); multi-agent and holonic systems encourage the formation of societal structures, such as holarchies, coalitions, teams, congregations, etc., limiting the interactions among entities and creating tight collaborations (Isern et al., 2011). Entities count on limited processing capacities, avoiding complex decision making rules and forecasting methods. Taking those elements and some previous works (Adam et al., 2011; Zambrano Rey et al., 2011a), the following definition of myopic behavior is then proposed (Zambrano Rey et al., 2013):

"Myopic behavior in heterarchical-based control occurs when entities are not capable of balancing their local objectives with the system's global objectives, thus compromising the overall system's performance. A myopic decisional entity experiences a social myopic

behavior and a temporal myopic behavior in terms of, respectively, projecting its decisions towards other entities in the organization and projecting its decisions in the long-term"

Below, the social and the temporal dimensions of myopic behavior are explained before analyzing the implications of this issue for HFMS control. In Appendix C, an analysis of social and temporal myopia is carried out on two examples of FMS control approaches.

7.3. The social dimension of myopic behavior

Social myopia can be understood as a knowledge limitation of decisional entities. Knowledge constraints concern data awareness and decision techniques (Filip, 2008). Data awareness is a difficult task due to the highly dynamic nature of FMS and the amount of data to be collected and processed. In heterarchies, data is gathered locally by sensing the environment or inquiring other entities. Information sharing is the basis of cooperation[5]. Therefore, an entity becomes socially myopic when it is not able to directly access the data it requires, or it is able to obtain only partial information (e.g., myopia in robotics – informed coexistence (Mataric, 1992).

Once a certain amount of information has been gathered, it is the turn of the decision-making algorithm to analyze it, to achieve its local goals and contribute to the accomplishment of global goals as well. To do so, decisional entities usually put together some planning, which is the sequence of actions or sub-actions that will leads from their current state to the objective (Pomerol, 1997). A decisional entity is also socially myopic due to its finite computation capabilities to construct such planning. As such, the problem is usually partitioned because it not only results in less information to collect and analyze, but also low-complex decision methods can be used and faster responses can be obtained. Thus, the solution is usually constructed progressively and decisional entities proceed depending on how results confirm their choices (as in myopic loss aversion (Thaler et al., 1997; Reb and Connolly, 2009), managerial myopia and myopic path planning (Bajracharya et al., 2009)). A particular example of this behavior results from using priority and dispatching rules, given that each rule only needs current production conditions and local information (Sels et al., 2012).

Taking these elements as reference, social myopia can be defined *as a decision-making limitation concerning data gathering from the environment and from other entities, and its subsequent analysis to find alternative solutions.*

---

[5] Herein, cooperation is understood as the ability of a decisional entity to built mutually acceptable plans and execute them (McFarlane and Bussmann, 2003)

7.4. The temporal dimension of myopic behavior

Temporal myopia can be analyzed from two time-related limitations. The first limitation concerns the solution search process which depends on the time spent for searching alternative solutions. For instance, in cooperation-based approaches the communication costs can be determinant, especially in systems with a large number of participating entities (Becker et al., 2009). The second limitation is related with decision-making techniques and estimations of long-term consequences of short-term decisions. Given the high local autonomy of decisional entities in heterarchical-based control, it is very hard to anticipate reactions of other entities and the environment because of the combinatorial nature of entities interactions and delays between actions and reactions (Feigenbaum et al., 2003; Benaissa et al., 2008). From this assumption, it can be concluded that temporal and social myopic behaviors are closely related.

To reduce temporal myopic behavior, additional look-ahead capabilities may be used but at the price of incurring in supplementary costs issued from longer processing times. Therefore, temporal myopic entities end up by working on smaller decision horizons overemphasizing current-term results at the expense of long-term performance. Then, the quality of the final solution is very sensitive to the definition of such decision horizon (Jia et al., 2009).

As for social myopia, the following definition of temporal myopia is proposed: *temporal myopia is a time-related knowledge limitation concerning the long-term estimation of current decisions, on a local and a collective perspective.*

Though myopic behavior is beneficial for reactivity and low-complex decision making, it has certain implications on FMS control, which are presented in the next part.

7.5. Implications of myopic behavior for HFMS control

Myopic behavior has been recognized as an important barrier for adopting heterarchical control in industrial applications because it makes it hard to predict and guarantee a certain level of global performance (Duffie and Prabhu, 1994; Mařík and Lažanský, 2007). As it has been mentioned, the behavior expected from interacting myopic entities results in local optimality since there is no explicit way to relate the effect of those local actions to global system performance. Therefore, global performance is highly sensitive to the underlying myopic decision-making algorithms (Maione and Naso, 2003).

A second implication of myopic behavior has been identified as system nervousness (Hadeli et al., 2006). This issue arises when decisional entities react abruptly and without forecasting their decisions. The system becomes nervous because those decisions may trigger

subsequent disruptions in other planning activities such as maintenance, inventory, procurement, etc; resulting in additional rescheduling costs (Pujawan, 2004).

Last, another undesirable effect of myopic behavior is the unpredictability associated to the decision-making algorithm. On one hand, myopic behavior helps to respond quickly to unpredictable changes in the manufacturing system and its environment; but on the other hand, the way in which decision-making algorithms will react is also unpredictable, so it is hard to determine beforehand if a perturbation will be handled efficiently or not. In some cases, myopic decisions can lead to deadlocks, which probably results in more performance losses than those caused by the perturbation.

8. Synthesis and discussion

Manufacturing is one of the most important wealth generators for a country, and therefore manufacturing efficiency is at the most important matter. In the last decades, manufacturing industry has been facing several challenges (e.g., globalization, mass customization, market volatility, short product life-cycles), requiring highly adaptive, reactive and fault-tolerant manufacturing control. Flexible manufacturing systems have been implemented in the search of efficient production systems in environments with rapid changes, small batches and high variety of products. However, to achieve those requirements it is necessary an intelligent control system that makes an efficient use of flexibility.

Leading-edge technologies in computing, industrial communications and software platforms have allowed the upcoming of new control architectures and algorithms focused on heterarchical control and intelligent decisional entities, capable to communicate, make decisions and cooperate to reach global objectives. Those new paradigms are suitable for FMS control given its complexity and the heterogeneity of components and manufacturing operations. However, in spite of promising advances, heterarchical-based control leaves some important issues among which myopic behavior. Since this issue has been recognized as an important barrier for adopting heterarchical-based control in industrial applications and this behavior has not been formally studied in HFMS, this dissertation focuses on myopic behavior in HFMS.

Given that myopic behavior affects significantly FMS global performance of heterarchical-based control, several approaches have been proposed to improve global performance, indirectly tackling myopic behavior. Among the different possible solutions to reduce myopic behavior, two main options seem possible: increase the efficiency of the decision-making algorithm or integrate additional techniques to the underlying decision-making algorithms. The former is for instance achievable by using more complex algorithms,

such as enumerative methods embedded into local decisional entities. However, not only the construction of models is a difficult task and already induces inaccuracies due to modeling assumptions and errors in data collection, but also these methods are prohibitively expensive in computing cost (O'Grady and Menon, 1986). On the same track, decision making can be improved by reusing historical system states and data, so it is possible to better characterize, analyze, predict and even learn (Li and Yeh, 2008; Choudhary et al., 2009). This may seem possible if there is a strong regularity, causal relationships can be identified and large sets of data are available for analysis.

In the second set of approaches, integrating additional techniques to the underlying heterarchical-based approaches is another strategy to reduce myopic decision-making. These approaches have mainly considered the introduction of additional optimization and simulation techniques in order to improve solution exploration, solution evaluation and look-ahead assessment. The main benefit of doing so is to keep low complex control while global performance can be enhanced. Since we consider this spectrum of possibilities a promising path for this research, in the following chapter, simulation, optimization and simulation-based optimization techniques introduced into HFMS control are studied and analyzed in regards to myopic behavior.

Chapter II. Literature Review on Myopic Behavior Reduction Using Simulation and Optimization

1. Introduction

As concluded in the Chapter I, integrating simulation and optimization techniques seem a promising possibility to reduce myopic behavior in heterarchical-based FMS (HFMS) control, i.e., fully heterarchical or semi-heterarchical FMS control. Since the integration of such techniques may induce architectural changes, this chapter aims at studying how those techniques have been used to reduced temporal and social myopic behaviors in fully heterarchical FMS control (FHFMS) and semi-heterarchical FMS control architectures (SHFMS). To this end, the approaches reported in this chapter are classified taken into consideration the type of technique used for improving global performance: optimization, simulation or simulation-based optimization (Sections II-3, II-4 and II-5 respectively).

With the purpose of facilitating the comprehension of such classification and positioning this literature review, a general synthesis framework for reducing myopic behavior, emerged from the analysis of all approaches reported herein, is proposed beforehand in Section II-2. The last section of the chapter presents a summary and discussion on the effectiveness and drawbacks of the reported approaches from three perspectives: myopic behavior reduction, decision-making algorithm complexity and gain in the overall performance.

2. General synthesis framework for reducing myopia behavior

Generally speaking, several approaches have been proposed to improve the global performance of heterarchical control based on techniques such as optimization, simulation or simulation-based optimization. Architecturally, certain techniques have been added preserving fully heterarchical FMS control (FHFMS), while other techniques have required adding a hierarchical level to the underlying heterarchical level, resulting in semi-heterarchical FMS control architectures (SHFMS). This modeling choice influences aspects such as the decision-making algorithm and the control strategy. The control strategy determines the control system dynamics under normal conditions and in the occurrence of internal and external perturbations. Between fully reactive, proactive and predictive-reactive,

the latter is the most frequently used control strategy. More information of these control strategies is given in Appendix D.

With the purpose of differentiating decisional entities participating in both types of architectures, henceforth a decisional entity working on a heterarchical level is denominated local decisional entity (LDE), and that working on the hierarchical level is denominated global decisional entity (GDE). LDEs can be assigned to represent physical objects (e.g., machines, products) or logical objects (e.g., production orders) in the working environment, i.e., the FMS; while GDEs are more concerned on global tasks such as global performance monitoring, and LDEs coordination and guidance. The $GDE - LDE$ denomination also helps to distinguish the entity's decision-making span; while LDEs work on local solutions with local decision-making algorithm, GDEs work on more global solutions using global decision-making algorithm. More, each entity adopts differently the additional simulation/optimization (S/O) techniques.

Figure II-1 presents the proposed general synthesis framework making the distinction between FHFMS and SHFMS architectures (Zambrano Rey et al., 2014). As seen in Figure II-1, for those approaches that can be classified as FHFMS, additional S/O techniques can be introduced mainly for one of three roles: evaluating, cooperating or informing other decisional entities. These additional techniques are either adopted by the LDEs as part of their decision-making algorithm or as part of their control strategy. In the first category ((1) in Figure I-1), the S/O role aims to improve the entity's data analysis, yielding better solution evaluation. The S/O techniques work directly on the LDE's decision-making algorithm.

The second category ((2) in Figure II-1) gathers those techniques used to improve entities' cooperation (e.g., negotiation protocols) and the third category ((3) in Figure II-1) congregates those techniques that try to improve the way in which local information is exchanged between entities. These two types of techniques affect the control strategy and the entities' interactions by defining, for instance, the access to information. In general, the main objective of additional S/O techniques is to broaden the LDEs' solution exploration, improve their look-ahead solution assessment and reach a tradeoff between local and global performance.

By introducing a specialized entity (one or more GDEs) with a global view of the system, it has been possible to obtain a SHFMS in which a larger set of S/O techniques and notably simulation-based optimization techniques have been used. In most of the reported approaches, these techniques are localized within the GDE's decision-making algorithm. For instance, the PROSA reference holonic manufacturing architecture proposed by Van Brussel et al. (1998) already defines a specialized entity (i.e., the staff holon) that can hold centralized

algorithms, useful when basic holons perform badly. Contrary to FHFMS approaches, SHFMS architectures have demonstrated better overall performance, even in highly uncertain environments (Brennan, 2000; Bongaerts et al., 2000; Cavalieri et al., 2000). Furthermore, Wong et al., (2006) demonstrated that including an upper decisional level also reduced communication overheads. Nevertheless, the SHFMS architecture inherits the disadvantages of hierarchy (discussed in Section I-5.1) as well as an additional challenge related to the management of *LDE*'s autonomy. It is important to point out that, the two-level SHFMS is a basic structure that can be recursively replicated, for instance to comply with holonic and fractal principles. This fact still needs to be studied and it is out of scope of this dissertation.
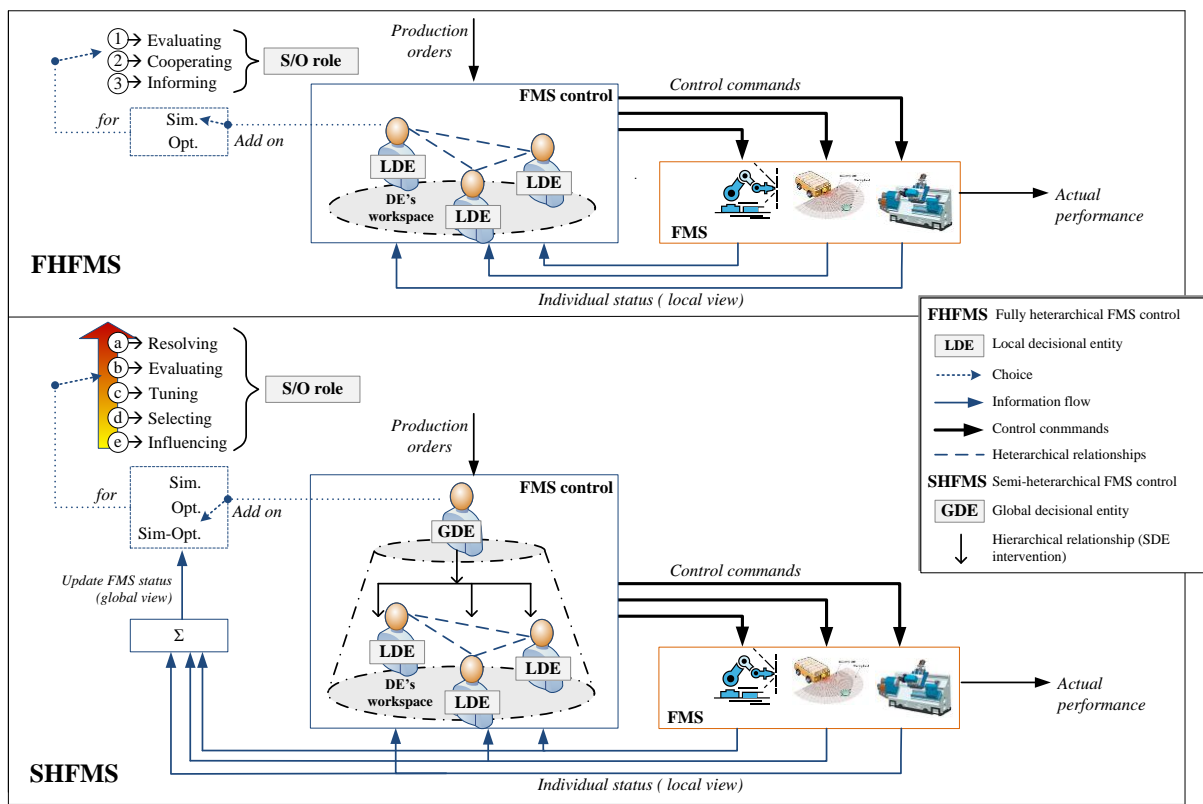
Figure II-1: General framework for myopic behavior reduction

As depicted in Figure II-1, there are five roles adopted by S/O techniques: resolving, evaluating, tuning, selecting or influencing *LDE*'s behavior. Those roles represent five distinct ways to handle *LDE*'s autonomy, going from the strong interventions proper of role (a), to the most indirect type of intervention displayed by approaches with role (e). The red-degraded arrow in Figure II-1 represents such spectrum of possibilities (from high autonomy (e) to low (a)).

With the first role ((a) in Figure II-1), the S/O resolves detailed FMS scheduling problems, for instance by generating complete or partial solutions that are subsequently executed by *LDE*s. This hierarchical intervention is often only effective under normal conditions, searching for near-optimal performance. Otherwise, it is expected that *LDE*s react using their own local decision-making algorithms (Leitão and Restivo, 2006). In the second role ((b) in Figure II-1), the *GDE* uses intelligently *LDE*s' processing capacities and their local decision-making algorithms. *LDE*s create proposals that are evaluated by the *GDE* using its S/O technique, since this latter can relate better those local plans in regards to the global objective.

As pointed out by Van Brussel et al. (1998), global performance in heterarchical control is extremely sensitive to the definition and fine-tuning of *LDE*'s decision-making algorithms, e.g., heuristics, artificial intelligence or dispatching/priority rules. More, it has been proven that static and pre-programmed algorithms can constraint the effectiveness of heterarchical control (Tay and Ho, 2008). Therefore, in the third and fourth roles ((c) and (d) in Figure II-1) the *GDE* uses S/O techniques to fine-tune the parameters of *LDE*'s decision-making algorithm or select which algorithms suits bests the current conditions. Hence, these two cases, the *GDE* is not directly involved in the *LDE*s' decision-making process but indirectly orchestrate their actions.

In the last category, ((e) in Figure II-1) the *GDE* influences *LDE*s' perception of the current situation by changing the values of environmental variables (e.g., traffic through a conveyor segment, state of an input buffer, traveling distance between two machines). The purpose of such indirect intervention is slowly taking the system towards the objective without applying drastic changes. *LDE*s have more autonomy to make their own decisions, keeping entirely their ability to react and adapt to changes. However, it is expected that the response time of these approaches is slower than approaches in the other categories.

In the following sections, optimization, simulation and simulation-based optimization approaches are reported and typified on the basis of the proposed framework.

3. Approaches based on optimization

Table II-1 presents different approaches based on optimization techniques, clearly identifying the type of control (i.e., FHFMS or SHFMS), the additional optimization technique, and its role, the underlying local control approach, control strategy, concerned myopic behavior and related reference.

Table II-1: Approaches based on optimization

Control architecture	Additional optimization technique	Optimization role	Local control approach	Control strategy	Concerned myopic behavior	Reference
FHFMS	Heuristic rule - Probability selection functions	Evaluate	MAS with contract-net protocol	Reactive	Social	(Boccalatte et al., 2004)
	Simulated annealing algorithm	Cooperate	MAS with currency-like mechanism	Reactive	Temporal	(Lim et al., 2009)
	Stigmergy-based approaches	Inform	MAS with negotiation protocol	Reactive	Social and temporal	(Valckenaers et al., 2006; Weyns et al., 2007)
	Heuristic rule based on intentions	Evaluate/ Inform	HMS with operation recursive division and delegation	Predictive-reactive	Social and temporal	(Zambrano Rey et al., 2012)
	Heuristic rule based on leveled commitment contracts	Cooperate	MAS with negotiation protocol	Reactive	Social and temporal	(Andersson and Sandholm, 2001)
	Multi-level contracts	Cooperate	HMS with contract net protocol	Reactive	Temporal	(Suesut et al., 2004)
SHFMS	Optimization algorithms	Resolve (complete)	HMS with self-organization	Predictive-reactive	Temporal and social	(Leitão and Restivo, 2006)
	Genetic algorithms	Resolve (complete)	MAS with negotiation protocol	Predictive-reactive	Social and temporal	(Wang et al., 2008)
	Mixed-integer linear program	Resolve (partial)	MAS with potential fields approach	Predictive-reactive	Social and temporal	(Zambrano Rey et al., 2011)
	Genetic algorithm	Evaluate	MAS with network representation and negotiation	Predictive-reactive	Social	(Li et al., 2010)
	Mathematical models from local information	Evaluate	MAS with negotiation protocol	Predictive-reactive	Social	(Nejad et al., 2011)
	Tabu Search	Evaluate	MAS with ant-colony optimization	Predictive-reactive	Social	(Böhnlein et al., 2011)
	Optimization techniques	Evaluate	MAS with negotiation protocol	Predictive-reactive	Social	(Heragu et al., 2002)
	Inductive decision tree	Tune up	MAS with negotiation protocol	Reactive	Social	(Zimmermann and Mönch, 2007)
	Evolutionary approach	Tune up	HMS with mixed-heuristic rules	Reactive	Social	(Walker et al., 2005)
	Scheduling policies	Influence	MAS with bidding mechanism	Reactive	Social	(Shen et al., 2000)
	Meta-heuristics	Influence	HMS with contract-net protocol	Predictive-reactive	Social and temporal	(Leitão, 2011; Leitão and Rodrigues, 2012)

MAS: Multi-agent systems   HMS: Holonic manufacturing system

3.1. Optimization approaches in fully heterarchical FMS control

Under a fully heterarchical architecture, local optimization is mostly based on basic and partial information (e.g., current status, current needs) issued from exchanged messages and proposals (i.e., negotiation, bids). With the purpose of better evaluating job proposals ((1) in Figure II-1) Boccalatte et al., (2004) proposed an extension of the Contract-Net basic protocol to update local schedules based on dynamic product priorities. A heuristic strategy based on a set of probability selection functions characterizes an agent's tendency to take into account other bidders and their objectives, reinforcing the *LDE*s' perception of the situation. In this case, agents (i.e., *LDE*s) are fully reactive and their social myopia is tackled using the generation of priorities, though they continue to be temporally myopic since negotiations only deal with current manufacturing operations.

For better cooperating, Lim et al., (2009) proposed to increase the solution-space exploration by introducing a currency-like mechanism hybridized with a simulated annealing (SA) algorithm ((2) in Figure II-1). The SA is called iteratively by job agents and aims to improve job-machine cooperation by adjusting currency values and minimizing production costs and lead times. More, machine agents forward job agent's bids in order to evaluate the plan as a whole. In this way, the job agent analyzes machines' bids and obtains larger decision horizon, reducing its temporal myopia. Nonetheless, the evident shortcoming of this strategy is that job agents have to record all the bids received and follow a heuristic rules to evaluate them (the first-in first-out rule in this case). Thus, the combinatorial explosion of the bid forwarding strategy may have an overwhelming effect on the job agent's tasks, slowing down the decision-making process.

Informing other entities aims at helping them to make more informed decisions ((3) in Figure II-1). Bio-inspired techniques bring interesting perspectives for achieving distributed optimization without sacrificing reactivity and adaptability (Barbosa et al., 2011). Certain bio-inspired mechanisms use pheromone trails to pass on information indirectly and without any compromise, respecting the principles of heterarchy. For instance, Valckenaers et al., (2006) and Weyns et al., (2007) used exploring and intention ant agents to create short-term forecasts and share local intentions through pheromones deposited in the environment. These ants go back and forth bringing updated information to the decider agent and depositing new intentions that emerge from those decisions. Heuristic rules are used to manage exploration and intentions, as well as local decision making. Likewise, Zambrano Rey et al., (2012) proposed a similar method sending representative *LDE*s to product service providers. Once these representatives are allocated they start working cooperatively to construct plans and inform others through intentions. These mobile representatives and the way they disperse out local information is a way to strengthen social bounds and reduce social myopia. Moreover, as these representatives work on different time horizons, they project decisions throughout the whole production sequence. In these approaches, the level of projection can be adjusted by determining a "traveling distance" tackling temporal myopia. The main advantage of these approaches is the distribution of control and the forecasting capacity achieved by the representatives, becoming an alternative method to simulation. Major issues are related to agent mobility, scalability issues, communication overhead and pheromone and intention modeling.

Other techniques to extend the regular immediate-future negotiation approaches are proposed by Andersson and Sandholm (2001) and Suesut et al. (2004). In the former, the authors studied algorithms for optimizing the contract itself and then propose de-commitment policies that expand the negotiation possibilities of each agent. Thus, agents become less socially and temporally myopic since they can explore and propagate other production

possibilities, even for sub-sequent operations. In the latter, multi-level contracts are used to enhance the agent look-ahead capabilities allowing them to better explore the long-term performance of each contract, avoiding a simple operation-based decision-making. Some potential drawbacks of extending contracts are the additional communication burden and non-negligible negotiation delays that might affect the *LDE*s' reactivity.

3.2. Optimization approaches in semi-heterarchical FMS control

Under the *resolving* role ((a) in Figure II-1), the *GDE* resolves completely or partially detailed FMS scheduling problems. In general, the *GDE* can, for example, host meta-heuristic approaches, such as genetic algorithms (Y. Wang et al., 2008) or mathematical models (Zambrano Rey et al., 2011b) to periodically generate optimized schedules that are proposed to *LDE*s which, despite their autonomy, follow these pre-defined plans. The level of social myopia can be defined by the size of the optimization problem, meaning the number of *LDE*s concerned (i.e., number of jobs, number of machines). In turn, the steadiness of conditions used by the *GDE* to run the global optimization determines the temporal validity of the solution. The most evident shortcoming of this type of approaches is the FMS scheduling problem complexity.

In order to speed up the *GDE* process, optimization can be used for evaluating *LDE*s' proposals and get more effective solutions, rather than generating a solution from scratch. The optimization agent (OA) proposed by Li et al. (2010) is a clear example of the *evaluating* role ((b) in Figure II-1). The OA evaluates plans created by job and machine agents with the aid of an evolutionary algorithm. The OA has a complete model of the shop floor (i.e., a flexible job-shop) that allows it to better explore the search space and get more effective solutions. Likewise, Nejad et al., (2011) introduced a coordination agent capable of making mathematical models according to information sent from the machine tool and product agents, and then it chooses the proposal that better meets the objective function. In the same way, Böhnlein et al., (2011) developed an ant-based MAS with AntTabu agents responsible for creating vehicle routing plans in a pre-optimization step. After all plans have been received, the AntTabu coordination agent performs a Tabu Search in order to post-optimize vehicle routing plans. The framework developed by Heragu et al. (2002) can also be included in this category. Middle-level and high-level global optimizer agents are integrated to analyze *LDE*s solutions and grant permission for their execution if those solutions meet global objectives. The *GDE* reduces *LDE*s' social myopia by running centralized optimization algorithms that orchestrate local proposals with global objectives. The main drawback of approaches in this category is that, since the supervisor evaluations are necessary during negotiation or decision-making, *LDE*'s reactivity is hardly compromised, under normal and

abnormal conditions. In addition, only current conditions are considered then the overall performance span is temporarily limited (temporal myopia).

The single parameterization agent proposed by Zimmermann and Mönch (2007) is an example of the *tuning* role ((c) in Figure II-1). In that work, the authors proposed optimization techniques to dynamically setup the decision rules (i.e., decision-making algorithms) hosted within *LDE*s. For instance, the maximum number of negotiation cycles in an auction-based protocol can be used as a parameterization attribute that influences agents' performances. Similarly, Walker et al., (2005) proposed a dynamic and responsive schedule by integrating traditional heuristic job-shop scheduling approaches and holonic manufacturing approaches. The *GDE*, called resource-scheduling dynamic mediator agent (RSDMA), coordinates local optimization for resource and order agents along with the global optimization that the work cell and FMS require. The *GDE* fine tunes the weights associated to the mixed heuristic rules used to create the schedules. Within this perspective, it is the decisional entity that evolves rather than the solution. The main advantage of the aforementioned techniques is that in any operating condition, the worst performance of the *LDE* is never lower than the performance obtained with the worst decision rule. In addition, the rule evolution is executed in parallel, thus the *GDE* does not, in theory, hold back the *LDE*'s decision making process. Conversely, in highly perturbed environments, it is highly probable that the *GDE* will have a hard time updating rule parameters, so *LDE*s risk using poor global performance rules.

Under the *influencing* role ((e) in Figure II-1), supervisors influence decisional entities through the environment, which in the case of MetaMorph (Shen et al., 2000) is defined by the virtual agent cluster dynamically created by mediator agents (i.e., *GDE*s). Intelligent agents find and cooperate with other agents through mediators, thus, the social myopia of an intelligent agent is dynamically adapted depending on the cluster to which the agent is assigned. *GDE*s can also influence *LDE*s' behaviors by adjusting environmental data or fine-tuning parameters used by the environmental optimization techniques. This concept was also introduced by Adam et al., (2009) as implicit control, in which the *GDE* influences *LDE*s externally without actually dictating them any specific orders or imposing any kind of rules. Leitão et al., (2012a) propose stigmergic approaches as a way to implement such implicit control. Another way in which the influencing role can be carried out is by using meta-heuristics to find patterns and analyze larger data set at the global level, aiming for giving some behavioral advices to the low level (Leitão, 2011; Leitão and Rodrigues, 2012). At the end, it is up to *LDE*s to decide whether or not such additional information must be taken into account for decision making. A possibly defect of those approaches is the necessary time *LDE*s and the entire FMS would take to reach the desired level of performance. Since *LDE*s

preserve a high level of autonomy, it is uncertain how they will react to those stimuli, being difficult to forecast their outcomes.

4. Approaches based on simulation

As briefly mentioned in Section I-4, the increased complexity of flexible manufacturing systems and decision-making algorithms, reduced computing costs, for both design and execution, and the improvements in simulation platforms have stimulated the use of simulation for manufacturing control (Habchi and Berchet, 2003). Simulation can be employed off-line and/or integrated in online control strategies (Leitão et al., 2012a; Cardin and Castagna, 2009). Off-line simulations can be used for solution validation, parameter sensitivity analysis and evaluation of robustness and performance, usually on the control design process (Jernigan et al., 1997; Smith, 2003). Conversely, online simulations are suited for anticipating deviations and prospectively analyzing multiple scenarios and strategies, before a decision is made (Pfeiffer et al., 2008; Monostori et al., 2010). In their study about the industrial applications of agent technology, Mařík and Lažanský, (2007) pointed out that simulation (and emulation) is the only way to predict the global behavior and detect patterns of highly autonomous systems exhibiting emergent behaviors, without the excessive cost of realistic experiments. Simulation can then support seamlessly the industrial adoption of heterarchical FMS control approaches.

Concerning myopic behavior, the purpose of using online simulation is twofold: first, simulations allow evaluating *LDE*s' interactions; hence the impact of a decision can be foreseen. Second, such evaluation can be extended to larger horizons, so mid- and long-term performance estimations are possible. The introduction of online simulation into heterarchical control approaches follows almost the same categories as in the optimization case, for FHFMS and SHFMS (Figure II-1). Table II-2 reports some approaches based on simulation, describing the type of simulation model, simulation role (according to Figure II-1), local control approach, control strategy, concerned myopic behavior and the work's reference.

4.1. Simulation approaches in fully heterarchical FMS control

In order to improve local decisions ((1) in Figure II-1), Papakostas et al., (2012) proposed that agents generate alternative solutions for each decision, and each alternative is evaluated by simulating its performance during a certain time span. Any decision is executed until its global effects are known, avoiding local optima. In the same way, Cardin and Castagna, (2009) placed the simulation tool into the staff holon, from which decisional holons request solution evaluation. The staff holon then helps basic holons to find a better local performance, ensuring also certain global coherence of their decisions. With the similar

purpose, Rolón and Martínez, (2012) proposed Gantt charts as the interaction mechanism in the @MES framework. Order and resource agents interact directly and indirectly through Gantt charts that are generated in a simulated scheduling world deployed with agent-based simulation. The Gantt chars depict the effect of a local decision, i.e., a manufacturing operation reservation, on the entire solution and for the complete order time span. One of the possible shortcomings of the precedent approaches is the requesting frequency of the simulation entity, especially if *LDE*s have to evaluate several alternatives, and for each alternative a simulation run is required. Then, the *LDE*'s decision-making is strongly dependant on the simulation model efficiency and the *LDE*'s decision-making algorithm.

Table II-2: Approaches based on simulation

Control architecture	Type of simulation	Simulation role	Local control approach	Control strategy	Concerned myopic behavior	Reference
FHFMS	ND	Evaluate	MAS with commitment protocol	Predictive-reactive	Social and temporal	(Papakostas et al., 2012)
	DES	Evaluate	HMS with negotiation protocol	Reactive	Social and temporal	(Cardin and Castagna, 2009)
	ABS	Evaluate	MAS with interaction mechanism	Predictive-reactive	Social and temporal	(Rolón and Martínez, 2012)
	DES	Evaluate	Look-ahead cooperative strategy	Reactive	Social and temporal	(Duffie and Prabhu, 1994)
	DES	Evaluate	Arrival-time control and priority rules	Predictive-reactive	Temporal	(Prabhu and Duffie, 1995)
	DES	Evaluate (off-line)	MAS with priority rules	Reactive	Social and Temporal	(Aissani et al., 2014)
	DES	Cooperate	MAS with auction-based approach	Predictive-reactive	Social and temporal	(Wang et al., 2013)
	ND	Evaluate/ Inform	Delegate MAS with bio-inspired mechanism	Reactive	Social and temporal	(Holvoet et al., 2009)
SHFMS	ND	Resolve	MAS with bidding process	Predictive-reactive	Social and temporal	(Wang and Lin, 2009)
	ABS	Resolve	MAS with auction-based negotiation	Predictive-reactive	Temporal	(Hodík et al., 2005)
	ND	Evaluate	Fractal Manufacturing	Predictive-reactive	Temporal	(Ryu and Jung, 2003)
	DES	Select	Dispatching rules	Reactive	Temporal	(O'Keefe and Rao, 1992)
	DES	Select	Dispatching rules	Reactive	Temporal	(Kim et al., 2012)
	ND	Select	Dispatching rules	Reactive	Temporal	(Metan et al., 2010)

ND: Not explicitly defined   DES: Discrete-event simulation   ABS: Agent-based modeling and simulation

Duffie and Prabhu, (1994) proposed a look-ahead cooperative scheduling strategy, in which tentative schedules are evaluated in time-scaled, distributed simulations using a replica of the real system. Simulations incorporate failures and other unexpected events into schedule evaluation, and once the simulation is complete, each virtual entity broadcasts the local merit of its proposal, so a global merit can be calculated and the best local plan can be executed. For most of these cases, temporal myopia is sensitive to the simulation horizon, defined as the extent of solution evaluation (e.g., number of operations in the job manufacturing sequence) and social myopia is related with the number of participating entities in the simulation model. (Prabhu and Duffie, 1995; Prabhu, 2003) also proposed a part-driven heterarchical manufacturing system in which the arrival time of each part is individually calculated using closed-loop controllers. In order to close the control loop, time-scaled

simulations are used to estimate part completion times, calculate the deviation with the part due date and evolve arrival times to a steady-state. Once arrival times are stabilized, parts are released into the shop floor (i.e., form one machine to job-shop like) and follow priority rules to allocate their manufacturing operations into appropriate resources. Temporal myopia is then reduced since simulations are run for the complete part production horizon, taking also into consideration part interactions.

Another way to improve the *LDE*'s decision-making is by allowing the entity to change its decision-making strategy according to current conditions. To do so, Aissani et al., (2014) developed a simulation study to generate a database related with different job-shop scheduling cases, so several rules can be automatically extracted. The extraction process is executed off-line but the set of rules is embedded into resource agents, which launch their cellular decisional module to choose the appropriate priority rule to execute. Since the chosen rule has proven its efficiency under normal conditions (or similar), entities are less myopic and make more informed decisions. The main limitation of this approach is the tedious work necessary to generate a consistent database based on the evaluation of a significant number of possible situations. It is also likely that *LDE*s' will use less adequate rules for non-record situations.

More focused on improving cooperation (role (2) in Figure II-1) to reach better decision-making, Wang et al., (2013) also proposed a control-based technique to develop a closed-loop feedback simulation (CLFS) approach. The CLFS included adaptive control of an auction-based bidding sequence in order to prevent the first-bid first-serve rule and dynamically allocate production resources to operations. The CLFS technique deals directly with myopic bids avoiding possible contradictions between local and global objectives. The CLFS iteratively adjusts the bidding sequence using the deviation between the predicted completion time issued from simulation and the due date fixed for each part. As approaches in Section II-3.1 the combinatorial explosion of subsequent bids is strongly dependant on FMS flexibility and the size of job sequences. Thus, since in this case subsequent bids are handled by tree exploration, the job agent can be affected by significant communication and decision costs.

The smart messages introduced by Holvoet et al. (2009) are autonomous and mobile messages that carry both, a behavior and state, and are an example of dissemination of local information (role (3) in Figure II-1). The behavior of a smart message is executed at every node, and determines how the entity will interact with others at that node, and decides on which node to move next. By doing so, smart messages aggregate information at every node and then they disseminate it in locations where such information is considered relevant. Smart messages trajectory choices are based on what-if symbiotic simulations on every node they pass. Simulation is then used as a technique to predict future system state and behavior,

reducing the smart messages temporal myopia. The coordination of several types of decisional entities (e.g., exploring, resource, product, etc.) with different behaviors is one of the main challenges of delegate-based methods.

4.2. Simulation approaches in semi-heterarchical FMS control

Generally speaking, in all these approaches, the *GDE* hosts a complete model of the FMS, being possible to obtain complete or partial solutions performing what-if simulations to explore more than one alternative solution ((a) in Figure II-1). For instance, Wang and Lin, (2009) integrated a simulation module employed to evaluate the feasibility of production and operations schedule in terms of a predetermined performance target. Scheduling agents request simulations before executing a solution, under normal and abnormal conditions. Instead of one single simulation entity, Hodík et al., (2005) introduced several simulation agents that model various properties of simulated resources such as production times or failure rates. The community of simulation agents emulates manufacturing resources (e.g., machines or human resources) and provides feedback to the planning system on the actual time spent on the plan realization. If the actual time differs from the plan, re-planning is executed. If on the first approach both dimensions of myopic behavior were addressed, the fact of having distributed and local simulators only reduces partially the resource agents' social myopia, since anyway each agent has partial results only corresponding to its local activities.

Proactive simulations are an interesting alternative for providing temporal visibility and conflict detection of local strategies proposed by *LDE*s ((b) in Figure II-1). In the fractal-based architecture of Ryu and Jung, (2003), an analyzer module first decides the best dispatching rule based on the status of fractals and the goal of the factory. Then, job profiles are scored using online simulations results and then, those job profiles are put together by resolver modules for execution. The proposed fractal model was implemented using multi-agent technology, thus a simulation agent performs such tasks using a centralized model. The main drawback of this approach is that each local solution has to be analyzed individually using a centralized model, possibly carrying out with high simulation costs.

Rather than analyzing each solution, look-ahead simulations can be used for evaluating the long-term performance of different myopic rules and choose the most appropriate one ((d) in Figure II-1) as proposed by O'Keefe and Rao (1992), Kim et al. (2012) and Metan et al., (2010). The selected rule is used by LDEs either during a pre-defined scheduling period or until an estimated performance value differs significantly (e.g., over a given limit) from the actual performance. In these cases, temporal myopia is not only related to the time period in which the rule is applied, but also the triggering policies for changing or adapting the rule. In

these approaches, the *GDE* is less intrusive because decision-making remains local. As mentioned before, the main drawback of these approaches is the possibly high cost of simulation, especially when different *LDE*s can host different myopic rules, given their needs and current situation.

5. Approaches based on simulation-based optimization

Recent technological advances in computing and the rapid evolution of simulation software support the use of online simulation in real-time manufacturing control (Habchi and Berchet, 2003; Fu et al., 2005; Cardin and Castagna, 2009). In fact, optimization methods have also been integrated in commercial simulation software as detailed by Hong and Nelson (2009). As explained by Klemmt et al. (2009), simulation-based optimization (SbO) can be obtained when online simulation and optimization algorithms are coupled in a closed-loop (Figure II-2).
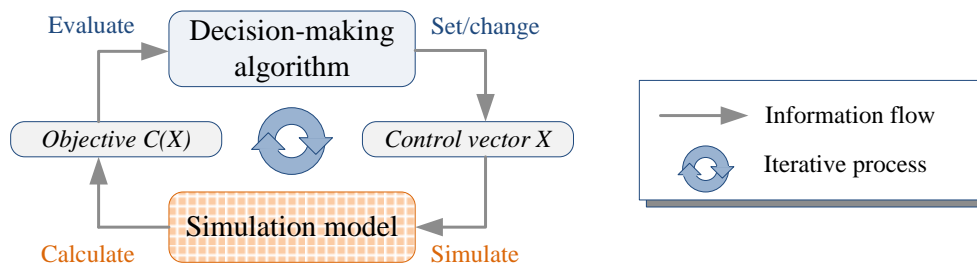
Figure II-2: Simulation-based optimization

A control vector X is modified by the optimization algorithm and introduced into the simulation model. In turn, the simulation model returns a value for the objective function $C(X)$, used by the optimization algorithm to evaluate the proposed solution. The model is simulated iteratively until a stopping criterion is met. Several optimization methods can be coupled with simulation, ranging from heuristics and meta-heuristics to operations research techniques (Weigert et al., 2006; Lemessi et al., 2011; Law and McComas, 2002; Tekin and Sabuncuoglu, 2004).

As seen in Table II-3, all approaches cited display semi-heterarchical architectures, given the centralized nature of the global optimization methods and the simulation models employed. Table II-3 reports, for each approach, the global optimization method, type of simulation, the SbO role as described in Section II-2, and the other aspects presented on precedent tables.

Chu et al., (2014) described a three-phase hybrid method integrating agent-based modeling and heuristic tree search to solve complex batch scheduling problems. In the first

phase the batch process is agent-based modeled, so in the second phase myopic agents can tackle the problem as a multi-stage decision problem by taking into consideration the constraints set up in the first phase. Each agent creates a tree of possible solutions using a negotiation protocol and evaluating each possibility with an agent-based simulation model. Phase three is executed at each branching point and controls the tree expansion by choosing the best solutions with a beam-search heuristic. Once each operation has been analyzed, it is assigned to a unit and executed immediately. By following that procedure, the simulation-optimization method resolves the scheduling problem by performing an intelligent exploration of the solution space, and taking into consideration the consequences of each decision (role (a) in Figure II-1). Hence, temporal myopia of agent decisions is reduced while social myopia still remains due to the local evaluation of each agent's decisions.

Table II-3: Approaches based on simulation-based optimization

Control architecture	Global optimization	Type of simulation	SbO role	Local control approach	Control strategy	Concerned myopic behavior	Reference
SHFMS	Beam search	ABS	Resolve (complete solution)	MAS with negotiation protocol	Predictive-reactive (iterative simulation runs)	Temporal	(Chu et al., 2014)
	Stigmergy for routing process	ABS	Resolve (partial solution)	Contract-net for allocation	Reactive (single simulation run)	Social	(Berger et al., 2010)
	Genetic algorithm	DES	Evaluate	MAS with bidding approach	Predictive-reactive (iterative simulation runs)	Social and temporal	(Zhang et al., 2007)
	Control theory heuristic	ABS	Evaluate	HMS with recursiveness and social factor	Predictive-reactive (iterative simulation runs)	Social and temporal	(Zambrano Rey et al., 2013)
	Iterative search method	ABS	Tune up	Priority rules	Reactive (parallel simulation runs)	Temporal	(Low et al., 2005)
	Genetic algorithm	DES	Tune up	Dispatching rules	Predictive-reactive (iterative simulation runs)	Temporal and social	(Maione and Naso, 2003)
	Pattern search	ND	Tune up/ Select	Dispatching rules	Predictive-reactive	Temporal and social	Kouiss et al., (1997)
	Genetic algorithm	DES	Select	Multi-attribute dispatching rules	Reactive (single simulation run)	Temporal and social	(Korytkowski et al., 2013; Geiger et al., 2006; Gaham et al., 2014)
	Neural network	DES	Select	Dispatching rules	Predictive-reactive (iterative simulation runs)	Temporal	(Mouelhi-Chibani and Pierreval, 2010)

ND: not explicitly defined   DES: Discrete-event simulation   ABS: Agent-based modeling and simulation

By dividing the FMS control problem into dynamic allocation and routing processes Berger et al., (2010) proposed a semi-heterarchical active-product-based control system in which partial solutions (role (a) in Figure II-1), only concerning the routing problem, are simulated in a virtual level. By following stigmergic principles, virtual active products make decisions in the virtual level that are afterwards downloaded to the physical level where physical active products make allocation decisions. By using travel history to update pheromones, virtual products share information with each other though the environment,

reducing their social myopic behavior. However, since the dynamic allocation process is executed on operation-basis, products only reason with a limited decision horizon, thus they continue to be temporarily myopic.

Instead of allocating the simulation functionalities externally, Zhang et al., (2007) proposed a simulation device allocated within the agent. Agents integrate their sub-models forming active clusters to create a more complete simulation model for evaluating their own decisions. An agent coordination algorithm, operating iteratively under the control of a genetic algorithm (GA), is developed to enable optimal planning and control, carried out through agent interactions. When satisfactory solutions cannot be found, subsystems are allowed to regroup to form new configurations which are evaluated using discrete-event simulations. For this example, simulation-optimization is used for *evaluating* agent configurations, so this approach can be classified in role (b) as described in Section II-2.

Another example of this category is the holonic framework proposed by Zambrano Rey et al. (2013). By using recursive divisions of the product holon, adjunct holons are created to generate complete plans for the product holon, avoiding temporarily myopic decisions. Each product holon hosts a control theory loop that generates release times, which are afterwards used by adjunct holons as the input parameter for plan generation. In order to improve cooperation and reduce social myopia, product holons calculate locally a social factor that changes their role in the organization from altruist to competition. Thus, product holons are more aware of other holons' needs and become able to balance local and global objectives, depending on their current status. The predictive phase is carried out using agent-based simulation supported in the control layer and when a solution is found, the operational layer of each holon is in charge of executing it, providing some sort of granularity.

Since myopic priority rules are very dynamic, rule fine-tuning (role (c) in Figure II-1) is a way to improve their efficiency. Low et al. (2005) described a symbiotic simulation system that employed software agents for monitoring, optimizing and controlling a semiconductor assembly and test operation. The local control is executed distributed by priority rules, tuned up by a control agent that sets lower and upper thresholds. The optimization agent (OA) carries out a simulation-based optimization to decide the best approach for handling a given situation. The OA creates a number of models with threshold combinations and distributes those models to simulation agents to carry out "what-if" analysis. Likewise, G. Maione and Naso, (2003) applied a genetic algorithm to adapt the decision strategy of autonomous part entities within a MAS-based heterarchical control structure. Part agents use a set of pre-assigned weighted decision rules and obey a weight adaptation policy as a reaction to unforeseen perturbations. In the latter, although a GDE is not defined, the genetic algorithm

is executed online and centralized. The length of the simulated scenario determines the evaluation horizon; hence it defines the temporal myopia of agent decisions.

But probably, one of the most used applications of simulation-optimization methods has been the analysis and selection (role (d) in Figure II-1) of myopic dispatching/priority rules (Frantzén et al., 2011). For instance, Kouiss et al., (1997) used pattern search to detect symptoms and change the low level agents behavior by for example selecting a particular dispatching rule that performs better under current conditions. Similarly, Korytkowski et al., (2013) developed an evolutionary simulation-based heuristic to construct near-optimal solutions for dispatching rule (DR) allocation. Different multi-attribute DRs may be assigned to each workstation by means of a genetic algorithm where the sequence of DRs is encoded into a chromosome. The fitness function is calculated as a mean value obtained from running a set of replications of discrete-event simulation runs. Similarly, Geiger et al. (2006) and Gaham et al. (2014) approached the problem of finding new DRs by combining simulation and genetic algorithm techniques. In order to reduce the cost of rule selection, Mouelhi-Chibani and Pierreval (2010) proposed a new approach based on neural networks for selecting the most suited DR in accordance with the current system state and the FMS operating conditions. The rule selection is executed on real time because the neural network does not need a training set and instead, its parameters are determined through simulation-optimization.

6. Synthesis and discussion

To synthesize, Figure II-3 positions the aforementioned approaches regarding three major aspects: the decision-making algorithm complexity, overall performance, understood as the balance between long-term global performance and reactivity; and a rough judgment of the level of myopia. The blue curve assumes that by increasing the algorithm complexity and global visibility of the controlled system, myopic behavior can be reduced. This could be true up to some point but after that there is no guarantee of that because of the required short-delay responses, which limits the amount of time for information gathering and analysis. Precisely, the red curve represents the assumption that global performance can be improved by enhancing the decision-making algorithm until it becomes hard to find a balance between reactivity and global performance; thus a good overall performance.

Fully heterarchical FMS control approaches remain in the low complexity area in Figure II-3, but because of their unpredictable behavior and myopic behavior, long-term optimality is not necessarily ensured. Global performance is then partially improved by introducing additional S/O techniques. However, as mentioned by Leitão et al., (2009) in their analysis of Simon's assumptions, intelligent decisional entities have bounded rationality due to their

finite computation and communication capabilities. Introducing S/O techniques require more communication effort and computation efficiency. Indeed, the overall system performance is improved compared to basic FHFMS control approaches, yet there will still be no guarantee of that.
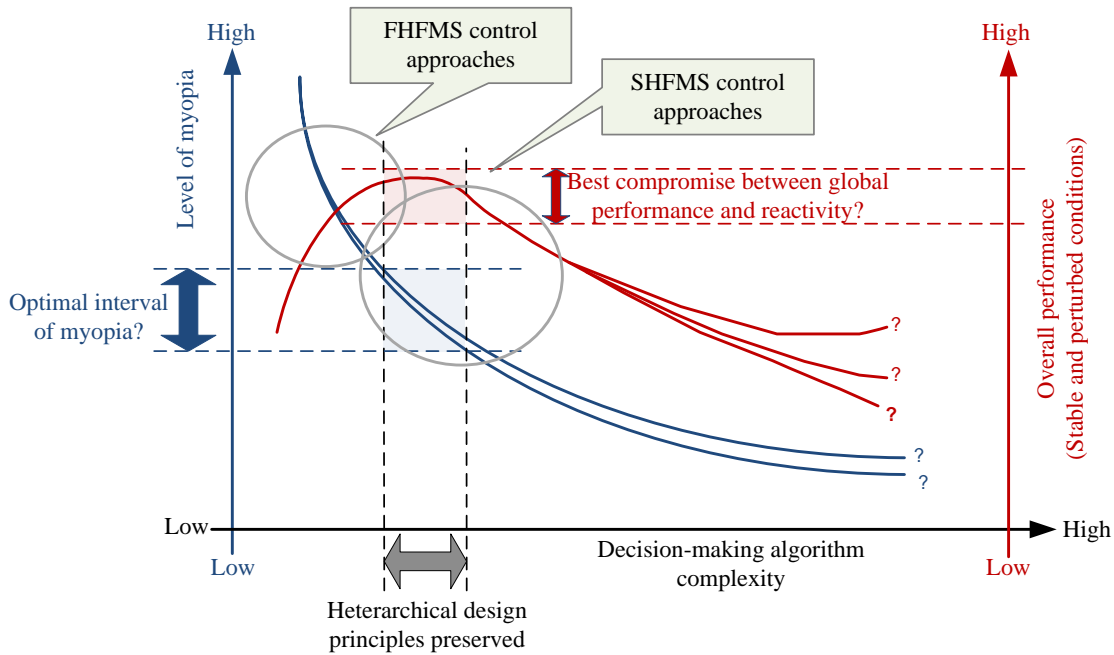
Figure II-3: Myopic behavior vs. decision-making algorithm complexity and overall performance

By introducing hierarchical interventions, better long-term optimality and predictability can be achieved; consequently myopic behavior can be reduced. In SHFMS control approaches the problem may turn to centralization and possible architectural rigidity that undeniably penalize important features achieved by heterarchy. The main issue with centralization lies on the S/O techniques that have been conceived for solving the entire detailed scheduling problem as if the system was centralized, overshadowing *LDE*s' control skills and processing capabilities. Therefore, the *GDE* usability becomes highly dependable on the combinatorial nature of the FMS problem, resulting in higher algorithm complexity translated into loss of reactivity and adaptability (Jeong and Kim, 1998; Baykasoğlu and Özbakır, 2010).

In order to avoid architectural rigidity, some works proposed a dynamic switch between hierarchy and heterarchy trying to balance global performance and reactivity (Pach, 2013). From that, it is possible to conclude that in those dynamic control architectures there is a preliminary effort to actually control myopia, by passing from a "less myopic" architecture to a "more myopic" architecture, taking into account the system's conditions. In order to actually control myopic behavior based on a desired overall performance, first it would be

necessary to establish a target level of myopia (or interval as seen in Figure II-3), then be capable of measuring its current level and finally be able to adjust local decisions to achieve the target myopia level and global objective. This is not an easy task because it is still not clear if having an optimal myopia level or interval means reaching an optimal overall performance (Figure II-3). More, it will be necessary to associate myopia measuring metrics to metrics used to determine system performance. In spite of its clear benefits concerning global performance and predictability, the switching mechanism and its required self-organization still need further research to avoid chaotic behavior and nervousness (Hadeli et al., 2006; Barbosa et al., 2012).

Another possibility to eliminate the architectural rigidity is by applying self-organizing principles allowing the possibility to completely change the structure of architecture (Leitão and Restivo, 2006; José Barbosa et al., 2013; Jose Barbosa et al., 2013). In addition to the internal adaptation of each decisional entity, the whole society of decisional entities is reconfigured to deal more effectively with perturbations. Hence, myopic behavior takes an additional dimension since not only local decisions are concerned by also architectural decisions need also to be made. Decisions regarding which entities should remain active, which entity should step out, and how to avoid continuous architectural changes that exacerbate system nervousness are some of the decisions that can be affected by myopia.

Several conclusions can be inferred from this literature review. First and foremost, in all reported works, S/O techniques have been introduced to enhance global performance. Meanwhile, despite the fact that they are all concerned with myopic behavior, none of them explicitly handle myopia, thus myopia is reduced indirectly. Second, as pointed out by Leitão and Restivo, (2006), to obtain a reactive, adaptable and fault-tolerant architecture, *"it is necessary to be as decentralized as possible and as centralized as necessary"*. Third, as synthesized in the proposed generic framework, the hierarchical interactions are not restricted to only provide solutions for the entire FMS scheduling problem, but they can adopt different roles from which different $GDE - LDE$ relationship schemes can be conceived.

From these conclusions, the following requirements can be highlighted:

✓ There is a need of control approaches that explicitly deal with myopic behavior, in which the benefits provided by heterarchy are preserved, notably the reactivity and adaptability. An improvement on global system performance should be then a consequence of reducing or controlling myopic behavior.

✓ Reducing or controlling myopic behavior should be done from a granular perspective, identifying particular myopic decisions and introducing different techniques for each one of them, so the control system can remain modular and re-configurable. In addition, that

identification will determine if myopic behavior should be reduced totally or partially, and the benefits and implications of the choice.

✓ In order to maintain an easy adaptation and evolution of HFMS control, combination and dynamic adaptation of the S/O techniques should also be possible. This would clearly ease the migration of current hierarchical approaches because current decision-making algorithms can possibly be re-used in conjunction with HFMS principles.

✓ Different kind of entity interactions should also be supported. A dynamic HFMS control should be capable of reducing myopic behaviors through different kind of entity interactions, accommodating different degrees of *LDE*'s autonomy. Again, this would allow a gradual migration from current rigid hierarchical to more heterarchical control approaches.

✓ Tackling myopic behavior should also bring other benefits, such as guaranteeing and predicting certain operational performance at all times. This is one of the main drawbacks of switching architectures because during perturbation handling, the system becomes fully myopic and possibly unpredictable. Hence, highly dynamic and adaptable semi-heterarchical approaches where centralization and decentralization coexists may lead to more stable approaches.

Some leads on accomplishing those functional requirements can be found in the works reviewed in this chapter. For instance, the fact that FMS control sub-problems can be addressed separately (S. Wang et al., 2008; Berger et al., 2010), make possible to have different techniques to deal with myopic decisions within each sub-problem. This FMS control problem decomposition is possible because of the logical sequence of FMS control decisions (e.g., a task needs to be chosen first, then a machine can be selected, and then a route to get to the destination machine can be selected). Thus, the introduction of simulation-based optimization into HFMS control seems to bring interesting benefits to deal with the social and temporal dimensions of those myopic decisions. On the optimization side, meta-heuristics, especially evolutionary algorithms have been widely used for simulation-based optimization. The main reason is that this type of algorithms allows addressing problems with mixed numerical and non-numerical variables (Pierreval and Paris, 2000). On the simulation side, given that the complexity of real FMS problems is too high to be solved by usual analytical or enumeration methods (Fu et al., 2005; Weigert et al., 2006), simulation becomes the only tool to depict and predict entity interactions within HFMS control. The use of agent-based simulation models has become more popular to this purpose.

It also seems interesting that, in regards to the proposed generic framework, the combination of simulation and optimization techniques may allow different roles and entity

interactions within semi-heterarchical control architectures. The resulting semi-heterarchical control architecture can certainly support the desired functional aspects. Thus, all these concepts are taken into consideration and translated into modeling features for the proposed semi-heterarchical FMS control approach described in the next chapter.

Chapter III. A Semi-Heterarchical Simulation-based Optimization Approach to Reduce Myopic Behavior in FMS

1. Introduction

In order to accomplish the requirements described in the previous chapter, a semi-heterarchical FMS (SHFMS) control approach is presented in this chapter. Therefore, we part from the premise that, for now, myopic behavior can be reduced with an architectural approach. This approximation allows us to be generic and give us the possibility of defining several instantiations of the proposed approach in regards to myopic behavior reduction and the target FMS. A key concept of our approach is the insertion of simulation-based optimization techniques into a global decisional level as a tool to reduce the impact of myopic control decisions made at the local decisional level.

This chapter starts by describing in Section III-2 the general features of the proposed approach. Then, the proposed SHFMS control architecture is described in Section III-3. Since the core of the SHFMS is the decisional entity, Section III-4 starts by describing a generic decisional entity's structure that is afterwards instantiated into decisional entities composing the proposed SHFMS control architecture. Then, the control strategy under normal and abnormal conditions is explained in Section III-5. As it will be seen through this chapter, there are various decisions necessary to instantiate and implement the propose semi-heterarchical architecture. Therefore, a possible procedure that can be helpful to realize the FMS control architecture is described in Section III-6. A final synthesis and assessment of the main attributes of the proposed approach is offered at the end of this chapter in Section III-7.

2. General features of the proposed approach

As concluded in the previous chapter, up to know most of research in heterarchical-based FMS (HFMS) control (i.e., fully heterarchical or semi-heterarchical FMS control) has focused on improving global performance, implicitly reducing myopic behavior. Our perspective goes on an alternative direction and the proposed approach relies on the combination of structural and functional features to explicitly reduce myopic behavior, hence achieving better global performance. The main objective to do so is to preserve some

important features of HFMS control, such as reactivity, adaptability and fault tolerance. In regards to the requirements listed in Section II-6, the general features of the proposed approach are:

✓ *"Explicitly deal with myopic behavior, aiming to preserve reactivity and adaptability"*. Our approach proposes to reduce myopic behavior by establishing a semi-heterarchy composed of global and local decisional levels, in which global and local decisional entities reside respectively. The global level's objective is to reduce myopic behavior of the local level, within a combined control strategy.

✓ *"Reducing or controlling myopic behavior should be done from a granular perspective"*. To this end, the FMS control problem is decoupled into sub-problems for which myopic behavior is treated individually. The purpose of this particularization is to accept configurations in which myopic behavior for some sub-problems is reduced while for other sub-problems is accepted. In regards to the previous requirement, this combined control strategy aims to preserve certain level of reactivity, adaptability and tolerance to perturbations.

✓ "*To maintain an easy adaptation and evolution of HFMS control, combination and dynamic adaptation of the S/O techniques should be possible*". Our approach proposes the integration of simulation-based optimization (SbO) techniques in the global decisional level to deal with different myopic decisions. Different SbO techniques can be used for each myopic decision, maintaining a modular design of the global decisional entity. On the local decisional level, local decisional entities are endowed with low complex decision-making algorithms for all control sub-problems.

✓ *"Different kind of entity interactions should also be supported"*. The aforementioned modular designed of the global decisional level is exploited by allowing the coexistence of a variety of roles for the SbO techniques, e.g., resolving, evaluating, tuning, selecting or influencing, as explained in Section II-2. Issued from these roles, the proposed approach tries to be as generic as possible by supporting different types of hierarchical interactions between global and local decisional entities. Therefore, as it will be further explained later on, different types of entity interactions (i.e., interactions between global and local entities) can be conceived, even for the same SbO role with the same SbO technique. One of the main benefits of this feature is the possibility to accommodate different degrees of local entities' autonomy, which should also be dynamically adjusted in the presence of abnormal conditions.

✓ "*Other benefits, such as guaranteeing and predicting certain operational performance at all times, should also be withdrawn while dealing with myopic behavior*". This

requirement is achieved by integrating simulation and maintaining myopic behavior reduced for at least some control sub-problems. Simulation allows the global level to have an estimation of global performance and be able to predict undesirable events resulting from local decisions, such as deadlocks. Only a priori feasible control decisions should be allowed for execution. If the simulation model reflects the local entities' behavior, it is probable to foresee such events.

In the following, these general features are described as part of the architecture, the internal decisional entities structure and the control strategy that drives the dynamic behavior of the proposed approach.

3. Description of the SHFMS architecture

The proposed semi-heterarchical FMS control (SHFMS) architecture is composed of two levels: a global decisional level (GDL) and a local decisional level (LDL) as depicted in Figure III-1. Each local decisional entity (LDE) is assigned to control an actuator (A), e.g., a product or a resource within the FMS. The proposed architecture has no restrictions on the modeling approach, e.g., multi-agent systems (MAS), holonic manufacturing systems (HMS), Product-driven (PD), used for implementation (some of these modeling approaches are described in Appendix B).

3.1. The local decisional level

The *local decisional level (LDL)* is formed by a set of local decisional entities ($LDEs$), such that $LDL = \{LDE_1, \dots LDE_j, \dots, LDE_J\}$ as seen in Figure III-1. Each LDE_j is assigned to control one of the actuators, $A_j \; \forall j \in \{1, \dots J\}$, such as products, or FMS resources, e.g., material-handling systems, machines, storage units. The LDL follows a set of local objective functions (LOF) so each LDE_j can be assigned with one those objectives $LO_j \in LOF = \{LO_1, \dots LO_j, \dots, LO_J\}$ as depicted in Figure III-1. Each LDE_j can host one or several decision-making algorithms to achieve the local objective (LO_j), providing local control decisions to the assigned control sub-problems. Other activities assigned to each LDE_j consist in calculating its local performance (LP_j) in regards to the local objective and survey the local status (LS_j) of the controlled actuator.
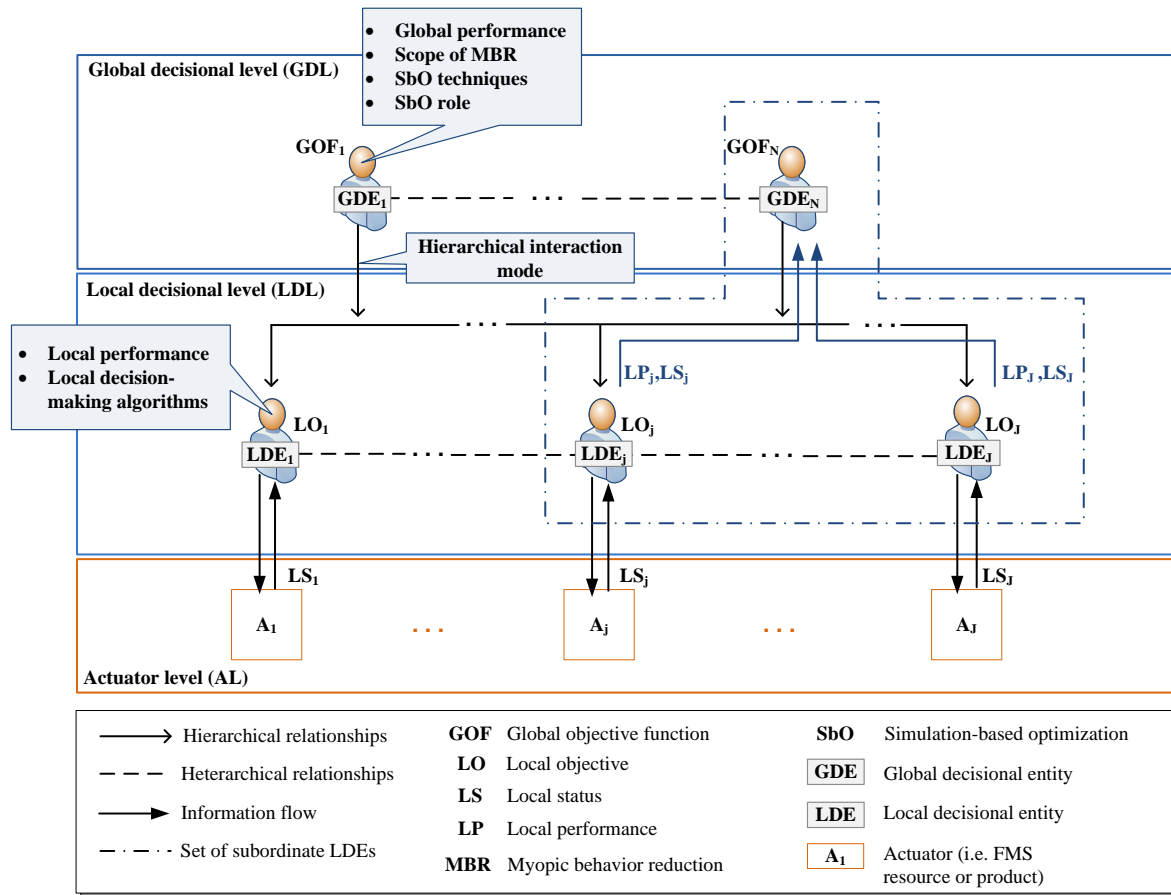
Figure III-1: Proposed semi-heterarchical FMS control architecture

3.2. The global decisional level

In the more general sense, the ***global decisional level (GDL)*** is composed of one or more global decisional entities ($GDEs$) such that $GDL = \{GDE_1, \ldots GDE_n, \ldots, GDE_N\}$, as seen in Figure III-1. The GDL works on a set of global objective functions (GOF), then each GDE_n can be assigned to one global objective function $GOF_n \in GOF = \{GOF_1, \ldots GOF_n, \ldots, GOF_N\}$ (examples of manufacturing objective functions can be found in Appendix A-II). Each GDE_n is hierarchically related to a set of subordinate LDEs as depicted in Figure III-1. For instance, one GDE can handle product LDEs and another GDE resource LDEs (i.e., machines, material handling, storage, etc.), each one with its own objectives. The hierarchical interaction mode is meant to reduce the myopic behavior of subordinate LDEs. Therefore each GDE_n can survey local performances (LP) of subordinate LDEs, making it possible to calculate the actual global performance associated to its assigned global objective.

The proposed GDL configuration aims to be generic by allowing the possibility of targeting multiple objectives. In that case, GDEs will have to cooperate with each other and coordinate to manage subordinate LDEs. This heterarchical configuration at the global level

allows managing strongly correlated FMS problems to yield more realistic industrial cases. For instance, manufacturing, maintenance and inventory, among other problems, can be managed by independent GDEs working together under objectives such as production rates, production and storage costs, maintenance costs, energy consumption, etc. Our motivation to propose such distribution at this level is twofold. First, different FMS problems handle different types of variables (e.g., release times that are continuous, machine selection that are discrete, maintenance times that are continuous, etc.) and have to deal with different constraints. Hence, myopic behavior for each problem can be handled individually by each GDE_n. Second, by proposing a heterarchical configuration at the GDL, complex models resulting from problem aggregation are avoided, improving the reactivity of each GDE_n. Indeed, myopic decisions related to FMS problems other than manufacturing are an interesting research topic envisaged within future works (see Conclusions and Further Work). Henceforth, only FMS control problems related to manufacturing will be taken into consideration.

3.3. $GDE - LDE$ hierarchical interaction modes

Our approach parts from the premise that it is not advantageous to eliminate myopic behavior because it is an essential behavior inherited from heterarchy and necessary to obtain desirable features such as reactivity, adaptability and fault tolerance. Given that the FMS control problem can be decoupled into control sub-problems (e.g., operation and product sequencing, product routing, machine selection as mentioned in Section II-6) myopic behavior for each of those sub-problems can be treated individually. Our approach proposes a limited scope of myopic behavior reduction (MBR) as described in Figure III-2. If the FMS control problem associated to GDE_n (denoted at CP_n) can be decoupled into K control sub-problems (csp), such that $CP_n = \{csp_n^1, ..., csp_n^k, ... csp_n^K\}$, the GDE_n will only focus on those k control sub-problems that are more sensitive to myopic control decisions and contribute the most to global performance. The other $K - k$ control sub-problems are dealt locally by LDEs', which have full autonomy for those control decisions. Given this shared control strategy, simulation is at the outmost important to provide a global performance estimation based on the proposals made by the optimization techniques and the local decision making algorithms (out of MBR scope).

With the purpose of reducing myopic behavior individually for each control sub-problem, the GDE_n is configured modularly so different optimization techniques (Opt-technique in Figure III-2) can be implemented to deal with each control sub-problem. The integration of optimization techniques and a simulation model can result in different SbO roles (e.g., *resolving, evaluating, tuning, selecting or influencing* as described in Section II-2), allowing the possibility of dealing with each control sub-problem differently. One of the possible

criteria to decide which control sub-problems should be dealt and which optimization technique and SbO role can be used, could be the computational complexity of the control sub-problem. The FMS control sub-problem complexity results from the level of the associated types of flexibility, FMS size and production volume. Another criteria can be the FMS state and conditions. If the FMS has over capacity or is saturated, reducing myopic behavior may not result in any major changes in regards to global performance.
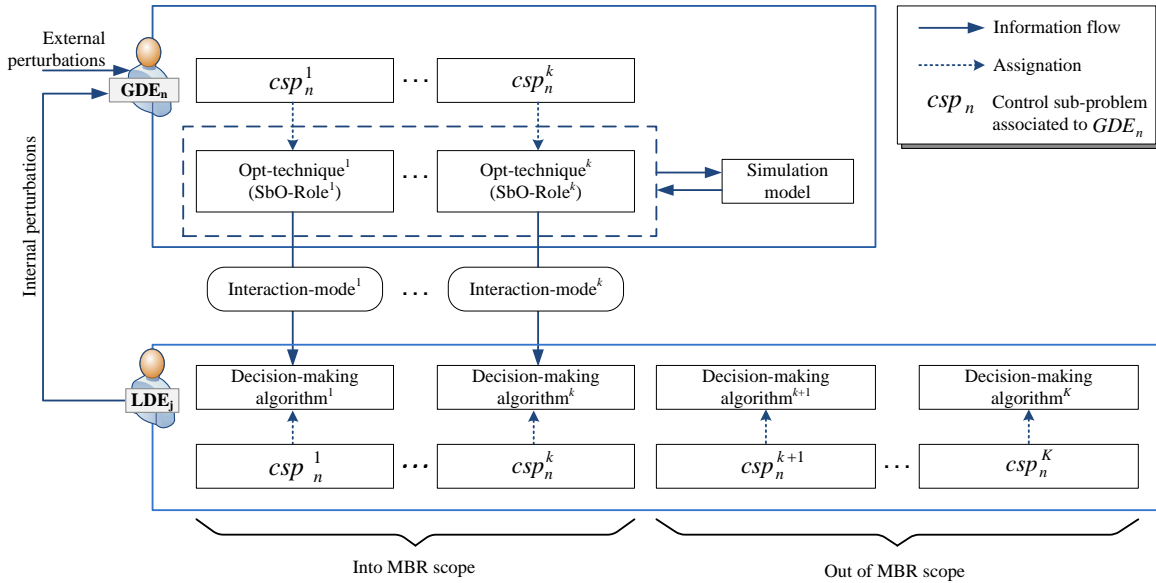
Figure III-2: Scope of myopic behavior reduction

As seen in Figure III-2, each GDE_n can be configured to work on k control sub-problems (csp), therefore each GDE_n handles the set of global decision variables GDV_n where $GDV_n = \{GDV_n^1, \dots, GDV_n^k\}$. The purpose of each GDE_n is to find the set of values $GDV_n^* = \{GDV_n^{1*}, \dots, GDV_n^{k*}\}$ that meet its global objective function GOF_n, under normal and abnormal conditions. The optimization technique that handles each control sub-problem works on the associated global decision variable. This one-to-one assignment is not a restriction but a practical configuration to handle different types of variables (e.g., numerical, continuous or discrete, or non-numerical), reduce problem complexity and provide better modularity and adaptability. For instance, the product release problem can be modeled with continuous variables, i.e., a release date; discrete variables, i.e., product release sequence or with non-numerical variables such as choosing one dispatching rule.

The GDE_n interacts with each LDE_j by taking the values GDV_n^* and sending them to the corresponding decision-making algorithm in the LDE_j. According to the chosen SbO role, there can be three possible hierarchical interaction modes. These hierarchical interaction modes determine the way in the value GDV_n^* is exploited by the local decision-making

algorithm in the LDE_j. GDV_n^* can for instance be assumed as the value or possible values for LDV_j (where LDV_j is the set of local decisional variable managed by local decision-making algorithms), or as a parameter of the decision-making algorithm. Three $GDE - LDE$ hierarchical interactions modes are proposed: *coercive, limitary* and *steering*. These interaction modes are described below and some examples are also given to explain them better.

✓ **Coercive interaction**: the definition of "coerce" fully explains this interaction mode. According to the Collins dictionary ("Collins Dictionary. http://www.collinsdictionary.com/ (visited 07/04/2014)," 2014) coerce means "*to compel or restrain by force or authority without regard to individual wishes or desire*". Indeed, this interaction mode overrules LDE's autonomy for the concerned control sub-problem. Though $LDEs$ can use their local decision-making algorithms to make their own decisions, it is strongly advised to assume the value GDV_n^* obtained by the SbO technique as local control decisions (LDV_j^*). This is particularly true if the SbO takes the resolving role. Other alternative solutions will affect the global dynamics, possibly affecting global performance.

   o *Example 1 - resolving role*: if the GDE_n is configured to reduce myopic machine allocations, then the SbO technique must provide the sequence of machines that each product needs to follow in order to accomplish their operation sequence. The GDE_n sends to each subordinate product LDE_j the machine sequences and this latter has to follow the imposed sequences unless a perturbation is detected. In this case, the GDV_n^* is a matrix in which, for instance, each column represents a product, each row an operation, and the selected machine is placed in the intersection of rows and columns.

✓ **Limitary interaction**: this interaction mode does not overrules completely $LDEs$' autonomy because GDV_n^* is a set of values that are adopted as boundaries by $LDEs$. Thus, $LDEs$ can use their local decision-making algorithms to make their own decisions to find LDV_j^*, but within the values provided by the associated SbO. *Resolving* and *evaluating* SbO roles can be used for such interaction mode. Remaining within the boundaries guarantee that performance will also remain bounded, if FMS conditions used to make those solutions are maintained. Two examples of this interaction mode are explained below:

   o *Example 2 - resolving role*: if the GDE_n is configured to reduce myopic machine allocations and the SbO converges to a population of possible solutions, i.e., machine sequences for each product; each product has the possibility to chose

from a set of machines for each manufacturing operation. Then, each LDE_j will assess the FMS current conditions and make a decision within the possibilities found by the SbO technique.

- o *Example 3 - evaluating role*: if the GDE_n is configured to reduce myopic routing, and each LDE_j makes several routing proposals, then the GDE_n can cluster a set of proposals with similar estimated performances and each LDE_j will only need to chose among one of them, taking into account the current traffic conditions.

✓ **Steering interaction**: is the less intrusive interaction mode since GDV_k^* is not used as a solution but as a parameter or a policy for the local decision-making algorithm (Carvalho et al., 2012). This interaction mode takes advantage of the LDEs' capabilities to gather and analyze data. LDEs' actions are in turn interpreted by the GDE_n, which may redirect LDEs' behavior by tuning up the parameters or sending a new policy. Therefore, each GDE_n steers the local decisions by adapting the LDEs' decision-making algorithm according to current FMS conditions. This interaction mode is suitable if the SbO technique takes the *tuning* or *selecting* roles. Two examples of this interaction mode are described below:

- o *Example 4 - tuning role*: if the GDE_n is configured to reduce myopic routing and the control decision to select a route results from a weighted sum of two variables, route load and distance to the destination, such that $selected\ route = max(\alpha * load_i + \beta * distance_i)$, then $GDV_n^k = \{\alpha, \beta\}$ and $LDV_n^k = \{load, distance\}$.

- o *Example 5 - selecting role*: if the GDE_n is configured to reduce myopic machine allocations and each LDE_j is endowed with a set of priority rules for making that choice, then the GDE_n can dynamically evaluate which of those priority rules suits best the current FMS conditions.

It is important to point out that the *influencing* SbO role has not been included in the aforementioned interaction modes since under the *influencing* role, both decisional entities (GDE_n and LDE_j) do not interact directly but indirectly through the environment. For instance, in approaches based on stigmergy (Hadeli et al., 2004; Sallez et al., 2009), pheromone values left in the environment can be treated as GDV_n in order to change LDEs' perception of their environment.

As seen in Figure III-2, the proposed approach has no restriction of combining various interaction modes and SbO roles in order to grant LDEs with different autonomy degrees for each control sub-problem. This is an interesting feature since autonomy is not anymore

defined for the entire entity but for each control sub-problem that the local decisional entity handles.

4. Decisional entities structure

To realize the proposed SHFMS control architecture it is fundamental to define and develop the building block of the control architecture: the decisional entity (DE) (Jung et al., 1996). The configuration of the DE can be inspired of any of the control units proposed by modeling approaches such as multi-agent systems (Shen et al., 2000), bionic manufacturing systems (Okino, 1993), holonic manufacturing systems (Christensen, 1994) or intelligent products (Sallez et al., 2010). In general, the aforementioned DE models have three internal modules in common: a control, an interaction and an information storage module. Since the internal architecture of an holon proposed by Babiceanu and Chen, (2006) contains those basic modules, herein, such configuration is adopted as generic DE and it is depicted in Figure III-3.

The core of the DE is its control module in which decision making resides. The information module stores, among others, the parameters required for decision making, tasks supported, performance metrics and other data needed to achieve the assigned objective. In turn, the interaction module supports decision making by allowing information exchange between decisional entities, the environment or the actuator as in the case of LDEs (Figure III-1). The global and local decisional entities constituting the proposed SHFMS are instances of the generic DE. The internal structure of the LDE and GDE are explained below.
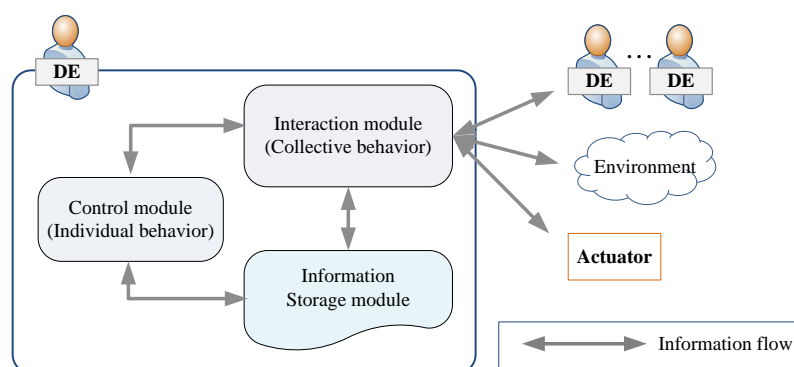
Figure III-3: The generic decisional entity (Babiceanu and Chen, 2006)

4.1. The local decisional entity (LDE)

Figure III-4 shows the internal configuration of an LDE. Each LDE_j is capable of handling a set of local decision variables $LDV_j = \{LDV_j^1, \dots, LDV_j^K\}$. The purpose of each

LDE_j is to find the set of values LDV_j^* that optimizes a local objective LO_j, taking into consideration the current status LS_j of its controlled actuator and its local performance LP_j.

4.1.1. The control module

The control module of LDE_j is divided into two sub-modules: high-level and low-level control sub-modules as shown in Figure III-4.
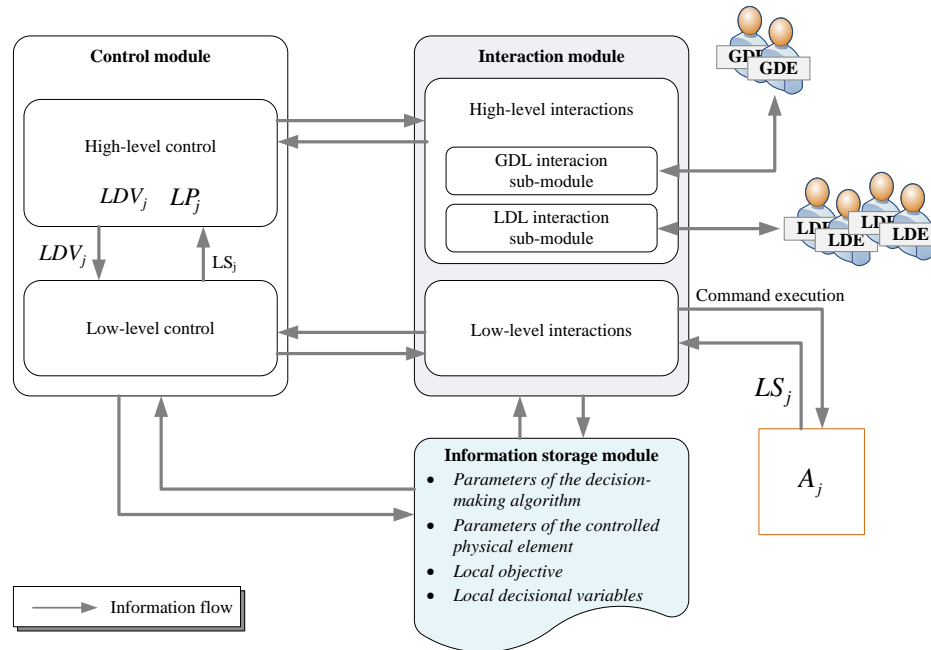
Figure III-4: The *LDE*s structure

This division of control is inspired by the different natures of both control processes as described in (Zambrano Rey et al., 2013). Though in the low-level decision-making can be executed periodically on high frequency basis, the high-level control is event-oriented. The high-level control sub-module hosts decision-making algorithms for dealing with the assigned control sub-problems, to achieve the entity's local objectives. In turn, the low-level sub-module is in charge of translating the high-level control decisions (LDV_j^*) into low-level commands that the controlled actuator (A_j) can execute. For example, if an *LDE* controls a material-handling component, e.g., an automated guided vehicle, product routing decisions are taken in the high-level control sub-module, and then those decisions are translated by the low-level control into moving commands (e.g. forwards, backwards). Also, the low-level control monitors the actual position of the vehicle, i.e., local status (LS_j), and informs the high-level control when the task has been accomplished or if an abnormal condition appears.

4.1.2. The interaction and information storage modules

The interaction module is also divided into high-level and low-level interaction sub-modules to handle data related to their corresponding control sub-modules. The high-level interaction sub-module discriminates between interactions with global and local decisional entities. Although each LDE_j has the capabilities to locally resolve the assigned FMS control sub-problems, in order to reduce its myopic behavior each LDE_j establishes a two-way interaction with the global decisional level by accepting global decisions GDV_n^* and returning local information such as its local performance LP_j and local status LS_j. High-level interactions with other LDEs are performed with the purpose of collecting the required information to execute the local decision-making algorithms and accomplish the tasks assigned. To do so, each LDE_j cooperates with other LDEs by exchanging a partial or a complete set of temporary values ldv_j (e.g., proposals, bids, etc.) where $ldv_j \subseteq LDV_j$; or their final decisions $ldv_j^* \subseteq LDV_j^*$ to inform others about their intentions or future actions. The fact that each LDE_j may only exchange a partial set of its decision variables (ldv_j) is one the causes of social myopia. The other cause of social myopia is more an internal issue of each LDE_j related to the exploration of possible alternatives for LDV_j, meaning a limitation of the local decision-making algorithms. Both, the high-level and low-level interaction sub-module implements the necessary protocols and technological requirements to ensure data exchange with other decisional entities and the controlled A_j.

Last, the information storage module contains all configuration parameters of the local decision-making algorithms (e.g., thresholds), parameters associated to the controlled A_j, the local decision variables (LDV_j) and their current values, as well as the local objective function (LO_j).

4.2. The global decisional entity (GDE)

Figure III-5 shows the internal configuration of each GDE_n. The control, interaction and information storage modules are described below.

4.2.1. The control module

As depicted in Figure III-5, in order to reduce the LDE's myopic behavior, in the control module of each GDE_n, simulation-based optimization techniques are coupled in a closed loop as proposed in (Zambrano Rey et al., 2014). Each GDE_n is configured to work only on k control sub-problems which are functions of GDV_n (① in Figure III-5). Those k control sub-problems are resolved by a simulation-based optimization loop, in which optimization techniques propose values for GDV_n (② in Figure III-5) and time-scaled simulations (③ in Figure III-5) evaluate the impact of GDV_n on LDEs' behavior. Simulation outputs (④ in

Figure III-5) are necessary to calculate an estimated global performance, EGP_n (⑤ in Figure III-5) from estimated local performances, ELP_j (in Figure III-5). The simulation-based optimization loop continues to explore alternative values for GDV_n until stopping criteria are met (⑥ in Figure III-5). For instance, the SbO loop can be stopped after a certain deadline, elapsed time or lack of solution improvement during a certain time.
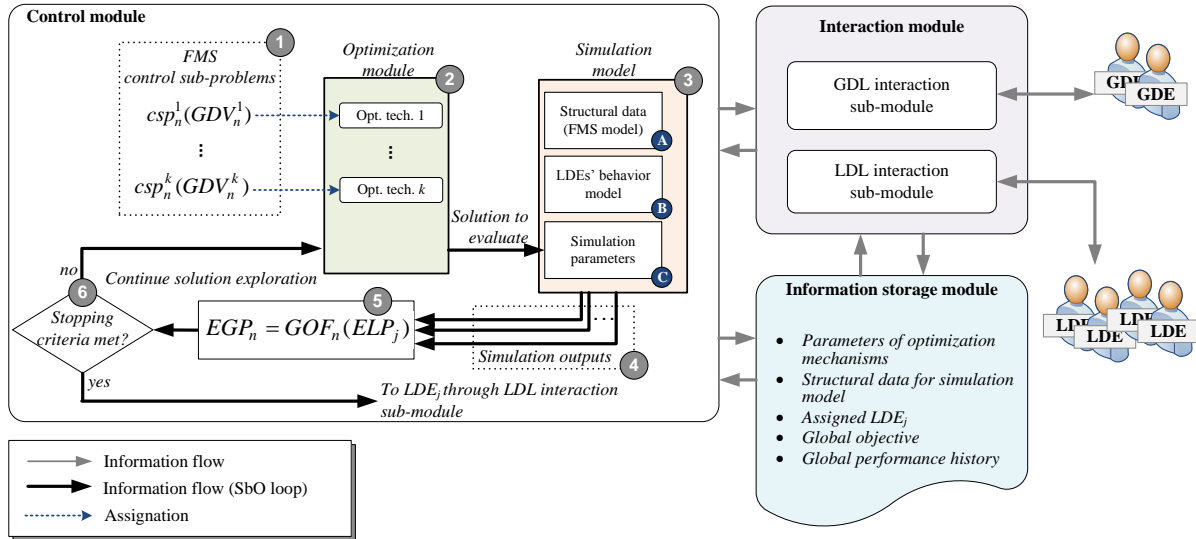
Figure III-5: The GDE_n structure

The use of simulation-based optimization is justified by the following benefits. First and foremost, the optimization mechanism allows an intelligent exploration of the solution space for GDV_n, taking also into consideration the LDEs' behavior. Indeed, it is the simulation model that allows integrating LDEs' behavior into GDE's solutions. Second, the simulation model provides an estimated global performance for the current and future states of the system, making possible to reduce the temporal myopia of local decisions. Third, simulation allows estimating the impact of local decisions of each LDE_j in regards to other LDEs, then the impact of their social myopic decisions can be reflected on the estimated local performances. Last, the EGP_n can be used as lower or upper bounds to monitor production execution and trigger recalculations if necessary. Additional description on the optimization module and the simulation model are provided below.

The optimization module

The proposed optimization module (② in Figure III-5) is characterized by a modular configuration in which different optimization techniques can be hosted. One important aspect of the optimization module is the type of connection between optimization mechanisms, which can be of two types: decoupled or coupled. In decoupled connection, each optimization technique works on each problem individually, creating successive SbO loops, as depicted in Figure III-6. Therefore, for each control sub-problem the value for the corresponding GDV_n^k

is fixed and passed to the next SbO loop as a parameter. For instance, in the kth SbO cycle, the set of values $GDV_n^{q*} \; \forall q \in \{1, \dots, (k-1)\}$ have already been fixed by the precedent (*k-1*) optimization techniques, the GDV_n^k variable is under evaluation and the set of (*K-k*) control sub-problems for which myopic behavior will not be reduced are dealt locally by *LDE*s using their local decision-making algorithms. As a result, global performance estimation is obtained through simulation replications and feedback to the optimization technique to drive the search process (Figure III-6).
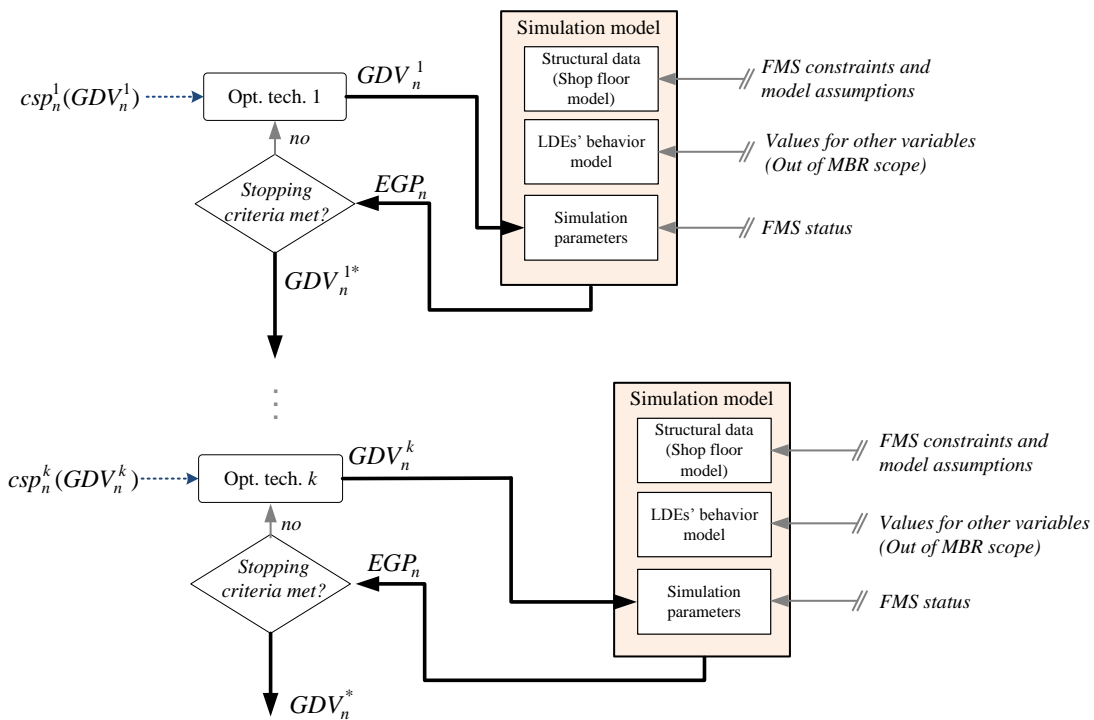
Figure III-6: Decoupled connection between optimization techniques

This type of connection aims to gradually obtained solutions for those control sub-problems assigned to the optimization module. More, since each control sub-problem is dealt separately, it is also expected that computational complexity would be reduced. Furthermore, any changes in the control sub-problem variables and constraints can be easily adopted in the concerned optimization technique without affecting others. Last, the GDE_n's modularity and adaptability is also preserved since one optimization technique can be easily replaced to maintain each GDE_n up to date.

Contrary to the decoupled type, in the coupled connection all optimization mechanisms work integrated possibly embedded into each other, sending a complete set of values for GDV_n to the simulation model (Figure III-7).

As in the previous case, all those (*K-k*) control sub-problems not assigned to optimization techniques are dealt locally by *LDE*s' decision-making algorithms (out of MBR scope). The main advantage of this configuration is that all decision variables are explored at the same time, so at any point in time a value for the entire set of GDV_n can be used to reduce *LDE*s' myopic behavior. However, coupling optimization techniques may entail a larger solution search space, possibly incurring in more computation costs that in the decoupled configuration. More, the configuration of each optimization technique needs to take into consideration the other techniques so they can actually work together. Furthermore, it is necessary to evaluate not only the efficiency of optimization techniques, individually (if possible), decoupled and coupled so the most appropriate techniques and connection type can be used.
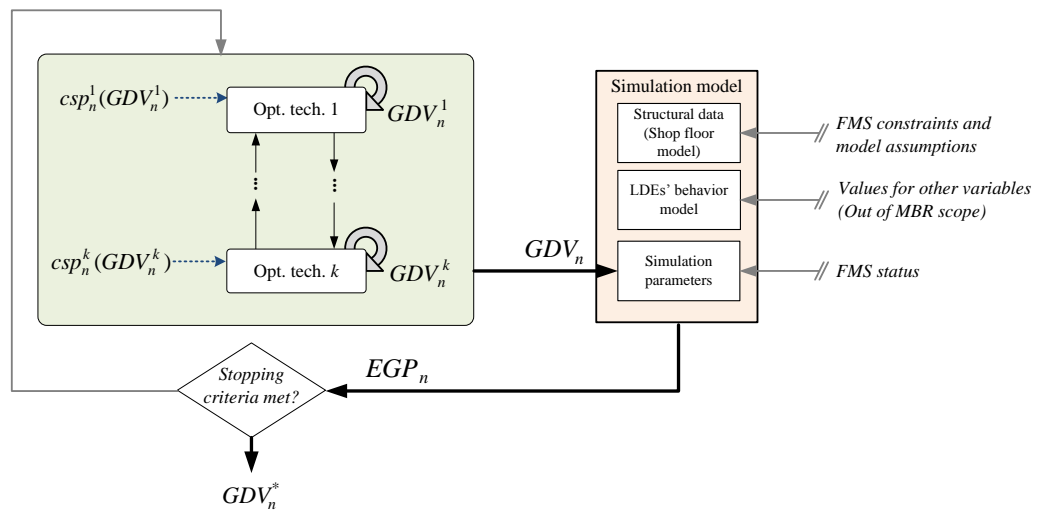
Figure III-7: Coupled connection between optimization techniques

The simulation model

Simulation has become a key tool for evaluating complex systems with complex relationships between entities, such as those driven by heterarchical approaches. Simulation outputs (④ in Figure III-5) not only help moving a solution towards finding a good or near-optimal value (inputs of the optimization mechanisms), but also help to describe *LDE*s interactions. In the proposed approach, simulation is of the outmost importance because it allows predicting how values of GDV_n affect *LDE*'s local control decisions and an estimated global performance can be calculated based on the *LDE*'s behavior.

The simulation model is mainly composed of three data types: the structural data, the simulation parameters and the *LDE*s' behavior model. The structural data (Ⓐ in Figure III-5) describes the FMS layout and its components. This type of information rarely changes and it

is the result of the FMS design process. In addition, some model assumptions, managerial and technical constraints are also used to structure the simulation model.

The *LDE*s' behavior model (Ⓑ in Figure III-5) contains the *LDE*s' decision-making algorithms, making possible to analyze *LDE*s' interactions and estimate how *LDE*s deal with FMS control sub-problems. It is important that this module reflects the *LDE*s' behavior resulting from their high autonomy, especially for those problems for which myopic behavior is not reduced. If the *GDE* takes into consideration the *LDE*s' behavior, *GDE*'s estimations will be more coherent with the actual *LDE*s' behavior during solution execution in the FMS.

Simulation parameters (Ⓒ in Figure III-5) congregate data that needs to be specified for each simulation replication. Simulation parameters consist in two types of input data: the first type is the solution to evaluate which correspond to the outputs of the optimization mechanisms (i.e., values for GDV_n). The second comes from current FMS status which enables adaptation in the presence of internal perturbations. The dynamic nature of the FMS (i.e., machine breakdowns, machine activation/deactivation, machine maintenance, etc.) is constantly monitored to update the simulation model and obtain valid solutions.

4.2.2. The interaction and information storage modules

The interaction module is composed of two sub-modules, one for cooperating with other decisional entities in the *GDL* and another for interacting with subordinate *LDEs*. Through the *GDL* sub-module, cooperation between *GDEs* is achieved by exchanging temporary values of GDV_n (for instance in the case of negotiation where proposals are exchanged) or GDV_n^* when a final decision has been made.

Through the *LDL* interaction sub-module, each GDE_n can hierarchically interact with subordinate *LDE*s, depending on the SbO roles and the interaction mode as described in Section III-3.3. Incoming information from *LDE*s (i.e., local performance and local status of the controlled actuator as shown in Figure III-1) allows each GDE_n to calculate deviations between the estimated and actual local performance for each subordinate LDE_j. More, reported status of the controlled actuator is needed to update parameters of the optimization mechanisms and simulation model. In the information storage, the assignation of each GDE_n to a particular global objective (GOF_n), subordinae *LDEs*, control sub-problems and global decisional variables (csp_n, GDV_n), corresponding parameters of the optimization mechanisms and incoming information from *LDE*s are stored and maintained up to date as shown in Figure III-5.

5. The control strategy

As defined in Section II-2, the control strategy determines the control system dynamics under normal conditions and in the occurrence of internal and external perturbations. Between fully reactive, predictive-reactive and proactive (Appendix D), our hypothesis is that the predictive-reactive strategy supports best the proposed SbO roles and hierarchical interaction modes. This section first covers the control strategy under normal conditions and then under abnormal conditions.

5.1. Control strategy under normal conditions

Figure III-8 depicts in detail how entities in the proposed architecture handle reactivity and global performance as described in Figure I-4.

On the basis of the predictive-reactive control strategy, the global decisional entity works in the predictive phase and the local decisional entity takes care of the reactive phase. The purpose of the control strategy is to maintain the global performance within a target profile established around a desired level of global performance. In the predictive phase, the GDE_n triggers its optimization mechanisms for exploring the solution space for the k control sub-problems assigned to the global decisional entity (global solution search in Figure III-8). Each alternative solution is composed of temporary values for GDV_n related to those control sub-problems. The simulation model is then in charge of evaluating those alternative solutions for a time horizon Th (solution evaluation in Figure III-8). It is then expected that after a certain number of SbO cycles, a set of values GDV_n^* satisfying the target performance profile are found. In this way, an estimated performance of the selected solution can be calculated. As shown in Figure III-8, certain global performance variability is anticipated and accepted given that the expected global performance remains within bounds. Such variability is mainly due to LDEs' full autonomy to deal with (K-k) control sub-problems. Since myopic behavior for those control sub-problems is not reduced, those myopic decisions appear during simulation replications and are strongly dependant on GDV_n values under evaluation. Model assumptions concerning the optimization mechanisms and simulation model may also contribute to such variability.

Once the GDE_n obtains GDV_n^*, these values are adopted by LDEs using one of the three interaction modes explained in Section III-3.3. Except for the *coercive interaction* mode in which GDV_n^* values are adopted as control solutions for the k control sub-problems, for other interaction modes LDEs have to gather the necessary information to find alternative solutions for all control sub-problems (local solution search Figure III-8). In such cases, the local objective (LO_j) is used to evaluate alternative solutions and select one of those alternatives for execution (solution evaluation and execution in Figure III-8). The reactive phase is

launched each time a control decision has to be made, since the local solution search is made for short decision horizons, contrary to the global solution search process in the predictive phase. As shown in Figure III-8 it is most likely that a deviation exists between estimated and actual performances due to modeling and data assumption, as well as local control decisions. It is important to have in mind that under normal conditions, the reactive phase is triggered to make decisions that are out of the MBR scope and not only to face a perturbation.
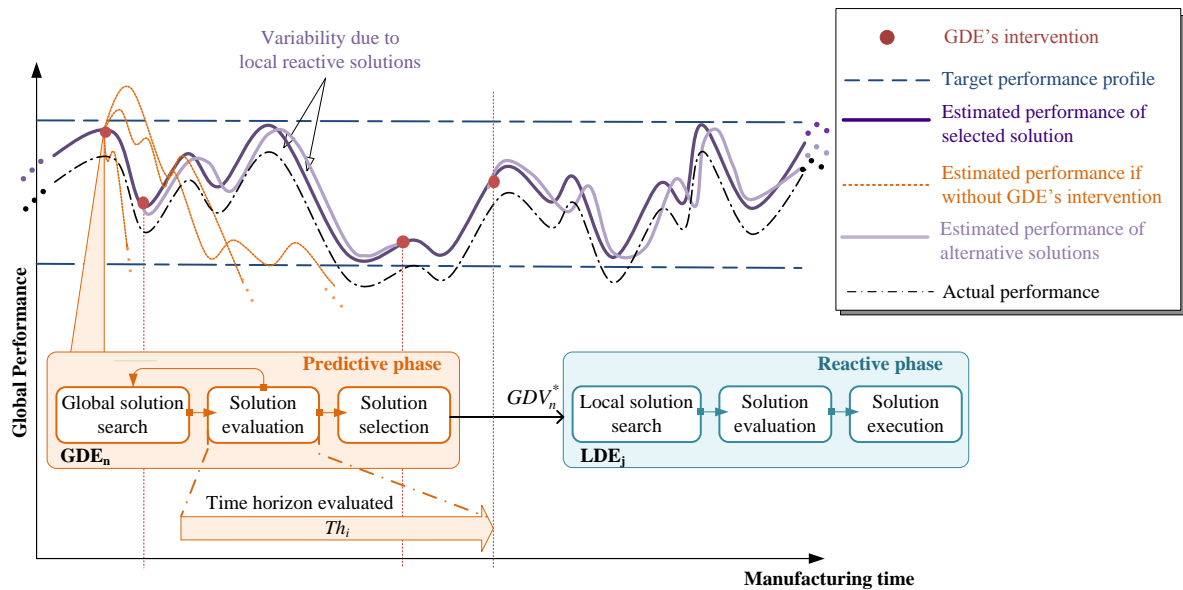
Figure III-8: Control strategy under normal conditions

5.2. Control strategy under perturbed conditions

The control strategy under perturbed conditions is categorized as those within *GDE*'s domain and those within *LDE*'s domain:

Perturbation handling within LDE's domain: these perturbations concern those control sub-problems for which myopic behavior is not reduced. Therefore, all perturbations in this category are handled locally by each LDE_j given that this latter has full autonomy for making the necessary control decisions. Each LDE_j must be capable of indentifying the perturbation and decide how to act. As depicted in Figure III-9, the reactive phase comprises local solution search, alternative solution evaluation and solution execution. Since the *LDE* is not guided to make such control decisions, the *LDE* makes myopic decisions given its reduced view of the entire FMS and the restricted amount of time it has to come up with a solution. In addition to the expected variability caused by myopic control decisions under normal conditions (forecasted during the predictive phase), myopic decisions to overcome internal perturbations may cause additional variations and deviations of the expected global performance. As

depicted in Figure III-9, since no simulation is carried out, it is not possible to estimate global performance, only the actual global performance monitored by the GDE will tell if GDE's intervention is necessary or not to maintain the system within the target profile.

Perturbation handling within GDE's domain: these perturbations concern those control sub-problems for which myopic behavior is reduced. In such cases, the GDE launches SbO cycles to calculate new GDV_n^* values. Two main features of the proposed approach are crucial for handling perturbed conditions within the GDE_n's domain. First and foremost, the different SbO roles and the proposed interaction modes allow several strategies. While *coercive* and *limitary* modes require recalculation at the global decisional level, the *steering* interaction does not require that LDEs wait for GDE_n's solutions. It is then possible to think in adaptive configurations in which an interaction mode is used under normal conditions and another mode is triggered for perturbed conditions. This adaptive strategy joint with the fact that interaction cooperation modes can be particular for each control sub-problem makes the proposed approach generic and highly adaptable. The main requirement would be to have multiple optimization mechanisms for the same control sub-problem or optimization mechanisms that allow multiple interaction modes.
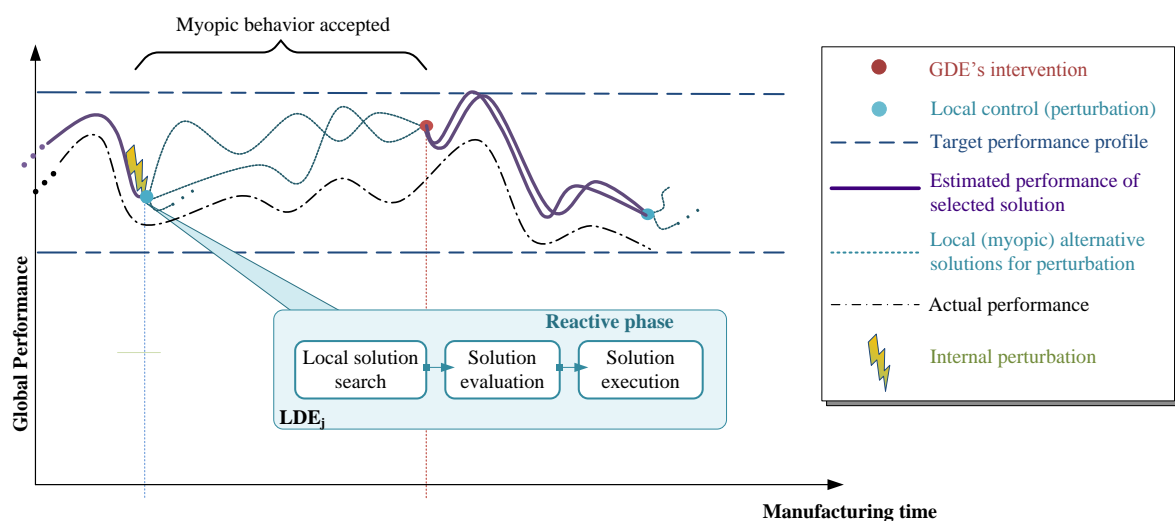
Figure III-9: Perturbation handling within LDE's domain

The second feature is issued from the modular configuration of the GDE's optimization module. In order to be more reactive, the optimization module can be re-configured to use only those optimization mechanisms that are concerned by the perturbation. In such case, the decoupled connection between optimization mechanisms would be the most suitable configuration to achieve such re-configuration. An adaptive strategy between the two

connection types would make this approach more reactive and can be envisaged in future works.

6. Procedure to instantiate the proposed approach

There are various design decisions to make in order to instantiate and implement the propose semi-heterarchical architecture. Figure III-10 depicts a possible procedure that can be helpful to realize the FMS control architecture.

The first stage is to gather all data necessary to build a FMS model. Data related to FMS resources, supported products, manufacturing operations, FMS layout, and production volume can be considered as static and dynamic parameters for the model. Technical and managerial constraints are also necessary to limit the model scope and allow its implementation. At the end of this stage, the global performance indicator(s) and the FMS control sub-problems issued from the model must be identified.

The second stage starts by taking those control sub-problems and allocating them into decisional entities in the local and global levels to achieve a general description of the architecture. To this end, global and local decisional entities participating in the architecture can be identified at this stage. By defining those entities, the interaction mode between *GDE*s and *LDE*s for each control sub-problem can be determined. Then, in the third and fourth stages, the internal structure of local and global decisional entities can be configured. Either these two stages are done in parallel, for instance for each control sub-problem, or sequentially. After defining the local decisional level, it should be possible to carry out an experimental study to assess the efficiency of the local decisional level. If a centralized version of the FMS model is achievable, then it should be possible to establish the deviation between results obtained with the *LDL* and optimal or near-optimal values obtained with the FMS model (e.g, linear or non-linear programming, meta-heuristics) for static and dynamic cases. Static studies aiming to determine the global performance under normal conditions can be accomplished through simulation or real experiments if an FMS is available. In turn, dynamic experimentations should be carried out to determine the architecture's reactivity, adaptability, and fault tolerance under perturbed conditions. At this time, myopic decisions and their impact on global performance can be assessed, becoming a critical point that may define and/or introduce changes in the configuration of the global decisional level, the interaction modes and the following stages of this procedure.

The next stage is focused on explaining the dynamic interactions between decisional entities, during normal and abnormal conditions (i.e., the control strategy). For these latter, it is then helpful to define processes and entities in charge of responding to different situations.

Last, the global evaluation stage looks for assessing the efficiency of the architecture in terms of the global performance indicator. As for the partial evaluation, simulation studies and real experimentations can be executed. In order to prove myopic behavior reduction, results obtained from partial and global evaluations should be compared. For both experimental studies, benchmark cases such as the ones proposed by Cavalieri et al., (2000), Wörner and Wörn, (2006) and Trentesaux et al., (2013) can be followed. On the basis of results analysis, new information, desired functionalities or changes in the FMS model, and the necessary adjustments and changes at each stage should be made aiming to keep the architecture up to date.
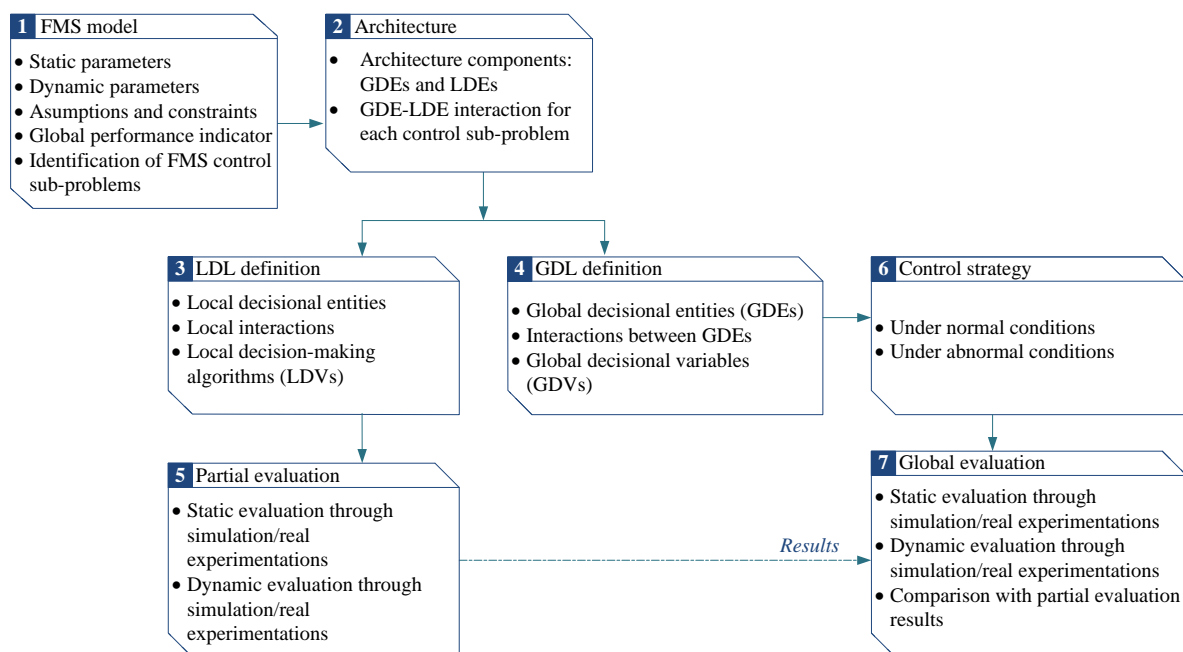
Figure III-10: Procedure to instantiate the proposed approach

7. Synthesis of the proposed approach

Now that all the constituent elements of the proposed approach have been described, herein the main features and pertinence of this proposition are highlighted. As expressed in the introduction of this chapter, our vision to reduce myopic behavior has an architectural perspective, with specific structural and functional features. Based on the concepts proposed herein, a detailed study on myopic behavior at the local decisional level, aiming for mathematical-based models that allows us to actually measure and control myopic behavior will be addressed in future works (see Conclusions and Further Work).

The first structural element and the main feature of the proposed approach is the configuration of a global decisional entity endowed with simulation-based optimization techniques. The main objective of such entity is to focus on control sub-problems that required the most attention according to the impact of those myopic control decisions on global performance. This division of control is a novelty in regards to similar semi-heterarchical architectures in which the global decisional level is redundant (Y. Wang et al., 2008; Li et al., 2010) from the control perspective to the local decisional level. In our approach, it is possible to accept myopic behavior as part of the control strategy because it allows rapid control responses given the limited information and temporal assessment needed to make those decisions.

The second structural element concerns the internal configuration of the global and local decisional entities. For the global decisional entity, the configuration of the optimization module avoids aggregation of control decisions into one optimization mechanism, which is one of the most common features found in semi-heterarchical approaches reported in Chapter II (Heragu et al., 2002; Walker et al., 2005; Nejad et al., 2011). In those cases, *GDE*'s interventions strongly depend on highly complex optimization mechanisms. In addition, the modular configuration of the optimization module makes possible to adapt optimization mechanisms and their connectivity to face changeable conditions. Such adaptation favors reactivity and adaptability of the whole architecture, always paying close attention to myopic behavior. Regarding the local decisional entity, the internal structure of the *LDE* differentiates between operational and strategic decisions, and high and low-level interactions making implementation more straightforward.

The third structural element is the integration of a simulation model. Simulation is necessary for *GDEs* to account for and assess *LDE*'s myopic control decisions. Simulation allows *GDEs* to estimate global performance out of solutions proposed by the optimization module and solutions proposed by *LDE*'s. Consequently, simulation plays an important role in predicting future states of the system and detecting issues such as deadlocks before they actually arise. The main different with SbO approaches reported in Section II-5 is the joint work between the optimization module and simulation model. In other works, such as the ones proposed by Low et al. (2005) and Zhang et al. (2007), the simulation model is only used for evaluating the outcomes of the solutions proposed by the optimization mechanism.

Functionally, the control strategy is based on the different roles the simulation-based optimization technique may have in regards to myopic behavior. Those roles allowed us to define three interaction modes between *GDL* and *LDL*: *coercive, limitary or steering*. These interaction modes state clearly how to integrate *GDE*'s results into *LDE*'s local decision-making algorithms to reduce myopic decisions. More, these interaction modes do not impose

any restriction on the type of local decision-making algorithms, yielding generic our myopic behavior reduction strategy. The second functional feature refers to the *LDE*s' autonomy. Since *LDE*s have a set of problems for which they have full autonomy, they can easily react to deal with perturbations affecting those control problems. On the contrary, if perturbations reach *GDE*s control sub-problems, these latter are called to react. As a result, reactivity is achieved at both levels. In most of semi-heterarchical approaches described in Chapter II, the global level is call to intervene every time an event perturbs the current plan, or it is up to the local level to do so (Shen et al., 2000; Leitão et al., 2005; Chu et al., 2014).

Following the procedure explained in the previous section, an instance of the proposed approach is described in the following chapter. Then, in Chapter V, a case study will serve for evaluating the concepts introduced herein.

Chapter IV. Reducing Myopic Behavior in a Flexible Manufacturing System: A Case Study

1. Introduction

This chapter aims at describing the applicability of the simulation-based optimization approach proposed in the previous chapter to reduce the myopic behavior when the FMS is controlled by a semi-heterarchical approach. The big dilemma encountered when controlling flexible manufacturing systems (FMSs) is to optimize a certain performance criteria by efficiently utilizing its flexibilities while maintaining reactivity to internal and external perturbations. In this chapter, the semi-heterarchical approach proposed previously was instantiated to control a particular type of FMS. The local decisional level (*LDL*) provides local reactivity but due to its myopic behavior, a global decisional level (*GDL*) was configured to reduce such myopic behavior, striking a balance between the ability to react to disturbances and global performance.

The choices on the different structural elements aimed to maintain modular, adaptable and low complex control architecture are explained in this chapter. To this end, this chapter follows the procedure explained in Section III-6. At first, the FMS control problem, its parameters, assumptions and constraints taken into account are described in Section IV-2. Also in this section, two global performance indicators are explained. In Section IV-3, the entire architecture is described as well as the chosen interaction modes between local and global decisional entities for the control sub-problems for which myopic behavior tries to be reduced. With the functional description of the local and global decisional levels, Section IV-4 explains in detail the internal configuration of local and global decisional entities. Section IV-5 deals with the control strategy and the global evaluation of the resulting architecture is explained in Chapter V. A synthesis of the chapter is presented in Section IV-6.

2. Description of the FMS control problem

This section describes the FMS scheduling model and the FMS control sub-problems issued from those scheduling decisions.

2.1. FMS scheduling model

The following FMS scheduling model provides the main parameters, assumptions and constraints that can afterwards be employed in this particular case study. This model was inspired from the flexible job-shop scheduling (FJSS) model used as reference by Trentesaux et al., (2013) in their benchmarking. Other FJSS models reported in literature (Demir and Kürşat İşleyen, 2013) did not consider some of the realistic features, conditions and constraints described below or included specifications and conditions that are for now not necessary in this particular case study (e.g., sequence dependent setup times, maintenance, overlapping, etc.). Further works can be focused on integrating such conditions and constraints, enlarging the variety, number and complexity of control sub-problems.

In this model, the FMS is composed of two types of resources: machines and transport devices. The FMS has been configured to assemble a set of products, each product with a different operation sequence and component requirements. The static parameters and dynamic parameters as well as the assumptions and constraints taken into consideration are described below.

2.2. Static Parameters

These parameters were considered as static because they were defined on the basis of FMS components, configuration and technical features, which remain unchanged for long periods of time. Parameters related to product design were also considered static and included herein.

T is the set of types of product supported that the FMS can process

R is the set of machines, $R = \{m_1, m_2, \ldots m_r, \ldots m_R\}$

I_p is the operation sequences for the supported types of products;
$$I_p = \{1, 2, \ldots |I_p|\}, p \in T$$

O is the set of operations supported by the FMS

o_{ip} is the operation i of product p, $o_{ip} \in I_p, o_{ip} \in O$

M_{m_r} is the set of manufacturing operations supported by machine m_r, $m_r \in R'$,
$$M_{m_r} \subseteq O$$

$D_{m_a m_b}$ is the distance between machines m_a and m_a; $m_a, m_b \in R$

MJ is the number of available transport devices

d_{sh} is the minimum distance between transport devices

V_{sh} is the transport device's nominal speed, i.e., linear speed

CW_{m_r} is the maximal input buffer capacity of machine m_r

$C_{m_a m_b}$ is the set of alternative transport routes to go from m_a and m_b; $m_a, m_b \in R$, i.e., transport network topology.

$TT_{m_a m_b}$ is the minimum transport time incurred to go from m_a and m_b by a transport device travelling at V_{sh}.

2.3. Dynamic Parameters

The following parameters were considered dynamic because they depend on FMS current status and production order data coming from the tactical level (Figure I-1).

P is the set of products to be manufactured, $P = \{1, 2, \dots, p \dots P\}, p \in T$

R' is the set of active redundant machines for each operation, $R' \in R$

S_p is the number of machine sequences available for product p

$\tau_{m_r}^{ip}$ is the processing time of operation i of product p on machine m_r

$C'_{m_a m_b}$ is the set of available transport routes to go from m_a to m_b

$Att_{m_a m_b}$ is the actual transportation time from m_a and m_b

2.4. Assumptions and constraints

The following assumptions and technical and managerial constraints are mostly taken into considerations in FMS studies with realistic implementations (Luh, 1998; Caumond et al., 2009; Herrero-Perez and Martinez-Barbera, 2010; Berger et al., 2010).

✓ A machine can process one operation at a time:

$$B_{ipyz} + B_{yzip} \leq 1 \qquad \text{Eq. IV-1}$$

where B_{ipyz} is a binary variable set to 1 if operation o_{ip} is performed before operation o_{yz}; otherwise zero (0).

✓ An operation is performed by only one machine:

$$\sum_{m_r \in R'} \mu_{ipm_r} = 1 \qquad \text{Eq. IV-2}$$

71

where μ_{ipm_r} is set to 1 if product j chooses machine m_r to execute operation o_{ip}; otherwise zero (0).

✓ Precedence constraints may exist between operations of the same product. In certain industrial cases such as automobile repair, quality control centers, semiconductor manufacturing and satellite communications, this constraint can be relaxed to achieve better machine utilization and reduce work-in-progress (Witkowski et al., 2010).

✓ Machine's input buffer capacity is limited. A limited input buffer capacity is the consequence of several physical characteristics (e.g., the actual size of the objects, the shop layout or the capacity of the material handling system) and/or managerial policies (e.g., minimizing the in-process inventory). The scheduling problems that take into account this constraint are not abundant because limiting buffers can lead to deadlocks, blocking or machine starvation (Mati et al., 2011).

✓ A product, once it has finished on a machine, is transferred directly to an available machine's input buffer (CW_{m_r}). When machines' input buffer is unavailable, there are two possibilities: the product blocks the machine until a downstream machine's input buffer becomes available or the product is taken care by the material-handling system at the cost of additional transport or intermediate storage times (Tavakkoli-Moghaddam et al., 2005; Mati et al., 2011).

If consecutive product operations are executed on different machines, there is a transport operation between those machines. If blocking machines is not considered, the product remains on transport until it finds a spot in the target machine's input buffer. Therefore, the actual transportation time depends on the input buffer capacity of the subsequent machine and the interactions between transport devices, which depend on the transport network utilization and the transport device circulation rules (e.g., priorities due to possible blockage):

$$Att_{m_a m_b} = TT_{m_a m_b} + att_{m_a m_b} \qquad \text{Eq. IV-3}$$

$$att_{m_a m_b} = \begin{cases} att_{m_a m_b} = 0 \; if \; TT_{m_a m_b} > rwt_{lm_b} \\ att_{m_a m_b} > 0 \; if \; TT_{m_a m_b} \le rwt_{lm_b} \end{cases} \qquad \text{Eq. IV-4}$$

where $att_{m_a m_b}$ is an additional transportation time, and rwt_{lm_b} is the remaining waiting time for product l in the input buffer of m_b, where l is the last product in the machine's input buffer. If blocking the machine is accepted, then the processing time of operation i of product p in machine m_a (being i the last operation of product p on the machine, if multiple operations for p are possible):

$$A\tau_{m_a}^{ip} = \tau_{m_a}^{ip} + a\tau_{m_a}^{ip} \qquad \text{Eq. IV-5}$$

$$a\tau_{m_a} = \begin{cases} a\tau_{m_a} = 0 \ if \ TT_{m_a m_b} > rwt_{lm_b} \\ a\tau_{m_a} > 0 \ if \ TT_{m_a m_b} \leq rwt_{lm_b} \end{cases} \qquad \text{Eq. IV-6}$$

where $A\tau_{m_a}^{ip}$ is the actual processing time of operation i of product p on machine m_a and the blocking time is denoted as $a\tau_{m_a}$.

✓ Product re-circulation is allowed, meaning that a product can visit a machine more than once, increasing machine-sequence flexibility.

✓ The number of simultaneous products in the FMS is limited. Limiting the number of products is a measure to avoid internal deadlocks (Caumond et al., 2009). Herein, we consider the case where a transport device is associated to a product during all its production time. Therefore, the number of transport devices (*MJ*) was chosen to determined the maximum number of products within the FMS at any instant t. This is usually a technical restriction issued from limited resources or space.

✓ Since machines can support various operations, a changeover from one operation to another may incur in setup times. The impact of this constraint can be reduced by for instance arranging similar products in batches. Within one batch, setup times can be neglected.

✓ Preemptions may or may not be forbidden depending on the manufacturing operation. When preemptions are not accepted, an operation has to be completed without interruption once it has started.

2.5. Control sub-problems

Control sub-problems can be defined based on flexibilities provided by the FMS and the aforementioned assumptions and constraints. Out of the FMSs flexibilities mentioned in Section I-3.1, the operation flexibility was not considered in the present study. Thus, it is assumed that products have a pre-defined and fixed operation sequence, which corresponds to FJSS models. For this case, the following control sub-problems were studied:

Release Sequencing: this problem deals with finding a sequence to release a set of *P* products into the FMS. The release sequencing problem can be modeled as a discrete or continuous problem. If modeled as a discrete problem, a product release order needs to be found. The product release order is a vector containing the product position in the sequence, which is treated as a discrete variable. Conversely, if the problem is modeled with continuous variables, the product's release time is treated as a real-value decision variable. Release

sequencing has a significant impact on shop floor workload, the work-in-progress and resource work balance (Shafaei and Brunn, 2000).

✓ Decision variable (discrete problem): U_P, release sequence for the set of P products to be manufactured, $U_P \subseteq P$ and $U_P \subseteq \mathbb{Z}^2$.

✓ Decision variable (continuous variable): rt_p, release time of product p, $rt_p \in \mathbb{R}$

Machine allocation: due to machine and machine-sequence flexibility, this problem deals with choosing one machine among the set of machines capable of performing each of the products' manufacturing operations. This problem is usually modeled with discrete variables.

✓ Decision variable: smr_p, the machine sequence for product p, and $smr_p \in S_p$

Product sequencing: this problem consists in ordering products within the machine's input buffer. This problem can be modeled with discrete variables or continues variables.

✓ Decision variable (discrete problem): qw_{m_r}, is the product's priority at m_r, where $qp_{m_r} = \{1 \prec 2, \prec \cdots \prec CW_{m_r}\}$ in decreasing order.

✓ Decision variables (continuous variable): at_{pm_r} is the arrival time of product p at m_r. Other variables related to the elapsed waiting time, remaining processing times, slack times can also be used.

Product routing: if multiple routes are available for transporting one product from m_a to m_b, then this problem deals with selecting one of those alternative routes. The routing problem can be modeled also with discrete variables as follows.

✓ Decision variables: $cc_{pm_am_b}$ is the selected route, where $cc_{pm_am_b} \subseteq C'_{m_am_b}$.

In the following section, the semi-heterarchical control architecture described in the previous chapter is instantiated to control the aforementioned sub-problems and reduce the myopic behavior emerged when those control sub-problems are dealt with local decision-making algorithms.

2.6. The global performance indicator

Our hypothesis to chose a particular global performance indicator is founded on the fact that myopic control decisions can cause longer waiting times at machines' input buffers and longer routing times, resulting in greater completion times, unbalanced machine utilization, among others. Hence, indicators measuring the completion time dispersion can be used to

calculate the FMS performance and also as a possible indirect myopic behavior measure. Among the non- regular performance indicators that can be used to measure dispersion, two were chosen for this study: the completion time variance (CTV) and the due date mean square deviation (MSD).

2.6.1. The completion time variance

(Kubiak, 1995) defined the CTV to measure the uniformity of the service provided to products. If myopic control decisions affect the way in which products use FMS resources, then control decisions that seem advantageous for some products may incur in greater completion time for others, resulting in greater dispersions around the mean completion time. Thus, it is expected that in the presence of myopic decisions, the CTV increases. The CTV is calculated with product completion times (ct_p) as follows:

$$\text{CTV} = \left(\frac{1}{|P|}\right) \sum_{p=1}^{|P|} (ct_p - \bar{C})^2 \qquad \text{Eq. IV-7}$$

$$\bar{C} = \left(\frac{1}{|P|}\right) \sum_{p=1}^{|P|} ct_p \qquad \text{Eq. IV-8}$$

where $|P|$ is the cardinality of P.

2.6.2. The due date mean square deviation

As seen in Eq. IV-9, the MSD evaluates the dispersion of completion times, but instead of doing it around the mean completion time ($\bar{C}$) it does it on a given due date.

$$\text{MSD} = \left(\frac{1}{|P|}\right) \sum_{p=1}^{|P|} (ct_p - dd_p)^2 \qquad \text{Eq. IV-9}$$

In order to have a meaningful result in terms of the completion time dispersion around the due dates, we adopt the square root of the MSD, which could be seen as the standard deviation of due-date deviations (Eq. IV-10). More, for this case we consider a common due-date for all products, then $dd_p = dd$ so all products have the same local objective.

$$\text{MSD}' = \sqrt[2]{\text{MSD}} = \sqrt[2]{\left(\frac{1}{|P|}\right) \sum_{p=1}^{|P|} (ct_p - dd)^2} \qquad \text{Eq. IV-10}$$

Having the MSD as an objective function allows us to position our approach in a Just-in-Time (JIT) context. Therefore, myopic behavior can be related with inventory costs, customer satisfaction and on-time deliveries, which are important performance criteria for industrialists. In JIT, products that are finished early than the due date carry out with higher

inventory costs and those that are finished after their due dates result in customer dissatisfaction (Jozefowska, 2007).

3. Description of the SHFMS architecture

The constituent elements of the semi-heterarchical control architecture described in Section III-3 were particularized on the basis of the FMS model described in the precedent section. Figure IV-1 shows the resulting semi-heterarchical architecture.

3.1. The local decisional level

As depicted in Figure IV-1 the local decisional level comprises two types of local decisional entities: transport decisional entities ($vLDE_j \forall j \in \{1, \dots, MJ\}$) and machine decisional entities ($mLDE_r \forall r \in \{1, \dots, R\}$). $vPLE$s are concerned with release sequencing, machine allocation and product routing sub-problems and $mPDE$s only make control decisions concerning product sequences within input buffers. For this particular case, machine decisional entities are temporally and socially myopic because their control decisions do not consider a global objective and local decisions are not shared with other $mLDE$s. Hence, $mLDE$s' myopic behavior is accepted and not reduced. The relationship between local decisional entities is fully heterarchical.

If the global decisional level did not exist, the FMS would be controlled as follows. As soon as a production order arrives, composed of P products, those products are released into the FMS in the order sent by the tactical level (Figure I-1). Once loaded in the FMS, products are assigned to available $vLDE$s. Then, each $vLDE$ makes machine allocation and routing decisions to fulfill the products' operation sequences. $vLDE$s make their decisions on the basis of data furnished by $mLDE$s and the state of FMS, e.g., route load, distances. At each machine, the $mLDE$ makes sequencing decisions to select one of the products in its input buffer. Once a product has been manufactured, the product is discharged from the FMS and the $vLDE$ becomes available for an upcoming product. $vLDE$s are temporally myopic since machine allocation and routing decisions are made stepwise.

For the machine allocation problem, machine allocations are made for one operation o_{ip} at a time without assessing the consequence of machine allocations for the rest of the operation sequence. A machine allocation that might seem beneficial (e.g., reduced input buffer time) may result in greater completion times. In turn, routing decisions experienced the same behavior by choosing one route at a time and possibly assessing routes by segments. $vLDE$s are also socially myopic because they do not share their local control decisions with other $vLDE$s. Based on the definitions of Pétin et al. (2007) and Meyer et al. (2011), this type

of control can be situated in the context of product-driven control. Product-driven control consists in providing the product with information, decision and communication capabilities so the product becomes an active decider capable of controlling and the executing its manufacturing sequence (Pétin et al., 2007).
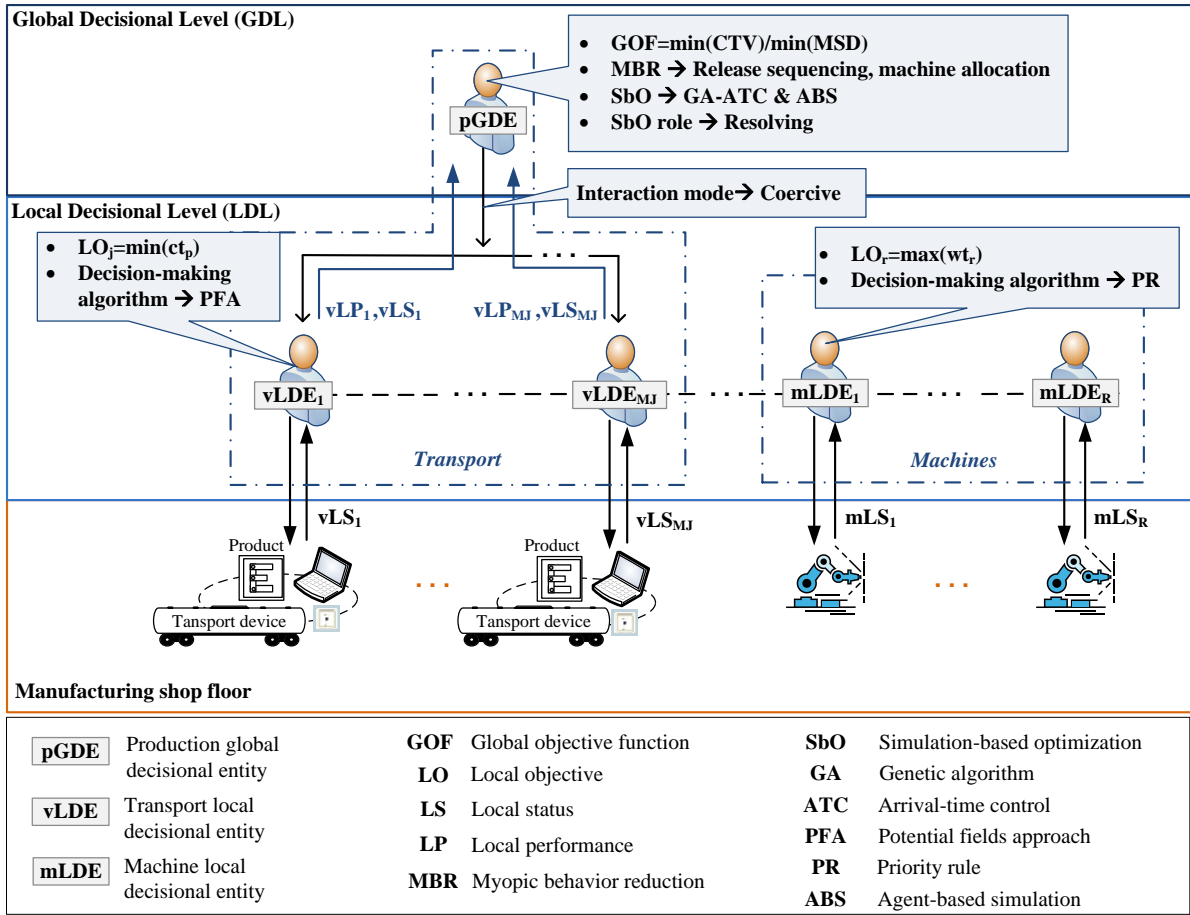
Figure IV-1: The SHFMS control architecture

Mostly issued from the distributed artificial intelligence field, several approaches can be used for implementing the local decisional level. Figure IV-2 makes a partial summary of possible approaches that can be used taking into account the aforementioned description (Baker, 1998; Monostori et al., 2006). Among these approaches, the potential fields approach (PFA) has proven its efficiency and reactivity for heterarchical FMS control (Zbib et al., 2012; Pach et al., 2012). Then, PFA with attractive fields has been chosen as the local decision-making algorithm for *vLDE*s and a simple priority rule is used by *mLDE*s to make their control decisions. More insights on these choices are given in Sections IV-4.1 and IV-4.2.
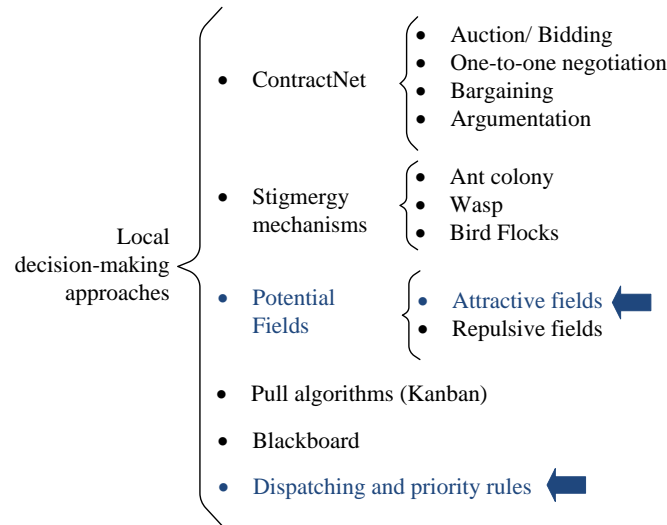
Figure IV-2: Possible local decision-making approaches

As mentioned in Section III-3, there are no restrictions on the modeling approach, e.g., multi-agent systems (MAS), holonic manufacturing systems (HMS), Product-driven (PD), used for implementing the local decisional level.

3.2. The global decisional level

The global decisional level is composed of only one global decisional entity, $pGDE_1$ ($N = 1$), assigned with one production-related global objective function GOF, i.e., the CTV or MSD. As seen in Figure IV-1, the global decisional level is configured to reduce myopic control decisions related to release sequencing and machine allocation sub-problems, given the proven significant effects of these control decisions in FMS performance (Shafaei and Brunn, 2000; Joseph and Sridharan, 2011b). For this case study, it is considered that product routing decisions can be dealt locally due to frequent interactions between transport devices, needing constant product routing decisions.

According to definitions given in the previous section, $pGDE_1$ has to be configured to deal with either the CTV or the MSD as objective functions. To reduce myopic decisions concerned by the $pGDE_1$, two optimization algorithms, the arrival-time control algorithm (ATC) by Duffie and Prabhu, (1994) and a genetic algorithm (GA) specifically conceived for this case study were chosen. The ATC was conceived to deal with release sequencing decisions and the GA with machine allocation decisions. As seen in Figure IV-3, other algorithms can be selected, from which heuristics and meta-heuristics are the most popular ones based on the literature review reported in Chapter II. Features and advantages of chosen algorithms are explained later in Section IV-4.3.
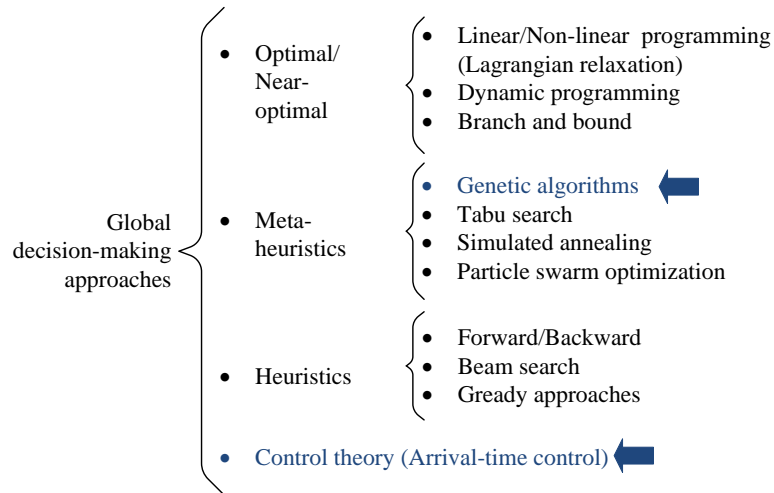
Figure IV-3: Possible global decision-making algorithms

3.3. $GDE - LDE$ hierarchical interactions modes

Figure IV-4 shows the chosen interactions between the global and the local decisional levels, only for transport decisional entities ($vLDE$s). For the two control sub-problems for which $pGDE_1$ is configured to reduce $vLDE$s' myopic behavior, the chosen interaction mode was the *coercive* mode because for both cases the proposed simulation-based optimization (SbO) technique takes the *resolving* role. This choice was straightforward because both optimization algorithms, GA and ATC, were conceived to deal with the machine allocation and release time variables, and values can be transparently adopted by $vLDE$s.
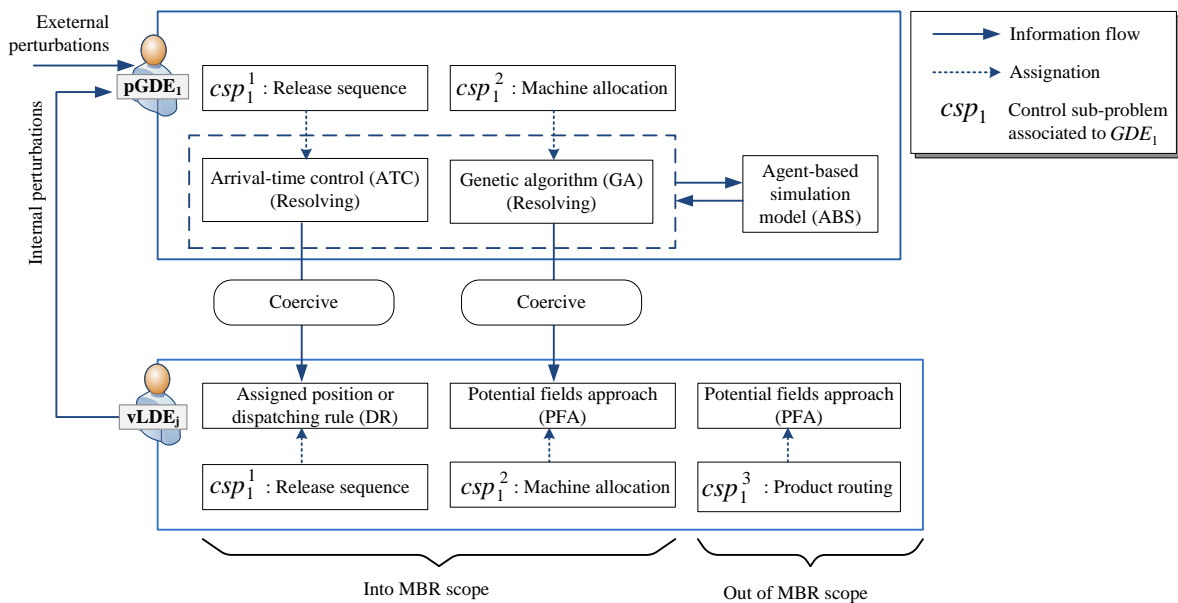
Figure IV-4: The chosen $GDL - LDL$ interaction mode

On the $vLDE$'s side, the potential fields approach deals with machine allocation and routing decisions. Since PFA is not concerned by the release sequence, a pre-defined release position (e.g., position in the batch, availability, etc.) for each product or a dispatching rule (DR), e.g., shortest or longest processing times, earliest due date, etc., can be used to locally determined how products are launched into the FMS. The following section explains in details each of these algorithms and the internal structure of decisional entities.

4. Decisional entities structure

This section describes first the internal structure of the local decisional entities, transport decisional entity $vLDE_j$ and machine decisional entity $mLDE_r$ and then it focuses on the global decisional entity $pGDE_1$.

4.1. Transport local decisional entity $vLDE_j$

The transport decisional entity is an instance of the local decisional entity described in Section III-4.1 (Figure III-4). As shown in Figure IV-4, the $vLDE_j$ is endowed with the potential fields approach to make machine allocation and routing decisions. For a product p assigned to $vLDE_j$, the local decision variables are $LDV_j = \{u, m_r, cc_{m_c m_r}\}$ where u is a release position of p and $u \in P$, m_r is a selected machine for each o_{ip} where $m_r \in R'$, $o_{ip} \in M_{m_r}$ and $cc_{m_a m_r}$ is a transport route connecting the current position (i.e., m_a) to the selected machine m_r and $cc_{m_a m_r} \in C'_{m_a m_r}$. The $vLDE_j$'s control and interaction modules are depicted in Figure IV-5.
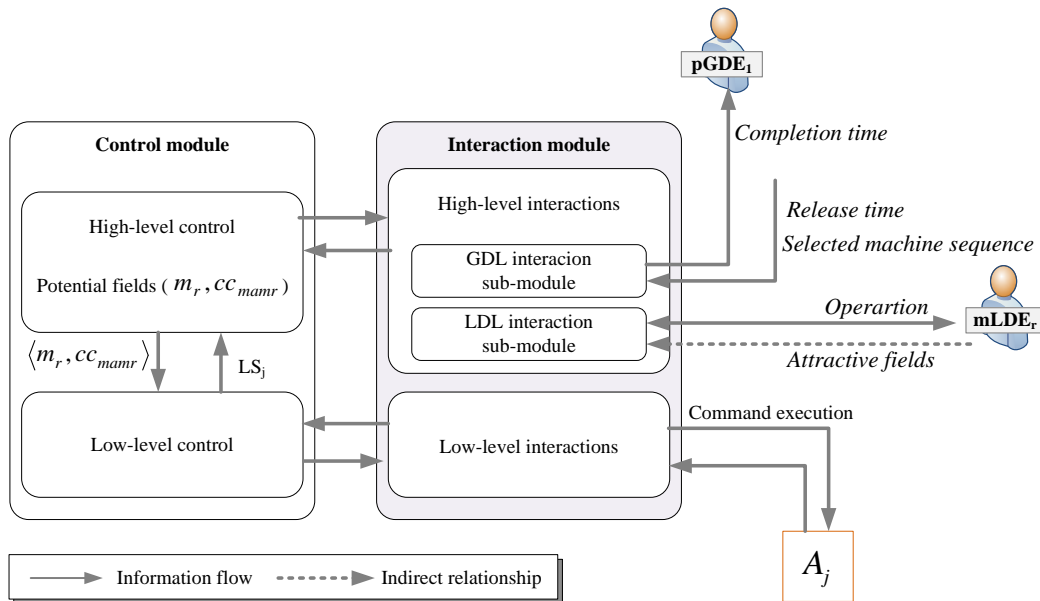
Figure IV-5: $vLDE_j$ control and interaction modules

4.1.1. The control module

For the high-level control, the potential fields approach proposed by Zbib et al., (2012) and Pach et al., (2012) was chosen as the decision-making algorithms. PFA is a real-time, totally reactive and heterarchical manufacturing control strategy based on attractive fields. Machines in the FMS emit 1D-attractive fields along the transportation system for each service they can provide. The field's intensity is reduced by the current status of the machine, the number of products (or operations) in its input buffer and the expected transportation time to reach the machine. Throughout the transportation network, the $vLDE_j$ gathers field values at decisional points where it can reassess its decision. At each decisional point, each $vLDE_j$ chooses the machine m_r with the highest field emitted for the manufacturing operation o_{ip} the product needs. As explained before, each $vLDE_j$ carries out a product p from the moment it is released into the FMS until it has finished its operation sequence I_p. During that time, PFA serves as an allocation strategy and also as a product routing strategy when there are multiple routes to reach a destination. More information on PFA can be found in Appendix E.

Our motivations for working with PFA can be explained vis-à-vis the following aspects. First, high myopic behavior regarding release sequencing, machine allocation and product routing has been identified in PFA (Zambrano Rey et al., 2011a; Zbib et al., 2012; Pach et al., 2012). Products are released into the FMS as they arrive and machine selection is executed stepwise for each task in the manufacturing sequence, without assessing the long term. In addition, each $vLDE_j$ is only concerned by its own objective, self-containing its decisions and avoiding information exchange with other $vLDE$s. Second, PFA is flexible, adaptable and reactive to internal and external changes. Since PFA is also used within the simulation model into the global decisional entity, updating FMS conditions is straightforward, thus favoring solution coherence. PFA's low complexity lies in the fact that machine allocation and product routing decisions depend exclusively on the information gathered by $vLDE$s, therefore there are no negotiations and less communication overhead. In addition, $vLDE$s constantly assess their decisions and do not require reservations or commitments of any kind. Clearly, the key aspect of this approach is the formulation of the attractiveness that drives the entire system and the way fields are propagated throughout the FMS (field attenuation). Last, PFA has been compared with other heterarchical approaches, i.e., contract-net (Zbib et al., 2012), and exact optimization models, i.e., an integer linear program (Pach et al., 2012), showing promising results. However, PFA has not been evaluated for the CTV and will require a dispatching rule or another algorithm that takes due dates into consideration in order to deal with the MSD.

The low-level control contains the control algorithms to move the transport device (A_j in Figure IV-5) to the selected destination. The distance between transport devices (d_{sh}) and the

device's nominal speed (V_{sh}) are handled at this level. In some cases, transfer priorities and collision avoidance could also be managed by low-level control.

4.1.2. The interaction and information storage modules

Through the strategic interaction sub-module each $vLDE_j$ receives data coming from the global decisional entity and reports its local performance (i.e., the completion time $ct_p(t)$) when the operation sequence has been finished (Figure IV-5). As mentioned before, the PFA restricts the interactions between entities of the same kind. Thus, through the *LDL* interaction sub-module the $vLDE_j$ only exchanges information with $mLDE$s, indirectly through attractive fields (as represented by the dotted arrow in Figure IV-5) and directly in form of two-way communication when the $vLDE_j$ arrives at the machine and requests for a specific operation o_{ip} (represented by the solid double arrow in Figure IV-5).

Given the low complexity of the potential fields approach, only the minimum distance between transport devices (d_{sh}) and the transport device's nominal speed (V_{sh}) are configuring parameters requiring storage. In addition to that, product-related information such as product identification, product type and product operation sequence are also saved in the storage module, and updated every time the transport device (TD_j) is assigned with a new product.

4.2. Machine local decisional entity $mLDE_r$

The machine decisional entity is also an instance of the local decisional entity described in Section III-4 (Figure III-3). The main purpose of this entity, in terms of control sub-problems, is to make decisions to sequence products within the input buffer. To do so, a priority rule is embedded into the high-level control module in charge of selecting one of the products from the machine's input buffer. The first-input first-output rule was chosen to solve the product sequencing problem, thus the product arrival time (at_{pm_r}) was used as variable.

The $mLDE$'s local objective is to maximize its working time ($max(wt_t)$ in Figure IV-1), for which a heuristic algorithm is also placed in the high-level control module. This heuristic algorithm is the core of the potential fields approach because it determines the attractiveness values according to the local machine's status. The attractiveness value is inversely proportional to the remaining processing time, which includes the processing time of the current operation and operations waiting in the input buffer (Pach et al., 2012). In order to maximize the working time, the attractiveness value is updated every time an operation has finished.

Once a product has been chosen, the $mLDE_r$ low-level control handles the $vLDE_j$ passage from the input buffer to the machine and proceeds to execute the requested operation.

The information storage module contains data related to the maximum attractiveness value, supported manufacturing operations (M_{m_r}), maximum buffer capacity (M_{m_r}) and the machine state (available or unavailable).

4.3. Global decisional entity $pGDE_1$

As mentioned before, $pGDE_1$ was configured to work on two control sub-problems: csp_1^1 the release sequence sub-problem and csp_1^2 is the machine allocation sub-problem (Figure IV-4). Thus, $GDV_1 = \{smr_p, rt(t)_p\} \, \forall p \in \{1 \dots P\}$. The purpose of $pGDE_1$ is to find the set of values GDV_1^* that optimize the global objective function GOF_1.

4.3.1. The control module

Taking as reference the internal GDE's structure proposed in Section III-4.2.1 (Figure III-5) the $pGDE_1$'s control module structure proposed for this case is presented in Figure IV-6[6]. The optimization module and the simulation model are described below.
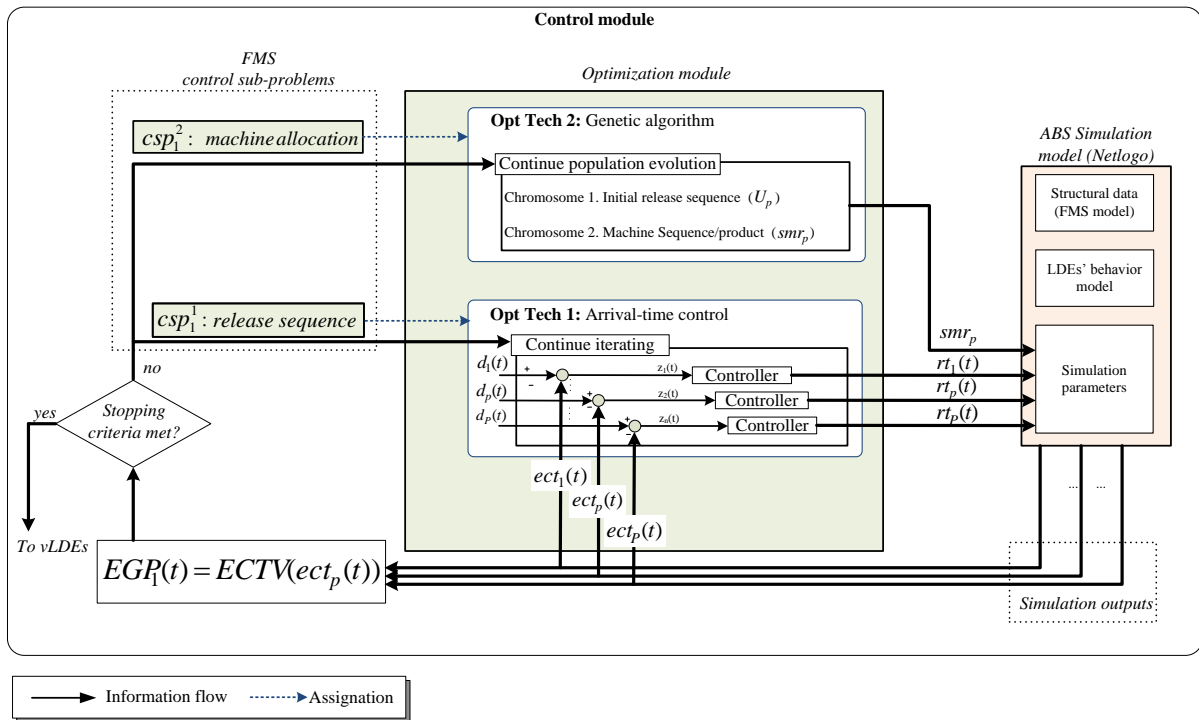
Figure IV-6: Control module of *pGDE₁*

The optimization module

Two optimization mechanisms were chosen to tackle myopic release sequencing and myopic machine allocations. The release sequencing sub-problem was treated with a continuous variable, the release time (rt_p(t)), because it provides more information that a

---

[6] For better visualization, control sub-problems and optimization mechanisms are presented in descending order.

simple sequence. In turn, the machine allocation sub-problem was modeled as a discrete combinatorial problem, aiming at selecting a complete machine sequence for each product (smr_p).

Among the various simulation-based optimization mechanisms that can be used to tackle myopic release sequences (e.g., dispatching rules, heuristics, and meta-heuristics capable of handling continuous variables such as particle swarm optimization), a distributed cooperative approach, the arrival-time control was chosen (Prabhu and Duffie, 1995)(Opt. Tech 1 in Figure IV-6). ATC is a distributed and autonomous decision-making algorithm for part-driven systems in which local schedules are generated in a purely distributed manner using minimal global information (Prabhu and Duffie, 1995). ATC is based on continuous variable control, hence its convergence and dynamic behavior can be analyzed and predicted (Prabhu and Duffie, 1996). Several reasons support our choice. First, ATC is a low complexity approach because each controller works independently without requiring explicit information from other controllers, resulting in no combinatorial complexity. Second, each decisional entity (i.e., $vLDE$) has its own controller in the $pGDE_1$, working on its local goal. Third, ATC requires time-scaled simulations that are used to provide feedback for local solution adjustment. The simulation model can use any decision-making algorithm for the *LDE*'s behavior model to solve machine allocation and product routing decisions. And last, due to its closed-loop structure, ATC can adapt gracefully to internal and external perturbations. The ATC is described in detailed in Appendix F-I.

Several priority rules, heuristics and meta-heuristics have been proposed to manage the machine allocation sub-problem (some examples are referenced in Section IV-3.2). Out of those optimization techniques, a genetic algorithm (GA) was conceived for such purpose (Opt. Tech. 2 in Figure IV-6). The encoding, operators and parameters are described in more detail in Appendix F-II. We have chosen genetic algorithms over other meta-heuristics due to the following reasons:

✓ Flexibility: GAs have been implemented effectively for many different problems in many fields. The numerous studies proposing encoding techniques, genetic operators and enhancement strategies have given GA the necessary maturity for their commercial usage (Grupe and Jooste, 2004; Nie et al., 2013). Thus, GA matches the requirements for the previously explained FMS model.

✓ Efficiency: GAs are known for their efficiency in finding nearly optimal solutions in a reasonable time (Honghong and Zhiming, 2003). Another indicator of the GA flexibility and efficiency is that several techniques can be adopted in order to improve the initial population generation and the population evolution in order to avoid falling into local optima (Pezzella et al., 2008; Gao et al., 2006).

✓ Scalability: GA have been tested for large-scale problems, and their behavior can be adapted depending on the size of the problem (He and Hui, 2007). A linear encoding technique or dividing the problem into multiple chromosomes (as proposed herein) can help to reduce the computing burden. Parameter control is also another tool that tries to adapt the genetic algorithm to problem conditions (Eiben et al., 2007). Although GA usually offer interesting computing costs, several strategies can be used to accelerate their processing speed (Rossi and Dini, 2000).

✓ Parallelism: GA maintain a population of good solutions instead of adjusting a single solution, making the search process more robust. In the context of real-time problems, GA can be stopped at any moment so the best solution out of the available solutions can be executed.

As described in Section III-4.2 the genetic algorithm can be connected with ATC in two ways: decoupled, where the two algorithms work separately; or coupled where ATC is embedded into the GA forming a close loop. More information on these two coupling modes and their efficiency to solve a flexible job-shop problem similar to the modeled considered herein can be found in Zambrano Rey et al., (2014). A feasible solution proposed by the two optimization approaches is composed of a release sequence (based on the release times for each product, rt_p) and a selected machine route (smr_p), for each product. In addition, the GA contributes with ATC's initial conditions in the form of a discrete release sequence U_p. The flow diagrams explaining these two types of interactions are presented in Appendix F-III and Appendix F-IV.

The simulation module

In the proposed approach, agent-based modeling and simulation (ABS) was used to model the FMS and basic heterarchical behavior of local decisional entities. Between discrete-event simulation (DES) and ABS, this latter offers individual-based models, bottom-up approaches and a more natural representation of heterarchical-based approaches where decisional entities have autonomous behavior (Siebers et al., 2010). Agent-based models can explicitly represent the complexity resulting from individual actions and interactions occurring in the real FMS. Siebers et al., (2010) summarized a list of problems that are good candidates to be modeled using ABS. All features mentioned in Siebers's paper apply to FMS control problems and make ABS suitable for the proposed semi-heterarchical approach.

For this particular case, the FMS simulation model was implemented using the multi-agent programmable modeling environment NetLogo (Wilensky, 1999) (the Agent-based Simulator in Figure IV-6). This tool was chosen for its functionalities to model complex systems evolving over time, such as heterarchical-based control approaches (Tisue and

Wilensky, 2004). NetLogo contains the appropriate elements to model each local decisional entity, its behavior(s) and its interactions with other local decisional entities. More, Netlogo has been used as an on-line tool in distributed applications (Wang, 2009; Rolón and Martínez, 2012) or as an evaluation tool (Barbosa et al., 2011; Berger et al., 2010).

4.3.2. The interaction and information storage modules

Since in this particular case there is only one global decisional entity, the GDL interaction sub-module is not required. On the contrary, the LDL interaction sub-module must be implemented using industrial networks, wired and wireless, this latter given the mobility of local decisional entities ($vLDE$). Machine sequences (smr_p) and product release times ($rt(t)_p$) for each product are send to $vLDE$s and actual completion times ($ct(t)_p$) are received to calculate the deviation with estimated local and global performances (respectively, $ect(t)_p$ and $ECTV(ect(t)_p)$ in Figure IV-6). In turn, the storage module contains the internal parameters to configure the two optimization mechanisms, as described in Appendix F, as well as the static parameters, constraints and assumptions issued from the FMS model (Section IV-2) and necessary for the ABS model.

5. Control Strategy

The control strategies under normal and abnormal conditions are detailed in this section, following the predictive-reactive strategy described in Section III-5.

5.1. Control strategy under normal conditions

The control strategy explained herein refers only to $pGDE_1$ and $vLDE$s given that $mLDE$s' myopic behavior is accepted. Thus, $mLDE$s follows a totally reactive strategy for controlling product sequences within input buffers.

Under stable conditions, reducing myopic behaviors is encouraged to achieve a better performance as explained in Section III-5.1. Therefore, in the predictive phase (Figure IV-7) $pGDE_1$ proceeds to explore iteratively several possible solutions for the machine allocation and release sequence sub-problems. The optimization mechanisms (GA and ATC in Figure IV-7) develop the solution until convergence (Eq. IV-11) and stop when the solution remains steady for an *itermax* number of iterations. Other stopping criteria can be used depending on the optimization mechanism(s), SbO role, problem complexity and desired reactivity. For the latter, it should be possible to stop the SbO cycle at any given time and use the best solution found until that moment, this is an advantage of population-based techniques.

$$|CTV_{iter} - CTV_{iter-1}|/|CTV_{iter}| < \varepsilon \qquad \text{Eq. IV-11}$$
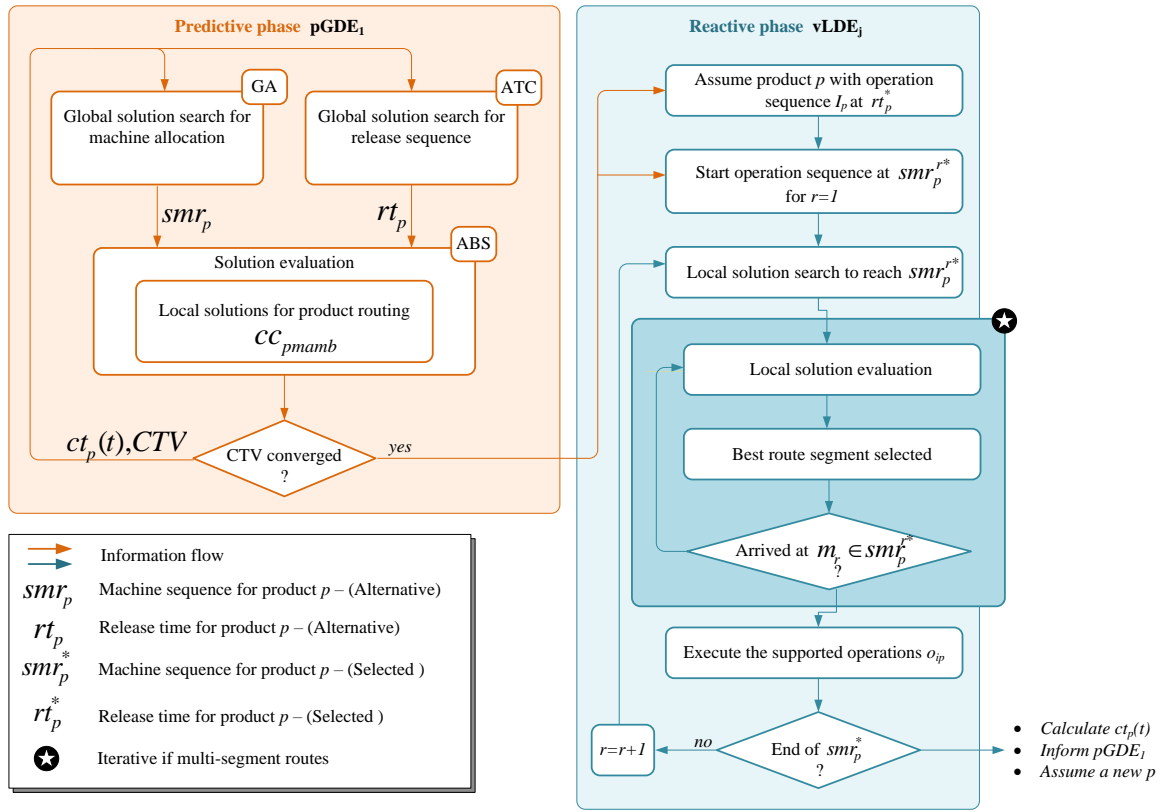
Figure IV-7: Control strategy under normal conditions (FMS case study)

The SbO techniques are in charge of evaluating those alternative solutions for a time horizon Th. Herein, the notion of time horizon is related to the number of products evaluated within the time horizon, as it is handled in rolling horizon approaches (Jia et al., 2009). If Th is small, i.e., few products, a myopic short-tem solution is obtained at a lesser cost for $pGDE_1$. This decision usually depends on the planning made by the tactical level and the size of the production order P and their operation sequences. For now, Th is setup to include all products in a production order P.

Once a solution is passed down to $vLDE_j$, this latter assumes $pGDE_1$'s solutions and execute them. However, since the product routing sub-problem is not dealt by the global decisional entity, an internal iterative process is triggered every time a routing decision has been made. The iterative process depicted in Figure IV-7 (marked as ✪) fits the most generic case in which each route can be divided in multiple segments with divergent and convergent nodes (examples of multi-segment routes can be found in (Herrero-Perez and Martinez-Barbera, 2010; Zhang et al., 2012; Zbib et al., 2012)). In a multi-segment route, decisions have to be made at each divergent point. Since $pGDE_1$ does not intervene in these decisions each $vLDE_j$ handles route selection myopically, making short term decisions at each divergent node (temporal myopia). More, $vLDE$s are socially myopic because they retain

information concerning route selection, so low communication burden can be maintained. Under these circumstances, $vLDE$s have to constantly assess traffic state, thus dealing locally with routing decisions seems a logical solution to ensure a high reactivity to traffic conditions. When the $vLDE_j$ has finished with the assigned product, it reports the actual completion time (ct_j) and waits for a new product assignation.

5.2. Control strategy under abnormal conditions

Abnormal conditions arise when the state of the FMS used to obtain control solutions changes, hence those solutions are either not valid or will potentially take the system out of the established target profile. As described in Section III-5.2, some perturbations are handled locally by $vLDE$s and others will require the $pGDE_1$ intervention. For this particular case, the perturbation handling procedure is predefined depending on the type of perturbation and the decisional entity in charge of responding. Perturbation handling proceeds as described below.

Perturbation handling within vLDE's domain: transport-related perturbations (e.g., longer than expected transportation times due to traffic or route segment blockage) are handled locally because $vLDE$s are fully autonomous for dealing with such control decisions. Exceptionally, $pGDE_1$ may intervene if $vLDE$s cannot handle a critical situation such as deadlock.

Perturbation handling within $pGDE_1$'s domain: under the *resolving* SbO role and *coercive* interaction mode, perturbations concerning the control sub-problems for which $pGDE_1$ is configured should be handled by this latter. Thus, $pGDE_1$ may be concerned by external and internal perturbations. External perturbations affect the current production order (P), for instance changing the number of products to be manufactured (e.g., urgent order, product/order cancellation) or the parameters of the current order (e.g., changes in due dates, type of products). Internal perturbations are the consequence of FMS components' health. Machine/tool performance variability (e.g., highly variable processing times) and breakdowns may perturbed control solutions provided by $pGDE_1$, making the machine partially or totally unavailable for production. The modular configuration of the proposed optimization module allows two possible strategies to react to external and internal perturbations. The first strategy aims at dynamically changing the connection mode from coupled to decoupled to speed up $pGDE_1$'s SbO cycles. The second strategy aims at only dealing with the release time sub-problem at the global level and leave machine allocations to be dealt at the local level. This strategy is suitable when FMS's machines deteriorate, so operation times cannot longer be considered deterministic, and the failure rate increases putting the machine frequently unavailable. In such cases, $vLDE$s can locally make machine allocation decisions using PFA, until the machine state gets back to more reliable conditions. On the $mLDE$'s side, any event affecting product sequences in the input buffer is treated locally by the FIFO rule.

6. Synthesis of the chapter

This chapter described one possible implementation of the proposed semi-heterarchical simulation-based optimization approach to reduce the myopic behavior of particular FMS control. Several choices were made throughout the stages of the implementation procedure. Indeed, other architectures can result by making slight changes of the proposed configuration or by taking other options at every decisional point of the procedure, i.e., interaction modes, simulation-based optimization techniques, SbO roles, local decision-making algorithms, etc. Hence, the fact that several configurations are possible makes our approach sufficiently generic, not only in the context of FMS control but also in other problems that have similarities with the FMS control problem, i.e., logistics, hospital management, transportation management, etc.

At first, a FMS model was proposed, taken into account realistic assumptions and constraints. From that model, four control sub-problems were identified (i.e., release sequencing, machine allocation, product sequencing, product routing). In this case, one global entity was configured with two optimization algorithms, a GA and the ATC, and an agent-based simulation model to reduce the myopic machine allocation and release sequence sub-problems. In turn, product and machine decisional entities were endowed with local decision-making algorithms, PFA and FIFO rule, that allowed them partial and total autonomy to deal with product routing and product sequencing sub-problems, respectively. This chapter particularly described the case in which myopic behavior is reduced only for some control sub-problems and accepted for other control sub-problems in order to preserve good local reactivity to perturbations. The control strategy clearly defines the procedures to follow and the entities responsible when handling external and internal perturbations.

As mentioned in the introduction, the next chapter relates simulation and real experimentation studies carried out in the AIP-PRIMECA assembly cell in order to evaluate our approach. Also, other results reporting the efficiency of the GA and GA-ATC on other FMS benchmark problems are also provided.

Chapter V. An Experimental Case Study: The AIP-PRIMECA Cell

1. Introduction

Now that all the constituent elements of the proposed approach have been described, herein the prototype implementation of the proposed approach into an assembly cell will be described. At first, the experimental data will be presented in Section V-2 and then, the implementation of each decisional entity will be detailed (Section V-3). The evaluation of the proposed approach was executed on the basis of a simulation study (Section V-4 and V-5) and hardware-in-the-loop experimental evaluations on the AIP-PRIMECA assembly cell (Section V-6). The simulation study was carried out for static and dynamic scenarios, taking into consideration two objective functions and three different configurations of the global decisional entity. Some conclusions and results analysis are offered at the end of the chapter (Section V-7).

2. General description of the AIP-PRIMECA cell

The flexible assembly cell located in the AIP-PRIMECA Center at Valenciennes University allowed us to evaluate our approach. The AIP-PRIMECA cell (in short AIP cell) shown in Figure V-1, can be considered as an FMS because it provides the different types of flexibilities, i.e., machine flexibility, material-handling flexibility, process flexibility and machine-sequence flexibility. This cell is composed of industrial elements such as industrial robots, conveyor system, sensors, actuators, programmable controllers. Therefore, the AIP cell is an interesting platform for hardware-in-the-loop evaluation of FMS control approaches. More, this assembly cell has been modeled as a Flexible Job Shop (Trentesaux et al., 2013). Therefore, linear and quadratic programming models have been conceived inspired from this cell, which can be used to obtain reference scheduling solutions.

To instantiate the FMS model presented in Section IV-2 it is necessary to analyze and describe in detail the AIP cell. Therefore, the AIP cell's data related to material-handling, product and machines can be assumed as parameters and model constraints. All these data is also necessary for the internal configuration of the global and local decisional entities. In the following sub-section the AIP cell is described in detail and then in Section V-3 the technological implementation of decisional entities is explained.

Figure V-1: The AIP-PRIMECA assembly cell

2.1. Material-handling data

The AIP cell's machines are linked by a unidirectional and flexible conveyor system as depicted in Figure V-2.
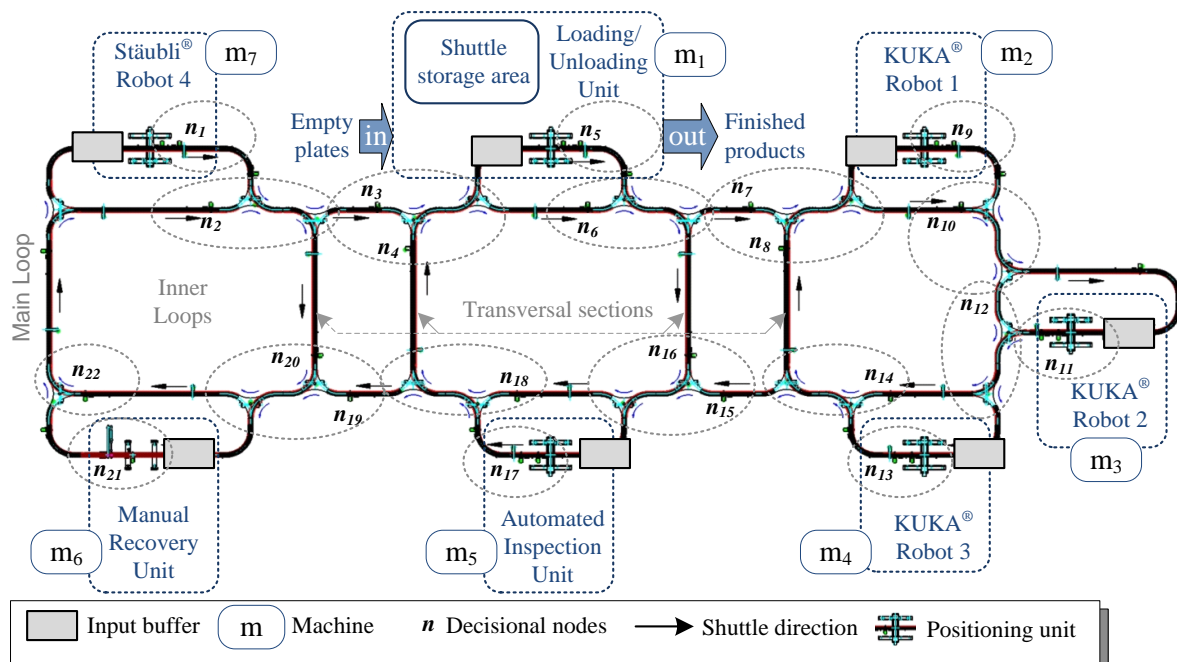
Figure V-2: The AIP cell's layout

The conveyor is composed of a main loop, several derivations to reach the machines, positioning units in front of machines, and four transversal sections composing multiple inner loops to allow material-handling flexibility (Montratec, 2014).

Wago® controllers (a. in Figure V-3), model 750-841, (WAGO Kontakttechnik, 2005), are used to drive the transfer gates allowing (when needed) to transfer a shuttle from one loop towards another (b. in Figure V-3). Also, the same type of controllers are used to handle the machine's input buffers and the positioning units in front of the machine so the shuttle can be maintained fixed in place while an operation is being executed by the machine, i.e., a robot (c. Figure V-3). Dotted shaded regions in Figure V-2 represent the areas concerned by each Wago® controller.
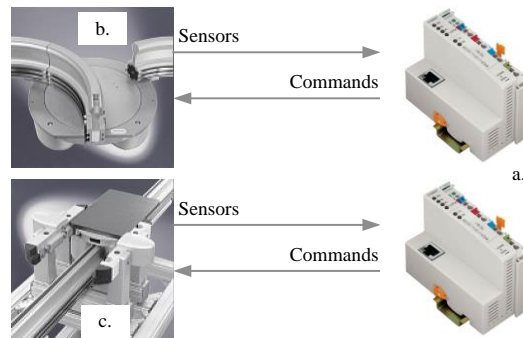
Figure V-3: a. Wago® controller, b. Transfer gate, c. Positioning unit.

Shuttles are self-propelled transport devices, placed on the conveying rail, which carry the physical product through the conveyor system (a. in Figure V-4). Each shuttle embeds an infrared control mechanism to avoid colliding with a shuttle position in front and maintain a minimal security distance between the two shuttles (d_{sh}). Other control mechanisms to manage their own speed in curves and straight segments (V_{sh}), dock into machines and stop before transfer gates make also part of shuttle's control. Throughout the conveyor, decisional nodes (**n** in Figure V-2) have been set up to let the shuttle communicate with dedicated Wago® controllers (d. in Figure V-4) that are in charge of tracking the shuttles' position. To this end, radio frequency identification (RFID) technology is used for shuttle identification and localization purposes (b. in Figure V-4). In this way, at each decisional node (c. in Figure V-4), the shuttle stops, localizes itself and gathers other data necessary to make a decision. In total, 11 Wago® controllers manage all transfer gates and shuttle tracking.

For hardware-in-the-loop evaluations, a maximum of 10 shuttles are available, thus at a given time *t* no more than ten products can be present in the AIP cell ($MJ = 10$). For the simulation model and the FMS formal model the number of shuttles is a variable that can take different values, with no restriction other than saturation.
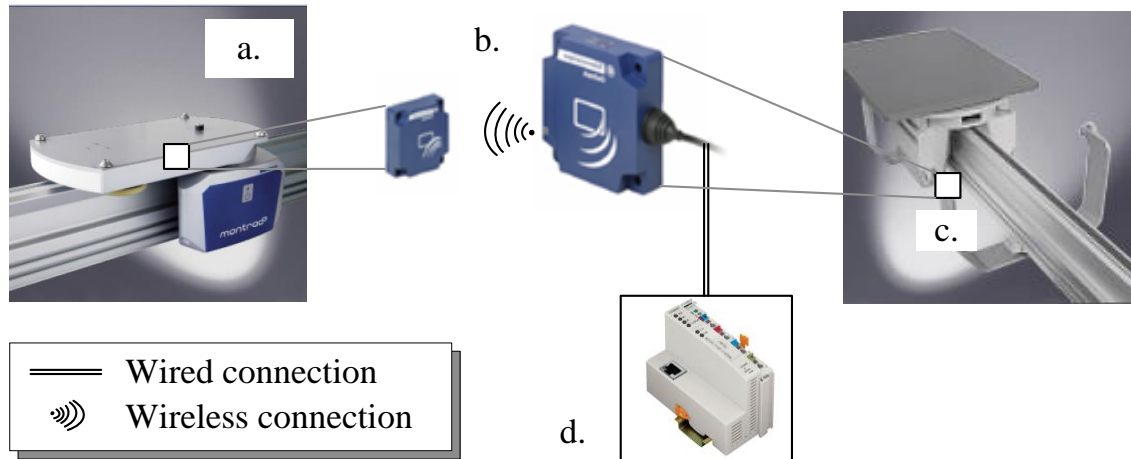
Figure V-4: a. Shuttle, b. RFID tag (small) and reader, c. Decisional node, d. Wago® controller for shuttle

Product routing flexibility is accomplished by transversal sections, proving more than one route ($C_{m_a m_b} > 1$) between two machines. Though in certain cases there is only one direct route (e.g., m_1 to m_2), looping around may be seen as a way to obtain multiple transport routes. Table V-1 reports the minimum transport times (in seconds) incurred to go from one node to the followings.

Table V-1: Minimum transport times (in seconds) between nodes

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1			6																	7		
2			4																	5		
3					5	4																
4					5	4																
5								6								7						
6								4								5						
7									5	4												
8									5	4												
9											13	5										
10											11	3										
11													7	6								
12													5	4								
13								7						6								
14								5						4								
15																5	4					
16																5	4					
17			7															6				
18			5															4				
19																				7	4	
20																				7	4	
21	13	12																				
22	10	9																				

From these values, minimum transport times between machines ($TT_{m_a m_b}$) can be obtained. These transport times were measured under normal conditions and for only one shuttle in the conveyor. For the FMS model these transport times are deterministic.

2.2. Product data

Contrary to AIP cell's components, which are industrial elements, the products assembled at the AIP cell are didactic products created for proof-of-concept evaluations. Therefore, products were conceive to exploit the AIP cell's flexibilities and allow evaluating various scenarios, e.g., product mixture, different operation sequences, etc. The AIP cell's machines are configured to assemble 6 different components (axis, r-shape, L-shape, I-shape, screw as seen in Figure V-5) into a matrix pallet carried by the shuttle. With these five components, seven different types of products ($T = 7$) can be assembled by the AIP: "B", "E", "L", "T", "A", "I" and "P" as illustrated in Figure V-5. Once a product has been assembled it is unloaded with its matrix pallet so an empty one can be loaded into the shuttle.
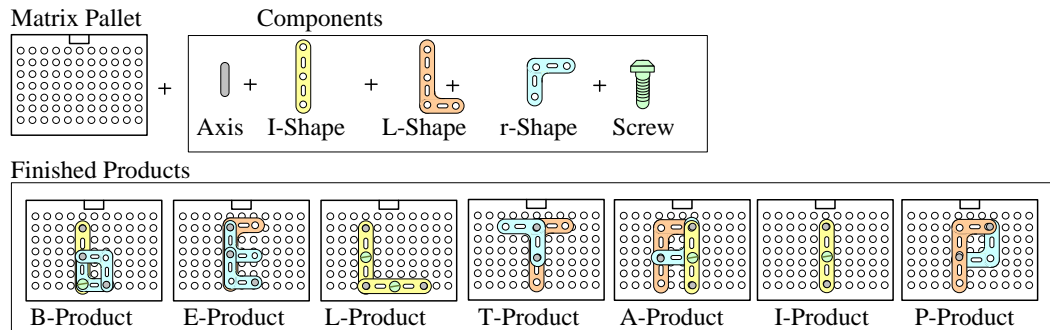
Figure V-5: Product types and components

To assemble each product, a sequence of manufacturing operations has to be followed (I_p). Product routing is not considered as a manufacturing operation. The AIP cell supports nine manufacturing operations ($O = 9$): "plate loading", "axis mounting", "r_comp mounting", "I_comp mounting", "L_comp mounting", "screw_comp mounting", "inspection", "manual recovery" and "plate unloading". Given that the automated inspection and the manual recovery unit do not offer any machine and machine-sequence flexibility, these operations are not taken into consideration for the experimental cases reported in this chapter, then $O = 7$ henceforth. Table V-2 presents the operation sequences for each type of product.

Table V-2: Product operation sequences

o_{ip}	B-product	E-product	L-product	T-product	A-product	I-product	P-product
1	plate loading	plate loading	plate loading	plate loading	plate loading	plate loading	plate loading
2	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting
3	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting	axis mounting
4	axis mounting	axis mounting	axis mounting	r_comp mounting	axis mounting	I_comp mounting	r_comp mounting
5	r_comp mounting	r_comp mounting	I_comp mounting	L_comp mounting	r_comp mounting	screw_comp mounting	L_comp mounting
6	r_comp mounting	r_comp mounting	I_comp mounting	plate unloading	L_comp mounting	plate unloading	plate unloading
7	I_comp mounting	L_comp mounting	screw_comp mounting		I_comp mounting		
8	screw_comp mounting	plate unloading	screw_comp mounting		screw_comp mounting		
9	plate unloading		plate unloading		plate unloading		

2.3. Machine data

The AIP cell is composed of 7 machines as follows: a loading/unloading unit (m_1), three assembly workstations (m_2, m_3 and m_4), an automatic inspection unit (m_5), a recovery unit (m_6), which is the only manual workstation in the cell, and an optional assembly workstation (m_7) that can be used to increase the machine-sequence flexibility. From these machines, only five (m_1, m_2, m_3, m_4 and m_7) are used in this case study, then ($R = 5$).

As depicted in Figure V-2, each machine has an input buffer. Due to conveyor derivation dimensions, the maximal input buffer capacity CW_{m_r} is setup at two shuttles for m_3, m_6 and m_7 and only one for the rest of machines. Herein, the raw material buffer is considered with unlimited capacity. The FIFO rule implemented to control product sequencing within machines' input buffers (Section IV-4.2) is the consequence of this technical restriction, given that input buffers can only accommodate shuttles sequentially in arrival order. Figure V-6 shows some of these machines.
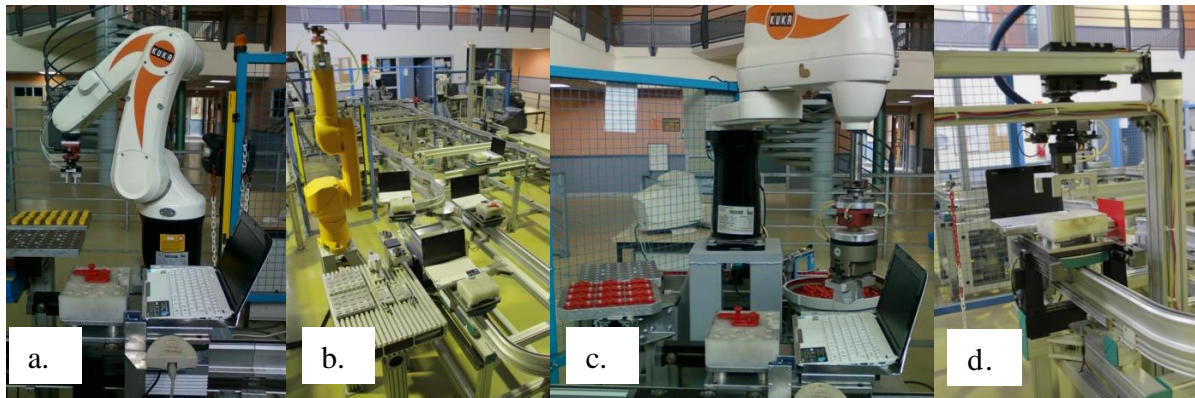
Figure V-6: a. machine m_2, b. machine m_7, c. machine m_4, and d. loading/unloading unit

Table V-3 presents the processing times (in seconds) of supported operations. An empty cell in the table means the operation is not supported by the machine. From Table V-3 it is possible to deduct the machine-sequence flexibility given the operation redundancy. For the FMS model this processing times are considered deterministic.

Table V-3: Processing times (in seconds) of operations

Operation (o_i)/processing time($\tau^i_{m_r}$)	m_1	m_2	m_3	m_4	m_7
axis mounting		20	20		20
r_comp mounting		20	20		20
screw_comp mounting			20	20	20
L_comp mounting		20		20	20
I_comp mounting			20	20	
plate loading	10				
plate unloading	10				

3. Decisional entities

This section describes the technological implementation of each decisional entity. This implementation has been possible due to collaboration with Cyrille Pach during the development of its holonic control architecture (Pach, 2013). Local decisional entities descriptions are given first, followed by the global entity. Further details on this implementation can be found in Appendix G-I and G-II.

3.1. Transport local decisional entity $vLDE_j$

The transport local decisional entity is composed of the transport device (i.e., the shuttle), the matrix pallet that supports the product and a processing unit in which the control, interaction and information storage modules are implemented (a. in Figure V-7).
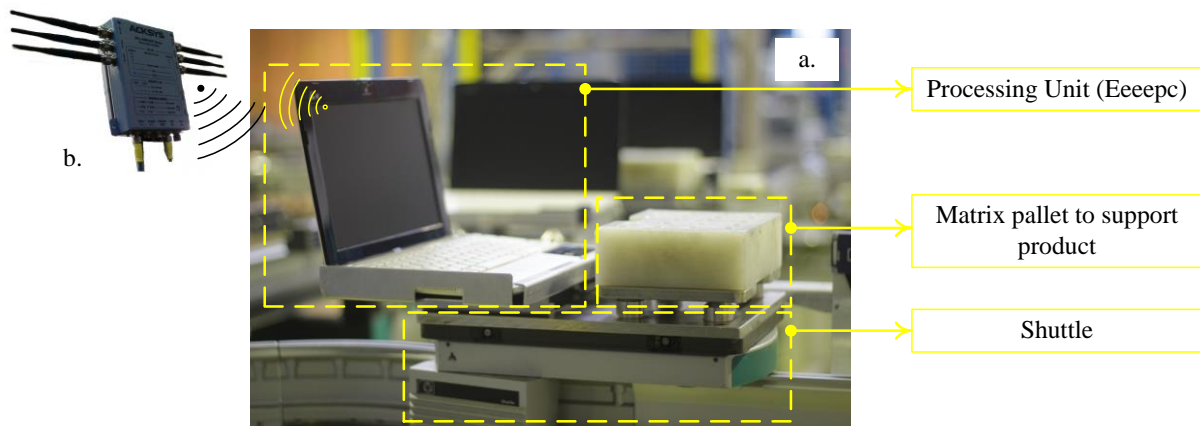
Figure V-7: a. The vLDE, and b. The wireless access point

The transport local decisional entity ($vLDE$) was conceived following the principles of "active" or "intelligent products", in which the physical product capabilities are augmented with informational, communicational and decisional capacities (Sallez, 2012). To this end, the processing unit is instrumented with a portable computer (for a proof-of-concept version), an Eeepc (AsusTek, 2014), given its portability and capabilities. Within this processing unit, the control module (Section IV-4.1.1) was developed in Java programming language and since the Wago® controllers communication protocol supports Modbus TCP, the Jamod class package was used to implement the interaction module (Section IV-4.1.2). By means of a wireless access point (b. in Figure V-7) and an Ethernet network, $vLDE$s can communicate with Wago® controllers managing transfer gates, machines and the global decisional entity. The Jamod class package allows $vLDE$s to read and write to Wago® controllers variables.

3.2. Machine local decisional entity $mLDE_r$

Seven (7) Wago® controllers (a. in Figure V-8) manage each of the AIP cell's machines (b. in Figure V-8). The Wago® controller handles the machine's input buffer, the list of operations supported by the machine, the attractiveness value under the potential fields approach and the information exchange with $vLDE$s when they arrive at the machine and request for operations. In addition, the Wago® controller ensures the information exchange with the machine controller. Consequently, the control, interaction and information storage modules of a $mLDE$ are all implemented in the Wago® controller.
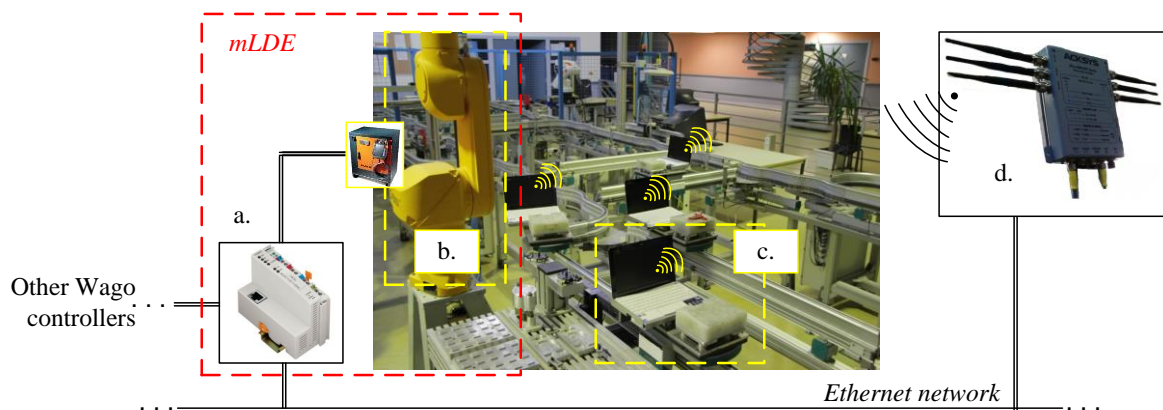
Figure V-8: The mLDE, a. The Wago® controller, b. The machine and its controller, c. vLDEs, and d. The wireless access point

As in the previous case, $mLDE$s and $vLDE$s use wireless connections to exchange information with each other (c. in Figure V-8). Instead, all 18 Wago® controllers exchange data through a wired connection, so potential fields can go around the entire AIP cell and $vLDE$s can gather machines' attractiveness values at any decisional node. More details on the potential fields approach (PFA) implemented at the AIP cell can be found in Appendix E and (Zbib et al., 2012; Pach et al., 2012).

3.3. Global decisional entity $pGDE_1$

For now, the $pGDE_1$ control module was programmed in Java programming language as classes that can be arranged in different ways according to the configuration of the optimization module and the global objective function. A set of methods to translate and pass variables, as well as to call NetLogo simulations, were also programmed in Java to ensure the integration of NetLogo with the rest of the $pGDE_1$'s control module. The Netlogo simulator and the Java programs run on a PC Intel(R) Core(TM) i5-3317U CPU@ 1.70Ghz with 4.00 GB of RAM memory. Appendix G-III describes in detail the configuration of the AIP cell

model into NetLogo. Figure V-9 presents the user interface of the NetLogo simulator that models the AIP cell.
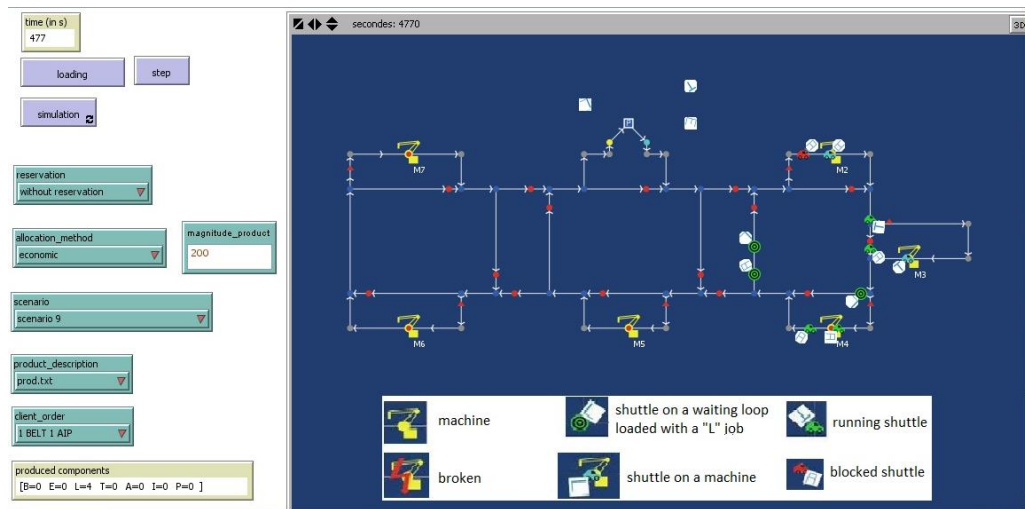
Figure V-9: The NetLogo graphical user interface for the AIP PRIMECA assembly cell

4. Static simulation study

The simulation study presented in this section was carried out to evaluate the proposed approach under static scenarios where no perturbations occur. Therefore, all theoretical values presented in the previous sections of this chapter apply. In Appendix H-I and H-II results from a static simulation study aiming to evaluate the efficiency of the genetic algorithm and the GA-ATC, decoupled and coupled, are reported.

4.1. General description of the static simulation study

The objective of this simulation study is to evaluate the efficiency of the proposed approach to deal with myopic decisions concerning release sequence and machine allocation sub-problems. As presented in the previous chapter, the product routing sub-problem is dealt exclusively by the local decisional entities, using a myopic decision-making algorithms. For this study, three independent instances of $pGDE_1$ were conceived and evaluated. As described in Table V-4, the optimization module (as explained in Section IV-4.3.1) is reconfigured to deal with myopic release sequences, myopic machine allocations or both types of myopic decisions. For this study, the coercive interaction is the only interaction mode that was evaluated.

For a better analysis of our approach, another myopic policy, the classical First Available Machine (FAM), was also used instead of PFA. The availability of a machine was determined by its remaining busy time, which includes the product on the machine and those in the

queue. Once a machine has been chosen, the product calculates the Shortest Path to reach the machine. Compared to PFA, FAM is more myopic regarding machine and routing choices since they are both stepwise and, sequentially. Regarding product release sequences, both rules are equally myopic. The three instances were evaluated for both decision-making algorithms, PFA and FAM.

Table V-4: Instances evaluated

Instance	Myopic Decision	Optimization Mechanism
Instance A	Release Sequence	Arrival-Time Control (ATC)
Instance B	Machine Allocation	Genetic Algorithm (GA)
Instance C	Release Sequence and Machine Routing	Arrival-Time Control and Genetic Algorithm (GA-ATC)

4.2. Factors and assumptions for the static simulation study

To evaluate the three instances presented before, three experimental factors were considered as seen in Table V-5. The first factor, $|R'|$, represents the set of active redundant machines. This factor allowed us to evaluate our approach with two levels of machine-sequence flexibility. The second factor, $|P|$, is the size of the production order. A production order is composed of a set of products with no constraints of product mixture. This factor allowed us to evaluate the scalability of the proposed approach. Last, the type of local decision-making algorithm, PFA or FAM, was also considered as an experimental factor.

Results of the three instance problems were compared with the results obtained from simulating the local control level as it worked with no global decisional level. This would be the case in which the AIP cell would be controlled by a fully heterarchical approach (FHFMS).

Table V-5: Factors for the static simulation study

Factors	Levels		Values
$|R'|$	2	3	m_2,m_3,m_4
		4	m_2,m_3,m_4,m_7
$|P|$	5	4	1*(BELT)
		8	1*(BELT),1*(AIP), 1*(B)
		16	2*(BELT),2*(LATE)
		24	3*(BELT),4*(AIP)
		28	4*(BELT),4*(AIP)
Local decision-making algorithm (*vLDE*)	2	1	PFA
		2	FAM

Since there are experimental factors with more than two levels, a factorial design was executed resulting in 20 design points for each instance in Table V-4. All the experimental results are reported in Appendix H-III. The experiments were executed as follows:

- ✓ Each design point for the FHFMS case had only one replication (given the low variability provided by the simulation model), but design points for other instances were replicated 10 times (10 independent trials) and mean values were used for the comparison.

- ✓ Each replication was independent and finished when the solution converged (ε=0.1 and remained steady for *itermax*=5 in Eq. IV-11).

- ✓ There was no warm-up period and the cell was considered empty at the beginning and at the end of each simulation replication.

- ✓ Simulation of the FHFMS case terminated when all the products had been manufactured.

- ✓ For comparing, the gaps between the FHFMS and the proposed problems instances are calculated as in Eq. V-1, where *GOF* (the CTV or MSD') is the global objective function and *LDMA* is the local decision-making algorithm, PFA or FAM.

$$Gap = \frac{GOF_{Instace-LDMA} - GOF_{FHFMS-LDMA}}{GOF_{FHFMS-LDMA}} * 100\% \qquad \text{Eq. V-1}$$

4.3. Simulation results for the static scenario

4.3.1. Results for Instance A

The control module of $pGDE_1$ for this instance is presented in Figure V-10. For analyzing the myopic behavior related to the release sequence sub-problem, *P* contains a certain product mixture (random combination of product types) as described in Table V-5. In the ABS (Agent-based model and simulation environment) the FHFMS approach (i.e., PFA or FAM) deals with machine allocation and product routing decisions.
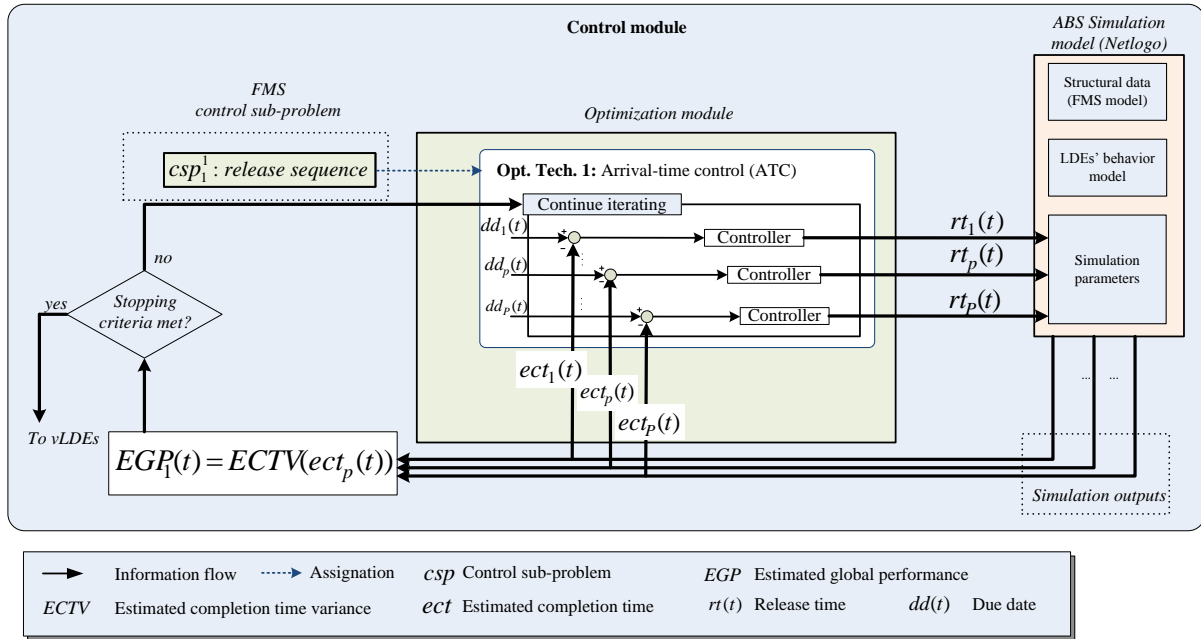
Figure V-10: The control module of $pGDE_1$ for problem instance A

As seen in Figure V-11, the FHFMS approaches are myopic to the release sequence, as expected. The arrival-time control (ATC) is capable of improving the CTV for both PFA and FAM, up to FMS saturation (28 products when |R'| =3). There are two main contributions of this simulation-optimization configuration. First, the balance between decisional levels since the global level partially interacts with the local level and only deals with one FMS sub-problem. Reactivity to internal perturbations is the ensured by the FHFMS approach, while ATC deals with incoming production orders and related perturbations or changes (i.e., urgent orders). Second, the low complexity of the ATC allowing important improvements without a prohibitive computing cost (see Appendix H-III).
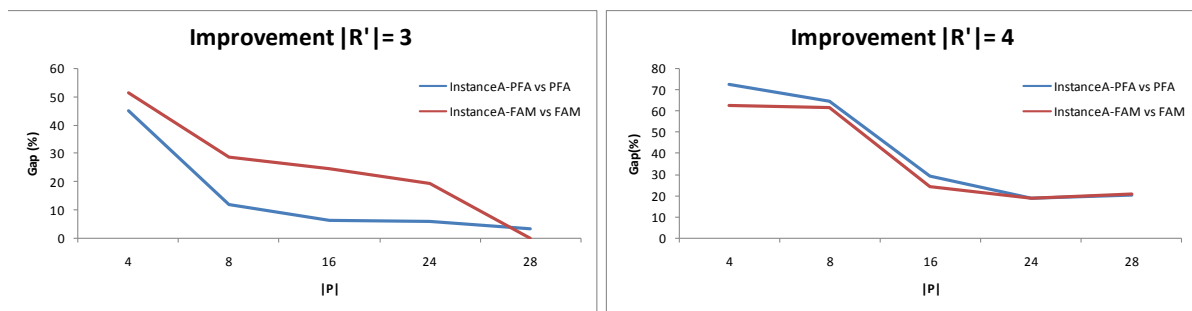
Figure V-11: Improvement by tackling the release sequencing sub-problem

Figure V-12 shows the convergence of the solution for the 4-product problem, |R'|=4, using FAM rule. The total time for this case is 10 seconds, including the time necessary to launch the NetLogo model with the static parameters (almost 3 seconds). Since the dynamic

102

parameters sent at each iteration only include the product release times, each iteration takes less than one second for this specific case.
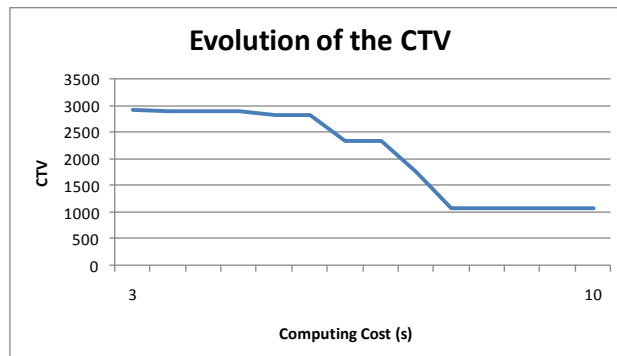
Figure V-12: Solution convergence of the ATC

4.3.2. Results for Instance B

The control module of $pGDE_1$ for this instance is shown in Figure V-13. In order to isolate the machine-sequencing sub-problem, P is composed of the same type of product. Product "A" was chosen because the AIP cell provides up to 13 different machine sequences when |R'|=4. In this configuration, the GA works until convergence and proposes machine sequences to the ABS for each product (smr_p Section IV-2.5). The ABS, in turn, simulates the product interactions due to product routing decisions and outputs the final completion times. Solution evaluation between then GA and the ABS continue until a stable solution is found.

The main contribution of this simulation-optimization configuration is that the global decisional entity deals with the combinatorial problem of machine sequences and PFA or FAM deals with product routing dynamics, which are highly dependent on product interactions and physical variables such as the transportation device speed, number of products, conveyor capacity and flexibility. The purpose of the ABS is to simulate such dynamics. Reactivity to all kinds of transportation issues is then ensured by the myopic rule, while the GA deals with machine and order issues.
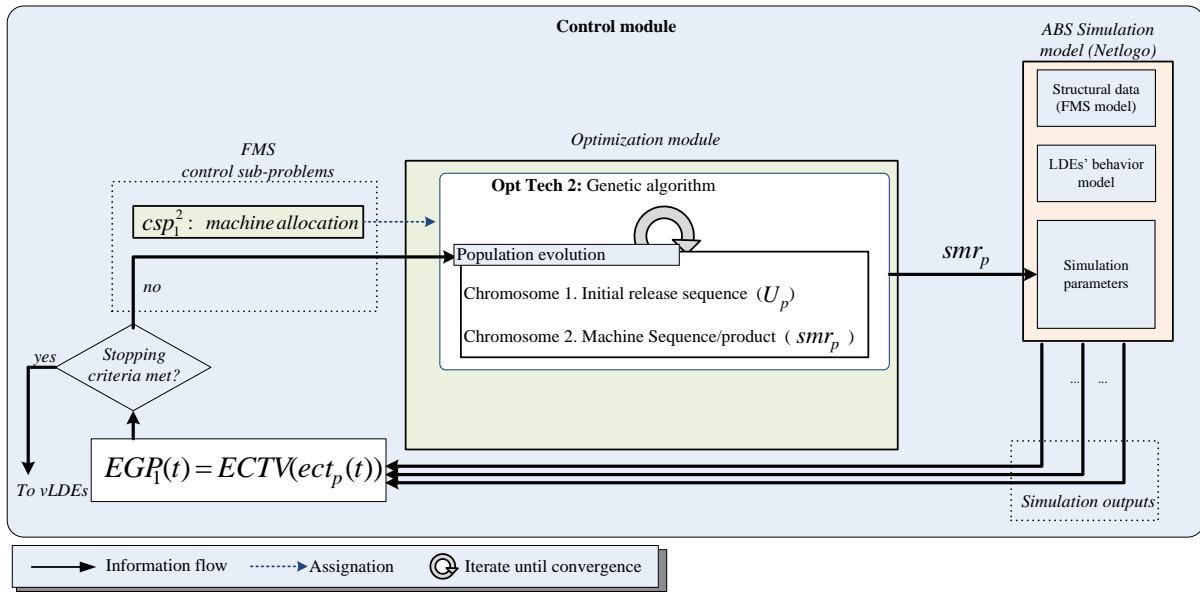
Figure V-13: The control module of $pGDE_1$ for problem instance B

Figure V-14 shows the improvement curves for an increasing number of products and the two levels of machine-sequence flexibility in regards to PFA and FAM. The results for PFA and FAM were the same at each design point, since both rules are only myopic for the material-handling sub-problem, and both end up using the shortest paths, since the cell does not offer that much material-handling flexibility, especially when |R'|=3. The minimum improvement obtained was around 79%, which demonstrates the high sensitivity of PFA and FAM to machine-selection decisions. This sensitivity is more evident when machine-sequence flexibility is increased.
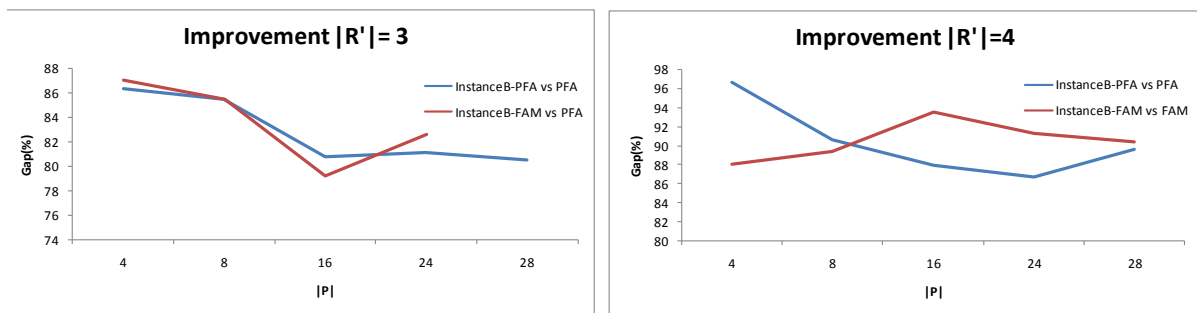
Figure V-14: Improvement by tackling the machine allocation sub-problem

Figure V-15 shows the convergence of the GA with |P|=8, FAM rule and |R'|=4. Since the GA finds a good solution in the first iteration, the GA-ABS loop converges rapidly at *itermax* iterations.

104
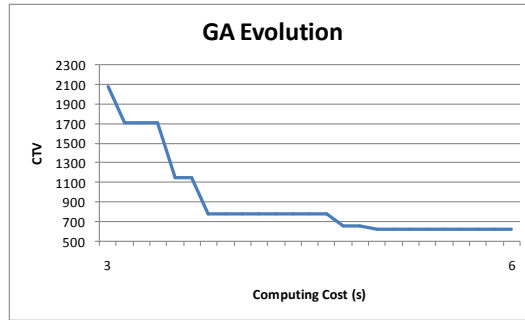
Figure V-15: Convergence of the CTV for the first iteration of the GA-ABS loop

For the last design point (28 products) with the FAM rule and low machine-sequence flexibility ($|R'|=3$) a deadlock appeared. Since all the products were in the FMS at the same time, predatory behavior emerged when a spot in an input buffer became available. At that point, all the products chose the same machine and overpopulated the same region on the conveyor. This is an issue of heterarchical approaches that, in this case, does not arise with the solution proposed by the global level.

Despite the increasing complexity of the machine allocation sub-problem, meta-heuristics (in this case GA) are a good choice for the global decisional level since the computing cost is not prohibitive (see Appendix H-III) and a certain level of reactivity can be preserved. Nonetheless, this can become an issue if the machine-sequence flexibility is improved or the number of products becomes much larger.

4.3.3. Results for Instance C

The control module of $pGDE_1$ for this instance is shown in Figure V-16. The two optimization mechanisms were coupled as described in Section III-4.2.1 and worked as follows: the GA proposes a population of solutions (machine sequences per product) to ATC, which in turn works on the release sequences until its convergence. The ATC returns the completion times to the GA which calculates the CTV for each individual (i.e., its fitness). The GA continues to evolve the population until a steady solution is found. The best solution is then sent to the ABS to obtain the final completion times, which might change due to local product routing decisions. This cycle is repeated until solution convergence.

For this problem instance, product orders are the same as in instance A. Improvement curves are reported in Figure V-17, comparing results from instance C with those of instance A and the FHFMS approach. The combination of GA-ATC not only improves the FHFMS solution, but also exceeds the improvement proposed by ATC alone. To deal with perturbations, in this case, the GA will deal with machine allocation issues and ATC with production order issues. Indeed, this configuration makes the coercive interaction more

significant but it achieved improvements of no less than 75% and 65% for both levels of machine-sequence flexibility, respectively. Hence, such improvement makes the computing cost worthy.
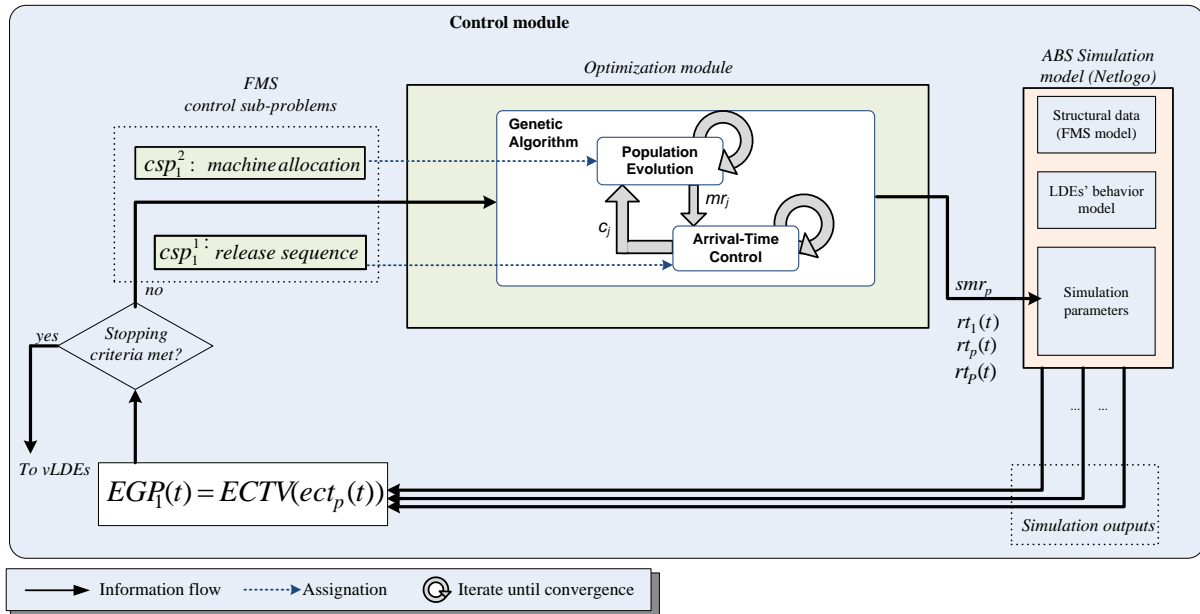
Figure V-16: The control module of $pGDE_1$ for problem instance C
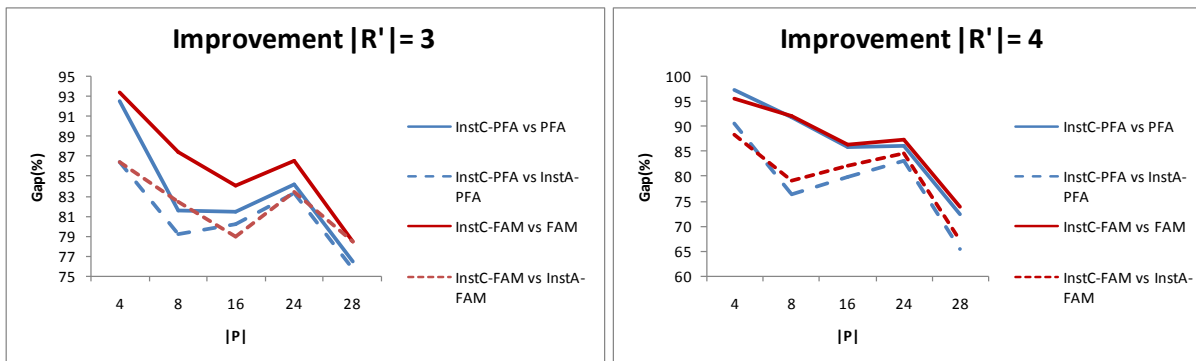
Figure V-17: Improvement by tackling the machine allocation and the release sequencing sub-problems

Figure V-18 shows the Gantt charts for the 8-problem products, FAM rule and |R'|=4 (one of the 10 trials). From these diagrams it is possible to see the myopic machine allocation of the FAM rule (FHFMS) and the resulting longer queuing and transport times (a. in Figure V-18). Indeed, the combination of myopic decisions for all sub-problems results in large deviations in product completion times, thus a bigger CTV. On the contrary, $pGDE_1$'s solution (b. in Figure V-18) balances better the machine weights, resulting in all the products finishing within short periods.
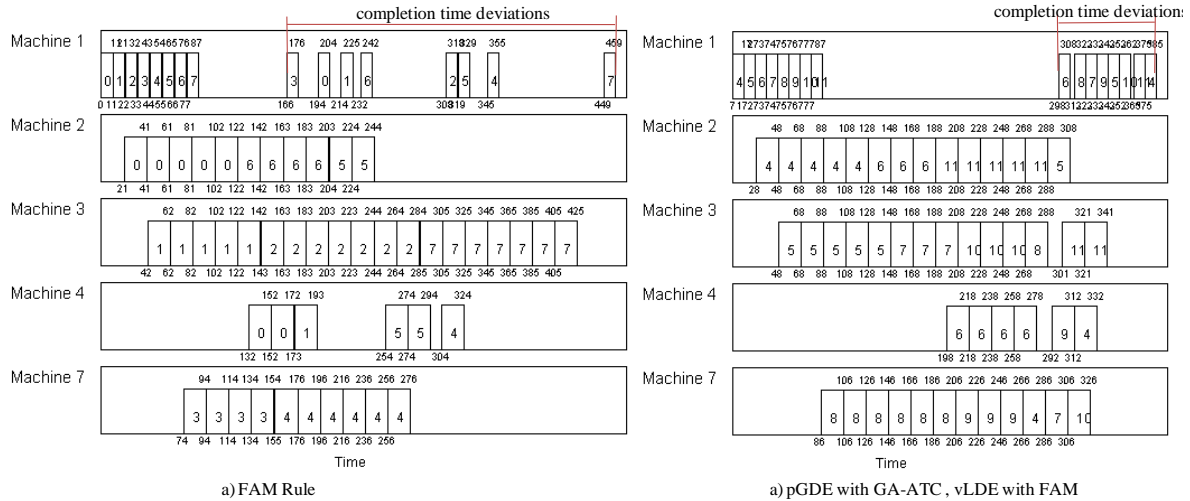
Figure V-18: Gantt charts

For the same trial, Figure V-19 shows the rapid convergence (in terms of computing cost) of the GA-ATC coupling. Although computing cost increases with problem complexity (see Appendix H-III), a non-negligible time is lost due to technical issues related to the interconnection between Java and the ABS platform (parameter exchange at each iteration).
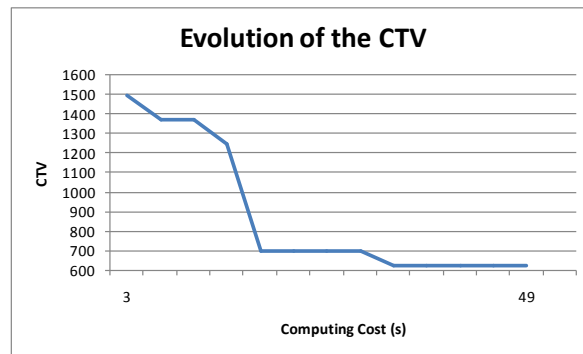
Figure V-19: Evolution of CTV within the GA-ATC loop

5. Dynamic simulation study

The main purpose of the dynamic scenario is to evaluate our approach in the presence of perturbations. From the possible internal and external perturbations reported in Trentesaux et al. (2013), we chose a maintenance task that arrives unexpectedly (scenario #PS9). This scenario simulates the limited machines' reliability. Once the machine is out for maintenance, the machine remains down for certain maintenance time that in this case depends on production volume. Herein, the affected machine is m_2. The necessary parameters, for this scenario are detailed in Table V-6. In this simulation study, neither the optimization module in the global decisional entity nor the interaction modes between global and local decisional

entities are adapted to overcome the perturbation. The optimization module (i.e., GA-ATC coupled) and the interaction mode (i.e, coercive) remained the same throughout all simulation time.

Table V-6: Parameters for the dynamic scenario

|P|	Product Mixture	Due Date (s)	[1]Duration of Maintenance (s)	[2]Maintenance start time
4	1 LATE	211	100	5
8	1 BELT + 1 LATE	303	200	5
16	2 BELT + 2 LATE	523	400	10
24	3 BELT + 3 LATE	658	600	20
28	3 LATE + 4 BELT	761	700	30

Given that local decision-making algorithms, PFA and FAM, were not designed to deal with due dates as local objectives, we cannot compare the aforementioned problem instances with the fully heterarchical approach (FHFMS) as did before. Then, for this dynamic scenario we compared instance C with instance A, to analyze the improvement gained when the machine allocation is dealt by the GA at the global level. Figure V-20 shows the dynamic behavior when the global decisional level (*GDL*) takes the configuration of problem instance C (GA-ATC).
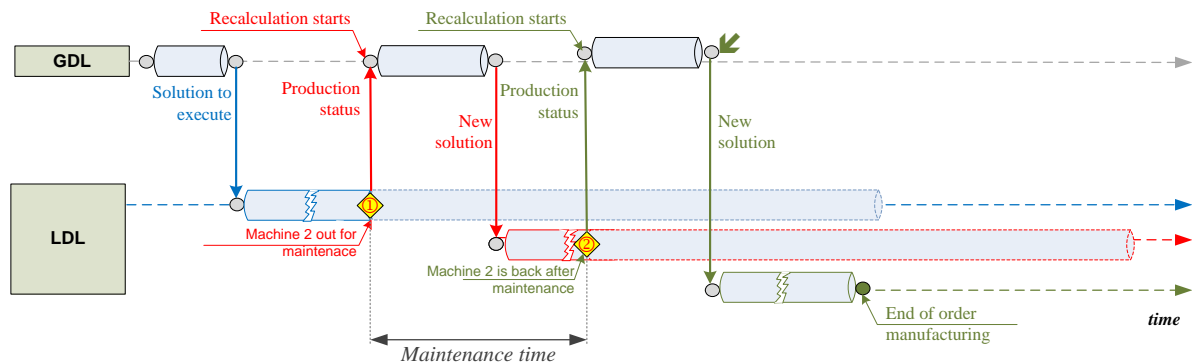
Figure V-20: Dynamic behavior under a maintenance perturbation

At first, the *GDL* calculates a solution and predict certain MSD'. The solution to execute is composed of a release sequence and machine sequences for each product. After certain execution time, m_2 breaks down and require maintenance (① in Figure V-20) and the global level triggers the GA-ATC to recalculate a new solution. For now, either the product in m_2 or in m_2's input buffer informs the global level and other *vLDE*s that a recalculation is needed. When maintenance has finished and the machine becomes available (②in Figure V-20) the recalculation process is triggered again. In this case, it is the global level that monitors the concerned machine and triggers recalculations. Other ways for recalculation triggering and handing will make part of future developments in association with a dynamic reconfiguration of the global decisional entity.

The difference between problem instance C and A is that in the latter both events (i.e., machine out for and machine back from maintenance) are taken care by PFA and the global level is not required. Table V-7 and Figure V-21 report the results and the gap (%) between problem instances A and C. The values reported represent the final MSD'.

Table V-7: Final MSD results for the dynamic scenario

|P|	Product Mixture	Instance A (MSD')	Instance C (MSD')	Gap (%)
4	1 LATE	83,5	69,7	20
8	1 BELT + 1 LATE	153,4	146,1	5
16	2 BELT + 2 LATE	301,5	286,5	5
24	3 BELT + 3 LATE	464,6	458,7	1
28	3 LATE + 4 BELT	478,3	474	1

From these results, it can be conclude that when the AIP's machine-sequence flexibility is affected for a significant amount of time (i.e., up to 42% of the total production time), using an optimization algorithm (i.e., GA) to deal with the machine-allocation sub-problem does not provide that much of improvement. In the best case the difference between both instances is around 20%. More, such improvement is drastically reduced when production volume increases, given that the AIP cell becomes saturated and there are not that much machine allocation choices for the number of products. Hence, PFA ended up offering the same performance than GA with the advantage of less processing time and locally made decisions.
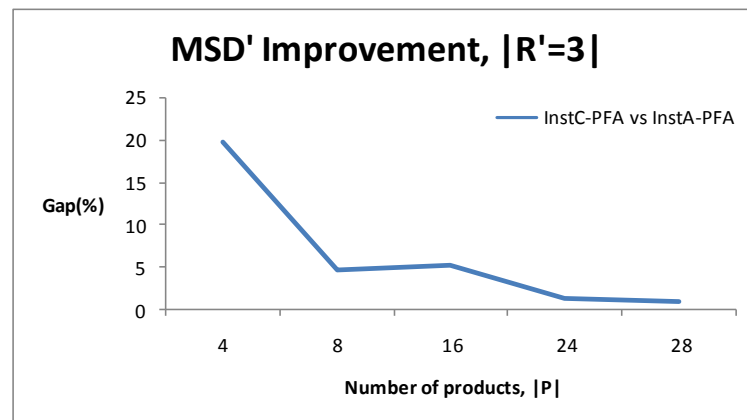
Figure V-21: Gap between the two problem instances, C and A

One important conclusion of this dynamic study is that a dynamic reconfiguration of the optimization module would allow the passage between instance C and A, offering almost the same performance with less computing cost. To achieve such dynamic reconfiguration it would be necessary to clearly establish the conditions under which each optimization technique is required, so the global decisional level is used only when a significant performance gain can be obtained. In addition, such dynamic reconfiguration would allow the local decision-making algorithm to take its role in improving reactivity. Other possibilities

regarding the interaction modes are also envisaged as explained later on (see Conclusions and Further Work).

6. Hardware-in-the-loop experimental study

The experimental study reported in this section was carried out in the AIP cell located at Valenciennes University. The objective of this study was to evaluate the real-time behavior of our approach, under normal and abnormal conditions. Only instance C was chosen for this experimental case given its efficiency. Since the AIP cell only counts with 10 shuttles, the two first cases proposed in Table V-6 (Section V-5) were carried out.

6.1. Hardware-in-the-loop results under normal conditions

Table V-8 reports results under normal conditions. The deviation between the predicted MSD' and actual MSD' values were between 8% and 13%. This difference is due to the fact that the simulation model is based on deterministic transport times and it does not take into account response delays due to communication between *vLDEs* and Wago$^{\circledR}$ controllers, mechanical issues related to transfer gate turning and other non-modeled times, e.g., time from shuttle storage to loading, time from input buffer to machine position unit, etc. In spite of this expected deviation, the global decisional level was able to predict a certain behavior for *vLDEs* and global performance value.

Table V-8: Results from the experimental results - normal conditions

|P|	Predicted MSD'	Actual MSD'	Gap (%)
4	46,9	51	8,8
8	81,5	92,4	13,4

6.2. Hardware-in-the-loop results under abnormal conditions

The maintenance scenario with 4 and 8 products was also implemented in the AIP cell. This time, maintenance times were reduced to 40 and 60 seconds to have more product interactions. Table V-9 reports results for these cases. The difference between the predicted MSD' and actual values ranged from 17% to 20% given that more shuttles are present at the same time within the AIP cell. More, during M_2's down time, products only have M_3 for the axis-mounting and r-component mounting so a predator behavior emerged in the vicinity of that machine. Differences between the simulation model and reality are evidently the main sources of deviation.

Table V-9: Results from the experimental results - abnormal conditions

|P|	Predicted MSD'	Predicted MSD' when M2 down	Predicted MSD' when M2 active	Actual MSD'	Gap (%)
4	38,2	91,1	66,6	78,2	17,3
8	73,3	188,3	134,5	161,4	19,9

For these small production orders the computing costs incurred by the global decisional entity were not prohibitive, remaining around 1 seconds for the 4-product order and 6 seconds for the 8-product order. Figure V-22 depicts the entire experiment.

In Figure V-22 Gantt charts on top show the predicted solutions, Figure V-22.a shows the actual production sequence (Gantt chart at the bottom) and Figure V-22.b plots the evolution of the predicted MSD' obtained by $pGDE_1$'s optimization module every time a calculation is triggered (in red the evolution of the average MSD' and in blue the evolution of the best MSD') and.

The Gantt chart displaying the actual production sequence was made by collecting information from the information storage module of each product. This information contained all the operation start and finish times and the actual machines where the operation was executed. It is then possible to see the small differences between simulation results and hardware-in-the-loop results given the differences between the actual AIP cell and its model. For instance, the time required from the machine's input buffer to the machine was considered into transport time while in reality it can take a few seconds, especially for the loading machine.

a. Final production sequence   *b. MSD evolution*
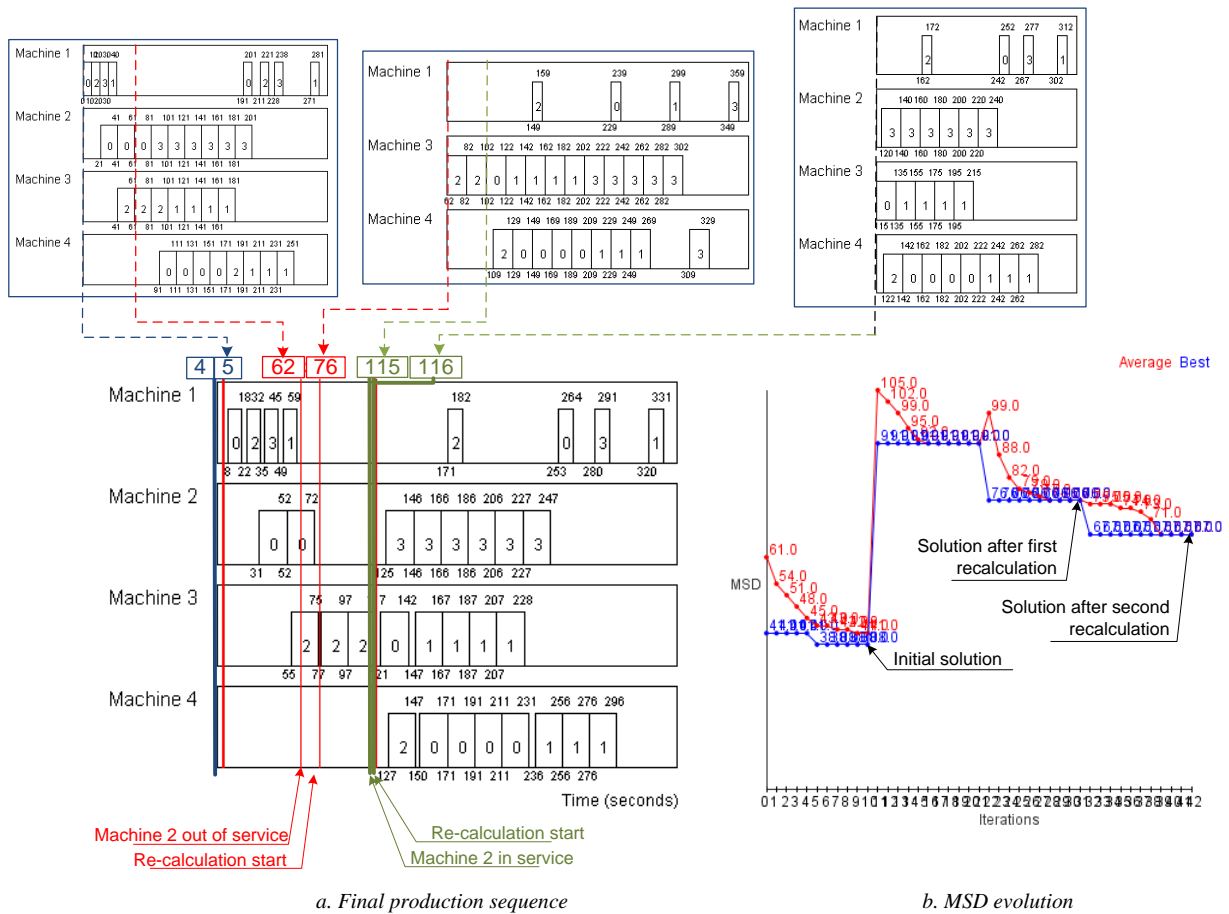
Figure V-22: Gantt charts and MSD' evolution for the 4-product order with maintenance perturbation

6.3. Return of experience

The fact that our approach could be implemented in a real assembly cell was a satisfactory achievement. A significant programming effort was made because four different programming environments were handled, MatLab for preliminary tests, NetBeans for Java language programming, CodeSys® for Wago® controllers and NetLogo® for the agent-based simulation (work entirely executed by Thérèse Bonte, Research Engineer at TEMPO-PSI). Once at the AIP cell, the debugging process took a significant amount of time. Indeed much work needs to be done in order to improve the current developments so the control system becomes more robust to several sources of perturbations, e.g., communications, mechanical and electrical issues, collisions, etc. An interesting feature of the current implementation is related to using an EeePC as processing unit for the *vLDE*. Such configuration allowed the debugging process to be done more easily in comparison with other devices, for instance without a screen.

7. Summary

This chapter mainly described the implementation on the proposed semi-heterarchical architecture to control a real assembly cell. After describing the AIP PRIMECA cell, the implementation of each one of the proposed decisional entities was described. Two campaigns were launched to evaluate the proposed concepts, the first relying only on simulation results and the second on the real assembly cell (hardware-in-the-loop).

The three problem instances proposed for the static scenario proved that the global decisional entity may have different configurations to deal with different myopic decisions. Each configuration reached non-negligible improvements compared to the myopic decision-making algorithms within a fully heterarchical architecture. In addition, by changing the machine allocation algorithm from the potential fields approach to a myopic priority rule (FAM), it was possible to show that the proposed approach is flexible and adaptable to different local decision-making algorithms. More, the fact that two optimization techniques were used to deal with two myopic decisions showed that dealing with myopic behavior from a granular perspective allows having modular and reconfigurable control architecture. Furthermore, the coercive interaction mode demonstrated its efficiency in dealing with myopic decision making, easing the execution of solutions obtained at the global decisional level. Figure V-23 positions the proposed SHFMS against other approaches reported in Chapter II. Mainly, the proposed features regarding the possible roles adopted by the simulation-based optimization techniques, the interaction modes and the limited myopic behavior reduction scope make possible to achieve a reduction of the impact of myopic behavior while striking a balance between global performance and reactivity. However, these conclusions can be inferred from separate evaluation of three different instances of the architecture. Therefore, a dynamic strategy to take advantage of the proposed features is envisaged to exploit all the benefits of the proposed architecture. In addition, open questions still remain concerning the evolution of the curves and the definition of an optimal myopic behavior region.

From these simulation results, it can also be concluded that performance improvement can be significantly dependent on FMS capacity. Based on results of Instance A, it is not worth it tackling myopic release sequences when the FMS reaches saturation. Conversely, dealing with machine-routing decisions can help to avoid deadlocks such as those that occurred with the FAM rule and a large number of products. In addition, coupling two optimization mechanisms can further improve the solution (Instance C vs. Instance A) with a certain compromise on reactivity.

Regarding the dynamic cases, the capability of the proposed approach to react to an internal perturbation was evaluated, under simulation and experimental study at the AIP.

113

From the simulation study it was possible to conclude that despite the fact that the global decisional entity was capable to react to the internal perturbation, a dynamic and possibly case-based reconfiguration strategy for the optimization module is necessary. In that way, the local decision-making algorithm can actually accomplish its role, reaching performance values similar to those proposed by the global level. This will definitely make part of the future work.
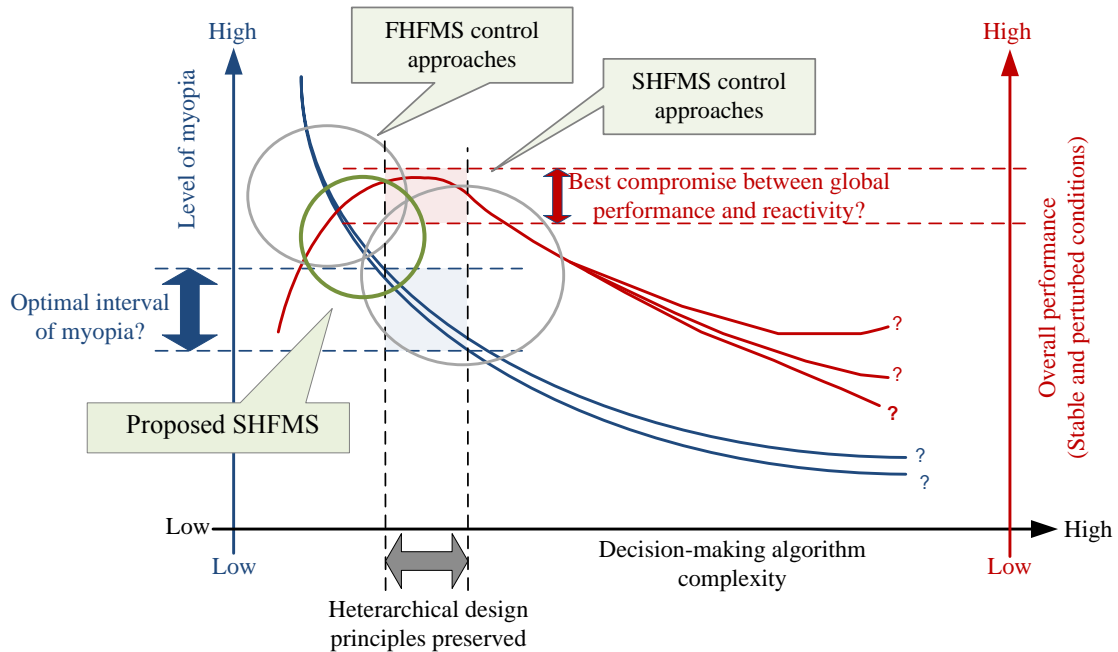
Figure V-23: The proposed SHFMS control approach

Results from the experimental campaigns on the real AIP cell (hardware-in-the-loop) confirmed the deviation caused by simulation model assumptions, communication delays, and deterministic values. However, an estimate of the actual global performance was able to be estimated beforehand. More general conclusions and further work are presented in the next part.

Conclusions and Further Work

Manufacturing efficiency is at the most important matter, especially in the current scenario in which manufacturing has to face globalized markets, product customization, higher client expectations, market volatility, and short product life-cycles among others. In the last decades, technological advances have permitted to rigid manufacturing systems to evolve toward more flexible manufacturing systems (FMS) capable to accommodate rapid changes, small batches and high variety of products. On the control of such flexible systems, new control architectures, paradigms, strategies and algorithms have appeared advocating for heterarchical architectures, low complex and highly reactive control residing on local decisional entities. However, in spite of promising advances, new issues have also emerged concerning their predictability, implementation costs, technologies and standards, as well as the guarantee of a minimal operational performance, in part due to myopic behavior of local decisional entities. Though such behavior has been recognized for some researchers as an important barrier for adopting heterarchical-based FMS (HFMS) control in industrial applications, until know this issue had not been formally studied in HFMS.

While analyzing other domains in which myopic decision making has been studied, similar aspects with FMS could be identified. Thus, our contribution starts by defining myopic behavior and identifying, defining and describing two dimensions of it, the social and temporal dimensions. In few words, myopic behavior results from the narrow visibility that each local decisional entity has on the current and future states of the FMS, given their focus on their own local objectives.

One of the conclusions that could be withdrawn after reviewing several papers concerning HFMS control was that until know most of works had focused on improving the global efficiency of HFMS control, implicitly dealing with myopic behavior. By doing so, features achieved by heterarchical principles, such as reactivity, adaptability and fault tolerance had been affected in regards to the underlying heterarchical approach. Therefore, our work part from an alternative perspective in which a global level, within a semi-heterarchical architecture, is configured to explicitly deal with myopic behavior resulting from those local control decisions. In the proposed semi-heterarchical architecture, the local decisional level ensures certain reactivity to perturbations, and the global decisional level focuses on myopic behavior and its impact on global performance. In regards to other semi-heterarchical approaches, our approach followed a more granular methodology in which the global

decisional level displays a flexible configuration to deal with myopic behavior of FMS control sub-problems. Therefore, myopic behavior can be reduced totally or partially depending on the number of control sub-problems that the global decisional level is configured for.

More, the proposed approach shows a modular configuration supported by simulation-based optimization (SbO) techniques. Different techniques into the global decisional level can coexist in order to reduce myopic control decisions. Such modularity is achieved by dedicated optimization techniques focused on reducing myopic decisions for each control sub-problem. In turn, given that some control sub-problems can be handled locally, the simulation model was used to determine the impact of local control decisions on the global objective function.

Furthermore, our approach also focused on the integration of solutions obtained at the global level into the local level. For that, three interaction modes, i.e., *coercive*, *limitary* and *steering*, between the two decisional levels were proposed in order to grant different autonomy levels to local decisional entities. Due to the modularity of the global decisional entity, the global decisional entity can take different roles (i.e., *resolving*, *evaluating*, *selecting*, *tuning* and *influencing*) and based on data obtained from the SbO technique(s), interaction modes can be different for each control sub-problem. Until know, the relationship between global and local levels had been understood as a whole. In our approach, such relationship is more granular so the local decisional entity may have different levels of autonomy for different control decisions. Consequently, our approach accepts configurations in which myopic behavior can be accepted for some control sub-problems (i.e., full local autonomy) and reduced for others (i.e., reduced local autonomy), favoring local reactivity and fault tolerance.

We were able to implement and evaluate our approach taking as a reference the AIP-PRIMECA cell at Valenciennes Universisty. At first static and dynamic simulation studies were executed and then hardware-in-the loop experimentations were carried out. From the static simulation study we could infer that myopic behavior affected importantly the global performance of local decision-making algorithms. Hence, significant improvements could be obtained with the proposed semi-heterarchical architecture under three different configurations of the global decisional entity. From the dynamic simulation study, it could be concluded that the global decisional entity could overcome an abnormal condition, always dealing with myopic control decisions and achieving improvements in terms of global performance.

The hardware-in-the-loop experiments allowed us to demonstrate a possible realistic implementation of our approach. An important programming effort was made since it was

necessary to work on different programming languages, communication protocols and networks, as well as with industrial controllers and conveying systems. Results from these experiments showed us that the simulation model was good enough and helped to obtain estimated global performances not only for preliminary studies but also for online experimentations. Expected deviations from simulation to actual results were confirmed.

Based on the requirements presented in Section II-6 and the return of experience from these evaluations, some research perspectives for the sort-, mid- and long-term could be identified. As mentioned before, our contribution relies on an architectural concept in which a global decisional entity with simulation-based optimization techniques deal with myopic decisions made in a local level. Indeed, several questions on this choice have to be made so other research paths can be undertaken. For instance, could it be possible to endow local decisional entities with more complex algorithms while respecting decentralization? Would those complex local algorithms take care of all decisions or a global decisional level would still be required? What type of complex algorithms may work in decentralized architectures? What type of entity interactions would apply, direct or indirect?

Since the proposed approach demonstrated promising results, the next step in the short-term could be to make the entire architecture dynamic and adaptive in the presence of internal and external perturbations. Mainly, a dynamic reconfiguration of the optimization module should be implemented according to the perturbation type and criticality. Hence, it is necessary to determine which control sub-problems should be handled by each decisional level and under which conditions. Indeed, with the current configuration, an evolution of interaction modes, from *coercive* to *limitary*, is achievable. In this way, the local decisional entity may experience different autonomy levels according to FMS conditions. It is thus possible to think on using a population of good solutions instead of only using the best solution given by the GA. A dynamic control strategy switching between the two interaction modes needs to be implemented. More for the mid-term, a second possibility for such dynamism and adaptability could be to implement optimization mechanisms that allow other roles for the global decisional entity. For instance, the *influencing* role can be achieved if an optimization technique is used to dynamically modify the machines' attractiveness values when using PFA in the local decisional level. The *tuning* role can also be envisaged if the product holds a more complex decision making rule in which other variables are considered, e.g., number of operations to be executed at the machine. Few modifications to the current configuration would allow evaluating these new approaches.

Continuing with myopic behavior, a shift between myopic behavior reduction to control can also be foreseen. A detailed study on myopic behavior at the local decisional level aiming for mathematical-based models should be done first so myopic behavior can be measured and

controlled. This would definitely allow us to have a dynamic architecture capable of changing its myopic behavior level to deal with abnormal conditions. Also, this study can be extended to other production-related problems with other control problems such as maintenance, inventory, quality, and personnel allocation in order to reach a complete model. To this end, it would be necessary to identify control sub-problems and the impact of myopic decisions for the global objective function related to each problem. The interaction between global decisional entities at the global level and with the local level needs to be studied given the multiple objectives and the target myopic behavior. Also, dynamic allocation of local decisional entities and resources to global decisional entities should be envisaged to achieve dynamic flexibility, instead of static flexibility as currently conceived in FMS.

An important aspect that is frequently neglected in FMS control is the human interaction with artificial decisional entities. Therefore, another perspective in the short and mid-term that allows us to follow a parallel research path is the integration of human entities in the global or in the local levels. For instance, for personnel scheduling problems, machine operators would be considered as local decisional entities while production managers would be placed in the global level. The challenge is to adapt the current *artificial* decisional entity to a *biological* decisional entity. In fact, putting the human-in-the-loop is not an easy task given the decentralized architecture, the high frequency of decisions, the difficulty of making behavior models so humans can have a global perspective of what is happening with decisional entities and the entire system, among others. Some work in such direction has already been started in collaboration with the Florida Institute of technology (Zambrano Rey et al., 2013). The fact that our approach counts on different interaction modes and roles of the artificial global decisional entity provides certain flexibility to deal with human interactions.

Last, in the long-term, our approach should also be tested in other contexts, e.g., other production objectives than JIT ones, other types of processes, and applied in other application areas such as healthcare engineering, supply chains, transport management. One interesting way to motivate industrialist to adopt heterarchical-based approaches is to take an industrial case, retake their current production decision-making algorithms and insert them into the proposed semi-heterarchical approach in conjunction with local decision-making algorithms. This exercise would show how to handle the migration between current hierarchical approaches and heterarchical-based approaches as well as the significant improvements in reactivity and fault tolerance that can be achieved.

References

3S-Smart Software, 2014. CodeSys http://www.codesys.com/ (visited 26/05/2014).

Adam, E., Berger, T., Sallez, Y., Trentesaux, D., 2009. An Open-Control Concept for a Holonic Multiagent System, in: Mařík, V., Strasser, T., Zoitl, A. (Eds.), Holonic and Multi-Agent Systems for Manufacturing, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 145–154.

Adam, E., Zambrano, G., Pach, C., Berger, T., Trentesaux, D., 2011. Myopic Behaviour in Holonic Multiagent Systems for Distributed Control of FMS. Trends Pract. Appl. Agents Multiagent Syst. 90, 91–98.

Aissani, N., Atmani, B., Trentesaux, D., Beldjilali, B., 2014. Extraction of Priority Rules for Boolean Induction in Distributed Manufacturing Control, in: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. Springer, pp. 127–143.

Al-Fawzan, M.A., Haouari, M., 2005. A bi-objective model for robust resource-constrained project scheduling. Int. J. Prod. Econ. 96, 175–187.

Anand, G., Ward, P.T., 2004. Fit, flexibility and performance in manufacturing: coping with dynamic environments. Prod. Oper. Manag. 13, 369–385.

Andersson, M.R., Sandholm, T.W., 2001. Leveled commitment contracts with myopic and strategic agents. J. Econ. Dyn. Control 25, 615–640.

Asadzadeh, L., Zamanifar, K., 2010. An agent-based parallel approach for the job shop scheduling problem with genetic algorithms. Math. Comput. Model. 52, 1957–1965.

AsusTek, 2014. EeePC http://eeepc.asus.com/ (visited 08/05/2014).

Atzori, L., Iera, A., Morabito, G., 2010. The Internet of Things: A survey. Comput. Netw. 54, 2787–2805.

Aytug, H., Lawley, M.A., McKay, K., Mohan, S., Uzsoy, R., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. Eur. J. Oper. Res., IEPM: Focus on Scheduling 161, 86–110.

Babiceanu, R., Chen, F., 2006. Development and Applications of Holonic Manufacturing Systems: A Survey. J. Intell. Manuf. 17, 111–131.

Back, T., 1994. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. Presented at the First IEEE Conference on Evolutionary Computation, pp. 57 –62 vol.1.

Bagchi, U., Sullivan, R.S., Chang, Y.-L., 1987. Minimizing Mean Squared Deviation of Completion Times About a Common Due Date. Manag. Sci. 33, 894–906.

Bajracharya, M., Howard, A., Matthies, L.H., Tang, B., Turmon, M., 2009. Autonomous off-road navigation with end-to-end learning for the LAGR program. J. Field Robot. 26, 3–25. doi:10.1002/rob.20269

Baker, A.D., 1998. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. J. Manuf. Syst. 17, 297–320.

Baker, K.R., Scudder, G.D., 1990. Sequencing with earliness and tardiness penalties: a review. Oper Res 38, 22–36.

Barbosa, J., Leitao, P., Adam, E., Trentesaux, D., 2013. Self-Organized Holonic Multi-agent Manufacturing System: The Behavioural Perspective. IEEE, pp. 3829–3834. doi:10.1109/SMC.2013.654

Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2012. Nervousness in Dynamic Self-organized Holonic Multi-agent Systems, in: Pérez, J.B., Sánchez, M.A., Mathieu, P., Rodríguez, J.M.C., Adam, E., Ortega, A., Moreno, M.N., Navarro, E., Hirsch, B., Lopes-Cardoso, H., Julián, V. (Eds.), Highlights on Practical Applications of Agents and Multi-Agent Systems, Advances in Intelligent and Soft Computing. Springer Berlin Heidelberg, pp. 9–17.

Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2013. Structural Self-organized Holonic Multi-Agent Manufacturing Systems, in: Industrial Applications of Holonic and Multi-Agent Systems. Springer, pp. 59–70.

Barbosa, J., Leitao, P., Pereira, A.I., 2011. Combining adaptation and optimization in bio-inspired multi-agent manufacturing systems. Presented at the 2011 IEEE International Symposium on Industrial Electronics (ISIE), pp. 1773 –1778.

Basnet, C., Mize, J.H., 1994. Scheduling and control of flexible manufacturing systems: a critical review. Int. J. Comput. Integr. Manuf. 7, 340–355.

Baykasoğlu, A., Özbakir, L., 2008. Analysing the effect of flexibility on manufacturing systems performance. J. Manuf. Technol. Manag. 19, 172–193.

Baykasoğlu, A., Özbakır, L., 2010. Analyzing the effect of dispatching rules on the scheduling performance through grammar based flexible scheduling system. Int. J. Prod. Econ. 124, 369–381.

Becker, R., Carlin, A., Lesser, V., Zilberstein, S., 2009. Analyzing myopic approaches for multi-agent communication. Comput. Intell. 25, 31–50.

Benaissa, K., Diep, D., Dolgui, A., 2008. Control of chaos in agent based manufacturing systems. Presented at the IEEE International Conference on Emerging Technologies and Factory Automation, 2008. ETFA 2008, pp. 1252 –1259.

Berger, T., Sallez, Y., Valli, B., Gibaud, A., Trentesaux, D., 2010. Semi-Heterarchical allocation and routing processes in FMS control: a stigmergic approach. J. Intell. Robot. Syst. 58, 17–45.

Blanc, P., Demongodin, I., Castagna, P., 2008. A holonic approach for manufacturing execution system design: An industrial application. Eng. Appl. Artif. Intell. 21, 315–330.

Boccalatte, A., Gozzi, A., Paolucci, M., Queirolo, V., Tamoglia, M., 2004. A multi-agent system for dynamic just-in-time manufacturing production scheduling. Presented at the 2004 IEEE International Conference on Systems, Man and Cybernetics, pp. 5548 – 5553 vol.6.

Böhnlein, D., Schweiger, K., Tuma, A., 2011. Multi-agent-based transport planning in the newspaper industry. Int. J. Prod. Econ. 131, 146–157.

Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm intelligence: from natural to artificial systems. Oxford university press New York.

Bongaerts, L., Monostori, L., McFarlane, D., Kádár, B., 2000. Hierarchy in distributed shop floor control. Comput. Ind. 43, 123–137.

Bordoloi, S.K., Cooper, W.W., Matsuo, H., 1999. Flexibility, Adaptability, and Efficiency in Manufacturing Systems. Prod. Oper. Manag. 8, 133–150.

Botti, V., Giret, A., 2008. ANEMONA A Mulit-Agent Methodology for Holonic Manufacturing Systems, Springer Series in Advanced Manufacturing.

Brandimarte, P., 1993. Routing and scheduling in a flexible job shop by tabu search. Ann. Oper. Res. 41, 157–183.

Brennan, R.W., 2000. Performance comparison and analysis of reactive and planning-based control architectures for manufacturing. Robot. Comput.-Integr. Manuf. 16, 191–200.

Brennan, R.W., Vrba, P., Tichy, P., Zoitl, A., Sünder, C., Strasser, T., Marik, V., 2008. Developments in dynamic and intelligent reconfiguration of industrial automation. Comput. Ind. 59, 533–547.

Brintrup, A., 2010. Behaviour adaptation in the multi-agent, multi-objective and multi-role supply chain. Comput. Ind. 61, 636–645.

Brooks, R.A., 1995. Intelligence without reason, in: The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents. Presented at the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), Sidney, pp. 25–81.

Browne, J., Dubois, D., Rathmill, K., Sethi, S.P., Stecke, K.E., 1984. Classification of flexible manufacturing systems. FMS Mag. 2, 114–117.

Buzacott, J.A., Mandelbaum, M., 2008. Flexibility in manufacturing and services: achievements, insights and challenges. Flex. Serv. Manuf. J. 20, 13–58. doi:10.1007/s10696-008-9052-x

Caramia, M., Dell'Olmo, P., 2006. Effective Resource Management in Manufacturing Systems Optimization Algorithms for Production Planning, Springer Series in Advanced Manufacturing. London.

Cardin, O., Castagna, P., 2009. Using online simulation in Holonic manufacturing systems. Eng. Appl. Artif. Intell. 22, 1025–1033.

Carvalho, M.M., Eskridge, T.C., Bunch, L., Bradshaw, J.M., Dalton, A., Feltovich, P., Lott, J., Kidwell, D., 2012. A Human-Agent Teamwork Command and Control Framework for Moving Target Defense (MTC2). Presented at the 8th Cyber Security and Information Intelligence Research Workshop, Oak Ridge.

Caumond, A., Lacomme, P., Moukrim, A., Tchernev, N., 2009. An MILP for scheduling problems in an FMS with one vehicle. Eur. J. Oper. Res. 199, 706–722.

Cavalieri, S., Garetti, M., Macchi, M., Taisch, M., 2000. An experimental benchmarking of two multi-agent architectures for production scheduling and control. Comput. Ind. 43, 139–152.

Chan, F.T.S., Chan, H.K., 2004. A comprehensive survey and future trend of simulation study on FMS scheduling. J. Intell. Manuf. 15, 87–102.

Cheng, R., Gen, M., Tsujimura, Y., 1996. A tutorial survey of job-shop scheduling problems using genetic algorithms–I. Representation. Comput. Ind. Eng. 30, 983–997.

Chirn, J.-L., McFarlane, D.C., 2000. A holonic component-based approach to reconfigurable manufacturing control architecture, in: Database and Expert Systems Applications, 2000. Proceedings. 11th International Workshop on. IEEE, pp. 219–223.

Chituc, C., Restivo, F., 2009. Challenges and Trends in Distributed Manufacturing Systems: Are wise engineering engineering systems the ultimate answer? Second Int. Symp. Eng. Syst. MIT Camb. Mass. 1–15.

Cho, S., Prabhu, V.V., 2007. Distributed adaptive control of production scheduling and machine capacity. J. Manuf. Syst. 26, 65–74.

Choudhary, A.K., Harding, J.A., Tiwari, M.K., 2009. Data mining in manufacturing: a review based on the kind of knowledge. J. Intell. Manuf. 20, 501–521.

Christensen, J., 1994. Holonic Manufacturing Systems: Initial Architecture and Standards Directions. Presented at the 1st Euro Wkshp on Holonic Manufacturing Systems.

Christo, C., Cardeira, C., 2007. Trends in intelligent manufacturing systems. Presented at the IEEE International Symposium on Industrial Electronics, 2007. ISIE 2007., IEEE, pp. 3209–3214.

Chryssolouris, G., 1992. Manufacturing systems: theory and practice, Mechanical Engineering Series. Springer-Verlag, New York.

Chryssolouris, G., Efthymiou, K., Papakostas, N., Mourtzis, D., Pagoropoulos, A., 2013. Flexibility and complexity: is it a trade-off? Int. J. Prod. Res. 51, 6788–6802.

Chryssolouris, G., Pierce, J., Dicke, K., 1991. An approach for allocating manufacturing resources to production tasks. J. Manuf. Syst. 10, 368–382.

Chu, Y., You, F., Wassick, J.M., 2014. Hybrid method integrating agent-based modeling and heuristic tree search for scheduling of complex batch processes. Comput. Chem. Eng. 60, 277–296.

Coffman, E.G., Yannakakis, M., Magazine, M.J., Santos, C., 1990. Batch sizing and job sequencing on a single machine. Ann. Oper. Res. 26, 135–147.

Collins Dictionary. http://www.collinsdictionary.com/ (visited 07/04/2014), 2014. . Collins Dict.

Colombo, A.W., 2005. Industrial agents: towards collaborative production, automation, management and organization. IEEE Ind. Electron. Soc. Newsl. 52, 17–18.

Colombo, A.W., Karnouskos, S., 2009. Towards the factory of the future: A service-oriented cross-layer infrastructure. ICT Shap. World Sci. View Eur. Telecommun. Stand. Inst. ETSI John Wiley Sons 65, 81.

Conway, R.W., Maxwell, W.L., Miller, L.W., 2003. Theory of scheduling. Dover Publications.

D'Souza, D.E., Williams, F.P., 2000. Toward a taxonomy of manufacturing flexibility dimensions. J. Oper. Manag. 18, 577–593.

Demir, Y., Kürşat İşleyen, S., 2013. Evaluation of mathematical models for flexible job-shop scheduling problems. Appl. Math. Model. 37, 977–988.

Dilts, D.M., Boyd, N.P., Whorms, H.H., 1991. The evolution of control architectures for automated manufacturing systems. J. Manuf. Syst. 10, 79–93.

Ding, H., Benyoucef, L., Xie, X., 2005. A simulation optimization methodology for supplier selection problem. Int. J. Comput. Integr. Manuf. 18, 210–224.

Dorigo, M., Birattari, M., 2010. Ant colony optimization, in: Encyclopedia of Machine Learning. Springer, pp. 36–39.

Du, J., Leung, J.Y.-T., 1990. Minimizing Total Tardiness on One Machine is NP-Hard. Math. Oper. Res. 15, 483–495. doi:10.1287/moor.15.3.483

Duffie, N.A., 1990. Synthesis of Heterarchical manufacturing systems. Comput. Ind. 14, 167–174.

Duffie, N.A., 1996. Heterarchical control of highly distributed manufacturing systems. Int. J. Comput. Integr. Manuf. 9, 270–281.

Duffie, N.A., 2008. Challenges in design of heterarchical controls for dynamic logistic systems, in: Dynamics in Logistics. Springer, pp. 3–24.

Duffie, N.A., Prabhu, V.V., 1994. Real-time distributed scheduling of heterarchical manufacturing systems. J. Manuf. Syst. 13, 94–107.

Eiben, A., Michalewicz, Z., Schoenauer, M., Smith, J., 2007. Parameter Control in Evolutionary Algorithms, in: Lobo, F., Lima, C., Michalewicz, Z. (Eds.), Parameter Setting in Evolutionary Algorithms, Studies in Computational Intelligence. Springer Berlin / Heidelberg, pp. 19–46.

El Khoukhi, F., Lamoudan, T., Boukachour, J., El Hilali Alaoui, A., 2011. Ant Colony Algorithm for Just-in-Time Job Shop Scheduling with Transportation Times and Multirobots. ISRN Appl. Math. 2011, 1–19. doi:10.5402/2011/165620

ElMaraghy, H.A., 2006. Flexible and reconfigurable manufacturing systems paradigms. Int. J. Flex. Manuf. Syst. 17, 261–276.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., Monostori, L., 2012. Complexity in engineering design and manufacturing. CIRP Ann. - Manuf. Technol. 61, 793–814.

Fattahi, P., Fallahi, A., 2010. Dynamic scheduling in flexible job shop systems by considering simultaneously efficiency and stability. CIRP J. Manuf. Sci. Technol. 2, 114–123.

Fattahi, P., Saidi Mehrabad, M., Jolai, F., 2007. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. J. Intell. Manuf. 18, 331–342.

Feigenbaum, J., Fortnow, L., Pennock, D.M., Sami, R., 2003. Computation in a distributed information market, in: Proceedings of the 4th ACM Conference on Electronic Commerce. ACM, pp. 156–165.

Ferber, J., 1999. Multi-agent systems: an introduction to distributed artificial intelligence. Addison-Wesley Reading, Reading.

Filip, F.G., 2008. Decision support and control for large-scale complex systems. Annu. Rev. Control 32, 61–70.

Frantzén, M., Ng, A.H.C., Moore, P., 2011. A simulation-based scheduling system for real-time optimization and decision making support. Robot. Comput.-Integr. Manuf. 27, 696–705.

Fu, M.C., Glover, F.W., April, J., 2005. Simulation optimization: a review, new developments, and applications, in: Proceedings of the 37th Conference on Winter Simulation. Winter Simulation Conference, pp. 83–95.

Fullerton, R.R., McWatters, C.S., 2001. The production performance benefits from JIT implementation. J. Oper. Manag. 19, 81–96.

Gaham, M., Bouzouia, B., Achour, N., 2014. An Evolutionary Simulation-Optimization Approach to Product-Driven Manufacturing Control, in: Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. Springer, pp. 283–294.

Gao, J., Gen, M., Sun, L., 2006. Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. J. Intell. Manuf. 17, 493–507.

Gao, J., Gen, M., Sun, L., Zhao, X., 2007. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems. Comput. Ind. Eng. 53, 149–162.

Geiger, C.D., Uzsoy, R., Aytu\ug, H., 2006. Rapid modeling and discovery of priority dispatching rules: An autonomous learning approach. J. Sched. 9, 7–34.

Ghezail, F., Pierreval, H., Hajri-Gabouj, S., 2010. Analysis of robustness in proactive scheduling: A graphical approach. Comput. Ind. Eng., Scheduling in Healthcare and Industrial Systems 58, 193–198.

Girden, E.R., 1991. ANOVA: Repeated Measures. SAGE Publications, Newbury Park.

Giret, A., Botti, V., 2004. Holons and agents. J. Intell. Manuf. 15, 645–659.

Goldberg, D., Kalyanmoy, D., 1991. Foundations of Genetic Algorithms. Morgan Kaufmann.

Gomes, M.C., Barbosa-Póvoa, A.P., Novais, A.Q., 2005. Optimal scheduling for flexible job shop operation. Int. J. Prod. Res. 43, 2323–2353. doi:10.1080/00207540412331330101

Gou, L., Hasegawa, T., Luh, P.B., Tamura, S., Oblak, J.M., 1994. Holonic planning and scheduling for a robotic assembly testbed, in: Computer Integrated Manufacturing and Automation Technology, 1994., Proceedings of the Fourth International Conference on. IEEE, pp. 142–149.

Grupe, F.H., Jooste, S., 2004. Genetic algorithms: A business perspective. Inf. Manag. Comput. Secur. 12, 288–297.

Gudehus, T., Kotzab, H., 2012. Comprehensive Logistics. Springer Berlin / Heidelberg, Berling.

Guinard, D., Trifa, V., 2009. Towards the web of things: Web mashups for embedded devices. Presented at the Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), Madrid, p. 15.

Guo, Z., Wong, W., Leung, S., Fan, J., Chan, S., 2008. A genetic-algorithm-based optimization model for scheduling flexible assembly lines. Int. J. Adv. Manuf. Technol. 36, 156–168.

Habchi, G., Berchet, C., 2003. A model for manufacturing systems simulation with a control dimension. Simul. Model. Pract. Theory 11, 21–44.

Hadeli, Valckenaers, P., Kollingbaum, M., Van Brussel, H., 2004. Multi-agent coordination and control using stigmergy. Comput. Ind. 53, 75–96.

Hadeli, Valckenaers, P., Verstraete, P., Germain, B.S., Brussel, H.V., 2006. A Study of System Nervousness in Multi-agent Manufacturing Control System, in: Brueckner, S.A., Serugendo, G.D.M., Hales, D., Zambonelli, F. (Eds.), Engineering Self-Organising Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 232–243.

Hallgren, M., Olhager, J., 2009. Flexibility configurations: Empirical analysis of volume and product mix flexibility. Omega 37, 746–756.

Hatvany, J., 1985. Intelligence and cooperation in heterarchic manufacturing systems. Robot. Comput.-Integr. Manuf. 2, 101–104.

He, D., Babayan, D.H.A., 2004. Agent–based agile manufacturing system scheduling. Presented at the Intelligent Agent-based Operations Management.

He, Y., Hui, C.W., 2007. Genetic algorithm for large-size multi-stage batch plant scheduling. Chem. Eng. Sci. 62, 1504–1523.

Heragu, S.S., Graves, R.J., Kim, B.-I., St Onge, A., 2002. Intelligent agent based framework for manufacturing systems control. Syst. Man Cybern. Part Syst. Hum. IEEE Trans. On 32, 560 – 573.

Herrero-Perez, D., Martinez-Barbera, H., 2010. Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems. IEEE Trans. Ind. Inform. 6, 166–180.

Hnaien, F., Delorme, X., Dolgui, A., 2010. Multi-objective optimization for inventory control in two-level assembly systems under uncertainty of lead times. Comput. Oper. Res. 37, 1835–1843.

Hodík, J., Becvar, P., Pechoucek, M., Vokrínek, J., Pospísil, J., 2005. ExPlanTech and ExtraPlanT: multi-agent technology for production planning, simulation and extra-enterprise collaboration. Int. J. Comput. Syst. Sci. Eng. 20, 357–367.

Hogg, T., Huberman, B.A., 1991. Controlling chaos in distributed systems. Syst. Man Cybern. IEEE Trans. On 21, 1325–1332.

Holland, J.H., 1975. Adaptation in natural and artificial systems. University of Michigan press.

Holvoet, T., Weyns, D., Valckenaers, P., 2009. Patterns of delegate mas. Presented at the Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems, 2009. SASO'09., IEEE, pp. 1–9.

Hon, K.K.B., 2005. Performance and evaluation of manufacturing systems. CIRP Ann.-Manuf. Technol. 54, 139–154.

Hong, L.J., Nelson, B.L., 2009. A brief introduction to optimization via simulation, in: Winter Simulation Conference. pp. 75–85.

Honghong, Y., Zhiming, W., 2003. The application of Adaptive Genetic Algorithms in FMS dynamic rescheduling. Int. J. Comput. Integr. Manuf. 16, 382–397. doi:10.1080/0951192031000077870

Huson, M., Nanda, D., 1995. The impact of just-in-time manufacturing on firm performance in the US. J. Oper. Manag. 12, 297–310.

Hussain, M.F., Joshi, S.B., 1998. A genetic algorithm for job shop scheduling problems with alternate routing, in: Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on. pp. 2225–2230.

Iassinovski, S., Artiba, A., Bachelet, V., Riane, F., 2003. Integration of simulation and optimization for solving complex decision making problems. Int. J. Prod. Econ. 85, 3–10.

Isern, D., Sánchez, D., Moreno, A., 2011. Organizational structures supported by agent-oriented methodologies. J. Syst. Softw. 84, 169–184.

Jeong, K.-C., Kim, Y.-D., 1998. A real-time scheduling mechanism for a flexible manufacturing system: Using simulation and dispatching rules. Int. J. Prod. Res. 36, 2609–2626.

Jernigan, S.R., Ramaswamy, S., Barber, K.S., 1997. A distributed search and simulation method for job flow scheduling. Simulation 68, 377–401.

Jia, Y., Wang, C., Wang, L., 2009. A rolling horizon procedure for dynamic pickup and delivery problem with time windows. Presented at the IEEE International Conference on Automation and Logistics, 2009. ICAL'09., pp. 2087–2091.

Johnston, K., 2009. Extending the marketing myopia concept to promote strategic agility. J. Strateg. Mark. 17, 139–148.

Jones, A.T., McLean, C.R., 1986. A proposed hierarchical control model for automated manufacturing systems. J. Manuf. Syst. 5, 15–25.

Joseph, O.A., Sridharan, R., 2011a. Evaluation of routing flexibility of a flexible manufacturing system using simulation modelling and analysis. Int. J. Adv. Manuf. Technol. 56, 273–289.

Joseph, O.A., Sridharan, R., 2011b. Effects of routing flexibility, sequencing flexibility and scheduling decision rules on the performance of a flexible manufacturing system. Int. J. Adv. Manuf. Technol. 56, 291–306.

Jozefowska, J., 2007. Just-In-Time Scheduling: Models and Algorithms for Computer and Manufacturing Systems. Springer Science + Business Media, LLC.

Jun, G., Prabhu, V.V., Jiafu, T., 2010. Simulation-based performance of assembly cells with real-time distributed arrival time control system, in: Control Conference (CCC), 2010 29th Chinese. IEEE, pp. 5291–5295.

Jung, M., Chung, M.K., Cho, H., 1996. Architectural requirements for rapid development of agile manufacturing systems. Comput. Ind. Eng. 31, 551–554.

Kacem, I., Hammadi, S., Borne, P., 2002. Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 32, 1 –13.

Kang, K., Zhang, R., Yang, Y., 2007. MAS Equipped with Ant Colony Applied into Dynamic Job Shop Scheduling, in: Huang, D.-S., Heutte, L., Loog, M. (Eds.),

Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 823–835.

Kannan, V.R., Tan, K.C., 2005. Just in time, total quality management, and supply chain management: understanding their linkages and impact on business performance. Omega 33, 153–162.

Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Robot. Res. 5, 90–98.

Kianfar, K., Moslehi, G., 2012. A branch-and-bound algorithm for single machine scheduling with quadratic earliness and tardiness penalties. Comput. Oper. Res.

Kim, J.-G., Kim, J.-S., Lee, D.-H., 2012. Fast and meta-heuristics for common due-date assignment and scheduling on parallel machines. Int. J. Prod. Res. 1–18. doi:10.1080/00207543.2011.644591

Klemmt, A., Horn, S., Weigert, G., Wolter, K.-J., 2009. Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems. Robot. Comput.-Integr. Manuf. 25, 917–925.

Koestler, A., 1967. The Ghost in the Machine. Hutchinson & Co Ltd., Oxford.

Korytkowski, P., Wiśniewski, T., Rymaszewski, S., 2013. An evolutionary simulation-based optimization approach for dispatching scheduling. Simul. Model. Pract. Theory 35, 69–85.

Kosanke, K., Vernadat, F., Zelm, M., 1999. CIMOSA: enterprise engineering and integration. Comput. Ind. 40, 83–97.

Kouiss, K., Pierreval, H., Mebarki, N., 1997. Using multi-agent architecture in FMS for dynamic scheduling. J. Intell. Manuf. 8, 41–47.

Kubiak, W., 1995. New results on the completion time variance minimization. Discrete Appl. Math. 58, 157–168.

Law, A.M., McComas, M.G., 1987. Simulation of manufacturing systems, in: Proceedings of the 19th Conference on Winter Simulation. pp. 631–643.

Law, A.M., McComas, M.G., 2002. Simulation optimization: simulation-based optimization, in: Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers. pp. 41–44.

Lee, A., Cheng, C.H., 1996. Metaplanning in FMS scheduling. Int. J. Oper. Prod. Manag. 16, 12–24.

Leitão, P., 2004. An agile and adaptive holonic architecture for manufacturing control. University of Porto, Porto.

Leitão, P., 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. Eng. Appl. Artif. Intell. 22, 979–991.

Leitão, P., 2011. A holonic disturbance management architecture for flexible manufacturing systems. Int. J. Prod. Res. 49, 1269–1284.

Leitão, P., Alves, J., Pereira, A.I., 2010. Solving Myopia in real-time decision-making using petri nets models' knowledge for service-oriented manufacturing systems, in: Proceedings of the 10th IFAC Workshop on Intelligent Manufacturing Systems. Presented at the 10th IFAC Workshop on Intelligent Manufacturing Systems, International Federation of Automatic Control, Lisbon, Portugal, pp. 144–149.

Leitão, P., Barbosa, J., Trentesaux, D., 2012a. Bio-inspired multi-agent systems for reconfigurable manufacturing systems. Eng. Appl. Artif. Intell. 25, 934–944.

Leitao, P., Colombo, A.W., Restivo, F.J., 2005. ADACOR: a collaborative production automation and control architecture. Intell. Syst. IEEE 20, 58–66.

Leitão, P., Marik, V., Vrba, P., 2012b. Past, Present, and Future of Industrial Agent Applications. IEEE Trans. Ind. Inform. PP, 1.

Leitão, P., Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. Comput. Ind. 57, 121–130.

Leitão, P., Rodrigues, N., 2012. Modelling and validating the multi-agent system behaviour for a washing machine production line, in: Industrial Electronics (ISIE), 2012 IEEE International Symposium on. IEEE, pp. 1203–1208.

Leitao, P., Valckenaers, P., Adam, E., 2009. Self-adaptation for robustness and cooperation in holonic multi-agent systems, in: Transactions on Large-Scale Data-and Knowledge-Centered Systems I. Springer, pp. 267–288.

Lemessi, M., Schulze, T., Rehbein, S., 2011. Simulation-based optimization of paint shops, in: Proceedings of the Winter Simulation Conference. pp. 2351–2362.

Li, D.-C., Yeh, C.-W., 2008. A non-parametric learning algorithm for small manufacturing data sets. Expert Syst. Appl. 34, 391–398.

Li, H., Li, Z., Li, L.X., Hu, B., 2000. A production rescheduling expert simulation system. Eur. J. Oper. Res. 124, 283–293.

Li, X., Zhang, C., Gao, L., Li, W., Shao, X., 2010. An agent-based approach for integrated process planning and scheduling. Expert Syst. Appl. 37, 1256–1264.

Lim, M.K., Zhang, Z., Goh, W.T., 2009. An iterative agent bidding mechanism for responsive manufacturing. Eng. Appl. Artif. Intell. 22, 1068–1079.

Lima, R.M., Sousa, R.M., Martins, P.J., 2006. Distributed production planning and control agent-based system. Int. J. Prod. Res. 44, 3693–3709.

Lin, G.Y.-J., Solberg, J.J., 1992. Integrated Shop Floor Control Using Autonomous Agents. IIE Trans. 24, 57–71.

Linn, R., Zhang, W., 1999. Hybrid flow shop scheduling: a survey. Comput. Ind. Eng. 37, 57–61.

López-Ortega, O., López-Morales, V., Villar-Medina, I., 2008. Intelligent and collaborative Multi-Agent System to generate and schedule production orders. J. Intell. Manuf. 19, 677–687.

Low, M.Y.H., Lye, K.W., Lendermann, P., Turner, S.J., Chim, R.T.W., Leo, S.H., 2005. An agent-based approach for managing symbiotic simulation of semiconductor assembly and test operation, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, pp. 85–92.

Luh, P.B., 1998. Scheduling of flexible manufacturing systems, in: Siciliano, P.B., Valavanis, P.K.P. (Eds.), Control Problems in Robotics and Automation, Lecture Notes in Control and Information Sciences. Springer Berlin Heidelberg, pp. 227–243.

Maglica, R., 1997. Improving the PAC shop-floor control architecture to better support implementation. Comput. Ind. 33, 317–322.

Maione, G., Naso, D., 2003. A genetic approach for adaptive multiagent control in heterarchical manufacturing systems. IEEE Trans. Syst. Man Cybern. Part Syst. Hum. 33, 573 − 588.

Maione, G., Naso, D., 2003. A soft computing approach for task contracting in multi-agent manufacturing control. Comput. Ind. 52, 199–219.

Mamei, M., Menezes, R., Tolksdorf, R., Zambonelli, F., 2006. Case studies for self-organization in computer science. J. Syst. Archit. 52, 443–460.

Mařík, V., Lažanský, J., 2007. Industrial applications of agent technologies. Control Eng. Pract. 15, 1364–1380.

Mařík, V., McFarlane, D., 2005. Industrial adoption of agent-based technologies. Intell. Syst. IEEE 20, 27 − 35.

Mataric, M.J., 1992. Minimizing complexity in controlling a mobile robot population. pp. 830 −835 vol.1.

Mati, Y., Lahlou, C., Dauzère-Pérès, S., 2011. Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints. Int. J. Prod. Res. 49, 2169–2182.

Mattfeld, D.C., Bierwirth, C., 2004. An efficient genetic algorithm for job shop scheduling with tardiness objectives. Eur. J. Oper. Res. 155, 616–630.

McFarlane, D.C., Bussmann, S., 2003. Holonic Manufacturing Control: Rationales, Developments and Open Issues, in: Agent Based Manufacturing: Advances in the Holonic Approach. pp. 303–326.

Metan, G., Sabuncuoglu, I., Pierreval, H., 2010. Real time selection of scheduling rules and knowledge extraction via dynamically controlled data mining. Int. J. Prod. Res. 48, 6909–6938.

Meyer, G.G., (Hans) Wortmann, J.C., Szirbik, N.B., 2011. Production monitoring and control with intelligent products. Int. J. Prod. Res. 49, 1303–1317.

Miller, K.D., 2002. Knowledge inventories and managerial myopia. Strateg. Manag. J. 23, 689–706.

Monden, Y., 2012. Toyota production system: an integrated approach to just-in-time. CRC Press, Boca Raton.

Monostori, L., Csáji, B.C., Kádár, B., Pfeiffer, A., Ilie-Zudor, E., Kemény, Z., Szathmári, M., 2010. Towards adaptive and digital manufacturing. Annu. Rev. Control 34, 118–128.

Monostori, L., Váncza, J., Kumara, S.R.T., 2006. Agent-Based Systems for Manufacturing. CIRP Ann. - Manuf. Technol. 55, 697–720.

Montoya-Torres, J.R., Dauzère-Pérès, S., Vermariën, L., 2005. A simulation-optimization approach for vehicle control in automated semiconductor manufacturing, in: Proceedings of the 17th European Modeling Simulation Symposium. pp. 137–142.

Montratec, 2014. Montratec conveyors http://www.montratec.com/ (visited 07/05/2014).

Morel, G., Panetto, H., Zaremba, M., Mayer, F., 2003. Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues. Annu. Rev. Control 27, 199–209.

Mouelhi-Chibani, W., Pierreval, H., 2010. Training a neural network to select dispatching rules in real time. Comput. Ind. Eng. 58, 249–256.

Nagalingam, S.V., Lin, G.C.I., 2008. CIM—still the solution for manufacturing industry. Robot. Comput.-Integr. Manuf. 24, 332–344.

Nagar, A., Haddock, J., Heragu, S., 1995. Multiple and bicriteria scheduling: A literature survey. Eur. J. Oper. Res. 81, 88–104.

Naim, M.M., Potter, A.T., Mason, R.J., Bateman, N., 2006. The role of transport flexibility in logistics provision. Int. J. Logist. Manag. 17, 297–311.

Nearchou, A., 2004. The effect of various operators on the genetic search for large scheduling problems. Int. J. Prod. Econ. 88, 191–203. doi:10.1016/S0925-5273(03)00184-1

Negahban, A., Smith, J.S., 2014. Simulation for manufacturing system design and operation: Literature review and analysis. J. Manuf. Syst.

Nejad, H.T.N., Sugimura, N., Iwamura, K., 2011. Agent-based dynamic integrated process planning and scheduling in flexible manufacturing systems. Int. J. Prod. Res. 49, 1373–1389.

Nie, L., Gao, L., Li, P., Li, X., 2013. A GEP-based reactive scheduling policies constructing approach for dynamic flexible job shop scheduling problem with job release dates. J. Intell. Manuf. 24, 763–774.

O'Grady, P.J., Menon, U., 1986. A concise review of flexible manufacturing systems and FMS literature. Comput. Ind. 7, 155–167.

O'Keefe, R.M., Rao, R., 1992. Part input into a flexible flow system: An evaluation of look-ahead simulation and a fuzzy rule base. Int. J. Flex. Manuf. Syst. 4, 113–127.

Okino, N., 1993. Biological Manufacturing Systems. Presented at the CIRP Seminar on Flexible Manufacturing Systems Past-Present-Future, Peklenik, J. (ed.), Bled, Slovenia, pp. 73–95.

Ouelhadj, D., Petrovic, S., 2009. A survey of dynamic scheduling in manufacturing systems. J. Sched. 12, 417–431.

Pach, C., 2013. ORCA: Architecture hybride pour le contrôle de la myopie dans le cadre du pilotage des Systèmes Flexibles de Production. Université de Valenciennes et du Hainaut-Cambresis.

Pach, C., Bekrar, A., Zbib, N., Sallez, Y., Trentesaux, D., 2012. An effective potential field approach to FMS holonic heterarchical control. Control Eng. Pract. 20, 1293–1309.

Pach, C., Berger, T., Bonte, T., Trentesaux, D., n.d. ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. Comput. Ind. doi:doi:10.1016/j.compind.2014.02.005

Pach, C., Zambrano Rey, G., 2011. Maitrise de la myopie des systèmes flexibles de production industriels. Presented at the 12e Colloque National AIP PRIMECA, Le Mont Dore, France.

Papadopoulos, C.T., Vidalis, M.J., O'Kelly, M.E.J., Spinellis, D., 2009. Manufacturing Systems: Types and Modeling, in: Analysis and Design of Discrete Part Production Lines, Springer Optimization and Its Applications. Springer New York, pp. 1–23.

Papakostas, N., Mourtzis, D., Michalos, G., Makris, S., Chryssolouris, G., 2012. An agent-based methodology for manufacturing decision making: a textile case study. Int. J. Comput. Integr. Manuf. 25, 509–526.

Paris, J.-L., Pierreval, H., 2001. A distributed evolutionary simulation optimization approach for the configuration of multiproduct kanban systems. Int. J. Comput. Integr. Manuf. 14, 421–430.

Parunak, H.V.D., 1991. Characterizing the manufacturing scheduling problem. J. Manuf. Syst. 10, 241–259.

Pawlewski, P., Golinska, P., Fertsch, M., Trujillo, J.A., Pasek, Z.J., 2009. Multiagent approach for supply chain integration by distributed production planning, scheduling and control system, in: International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008). pp. 29–37.

Peeters, P., Van Brussel, H., Valckenaers, P., Wyns, J., Bongaerts, L., Kollingbaum, M., Heikkilä, T., 2001. Pheromone based emergent shop floor control system for flexible flow shops. Artif. Intell. Eng. 15, 343–352.

Pereira, A., 2013. FMS performance under balancing machine workload and minimizing part movement rules. Int. J. Simul. Model. 10, 91–103.

Pereira, C.E., Carro, L., 2007. Distributed real-time embedded systems: Recent advances, future trends and their impact on manufacturing plant control. Annu. Rev. Control 31, 81–92.

Pétin, J.-F., Gouyon, D., Morel, G., 2007. Supervisory synthesis for product-driven automation and its application to a flexible assembly cell. Control Eng. Pract. 15, 595–614.

Petkova, B.N., van Wezel, W., 2006. Disentangling manufacturing flexibility, in: Preprints of the Fourteenth International Working Seminar on Production Economics. pp. 287–296.

Pezzella, F., Morganti, G., Ciaschetti, G., 2008. A genetic algorithm for the Flexible Job-shop Scheduling Problem. Comput. Oper. Res. 35, 3202–3212.

Pfeiffer, A., Kádár, B., Monostori, L., Karnok, D., 2008. Simulation as one of the core technologies for digital enterprises: assessment of hybrid rescheduling methods. Int. J. Comput. Integr. Manuf. 21, 206–214.

Pichler, F., 2000. Modeling complex systems by multi-agent holarchies, in: Computer Aided Systems Theory-EUROCAST'99. Springer, pp. 154–168.

Pierreval, H., Paris, J.-L., 2000. Distributed evolutionary algorithms for simulation optimization. IEEE Trans. Syst. Man Cybern. Part Syst. Hum. 30, 15 –24.

Pinedo, M., 2008. Scheduling: Theory, Algorithms, and Systems. Springer.

Pomerol, J.-C., 1997. Artificial intelligence and human decision making. Eur. J. Oper. Res. 99, 3–25.

Prabhu, V.V., 2003. Stable fault adaptation in distributed control of heterarchical manufacturing job shops. Robot. Autom. IEEE Trans. On 19, 142 – 149.

Prabhu, V.V., Duffie, N.. A., 1995. Modelling and Analysis of Nonlinear Dynamics in Autonomous Heterarchical Manufacturing Systems Control. CIRP Ann. - Manuf. Technol. 44, 425–428.

Prabhu, V.V., Duffie, N.A., 1996. Modelling and Analysis of Heterarchical Manufacturing Systems Using Discontinuous Differential Equations. CIRP Ann. - Manuf. Technol. 45, 445–448.

Prabhu, V., Duffie, N., 1995. Modelling and Analysis of Nonlinear Dynamics in Autonomous Heterarchical Manufacturing Systems Control. CIRP Ann. - Manuf. Technol. 44, 425 – 428. doi:10.1016/S0007-8506(07)62356-7

Pujawan, I.N., 2004. Schedule nervousness in a manufacturing system: a case study. Prod. Plan. Control 15, 515–524.

Pujo, P., Broissin, N., Ounnar, F., 2009. PROSIS: An isoarchic structure for HMS control. Eng. Appl. Artif. Intell. 22, 1034–1045.

Pyoun, Y.S., Choi, B.K., 1994. Quantifying the flexibility value in automated manufacturing systems. J. Manuf. Syst. 13, 108–118. doi:10.1016/0278-6125(94)90026-4

Rahimifard, S., 2004. Semi-heterarchical production planning structures in the support of team-based manufacturing. Int. J. Prod. Res. 42, 3369–3382.

Rajabinasab, A., Mansour, S., 2011. Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach. Int. J. Adv. Manuf. Technol. 54, 1091–1107.

Rajasekharan, M., Peters, B.A., Yang, T., 1998. A genetic algorithm for facility layout design in flexible manufacturing systems. Int. J. Prod. Res. 36, 95–110.

Reaidy, J., Massotte, P., Diep, D., 2006. Comparison of negotiation protocols in dynamic agent-based manufacturing systems. Int. J. Prod. Econ. 99, 117–130.

Reb, J., Connolly, T., 2009. Myopic regret avoidance: Feedback avoidance and learning in repeated decision making. Organ. Behav. Hum. Decis. Process. 109, 182–189. doi:10.1016/j.obhdp.2009.05.002

Rolón, M., Martínez, E., 2012. Agent-based modeling and simulation of an autonomic manufacturing execution system. Comput. Ind. 63, 53–78.

Roshanaei, V., Azab, A., ElMaraghy, H., 2013. Mathematical modelling and a meta-heuristic for flexible job shop scheduling. Int. J. Prod. Res. 1–28.

Rossi, A., Dini, G., 2000. Dynamic scheduling of FMS using a real-time genetic algorithm. Int. J. Prod. Res. 38, 1–20.

Rossi, A., Dini, G., 2007. Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robot. Comput.-Integr. Manuf. 23, 503–516.

Ryu, K., Jung, M., 2003. Agent-based fractal architecture and modelling for developing distributed manufacturing systems. Int. J. Prod. Res. 41, 4233–4255.

Sallez, Y., 2012. The Augmentation Concept: How to Make a Product "Active" during Its Life Cycle, in: Borangiu, T., Thomas, A., Trentesaux, D. (Eds.), Service Orientation in Holonic and Multi-Agent Manufacturing Control, Studies in Computational Intelligence. Springer Berlin Heidelberg, pp. 35–48.

Sallez, Y., Berger, T., Deneux, D., Trentesaux, D., 2010. The lifecycle of active and intelligent products: The augmentation concept. Int. J. Comput. Integr. Manuf. 23, 905–924.

Sallez, Y., Berger, T., Trentesaux, D., 2009. A stigmergic approach for dynamic routing of active products in FMS. Comput. Ind. 60, 204–216.

Sawik, T., 2009. Multi-objective due-date setting in a make-to-order environment. Int. J. Prod. Res. 47, 6205–6231.

Seleim, Azab, A., AlGeddawy, T., 2012. Simulation Methods for Changeable Manufacturing. Procedia CIRP 3, 179–184.

Sels, V., Gheysen, N., Vanhoucke, M., 2012. A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. Int. J. Prod. Res. 50, 4255–4270.

Senehi, M.K., Kramer, T.R., Ray, S.R., Quintero, R., Albus, J.S., 1994. Hierarchical control architectures from shop level to end effectors, in: Computer Control of Flexible Manufacturing Systems. Springer, pp. 31–62.

Sethi, A.K., Sethi, S.P., 1990. Flexibility in manufacturing: A survey. Int. J. Flex. Manuf. Syst. 2, 289–328.

Sevaux, M., Sörensen, K., 2004. A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates. 4OR Q. J. Oper. Res. 2, 129–147.

Shafaei, R., Brunn, P., 2000. Workshop scheduling using practical (inaccurate) data Part 3: A framework to integrate job releasing, routing and scheduling functions to create a robust predictive schedule. Int. J. Prod. Res. 38, 85–99.

Shaikh, N.I., Prabhu, V.V., 2009. Monitoring and prioritising alerts for exception analytics. Int. J. Prod. Res. 47, 2785–2804.

Shannon, R.E., 1998. Introduction to the art and science of simulation, in: Proceedings of the 30th Conference on Winter Simulation. IEEE Computer Society Press, pp. 7–14.

Shen, W., Hao, Q., Yoon, H.J., Norrie, D.H., 2006a. Applications of agent-based systems in intelligent manufacturing: An updated review. Adv. Eng. Inform. 20, 415–431.

Shen, W., Maturana, F., Norrie, D.H., 2000. MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing. J. Intell. Manuf. 11, 237–251.

Shen, W., Norrie, D.H., 1999. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. Knowl. Inf. Syst. Int. J. 1, 129–156.

Shen, W., Wang, L., Hao, Q., 2006b. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 36, 563 –577.

Shirazi, B., Mahdavi, I., Mahdavi-Amiri, N., 2011. iCoSim-FMS: An intelligent co-simulator for the adaptive control of complex flexible manufacturing systems. Simul. Model. Pract. Theory 19, 1668–1688.

Shnits, B., Rubinovitz *, J., Sinreich, D., 2004. Multicriteria dynamic scheduling methodology for controlling a flexible manufacturing system. Int. J. Prod. Res. 42, 3457–3472.

Siebers, P.O., Macal, C.M., Garnett, J., Buxton, D., Pidd, M., 2010. Discrete-event simulation is dead, long live agent-based simulation! J. Simul. 4, 204–210.

Smith, J.S., 2003. Survey on the use of simulation for manufacturing system design and operation. J. Manuf. Syst. 22, 157–171.

Smith, J.S., Hoberecht, W.C., Joshi, S.B., 1996. A shop-floor control architecture for computer-integrated manufacturing. IIE Trans. 28, 783–794.

Smith, R.G., 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Trans. Comput. C-29, 1104 –1113.

Sourd, F., Kedad-Sidhoum, S., 2003. The One-Machine Problem with Earliness and Tardiness Penalties. J. Sched. 6, 533–549. doi:10.1023/A:1026224610295

Sousa, P., Ramos, C., Neves, J., 2007. Scheduling in holonic manufacturing systems, in: Process Planning and Scheduling for Distributed Manufacturing. Springer, pp. 167–190.

Spano, M.R., O'Grady, P.J., Young, R.E., 1993. The design of flexible manufacturing systems. Comput. Ind. 21, 185–198. doi:10.1016/0166-3615(93)90135-N

Su, L.-H., 2009. Minimizing earliness and tardiness subject to total completion time in an identical parallel machine system. Comput. Oper. Res. 36, 461–471.

Suesut, T., Tipsuwanporn, V., Nilas, P., Remgreun, P., Numsomran, A., 2004. Multi level contract net protocol based on holonic manufacturing system implement to industrial networks, in: Robotics, Automation and Mechatronics, 2004 IEEE Conference on. pp. 253–258.

Sugimori, Y., Kusunoki, K., Cho, F., Uchikawa, S., 1977. Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system. Int. J. Prod. Res. 15, 553–564.

Sun, W., Pan, Y., Lu, X., Ma, Q., 2010. Research on flexible job-shop scheduling problem based on a modified genetic algorithm. J. Mech. Sci. Technol. 24, 2119–2125.

Tàkindt, V., Billaut, J.C., 2006. Multicriteria scheduling [electronic resource]: theory, models and algorithms. Springer.

Tavakkoli-Moghaddam, R., Jolai, F., Vaziri, F., Ahmed, P.K., Azaron, A., 2005. A hybrid method for solving stochastic job shop scheduling problems. Appl. Math. Comput. 170, 185–206.

Tay, J.C., Ho, N.B., 2008. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. Comput. Ind. Eng. 54, 453–473.

Tekin, E., Sabuncuoglu, I., 2004. Simulation optimization: A comprehensive review on theory and applications. IIE Trans. 36, 1067–1081.

Thaler, R.H., Tversky, A., Kahneman, D., Schwartz, A., 1997. The effect of myopia and loss aversion on risk taking: An experimental test. Q. J. Econ. 112, 647–661.

Thomas, A., Trentesaux, D., Valckenaers, P., 2012. Intelligent distributed production control. J. Intell. Manuf. 23, 2507–2512.

Tisue, S., Wilensky, U., 2004. NetLogo: Design and implementation of a multi-agent modeling environment, in: Proceedings of Agent. pp. 7–9.

Toth, P., 2000. Optimization engineering techniques for the exact solution of NP-hard combinatorial optimization problems. Eur. J. Oper. Res. 125, 222–238.

Trentesaux, D., 2009. Distributed control of production systems. Eng. Appl. Artif. Intell. 22, 971–978.

Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P., Barbosa, J., 2013. Benchmarking flexible job-shop scheduling and control systems. Control Eng. Pract. 21, 1204–1225.

Truong, T.H., Azadivar, F., 2003. Simulation optimization in manufacturing analysis: simulation based optimization for supply chain configuration design, in: Proceedings of the 35th Conference on Winter Simulation: Driving Innovation. Winter Simulation Conference, pp. 1268–1275.

Ueda, K., 1992. A concept for bionic manufacturing systems based on DNA-type information, in: OLLING, G.J., KIMURA, F. (Eds.), Human Aspects in Computer Integrated Manufacturing. Elsevier, Amsterdam, pp. 853–863.

Ueda, K., Hatono, I., Fujii, N., Vaario, J., 2001. Line-less production system using self-organization: A case study for BMS. CIRP Ann.-Manuf. Technol. 50, 319–322.

Ueda, K., Kito, T., Fujii, N., 2006. Modeling biological manufacturing systems with bounded-rational agents. CIRP Ann.-Manuf. Technol. 55, 469–472.

Ueda, K., Vaario, J., Ohkura, K., 1997. Modelling of Biological Manufacturing Systems for Dynamic Reconfiguration. CIRP Ann. - Manuf. Technol. 46, 343–346.

Vaario, J., Ueda, K., 1998. An emergent modelling method for dynamic scheduling. J. Intell. Manuf. 9, 129–140.

Valckenaers, P., Hadeli, Saint Germain, B., Verstraete, P., Van Brussel, H., 2006. Emergent short-term forecasting through ant colony engineering in coordination and control systems. Adv. Eng. Inform. 20, 261–278.

Valckenaers, P., Kollingbaum, M., Van Brussel, H., Bochmann, O., 2001. The design of multi-agent coordination and control systems using stigmergy, in: In Proceedings of the Third International Workshop on Emergent Synthesis (IWES01. Citeseer.

Valente, J., Moreira, M., Singh, A., Alves, R., 2011. Genetic algorithms for single machine scheduling with quadratic earliness and tardiness costs. Int. J. Adv. Manuf. Technol. 54, 251–265.

Van Brussel, H., Bongaerts, L., Wyns, J., Valckenaers, P., Van Ginderachter, T., 1999. A conceptual framework for holonic manufacturing: identification of manufacturing holons. J. Manuf. Syst. 18, 35–52.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. Comput. Ind. 37, 255–274. doi:10.1016/S0166-3615(98)00102-X

Vieira, G.E., Herrmann, J.W., Lin, E., 2003. Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. J. Sched. 6, 39–62.

Vieira, G.E., Veiga, C.P., 2009. Hierarchical and centralized architectures for distributed production planning, scheduling and control activities, in: IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, 2009. SOLI '09. Presented at the IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics, 2009. SOLI '09, pp. 341–346.

Vilcot, G., Billaut, J.-C., 2011. A tabu search algorithm for solving a multicriteria flexible job shop scheduling problem. Int. J. Prod. Res. 49, 6963–6980.

Vincent, T., 2011. Multicriteria models for just-in-time scheduling. Int. J. Prod. Res. 49, 3191–3209.

WAGO Kontakttechnik, 2005. Wago I-O System 750 Manual http://www.wago.com/wagoweb/documentation/750/eng_manu/system/m0750xxxx_x xxxxxxx_0en.pdf (visited 05/07/2014).

Walker, S.S., Brennan, R.W., Norrie, D.H., 2005. Holonic job shop scheduling using a multiagent system. Intell. Syst. IEEE 20, 50–57.

Wang, L., 2009. An agent-based simulation for workflow in emergency department, in: Systems and Information Engineering Design Symposium, 2009. SIEDS'09. pp. 19–23.

Wang, L.-C., Cheng, C.-Y., Lin, S.-K., 2013. Distributed feedback control algorithm in an auction-based manufacturing planning and control system. Int. J. Prod. Res. 51, 2667–2679.

Wang, L.-C., Lin, S.-K., 2009. A multi-agent based agile manufacturing planning and control system. Comput. Ind. Eng. 57, 620–640.

Wang, S., Xi, L., Zhou, B., 2008. FBS-enhanced agent-based dynamic scheduling in FMS. Eng. Appl. Artif. Intell. 21, 644–657.

Wang, Y., Li, H., Liu, H.-W., 2008. Multi-Agent and Hybrid Genetic Algorithm Approach for Distributed Jobshop Scheduling, in: Apperceiving Computing and Intelligence Analysis, 2008. ICACIA 2008. International Conference on. IEEE, pp. 404–407.

Wei, Y., Gu, K., Liu, H., Li, D., 2007. Contract net based scheduling approach using interactive bidding for dynamic job shop scheduling, in: Proceedings of the 2007 IEEE International Conference on Integration Technology. Presented at the International Conference on Integration Technology, IEEE, Shenzhen, China, pp. 281–286.

Weigert, G., Horn, S., Werner, S., 2006. Optimization of manufacturing processes by distributed simulation. Int. J. Prod. Res. 44, 3677–3692.

Weng, M.X., Wu, Z., Qi, G., Zheng, L., 2008. Multi-agent-based workload control for make-to-order manufacturing. Int. J. Prod. Res. 46, 2197–2213.

Weyns, D., Boucké, N., Holvoet, T., 2008. A field-based versus a protocol-based approach for adaptive task assignment. Auton. Agents Multi-Agent Syst. 17, 288–319.

Weyns, D., Holvoet, T., Helleboogh, A., 2007. Anticipatory vehicle routing using delegate multi-agent systems. pp. 87–93.

Wilensky, U., 1999. NetLogo http://ccl.northwestern.edu/netlogo/ (visited 11/05/2014). http://ccl.northwestern.edu/netlogo/.

Witkowski, T., Antczak, P., Antczak, A., 2010. Solving the Flexible Open-Job Shop Scheduling Problem with GRASP and Simulated Annealing. IEEE, pp. 437–442.

Wong, T.N., Leung, C.W., Mak, K.L., Fung, R.Y.K., 2006. Dynamic shopfloor scheduling in multi-agent manufacturing systems. Expert Syst. Appl. 31, 486–494.

Wörner, J., Wörn, H., 2006. Benchmarking of Multiagent Systems in a Production Planning and Control Environment, in: Multiagent Engineering. Springer, pp. 115–133.

Wu, Z., Weng, M.X., 2005. Multiagent scheduling method with earliness and tardiness objectives in flexible job shops. Syst. Man Cybern. Part B Cybern. IEEE Trans. On 35, 293–301.

Xia, W., Wu, Z., 2005. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Comput. Ind. Eng. 48, 409–425.

Zambrano Rey, G., Aissani, N., Bekrar, A., Trentesaux, D., 2012. A Holonic Approach to Myopic Behavior Correction for the Allocation Process in Flexible-Job Shops Using Recursiveness, in: Service Orientation in Holonic and Multi-Agent Manufacturing Control, Studies in Computational Intelligence. Springer Berlin / Heidelberg, pp. 115–128.

Zambrano Rey, G., Bekrar, A., Prabhu, V., Trentesaux, D., 2014. Coupling a genetic algorithm with the distributed arrival-time control for the JIT dynamic scheduling of flexible job-shops. Int. J. Prod. Res. 52, 3688–3709.

Zambrano Rey, G., Bonte, T., Prabhu, V., Trentesaux, D., n.d. Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. Simul. Model. Pract. Theory. doi:10.1016/j.simpat.2014.01.005

Zambrano Rey, G., Carvalho, M., Trentesaux, D., 2013a. Cooperation models between humans and artificial self-organizing systems: Motivations, issues and perspectives, in: Resilient Control Systems (ISRCS), 2013 6th International Symposium on. pp. 156–161.

Zambrano Rey, G., Pach, C., Adam, E., Berger, T., Trentesaux, D., 2011a. Myopic behavior in heterarchical control of FMS, in: Proceedings of the International Conference on Industrial Engineering and Systems Management (IESM'2011). Presented at the (IESM-2011), Metz, France, pp. 1352–1362.

Zambrano Rey, G., Pach, C., Aissani, N., Bekrar, A., Berger, T., Trentesaux, D., 2013b. The control of myopic behavior in semi-heterarchical production systems: A holonic framework. Eng. Appl. Artif. Intell. 26, 800–817.

Zambrano Rey, G., Pach, C., Aissani, N., Berger, T., Trentesaux, D., 2011b. An approach for temporal myopia reduction in Heterarchical Control Architectures. Presented at the 2011 IEEE International Symposium on Industrial Electronics (ISIE), pp. 1767–1772.

Zbib, N., Pach, C., Sallez, Y., Trentesaux, D., 2012. Heterarchical production control in manufacturing systems using the potential fields concept. J. Intell. Manuf. 23, 1649–1670.

Zeigler, B.P., Praehofer, H., Kim, T.G., 2000. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. Academic press.

Zhang, D.Z., Anosike, A., Lim, M.K., 2007. Dynamically Integrated Manufacturing Systems (DIMS): A Multiagent Approach. IEEE Trans. Syst. Man Cybern. Part Syst. Hum. 37, 824 –850.

Zhang, Q., Manier, H., Manier, M.-A., 2012. A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. Comput. Oper. Res. 39, 1713–1723.

Zimmermann, J., Mönch, L., 2007. Design and implementation of adaptive agents for complex manufacturing systems, in: Holonic and Multi-Agent Systems for Manufacturing. Springer, pp. 269–280.

Zribi, N., Kacem, I., El-Kamel, A., Borne, P., 2006. Minimizing the total tardiness in a flexible job-shop. Presented at the World Automation Congress.

Appendix A. The Manufacturing Environments

A-I. Types of Manufacturing Shop Floors

In practice, discrete manufacturing systems are classified by their shop floor layout. The layout determines the production flow through the shop floor resources, i.e., human operators or machines. Mainly, there are five different shop floor layouts: the single machine shop, project shop, flow shop, cellular system and job shop (Chryssolouris, 1992; Caramia and Dell'Olmo, 2006).

a. The single machine shop

It is the simplest layout consisting in only machine performing all the operations (Figure A-1 a.). Usually, the machine operates continuously until a new setup is required for manufacturing a different product. Therefore, the operation of a single-machine is dictated by the sequence in which products pass by the machine, and the formation of product batches for scheduling setup operations (Coffman et al., 1990).

b. The project shop

It is also known as fixed-position shop because, contrary to other shop floor layouts, the product remains fixed and resources are brought to the product as requested (Figure A-1 b.). This shop layout is reserved to large size products such as aircrafts and ships.

a. The flow shop

In this shop floor layout, machines and resources are organized according to the product sequences. Therefore, products have to visit all the machines of the flow line and each operation has to be executed on a specific machine, in a specific order (Figure A-2 a.). The flow of products is unidirectional. This organization results in dedicated flow lines, in order to avoid expensive and frequent setups. The flexible (also known as hybrid) flow shop is an extension of this layout, in which identical, uniform or unrelated machines are placed in parallel, at least at one stage of the flow line. Flow-shop-based layouts are often found in the electronic manufacturing environment such as IC packaging and PCB fabrication (Linn and Zhang, 1999).
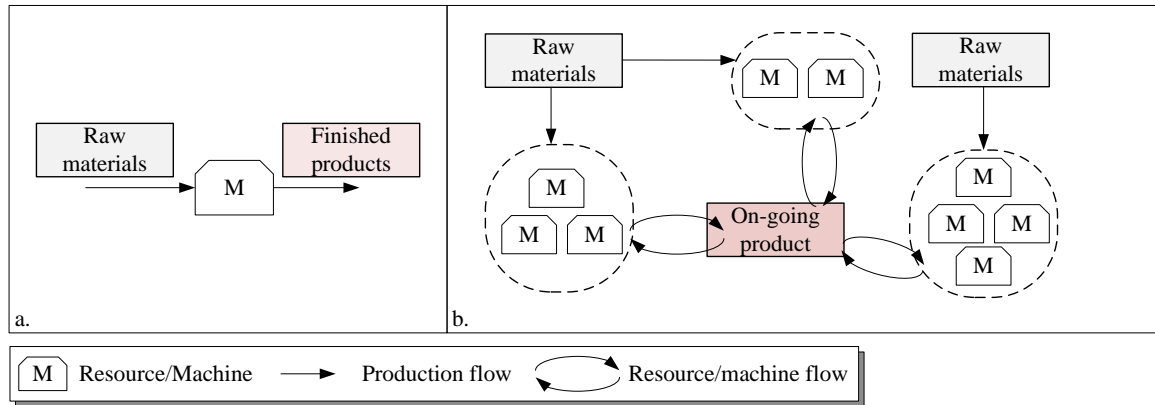
Figure A-1: a. Single machine layout and b. Project layout

b. The cellular system

As it name infers, cells of machines and resources are created according to the process combinations that occur in a particular product family. Then, a cell is configured for a specific product family. Within a cell, the material flow is determined by the product operation sequence (Figure A-2 b.). The main objective of this layout is to simplify the scheduling process by creating independent cells and reduce transportation times.
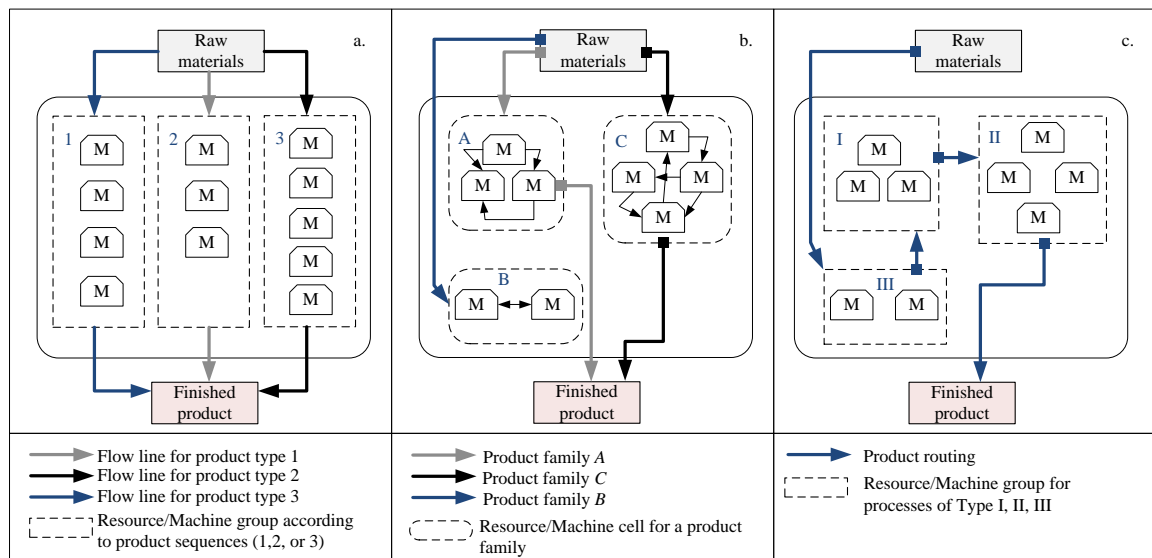
Figure A-2: a. Flow shop layout, b. Cellular system layout and c. Job shop layout

c. The job shop

In a job shop machines and resources are grouped together in the basis of their processing capabilities (e.g., lathes, milling machines, etc). Machines are usually general-purpose machines and due to their process flexibility, they can perform several manufacturing operations, thus they can handle different types of products (Figure A-2 c.). Products visit

machines depending on their operation sequence since there is *a priory* assignment of operations to machines. Product use material-handling systems to be transferred from one machine to the other. An extension of the classic job shop is the flexible job shop. In this new version, for each manufacturing operation there is a set of equivalent machines, possibly with different processing characteristics (e.g., processing time, quality of service, etc). Hence, there exist different machine sequences for the same type of product. If the manufacturing operations of a product can be swapped, then the flexible job shop becomes an open shop (Pinedo, 2008).

An FMS can present several shop floor layouts: spine, circular, ladder and open field (respectively a., b., c. and d. in Figure A-3).
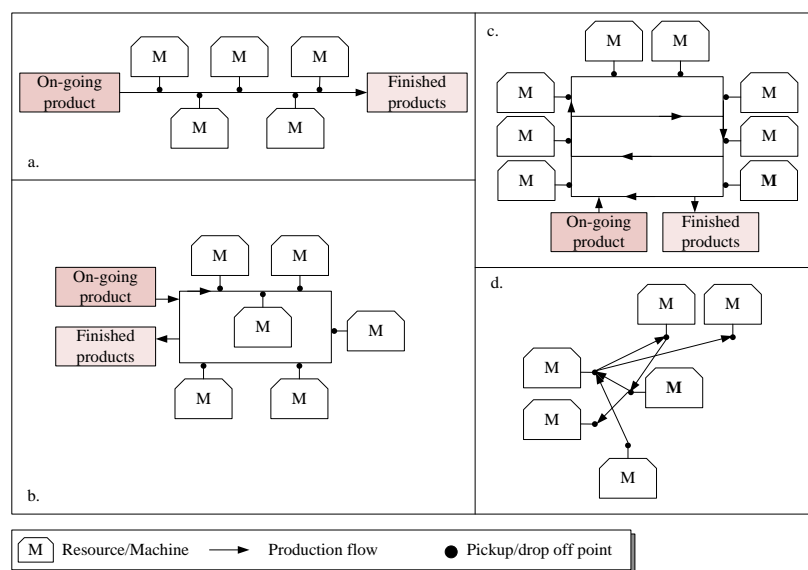
Figure A-3: FMS layouts

In the spine layout, workstations are disposed in a line and products flow through in one direction. The circular or loop layout is similar to the previous one, but the loading/unloading stations coincide. The ladder layout differs from the precedent configurations by proposing alternative product routings in order to reduce transport times. The open field layout is the most flexible one because there is no predetermined layout pattern, and the FMS is divided in specialized cells according to the group technology concept (Rajasekharan et al., 1998).

A-II. Performance criteria in manufacturing control

One of the main functions of manufacturing control is to allocate (i.e., where and when) products to manufacturing resources in order to accomplish the objectives set up at the planning level. Such allocation process generates detailed short-term schedules, which purpose is to minimize or maximize certain performance criteria. Accomplishing certain

performance level depends on the shop floor layout, its physical and operational constraints, current production and shop floor status, and production requirements (Caramia and Dell'Olmo, 2006; Chryssolouris, 1992).

In manufacturing control, performance measures are important because they allow the controller to understand the current state of the manufacturing system and take the appropriate actions necessary to reach the goals (Hon, 2005). But, before mentioning the various performance criteria used in manufacturing control, it is important to differentiate between a performance criterion and a constraint. A performance criterion allows certain flexibility or degree or freedom, while a constraint must always be respected. In some cases, a constraint can determine if there is a feasible solution or no solution at all. For instance, a constraint for a shop floor controller can be that no products can be late regarding a specific due date, leaving no margin for the controller. Contrary, a performance criterion would look for reducing the number of late products, though there can be products that finish after the due date (Tàkindt and Billaut, 2006).

Performance criteria have been divided in two types: regular and non-regular (Nagar et al., 1995). A regular criterion is a non-decreasing function of the product completion times. Examples of regular criteria are: job flow time (F), makespan ($Cmax$), tardiness (T) and their derivatives, e.g., mean flow time ($\bar{F}$), mean completion time ($\bar{C}$), mean tardiness ($\bar{T}$) and number or tardy jobs (n_T).

Contrary to regular criteria, a non-regular measure is usually not a monotone function of the product completion times (Baker and Scudder, 1990). With the advent of Just-in-time (JIT) production (more information on Just-in-time is provided in the next section of this appendix) new non-regular performance criteria have been proposed based on earliness (E) and tardiness (T), and the combination of both.

Another way to differentiate performance criteria is on the basis of the type of production strategy: make-to-stock (MTS) or make-to-order (MTO). The main difference between MTO and MTS is the time in which an order is received and treated. In MTS a product is already available in stock when the order is placed by the customer, thus the product can be dispatched immediately to the customer (Weng et al., 2008). Then, the objective of the manufacturing system under MTS production is to replenish the inventory. Since for MTS production due dates are not explicitly defined, regular performance measures are usually utilized (Jozefowska, 2007).

In MTO production some part of the production process or even the entire production is done after the client order has been accepted. Production under MTO imposes two problems: the due-date setting and the posterior due-date accomplishment (Sawik, 2009). The company

competitiveness is determined by how the company replies to the customer and how the company respects its engagements. On one hand, customers would like to have early due dates and on the other hand manufactures would prefer to extend those due dates in order to better plan and execute production and ensure on-time deliveries. However, order acceptance and due-date setting is not an easy task, not only because of the unpredictable nature of order arrivals, but also because of the additional costs in relation to inventory (if the order is finished earlier than the due date) and customer service level (if the order is finished after the quoted due date) (Weng et al., 2008). Non-regular performance measures penalizing earliness and tardiness are usually utilized for MTO production and JIT production.

A-III. Just-in-time production

Just-in-Time (JIT) manufacturing was introduced by Toyota in the 1950s to reduce inventory costs and satisfy customer demands with on time deliveries. Sugimori et al., (1977) describe JIT as a *"method whereby the production lead time is greatly shortened by maintaining the conformity to changes by having all processes produce the necessary parts at the necessary time and have on hand only the minimum stock necessary to hold the processes together"*. As Kannan and Tan, (2005) pointed out, JIT principles improve the production process by integrating customers and suppliers into the production process and reducing all sources of waste including: waste due to overproduction, longer storage times, waiting times due to machine or shop floor organizational issues, badly prepared production processes, among others (Tàkindt and Billaut, 2006).

The impact of JIT strategies on manufacturing performance has been the subject of several studies (Fullerton and McWatters, 2001; Huson and Nanda, 1995; Kannan and Tan, 2005). These studies concluded that benefits from JIT implementation extent to better product quality, shorter delivery times, better production mix and volume, more efficient utilization of workers capabilities, and better relationships with customer and suppliers.

Regarding manufacturing control, the most important objective of JIT production is to keep production rate for each type of product, per unit of time, as smooth as possible (Monden, 2012). If the customer demands are known in advance, JIT production tries to minimize the deviation concerning customer delivery times. On one hand, completing production beforehand incurs in additional inventory costs, which depends on the type of product, quantity and total time the inventory is carried. On the other hand, tardy deliveries result in customer dissatisfaction, which can carry out additional costs due to contractual penalties (Jozefowska, 2007).

From the optimization perspective, among the JIT objectives to optimize, the attention has been focused on inventory costs minimization, resulting in criteria such as earliness and tardiness costs (Baker and Scudder, 1990). The main difficulty arises with the formulation of inventory and customer dissatisfaction costs and their corresponding relationship (Jozefowska, 2007). Different performance criteria resulting from the combination of earliness and tardiness costs can be found in (Jozefowska, 2007; Baker and Scudder, 1990). An additional difficulty is that the JIT problem is considered intractable because even the minimization of the tardiness cost in a one-machine environment is proven to be NP-hard (Du and Leung, 1990). Consequently, most of the studies tackle the one-machine problem with both criteria, earliness and tardiness (Sourd and Kedad-Sidhoum, 2003) or specific cases of the multiple machine problems: job-shops with tardiness costs and other regular measures (Mattfeld and Bierwirth, 2004), identical parallel machines with earliness and tardiness costs (Su, 2009) or problems with common due-dates with earliness and tardiness penalties (Kim et al., 2012).

Appendix B. Modeling Approaches for Manufacturing Control

B-I. Bionic and Bio-inspired approaches

Bionic or biological manufacturing systems (BMSs) were proposed by Okino, (1993), Ueda, (1992) and Ueda et al., (1997). BMS adopt biologically-inspired features such as self-organization, evolution, adaptation and learning to deal with complex environments subject to uncertain events (Ueda et al., 2006). By analogy, shop floor machines can be compared with cells, in which materials enter (raw materials) and exit (processed products) through the membrane. A BMS is composed of cells, which in turn can contain lower level cells forming a hierarchical structure. Enzymes are in charge of preserving the harmony between entities and resolve potential conflicts, while genes store information (Figure B-1). In BMS, there are two types of biological information: (a) genetic information that evolves through generations, DNA-type and (b) the information learned individually during the lifetime of single organism, BN-type (Ueda et al., 1997).
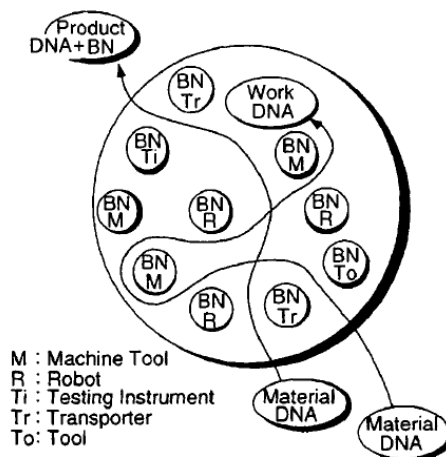
Figure B-1: Concept of BMS at the shop floor level (Kanji Ueda, Vaario, and Ohkura 1997)

Instead of focusing on the internal structure of living organisms, other bio-inspired methods focus on the interaction between living entities and the emergent behavior issued from their societies. Such emergent behavior has been denominated swarm intelligence, usually seen in ant and bee societies. Swarm intelligence has been used to model distributed systems where interactions between entities lead to an intelligent global behavior (Bonabeau et al., 1999). In such societies, entities use indirect communication means instead of direct

message exchange as it happens in human societies. For example, ants communicate using pheromones and bees through a waggle dance. Stigmergical approaches are based on pheromones laid down on the environment that are used for guidance. Pheromones pass information and thus they can be reinforced or evaporated depending on their value for reaching the objective. For an ant colony, a strong pheromone trace usually defines the shortest path form the nest to the food (Leitão et al., 2012a).

Some examples in manufacturing control using the concept of stigmergy are proposed by Peeters et al., (2001), Sallez et al., (2009), Valckenaers et al., (2001) and Hadeli et al., (2006). For instance, Holvoet et al., (2009) and Leitão et al., (2012a) combine stigmergy with multi-agent systems to introduce self-adaptation and self-optimization in vehicle routing and manufacturing systems, respectively.

Bio-inspired systems adopt emerging system methodologies inspired on self-organization, evolution and learning. Such features allow BMS to be highly adaptable and re-configurable in uncertain and dynamic environments (Ueda et al., 1997). In addition, bio-inspired methods have been widely adopted for solving hard optimization problems including dynamic and stochastic features, as well as with multiple objectives (Dorigo and Birattari, 2010).

B-II. Multi-agent systems

The multi-agent system (MAS) paradigm emerged from the distributed artificial intelligence (DAI) field, in which autonomous entities, called agents, are organized in a decentralized structure (Leitão, 2009). An "agent" according Ferber, (1999) is a virtual or real entity that possesses its own resources and skills used to pursue its own objectives. In MAS, agents interact and cooperate with each other and with the environment. An agent is highly dependent on its perception and representations of the environment and other agents (Pichler, 2000).

Three types of architectures are commonly studied in MAS: functional, blackboard, and heterarchical architectures (Baker 1998). In a functional architecture, each agent represents a functional capability and there is usually only one agent per function. In a blackboard architecture, each agent has expertise in a certain area and agents share their expertise by posting partial solutions to a problem on a central blackboard (Baker, 1998). Agents organized in heterarchical structures are characterized by their high-level of autonomy and cooperation, restraining tight master/slave relationships. Information is handled locally and agents communicate with each other when needed (Leitão, 2009).

In agent-based manufacturing control, an agent can represent a physical resource such as a machine or a product or a logical object such as production order, as shown in Figure B-2

(Shen et al., 2006a; Colombo, 2005). For instance, by using negotiation protocols, machines and product agents can cooperate with each other and make their own manufacturing control decisions to allocate product operations into available resources. Several rules or protocols can be used for agent negotiation. More information about frameworks, applications and examples can be found in (He and Babayan, 2004; Shen and Norrie, 1999; Shen et al., 2006a; Leitão, 2009; Kouiss et al., 1997; G. Maione and Naso, 2003). In turn, Mařík and Lažanský, (2007) and Leitão et al., (2012b) offer an interesting summary of industrial applications of agent-based approaches.

As pointed out by (Mařík and McFarlane, 2005), the benefits of agent technology rely on the robustness, flexibility, adaptability and re-deplorability of the control structure. However, there are still some barriers to overcome concerning the economical investment, the unpredictability due to their emergent behavior when there is no supervisory level, the number of agents required for an industrial application, the lack of commercial platforms and standards, among others.
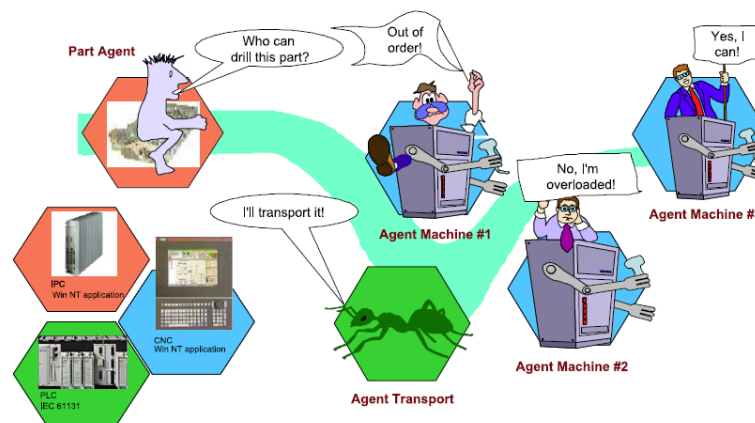
Figure B-2: Multi-agent system for manufacturing control (Colombo 2005)

B-III. Holonic manufacturing systems

The Holonic Manufacturing System (HMS) concept came up in the early 1990s when the Intelligent Manufacturing Consortium put together a program to widespread the use of distributed manufacturing systems. The HMS paradigm is based on the concepts from Arthur Koestler's studies about biological and social systems (Koestler, 1967). The main idea behind Koestler's view is that an entity is at the same time a part and a whole.

HMSs are composed of autonomous and cooperative entities called holons, that normally are assigned to manufacturing physical units, such as resources, products, material-handling systems, etc; or manufacturing operations namely, planning, scheduling, maintenance, etc. Holarchies are the societies formed by holons in order to accomplish their individual and

collective goals (Botti and Giret, 2008). The basic rules for cooperation and holon autonomy specifications are also defined by the holarchy. Holons are encouraged to cooperate with each other to develop and execute mutually acceptable plans.

The first holon structure was described by Christensen, (1994) establishing a explicit separation between the informational and physical parts of an holon. In turn, various holonic manufacturing architectures, such as PROSA by Van Brussel et al., (1998), ADACOR by Leitão and Restivo, (2006), HCBA by Chirn and McFarlane, (2000), and PROSIS by Pujo et al., (2009) have been proposed. As examples, Figure B-3 shows the set of four holons: resource, order, product and staff of PROSA and Figure B-4 shows the set of ADACOR holons and their interactions.
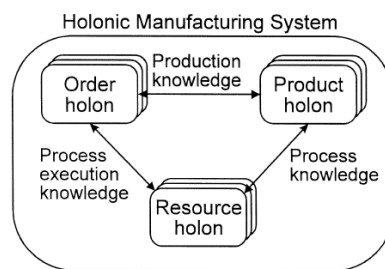
Figure B-3: Basic holons of HMS (Van Brussel et al. 1998)

Holonic Manufacturing Systems aim to provide effective decision-making processes by empowering autonomy, adaptability, agility and reactivity, among other characteristics (Sousa et al., 2007). So far, distributed agent architectures have made MAS theory appropriate for HMS implementation. Thus, HMS and MAS benefits and challenges are strongly related (Babiceanu and Chen, 2006). As concluded by Giret and Botti, (2004) the main difference between the two paradigms is the concept of recursiveness, which introduces hierarchical relationships into an heterarchical structure.

Before HMSs are widely implemented in industry, there are still several open issues to addressed (McFarlane and Bussmann, 2003): there is a need for definitive and proven design methodologies providing unambiguous guidelines. More, it is necessary to demonstrate the potentials of HMSs to improve manufacturing systems performance, through serious studies using realistic industrial context. Furthermore, the migration process between current manufacturing systems to HMS control implies further studies on more holistic approaches gathering multiple manufacturing functions that are currently treated separately, such as production planning and control.
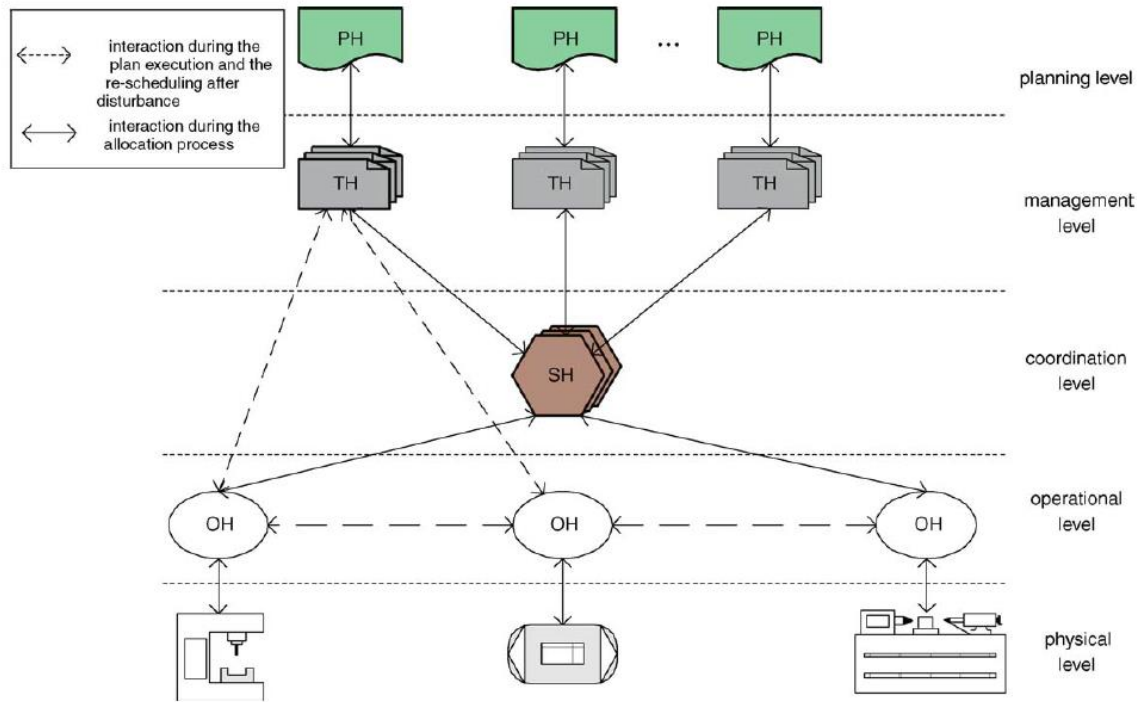
Figure B-4: ADACOR holons and their interactions (Leitão and Restivo, 2006)

Appendix C. Examples of Myopic Behavior in Heterarchical FMS Control

C-I. A soft computing approach for task contracting in multi-agent manufacturing control

In this work, Maione and Naso (2003) proposed a new multi-objective task-contracting mechanism based on fuzzy decision-making for real-time part flow control in flexible manufacturing systems. Based on multi-agent systems (MAS) design principles, information, decision and control is, as much as possible, physically and logically distributed across autonomous agents in the plant. Agents are software entities associated to physical entities operating in the manufacturing plant such as machines, raw parts, automated guided vehicles (AGV). Manufacturing control results from the concurrent actions of the multiple agents operating in the plant.

Herein, the decision-making process is based on the Contract-Net protocol (CNP) proposed by Smith, (1980). Two interacting agents are modeled: a part agent (PA) and a workstation agent (WA). The part processing priority at any workstation is solved by a negotiation process. The PA retains all the information related to its manufacturing operation sequence and the workstations that can execute each operation. The PA's main objective is to minimize the total time required to fulfill its operation sequence, i.e., part flow time, and the cost of service. In turn, the task of each WA is to assign priorities to PAs requesting an operation on the associate workstation. The processing priority depends on the buffer capacity and it is inversely proportional to the waiting time. WAs respond to PAs proposals depending on their current capacity (own capacity), setup changes with respect to the preceding and next part in queue, and estimated delivery time. The objective of WAs is to maximize the rate of parts processed per time unit, i.e., the throughput, and minimize the setup and idle times.

Herein, the analysis of social myopia is carried out only for the PA's decision algorithms (described in Figure C-1) but it can be equally done for the WA's. The decision-making process carried out by PAs goes as follows. When a part is ready for an operation, the associated PA can request such operation only to the set of m machines that can execute it, where $m \subseteq M_s$ and M_s is the set of workstations in the manufacturing system. WAs that were requested reply with a message containing h_j offers, $j=1...m$. Hence, the total number of

alternatives available for PA is $q = \sum_j h_j$. Each offer $O_{ij}=[p_i,\ c(p_i)]_j$,$i=1... \ h_j$ is a duple. The first value is the offered position in the workstation's buffer and the second value is the associated cost of the position based on the workload.
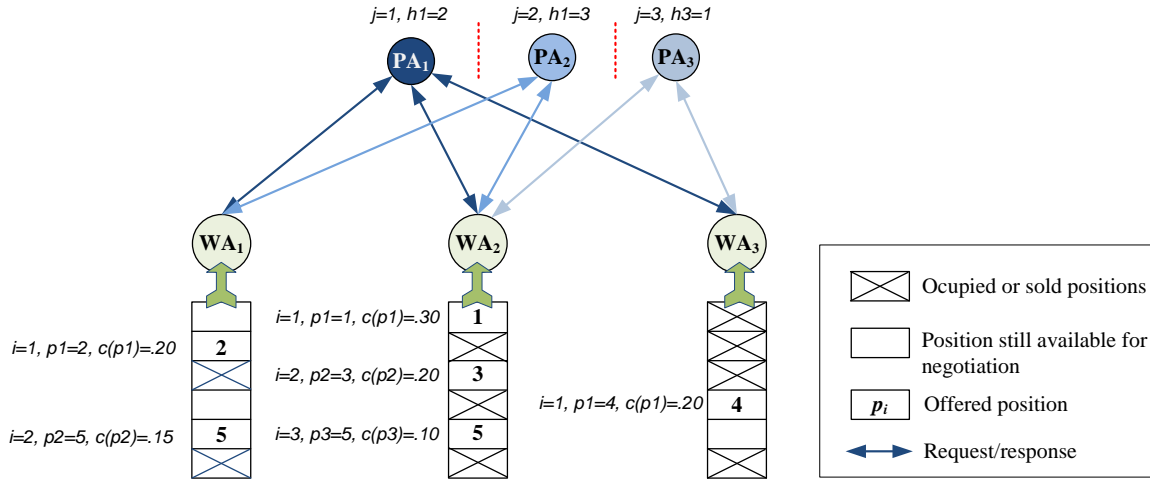
Figure C-1: Decision making based on negotiation

Now, let $m' \subseteq m$ be the subset of WA that can propose further offers, and $m'' \subseteq m$ the subset of WA that have no other places to offer ($m' \cup m''=m$ and $m' \cap m''=\varnothing$). In Figure C-1, $m'=\{WA_1,WA_3\}$ and $m''=\{WA_2\}$. Available positions are left hoping to maximize current setups, give opportunity to other parts, and/or other upcoming operations of the same part. At this point, the PA can either decide with current offers or start new negotiations, with different priorities and fictitious offers, with any of the m'. The purpose of fictitious offers is to enlarge the set of alternatives and may be, improve the cost/priority tradeoff of the offer. The difference between regular and fictitious offers is a cost reduction (i.e., a discount) managed by a tuning parameter called *negotiation factor* that is set according to the WA pricing strategy. The higher the *negotiation factor*, the more the PA tends to negotiate. A negotiation is carried out until the PAs selects an offer as final decision. To avoid indefinite negotiation loops, a *negotiation counter* (NPI) is defined. In case the NPI reaches the maximum number of allowed iterations, the PA accepts the best available offer in the last negotiation. Then, the PA informs the selected WA and waits for a confirmation. A timeout bound is set to avoid infinite waits due to lost messages or communication faults. In case of WA rejection or timeout expiration, a new negotiation cycle must be started.

On the PA side, social myopia is manifested by the PA's decision to enlarge the set of alternatives and take further negotiations. Also, PAs are socially myopic since they do not know the evolution of other PAs' negotiations (red dotted line in Figure C-1). On the WA side, tuning the negotiation factor limits also PAs perception of the negotiation process, thus it restricts the search of further alternatives. PAs are also socially myopic due to reduced

decision-makings costs obtained by avoiding interactions between PAs (reducing communication cost) and by implementing low-complex rules.

The negotiation factor, the negotiation counter and timeout bounds are time-related parameters that render PAs temporally myopic. Those parameters constraint the solution search, thus they limit the set of alternatives to analyze. In addition, the decision-making horizon is reduced to its minimum since each PA deals with one operation at a time. Furthermore, there are no look-ahead techniques to assess the outcomes of a decision so at any point in time overall performance is estimated.

C-II. Market-like multi-agent architecture

This model, proposed by Lin and Solberg, (1992), is inspired from the negotiations carried out in a marketplace where sellers aim to sell their goods at the highest price, while buyers try to buy goods of the desired types and of the required quality at the lowest prices (Cavalieri et al., 2000). Two types of highly autonomous agents represent manufacturing objects, *part* agents and *resource* agents. The part agent contains all data regarding its type, process plan and managerial information such as local goals, current state of negotiations and a budget for each manufacturing operation. The *part* agent is endowed with decision-making rules for carrying out the negotiation process and controlling its production. In turn, the *resource* agent represents any production resource. It also contains manufacturing and managerial information, as well as decision making rules for carrying out the negotiation process with *part* agents.

The negotiation process between part and resource agents aims at allocating part agents' operations at resources. This process is carried out in a way similar to an open marketplace. Part agents selfishly try to achieve their local objectives, for which they have a certain amount of currency, representing their capacity to purchased resource services. In turn, resource agents sells services at a fixed price according to their current capacity and the global state of the system, also with the objective of maximizing their own local objectives (e..g, utilization rate). Concurrent behaviors emerge when part agents try to obtain, at the same time, services offered by resources, which in fact are limited. As depicted in Figure C-2 the control strategy is carried out through the ContractNet protocol with a combination of price and objective mechanism. The bid submission is broadcasted with a purchase price of the requested service and on the basis of the unit prices offered, the resource selects which of the announced tasks is more interesting. Indeed, the price system plays a fundamental role since the bid selection between two task announcements takes place according to their unit price. Finally, the model also foresees information and monitoring agents which upon request provide explicit information about the global current state of the system.

In regards to temporal myopia, as described by Lin and Solberg, (1992) the negotiation process is executed for only one operation at a time. Then, *part* agents are not able to estimate the future consequences of task commitments and budget management. In turn, *resource* agents only count of unit prices to determine part priorities, which may also be temporally valid. Given the competitive behavior of *part* agents and the fact that there is no negotiation between *part* agents, these latter are socially myopic since their decisions only rely on information coming from *resource* agents. Though certain global information is available, it only gives a glimpse of how the entire system is being directed towards the global objective. More precise information related to each *part* agent will be then necessary to adjust local decisions and improve global performance.
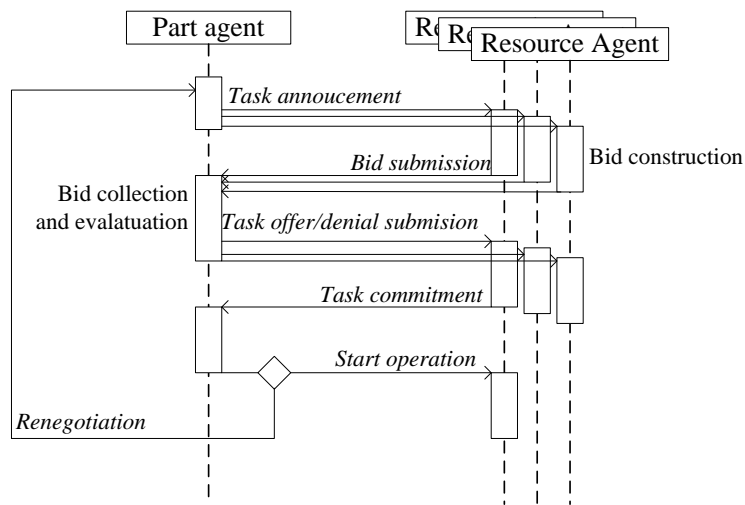
Figure C-2: Model of the market-like negotiation

Appendix D. Control Strategies in Heterarchical FMS Control

This appendix described three control strategies issued from dynamic scheduling: *fully reactive*, *predictive-reactive*, and *proactive* control strategies (Ouelhadj and Petrovic, 2009).

D-I. Fully reactive control strategy

Fully reactive strategies are supported on total distribution of control since there is no central unit capable of generating schedules beforehand and decisions are made locally on short-term basis. Usually a production order is divided into short-term decisions, scheduled stepwise by a decisional entity, revising shop floor conditions at each decision (i.e., task) (Figure D-1).
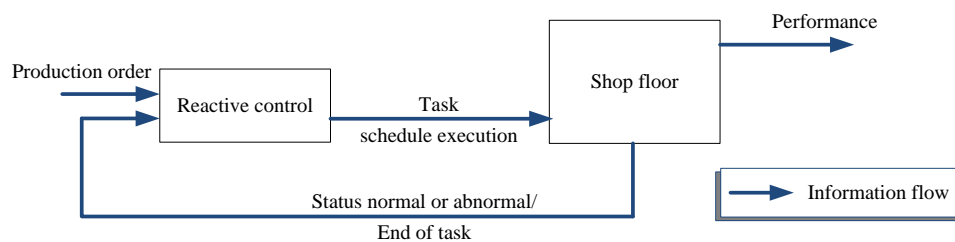
Figure D-1: Fully reactive control

If the control architecture is fully heterarchical, decisional entities mostly hold distributed artificial intelligence (DAI) algorithms or heuristics. An example of a fully decentralized and reactive FMS control approach is PROSIS, the holonic-based isoarchic structure proposed by Pujo et al., (2009). In the case of semi-heterarchical architectures, the role of the upper level is focused more on coordination rather than on control, and then low-level entities have a significant level of autonomy. In most of the cases, the upper level is called to solve operational issues between low-level decisional entities such as conflicts or deadlocks, or is engaged with performance monitoring, entity registration and identification to ensure the well functioning of the organization (Boccalatte et al., 2004; López-Ortega et al., 2008; Rajabinasab and Mansour, 2011). Then, the low level is completely in charge of short-term scheduling.

157

D-II. Predictive-reactive control strategy

Predictive-reactive strategies are probably the most common strategies reported in literature. In the predictive phase, a schedule is generated for a production order and revised in the reactive phase, in case the shop floor conditions used in the predictive phase have changed (Figure D-2). Usually, this strategy requires a semi-heterarchical structure since in order to generate predictive schedules, a complete or at least a partial model of the manufacturing system is needed. This functionality is regularly held in the upper level of the architecture. Reactivity and perturbation handling is mostly entitled to low-level decisional entities due to their low-complex decision-making algorithms, though the upper level can also participate. Optimal, near-optimal and artificial intelligence approaches usually utilize this control strategy in the upper decisional level, while priority rules and DAI approaches are embedded into low-level entities.
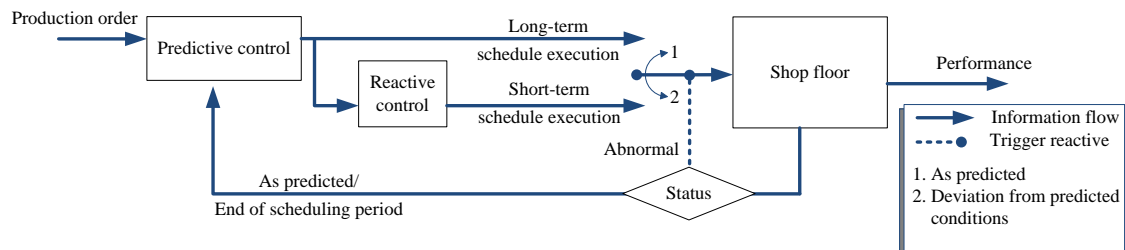
Figure D-2: Predictive-reactive control

Compared to the fully reactive strategies, the benefit of this control strategy is twofold. Firstly, the upper level is capable of estimating the system output on a longer term than entities at the lower level; and secondly, based on plans proposed by the upper level, performance can be estimated *a priori* and used as an objective for low-level entities to achieve and maintain. The studies made by Cavalieri et al., (2000) and Brennan, (2000) showed than upper-level entities with certain planning capabilities help the system guarantee a better global performance even under the presence of uncertain events. Nevertheless, this strategy also implies several drawbacks. First, the upper-level entities require a complete or at least a partial model of the system, which not only can be difficult to build, but can restraint the control system adaptability. Second, the upper-level entities require a whole view of the entire system, which is not necessarily easy to achieve, and may involve an important communication burden. Third, there could be a significant hierarchical dependence on the high level, which may result in non-negligible lags, penalizing the control system's agility. Fourth, it is necessary to clearly define the conditions that trigger reactive control, otherwise with a minimal deviation reactive control may take over without being necessary. On this switching issue, Pach et al., (n.d.) reviewed various approaches and

proposed a typology depending on the homogeneity or heterogeneity of the switching mechanism.

D-III. Proactive control strategy

Contrary to predictive-reactive approaches in which any disruption affecting the predictive schedule triggers a re-scheduling process; proactive strategies propose a less efficient schedule in theoretical conditions which can turn out to yield better results once completed (Al-Fawzan and Haouari, 2005). Consequently, the predictive schedule introduces some slack so one solution among a set of possible solutions can be accommodated to remain efficient if unexpected disturbances occur.

Most of works using this control strategy use an evaluation of the candidate schedules' robustness, so as to be able to determine the best solution to select (Aytug et al., 2005). This evaluation is generally based on measures, which are used to quantify the capacity of a schedule to remain efficient in various types of circumstances. Although such quantitative approach is convenient for optimization objectives, characterizing the robustness of a scheduling solution through a single number may be quite restrictive when many solutions have to be compared (Ghezail et al., 2010).

Appendix E. The Potential Fields Approach

This appendix briefly describes the potential fields concept for heterarchical machine allocation and product routing within a FMS. The potential fields concept was first used in the field of robot navigation to plan the movement of mobile robots in real time (Khatib, 1986; Mamei et al., 2006). In this application, attractive fields serve to guide robots to their destination while repulsive fields help them to avoid obstacles. In manufacturing systems, this concept has been used by Vaario and Ueda, (1998) for dynamic product allocation among a set of resources, Ueda et al., (2001) for line-less production system, and Weyns et al., (2008) for AGV transportation systems. Herein, the potential fields approach (PFA) proposed by Zbib et al., (2012) and Pach et al., (2012) is used since it takes multiple aspects into account in order to define attractiveness such as resource workloads, extra events and the influence of travel times for a FMS network topology.

In this PFA, the product is an active entity capable of making allocation and routing decisions. As depicted in Figure E-1at every divergent node where a product needs to make a decision, the product senses the attractive fields emitted by FMS resources (e.g., machines, inspection units, etc). The product looks for those fields related to the manufacturing operation it is interested. Products take care of their manufacturing operation sequence stepwise, so short-term schedules are generated by each product. Decision making is based on the strongest field.
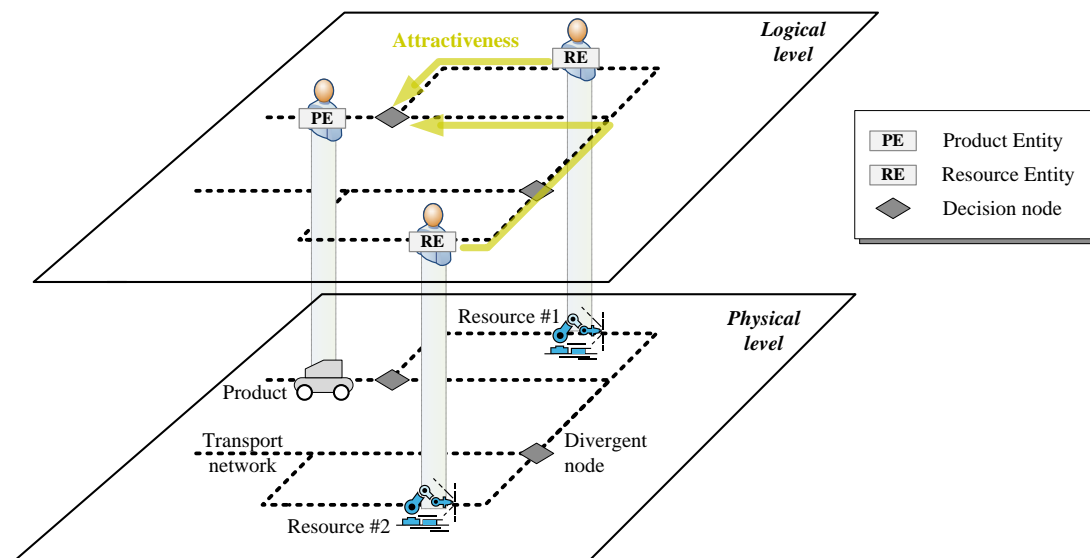
Figure E-1: Example of potential fields from (Pach et al. 2012)

In this study, resources emit 1D- fields for each service they can provide and repulsive fields are not considered. If there are multiple divergent nodes between the chosen destination, decisions are reassessed at each divergent node looking always for the strongest field. Therefore, if a resource state changes between two divergent nodes (i.e., the traveling time), the product will take into account new conditions for decision making. This feature makes this approach highly reactive and adaptable.

Each potential field depends on the attractiveness value of the emitting resource. Four types of data are used to calculate attractiveness:

✓ Types of services the resource provides

✓ The resource sate in the current time, e.g., available, unavailable

✓ The number of free places in the resource's input buffer

✓ The time at which the resource will be available again for new products.

One key aspect for decision making is the distance between the product and the possible destination resources. In order to account for that, the attractiveness value is affected by the distance. Then, at each divergent node the modified attractiveness becomes the potential field sensed by the product. Thus, products follow uphill gradients to route themselves toward their chosen resources. The attractiveness at any time t providing service Se is calculated as follows:

$$\alpha_{re}(t) = \frac{S_{re}(t) * Z_{re}}{1 + \sum_u \sum_v p_{uv} w_{uvr}(t) + p_{kl}(1 - z_{klr}(t))} \qquad \text{Eq. E-1}$$

where $S_{re}(t)$ is a binary variable set to 1 if the resource r is available for service e and Z_{re} is the maximum attractiveness value. If the resource is unavailable or it does not support service e this variable is set to 0; p_{uv} is the processing time of operation u of product v; w_{uvr} is a binary variable set to 1 if operation u of product v is waiting in the input buffer of r at time t; p_{kl} is the processing time of operation k of product l which is under manufacturing; and z_{klr} is a real variable in the interval $[0,1]$ that express the percentage of in-progress operation. This variable is set to 0 at the beginning and 1 when the operation has ended. This variable increases linearly with time.

The potential field at any time t, sensed by product on a divergent node d, for the service e, emitted from resource r is as follows:

$$F_{dre}(t) = \frac{\alpha_{re}(t)}{T_{dr}} \qquad \text{Eq. E-2}$$

where T_{dr} is the distance (expressed in time units) between the node d and the resource r. A linear decreasing envelope was used by Zbib et al., (2012). Though the field evolves over time, the envelope is the same. Figure E-2 shows the variation of attractiveness with the time, as well as the decreased attractiveness of the potential field over distance.

At every divergent node, the product senses all fields, filters the one related to the service it is looking for and chooses the resource r^* by applying the following rule:

$$r^* = \text{argmax}_r (F_{dre}(t)) \qquad \text{Eq. E-3}$$

According to some previous evaluations, the maximum attractiveness value (Z_{re}) is fixed at 200 units for all machines and all services (Pach et al., 2012).
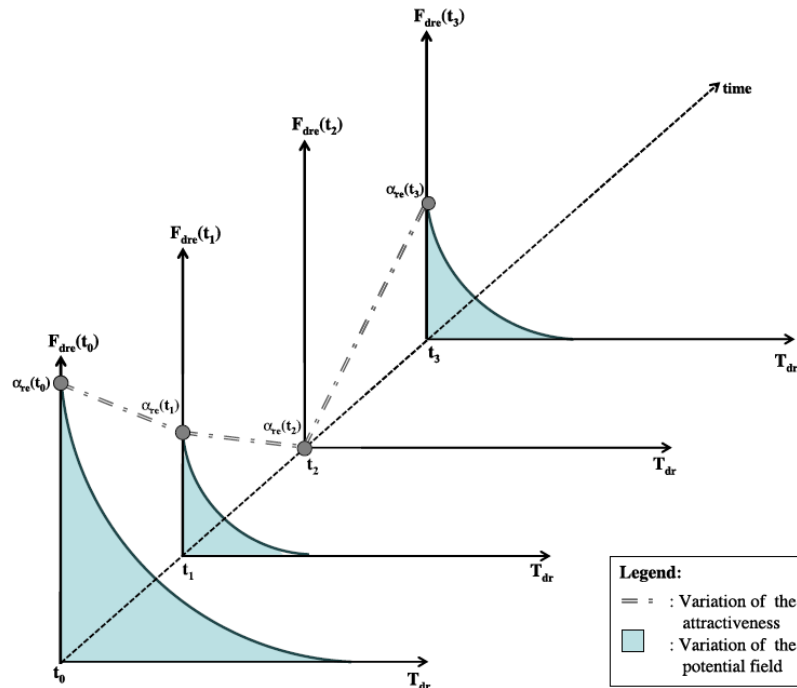
Figure E-2: Variation of attractiveness over time from (Pach et al. 2012)

Appendix F. Optimization Techniques for the Global Decisional Entity

This appendix described two optimization algorithms used within the optimization module of the global decisional entity. The arrival-control (ATC) and the genetic algorithm (GA) described in Zambrano Rey et al., (2014), were designed to minimize the due-date mean square deviation (MSD) and the completion time variance (CTV). After describing each algorithm, two connection modes between these algorithms are presented as proposed by Zambrano Rey et al., (2014).

F-I. The arrival-time control (ATC)

In discrete production processes, such as in manufacturing, the arrival time is the time in which the part[7] enters into the manufacturing system, reaches a queue of a machine or starts a specific production process. Controlling the arrival times of all parts in the system have a significant impact of system's performance because it determines how parts interact with each other. For instance, if the first-come first-serve policy is applied by manufacturing machines, the arrival times of parts influence the sequence and the order in which these are processed, the amount of machine idle times, and the accomplishment of part objectives. Based on this assumption, Prabhu and Duffie, (1995) proposed the arrival-time feedback control as a scheduling methodology for any part-driven system. Because arrival times can be modeled as continuous variables, a control loop can adjust iteratively the arrival times until the objective is closely or completely achieved.

Each part has its own control loop, resulting in the multivariable and distributed arrival-time control shown in Figure F-1. For a system with n parts, there will be n arrival time controllers (ATCs), one for each job j. Using a simulation plant, the completion times (c_j) are estimated and feedback into the loops in order to calculate the deviations (z_j) around the due dates (d_j). Each loop works simultaneously and iteratively until the arrival times (at_j) reach their steady-state. Control loops are capable of reacting and adapting to any internal or external changes. When the arrival times reach their steady-state, a schedule is ready for

---

[7] The term part is equivalent to job for the terminology used in job-shop like problems or product in FMS. This is the term used in ATC for the sake of generality.

execution, and then parts can be released according to the sequence imposed by their arrival times.
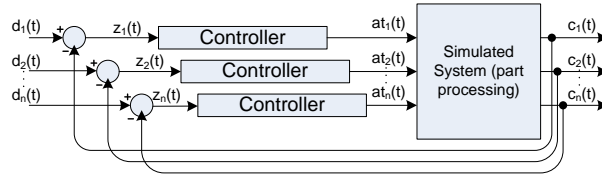
Figure F-1: Multi-variable arrival time control

The choice of the control law is critical because it impacts the system dynamics as well as its stability and convergence. The arrival time control for the jth part has the form in Eq. F-1 where k_j is the controller gain, z_j is the due-date deviation, $at_j(o)$ is an arbitrary initial condition, and τ is the integration variable, which can take values from zero to t (Prabhu, V. and Duffie, N., 1995). When capacity is sufficient for the requested parts and there is no queuing, due-date deviations converge to zero. When capacity is not sufficient, the controller adjusts the arrival times, penalizing equally earliness and tardiness. The only requirements of arrival time control are that processing times, due dates and non-modeled errors are guaranteed to be bounded.

$$at_j = k_j \int_0^t z_j(\tau)d\tau + at_j(0) \quad \text{where} \quad z_j(\tau) = d_j - c_j(\tau) \qquad \text{Eq. F-1}$$

The ATC does not make any hard assumptions about the system configuration. For shop floor layouts without routing flexibility, the ATCs solve the sequencing problem by calculating the arrival times and then the machine allocation needs to be solved with another algorithm. The only ATC configuration parameter is the controller gain k_j, which is usually the same value for all control loops. By selecting control system parameters such that $0 \leq k_j \leq 2$, the bounding of the digital arrival time control algorithm can be guaranteed (Jun et al., 2010).

F-II. A genetic algorithm for solving the flexible job-shop scheduling problem (GA)

The flexible job-shop scheduling problem (FJSSP) has been classically decomposed into two sub-problems: the product-sequencing problem, which is responsible for ordering the jobs' manufacturing operations, and the machine-sequencing problem, which is responsible

for allocating each manufacturing operation to one of the redundant machines (Xia and Wu, 2005). Based on this problem decomposition, two types of approaches have been proposed to deal with the FJSSP: integrated approaches and hierarchical approaches (Fattahi and Fallahi, 2010). Unlike integrated approaches, hierarchical approaches handle sub-problems independently and sequentially. According to this typology, Table F-1 provides a summary of a literature review, clearly describing the type of approach, the objective function(s) and the realistic characteristics necessary to model a flexible manufacturing system (FMS) based on a FJSS model, i.e., intermediate storage capacity, maximum number of jobs (limited capacity), possibility of job re-circulation and rework, and transportation time.

Table F-1: Summary of a literature review on FJSSP

Author	Type	Approach	Objective function	Inter. storage capacity	Max. number of jobs	Job re-circulation	Transport times
(Brandimarte 1993)	FJSSP	Hierarchical	*min*(makespan, weighted tardiness)	No	No	No	No
(Zribi et al. 2006)	FJSSP	Hierarchical	*min*(total tardiness)	No	No	No	No
		Integrated	*min*(total tardiness)	No	No	No	No
(Gao et al., 2007)	FJSSP	Hierarchical	*min* (makespan, max. machine workload, total workload)	No	No	No	No
(Vilcot and Billaut, 2011)	FJSSP	Hierarchical	*min*(max. lateness, total tardiness, makespan)	No	No	No	No
(Guo et al., 2008)	FAL[1]	Integrated	*min*(weighted sum of earliness/tardiness, min. production flow)	No	No	No	No
(Sun et al., 2010)	FJSSP	Integrated	*min*(makespan)	No	No	No	No
(Hussain and Joshi 1998)	FJSSP	Hierarchical	*min*(sum of squared due-date deviation)	No	No	No	No
(Prabhu 2003)	FSP[2]	N/A	*min*(mean square due-date deviation	No	No	No	No
(Valente et al., 2011) and (Kianfar and Moslehi, 2012)	SM[3]	N/A	*min*(sum of the weighted quadratic earliness and tardiness costs)	No	No	N/A	N/A
(Gomes, Barbosa-Póvoa, and Novais 2005)	FJSSP	Integrated	*min*(weighted sum of earliness/tardiness)	Yes	No	Yes	No
(Caumond et al. 2009)	FMS	N/A	*min*(makespan)	Yes	Yes	No	Yes
(Mati, Lahlou, and Dauzère-Pérès 2010)	FJSSP	Integrated	*min*(makespan)	Yes	Yes	No	Yes
(Zhang, Manier, and Manier 2011)	FJSSP	Integrated	*min*(makespan and min. of the storage capacity)	Yes	No	No	Yes
(El Khoukhi et al. 2011)	JSP[2]	N/A	*min*(sum of earliness/tardiness, empty movements)	Yes	No	No	Yes
(Zbib et al. 2012; Pach et al. 2012)	FMS	Integrated	*min*(makespan)	Yes	No	Yes	Yes

[1]Flexible Assembly Line   [2] Job-shop scheduling problem   [3]Single Machine

From an optimization perspective, the general FMS scheduling problem is a more complex version of the classical flexible job-shop scheduling problem (FJSSP), which is known to be NP-hard (Conway et al., 2003). According to Luh, (1998), the realistic characteristics already mentioned make part of the major differences between job-shop problems and FMS scheduling problems, which complicate even more the FMS scheduling problem but yield more realistic models. These characteristics need to be considered to

accomplish more applicable schedules and to avoid problems such as congestion, deadlocks or other blocking situations.

Limited intermediate storage capacity and the maximum number of jobs are limitations due to several physical characteristics (e.g., the actual size of the objects, the shop layout or the capacity of the material handling system) and/or managerial policies (e.g., minimizing the in-process inventory). Transportation times are often neglected or modeled using simple constant times in FJSS. Transportation times depend on material handling flexibility and their costs compared to those of the manufacturing operations. Material handling flexibility assumes that there are different paths to transfer jobs between machines. According to Sethi and Sethi, (1990), this type of flexibility increases machine availability and use, thus reducing throughput times. For a FMS with this type of flexibility, transportation times must be considered with more accuracy since the actual transportation time incurred by each job affects the sequence in which the jobs are processed. Job re-circulation occurs when a job visits a machine more than once, and is a phenomenon that is quite common in the real world. This condition has a direct impact on the machine-sequence flexibility.

In most of the aforementioned papers (Table F-1), the FJSSP addresses makespan minimization or a linear combination of the makespan with other objectives, such as machine workload, flow time, and maximum or total tardiness ((Brandimarte, 1993; Zribi et al., 2006; Gao et al., 2007; Vilcot and Billaut, 2011). Generally speaking, when JIT production objectives are considered, the scheduling models in the literature assume a linear relationship between earliness and tardiness costs (Guo et al., 2008; Sun et al., 2010; Gomes et al., 2005; El Khoukhi et al., 2011). Nevertheless, simply minimizing tardiness costs, or the linear combination of earliness and tardiness, is too simplistic since tardiness costs comprise other qualitative indicators related to customer loyalty (i.e., customer dissatisfaction and the risk of losing the customer). In these cases, a quadratic function represents the penalties for due date non-compliance better, which increases non-linearly (Jozefowska, 2007).

To the best of our knowledge, and despite the industrial need for JIT scheduling methods (Vincent, 2011), few studies have dealt with quadratic earliness and tardiness costs for the FJSSP. One of the few studies dealing with quadratic JIT costs was carried out by Hussain and Joshi (Hussain and Joshi, 1998). They addressed the job-shop problem with alternative routing and minimization of the sum of the mean square due date deviation. They proposed a genetic algorithm that selects one of the machines from the alternative machines, and then a pure non-linear program (NLP) determines the job order and job start time. Other studies found in the literature dealing with these types of quadratic costs deal with the flow-shop-like problem (Prabhu, 2003) or the single machine problem (Valente et al., 2011; Kianfar and Moslehi, 2012). A second issue that was identified through the literature review presented in

Table F-1 was that, in addition to JIT scheduling objectives, several realistic characteristics are usually neglected by researchers when modeling the FJSSP.

To sum up this literature review, either the FJSS models reported did not considered the MSD and the CTV as objectives functions, or hard assumptions were made to simplify the model. Therefore, a genetic algorithm (GA) in which a dual-chromosome encoding technique enables a hierarchical approach to the FJSSP was proposed. Since they were proposed by Holland (Holland, 1975), genetic algorithms (GAs) have become one of the most developed and most applied evolutionary-based meta-heuristics for combinatorial optimization problems. In the following, the GA procedure and configuration is detailed.

a. GA procedure

The pseudo code presented in Figure F-2 describes the procedure followed by the proposed genetic algorithm.

```
Step 1:    Population generation
Step 2:    Evaluate fitness of each individual
Step 3:    If apply, Select parents for crossover
                    Generate offspring
Step 4:    Select offspring for mutation
           if apply, Mutate the inviduatal
Step 5:    Evaluate offspring's fitness
           Insert offspring within the population
Step 6:    Insert random population
           if apply, Evaluate fitness of random population
                    Insert random individuals within the population
Step 7:    Select best individuals
           If population stable, then goto Step 8
           If population no stable, Goto Step 4
Step 8:    Select the best individual
           Print out solution
```

Figure F-2: Pseudo code to explain GA procedure

b. GA encoding

In this model, the product sequence sub-problem is dealt as a release sequence meaning that the GA focuses only on the product entry order. The order in which products are treated at each machine is solved locally by the machine, for instance using a priority rule such as first-input first-output. Due to the different nature of the machine sequence and release sequence problems, we propose an encoding technique that handles the two problems separately through a dual-chromosome encoding. Different advantages of this dual-chromosome encoding are listed below:

✓ It reduces the problem complexity, in some cases, this hierarchical division has proven to be more efficient than integrated approaches (Fattahi et al., 2007).

✓ Dual-chromosome encoding allows generic problem modeling, making it possible to use the same structure for other kinds of problems (e.g., the job-shop problem, flow-shop-like problems).

✓ If both chromosomes are independent, various genetic operations can be applied, allowing more diversity and easier chromosome manipulation, thus improving the efficiency of the algorithm.

✓ This division allows integrating the GA with other optimization techniques such as the ATC

Among the various encoding techniques applied to job-shop-like problems (Cheng et al., 1996), the job-based representation was chosen. Each chromosome is a permutation of the integers from 1 to *n*, and represents a solution for the release sequence sub-problem (Figure F-3 a). The main advantage of this encoding technique is that it constraints the chromosome size to its minimum length. For the machine-sequence chromosome, an indirect encoding technique (inspired from Hussain and Joshi, (1998)) was used to explore the machine-sequence alternatives available on the FMS.

<table>
<tr><td></td><td colspan="2">Chromosome encoding</td><td></td><td>Example for a 3-job client order</td></tr>
<tr><td>a.</td><td>π_1 π_2 ... ... π_{n-2} π_{n-1} π_n</td><td>where $\pi = j \in J$</td><td></td><td>2 1 3</td></tr>
<tr><td>b.</td><td>g_1 g_2 ... g_i ... g_{n-2} g_{n-1} g_n</td><td>where $g_i = u(0,1)$</td><td></td><td>0.7 0.1 0.6</td></tr>
</table>
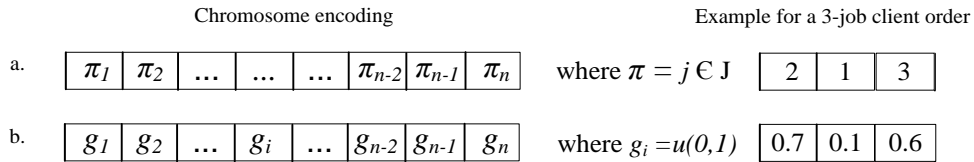
Figure F-3: Chromosome encoding

In this case, the machine sequence chromosome genes were generated by a uniform distribution and provide a value to choose one feasible machine sequence to their corresponding genes in the release sequence chromosome (Figure F-3 b). Figure F-3 also shows the dual-chromosome encoding for a 3-job order.

The sequencing chromosome is ready to be used, but the raw chromosome needs a decoding process. First, the Machine-Sequence Flexibility vector (MSF) in Eq. F-1) is formed with the number of available machine sequences for each job type. A machine sequence is a sequence of machines that a job must follow in order to fulfill its manufacturing operation sequence.

$$MSF = \{ S_1, .., S_t, .., S_T \} \qquad \text{Eq. F-1}$$

Out of all these machine sequences, one, called the selected machine-sequence (smr_j in Eq. F-2), is selected on the basis of the corresponding gene in the raw machine-sequence chromosome (g_j in Eq. F-2) and the maximum number of available machine sequences (S_{tj} in Eq. F-2) available for the job. The value of smr_j is the position of the selected machine sequence within the vector containing all the machine sequences for the job.

$$smr_j = \lceil S_{tj} * g_j \rceil \qquad\qquad \text{Eq. F-2}$$

Figure F-4 shows the decoding process for the 3-job order of Figure F-3 b. The MSF is constantly updated to be coherent with FMS conditions.
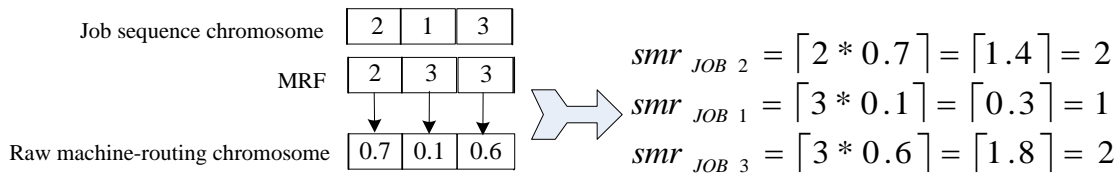
Figure F-4: Decoding process for a 3-job order

This encoding-decoding technique requires a pre-processing algorithm that constructs the selected machine sequence. When new machines are added to or withdrawn from the FMS (or machine operations modified), the MSF must be updated in order to obtain feasible schedules.

c. The initial population and fitness evaluation

The initial population of dual-chromosomes (individuals) is created randomly, and all the individuals are then classified according to their fitness. The fitness evaluation can be explained by the flow diagram in Figure F-5.

For the common due-date problem, Bagchi et al., (1987) defined two types of due-date mean square deviation (MSD) problems: the unconstrained MSD problem and the constrained MSD problem. With the unconstrained problem, increasing the due date does not result in any further increment of the MSD. Bagchi et al., (1987) proved that the schedule that minimizes the Completion Time Variance (CTV) also minimizes the MSD for any due date greater than or equal to its mean completion time ($\bar{C}$). As seen in Figure F-5, for each individual (δ), the mean completion time is calculated and depending on the common due date (dd_c), the order release time is set to zero ($rt_o=0$) or to the difference between the two values so completion times (ct_p) finish around the due date.
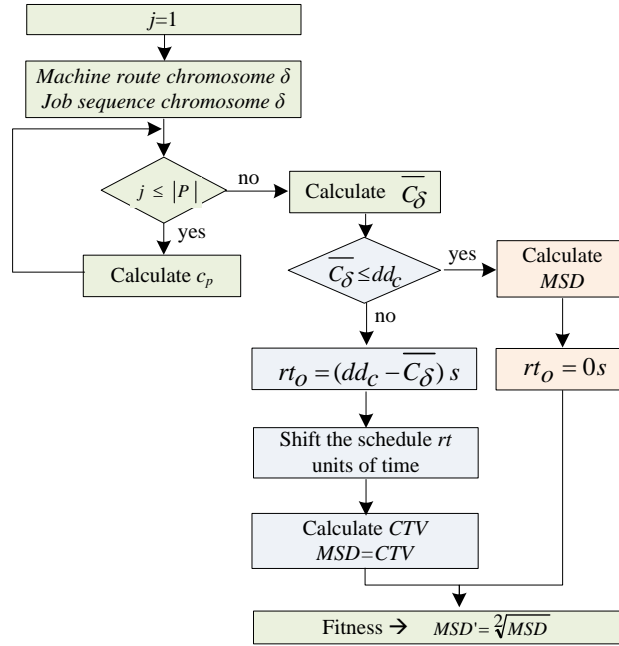
Figure F-5: Fitness evaluation

d. Crossover

To apply a crossover operator, a parent-selection mechanism needs to be chosen. The most common selection methods are the proportional (Holland, 1975), tournament and roulette wheel (Goldberg and Kalyanmoy, 1991) and linear methods (Back, 1994). Among techniques using fitness for selection the ranking technique proposed by Sevaux and Sörensen, (2004) is used herein. The first parent was chosen with the probability distribution p in (Eq. F-3) (where k is the chromosome's ranking position), which gives a higher probability (2/(Pop_size+1)) to the best individual. The second parent was selected randomly.

$$p = \frac{2(\text{Pop\_size}+1-k)}{\text{Pop\_size}(\text{Pop\_size}+1)} \qquad \text{Eq. F-3}$$

Once the parents were chosen, two offspring were created, using one of the available crossover operators. Most of the available crossover operators are improved or modified versions of the classic one-, two- or three-cut point methods or the position-based method (Nearchou, 2004). In this case, we chose the two-cut point method since the size of the chromosome was limited to its minimum. The same selection criterion and crossover operator were applied to the sequence and machine-routing chromosomes, but the process was done independently. The crossover operator was applied if a value $\theta \sim u(0,1)$ was lower than a crossover probability P_{cross}. Otherwise, each child was a fair copy of its associated parent.

e. Mutation

The most common mutation operators are adjacent exchange, swapping, shift, displacement, inversion and random insertion. Since the sequencing chromosome is a permutation chromosome, any of these operators can be used, with the exception of random insertion. Combining two-cut point crossover with shift and swapping operators gave the most efficient results for the flow-shop problem, when applied to multiple public benchmarks (Nearchou, 2004). The mutation operator was applied to the chromosome if a value $\beta \sim u(0,1)$ was lower than a mutation probability P_{muta}. Each parent generated a new child chromosome if mutation was applied. Otherwise, each child was a fair copy of its associated parent.

By using the crossover and mutation operators, *Pop_size/2* new offspring were generated and inserted into the population, expanding it to *Pop_size+Pop_size/2* individuals. However, the best *Pop_size* population was selected as the next generation, according to the natural law of survival of the fittest.

f. Random offspring insertion

In order to add diversity and avoid rapid convergence, thus falling into local minima, a number of random generated chromosomes, calculated as *Pop_size/x*, were created and inserted into the population at a certain insertion rate *gr* (Shaikh and Prabhu, 2009). The objective of this insertion rate was to perturb the population during stabilization, taking the solution from local minima if the randomly inserted individuals found a better result. The insertion rate (Eq. F-4) is inversely proportional to the number of jobs in order to favor solution space exploration as the problem size increases. The number of individuals NI to be inserted was obtained from (Eq. F-5), where *x* is the population ratio to be inserted. Although the population increased with random insertion, the selection of the best *Pop_size* individuals prevailed throughout all generations.

$$gr = \text{itermax}/_n \qquad \text{Eq. F-4}$$

$$NI = \text{Pop\_size}/_x \qquad \text{Eq. F-5}$$

g. Stop criterion

The most common criterion is the maximum number of elapsed generations. In this case, to be more robust in relation to problem size and flexibility, the algorithm stopped when there was no further solution improvement after a prescribed number of generations (*itermax*). For

each generation *iter*, the solution was considered steady if (Eq. F-6) held true. When the solution was steady for *itermax* generations, the algorithm stopped, and the chromosome pair with the best fitness was selected. The CTV can be used instead of the MSD, if apply.

$$|\text{MSD}'_{\text{iter}} - \text{MSD}'_{\text{iter}-1}|/|\text{MSD}'_{\text{iter}}| < \varepsilon] \qquad \text{Eq. F-6}$$

where ε determines the maximum range of solution variation. Table F-2 summarizes the configuration parameters of the GA used for the experimental cases.

Table F-2: GA parameterization.

Parameter	Values
Population Size	*Pop_size= n* σ *n= number of jobs in the order* *σ=10*
Crossover Operator	*Two-cut point*
Crossover Probability	*Pc=0.7*
Mutation Operator	*Shift*
Mutation Probability	*Pm=0.4*
Stopping Criterion	*No further improvement during a prescribed number of iterations has elapse (ITERMAX=10)*
Randomly Inserted Population Size	*Pop_size/4*
Frequency of Insertion	*gr= itermax /n*

F-III. Decoupled connection between ATC and GA

This first connection mode comprises two phases in which the GA and ATC worked sequentially (Figure F-6). In the first phase, the GA runs until the population stabilizes and one solution is found. The fitness evaluation of each individual (i.e., dual-chromosome) is executed using Eq. F-7 with release-times (rt_p) set to zero. The relationship between release times (release sequence), machine routes, completion times and product interaction can be explained by Eq. F-7. The first term is the product release time. The second term is the sum of all the processing times ($\tau_{m_\emptyset}^{ip}$), which depend on the machines selected (smr_p). The third term is the total waiting time experienced by the product when it is placed in the machine's input buffer ($m_\emptyset$). The last terms refers to the transportation times from one machine to the next. In this case, the transportation times depend on the queue capacity of the following machine ($CW_{m_{\emptyset+1}}$). If the following queue is at its maximum capacity, products remain in transfer until there is a place in the machine's input buffer.

$$\mathbf{c_p} = \mathbf{rt_p} + \sum_{\substack{o_{ip} \in I_p, \\ m_\emptyset \in smr_p}} \boldsymbol{\tau}_{m_\emptyset}^{ip} + \sum_{m_\emptyset \in smr_p} \mathbf{q_{pm_\emptyset}} \left(\mathbf{at_{m_\emptyset p}}\right) + \qquad \text{Eq. F-7}$$

$$\sum_{m_\emptyset \in smr_j} \mathbf{Att_{m_\emptyset m_{\emptyset+1}}} \left(\mathbf{CW_{m_{\emptyset+1}}}\right)$$

The objective of the GA is to narrow down the entire machine-allocation solution-space to find just one selected machine sequence per job. The GA uses Eq. F-7 to calculate the fitness of each individual and evolve the population towards convergence. The solution given by the GA is thus formed by a job sequence, one machine sequence per job and a temporal MSD or CTV. In the second phase, the ATC uses the sequence given by the GA as its initial condition, and the machine sequence is used by the simulation model to calculate the completion times. The number of ATCs required is the same as the number of jobs. The objective of the second phase is to calculate the release times (i.e., arrival times at the FMS) that minimize the MSD or the CTV. A solution is found when all ATCs reach their steady-state. The solution is formed by a job sequence, the release times per job, a machine sequence per job and an estimated MSD or CTV.
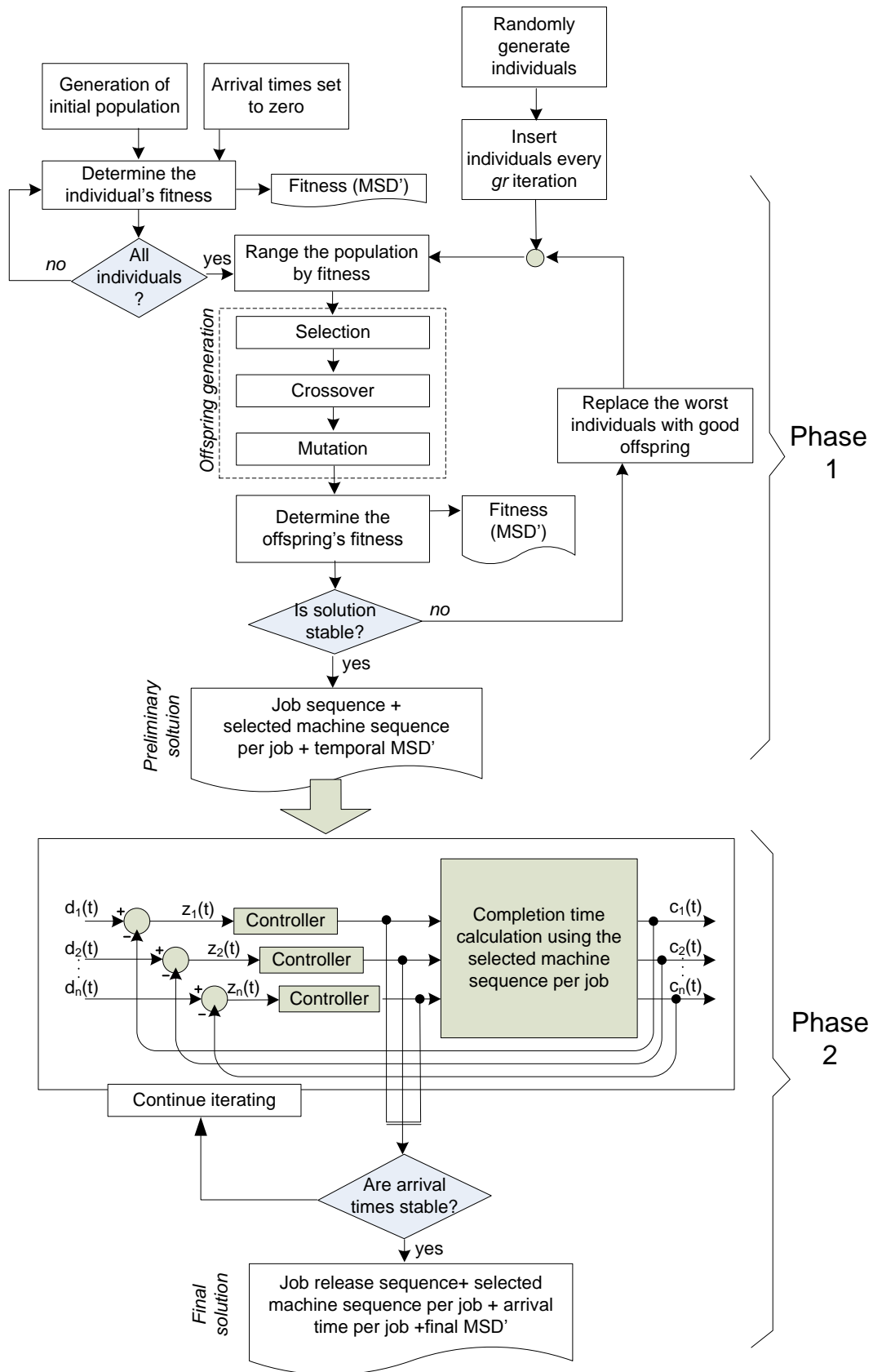
Figure F-6: GA-ATC decoupled connection for the MSD

F-IV. Coupled connection between GA and ATC

Contrary to the previous one, the purpose of this connection mode is to maintain a pool of valid solutions, and make the two algorithms work together by coupling them. In this way, the ATC efficiency is not constrained by the solution obtained by the GA. Hence, the ATC helps the GA to evolve its population including actual release time values for each product in each chromosome.

This connection mode is described in Figure F-7. The genetic algorithm continues its role as the machine-sequence explorer, but the ATC is inserted into the GA loop becoming the fitness evaluator. The algorithm works as follows. The genetic algorithm generates a *Pop_size* number of individuals and each individual becomes the input of a set of arrival-time controllers. The ATCs work on the release times until the MSD or CTV (fitness) reaches its steady-state. Then, the set of given chromosomes is ranked by their fitness, and a selected number of individuals undergo the crossover and mutation operators to create new offspring. The offspring is also evaluated by sets of arrival-time controllers and then, these individuals are inserted into the population, replacing the worst parents. Randomly-generated individuals are also introduced to add diversity and minimize the possibility to fall into local minima. However, only the fittest individuals survive throughout generations. The termination criterion is based on the stabilization of the population's MSD or CTV. The solution contains the same elements as in the sequential coupling.

Intuitively, although the coupled connection may incur in more computing cost than the decoupled connection, the intention is to fully take advantage of the capability of GA to evolve a certain population towards optimal or near optimal results. Anyway, since the number of machine-sequence combinations to explore is restricted, the computational cost is expected to be useful for FMS control.
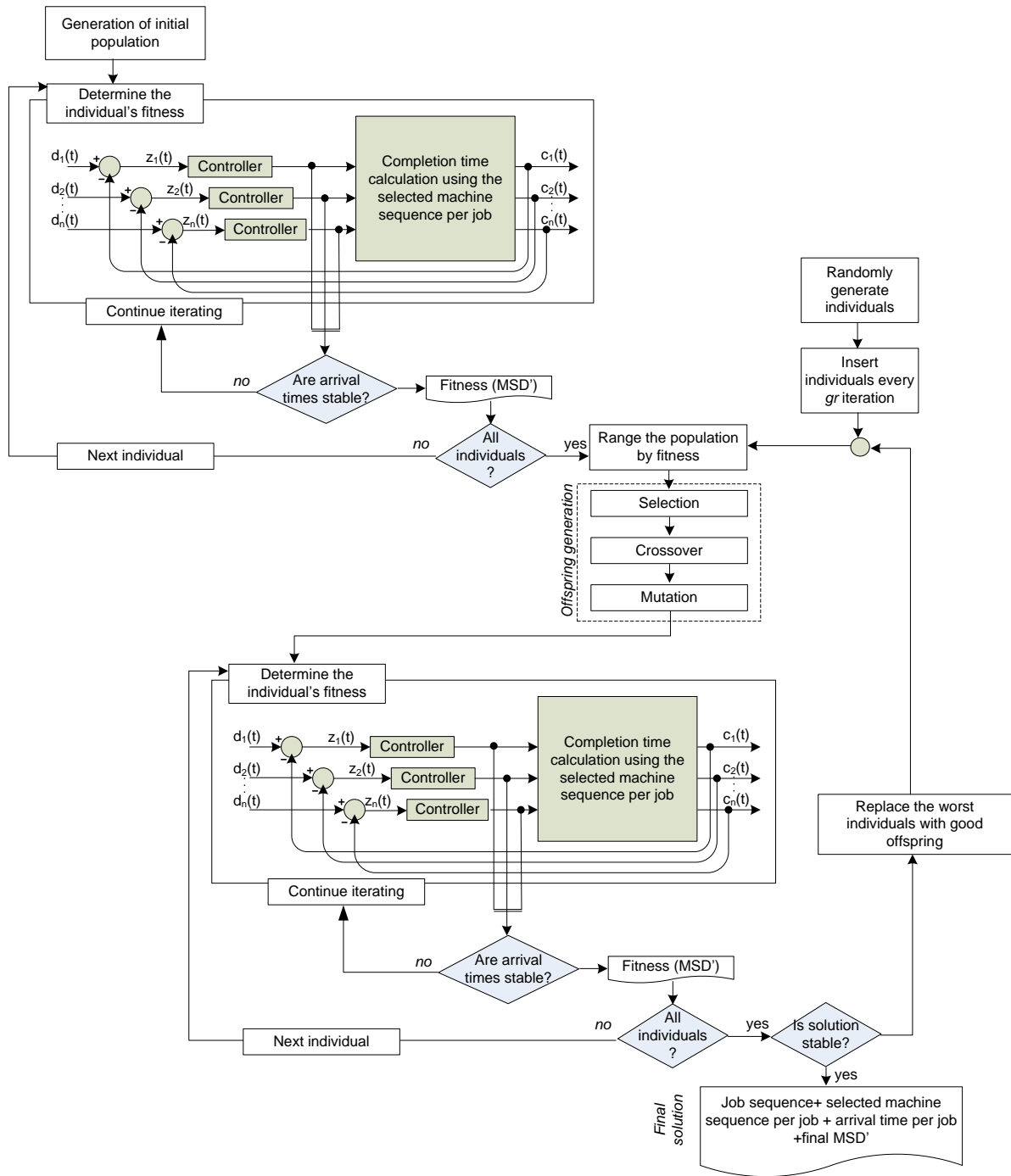
Figure F-7: GA-ATC coupled connection for MSD

Appendix G. Software Developments for Implementation

G-I. Wago® Programs

There are 18 in the AIP cell as depicted in Figure G-1. Wago® controllers are programmed using CodeSys® which is an IEC 61131-3 development system (3S-Smart Software, 2014). In order to program Wago® controllers, these latter are arranged in three groups depending on the nodes they handle. Therefore, three types of programs were developed but each Wago® controller has to be coded individually given that each controller handles different nodes, hence different sensors and actuators.
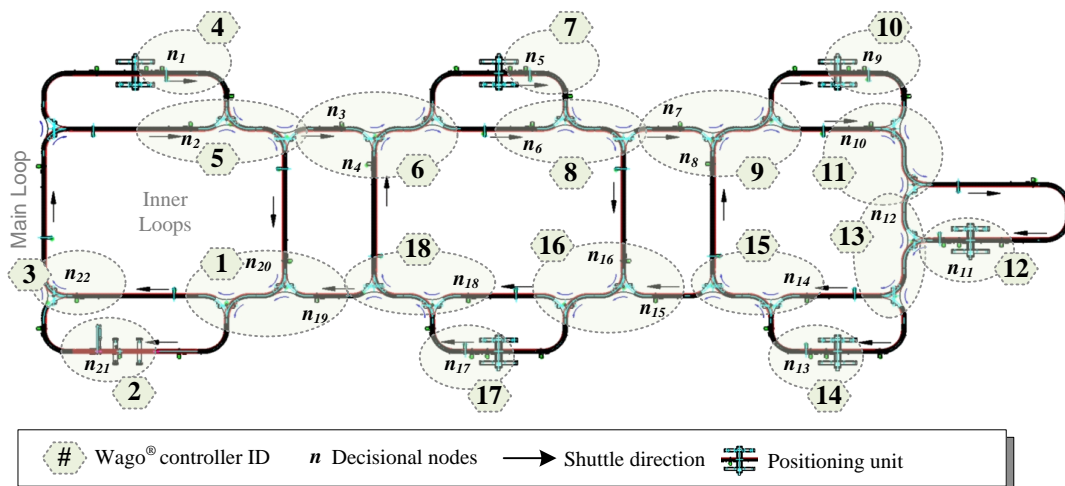
Figure G-1: Node assignation to Wago® controllers

✓ Type A controllers: these are Wago® controllers handle shuttle mobility from two regular source nodes to two destination nodes. Wago® controllers 1, 6, 9 and 16 are of this kind. For instance, Wago® controller 1 handles shuttles arriving at nodes 19 and 20, requesting transfer to nodes 21 or 22.

✓ Type B controllers: These are Wago® controllers that handle the machines at the AIP cell, hence the machine local decisional entities. These controllers handle the waiting line but they cannot control the transfer gates to let the shuttle depart from the machine. In order to let the shuttle pass they have to request type C Wago® controllers. Wago® controllers 2, 4, 7, 10, 12, 14 and 17 are of this kind.

179

✓ Type C controllers: These are Wago® controllers that handle a regular node (i.e., two source nodes and one destination node). One of the source nodes can be a machine-related node then these Wago® controllers receive requests from type B Wago® controllers. Wago® controllers 3, 5, 8, 11, 13, 15 and 18 are of this kind.

One important feature of the Wago® controller programs coded is their generality. Therefore, decisional entities can be coded in any language to control production and generate high-level control decisions (Section IV-4). Once control decisions are made, low-level control actions are executed by the Wago® controllers. However, in some cases as in the machine local decisional entity, high level and low level control is programmed in the Wago® controller.

a. Wago® controllers Type A and C

Figure G-2 depicts, in a general way, the GRAFCET program for these types of Wago® controllers.
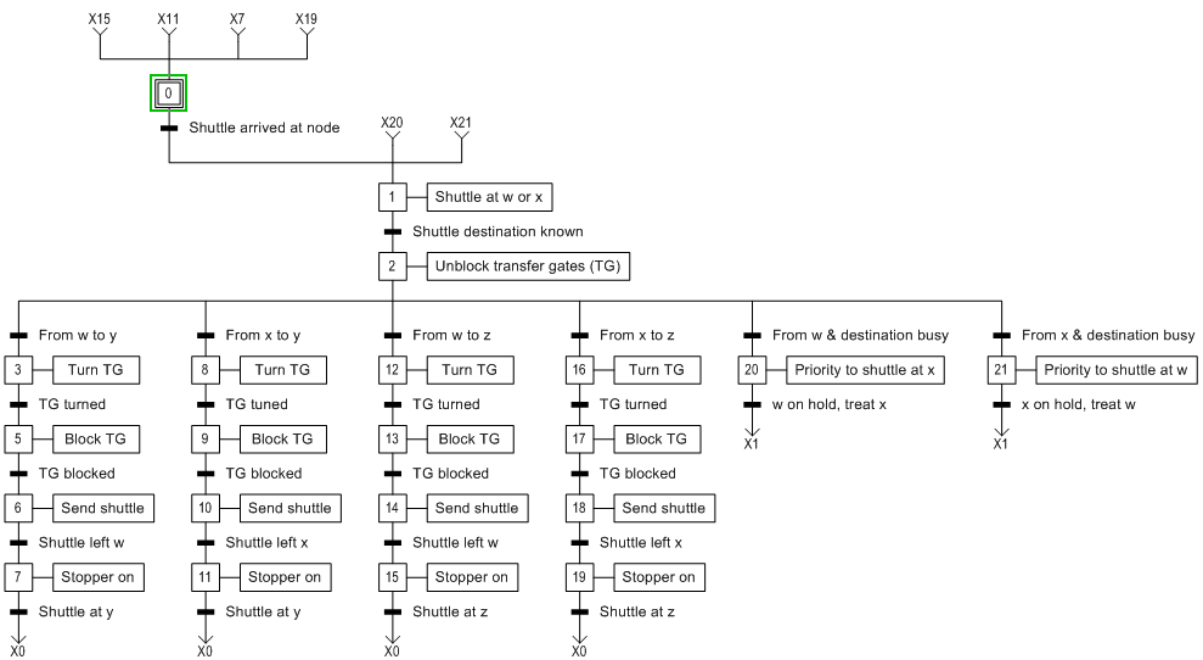
Figure G-2: GRAFCET diagram for Wago® controllers type A and C

Nodes *w* or *x* represent source nodes and nodes *y* and *z* represent destination nodes. In the case of type C Wago® controllers there is only one destination node so branches related to node *z* are not coded.

b. Wago® controllers Type B

Figure G-3 depicts the GRAFCET program for Type B controllers. The main purpose of these controllers is to provide the operation sequences requested by the product. In addition, when the product has finished all the possible operations at the machine, these controllers request type C controllers for transfer to the destination node. Communication between Wago® controllers is achieved through environment variables that are exchanged through the network. Internal procedures are used to determine when such variables change their values so updated values can be used. For instance, potential fields are exchanged through those type of variables with every Wago® controller. In that way, once a value changes all variables concerned in other Wago® controllers are updated.
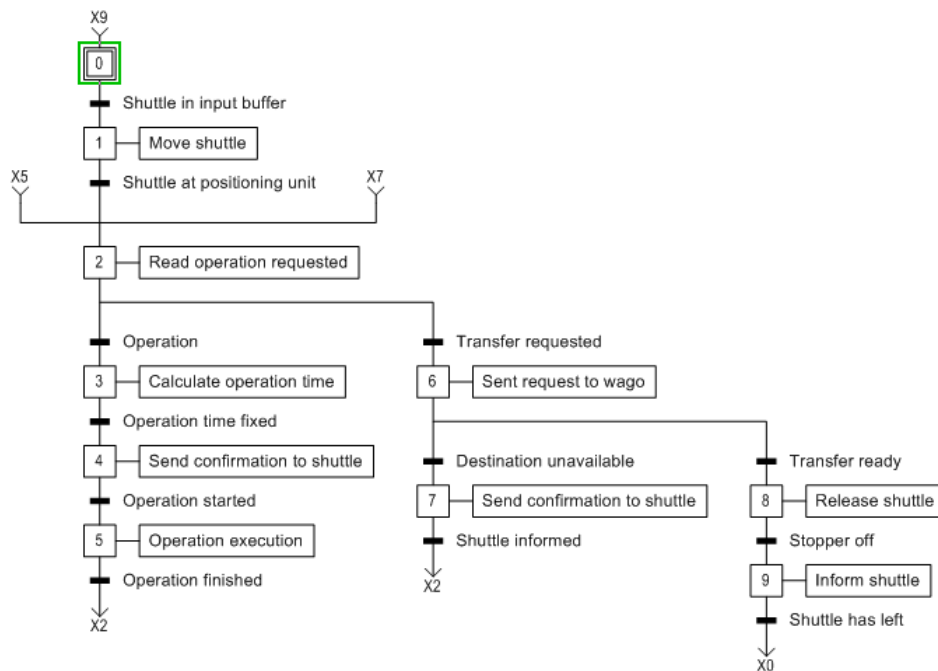
Figure G-3: GRAFCET diagram for Wago® controllers type B

G-II. Java Programs

The global decisional entity and the transport local decisional entity were implemented in Java programming language. For the global decisional entity two java programs were coded, one for the control module and another for the interaction module, this latter called dispatcher program. Figure G-4 shows a flow diagram representing the main parts of the of the control module of the global decisional entity. The optimization mechanisms were program as methods so it is possible to use one or the other or both of them, coupled or decoupled. A TCP/IP communication method allows information exchange with the dispatcher program.
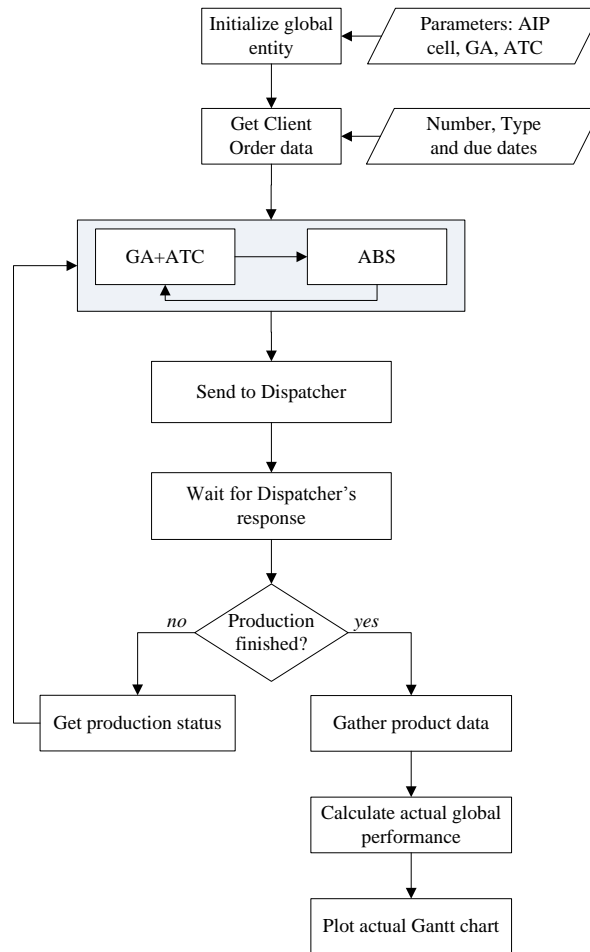
Figure G-4: Flow diagram of the control module of the global decisional entity

Figure G-5 shows a flow diagram for the interaction module of the global decisional entity. This dispatcher program is in charge of monitoring production, establishing communication with transport local decisional entities and update AIP cell data in case of changes. It also collects production information either for recalculation or final global performance calculation when production has finished.
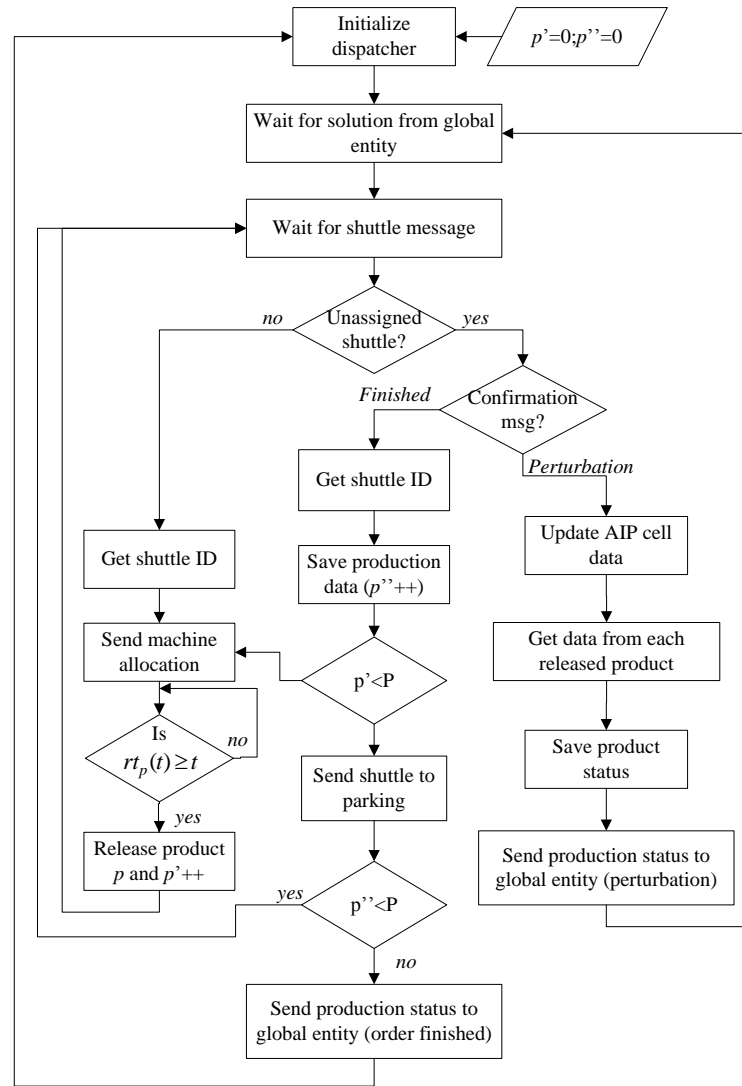
Figure G-5: Flow diagram of the dispatcher program (interaction module of $pGDE_1$)

Figure G-6 shows the flow diagram of the control and interaction modules of the transport local decisional entity. One particular characteristic of the interaction module is that it communicates with the dispatcher program via TCP/IP protocol and Wago® controllers (including the machine local decisional entities) via Modbus TCP/IP. Two threads were coded: one for handling the normal operation sequence and another thread for paying attention if another transport local decisional entity announces a perturbation. Perturbations are announced by broadcasting a message with a perturbation code. When a perturbation has been announced each entity finishes their current task, i.e., arrive at the next node, finishes an operation, and reports its current status and production history to the dispatcher (i.e., operations already executed). Afterwards the entity waits for new information coming from the global level.

Flowchart nodes:

Initialize Product Program ← Node list, Wago IP list

Localize at loading station

Get the shuttle's ID

Send shuttle's ID to dispatcher

Wait for dispatcher's response

Product assgined? — *yes* / *no*

no → Go to parking → End

Thread 1

p_type, p_Id, smr_p, rt_p(t) and I_p → Get product information

Get product information → $o_{ip} < |I_p|$ — *Finished*

Look for shortest path to the jth m in smr_p

Is m's input buffer full? — *yes* → Chose an inner loop to loop around

no → Get into m's input buffer

At m's positioning unit? — *no* / *yes*

Sequentially requests operations allocated at this m

Operations Finished at m? — *no* / *yes*

m unavailable? — *no* / *yes* → Send broadcast message to products and dispatcher

Send production information to dispatcher

Thread 2

Start thread to receive perturbation announcements

Perturbation announced? — *no* / *yes*

Finish current operation

Wait for dispatcher's response

G-III. The NetLogo simulation model

This NetLogo model was developed by Thérèse Bonte, research engineer at TEMPO-PSI. The AIP cell simulation model realized in Netlogo has four types of agents: *"turtles"*, mobile or static, which are decisional entities; *"patches"*, static, which provide a grid representation of the environment; *"links"*, which are agents that connect two turtles; and "*the observer*"

who is in charge of giving instructions to the other agents. In this study, the FMS components were modeled with just turtles and links, as shown in the UML (Unified Modeling Language) class diagram in Figure G-7. The FMS topology is represented by a directed graph composed of *Nodes* linked by *NodeLinks*. *Nodes* can be structural, which define the FMS layout; decisional, to model products that collect information and make decisions; or directional, to define transport directions that guide products in the FMS (e.g., convergent or divergent nodes).
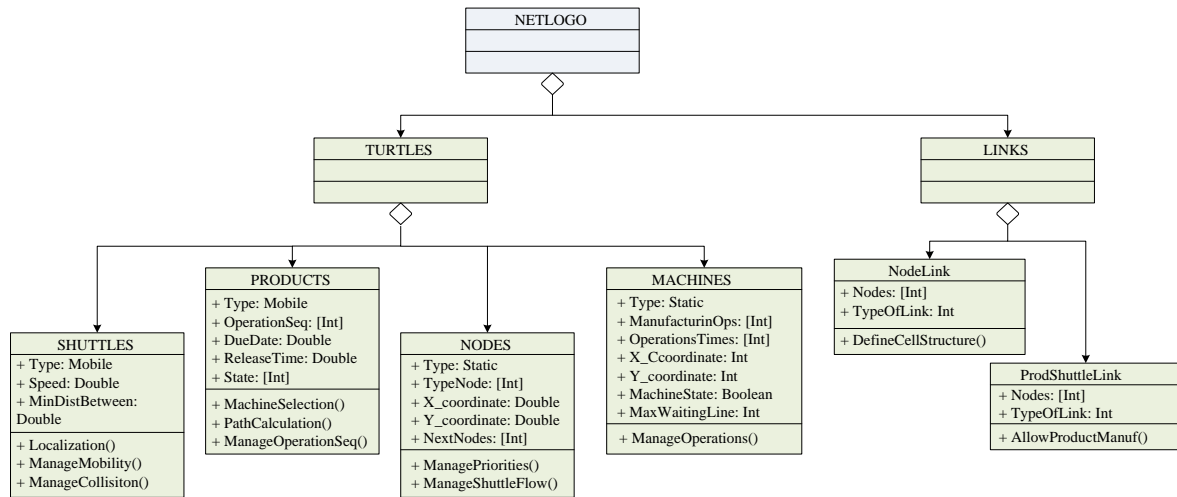
Figure G-7: The static structure of the simulation model in NetLogo

Transport decisional entities (*vLDE*s) are composed of two turtles: *transport* and *product* to differentiate between control decisions related to product routing, taken care by the *transport turtle*, and machine allocation that is taken care by the *product turtle*. *Transport turtles* are in charge of taking *products turtles* around the FMS. For instance, for a particular problem with *P* products and *MJ transport turtles*, once a *product p* is released into the FMS, if there is an empty *transport turtle*, a link (*ProdShuttleLink*) is created between the *product* and the *transport turtles*. While the *transport turtle* manages the mobility, traffic and possible collisions through the *NodeLinks*, the *product turtle* makes machine allocation decisions to accomplish its operation sequence. At the end of the sequence the link is broken; the *product turtles* change their state to finished and calculate their completion times (i.e., simulation outputs); and *transport turtles* are liberated to assist new *product turtles* or go to the transport stock area.

A simulation replication starts by an initialization process consisting in loading the network topology (node and machine attributes), and the product and transport device attributes. These data along with the parameters, assumptions and constraints defined in Sections IV-2.2 to IV-2.4 are considered as static parameters that do not change during a

simulation replication. Then *"ProdShuttleLink"* are created depending on the number of available transport devices and the number of products. These links are created every time a new product is released into the FMS. Every time the simulation clock is incremented, *transport turtles* calculate their next position and after dealing with priorities, only those that are able to move will do so. For instance, a *transport turtle* in a machine's input buffer only enters the machine when the machine lets it do it; otherwise the *transport turtle* remains in its place. Depending on the new *transport turtle* position, e.g., a decisional node or a machine, the *transport turtle* executes the appropriate process, e.g., makes a decision, or starts an operation. In the *LDE*'s behavior model *transport*, *product* and *machine turtles* are customized with the decision-making algorithms chosen for $vLDE$s and $mLDE$s, respectively, as described in Sections IV-4.1 and IV-4.2.

The Netlogo simulation model can be controlled in two ways: manually and from a Java program. Figure G-8 shows the simulation model interface when the simulation model is controlled manually. The first step to manually launch a simulation replication is to choose the type of local decision-making algorithm, i.e., the potential fields approach or the first available machine rule. (① in Figure G-8). Based on the benchmark study proposed by (Trentesaux et al., 2013) the simulation scenario must be chosen (② in Figure G-8). Then, the source files for the client order and product configuration must be specified (③ in Figure G-8). The model can then be launched (④in Figure G-8) and simulation can start either stepwise or continuously until the client order finishes (⑤ in Figure G-8). In order to see product interactions or get results quickly, the simulation clock can be modified (⑥in Figure G-8). Intermediate results in terms of number and type of finished products are displayed (⑦in Figure G-8) and the actual cell dynamics can be seen through the simulation screen (⑨in Figure G-8). Figure G-9 shows the simulation screen for a simulation replication under normal conditions and Figure G-10 shows also a 3D view when machine m_2 breaks down (in Figure G-10).
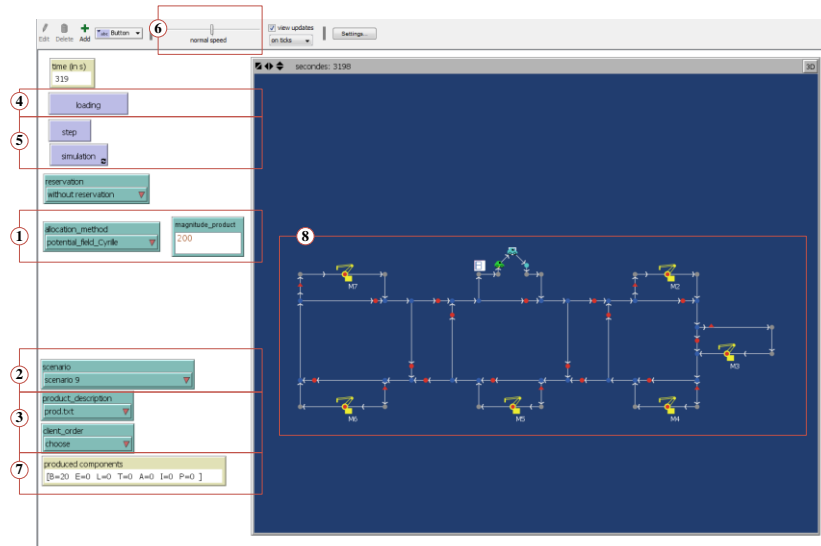
Figure G-8: View of the NetLogo simulator in manual mode
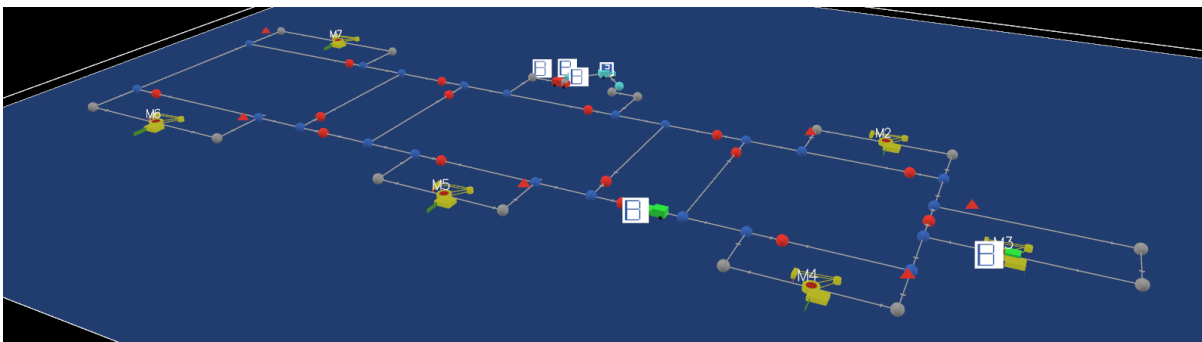
Figure G-9: Simulation screen in 3D view for normal conditions

When the Netlogo simulation model is controlled by a Java program, the model is launched without visualization to avoid higher simulation costs. The model accepts the client order (types of products, release times and machine allocations if apply) as entry data, and returns completions times for each product when the simulation replication has finished. The Java program launches the simulation replication and waits for simulation results.
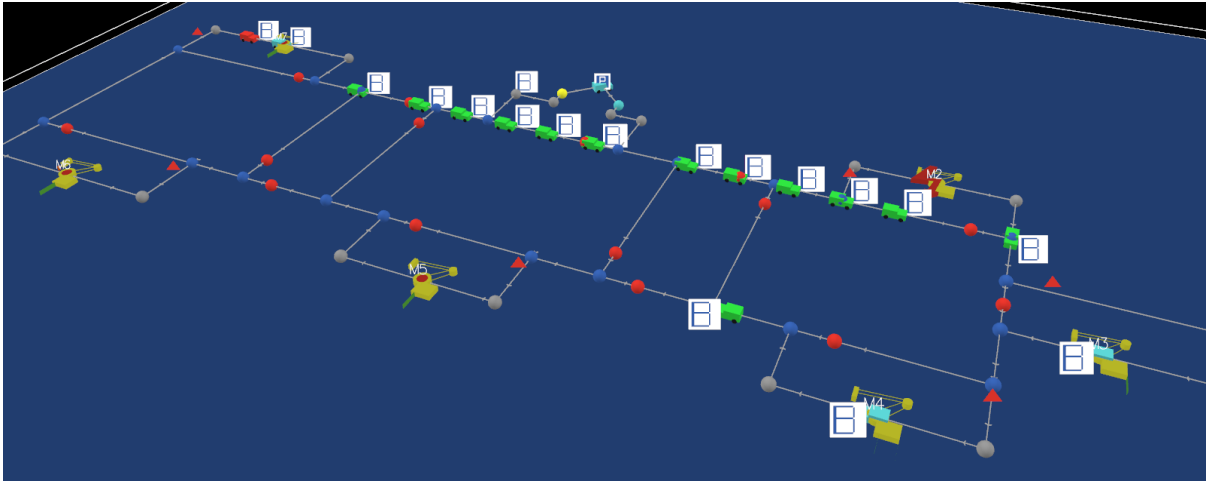
Figure G-10: Simulation screen in 3D view for the breakdown case

Appendix H. Results from Simulation Studies

This appendix first describes the experimental protocol followed to evaluate the efficiency of the proposed genetic algorithm and the GA-ATC connection types, decoupled and coupled. The ATC efficiency to deal with the flexible job-shop scheduling problem (FJSSP) was also evaluated by evaluating all the combinations of machine sequences. These results are also compared with a quadratic linear program (QLP) conceived for the FJSSP for the minimization of the due-date mean square deviation (MSD). In the last section of this appendix, results from the simulation study described in Sections V-4 are also reported.

H-I. Efficiency of the genetic algorithm

The genetic algorithm, described in Appendix F was conceived to minimize the due-date mean square deviation (MSD) for the common due-date case. As mentioned in Appendix F, for the common due-date problem two types of MSD problems can be defined: the unconstrained MSD problem and the constrained MSD problem. A simulation study was carried out for both types of MSD cases and results were compared with those obtained from a quadratic linear program (QLP). The QLP used for comparison was inspired from the Mixed-Integer Linear Program (MILP) proposed by Trentesaux et al., (2013). Such model already contains the necessary constraints to handle transportation times, intermediate storage capacity and the maximum number of jobs, agreeing with the AIP cell case. However, the model proposed by Trentesaux et al. was conceived for the minimization of the makespan, thus it had not been tested for the minimization of the MSD. The genetic algorithm (GA) was compared with a QLP using three benchmarks, one inspired from a real manufacturing cell (the AIP-PRIMECA cell) and two adapted from the related literature.

Ten different client orders, ranging from 21 to 96 operations, were tested. For each client order, six different due dates were studied. Eq. H-1 is based on the work of Sourd and Kedad-Sidhoum, (2003) that allows due dates to be selected according to two range factors: φ, which controls tightness, and ω, which controls due-date variance. Since transportation is a key completion time contributor, the worst case scenario for the total transportation time of the order TT_o was used in the formulation. For all scenarios, $\varphi = 0.2$ and $\omega = 0.2$ were chosen in order to obtain tight due dates, for which the constrained MSD problem was most likely to occur.

$$d_{cc} = U \left[\begin{array}{c} \max\left(0, \left(1 - \varphi - \frac{\omega}{2}\right) * \left(tt_o + \sum_{j \in J, i \in I_j} pij\right)\right), \\ \left(1 - \varphi + \frac{\omega}{2}\right) * \left(tt_o + \sum_{j \in J, i \in I_j} pij\right) \end{array} \right] \qquad \text{Eq. H-1}$$

Results obtained from the GA were compared with the solutions given by the QLP (i.e., *MSD'qlp*). The gap between the GA and the QLP is expressed in Eq. H-2.

$$\text{Gap} = \frac{MSD' - MSD'qlp}{MSD'qlp} * 100\% \qquad \text{Eq. H-2}$$

a. Results and analysis for the AIP-PRIMECA cell

For this evaluation protocol, the GA was solved using MatLab, running on an Intel Pentium® D PC with a 3.40 GHz processor and 1GB RAM. The QLP was solved using IBM Cplex 12.2 Concert Technology, running on an Intel Dual-Core PC with a 1.73 GHz Processor and 2GB RAM. Results from the 60 tests are shown in Table H-1. For the GA, the reported results are the average MSD' for 20 trials for each test case, the minimum and maximum solutions for the MSD', the computing cost and its minimum and maximum solutions.

Performance comparison: As expected, the QLP had difficulty finding solutions for large problems. In this particular case, the QLP found optimal solutions for up to 4-job orders and some good heuristic solutions for 6-job orders. For the rest of the orders, the QLP found heuristic solutions and these results were, most of the time, outperformed by the GA (if the QLP found a solution after running for 5 hours). Figure H-1 shows the results for the first 18 cases because, for these cases only, we can make a fair comparison. The good behavior of the GA is represented not only for the very low gaps, but also for the low variation of its solutions. The largest QLP gap was 2.93% (compared with a lower bound) and the largest GA variability for this set of results was 7% around the average value.
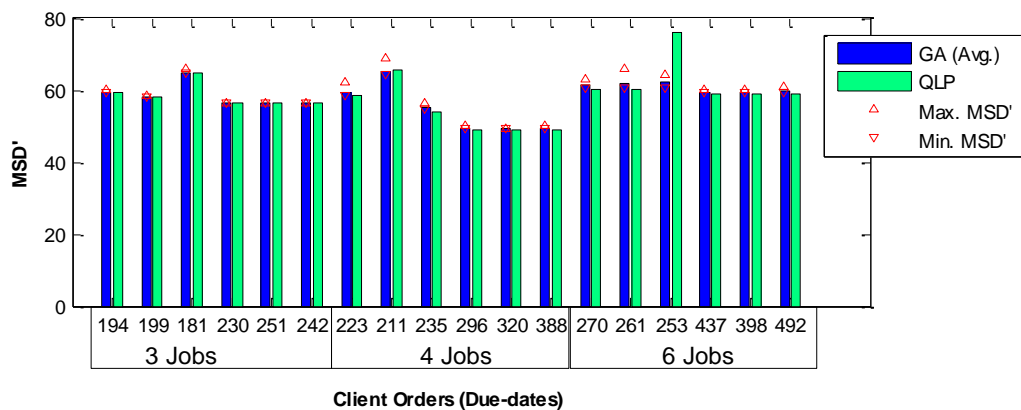
Figure H-1: MSD' for the first 18 cases (Sc0t01-01, 02 and 03)

Table H-1: Results for 60 tests with an increasing number of jobs

| ID test	Order	Due-min	Due-max	Due-date	MJ	Avg. MSD'	Min MSD'	Max MSD'	Avg. Comp. Time (s)	Min. Comp. Time (s)	Max. Comp Time (s)	MSD' (QLP)	Time (QLP)	% Exploration	GAP (%)	MSD Variability (R)
Sc0t01-01	1 AIP	180	252	194	2	59,54	59,47	60,19	1,83	1,71	2,41	59,47	2,49	Optimal	0,12	0,72
	1 AIP			199		58,20	58,09	58,63	1,74	1,70	1,79	58,09	2,45	Optimal	0,19	0,54
	1 AIP			181		64,86	64,74	65,86	1,73	1,70	1,84	64,74	3,38	Optimal	0,19	1,13
	1 AIP			230		56,41	56,41	56,41	1,78	1,72	2,19	56,34	2,4	Optimal	0,13	0,00
	1 AIP			251		56,41	56,41	56,41	1,75	1,72	1,87	56,34	2,47	Optimal	0,13	0,00
	1 AIP			242		56,41	56,41	56,43	1,77	1,72	2,18	56,34	3,08	Optimal	0,13	0,03
Sc0t01-02	1 LATE	206	392	223	3	59,19	58,57	62,47	4,45	3,41	6,26	58,56	47,13	Optimal	1,09	3,89
	1 LATE			211		65,59	64,52	69,07	5,13	3,39	7,08	65,52	128,38	Optimal	0,10	4,55
	1 LATE			235		55,13	54,65	56,25	4,49	3,40	7,42	53,98	92,39	Optimal	2,14	1,60
	1 LATE			296		49,37	49,30	50,41	3,98	3,47	5,09	48,91	32,42	Optimal	0,93	1,10
	1 LATE			320		49,35	49,30	49,46	4,12	3,50	6,58	48,91	33,4	Optimal	0,90	0,16
	1 LATE			388		49,37	49,30	50,21	4,29	3,52	5,36	48,91	89,02	Optimal	0,94	0,90
Sc0t01-03	2 AIP	250	504	270	4	61,30	60,81	63,10	15,03	10,45	18,79	60,32	3600	54,25	1,62	2,30
	2 AIP			261		61,82	60,81	65,87	15,02	10,40	22,80	60,06	3600	14,82	2,93	5,07
	2 AIP			253		62,24	60,81	64,33	16,27	10,29	22,05	75,93	3600	98,16	-18,03	3,52
	2 AIP			437		59,46	59,36	60,28	13,70	10,42	18,79	58,8	3600	21,79	1,12	0,92
	2 AIP			398		59,46	59,36	60,29	14,20	10,56	21,67	58,8	3600	55,07	1,12	0,92
	2 AIP			492		59,59	59,36	60,88	14,09	10,38	18,84	58,806	3600	0,27	1,33	1,52
Sc0t01-04	1 BELT + 1 AIP	262	728	291	4	74,79	70,91	86,60	26,74	14,97	38,21	87,7	18000	97,78	-14,72	15,69
	1 BELT + 1 AIP			267		84,67	78,92	95,17	29,19	14,80	48,46	100,74	18000	98,12	-15,95	16,25
	1 BELT + 1 AIP			302		72,22	69,74	77,36	26,01	14,94	40,83	81,55	18000	59,54	-11,44	7,63
	1 BELT + 1 AIP			610		71,36	69,71	78,18	26,75	15,01	38,63	67,52	18000	30,85	5,69	8,47
	1 BELT + 1 AIP			546		70,55	69,71	75,70	28,68	17,79	48,91	67,52	18000	52,45	4,48	5,99
	1 BELT + 1 AIP			709		71,11	69,71	73,64	28,81	17,54	39,85	67,52	18000	12,98	5,31	3,93
Sc0t01-05	2 LATE	279	784	284	4	114,09	111,59	116,74	35,57	20,78	53,83	124,69	18000	99,75	-8,50	5,15
	2 LATE			298		108,53	101,26	115,18	37,13	20,85	91,70	120,03	18000	99,52	-9,58	13,91
	2 LATE			311		102,31	96,70	109,50	39,65	20,71	63,99	128,51	18000	99,62	-20,39	12,80
	2 LATE			573		95,00	92,48	101,01	40,35	20,87	59,45	106,34	18000	98,89	-10,67	8,53
	2 LATE			668		95,51	92,48	100,95	36,84	22,73	67,47	99,52	18000	99,28	-4,03	8,47
	2 LATE			731		95,25	92,48	100,10	39,03	20,91	59,45	102,53	18000	98,81	-7,10	7,62
Sc0t01-06	1 BELT + 1 LATE	284	798	289	4	111,87	104,73	116,52	36,32	20,65	63,36	139,59	18000	99,39	-19,86	11,79
	1 BELT + 1 LATE			292		110,00	103,39	114,56	38,35	20,41	61,05	128,26	18000	99,79	-14,24	11,16
	1 BELT + 1 LATE			303		105,21	99,13	112,14	35,43	22,11	52,20	NR	18000	N/A	N/A	13,01
	1 BELT + 1 LATE			589		94,96	92,84	97,69	36,73	20,73	56,98	106,35	18000	99,56	-10,71	4,85
	1 BELT + 1 LATE			686		95,53	92,84	99,98	39,37	22,53	58,04	NR	18000	N/A	N/A	7,14
	1 BELT + 1 LATE			797		96,38	92,84	101,15	34,16	20,80	51,58	131,06	18000	99,75	-26,46	8,31
Sc0t01-07	3 AIP	304	816	323	4	96,97	94,91	102,84	54,06	33,83	83,42	181,6	18000	99,89	-46,60	7,92
	3 AIP			312		101,06	98,54	106,40	48,47	26,84	73,32	NR	18000	N/A	N/A	7,86
	3 AIP			307		100,77	98,56	103,32	57,74	34,01	82,41	138,73	18000	99,9	-27,36	4,75
	3 AIP			752		92,69	91,44	94,46	53,17	27,10	72,25	109,44	18000	99,64	-15,31	3,02
	3 AIP			644		92,89	91,44	95,17	48,99	34,48	72,03	112,84	18000	99,71	-17,68	3,74
	3 AIP + 1 BELT			689		93,07	91,44	95,63	52,82	29,58	74,24	110,56	18000	99,77	-15,82	4,19
Sc0t01-08	2 AIP + 1 BELT	315	839	321	4	128,06	119,65	137,00	46,90	25,67	78,17	NR	18000	N/A	N/A	17,35
	2 AIP + 1 BELT			327		125,89	120,51	130,94	48,38	27,57	77,72	NR	18000	N/A	N/A	10,43
	2 AIP + 1 BELT			317		129,43	125,41	137,34	44,14	25,28	62,63	308,86	18000	99,96	-58,09	11,92
	2 AIP + 1 BELT			682		111,95	108,02	116,64	48,90	25,73	72,18	253,63	18000	99,95	-55,86	8,62
	2 AIP + 1 BELT			759		113,11	110,31	117,35	44,05	30,11	58,22	NR	18000	N/A	N/A	7,04
	2 AIP + 1 BELT			825		112,72	107,30	119,19	52,95	34,03	102,10	NR	18000	N/A	N/A	11,89
Sc0t01-09	2 BELT + 1 AIP	323	847	358	4	150,18	141,63	153,54	50,63	32,23	84,73	NR	18000	N/A	N/A	11,91
	2 BELT + 1 AIP			416		136,64	132,79	141,38	62,02	37,54	85,93	NR	18000	N/A	N/A	8,59
	2 BELT + 1 AIP			403		138,24	132,22	147,54	63,17	32,75	92,78	NR	18000	N/A	N/A	15,32
	2 BELT + 1 AIP			749		136,12	133,22	139,91	63,56	37,97	96,28	NR	18000	N/A	N/A	6,68
	2 BELT + 1 AIP			691		135,60	131,78	142,08	61,48	40,37	81,59	NR	18000	N/A	N/A	10,30
	2 BELT + 1 AIP			563		136,15	133,05	140,96	58,63	35,27	81,95	NR	18000	N/A	N/A	7,91
Sc0t01-10	2 LATE + 1 BELT	329	851	390	4	168,23	161,78	175,63	81,86	53,79	119,30	NR	18000	N/A	N/A	13,85
	2 LATE + 1 BELT			412		164,15	157,31	173,02	69,10	40,80	126,51	NR	18000	N/A	N/A	15,72
	2 LATE + 1 BELT			456		159,45	154,50	167,74	72,13	44,38	120,68	NR	18000	N/A	N/A	13,24
	2 LATE + 1 BELT			632		159,05	156,09	164,42	73,82	50,87	91,48	NR	18000	N/A	N/A	8,33
	2 LATE + 1 BELT			619		158,77	155,03	163,58	77,89	44,38	133,77	NR	18000	N/A	N/A	8,55
	2 LATE + 1 BELT			703		158,83	153,18	166,10	74,70	41,03	136,76	NR	18000	N/A	N/A	12,92

NR= No result for the prescribed time   N/A= not applicable

For larger instances, the QLP ran for 5 hours and still did not find any good solutions or no solutions at all (e.g., instances Sc0t01-08 thru Sc0t01-10). The largest GA variability attained, reported for a 10-job order, was 14% around the average value.

Computing cost: Figure H-2 reports the computational burden of the two methods. For orders of 6 jobs or more, the QLP was stopped between 1 and 5 hours while the GA only needed just over 2 minutes to find a solution (i.e., for a 12-job order).
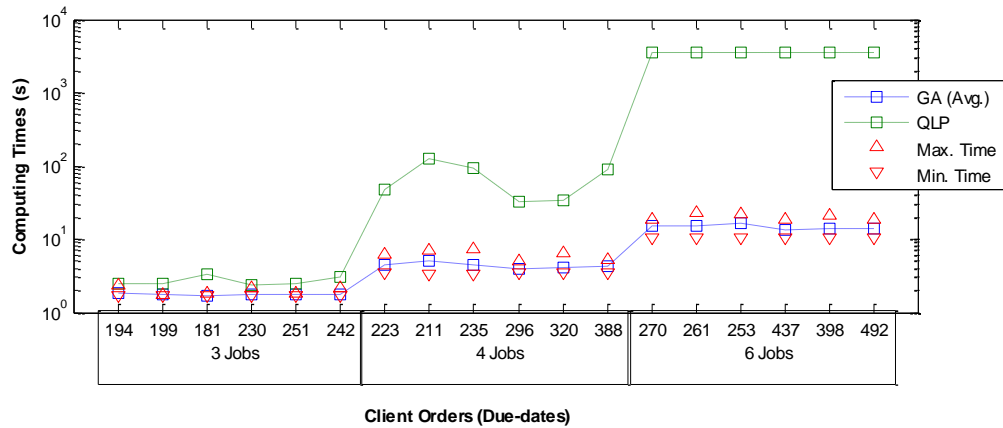
Figure H-2: Computing time for the first 18 cases (Sc0t01-01, 02 and 03)

The GA's behavior: Figure H-3 shows the GA's behavior for the best MSD' and the average MSD' for a 6-job order and a 398-second due date. This figure shows the good and rapid convergence of the best MSD' and the average MSD'.
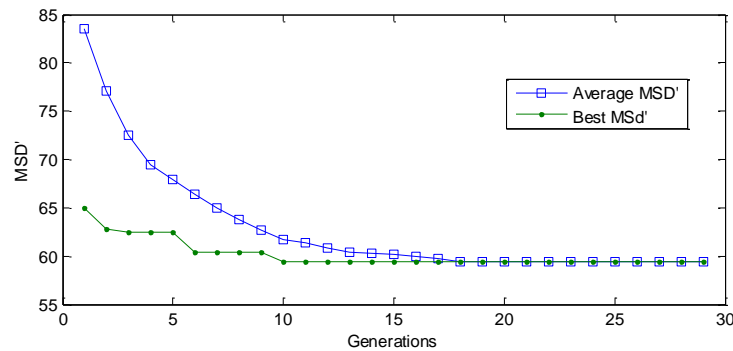
Figure H-3: Best MSD' and average MSD' for a client's 6-job order

An additional advantage of our GA over the QLP is the calculation of release times. As shown in the Gantt diagrams in Figure H-4, although both methods obtained very similar results for this given case (gap of 0.8%), the GA released the jobs 120 seconds after the QLP did. This behavior was accomplished by the shifting process that the GA executes over the schedule, due to the unconstrained nature of this particular problem. To accomplish the same

192

behavior with the QLP it would be necessary to use a multi-objective function, integrating flow time minimization with the MSD function.
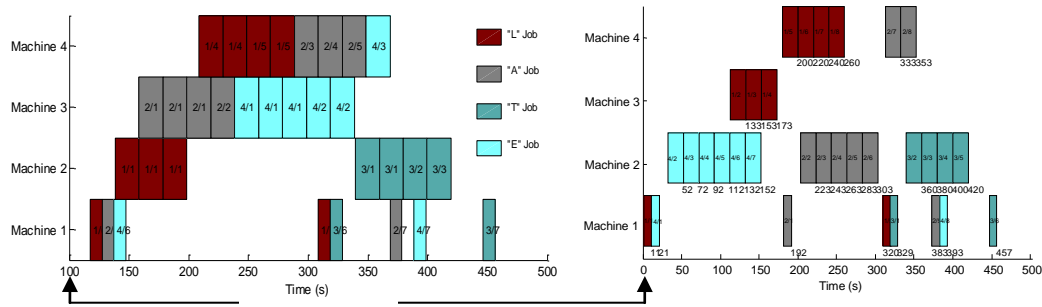
a) GA solution   b) QLP solution

Figure H-4: Gantt charts for the scenario Sc0t01-02: 1 "LATE" order with a 388-second due date.

The encouraging results obtained with our GA validate our choices, especially regarding the chromosome encoding technique. Our technique guarantees the minimal chromosome length, which equals the number of jobs in the order. For instance, the 10-job problem (2 LATE+1 BELT) has 96 operations. Our chromosome length remains at 10, whereas with an operation-based representation, the chromosome length would be increased up to 96 genes. Handling the genetic operators for that number of genes is much more complex, and the solution is more costly.

b. Results and analysis from two benchmarks from the literature

Though GA and QLP results cannot be compared with results reported by other authors (due to the objective function), two sets of problem instances taken from the FJSSP literature have been considered in order to compare the QLP and the GA proposed. The reason for executing such evaluation is that these problem instances provide more generality (partial and full flexibility), thus a larger search space:

✓ Kacem instances: the three problem instances from Kacem et al., (2002): problem *8x8*, problem *10x10* and problem *15x10* are used for comparison. The first problem has partial flexibility and a total of 27 operations. The second and third problems have full flexibility and a total of 30 and 56 operations respectively.

✓ BR instances: the first ten instances proposed by Brandimarte, (1993), denoted *mk1* to *mk10* are selected for comparison. The number of jobs varies from 10 to 20, the number of machines from 6 to 15 and the number of operations from 58 to 232.

For each problem instance, two due dates were tested in order to obtain one constrained and one unconstrained MSD problem. Results from the GA (mean of 10 independent trials)

193

were compared with those obtained with the QLP. For these tests, transport times, queue capacity and the maximum number of jobs were not considered as they are not taken into consideration by the benchmarks. Given the machine-sequence flexibility in these problem instances, a simple local search procedure focused on searching a better allocation for the last operation of each job was executed on the final solution given by the GA. Three random instances, of different size and with different data to those used for validation, were used for this purpose. The only parameter that required adjustment from the AIP cell was the population size (σ), which was increased from 10 (Appendix F) to 3000. This change is evident since the search space for these problem instances is larger than that of the AIP cell case. Table H-2 and Table H-3 show the results for Kacem's instances, for the constrained and unconstrained problems, respectively. For these instances, results with local search (GA^{+LS}) and without local search (GA^{-LS}) are reported. The solution improvement obtained with the simple local search procedure ranged from 10 to 38%.

Table H-2: Results for Kacem's instances – Constrained problems

Instance	Due date (s)	GA^{-LS}		GA^{+LS}		QLP		Gap (GA^{+LS}/QLP) (%)
		MSD'	CT(s)	MSD'	CT(s)	MSD'	CT(s)	
8 x 8	10	4.64	97	4.08	100	2.93	3600	39
10 x 10*	10	0.69	197	0.51	206	0.22	3600	N/A
15 x 10	10	7.58	430	6.11	454	N/S	oom	N/A

N/A: not applicable   N/S: no solution found   oom: out of memory

Table H-3: Results for Kacem's instances – Unconstrained problems

Instance	Due date (s)	GA^{-LS}		GA^{+LS}		QLP		Gap (GA^{+LS}/QLP) (%)
		MSD'	CT(s)	MSD'	CT(s)	MSD'	CT(s)	
8 x 8*	100	1.14	74	0.70	78	0	74.68	N/A
10 x 10*	100	0.95	101	0.79	117	0	26.65	N/A
15 x 10	100	2.07	551	1.86	1.36	30.12	3600	-93

N/A: not applicable

In Table H-2 and Table H-3, instances marked with a star (*) represent those for which the QLP found the optimal solution. The very high gaps (>39%) for these particular problems do not mean that the GA performance was poor. The optimal solution is zero (or <1), so any small deviation results in a considerable gap. For instance, for the *8x8* problem, a deviation of one second for only one job, leads to a gap of 35%. Table H-4 reports the results for the BR instances.

In general terms, the GA^{+LS} exceeded the QLP for almost all the problems (negative gaps) and the QLP was unable to find an optimal solution for any of them. The QLP only reported better results than the GA^{+LS} for one problem: *Mk1* with a due date set at 100s (gap=54%). For this case the completion time deviations between the GA and the QLP do not go over 2 seconds.

Table H-4: Results for BR instances

Inst.	Constraint Problems						Unconstraint Problems					
	Due date (s)	GA^{+LS}		QLP		Gap (GA^{+LS},QLP) (%)	Due date (s)	GA^{+LS}		QLP		Gap (GA^{+LS}/QLP) (%)
		MSD'	CT (s)	MSD'	CT (s)			MSD'	CT (s)	MSD'	CT (s)	
Mk1	10	17.4	107	22.1	3600	-21	100	54	252	0.77	3600	54
Mk2	10	10.6	153	113.5	3600	-90	100	-48	178	1.72	3600	-48
Mk3	10	176.4	401	N/S	oom	N/A	300	N/A	639	N/S	oom	N/A
Mk4	30	34.1	356	40.4	3600	-24	300	-6	656	3.43	3600	-6
Mk5	40	93.9	585	180.7	3600	-48	350	-30	437	14.17	3600	-30
Mk6	70	25.6	625	N/S	oom	N/A	350	N/A	300	N/S	oom	N/A
Mk7	70	9.24	678	N/S	oom	N/A	500	N/A	1190	N/S	oom	N/A
Mk8	100	116.8	718	244.9	3600	-60	500	-64	1065	34.82	3600	-64
Mk9	100	200.3	849	N/S	oom	N/A	750	N/A	1643	N/S	oom	N/A
Mk10	120	126.3	1118	N/S	oom	N/A	850	N/A	1070	N/S	oom	N/A

N/A: not applicable   N/S: no solution found   oom: out of memory

H-II. Efficiency of the optimization module

In this section we report the results obtained from the two connection types (i.e., also named hybrid approach) between the genetic algorithm and the ATC: the decoupled (denoted as SG-ATC) and the coupled coupling (IG-ATC) as presented in Zambrano Rey et al., (2014). We compared the two approaches against the quadratic linear program (QLP) to determine the gap with optimal results, the ATC with full exploration of machine sequences (detailed ATC, DATC) to determine the impact of the genetic algorithm in the searching process, and the pure version of the GA to determine the gain of introducing the ATC.

The two hybrid approaches, the DATC and he GA were programmed in MatLab$^{®}$, running on an Intel Pentium$^{®}$D PC with a 3.40 GHz processor and 1 GB of RAM. The QLP was solved using IBM Cplex 12.2 Concert Technology, running on an Intel Dual-Core PC with a 1.73 GHz Processor and 2GB of RAM. Herein, results from simulations for static scenarios are reported. All results are detailed in Table H-5. Figure H-5 depicts the tendency for an increasing number of jobs, by plotting the case of one particular due-date for each scenario "*Sc0t01-0#*" in Table H-5. MSD' values are drawn in two figures depending on the type of MSD problem, constrained or unconstrained. The tables accompanying the figures present the scenario number, the number of jobs considered (which coincides with those in the horizontal axis of the figure) and the due date set for that particular instance. As with the GA, comparisons with optimal results (QLP) can be made for 3- and 4-jobs orders. Up to 9- and 10-job orders, the QLP had to be stopped and some heuristic solutions were found, but not for all cases. For orders with more than 10 jobs there are no solutions at all.
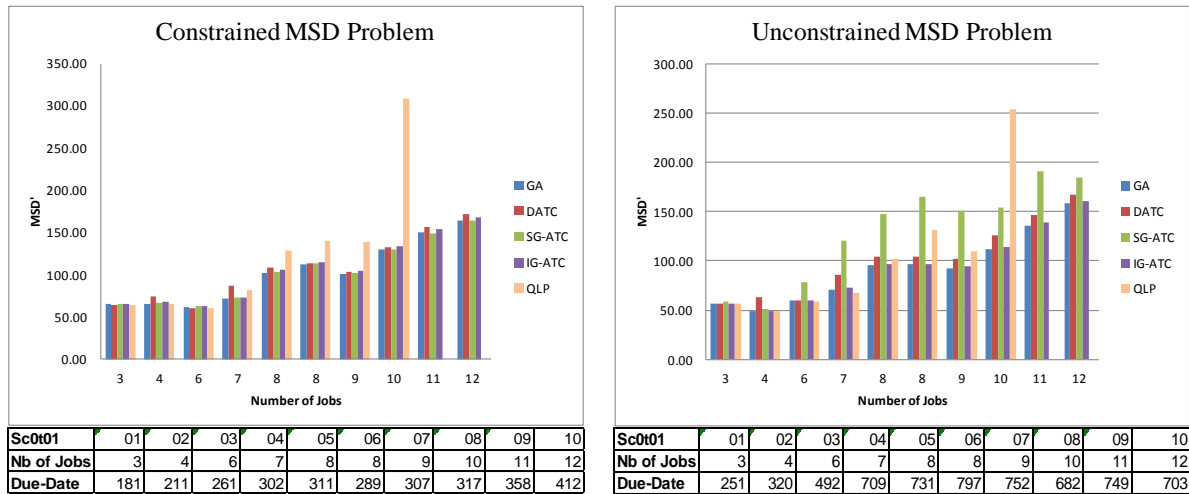
Figure H-5: Some results from the static scenario.

The IG-ATC reports the lower gaps compared to the SG-ATC that behaves badly for the unconstrained MSD problems, reaching an 80% gap. This behavior is due to the lack of feedback between the ATC and the GA. The job sequence found by the GA is related with the machine sequence chosen, and since the ATC finds new job sequences, these do not match well the machine sequences proposed by the GA. By coupling the ATC with the GA, both algorithms achieve better results and display a more consistent behavior for the two kinds of problems.

Compared to the ATC with complete exploration (DATC) hybrid approaches reported similar results for some tests, meaning that the GA accomplishes well its machine-sequence screening process. For other cases, hybrid approaches proposed better solutions than the complete exploration, because the GA also explores the job sequence and the ATC is sensitive to the initial conditions.

Table H-5: Simulation results for static scenarios.

ID test	Order	Due min	Due max	Due-date	MJ	Type MSD Problem	GA Avg. MSD'	GA Avg. Comp. Time (s)	DATC Avg. MSD'	DATC Avg. Comp. Time (s)	SG-ATC Avg. MSD'	SG-ATC Avg. Comp. Time (s)	IG-ATC Avg. MSD'	IG-ATC Avg. Comp. Time (s)	QLP MSD'	QLP Time	QLP % Exploration	GAP (%) GA	GAP(%) DATC	GAP (%) SG-ATC	GAP (%) IG-ATC
Sc0t01-01	1 AIP	180	252	194	2	Constrained	59,54	1,83	59,47	1,25	59,52	1,91	60,26	4,96	59,47	2,49	Optimal	0,12	0,00	0,09	1,33
	1 AIP			199			58,20	1,74	58,09	0,90	58,18	1,81	58,79	5,68	58,09	2,45	Optimal	0,19	0,00	0,14	1,20
	1 AIP			181			64,86	1,73	64,74	0,89	64,79	1,84	65,48	5,52	64,74	3,38	Optimal	0,19	0,00	0,09	1,14
	1 AIP			230		Unconstrained	56,41	1,78	56,41	0,96	56,38	2,06	56,46	5,13	56,34	2,4	Optimal	0,13	0,13	0,08	0,22
	1 AIP			251			56,41	1,75	56,64	1,24	58,63	1,95	56,77	5,97	56,34	2,47	Optimal	0,13	0,53	4,07	0,77
	1 AIP			242			56,41	1,77	56,59	1,13	56,65	1,85	56,69	5,84	56,34	3,08	Optimal	0,13	0,45	0,55	0,62
Sc0t01-02	1 LATE	206	392	223	3	Constrained	59,19	4,45	69,36	3,95	59,60	4,83	62,23	11,01	58,56	47,13	Optimal	1,09	18,46	1,79	6,27
	1 LATE			211			65,59	5,13	73,64	3,93	66,04	5,16	67,83	10,72	65,52	128,38	Optimal	0,10	12,39	0,80	3,52
	1 LATE			235			55,13	4,49	66,99	3,88	55,38	4,95	56,32	10,17	53,98	92,39	Optimal	2,14	24,09	2,59	4,33
	1 LATE			296		Unconstrained	49,37	3,98	63,34	4,56	49,46	5,13	49,57	12,82	48,91	32,42	Optimal	0,93	29,50	1,12	1,35
	1 LATE			320			49,35	4,12	63,34	5,11	51,36	4,43	49,52	14,66	48,91	33,4	Optimal	0,90	29,50	5,00	1,25
	1 LATE			388			49,37	4,29	63,34	7,39	60,72	4,79	49,90	21,64	48,91	89,02	Optimal	0,94	29,50	24,15	2,03
Sc0t01-03	2 AIP	250	504	270	4	Constrained	61,30	15,03	60,81	47,45	61,73	16,54	62,29	40,34	60,32	3600	54,25	1,62	0,81	2,34	3,27
	2 AIP			261			61,82	15,02	60,87	45,66	62,24	16,78	62,27	44,06	60,06	3600	14,82	2,93	1,34	3,63	3,67
	2 AIP			253			62,24	16,27	60,81	45,58	62,01	16,34	62,39	44,72	75,93	3600	98,16	-18,03	-19,92	-18,34	-17,83
	2 AIP			437		Unconstrained	59,46	13,70	59,37	80,67	73,03	15,62	59,68	74,66	58,8	3600	21,79	1,12	0,97	24,21	1,50
	2 AIP			398			59,46	14,20	59,37	70,27	72,28	17,54	59,88	60,20	58,8	3600	55,07	1,12	0,97	22,93	1,84
	2 AIP			492			59,59	14,09	59,38	89,48	77,78	15,85	59,67	83,90	58,806	3600	0,27	1,33	0,97	32,27	1,47
Sc0t01-04	1 BELT + 1 AIP	262	728	291	4	Constrained	74,79	26,74	88,58	114,87	74,53	25,10	74,84	78,70	87,7	18000	97,78	-14,72	1,00	-15,01	-14,66
	1 BELT + 1 AIP			267			84,67	29,19	93,14	117,78	83,95	27,45	86,06	71,16	100,74	18000	98,12	-15,95	-7,54	-16,66	-14,57
	1 BELT + 1 AIP			302			72,22	26,01	87,41	114,28	72,47	29,00	73,38	76,00	81,55	18000	59,54	-11,44	7,19	-11,13	-10,01
	1 BELT + 1 AIP			610		Unconstrained	71,36	26,75	85,39	206,53	116,72	25,68	73,25	163,04	67,52	18000	30,85	5,69	26,46	72,87	8,49
	1 BELT + 1 AIP			546			70,55	28,68	85,58	191,49	106,18	24,84	72,44	151,40	67,52	18000	52,45	4,48	26,74	57,26	7,28
	1 BELT + 1 AIP			709			71,11	28,81	85,50	226,79	121,00	28,09	72,53	160,49	67,52	18000	12,98	5,31	26,63	79,20	7,42
Sc0t01-05	2 LATE	279	784	284	4	Constrained	114,09	35,57	115,68	581,41	114,63	33,95	117,55	82,93	124,69	18000	99,75	-8,50	-7,22	-8,07	-5,72
	2 LATE			298			108,53	37,13	111,40	569,20	108,10	43,46	110,42	93,58	120,03	18000	99,52	-9,58	-7,19	-9,94	-8,01
	2 LATE			311			102,31	39,65	108,89	567,52	103,12	40,39	105,33	92,19	128,51	18000	99,62	-20,39	-15,27	-19,76	-18,04
	2 LATE			573		Unconstrained	95,00	40,35	103,39	911,94	118,26	38,20	96,03	188,33	106,34	18000	98,89	-10,67	-2,78	11,21	-9,69
	2 LATE			668			95,51	36,84	103,90	1028,31	136,89	34,14	97,37	188,44	99,52	18000	99,28	-4,03	4,40	37,55	-2,16
	2 LATE			731			95,25	39,03	104,15	1103,26	147,39	37,79	96,50	214,36	102,53	18000	98,81	-7,10	1,58	43,76	-5,88
Sc0t01-06	1 BELT + 1 LATE	284	798	289	4	Constrained	111,87	36,32	114,03	401,69	113,01	32,40	114,35	89,72	139,59	18000	99,39	-19,86	-18,31	-19,04	-18,08
	1 BELT + 1 LATE			292			110,00	38,35	113,22	397,07	110,71	33,38	113,01	97,77	128,26	18000	99,79	-14,24	-11,73	-13,68	-11,89
	1 BELT + 1 LATE			303			105,21	35,43	110,87	382,02	106,73	32,02	109,41	98,03	NR	18000	N/A	N/A	N/A	N/A	N/A
	1 BELT + 1 LATE			589		Unconstrained	94,96	36,73	103,62	626,58	127,59	35,18	98,39	144,03	106,35	18000	99,56	-10,71	-2,57	19,98	-7,48
	1 BELT + 1 LATE			686			95,53	39,37	103,90	699,86	151,36	33,33	98,30	174,35	NR	18000	N/A	N/A	N/A	N/A	N/A
	1 BELT + 1 LATE			797			96,38	34,16	103,82	768,96	164,78	33,39	96,65	203,52	131,06	18000	99,75	-26,46	-20,79	25,73	-26,26
Sc0t01-07	3 AIP	304	816	323	4	Constrained	96,97	54,06	103,30	1270,66	97,41	56,65	99,87	142,06	181,6	18000	99,89	-46,60	-43,12	-46,36	-45,01
	3 AIP			312			101,06	48,47	103,11	1280,32	100,90	50,79	102,83	125,57	NR	18000	N/A	N/A	N/A	N/A	N/A
	3 AIP			307			100,77	57,74	103,34	1304,91	102,06	46,42	105,10	131,20	138,73	18000	99,9	-27,36	-25,51	-26,44	-24,24
	3 AIP			752		Unconstrained	92,69	53,17	101,84	2526,98	150,54	64,26	94,75	276,93	109,44	18000	99,64	-15,31	-6,95	37,55	-13,43
	3 AIP			644			92,89	48,99	101,83	2265,69	133,17	63,17	94,37	247,36	112,84	18000	99,71	-17,68	-9,75	18,01	-16,37
	3 AIP			689			93,07	52,82	101,93	2384,78	137,53	60,12	93,81	266,98	110,56	18000	99,77	-15,82	-7,80	24,39	-15,15
Sc0t01-08	2 AIP + 1 BELT	315	839	321	4	Constrained	128,06	46,90	132,01	2914,88	129,41	42,38	134,37	103,32	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 AIP + 1 BELT			327			125,89	48,38	130,90	2907,48	126,56	47,14	130,82	110,84	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 AIP + 1 BELT			317			129,43	44,14	132,89	2924,55	130,03	45,14	133,79	110,61	308,86	18000	99,96	-58,09	-56,97	-57,90	-56,68
	2 AIP + 1 BELT			682		Unconstrained	111,95	48,90	126,38	4844,60	154,37	47,03	114,16	217,12	253,63	18000	99,96	-55,86	-50,17	-39,14	-54,99
	2 AIP + 1 BELT			759			113,11	44,05	126,19	5351,54	167,40	49,98	116,05	226,64	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 AIP + 1 BELT			825			112,72	52,95	126,16	5533,42	171,19	51,91	115,28	240,27	NR	18000	N/A	N/A	N/A	N/A	N/A
Sc0t01-09	2 BELT + 1 AIP	323	847	358	4	Constrained	150,18	50,63	156,26	6621,85	149,22	57,51	153,54	132,27	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 BELT + 1 AIP			416			136,64	62,02	146,40	6581,98	137,45	60,02	140,42	150,38	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 BELT + 1 AIP			403			138,24	63,17	147,54	6578,21	139,19	59,47	143,83	131,89	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 BELT + 1 AIP			749		Unconstrained	136,12	63,56	146,70	10624,27	191,37	51,73	139,02	239,46	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 BELT + 1 AIP			691			135,60	61,48	145,60	9913,77	167,42	59,72	139,05	221,32	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 BELT + 1 AIP			563			136,15	58,63	145,91	8020,05	145,99	63,82	137,54	183,35	NR	18000	N/A	N/A	N/A	N/A	N/A
Sc0t01-10	2 LATE + 1 BELT	329	851	390	4	Constrained	168,23	81,86	174,82	18000,00	169,57	72,81	174,92	155,32	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 LATE + 1 BELT			412			164,15	69,10	171,40	18000,00	164,01	76,71	167,59	176,96	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 LATE + 1 BELT			456			159,45	72,13	174,82	18000,00	158,14	81,61	163,11	160,97	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 LATE + 1 BELT			632		Unconstrained	159,05	73,82	167,86	18000,00	167,03	82,59	161,65	222,03	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 LATE + 1 BELT			619			158,77	77,89	168,8	18000,00	169,17	73,93	162,50	203,99	NR	18000	N/A	N/A	N/A	N/A	N/A
	2 LATE + 1 BELT			703			158,83	74,70	166,79	18000,00	184,18	76,68	160,39	302,99	NR	18000	N/A	N/A	N/A	N/A	N/A

NR= No result for the prescribed time   N/A= not applicable

Compared to the pure version of the GA, the IG-ATC behaves similarly for the common due date case. Indeed, the IG-ATC offers a more generic approach since it could also be used for non-common due date cases, for which the pure version of the GA is restricted. Figure H-6 a. reports the computing cost for the constrained MSD case and 6 b. for the unconstrained ones.
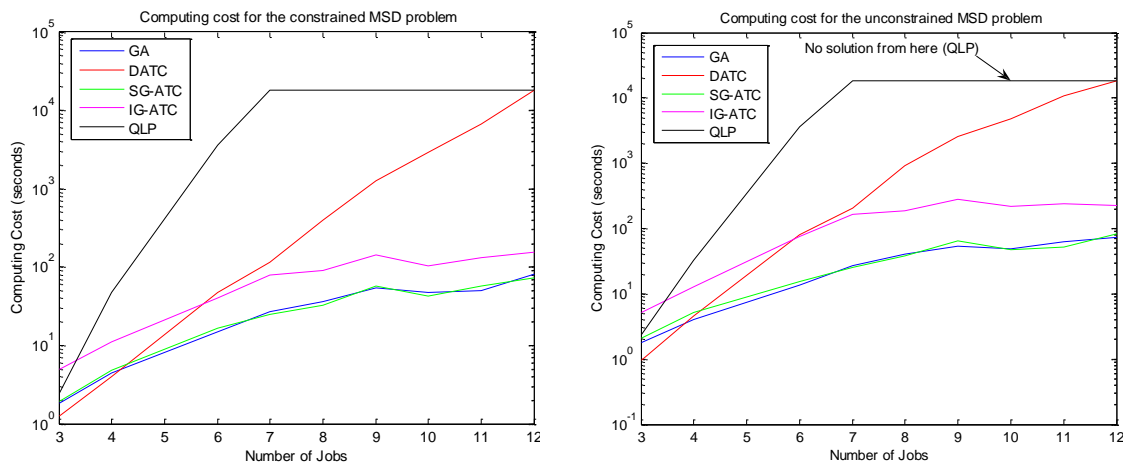
Figure H-6: Computing cost

From these graphs we can see how the second phase of the SG-ATC does not add any major cost, since the values are similar to those of the pure GA. It is also evident the increasing computational cost of the full exploration approach and the QLP, as the problem becomes more complex. The DATC had to be stopped at 5 hours for the 12-job problem in the constrained case and at 11- and 12-job problems in the unconstrained case. Starting from 7-job problems, the QLP was stopped at 5 hours, and sometimes after all this time it did not find a solution. The IG-ATC reports a similar behavior than the SG-ATC and the GA.

Even though results favor the IG-ATC, simulation results of 6 cases are subjected to an analysis of variance (ANOVA) *F*-test procedure (Girden, 1991) to look for an statistical difference that supports the choice. The different algorithms are considered as factors and the MSD' as the response variable. The assumptions of similar variances were examined and they hold true for all cases. Table H-6, Table H-7 and Table H-8 show the ANOVAs for 4-, 7- and 9-job orders (constrained MSD); and Table H-9, Table H-10 and Table H-11 report the ANOVAs for the same orders but for the unconstrained MSD problem. Table H-12 shows the Tukey's multiple comparison tests for the six cases.

Table H-6: Analysis of variance for a 4-job order (1 LATE) with due date at 211s

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	159.9734	3	53.3245	2.87	0.041881
Within groups	1414.0908	76	18.6065		
Total	1574.0642	79			

Table H-7: Analysis of variance for a 7-job order (1 BELT + 1 AIP) with due date at 267s.

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	798.1175	3	266.0392	13.21	<.0001
Within groups	1530.0377	76	20.1321		
Total	2328.1553	79			

Table H-8: Analysis of variance for a 9-job order (3 AIP) with due date at 307s.

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	672.5915	3	224.1972	11.67	<.0001
Within groups	1459.4798	76	19.2037		
Total	2132.0713	79			

Table H-9: Analysis of variance for a 4-job order (1 LATE) with due date at 388s.

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	2970.8778	3	990.2926	31.06	<.0001
Within groups	2422.7709	76	31.8786		
Total	5393.6488	79			

Table H-10: Analysis of variance for a 7-job order (1 BELT + 1 AIP) with due date at 709s.

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	36032.9472	3	12010.9824	293.31	<.0001
Within groups	3112.1726	76	40.9496		
Total	39145.1198	79			

Table H-11: Analysis of variance for a 9-job order (3 AIP) with due date at 644s.

Source	Sum of Squares	Df	Mean Square	F	P
Between groups	13351.5866	3	4450.5289	168.25	<.0001
Within groups	2010.2919	76	26.4512		
Total	15361.8785	79			

Table H-12: Pair wise comparison by Tukey's HSD Test.

Methods	Constrained MSD			Unconstrained MSD		
	4-jobs	7-jobs	9-jobs	4-jobs	7-jobs	9-jobs
M1 vs M2	NS	P<.01	P<.01	P<.01	P<.01	P<.01
M1 vs M3	NS	P<.01	P<.01	NS	NS	P<.01
M1 vs M4	P<.05	P<.01	P<.01	P<.01	P<.01	P<.01
M2 vs M3	NS	NS	NS	P<.01	P<.01	P<.01
M2 vs M4	NS	NS	NS	NS	NS	NS
M3 vs M4	NS	P<.05	NS	P<.01	P<.01	P<.01
M1=DATC	M2=GA	M3=SG-ATC		M4=IG-ATC		NS=non significance

For all cases, there is a significant statistically difference between the means MSD' values, with 95% confidence (*p-value=0.041*) for the case with a 4-job order and 99% for the other cases (*p-value<0.0001*). According to the Tukey's pair wise comparison (Table H-12), the IG-ATC is statistically different from the DATC with complete exploration (for all cases)

and with the SG-ATC just for the unconstrained MSD cases. In all cases, the IG-ATC offers a better performance than the two other approaches. Concerning the GA, there is no significant statistical difference with the IG-ATC for any case.

As a conclusion from these tests, we chose the IG-ATC approach as the most suitable for the global decisional entity, better responding to different due date problems and ensuring an excellent performance, compared to the other four approaches. However, further ANOVA tests could be executed to determine the impact of larger instances, and other flexibility degrees.

H-III. Results from the static simulation study

In this section, results from the simulation study described in Section V-4 are reported in Table H-13. CTV values reported in this table are mean values out of 10 independent trials. The only design point that generated an issue was |R'=3|, with FAM policy (FHFMS) and |P|=28. For this case, a lot of products travel in the inner loop close to machines m_2, m_3 and m_4 until they all get stuck and no shuttle can move. This situation is overcome with |R'=4| because products are attracted to the other side of the AIP cell, then the inner loop close to machines m_2, m_3 and m_4 is not overcrowded.

An additional conclusion that can be withdrawn from these data is the small gap between the purely heterarchical approach (FHFMS) and Instance A when the number of products gets over 24. Thus, when the FMS becomes saturated, dealing with myopic behavior is not worthy. However, as seen in these results, setting up a threshold is not an easy tasks because saturation not only dependant on the FMS capacity but also on the local decision-making algorithm, the machine-sequence flexibility and possibly on the product mixture. This latter hypothesis needs to be evaluated. Last, though the increasing computing cost makes necessary a more thorough study to determine the applicability of the global decisional level, based on these results, we can conclude that it is still worthy to deal with both types of myopic decisions, i.e., release sequence and machine allocation, even for a large number of products.

Table H-13: Result from simulation study

|R'|	FHFMS	|P|	FHFMS Product Mixture		FHFMS Single Type Product		Instance A			Instance B			Instance C			
			CTV	Computing Cost (s)	CTV	Computing Cost (s)	CTV	Computing Cost (s)	Gap (%)	CTV	Computing Cost (s)	Gap (%)	CTV	Computing Cost (s)	Gap (%) with Instance 0	Gap (%) with Instance A
3	PFA	4	1667	3	1653	3	917	10	45	225	19,4	86	125	25	93	86
		8	6892	4	9738	4	6091	19	12	1411	35	86	1268	50	82	79
		16	44978	6	46931	6	42255	51	6	9041	40	81	8350	141	81	80
		24	82293	9	105424	10	77698	101	6	19891	85	81	12973	375	84	83
		28	111018	12	140695	13	107550	94	3	27444	114	80	26136	476	76	76
	FAM	4	1885	3	1744	3	917	8	51	225	19,4	87	125	25	93	86
		8	10097	4	9730	4	7216	27	29	1411	35	85	1268	50	87	82
		16	52574	6	43532	6	39626	60	25	9041	40	79	8350	141	84	79
		24	96858	10	114330	10	78121	124	19	19891	85	83	12973	375	87	83
		28	120967	12	N/S	N/A	120967	154	0	27444	114	N/A	26136	476	78	78
4	PFA	4	4831	3	4873	3	1318	12	73	163	20,1	97	125	25	97	91
		8	7492	4	6106	4	2637	14	65	573	25	91	620	49	92	77
		16	23158	6	25385	7	16362	39	29	3082	40	88	3285	160	86	80
		24	42886	8	60603	9	34801	79	19	8091	77	87	5918	402	86	83
		28	60700	9	95916	11	48197	125	21	9976	103	90	16728	434	72	65
	FAM	4	2914	3	1367	3	1081	10	63	163	20,1	88	125	25	96	88
		8	7810	4	5408	4	2978	14	62	573	25	89	620	49	92	79
		16	24187	6	47678	7	18291	41	24	3082	40	94	3285	160	86	82
		24	46939	7	93271	9	38186	92	19	8091	77	91	5918	402	87	85
		28	64022	10	104263	11	50922	162	20	9976	103	90	16728	434	74	67

N/A: not applicable

201

Résumé

Introduction. Les systèmes manufacturiers doivent faire face à la globalisation des marchés, la personnalisation des produits, les exigences élevées des clients, la volatilité du marché et les cycles de vie des produits sans cesse réduits. Dans les dernières décennies, les avancées technologiques ont permis aux systèmes manufacturiers classiques d'évoluer vers des systèmes manufacturiers plus flexibles (FMS en anglais, Flexible Manufacturing Systems). Ces systèmes sont capables de s'adapter rapidement aux changements, aux tailles réduites de lots et à une grande variété de produits. Afin de contrôler ces systèmes flexibles, de nouvelles architectures, paradigmes, stratégies et algorithmes de contrôle ont vu le jour. Ceux-ci préconisent l'hétérarchie au lieu de la hiérarchie basée sur un contrôle local, peu complexe et hautement réactif supporté localement par des entités décisionnelles. Cependant, en dépit d'avancées prometteuses, de nouveaux problèmes ont aussi émergé (e.x, la prévisibilité, les coûts d'implémentation, les technologies et les normes) notamment celui lié à la garantie d'une performance opérationnelle minimale, et cela principalement à cause du comportement myope des entités décisionnelles locales. Même si ce comportement a été reconnu par certains chercheurs comme une barrière importante pour l'adoption du contrôle hétérarchique des FMS, jusqu'à maintenant ce comportement reste encore peu étudié.

Cette thèse se concentre sur le comportement myope des entités décisionnelles constituant les architectures de contrôle et plus particulièrement sur les approches permettant de réduire ce comportement. Les approches de contrôle hétérarchique des FMS existants se concentrent actuellement sur l'amélioration de la performance globale mais ne traitent pas explicitement le problème de myopie. En conséquence, ces approches deviennent plus complexes en entraînant la perte de certaines caractéristiques de l'hétérarchie telles que la réactivité, la tolérance aux perturbations et l'adaptabilité. Cette thèse est organisée en 5 chapitres.

Chapitre 1. Ce chapitre présente les définitions importantes du contrôle des systèmes manufacturiers, les FMS, ainsi que les algorithmes et les architectures de contrôle existants (centralisés, entièrement hiérarchiques, entièrement hétérarchiques et semi-hétérarchiques) qui peuvent être utilisés pour le contrôle des FMS. Pour d'étudier le comportement myope, une étude de la littérature a été réalisée dans des domaines autres que celui du contrôle des FMS afin de trouver des caractéristiques communes au niveau de la prise de décision. Le comportement myope résulte de la visibilité réduite que chaque entité décisionnelle locale a sur l'états actuel et futur du FMS. Cette visibilité réduite est la conséquence de la priorité donnée par les entités décisionnelles à leurs objectifs locaux plutôt qu'aux objectifs globaux. Deux dimensions de ce comportement myope ont aussi été identifiées : sociale et temporelle.

La dimension sociale traduit une capacité limitée soit à récupérer des donnée globales (notamment auprès des autres entités), soit à traiter correctement ces données lors de la prise de décision. La dimension temporelle peut être définie d'une part comme un temps limité pour la rechercher de solutions alternatives et d'autre part comme une capacité limitée à se projeter dans le futur et évaluer les conséquences à long terme des décisions de court terme. En plus d'être une barrière pour l'adoption industrielle du contrôle hétérarchique, ce comportement provoque d'autres effets indésirables comme la nervosité du système, le manque de prévisibilité et la perte de performance. Pour faire face à ce comportement myope, l'efficacité de la prise de décision au niveau local doit être renforcée, soit en utilisant des algorithmes de contrôle plus complexes, soit en intégrant des techniques supplémentaires aux algorithmes de prise de décision locale. La première possibilité est réalisable, par exemple, en utilisant des méthodes d'énumération ou en utilisant un historique des données pour mieux modéliser le système. La seconde possibilité peut être réalisée principalement en ajoutant des techniques de simulation/optimisation. Etant donné le spectre important des possibilités dans cette seconde voie, dans cette thèse nous avons choisi l'utilisation de l'optimisation et de la simulation pour faire face au comportement myope dans le contrôle des FMS basé sur l'hétérarchie.

Chapitre 2. Ce chapitre est consacré à une revue de l'état de l'art sur la réduction du comportement myope. Plusieurs approches ont été examinées et classées selon le type d'architecture de contrôle, entièrement hétérarchique (FHFMS: Full Heterarchical FMS control) ou semi-hétérarchique (SHFMS: Semi-Heterarchical FMS control) d'une part et la technique utilisée pour réduire ce comportement myope, l'optimisation, la simulation ou l'optimisation basée sur la simulation d'autre part. Suite à cette analyse de la littérature, une typologie générale pour la réduction du comportement myope a été réalisée. D'une manière générale, l'introduction de techniques simulation/optimisation (S/O) dans les FHFMS peut être réalisée dans le but de mieux évaluer les décisions locales, renforcer la coopération ou améliorer l'échange d'informations entre les entités décisionnelles. Cependant, l'architecture résultante peut nécessiter un coût plus élevé en communication et en traitement de l'information. La performance globale du système est alors améliorée par rapport aux approches de base, sans qu'on puisse le garantir. De plus, les techniques de S/O dans les SHFMS ont été principalement introduites avec cinq rôles différents : la « résolution», l'« évaluation », la « sélection », le « réglage » et l'« influence ». Les techniques S/O sont intégrées dans une entité décisionnelle globale qui, selon le rôle S/O, intervient directement ou indirectement dans le processus décisionnel des entités locales subordonnées. De cette analyse de l'état de l'art, une liste des exigences a été proposée à la fin de ce chapitre. L'objectif est de trouver un équilibre entre la réduction du comportement myope et la préservation des caractéristiques importantes des architectures hétérarchiques.

Chapitre 3. A partir de la typologie générale proposée pour la réduction du comportement myope, une approche de contrôle semi-hétérarchique des FMS pour réduire le comportement myope en intégrant des techniques d'optimisation basées sur la simulation est décrite dans ce chapitre. L'architecture proposée est basée sur deux niveaux décisionnels, un niveau global et un niveau local, dans lesquels résident des entités décisionnelles globales et locales. Chaque entité décisionnelle globale (GDE) se concentre sur un objectif global et peut gérer un ensemble d'entités décisionnelles locales (LDE). Si plusieurs GDE existent, des relations hétérarchiques sont établies entre elles. A l'intérieur d'une GDE, une boucle d'optimisation basée sur la simulation est implantée. Le problème de contrôle d'un FMS est alors découpé en plusieurs sous-problèmes de contrôle et pour chaque sous-problème une technique d'optimisation est configurée. Les techniques d'optimisation explorent l'espace de recherche de chaque sous-problème et le modèle de simulation permet d'évaluer l'efficacité de ces solutions par rapport à la fonction objectif globale. Une des principales nouveautés de cette approche est que l'entité globale se concentre uniquement sur les sous- problèmes de contrôle qui ont le plus d'impact sur la performance globale. Par conséquent, il est possible d'accepter le comportement myope dans le cadre de la stratégie de contrôle, car il permet des réponses rapides. Une autre nouveauté est la proposition de trois modes d'interaction entre les entités décisionnelles globales et locales : coercitif, limitatif et directif. Ces modes sont définis sur la base du rôle de la GDE et le degré d'autonomie souhaité pour le niveau local. La stratégie de contrôle dans des conditions normales et anormales est également décrite, ainsi que le processus pour générer une instance de l'approche proposée.

Chapitre 4. Ce chapitre décrit une instance de l'approche proposée dans le chapitre précédent. Tout d'abord, le problème du contrôle d'un FMS, ses paramètres, hypothèses et contraintes prises en compte sont décrits. Le problème de contrôle d'un FMS est divisé en plusieurs sous-problèmes : la séquence d'entrée des produits dans le FMS, l'allocation des tâches, le routage des produits et la séquence des produits sur chaque machine. Dans cette instance, une entité décisionnelle globale (pGDE) et deux types d'entités décisionnelles locales sont proposées: une entité machine (mLDE) et une entité véhicule (vLDEs). Le niveau global se concentre uniquement sur la réduction des décisions myopes des vLDEs. Dans le niveau global, la pGDE est dotée d'un algorithme génétique en charge de l'exploration du sous-problème d'allocation des tâches et d'un algorithme basé sur la théorie du contrôle pour la séquence d'entrée. Les deux algorithmes sont couplés et un modèle de simulation à base d'agents est utilisé pour évaluer les solutions proposées par les techniques d'optimisation. Dans le niveau local, les vLDEs sont dotées d'une technique basée sur les champs de potentiel pour réaliser l'allocation des tâches et le routage des produits. Les mLDEs utilisent une règle myope de priorité (premier entré-premier sorti) pour gérer la séquence des produits dans la file d'attente de chaque machine. Le mode d'interaction coercitif est utilisé pour intégrer les décisions de la pGDE dans le module de contrôle des

vLDEs. Par conséquent, la stratégie de contrôle dépend du type de perturbation. Les perturbations liées au routage des produits et la séquence des produits dans la file d'attente sont traitées localement par les vLDEs et mLDEs, tandis que les perturbations liées à l'allocation des tâches et la séquences d'entrée des produits sont traitées au niveau global.

Chapitre 5. L'instance décrite dans le chapitre précédent a été mise en œuvre et implémenté pour contrôler une cellule flexible d'assemblage afin d'évaluer l'approche proposée. Dans un premier temps, les données expérimentales sont détaillées puis la mise en œuvre de chaque entité décisionnelle est décrite. L'évaluation de l'approche proposée a été réalisée sur la base d'une étude en simulation, puis avec des expérimentations réelles sur la cellule flexible de l'AIP-PRIMECA de l'université de Valenciennes. L'étude en simulation a été réalisée pour divers scénarios statiques et dynamiques, en tenant compte de deux fonctions objectives : la variance du temps d'achèvement (CTV) et l'écart quadratique moyen autour d'une date d'échéance (MSD). Trois configurations différentes de l'entité décisionnelle globale ont été évaluées tout d'abord afin de déterminer l'applicabilité de notre approche lorsqu'il s'agit de différents types de décisions myopes. Dans tous les cas, le niveau global a permis d'atteindre une amélioration significative de la performance globale par rapport au scénario où il n'existait que le niveau local. Puis une étude en simulation avec des scenarios dynamiques, dans laquelle une tâche de maintenance curative affecte l'une des machines redondantes, a été réalisée. Les résultats ont montré que la performance globale peut être améliorée en ajoutant un niveau global, sans pour autant perdre en réactivité. Des expérimentations réelles sont été réalisées sur la cellule AIP-PRIMECA. Pour ces expérimentations, certains cas déjà évalués dans l'étude de simulation avec des scenarios dynamiques ont été réalisées et des résultats prometteurs ont été obtenus.

Conclusions et travaux futurs. Jusqu'à maintenant, la plupart des travaux concernant le contrôle hétérarchique des FMS ont porté sur l'amélioration de la performance globale en, traitant implicitement le comportement myope. Notre travail s'est focalisé sur une autre alternative dans laquelle un niveau global, dans une architecture semi- hétérarchique, est configuré pour traiter explicitement le comportement myope résultant de décisions de contrôle locales. Dans l'architecture semi-hétérarchique proposée, le niveau décisionnel local assure une certaine réactivité aux perturbations alors que le niveau décisionnel global se concentre sur la réduction du comportement myope et de son impact sur la performance globale. Par rapport à d'autres approches semi-hétérarchiques, notre approche a suivi une méthodologie plus granulaire dans laquelle le niveau décisionnel global permet une configuration flexible et modulaire pour faire face au comportement myope du niveau local. Par conséquent, le comportement myope peut être réduit en totalité ou en partie en fonction du nombre de sous-problèmes de contrôle pour lesquels le niveau de décision global est configuré.

Basé sur les exigences présentées dans le deuxième chapitre et le retour d'expérience de ces évaluations (en simulation et en réel) de l'architecture proposée, des perspectives de recherche pour le court, moyen et long terme ont pu être identifiées. Etant donné que l'approche proposée a montré des résultats prometteurs, l'étape suivante à court terme pourrait être de rendre l'ensemble de l'architecture plus dynamique et adaptatif, en présence de perturbations internes et externes. Principalement, une reconfiguration dynamique du module d'optimisation peut être mise en œuvre en fonction du type de perturbation et la criticité de celle-ci. Pour aller plus loin sur le comportement myope, une orientation vers le contrôle de la myopie peut être envisagée. Une étude détaillée du comportement myope au niveau décisionnel local, utilisant des modèles mathématiques, pourrait être faite afin de le mesurer et de le contrôler. Un aspect important qui est souvent négligé dans le contrôle des FMS est l'interaction humaine avec les entités décisionnelles artificielles. Par conséquent, une autre perspective à court et moyen terme est l'intégration d'entités humaines dans les niveaux globaux et locaux. Finalement, à long terme, notre approche doit aussi être évalué dans d'autres contextes, par exemple, dans le domaine hospitalier, les chaînes d'approvisionnement, la gestion des transports.

Réduction du Comportement Myope dans le contrôle des FMS : Une Approche Semi-Hétérarchique basée sur la Simulation-Optimisation

Résumé

Le contrôle hétérarchique des systèmes de production flexibles (FMS) préconise un contrôle peu complexe et hautement réactif supporté par des entités décisionnelles locales (DEs). En dépit d'avancées prometteuses, ces architectures présentent un comportement myope car les DEs ont une visibilité informationnelle limitée sue les autres DEs, ce qui rend difficile la garantie d'une performance globale minimum. Cette thèse se concentre sur les approches permettant de réduire cette myopie. D'abord, une définition et une typologie de cette myopie dans les FMS sont proposées. Ensuite, nous proposons de traiter explicitement le comportement myope avec une architecture semi-hétérarchique. Dans celle-ci, une entité décisionnelle globale (GDE) traite différents types de décisions myopes à l'aide des différentes techniques d'optimisation basée sur la simulation (SbO). De plus, les SbO peuvent adopter plusieurs rôles, permettant de réduire le comportement myope de plusieurs façons. Il est également possible d'avoir plusieurs niveaux d'autonomie en appliquant différents modes d'interaction. Ainsi, notre approche accepte des configurations dans lesquelles certains comportements myopes sont réduits et d'autres sont acceptés. Notre approche a été instanciée pour contrôler la cellule flexible AIP- PRIMECA de l'Université de Valenciennes. Les résultats des simulations ont montré que l'architecture proposée peut réduire les comportements myopes en établissant un équilibre entre la réactivité et la performance globale. Des expérimentations réelles ont été réalisées sur la cellule AIP-PRIMECA pour des scenarios dynamiques et des résultats prometteurs ont été obtenus.

MOTS CLES : Pilotage Semi-Hétérarchique, Myopie, Réactivité, Performance Globale, Simulation, Optimisation, FMS

Reducing Myopic Behavior in FMS Control: A Semi-Heterarchical Simulation-Optimization Approach

Abstract

Heterarchical-based control for flexible manufacturing systems (FMS) localizes control capabilities in decisional entities (DE), resulting in highly reactive and low complex control architectures. However, these architectures present myopic behavior since DEs have limited visibility of other DEs and their behavior, making difficult to ensure certain global performance. This dissertation focuses on reducing myopic behavior. At first, a definition and a typology of myopic behavior in FMS is proposed. In this thesis, myopic behavior is dealt explicitly so global performance can be improved. Thus, we propose a semi-heterarchical architecture in which a global decisional entity (GDE) deals with different kinds of myopic decisions using simulation-based optimization (SbOs). Different optimization techniques can be used so myopic decisions can be dealt individually, favoring GDE modularity. Then, the SbOs can adopt different roles, being possible to reduce myopic behavior in different ways. More, it is also possible to grant local decisional entities with different autonomy levels by applying different interaction modes. In order to balance reactivity and global performance, our approach accepts configurations in which some myopic behaviors are reduced and others are accepted. Our approach was instantiated to control the assembly cell at Valenciennes AIP-PRIMECA center. Simulation results showed that the proposed architecture reduces myopic behavior whereby it strikes a balance between reactivity and global performance. The real implementation on the assembly cell verified the effectiveness of our approach under realistic dynamic scenarios, and promising results were obtained.

Keywords: heterarchy, myopic behavior, simulation-based optimization, FMS, reactivity, global performance