NTNU
Norwegian University of
Science and Technology

Decentralized Decision Making in Multi-Agent Systems

Sindre Sønvisen Pedersen

Master of Science in Engineering Cybernetics
Submission date: June 2011
Supervisor: Bjarne Anton Foss, ITK
Co-supervisor: Johannes Tjønnås, SINTEF IKT

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of Science and Technology

Faculty of Information Technology, Mathematics and Electrical Engineering

Department of Engineering Cybernetics

Master project

Name of candidate: Sindre Pedersen

Subject: Engineering Cybernetics

Title: Decentralized decision making in Multi-agent systems
Title (in Norwegian): Desentralisert beslutningsstøtte i multi-agent systemer

Background
SINTEF participates in the EU research project GRACE (inteGration of pRrocess and quAlity Control using multi-agEnt technology). SINTEF will in the GRACE project develop control mechanisms at local and global level for a washing machine production line.. Furthermore, adaptivity will be implemented on the final product, by updating the control parameters of the on board micro controller based on the quality measures of the actual outcome of the assembly process. The report "Steps towards the integration of control theory in multi-agent system" by S.Pedersen is a part of this effort. This activity is continued in this master project.

Task description
Multi-agent systems may be perceived as an architecture for decentralized control of networked systems. This places constraints on the choice of algorithms since centralized methods are incompatible with such an architecture. In this work the candidate shall study and evaluate suitable algorithms for control and real-time optimization based on a multi-agent system architecture for production lines, through analysis and numerical simulation.

1. Develop a suitable example for the analysis. The example may preferably extend the example reported in project work TTK4500. Extensions may include a model description of a production line with more work stations and intermediate storage facilities, material feedback and an economic model.
2. Study the real-time resource allocation problem for a production system by describing this as an optimization problem, and by formulating alternative decentralized solution methods. This may include both heuristics and analytically based methods. Candidate methods are self-organizing approaches and a variety of decentralized optimization methods like Dantzig-Wolfe decomposition and distributed constraint optimization. Methods should be compared, if possible analytically and through suitable tests on the example developed in 1. Tests should include stochasticity by for instance using Monte-Carlo simulation. The production system is a dynamic system. Hence, analysis of dynamic properties related to system equilibrium stability and robustness should also be addressed.
3. In an ideal case a fully integrated (centralized) control approach will probably provide the best possible solution since all parts of a system have complete knowledge of the entire system at all times. On the other hand a decentralized approach with minimal interaction between components may be much simpler, but at the price of lowered overall performance. Hence, it may be possible to provide some justification for choosing a certain level of integration between agents based on the characteristics of a given production systems. This hypothesis should be explored using, among other sources, the results in 2.

The thesis report may include a paper to a selected conference with the main results of this work.

Starting date:	10.01.2011
End date:	06.06.2011
Co-supervisor:	Johannes Tjønnås, SINTEF

Trondheim, 13.01.2011

Bjarne Foss
Professor/supervisor

When problems overwhelm us, and sadness smothers us, where do we find the will and the courage to continue? Well, the answer may come in the caring voice of a friend, a chance encounter with a book, or from a personal faith. For Janet help came from her faith, but it also came from a squirrel. Shortly after her divorce, Janet lost her father, then she lost her job. She had mounting money problems. But Janet not only survived, she worked her way out of despondency and now she says, life is good again. How could this happen? She told me that late one Autumn day when she was at her lowest she watched a squirrel storing up nuts for the winter, one at a time he would take them to the nest. And she thought, if that squirrel can take care of himself with the harsh winter coming along, then so can I. Once I broke my problems into small pieces I was able to carry them, just like those acorns, one at a time.

- "Little Acorns" by The White Stripes

Abstract

The literature in the field of multi-agent manufacturing control predominantly presents qualitative arguments to motivate its usage. This work presents quantitative meaning to some of these arguments. The work also discusses the control structuring process and puts it into perspective with multi-agent control. A suitable example for the analysis of multi-agent control is developed. A two layered approach for the control of this example process is proposed and it is shown that agents can be used in both of these layers. An ad-hoc analysis is made of the top layer and simulation results shows that a distributed control approach introduces a optimality gap when compared to a centralized approach. Further simulation results show that a layered approach to multi-agent control for the proposed process improves overall performance by increasing agent coordination.

To date there has been a lack of industry adoption of multi-agent technologies. The industry seems to favor control approaches that are well tested and with quantifiable evidence supporting their efficiency. The development of simulators that capture the full complexity of manufacturing processes is a time consuming task. The example process proposed in this work only captures some core dynamics. Further work on presenting more quantifiable arguments for the use of multi-agent control is thus motivated throughout the thesis.

The ideas presented in this thesis are included in the article 'MAS for manufacturing control: A layered case study' by Pedersen et al. The article is aimed for submission to the AAMAS 2012 conference in Valencia.

Acknowledgements

This thesis is the result of the work undertaken in the spring of 2011 at NTNU. Many thanks to professor Bjarne Foss at the Department of Engineering Cybernetics, first of all for accepting me as his student, and also for always taking the time for quality advisory sessions during the semester. Special thanks also goes to Johannes Tjønnås and Ingrid Schjølberg at SINTEF ICT for all their support and ideas, and for being great mentors and friends.

Further, I am grateful for the work of Martin Hanger in auditing this thesis.

Lastly, I offer my regards and blessings to family and close friends for their continuous support and encouragements during my years at NTNU. You all know who you are!

Contents

List of Figures

Abbreviations

BDI	Belief Desire Intention model
DSA	Distribution Scheduling Agent
HMS	Holonic Manufacturing System
JADE	Java Agent Development Framework
GRACE	inteGration of pRocess and quAlity Control using multi-agEnt technology
PID	Proportional Integral Derivative
PSA	Production Scheduling Agent
MAS	Multi-agent System
MPC	Model Predictive Control

Chapter 1

Introduction

1.1 Background

SINTEF ICT Applied Cybernetics is a partner in the FP 7 NMP project GRACE[1]. The main objective of this project is to conceive, study, develop, implement and validate a collaborative multi-agent system (MAS) which operates at all stages of a production line, integrating process control with quality control. The main focus is on appliances. One of the tasks of SINTEF is to develop adaptability and optimization mechanisms at different levels in various processes across the factory. At a local level process control of manufacturing and assembly resources will be considered and at a global level the system will adapt to specific events that may occur in the production environment by combining decisions of individual agents. In order to develop these mechanisms a thorough understanding of MAS is needed. Moreover, a study on how stability, robustness and optimality are achieved for the system is required. The report 'Steps towards the integration of control theory in multi-agent system' by S. Pedersen [27] is a part of this effort. This thesis is a continuation of the work undertaken in that report.

Agent-based systems is an area of research that has it roots in information technology. Put at its simplest, an agent is a computing element that is capable of flexible autonomous behavior in unpredictable domains. In particular, there has in the latest years been significant research interest for multi-agent based manufacturing systems. In the increasingly global economy, the companies experience highly fluctuating marked conditions. Coupled with the fact that products tend towards being more customized and with shorter life cycles, publications emphasize the need for a *flexible* enterprise. Papers like [4],[21],[20],[19],[18],[15],[14],[13],[12],[5] and [2] all present different multi-agent architectures for manufacturing systems that claim to fulfill the need for flexibility better than the current, more rigid hierarchical manufacturing systems. Such designs are regarded as state-of-the-art [16] and will serve as a foundation for the GRACE project.

[1]http://grace-project.org/

A note on flexibility
The word 'flexibility' will be used a lot in this thesis, creating the need to write some words abouts its intended meaning. Some control theorists reading this may scratch their heads when reading flexibility because of the two commonly used and somewhat related terms of 'robustness' and 'adaptivity'. On the other hand it seems like flexibility is very commonly found in the multi-agent literature. If, for example, a control system is influenced by a disturbance or a small perturbation in the process, robustness certainly falls under the term of flexibility. Also, if some slow varying variable changes, adaptivity can also be perceived as flexibility (notice however that adaptivity makes more assumptions on the underlying model). **However, flexibility is used in this thesis because it has a broader meaning than robustness and adaptivity in the classical control theoretical sense. It does not only entail control theoretical aspects, but also economical, software and structural aspects. The need for flexibility arises when the environment appears non-deterministic. A flexible control architecture is a system that can not only operate under the conditions that are at present, but also under tomorrow's possibly different conditions.** It can be used on multiple levels. On a local level, entities in a system can be flexible in the sense that they are robust or adaptive. On a global level, a system of entities can also be flexible in the sense that they have the ability to restructure and interact in accordance with the operating conditions. As an example, an enterprise can be said to be flexible if it has a diversified portfolio such that it can restructure to benefit from the most profitable markets at all times.

This topic will be explored in further detail in Chapter 2.

In Leitão and Restivo 1999 [17], the authors give an overview of manufacturing paradigms and describe the evolution of control architectures. Multi-agent control is proposed as a natural next step to the evolution of manufacturing control because of arguments like the need for robustness, flexibility and scalability. Further, the article suggests that MAS can be applied at multiple levels in an organization, all the way from a virtual enterprise level down to the shop floor level at the factory, thus resulting in a layered approach to multi-agent control.

Multi-agent systems are conceptually an *architecture* that puts constraints on the possible control approaches. Although to date control theorist seem to have taken little interest in multi-agent manufacturing control directly, there are clear similarities in the approach found in some of the literature on distributed control. Where as the multi-agent literature is almost exclusively focused on design, the distributed control literature is more rooted in mathematical properties like stability, optimality, robustness and adaptivity. Decomposition methods for mathematical optimization is an example of a field that has strong similarities with multi-agent systems in the sense that computational nodes may be distributed in space. Another field which touches base with multi-agent systems is distributed control of robotic networks. Bullo et al. [3] puts the analysis of such networks in the framework of graph theory where they proceed to use algebraic tools to study issues like stability and convergence.

It has long been known that decentralized control can cause arbitrarily poor performance if special care is not taken. Rawlings and Stewart 2008 [28] show that multiple optimization based controllers can cause both instability and suboptimality if the level of coordination is too low. Pedersen December 2010 [27] also highlights other issues concerning stability and optimality in decentralized decision making.

Skogestad 2000 [32] emphasizes the importance of control system *structuring* and *design*. Important questions like which variables should be controlled, which variables should be measured, which inputs should be manipulated, and which links should be made between them, is explored. Answering these questions will result in a particular control structure that serves as a basis for control system implementation.

1.2 Problem statement

The multi-agent publications mentioned in the background all give excellent qualitative arguments for the use of multi-agent systems in manufacturing. It is the purpose of this thesis to give *quantifiable* meaning to these arguments, as this seems to be somewhat missing in the literature to date. Also, many of these articles start with the presumption that a multi-agent control system is the best choice at the outset. This thesis will put multi-agent systems in relation with the *structural* decision process of control system selection. Further, it is the authors belief that a layered approach to multi-agent control can provide increased coordination, and give some degree of guaranteed performance. Thus, how agent-usage on different levels in a control system effects performance will also be explored.

SINTEF's main role in GRACE will be to offer competence on control engineering. It is then important to look at how one can incorporate control principles in the design of new manufacturing systems. At the time of this writing, there is still a clear missing link between control theory and multi-agent systems [26]. It is thus also the purpose of this paper to fill some of the gap between these two fields. This will be done by studying ideas in control theory that share common ground with multi-agent systems and through investigation of problem domains from the manufacturing industry. This will allow us to not only describe *how* the system is implemented, but also which *implications* the specific implementation gives.

On the basis of literature review and general knowledge of control theory and mathematical programming, the following hypotheses are formulated.

Hypothesis 1. *A decentralized control approach generally suffers from an optimality gap when compared to a fully centralized control approach.*

Hypothesis 2. *Choosing a decentralized control approach may be justified in cases where the system needs to exhibit a high degree of flexibility and/or if there are specific design constraints present.*

Hypothesis 3. *Pure multi-agent control systems may benefit from implementing some degree of layered control techniques to improve coordination and overall performance of the system.*

These hypotheses are fairly general, and they should only be expected to hold under certain assumptions. However, it is the purpose of this thesis to investigate these in a quantifiable manner on a suitable test case. Such results could contribute to demystifying some of the qualitative arguments given in multi-agent literature. It should be noted that developing simulators that capture the full complexity of a manufacturing plant is a time consuming task. This thesis will thus only aim to develop a limited simulator with some key properties. Hopefully this will serve as a motivation for more quantifiable research on the conditions favoring multi-agent control.

1.3 Organization of the thesis

Chapter 2 considers the plantwide structuring process. This includes how to structure a control system, in addition to highlighting the trade offs between flexibility and optimality. Chapter 3 gives the theoretical foundation associated with multi-agent systems.

Further, in Chapter 4, the product marriage process is presented as a test case for investigating the proposed hypotheses. A simulator for the process is built in JAVA. A two layered approach is suggested for the control of this process, where the top layer is optimization based.

Chapter 5 proceeds to investigate the top level of this control system. A model for the layer is developed and possible agent usage is explored. Simulation results from both agent-based and centralized optimization are presented. This Chapter is aimed at investigating Hypothesis 1. In Chapter 6, the top level controller is implemented on the process simulator along with a low level multi-agent control system. This chapter presents simulation results from different control schemes on the simulator, including a layered and a non-layered approach. This Chapter is aimed at investigating Hypothesis 2 and Hypothesis 3.

Chapter 7 gives an overall discussion of the work undertaken in this thesis. Finally, Chapter 8 gives some concluding remarks and suggestions for future work.

Note also that an article which includes the main ideas in this thesis is included in Appendix C. The article is aimed for submission to the AMMAS 2011 conference in Valencia.

Chapter 2

Plantwide control selection

There have been many articles published on different multi-agent architectures for manufacturing control. Most of the arguments they present for motivating the usage of MAS in manufacturing control falls under terms like modularity, robustness, reconfigurability and reusability. Although these articles all present valid arguments, they often start with the presumption that a MAS architecture is the right choice for manufacturing control at the outset. As such, they often do not put the MAS architecture in context with other possible approaches. This chapter aims to take a step back and conceptually look at which motivations are underlying the structuring of any control system. The chapter will also try to define some core characteristics of a control system and place the multi-agent approach in context with other possible approaches.

2.1 Control system structuring

When considering a production plant, be it a chemical or batch oriented one, it may have thousands of measurements and control loops. In the field of control theory it is often assumed that the control structure is given at the outset [32]. That is, the controlled variables, manipulated variables and control objectives are already determined. The issue of plantwide control, however, considers the underlying structure of the overall plant with emphasis on structural decisions. These structural decisions include the appropriate selection of controlled and manipulated variables, as well as a suitable decomposition of the overall problem into smaller subproblems.

It is in the realm of plantwide control that the justification for the usage of multi-agent systems is found. Multi-agent systems are architectures that implicates a decentralized control approach for plantwide control that aims to provide the system with a degree of robustness to variances. These variances can often be divided into internal variances, like rate of throughput, or external variances, like marked conditions. That is, the system should be able to function under the full range of operating conditions, internal and external, without the need for reconfiguration. Since it is a decentralized approach, a key question is thus how to divide the system for different agents to control. Morari 1983 [22] suggests that there are two principal ways of decomposing an overall system problem:

1. **Vertical decomposition** The system is separated by operating frequencies (time scales). The control approach is layered in a hierarchy from slow to fast dynamics. Slow dynamics are controlled by top-level controllers which serve as setpoints for the lower layers. The lower layers control the faster dynamics of the system. The top level control is typically optimization based. As an example a plant can be divided into scheduling (frequency: weeks), site-wide optimization (frequency: days), local optimization (frequency: hours), predictive control (frequency: minutes) and regulatory control (frequency: seconds).

2. **Horizontal decomposition:** The system is split into separate subsystems which are separately controlled by non-communicating controllers.

Skogestad 2000 [32] argues that most control system today uses both of these decompositions. The layers in the vertical decomposition are linked by the controlled variables, whereby the setpoints are computed by the upper layers and implemented by the lower layers (see Figure 2.1). As such, an important issue is the selection of these variables. Skogestad continues to give four requirements that a controlled variable should meet:

1. Its optimal value should be insensitive to disturbances.

2. It should be easy to measure and control accurately.

3. Its value should be sensitive to changes in the manipulated variables.

4. For cases with two or more controlled variables, the selected variables should not be closely correlated.

When designing a multi-agent control structure these requirements can also be beneficial to look at when selecting the control area of each agent. Other sources of inspiration for system division can be found in Alonso and Ydstie 2002 [1] where the separation principle is dividing the system into passive subsystems.

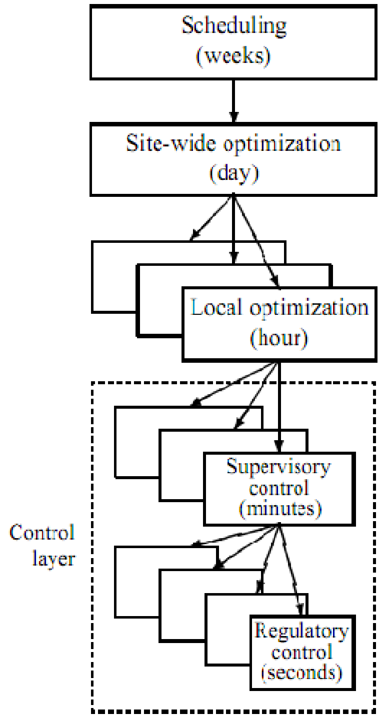


Figure 2.1: Common layered control approach [32]

In multi-agents systems proposed for manufacturing control, a conceptually similar decomposition to that in Figure 2.1 is some times used. The system is first divided into different levels based on frequency of the dynamics[1]. Further, each level is divided into different entities based on function[2]. With the term function it is meant that the particular subsystems on one level operates approximately on the same frequency but provides a different service. As an example, in a production line, each machine executes different procedures on the product. However, each machine typically operates on nearly the same frequency and thus all belong to the same level. The control of each subsystem is then assigned to individual *agents* which *communicate* to improve global performance at each level. Notice how this is different from the horizontal decomposition proposed by Morari, which is a non-communicating approach. Notice also that in the term agent (Definition 3) lies a presumption that the controllers are self contained and should be able to function on their own. That is, they do not require any information or services from other agents to function locally. The idea is shown in Figure 2.2.

[1] Although pure MAS is not a layered approach.
[2] As an example, see Leitao 1999 [17]

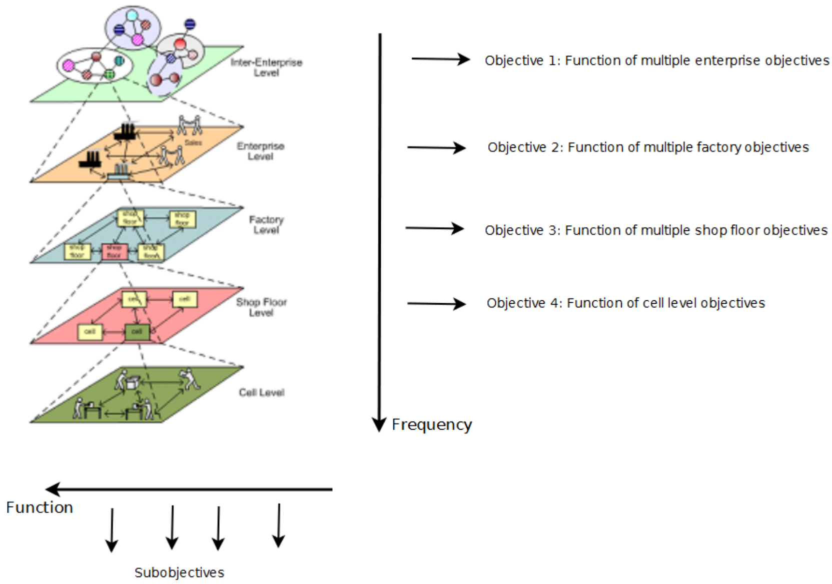


Figure 2.2: Common decomposition for a MAS control approach in a manufacturing environment

Relating to Skogestads division of plantwide control systems, multi-agent systems differ in the way that each entity and each layer typically communicate to achieve local and global goals. Thus, communication is not only downwards, but possibly also horizontally and upwards (Figure 2.4). Note also that in a MAS the communication structure is not rigid, like in Figure 2.1, but dynamic. This comes from the fact that agents should in principle be able to decide on-line which other agents it wants to communicate with (Figure 2.3). This dynamic communication structure is often facilitated by yellow-book services. Thus, some of the structural decisions are left to the MAS to decide on its own. Also note that Skogestads control structure is based on continuous processes (chemical, petrochemical etc.) while these MAS control structures are generally aimed at discrete / batch processes (cars, household appliances etc.). However, in principle both approaches may be applied in a generalized process framework.

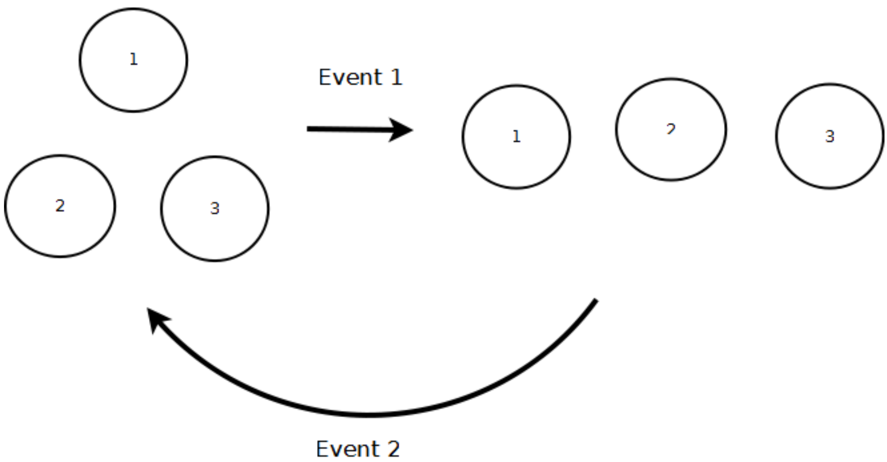


Figure 2.3: Example of restructuring in communication structure due to events.

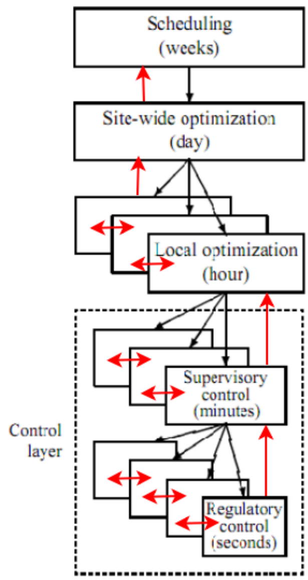


Figure 2.4: Multi-agent version of Skogestads layered approach

Putting the multi-agent system into a traditional block diagram, each level of the multi-agent system can be seen as a control allocation block mapping upper level goals to lower level goals. The idea is shown in Figure 2.5.

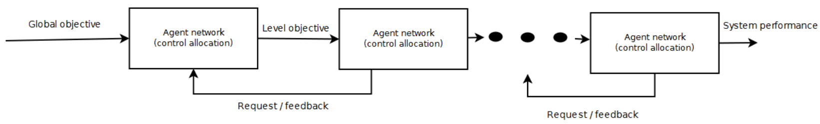


Figure 2.5: Control allocation view of a layered MAS control system

2.2 Common control approaches

2.2.1 The generalized-specialized scale

When looking at nature one will find many different strategies present in various species for survival and replication. These strategies have evolved over millions of years of evolution and are adapted to the environment in which the species live. Among the many unique strategies found, they can all be placed on a continuum between *generalist* and *specialist* species. A generalist species is able to thrive in a wide variety of environmental conditions and can make use of a variety of different resources. A specialist species can only thrive in a narrow range of environmental conditions or has a limited diet. As an example, omnivores (eats both plants and animals) are more generalists than herbivores (eats only plants) because they can eat a greater range of foods. However, the herbivores will be more effective at utilizing the energy from a plant diet because their digestive system and metabolism are specifically designed for such a diet. As a general rule, ***when environmental conditions change, generalist are able to adapt, while specialist tend to fall victim to extinction much easier*** [34]. On the other hand, a species with a highly specialized ecological niche is in general more effective in that particular niche as long as the environment does not change. A species that has too little specialization for a niche will be dominated by other, more specialist species. Thus, the degree of specialization and generalization is a balancing issue for which evolution has given different solutions. Figure 2.6 shows some species on a generalist-specialist scale relating to food consumption. Notice that the true location of any species on a more abstract generalist-specialist scale is a complex functions of many such scales. For example, one could place the same species on a generalist-specialist scale relating to intelligence, like in Figure 2.7[3], which gives a different result. The cockroach is now placed on the very end of the specialist side because its brain is highly specialist to process specific things like variations in light, patterns in pheromones from other cockroaches and such. It probably does this way better than the human brain is able to. However, most would argue that the average human brain has the ability to process a wider range of information.

In fact, this scale can be used in a wide variety of fields. For example, when looking at economic systems, marked economy would be placed further on the

[3]The placement of the panda bear and the cow should be taken with a grain of salt in this figure.

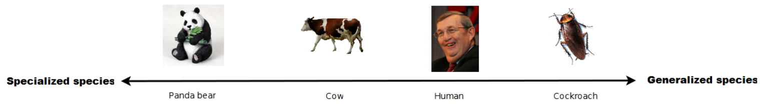


Figure 2.6: Different species on a generalist-specialist scale relating to food consumption

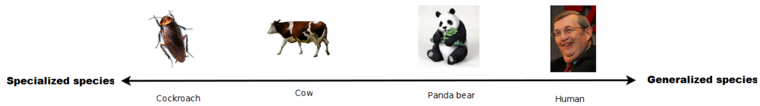


Figure 2.7: Different species on a generalist-specialist scale relating to intelligence

generalist side of the scale than planned economy. Surely, history has proved countries with a marked economy to be more versatile than others with a purely planned economy. Similarly, in politics, democracy could be placed further on the generalist side of the scale than dictatorship. Dictatorship relies heavily on the optimal functioning of one person, and if that person is disabled, the system falls down. The reader is encouraged to reflect on these scales and think of other areas in which the generalist-specialist scale can be used.

2.2.2 Classifying control approaches

As the reader should suspect, the biology-inspired introduction is meant to motivate a discussion of control systems. Just like with living organisms, control structures must implement a strategy for control in a given environment. The structure must be adapted to the plant, just as the organism must be adapted to its biotope. The author would like to argue that a control structure can also be placed on a continuum between generalized and specialized control. Thus, the following definitions will now be introduced:

Definition 1. *A **generalized control structure** is a control structure that can function on a variety of different plants. It has a high degree of flexibility but is not tailored for any specific set of conditions.*

Definition 2. *A **specialized control structure** is a control structure that is tailored for a specific set of conditions. It has a low degree of flexibility and typically only functions in one kind of plant.*

Although intentionally vague, these definitions can be used to look at some of the more common control strategies used today. The following approaches will be considered in the light of the model structure they use to calculate control signals.

- **Open-loop control:** The system control is calculated a priori with no feedback. As such, the controller is static. This approach can be regarded as highly specialized because it only works on the specific system it was design for. It is also very sensitive to model errors.

- **Closed-loop control:** The system control is calculated on-line by introducing feedback based on core process structures and properties. This approach is more generalized than open-loop control as it uses feedback to compensate for errors in the system model.
- **Adaptive control:** Some system or control parameters are estimated on-line using feedback from measurements. This allows the controller to operate in conditions where there are parameters that are not known a priori. Adaptive control does however make crucial presumptions about the model *structure*, like the rate of change of the system parameters. If, as an example, a parameter that is estimated does not change sufficiently slow this approach can lead to instability. In terms of model structure, adaptive control will be classified as more specialized.
- **Robust control:** Deals explicitly with uncertainty in its approach to controller design. Controllers designed using robust control methods tend to be able to cope with small differences between the true system and the nominal model used for design. This approach is thus a somewhat generalized with respect to model structure.
- **Stochastic control:** Deals with the presence of stochastic variables in a system model. This allows the controller to operate in conditions with random events. This control approach makes the assumption that the certainty equivalence property [4] holds. It also typically assumes the presence of Gaussian noise. Because of these assumptions made about the model, stochastic control will be classified as more specialized on model structure.
- **Artificial intelligence:** Uses various approaches like neural networks, fuzzy logic, evolutionary computation or machine learning. Typically these approaches are highly generalized with respect to model structure.

It is difficult to place these approaches definitely on a specialized-generalized scale, as each controller can be placed differently on a case dependent basis. As an example, a model based controller can have a linear or non-linear model. If the controller has a non-linear model it can be assumed that it is more generalized because the system can be linearized around new operating points. However, one can generally assume that Figure 2.8 is a good coarse representation. The reader is once again encouraged to reflect on this scale.

2.3 Choosing the right control structure

The argumentation for switching from the traditionally more rigid hierarchical control approach to multi-agent control is often the need for more flexibility in the

[4] See Van de Water 1981[35] for an introduction to the certainty equivalence property in relations to stochastic control.

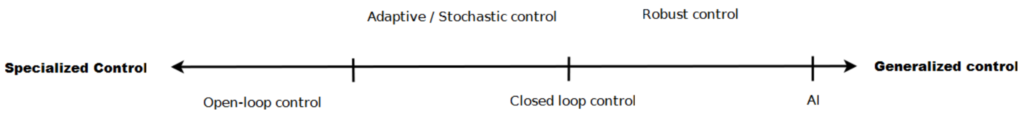


Figure 2.8: Common control techniques on a specialized-generalized scale relating to model structure

factory (as discussed earlier). What this really means is that one needs to push the control system further out on the generalized end of the scale. This is due to increased requirements for allowed variance in the system. Conceptually, one can look at the control system designer's task as to solve an optimization problem. Imagine that $x \in X$ is an instance of a stochastic state vector X. The distance from some optimal operating point x^* after convergence for the system for the instance x is defined to be $\tilde{x} := x^* - \lim_{t \to \infty} c \cdot x$ for some selection vector c. The maximum allowed variance for X is defined to be $\bar{\sigma}$. This upper limit is given by constraints in the implementation. For example, it could be calculated that the system can become unstable if the allowed variance is greater than $\bar{\sigma}$. Given two weighting coefficients $C_1, C_2 \geq 0$, the optimization for the designer can then be defined as to solve

$$\min_{\tilde{x},\bar{\sigma}} \qquad C_1 \|\tilde{x}\| - C_2 \|\bar{\sigma}\| \tag{2.3.1}$$

Subjected to

$$\tilde{x} = f(\bar{\sigma}) \tag{2.3.2}$$
$$\bar{\sigma} = g(\tilde{x}) \tag{2.3.3}$$
$$Var(X) \leq \bar{\sigma} \tag{2.3.4}$$

If the system is operating close to optimal ($\tilde{x} \approx 0$) and has a low allowance for variance ($\bar{\sigma} \approx 0$) it is natural to assume that it is a somewhat specialist system. The functions f and g represents the trade off between optimality and flexibility. It will be assumed that as a general rule, the allowed variance decreases with increased optimal performance. Thus, f and g are functions such that $\frac{df}{d\bar{\sigma}}, \frac{dg}{d\tilde{x}} \geq 0$. This follows from the belief that the control structure can be placed on a generalized-specialized scale (note however that there may be special cases where the relationship is not 100% linear, as shown by the dotted circle in Figure 2.9). The weighting coefficients will represent to which degree the system needs to be specialized and generalized.

The process of deciding upon a control system will then typically be

1. Do a system requirement analysis to specify the weighting coefficients C_1 and C_2.
2. Solve optimization problem 2.3.1 to find $\tilde{x}$ and $\bar{\sigma}$.
3. Design/choose a control system that most closely resembles the desired $\tilde{x}$ and $\bar{\sigma}$.

According to the multi-agent control literature, C_2 should be increased relative to C_1 due to, in particular, trends in marked conditions. There is then an increased requirement for the handling of more variance in the system. Because multi-agent systems typically are very flexible they are thus proposed as good solutions for some factories today.

Two simple examples will now be presented to give the reader an intuition about this selection process. Notice how human beings fit perfectly into Definition 3 and are used as analogies for software agents in these examples.

2.3.1 Example: Resource redundancy

As an example, consider x to be the total number of workers in a factory. It is calculated that at least n workers need to be operative to meet the production requirements today. The optimal operating point from an economical standpoint for this system is thus $x^* = n$ because each worker gives an additional salary expenditure. Say that the system designer chooses $C_1 = 1$ and $C_2 = 0$ because a marked analysis shows that the price of the product needs to be minimized due to competition. Optimization problem 2.3.1 then gives the solution $\tilde{x} = 0$ and $\bar{\sigma} = 0$. The designer then chooses to implement only n workers in the the factory. Imagine that this *is* feasible under todays operating conditions because all workers are in good health and can operate with full efficiency. However, the system has zero *flexibility* because if only one of the worker's efficiency is reduced, in the future the system will not be able to meet the production requirements. If instead the coefficients had been set to $C_1 = 1$ and $C_2 = 1$ this could have resulted in the implementation of $n + k$ workers instead, which would allow for more variance in the worker efficiency. Notice the direct trade off between optimality and flexibility in this example; the total revenue today is reduced as a direct function of the number of workers and the flexibility of the system increased as a direct function of the number of workers present in the factory.

2.3.2 Example: Resource specialization

Now imagine that our designer can either implement a specialized robot to carry out a task of a human worker. Let $x = [x_1, x_2]$ where x_1 is the efficiency measured in system throughput (the rate that finished products are being produced) and x_2 is the type of product currently being produced (given by an integer number). Optimality is measured in total throughput of the system, such that $x^* = \infty$. Choosing $C_1 >> C_2$ when solving 2.3.1 results in a small $\tilde{x}$ and a small $\bar{\sigma}$. The designer will then choose to implement mostly robots, because they give a higher throughput for the particular product which the marked demands *now*. However, the system will then have a low degree of flexibility because the robots will not be able to produce products that are not closely related to the one it was designed for. Choosing $C_2 >> C_1$ when solving 2.3.1 results in a larger $\tilde{x}$ and a larger $\bar{\sigma}$. This can be achieved by mostly using humans. Humans will be slower than the robots to produce one particular product than the robots, but in return they are highly flexible in the way that they can adapt and learn do produce new products. It is thus fair to say that using humans gives a more generalized control structure.

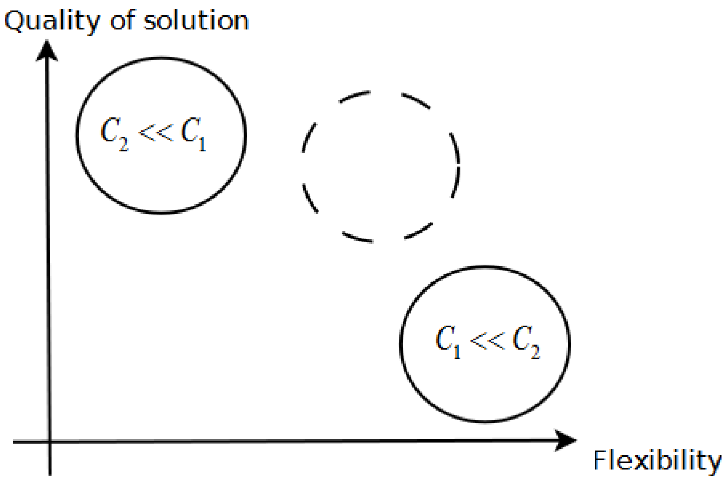


Figure 2.9: Qualitative representation of control system selection.

Remark. *It should be noted that this selection process is merely the authors own attempt to formulate control structure selection as an optimization problem which highlights the balance between flexibility and optimality. It should not be considered as any proven or tested theory.*

2.3.3 Understanding process variation

In relation to step 1 in the proposed control structure selection process, it is important to understand which types of variation is going to be present in the process. This is a vast scientific field in itself and this section will only briefly present some typical sources of variation. Doty 2006 [8] suggests that in a typical manufacturing process these can include the following:

- **Process.** This includes such things as poor workholding and positioning, machine vibration, machine looseness, hydraulic and electrical fluctuation, machine breakdown, machine wear, machine speeds, poor preventive maintenance, poor repairs, dirty machinery, poor setup, change in setup, poor fixture design etc.
- **Material.** These types of variations are caused by differences in material characteristics, such as hardness, moisture content, tensile strength, ductility, high or low concentration, mixing of different lots, change in supplier etc.
- **Operator.** This is can be a great source of variation. The personal, emotional and mental problems of the operator, along with inattentiveness and

lack of understanding, can lead to misalignment, frequent machine adjustments, improper handling etc. Other sources of operator problems are: untrained operator, operator fatigue, changes in shift and operator morale.

- **Tooling.** Tooling problems can be caused by such things as: using the wrong tool, tool made incorrectly, tool used incorrectly, tool wear etc.
- **Measurement.** Errors is measurement induced by: poor signal processing, using wrong gauge type, inaccurate gauge, poor gauge maintenance etc.
- **Procedures.** Some common errors in this area are incomplete operation and missing operation to a product.
- **Environment.** This includes temperature, moisture, air pressure, dust etc.

In addition to these, the author would like to add

- **Marked.** Rapid and unpredicted changes in marked conditions can cause variance on many levels. Examples include variance in required system throughput, type of product, quality, logistics etc.

An important task for system design is to identify which of these or other variances that will be present in the system and to which degree. Statistical process control (SPC) is the application of statistical methods to examine a process and the sources of variation in that process. Tools used include histograms, cause-and-effect diagrams, Pareto diagrams, control charts, run chart and process flow diagrams. SPC is often used when the system is in operation to ensure that it operate at its full potential to produce conforming products. However, much of these tools can also be used in the plantwide control design to identify process variation. For an excellent introduction to SPC, see Oakland 2007 [24].

Chapter 3

Theoretical foundation

This chapter will give a short introduction to multi-agent systems. Section 3.1 defines the agent and different types of agents. Section 3.2 explains networks of agent, called multi-agent systems. Finally, Section 3.3 and Section 3.4 gives two different examples of multi-agent control usage.

3.1 The intelligent autonomous agent

Multi-agent systems are systems composed of multiple interacting computing elements, known as agents. Before we explore the field of multiple agents, it is necessary to have a concept of what an agent really is. Since agent technology has sprung out of multiple academic fields, there is no unifying definition. There is however a wide agreement that an agent should be something more than a simple object. It needs to have some sort of internal system for reasoning. Wooldridge and Jennings [38] will now be used as an inspiration to form a definition which suggest four key qualities the intelligent autonomous agent should have .

Definition 3. *An agent is a software entity that should have the following four qualities:*

- ***Autonomous:*** *The agent should function without the need for external intervention.*

- ***Proactive:*** *The agent should have some goal directed behavior.*

- ***Reactive:*** *The agent should be able to perceive and react to a changing environment.*

- ***Social ability:*** *The agents should be able to interact with other agents.*

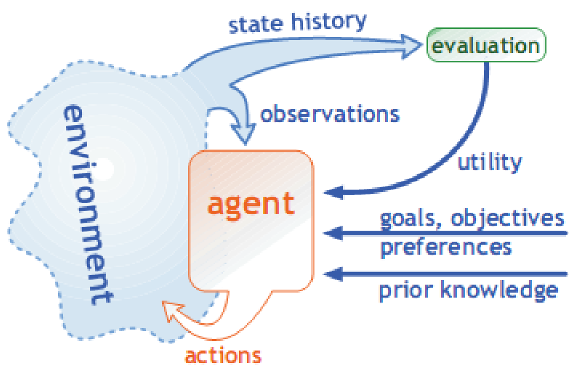


Figure 3.1: The agent and it's environment [21]

Notice how this makes an agent different from a mere object. The locus of control with respect to the decision about whether to execute an action or not is within the agent. It only executes if its internal logic decides upon it. Objects will however execute unconditionally if its methods are called. Informally one can say that objects do it for free, while agents do it because they want to.

The key problem facing an agent is thus that of deciding which of its actions it should perform in order to best satisfy its defined objectives. To solve this decision problem, several agent models and architectures have been proposed. Most of these fall under one of the following three categories [37]

- ***Reactive agents:*** Sensory input is directly linked to the action capabilities of the agent. The agent is autonomous and chooses on its own - even if the choice is hardwired - how to react to a specific situation. However, it is difficult to implement pro-activeness and goal-directed behavior.

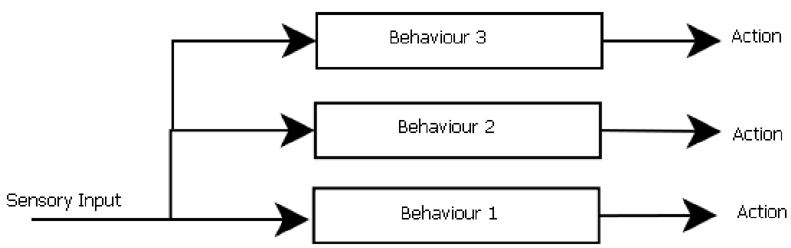


Figure 3.2: Reactive agent

- ***Deliberative Agent:*** Explicitly represents goals and form plans about how the agent wants to behave in the future in order to achieve its goals. The belief-desire-intention (BDI)[1] model falls in to this category.

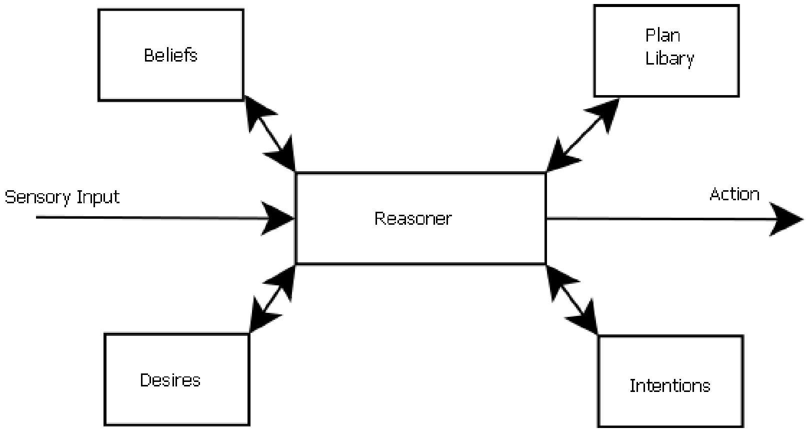


Figure 3.3: Delibarative agent

- ***Hybrid Agent:*** Incorporate both reactive and deliberative mechanisms into one architecture. Typically achieved by introducing a layer for each mechanism. Sensory input is provided to the behavior layer, and passed further up if no rule in the layer triggers.

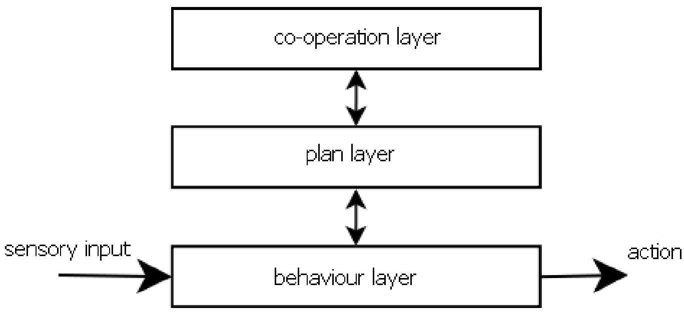


Figure 3.4: Hybrid agent

[1]See [37] for an introduction to the BDI model.

3.2 Multi-agent systems

Multi-agent systems (MAS) occur when multiple agents interact in an environment. Each agent pursues its own local goals, but they also work together solving global problems. Panait and Mason [25] argue that for true MAS the environment should also be constrained in some way; such that each agent does not have absolute knowledge of the world that the other agents know. If this was not the case there would be no need to communicate information and the system could in reality be called a single agent system.

In order to communicate, the agents need to understand each other. Thus the need for a common message structure arises. The Foundation for Intelligent Physical Agents (FIPA) is an organization that promotes such standards in the academia and industry [11][10]. The communication can also be indirect, in which an action taken by one agent affects the environment in such a way that other agents senses it and makes decisions based on that change. An example of such communication is seen in ant colonies where ants communicate by pheromones left in the environment [7].

MAS are either characterized as heterogeneous or homogeneous. In homogeneous multi-agent systems all the agents are similar. Also in heterogeneous systems there are often subgroups of homogeneous agents. It should be noted however that an agent is a software entity, such that the actual physical devices the agents are associated with need not be alike. Holonic manufacturing systems (HMS), on the other hand, divide the whole manufacturing process into pieces. Each piece is called a holon, and it includes both software and hardware.[2] For an introduction to holonic manufacturing systems, see [2].

Further, the agents can be divided into several possible organizational forms. The individual agent is given a role in the organization. Each agent can of course also be a member of multiple organizations. It is important to notice, however, that on an authoritative level the organization is completely flat. That is, there is no agent that has control over another. One agent can merely suggest to another agent to perform something. The organization principle is instead one of communication, responsibilities and coordination.

[2]The term agent will however be used for all purposes throughout this thesis

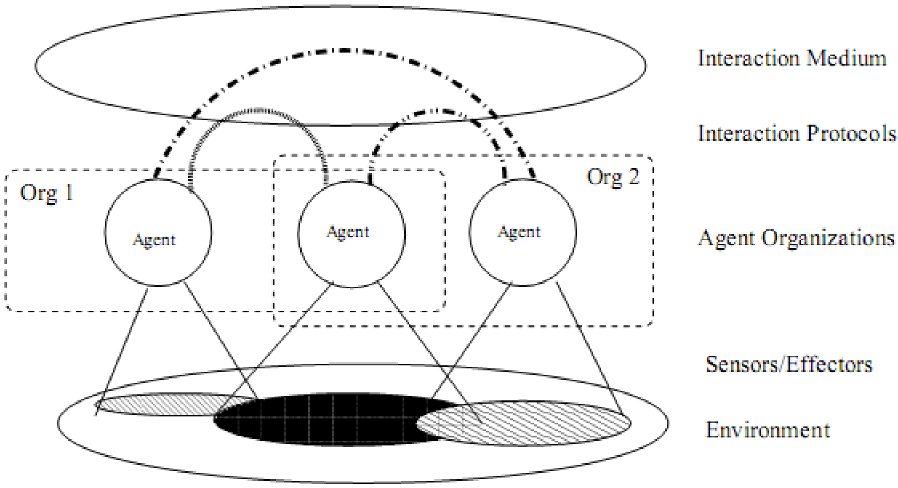


Figure 3.5: Layers of a multi-agent system [39]

3.3 Example: Cooperating autonomous vehicles

In particular, the field of homogeneous agents are a very hot topic among control engineers. Consider now the problem of unmanned planes flying in formation. On one hand, you want the airplanes to keep a specific V-formation. On the other hand, you also want to minimize fuel consumption and keep the constraints of the individual airplanes. Further, each airplane can only communicate with its closest neighbors. To solve such an optimization problem, one could decompose the global problem (to keep the formation) and make each agent (airplane) try to keep a constant distance to its neighbors, while also satisfying its local goal of keeping fuel costs low. The objective function to be optimized by the individual agent would then be a sum of the local fuel consumption cost function and the specific decomposed part of the global problem of its neighbors and itself. For an example of an algorithm that solves such a problem and at the same time compares it with a centralized method, see Shi and Hou 2004 [31].

3.4 Example: Manufacturing Application

Although there has been much research on the application of multi-agent systems in manufacturing systems, little has been actually implemented in the industry. There exist some examples, however. Probably the first full-scale industrial agent-based production system was the cylinder head manufacturing system Production

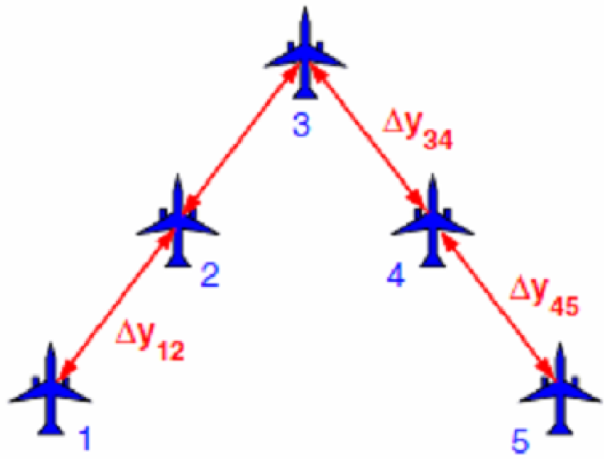


Figure 3.6: Formation flight

2000+ (P2000+) of DaimlerChrysler which was implemented in 1999.[3] This system consisted of flexible machines which were configured to process a range of products. To achieve robustness, each operation of the manufacturing system is provided by at least two machines. In case of machine failure, there is thus at least one other machine able to process the work-pieces. The system has a flexible transportation system that can move a work-piece from any machine to any other machine. DaimlerChrysler had designed and implemented an agent-based control system consisting of agents for each machine, each transportation switch and each work-piece. These agents interact in order to achieve a robust and flexible material flow through the manufacturing system. The machine agent controls the workload of the machine and bids for suitable work-pieces. The work-piece agent in turn chooses the best machine for the next operation based on processing and workload criteria. Finally, the transportation agent chooses a route to the next machine taking into account the current load of the transportation system.

The system proved to be extremely robust against disturbances on machines, as well as failures of control units. When compared with the maximal throughput achievable in theory, the performance turned out to be nearly optimal. For more details on the P2000+ system, see [4].

[3]The old control system was five years later reinstated due to problems in maintaining the P2000+ system when the designer, Stefan Bussman, left the company. This should be a valuable lesson for the standardization of agent-based control systems.

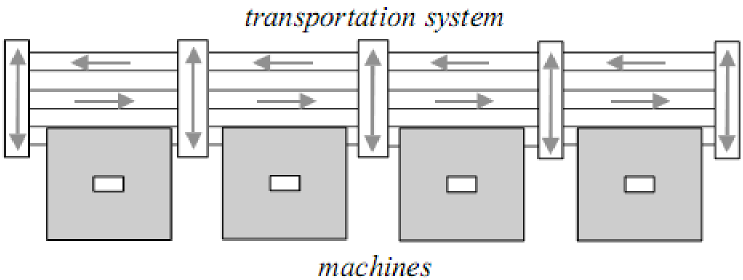


Figure 3.7: Production layout of P2000+ [4]

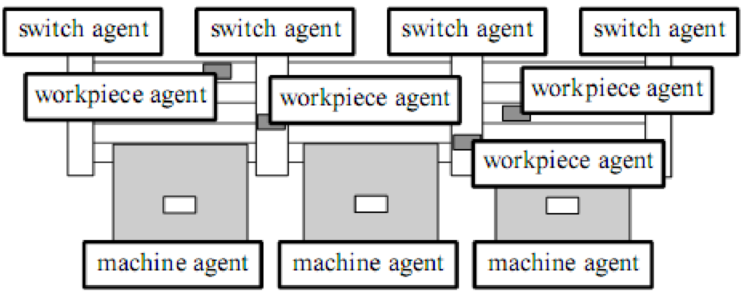


Figure 3.8: Conrol agents of P2000+ [4]

Chapter 4

The product marriage process

The product marriage process is very commonly found in nearly all manufacturing industries. At its simplest, it is an intersection in a production line where some sub-products, most typically two, are married into a more complex product. As an example, in the production of washing machines, at one point in the production process a drum is inserted into the washing machine tub. A multi-agent control system for a generic product marriage process was analyzed and implemented into a JADE simulator in Pedersen December 2010 [27]. This implementation will be briefly summarized in Section 4.1. Further, an extension for the process will be presented in Section 4.2. Problems related to this new, extended process will be investigated for the remainder of this thesis.

4.1 The product marriage process

Several parts are produced by separate machines and married into one final product. The production line is terminated by a storage space with a output given by consumer demand. Each machine in the system has a given speed constraint which change in a stochastic manner. The final product has an associated quality parameter. If the quality is out of a given bound, the product is deemed as trash. The process is shown in Figure 4.1.

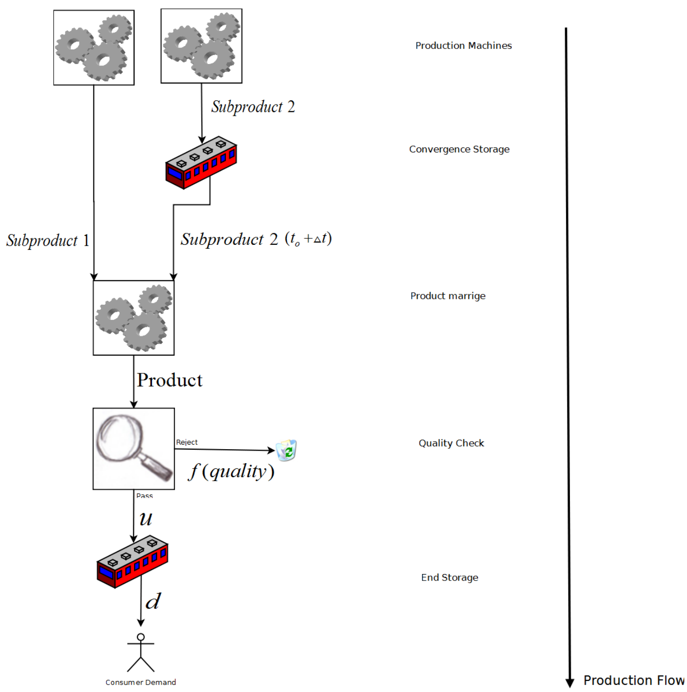


Figure 4.1: Product marriage process

For the control system, there are four global objectives:

1. **Satisfy consumer demand.** This is done by using a piecewise linear controller with feed forward from the consumer demand to keep a set point on the system end storage. The idea is that by controlling the end storage level this objective is indirectly made near-optimal because the system has a buffer in the event of reduced system performance (driven by stochastic events). This control strategy is an example of self-optimizing control as proposed by Skogestad [33].

2. **Avoid bottlenecks.** This is done by the local interaction between agents which aims to synchronize the speed of the machines to the lowest speed constraint.

3. **Optimize the worst bottleneck.** This is done by self-organization of some system resources. The resources is distributed in such a way that the worst constraint of all the machines is as high as possible.

4. **Minimize trash.** The product has a quality parameter attached to it which is given to be a function of the difference between two subproduct parameters.

> If the quality parameter exceeds some threshold the product is deemed as trash. The trash is reduced by introducing an adaptive controller which estimates the quality function and communicates this to other agents which can form control strategies based on the information.

The control structure is shown in Figure 4.2 and the three different flows of the system in Figure 4.3. The JADE implementation of the simulator is shown i Appendix A. For more information on the process, please refer to Pedersen August 2010 [26] and Pedersen December 2010 [27]

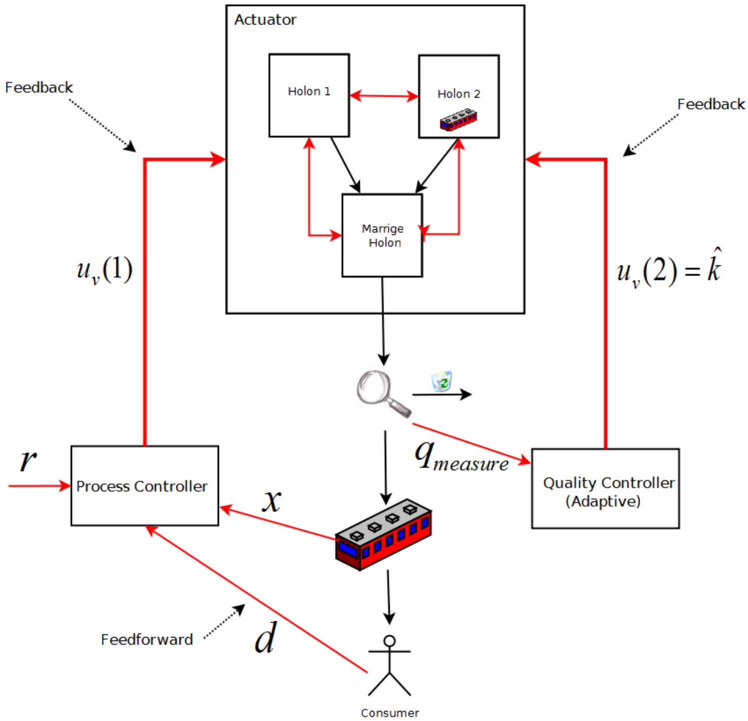


Figure 4.2: Control structure

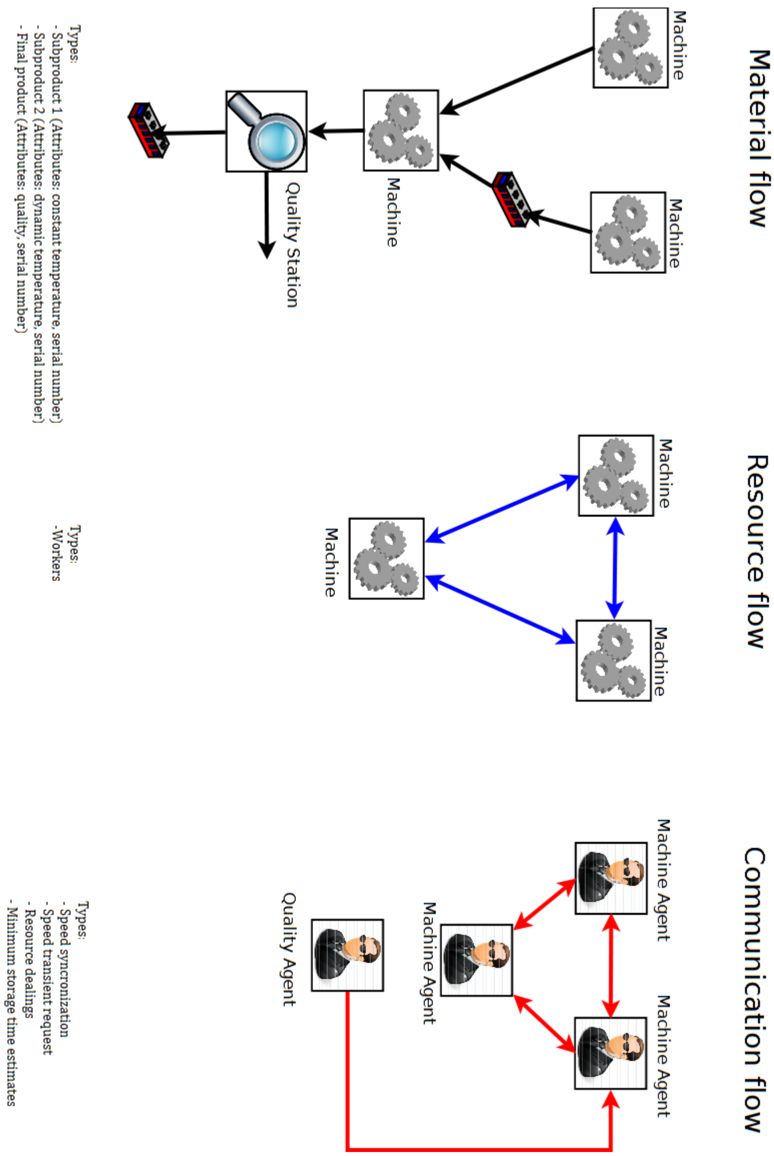


Figure 4.3: Flows of the process

4.2 Extending the product marriage process

An extension to the system is now proposed. Its main purpose is to investigate Hypothesis 1 and Hypothesis 2. The extension is thus made with the following considerations in mind:

- The system should build upon the system developed in the project report [27]. This is important because of simulator availability.
- The system should facilitate the comparison of different agent-based control approaches with varying degrees of integration between the agents.
- The system should facilitate the investigation of agents with different levels of responsiveness and local knowledge. As an example: higher level agents could make coarse plans which can be made more robust with the use of lower level responsive agents that can make local adjustments.

From these considerations the following extensions are proposed

1. Two parallel production lines. This will add redundancy, and as a consequence, more flexibility to the system.
2. The production lines can now produce not one, but four closely related products p_1, p_2, p_3 and p_4. It is possible to switch between the production of products on each line at a fixed setup cost μ.
3. The products are to be distributed to m retailers $R \in \{1, \ldots, m\}$ in batches. Each batch has a fixed capacity C. Each retailer has demand for one batch C in each distribution cycle which is a combination of the products.
4. Distribution to all the retailers must be completed in a distribution cycle $S \in \{1, \ldots, n\}$ where $n \geq m$. At each period s at most one retailer can be served.
5. Both lines terminate in a switch between four end storages; one for each product. These storages all have outputs to the current retailer batch. There is an associated holding cost for each product kept on storage each period, denoted by h_p

The extension is shown in Figure 4.4. To control this system a three layered control approach will be used, as shown in Figure 4.5. The blue arrows indicate control signals downwards in the pyramid. Red arrows indicate feedback upwards in the layers. Notice that the red arrow between the system control layer and the scheduling layer is not solid. This is because feedback between these layers will not be implemented in this thesis, although it will be discussed.

The top layer will have an economic goal oriented control approach, while the system control layer will have more production oriented goals. To handle the scheduling problem of the system, a top layer scheduler will be explored and implemented in Chapter 5. This includes the comparison between different agent-based and centralized scheduling approaches. This investigation primarily aims to investigate Hypothesis 1. Further, the scheduler will be integrated with the extended JAVA-based[1] product marriage simulator in Chapter 6. This chapter will seek to investigate Hypothesis 2 and Hypothesis 3.

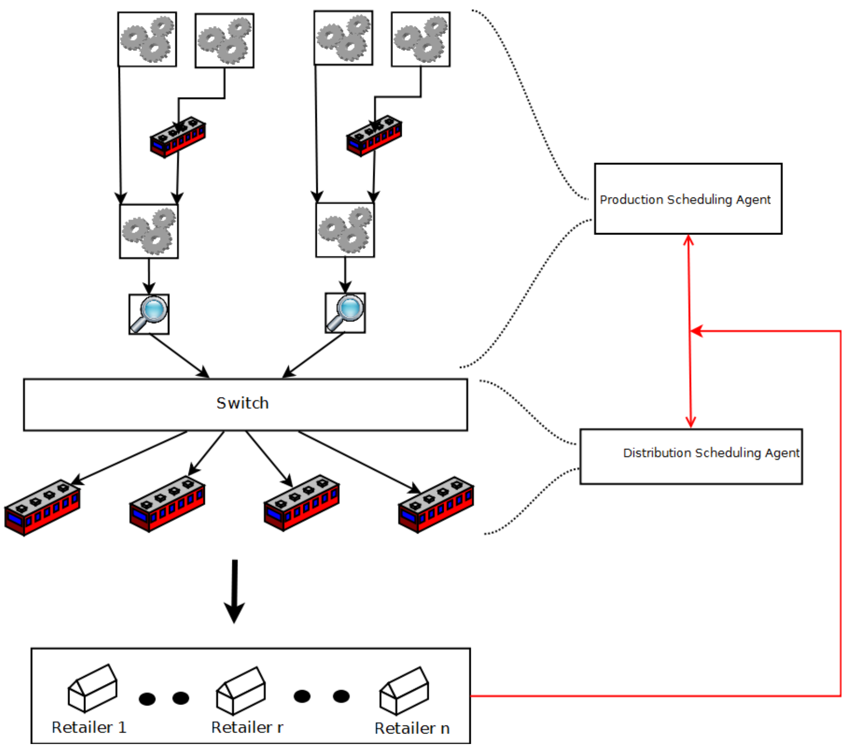


Figure 4.4: Extension of the product marriage process. Black arrows indicate material flow, red arrows indicate communication

[1]Where agents are implemented in JADE, a JAVA library.

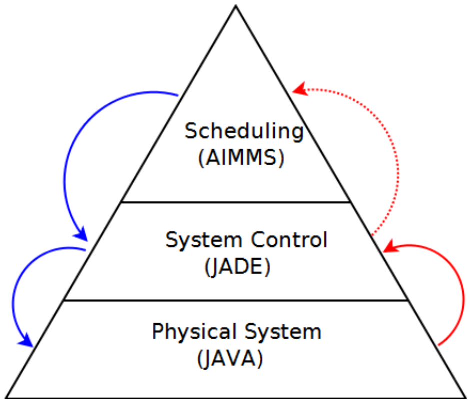


Figure 4.5: Control approach for the extended system

Chapter 5

The scheduling problem

On an abstract level there can be identified two main parts of the extended product marriage process; the production station and the distribution station. The new control problem on this level will be to make a production schedule in each cycle to satisfy all costumer demands, but at the same time minimize the production cost. The holding cost should also be minimized. In a non-integrated case, the first one is exclusively managed by the production station while the latter is managed by the distribution station. The idea is illustrated in Figure 5.1

It should become clear to the reader that this extension gives the possibility for a new level of agent usage. The production station can be assigned to one agent, while the distribution station can be assigned to another. Thus, the ***Production Scheduling Agent (PSA)*** and the ***Distribution Scheduling Agent (DSA)*** are now defined.

In this Chapter, different levels of integration between the production system and the distribution system are investigated, as literature review [6] gives indications of possible performance benefits by introducing cooperation between the production scheduler and the distribution scheduler. Thus, Hypothesis 1 is in focus.

This study also gives creates possibility to integrate the scheduling agents with the simulator already in place. The lower level machine agents present in the simulator can then be reprogrammed to follow the coarse scheduling plans made by the scheduling agents. Stochastic events can then be introduced into the system which may give rise to the need for local adjustment to the schedules. Simulations can be done on the simulator with different real time strategies for the machine agents to deal with these stochastic events. This will be explored in Chapter 6.

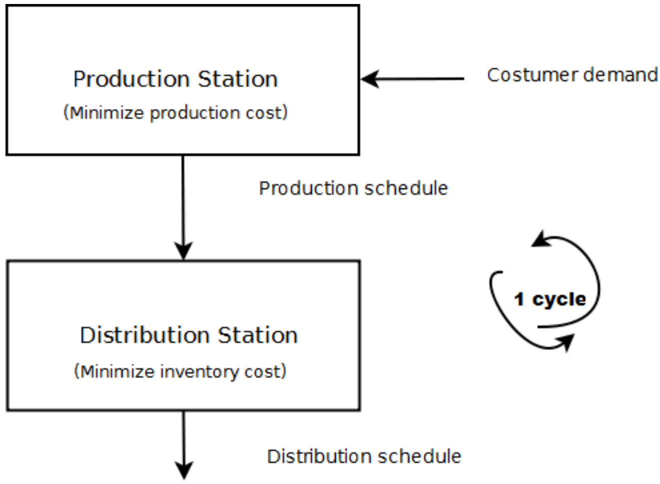


Figure 5.1: Scheduling problem with no integration between agents

5.1 Formulating the optimization problem

$S(\sigma)$ is defined as as the total production cost for the producer for a schedule σ. Further, the total holding cost for the distributor is defined as $T(\sigma, \nu)$ for a distribution schedule ν. The overall goal for the system is to minimize a convex combination of the costs $\alpha T + (1-\alpha)S$ for some $0 \leq \alpha \leq 1$. However, the PSA and the DSA are thought of as separate entities, and as a result, it may not be possible or even wanted in some cases to implement a centralized optimization model. Thus, this chapter seek to explore agent based (decentralized) optimization approaches and the possible overall performance loss they imply.

Three different cases can be distinguished for this problem; producer domination, distributor domination and cooperation. In the case of producer domination the PSA finds a production schedule that minimizes the production cost without any thought of the DSA. This implies that the cost function for the producer agent is defined as $S(\sigma)$ and the cost function for the distributor is $T(\sigma, \nu)$ for the given production schedule σ. If the distributor is dominant it minimizes the same cost function $T(\sigma, \nu)$ for both σ and ν. The given production schedule is then forced upon the PSA. Thus, the PSA does no optimization in this case. Finally, in the cooperative case both agents have some level of cooperation to reduce the total cost. In fully integrated (centralized) case can be thought of as full cooperation. In this case both agents has complete knowledge of the system and try to minimize $\alpha T + (1-\alpha)S$ directly. All three cases will now be explored further.

Section 5.1.1 defines the notation used. In Section 5.1.2 producer domination and distributor domination approaches are investigated. Section 5.1.3 discusses the effects of the initial inventory level on the problem. Finally, in Section 5.1.4, cooperative approaches are discussed.

5.1.1 Notation

The notation used in the formulation of the optimization problem is summarized in tables 5.1 - 5.4. The tables are split to show variables, sets, indexed parameters and other (miscellaneous) parameters which can not be indexed. The time scale of the problem is shown in Figure 5.2.

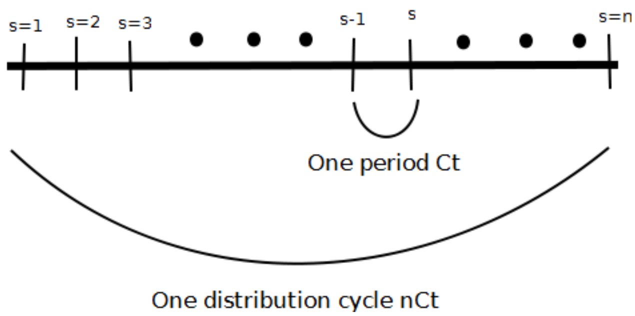


Figure 5.2: Time scales of the problem

Table 5.1: Miscellaneous list

Parameter	**Symbol**	
Number of retailers	m	
Number of periods	n	$n \geq m$
Number of lines	l	
Batch capacity	C	
Unit production time	t	
Batch period	Ct	
Distribution cycle	nCt	
Production Schedule	σ	
Distribution Schedule	ν	

Table 5.2: Set list

Set	**Symbol**	**Index**	**Set range**
Products	P	p	$\{1, 2, 3, 4\}$
Production lines	L	l	$\{1, 2\}$
Retailers	R	r	$\{1, \ldots, m\}$
Periods	S	s	$\{1, \ldots, n\}$

Table 5.3: Parameter List

Parameter	**Symbol**	**Index Domain**
Setup cost	μ	Constant
Holding cost	h	p
Demand	d	(p,r)

Table 5.4: Variable List

Variable	**Symbol**	**Index Domain**	**Value range**
Inventory level	I	(p,s)	$I \in \mathbb{Z}_{\geq 0}$
Distribution variable	x	(r,s)	$x \in \{0,1\}$
Production variable	y	(l,p,s)	$y \in \{0,1\}$
Setup variable	g	(l,s)	$g \in \{0,1\}$

5.1.2 No cooperation

In the case of no cooperation the agents has a greedy approach to solving their scheduling problem. Two non-cooperative cases are explored in this section; producer domination in section 5.1.2.1 and distributor domination in section 5.1.2.2

5.1.2.1 Producer domination

If the producer agent is the dominant part, it creates a schedule that is locally optimal for itself which it imposes on DSA. The production cost is dependent on the sum of all setup costs. The setup variable $g_{l,p,s} \in \{0,1\}$ is defined as having the value 1 if a setup is required for product p on line l in period s, and zero otherwise. It is assumed a setup cost is due on both stopping production of a product and starting up production of a new one. All setups (stopping and starting) have a fixed setup cost μ. The total setup cost then becomes $\mu \sum_{l=1}^{2} \sum_{p=1}^{4} \sum_{s=1}^{n} g_{l,p,s}$. Further, the producer must produce at least $k_p := \sum_{r=1}^{m} d_{p,r}$ of product p to satisfy consumer demand. The production variable $y_{l,p,s}$ is defined to have the value 1 if product p is being produced on line l in period s, and zero otherwise. Thus, the following optimization problem for the producer agent is formulated.

$$\underset{y_{l,p,s}}{minimize} \qquad \mu \sum_{l=1}^{2} \sum_{p=1}^{4} \sum_{s=1}^{n} g_{l,p,s} + \Upsilon \|w\| \tag{5.1.1}$$

Subjected to

$$C\sum_{l=1}^{2}\sum_{s=1}^{n} y_{l,p,s} \geq k_p - w \qquad \forall p \in \{1,2,3,4\} \tag{5.1.2}$$

$$\sum_{p=1}^{4} y_{l,p,s} \leq 1 \qquad \forall s \in \{1\ldots,n\} \quad \forall l \in \{1,2,\} \tag{5.1.3}$$

$$g_{1,p,s} \geq y_{1,p,s} - y_{1,p,s-1} \qquad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\ldots,n\} \tag{5.1.4}$$

$$g_{1,p,s} \geq y_{1,p,s-1} - y_{1,p,s} \qquad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\ldots,n\} \tag{5.1.5}$$

$$g_{2,p,s} \geq y_{2,p,s} - y_{2,p,s-1} \qquad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\ldots,n\} \tag{5.1.6}$$

$$g_{2,p,s} \geq y_{2,p,s-1} - y_{2,p,s} \qquad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\ldots,n\} \tag{5.1.7}$$

$$y_{l,p,s} \in \{0,1\} \tag{5.1.8}$$

$$g_{l,p,s} \in \{0,1\} \tag{5.1.9}$$

$$w \in \mathbb{R}^4 \tag{5.1.10}$$

where w is a slack variable and Υ is a weighting parameter.

Constraint 5.1.2 is to ensure that enough of product p is being produced to satisfy demand. Note that a slack variable w has been added to the objective function to ensure feasibility. This is due to the fact that the demand may be to large and in these cases it may not be feasible to refill the inventory. Constraint 5.1.3 is to ensure that only one product is being produced on each line at each period. Constraints 5.1.4 - 5.1.7 is to assign the right value to the setup variable.

After a optimal producer schedule σ is created, it is passed to the DSA. It must then create a distribution schedule ν which minimizes its total holding cost $T(\sigma, \nu(\sigma))$ given the production schedule σ.

To determine a optimization problem for DSA, $I_{p,s}$ is now defined as the inventory level of product p at the end of period s. For cost calculation the average inventory level for each period $(I_{p,s} + I_{p,s-1})/2$ is used. The total holding cost for one distribution cycle is then $\frac{1}{2}\sum_{p=1}^{4} h_p(I_{p,0} + 2\sum_{s=1}^{n-1} I_{p,s} + I_{p,n})$. It is further assumed that the system always controls the inventory level at the end of a distribution cycle approximately back to the initial inventory level, such that $I_{p,0} \approx I_{p,n}$. This is ensured by constraint 5.1.2. The total holding cost can then be written as $\sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s}$. The distribution variable $x_{r,s}$ is defined to have the value 1 if retailer r is being serviced at the end of period s. Thus, the optimization problem for the DSA can be formulated as

$$\underset{x_{r,s}}{minimize} \qquad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{5.1.11}$$

Subjected to

$$\sum_{r=1}^{m} x_{r,s} \leq 1 \qquad \forall s \in \{1,\dots,n\} \tag{5.1.12}$$
$$\sum_{s=1}^{n} x_{r,s} = 1 \qquad \forall r \in \{1,\dots,m\} \tag{5.1.13}$$
$$I_{1,s} = I_{1,s-1} + C\sum_{l=1}^{2} y_{l,1,s} - \sum_{r=1}^{m} x_{r,s}d_{1,r} \qquad \forall s \in \{1,\dots,n\} \tag{5.1.14}$$
$$I_{2,s} = I_{2,s-1} + C\sum_{l=1}^{2} y_{l,2,s} - \sum_{r=1}^{m} x_{r,s}d_{2,r} \qquad \forall s \in \{1,\dots,n\} \tag{5.1.15}$$
$$I_{3,s} = I_{3,s-1} + C\sum_{l=1}^{2} y_{l,3,s} - \sum_{r=1}^{m} x_{r,s}d_{3,r} \qquad \forall s \in \{1,\dots,n\} \tag{5.1.16}$$
$$I_{4,s} = I_{4,s-1} + C\sum_{l=1}^{2} y_{l,4,s} - \sum_{r=1}^{m} x_{r,s}d_{4,r} \qquad \forall s \in \{1,\dots,n\} \tag{5.1.17}$$
$$I_{1,s} \geq 0 \qquad \forall s \in \{1,\dots,n\} \tag{5.1.18}$$
$$I_{2,s} \geq 0 \qquad \forall s \in \{1,\dots,n\} \tag{5.1.19}$$
$$I_{3,s} \geq 0 \qquad \forall s \in \{1,\dots,n\} \tag{5.1.20}$$
$$I_{4,s} \geq 0 \qquad \forall s \in \{1,\dots,n\} \tag{5.1.21}$$
$$x_{r,s} \in \{0,1\} \tag{5.1.22}$$

Constraint 5.1.12 is to ensure that no more than one retailer is being serviced at the end of each period. Constraint 5.1.12 is to ensure that each retailer is served once. Constraints 5.1.14 - 5.1.17 are to ensure that the inventory level at each period is being updated correctly. Constraints 5.1.18 - 5.1.21 are to ensure that the inventory level is always positive.

Note that the initial inventory level may need to be a non-zero number to make the optimization problem feasible.

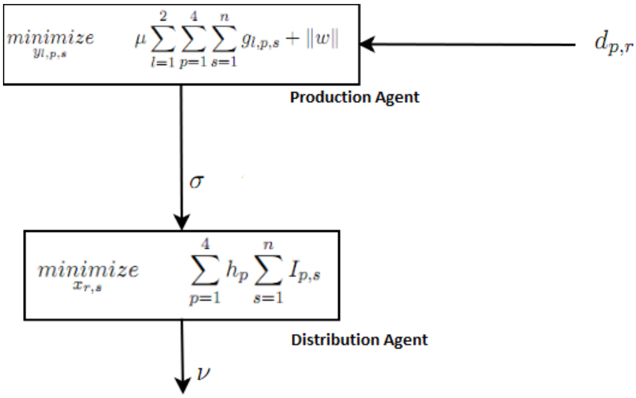


Figure 5.3: Scheduling with producer domination

5.1.2.2 Distributor domination

If the distributor is the dominant part it can impose a limit I^* on the total holding cost resulting from the producer agents schedule. In this case the PSA solves

$$\underset{y_{l,p,s}}{\mathit{minimize}} \qquad \mu \sum_{l=1}^{2} \sum_{p=1}^{4} \sum_{s=1}^{n} g_{l,p,s} + \Upsilon \|w\| \tag{5.1.23}$$

Subjected to

$$\sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \leq I^*$$
$$\mathit{Constraints} \quad 5.1.2 - 5.1.7$$
$$\mathit{Constraints} \quad 5.1.12 - 5.1.22$$

The distributor then solves optimization problem 5.1.11 subjected to 5.1.12 - 5.1.21. The communication flow of the problem is shown in Figure 5.4. Notice that the PSA in this case needs full knowledge of all parameters and variables in *both* subproblems. For this reason, a different approach will now be discussed.

Another possible approach for distributor domination is to let DSA solve its objective function with respect to $x_{r,s}$ and a new variable $y^*_{p,s}$. This is the total production from both lines, thus abstracting some of the details in the production scheduling problem. The DSA solves

$$\underset{x_{r,s},y^*_{p,s}}{\mathit{minimize}} \qquad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{5.1.24}$$

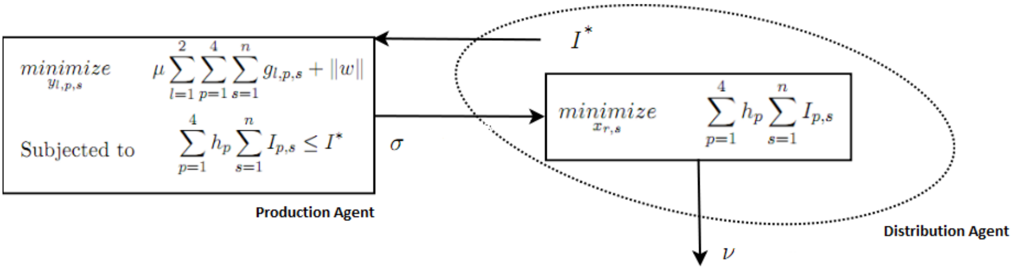


Figure 5.4: Scheduling with distributor domination, case 1

Subjected to

$$
\begin{aligned}
C \sum_{s=1}^{n} y^*_{p,s} \geq k_p - w \qquad & \forall p \in \{1,2,3,4\} \\
y^*_{p,s} \in \mathbb{R} & \\
\mathit{Constraints} \quad 5.1.12 - 5.1.22 &
\end{aligned}
$$

The optimal value of $y^*_{p,s}$ is then passed to the PSA which solves

$$
\underset{y_{l,p,s}}{\mathit{minimize}} \qquad \mu \sum_{l=1}^{2} \sum_{p=1}^{4} \sum_{s=1}^{n} g_{l,p,s} + \Upsilon \|w\| + \Omega \left\| y^*_{p,s} - \sum_{l=1}^{2} y_{l,p,s} \right\| \tag{5.1.25}
$$

Subjected to

$$
\mathit{Constraints} \quad 5.1.2 - 5.1.10
$$

where Ω is a weighting coefficient for the deviation from $y^*_{p,s}$. The actual production plan $y_{l,p,s}$ is then passed to the DSA which does another optimization if $y^*_{p,s} \neq \sum_{l=1}^{2} y_{l,p,s}$. The communication flow is shown in Figure 5.5

5.1.3 The effects of the initial inventory level

Given a set of products P_1, P_2, P_3, P_4 the optimal production schedule for the PSA is any sequence that produces all the product in the largest possible batches. As an example both $\{P_1, \ldots, P_1, P_2, \ldots, P_2, P_3, \ldots, P_3, P_4, \ldots, P_4\}$ and $\{P_4, \ldots, P_4, P_3, \ldots, P_3, P_2, \ldots, P_2,$ are optimal sequences for the PSA. The following lemma is thus formulated

Lemma 1. *A optimal production schedule for the PSA is any schedule that requires only 4 changes in production.*

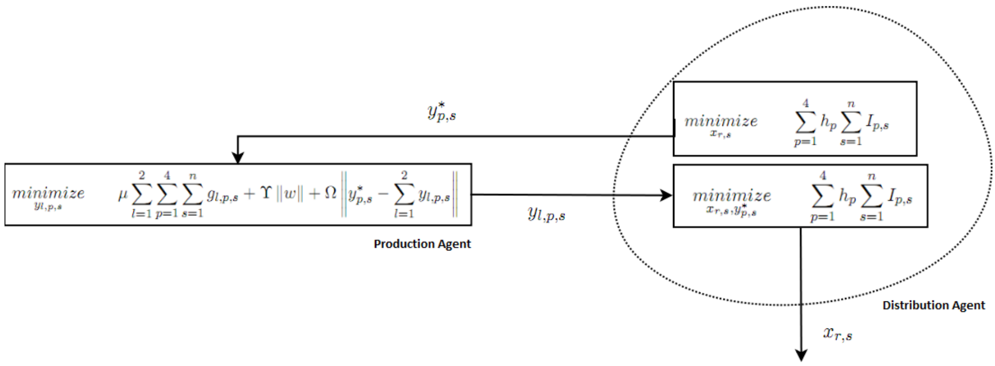


Figure 5.5: Scheduling with distributor domination, case 2

The lemma is stated without proof as the production cost is just the sum of all setups.

Given a production schedule the DSA wants to empty as much of the most expensive products to hold as quickly as possible. The following lemma is now formulated

Lemma 2. *Given a initial inventory level* $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ *and a fixed production sequence* σ*. The scheduling agent will at any instant* s *choose to serve the retailer* r *with the highest value of the sum* $\sum_{p=1}^{4} h_p * d_{p,r}$

Proof. The total cost for holding a order for retailer r is $\sum_{s=1}^{k_r}\sum_{p=1}^{4} h_p d_{p,r}$ where k_r is the period that the retailer is served. If the initial inventory level is $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ any distribution schedule is feasible because the initial inventory is large enough to serve all orders. Given a fixed production schedule the holding cost is the sum of holding all the orders plus the cost impended by the production, that is $\sum_{r=1}^{m}\sum_{s=1}^{k_r}\sum_{p=1}^{4} h_p d_{p,r} + T(\sigma)$. Since the two terms do not have any coupling the expression is minimized by minimizing both terms independently. Clearly the sum is minimized by making k_r smallest for the most expensive orders to hold, i.e. the largest $\sum_{p=1}^{4} h_p d_{p,r}$. □

If the DSA is dominant it will choose a production schedule that minimizes the holding cost. In the case of a large initial inventory level the following lemma is formulated

Lemma 3. *Given a initial inventory level* $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ *and the product holding costs* $h_1 \leq h_2 \leq h_3 \leq h_4$*. The optimal production schedule for the DSA is then* $\{P_1, \ldots, P_1, P_2, \ldots, P_2, P_3, \ldots, P_3, P_4, \ldots, P_4\}$*.*

Proof. If the initial inventory level is $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ any order that is produced is kept on storage until the end of the distribution cycle, i.e up to and including period n. This is true for any distribution schedule ν. The accumulated holding cost for holding a batch of product p produced in period k_p is $\sum_{k_p}^{n} Ch_p$. The total holding cost impended by the production is thus $\sum_{\forall p \in \sigma} \sum_{k_p}^{n} Ch_p$ where the notation $\forall p \in \sigma$ means for all the products in the production schedule. Clearly the sum is minimized by making k_p largest for the most expensive products to hold, i.e the largest h_p. □

From lemmas 1 - 3 the following theorems can be now be formulated.

Theorem 1. *When* $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ *the effect of* σ *and* ν *on the holding cost is fully decoupled.*

Proof. Follows directly from Lemma 2 and Lemma 3 □

Theorem 2. *When* $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ *the optimal* σ *for the DSA is in the set of optimal solutions for the PSA. That is, a optimal production schedule for the DSA is also a optimal production schedule for the PSA, but not necessarily the other way.*

Proof. Follows directly from Lemma 1 and Lemma 3 □

It can thus be concluded that when the initial inventory gets sufficiently large the problems decouple. In fact, experience with simulations show that the coupling between the problems generally decreases as the initial inventory is increased. This is because the dependency of the DSA on the production schedule to be able to serve the orders it wants to minimize its holding cost is decreased. The concept is shown in Figure 5.6.

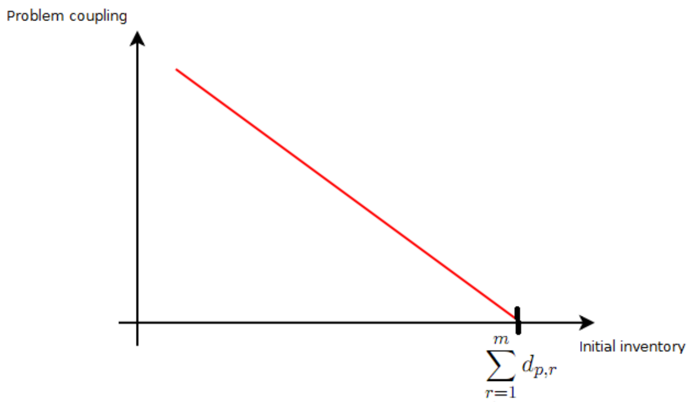


Figure 5.6: Sketch of the problem coupling as a function of the initial inventory

5.1.4 Cooperative approaches

In the case of a large initial inventory level ($I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$) Theorem 2 can be referred to. In this case full cooperation can be achieved by simply making the DSA share its holding costs. The PSA can then choose that of its optimal production schedules which produces the cheapest products to store first.

However, for the most part the initial inventory can not be assumed to be that large. In the general case a production- and distribution- schedule that globally optimizes the total cost can be found by solving

$$\underset{y_{l,p,s},x_{r,s}}{minimize} \qquad \mu \sum_{l=1}^{2} \sum_{p=1}^{4} \sum_{s=1}^{n} g_{l,p,s} + \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{5.1.26}$$

Subjected to

$$\begin{aligned} Constrains \quad & 5.1.2 - 5.1.10 \\ Constraints \quad & 5.1.12 - 5.1.22 \end{aligned}$$

This is a fully integrated (centralized) approach to the scheduling problem. There is however some possibilities for partial integration where each agent does not need complete knowledge of all the variables. One idea is the sharing of Lagrangian multipliers in a negotiation strategy between the agents. If g_k is a constraint for a optimization problem and Λ is the Lagrangian function[1], observe that

$$\frac{\partial \Lambda}{\partial g_k} = \lambda_k$$

[1] See [29] for an introduction to Lagrangian functions and their use in mathematical optimization

where λ_k is the Lagrangian multiplier associated with constraint g_k. Thus, λ_k is the rate of change of the quantity being optimized as a function of the constraint variable. Consider, for example, the case of producer domination (Section 5.1.2.1). After both optimization problems are solved the DSA could initiate a negotiation process. It could then share, for example, its largest Lagrangian multiplier with the PSA. This is a measure of the decrease in the value of the DSA cost due to the relaxation of that given constraint. The constraint in question is in this case a part of the production schedule given by the PSA. The PSA can then evaluate this value by some means and either choose to accept a small change to the production schedule or to refuse any changes. It could, for example, compare the Lagrangian multiplier with its relative increase in cost by changing the associated part of the production schedule.

The Lagrangian multipliers share core information with the other agents which yield efficient cooperation without the need of sharing 'unnecessary' information. Thus, the encapsulation principle of the agent is maintained. Note that Lagrangian multipliers cannot be obtained directly in integer programs because the Lagrangian function is not continuous. However this can be overcome by using the generalized Lagrangian multiplier method [30].

5.2 Implementation

The optimization problems were implemented in the AIMMS[2] optimization software. AIMMS is an advanced development environment for building optimization based decision support applications and advanced planning systems. It is used by leading companies in a wide range of industries in areas such as supply chain management, production planning, logistics, forestry planning and risk-, revenue- and asset- management. Next to a mathematical modeling language, AIMMS offers a number of advanced modeling concepts and a full graphical interactive user interface both for developers and end-users.

Three models where chosen for simulation purposes; producer dominance, distributor dominance and centralized approach. All of these are integer problems and as such they are in the NP-hard[3] class of decision problems [36]. One consequence of this is that there does not exist any proven algorithms to solve the problem in polynomial time [9]. Instead, the solution time typically grows exponentially with the number of variables in the problem. As such it is important to find clever ways to speed up the solution time. Some examples of techniques for faster solution times are:

- **Hot start:** After solving a problem for one distribution cycle for a set of inputs the solution can be stored and used as a starting point for the next set

[2] http://www.aimms.com/

[3] Problems that are not proved to be able to solve in polynomial time.

of inputs. This starting point will typically be much closer to optimal than the generic starting point used by the algorithm.

- **Stop before finish:** The algorithm can be stopped after a given time. The solution will not be optimal, but experience with the stopping time can provide near-optimal solutions that are deemed 'good enough'.
- **Solving in parallel:** If multiple computational units (CPUs) are available the problem can in some cases be solved in parallel by using decomposition techniques. As an example, if $I_{p,0} \geq \sum_{r=1}^{m} d_{p,r}$ Theorem 1 is valid and the producer- and distributor- subproblems can be solved i parallel.

Microsoft Excel was used for generating the input to the CPLEX solver in AIMMS. The input is then passed to AIMMS through a AIMMS EXCEL plug-in. After the simulations are done the result is written back into Excel. The test setup is shown in Figure 5.7.

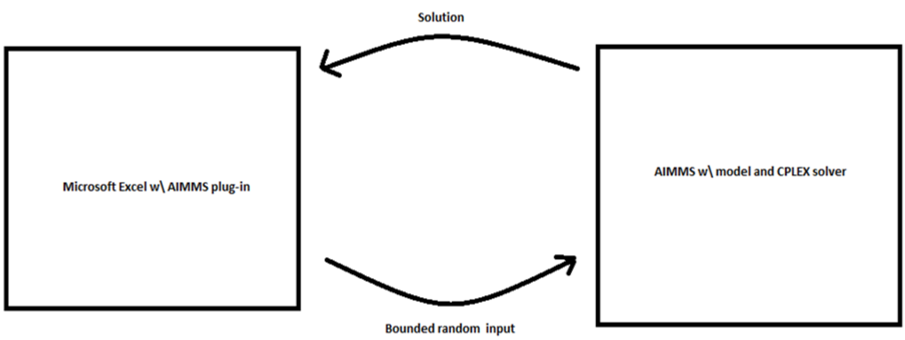


Figure 5.7: Simulation setup

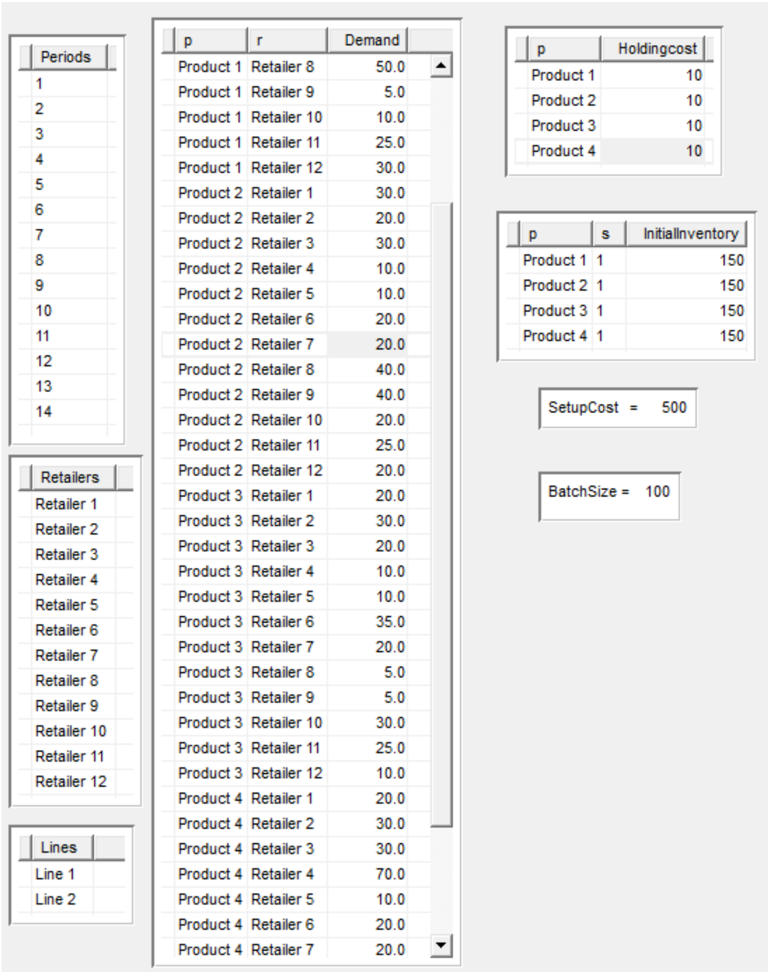
5.3 Results

The three models were solved using different input parameters. Two different setup cost $\mu \in \{500, 10000\}$ were used. For each setup cost, four different sets of holding costs $h \in \{(10,10,10,10),(8,9,10,13),(1,5,10,24)\}$ were used. For each of these instances an Excel macro generated 10 different demand matrices to be simulated. The demand was generated randomly such that for each retailer the total demand is less than 150 units. For the same demand matrix the three models were solved. Thus, there was done $2*3*10*3 = 180$ distinct simulations. Table 5.5 summarizes all the input data for the simulations. Figure 5.8 shows an example input for the producer domination model. Figure 5.3 shows the corresponding output from AIMMS.

For each instance of (μ, h) the average producers cost S_{av}, the average distributors cost T_{av} and the average total cost Ξ_{av} where recorded. Let S^*_{av}, T^*_{av} and Ξ^*_{av} denote the optimal values for the producer in producer domination, the distributor in distributor domination and the total cost in the centralized approach respectively. The average cost of conflict was calculated as $(S_{av} - S^*_{av})/S^*_{av}$ for the producer and $(T_{av} - T^*_{av})/T^*_{av}$ for the distributor in all three models. The distance from optimal solution for the total cost was calculated as $(\Xi_{av} - \Xi^*_{av})/\Xi^*_{av}$. The results for $\mu = 500$ is shown in Table 5.6. The results for $\mu = 10000$ is shown in Table 5.7.

Table 5.5: Input list

Input	**Value**
Number of retailers	12
Number of periods	14
Number of lines	2
Batch capacity	100
Setup cost	$\{500, 10000\}$
Holding cost	$\{(10,10,10,10),(8,9,10,13),(1,5,10,24)\}$
Demand	Bounded random



Periods
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Retailers
Retailer 1
Retailer 2
Retailer 3
Retailer 4
Retailer 5
Retailer 6
Retailer 7
Retailer 8
Retailer 9
Retailer 10
Retailer 11
Retailer 12

Lines
Line 1
Line 2

p	r	Demand
Product 1	Retailer 8	50.0
Product 1	Retailer 9	5.0
Product 1	Retailer 10	10.0
Product 1	Retailer 11	25.0
Product 1	Retailer 12	30.0
Product 2	Retailer 1	30.0
Product 2	Retailer 2	20.0
Product 2	Retailer 3	30.0
Product 2	Retailer 4	10.0
Product 2	Retailer 5	10.0
Product 2	Retailer 6	20.0
Product 2	Retailer 7	20.0
Product 2	Retailer 8	40.0
Product 2	Retailer 9	40.0
Product 2	Retailer 10	20.0
Product 2	Retailer 11	25.0
Product 2	Retailer 12	20.0
Product 3	Retailer 1	20.0
Product 3	Retailer 2	30.0
Product 3	Retailer 3	20.0
Product 3	Retailer 4	10.0
Product 3	Retailer 5	10.0
Product 3	Retailer 6	35.0
Product 3	Retailer 7	20.0
Product 3	Retailer 8	5.0
Product 3	Retailer 9	5.0
Product 3	Retailer 10	30.0
Product 3	Retailer 11	25.0
Product 3	Retailer 12	10.0
Product 4	Retailer 1	20.0
Product 4	Retailer 2	30.0
Product 4	Retailer 3	30.0
Product 4	Retailer 4	70.0
Product 4	Retailer 5	10.0
Product 4	Retailer 6	20.0
Product 4	Retailer 7	20.0

p	Holdingcost
Product 1	10
Product 2	10
Product 3	10
Product 4	10

p	s	InitialInventory
Product 1	1	150
Product 2	1	150
Product 3	1	150
Product 4	1	150

Figure 5.8: Input

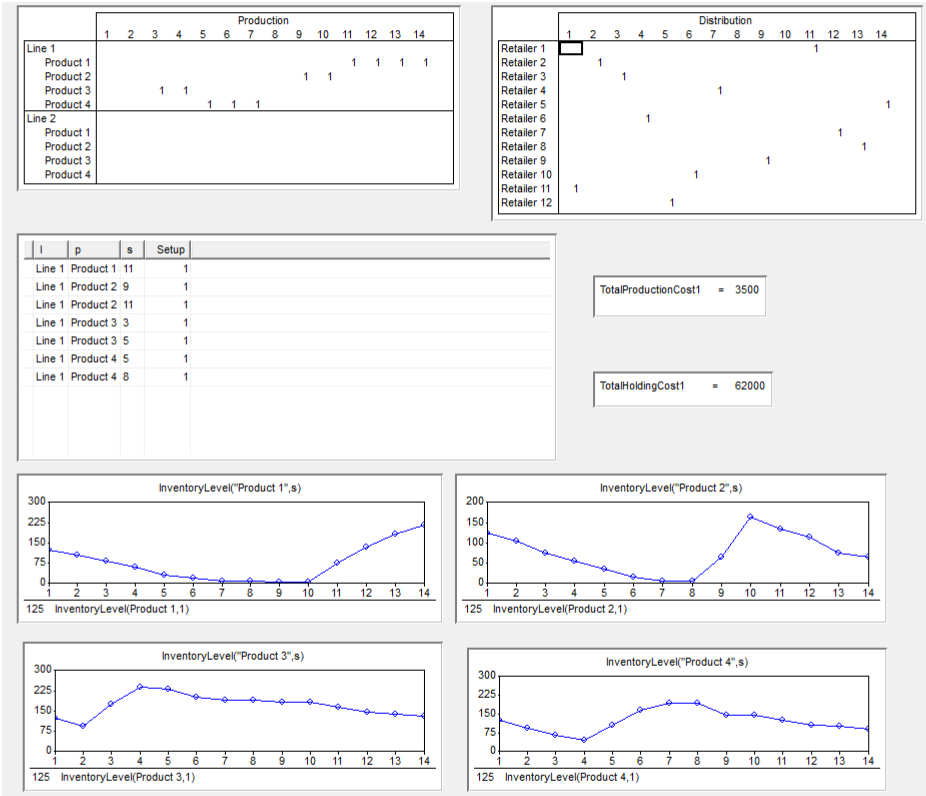


Figure 5.9: Output

Table 5.6: Results for $\mu = 500$

Scheme	**Cost of conflict** (%) **Producer**	**Distributor**	**Optimality gap**(%)
$\boldsymbol{h_1 = h_2 = h_3 = h_4 = 10}$			
Producer dom.	0	156,9	126,4
Distributor dom.	114,3	0	3,9
Centralized	62,9	0,5	0
$\boldsymbol{h_1 = 8, h_2 = 9, h_3 = 10, h_4 = 13}$			
Producer dom.	0	161,2	123,6
Distributor dom.	139,1	0	5,7
Centralized	63,1	3,5	0
$\boldsymbol{h_1 = 1, h_2 = 5, h_3 = 10, h_4 = 24}$			
Producer dom.	0	237,6	168
Distributor dom.	127,6	0	5,7
Centralized	51,4	6,1	0

Table 5.7: Results for $\mu = 10000$

Scheme	**Cost of conflict** (%)		**Optimality gap**(%)
	Producer	**Distributor**	
$\boldsymbol{h_1 = h_2 = h_3 = h_4 = 10}$			
Producer dom.	0	98,4	28,9
Distributor dom.	171,4	0	102,8
Centralized	0	12,23	0
$\boldsymbol{h_1 = 8, h_2 = 9, h_3 = 10, h_4 = 13}$			
Producer dom.	0	167,5	36
Distributor dom.	169,4	0	114,5
Centralized	0	19	0
$\boldsymbol{h_1 = 1, h_2 = 5, h_3 = 10, h_4 = 24}$			
Producer dom.	0	191,8	31,6
Distributor dom.	152,4	0	113,9
Centralized	0	61,7	0

5.4 Discussion

Three immediate observations can be made from the results:

- All diagonal elements in Table 5.6 and Table 5.7 are zero. This is because this is the same number that is being used as a reference for calculating the cost of conflict.

- It can be seen from Table 5.7 that when $\mu >> \|h_p\|$ the optimal production schedule for the PSA is equal to the centralized optimal solution. However, when μ is closer to $\|h_p\|$, as in Table 5.6 the centralized solution is more close to the optimal schedule for the DSA.

- It can also be seen from both tables that when the spread between the holding costs for the products increases the cost of conflict for the distributor increases[4]. This is reasonable because when the spread increases, it becomes more important for the distribution agent to get the more expensive products late in the distribution cycle.

Thus it is more beneficial with respect to the total cost to use producer domination in cases where $\mu >> \|h_p\|$. Distributor dominance is good in the cases where μ is closer to $\|h_p\|$ and particularly if the spread in h_p is large.

Of course, it can be seen that the non-cooperative approaches does in general suffer an optimality gap when considering the total cost, which supports Hypothesis 1. It may be natural to ask why a centralized approach should not be used in every case. For one, when considering supply chains the 'agents' may very well be different organizations. If this is the case the agents are by nature greedy[5]. This is in general also the case even if the agents are within the same organization but belonging to different departments. There may also be information that the organizations considers confidential. As such, a fully centralized approach may not be possible to implement because all information needs to be processed centrally. Another argument for not using a centralized approach may be limitation in the available computational power. Also, breaking down the model into smaller submodels increases the maintainability and flexibility with respect to alterations or re-optimizations.

However there are strong arguments for introducing some degree of cooperation between the agents. When a centralized approach is implemented the overall solution is improved by increasing the cost for one agents while decreasing the cost more for the other agent. One way to overcome the greedy nature of the agents is to compensate the 'loosing' agent for switching to a centralized approach. Classical game theory [23] states that this compensation need only be the amount by which the loosing parts cost increases, plus an infinitesimally small value. Still there is

[4]Notice also that the L_1 norm for all sets of holding costs are the same, such that this is not a variable.

[5]At least in a market economy

the issue concerning the sharing of confidential information. A more integrated approach could however be possible also with local processing of sensitive information with the sharing Lagrangian multipliers (Section 5.1.4).

In some cases even very simple cooperation can achieve great outcomes. As an example, in the cases where Theorem 2 is valid simply communicating h_p to the PSA can ensure a optimal solution for both agents in the case of producer domination.

Chapter 6

Total system analysis

In this chapter the scheduler will be implemented on the product marriage simulator. The purpose of this implementation is to explore Hypothesis 2 and Hypothesis 3. This is done by introducing a significant disturbance to the production system and noting performance for three different control approaches: open-loop control, multi-agent control and multi-agent control with scheduling. The control approaches will be evaluated on the basis on how well they meet the following goals:

1. Deliver all orders
2. Produce enough product to replenish inventory[1]
3. Minimize cost

In section 6.1 the product marriage MAS is presented. Section 6.2 explains the test setup and Section 6.3 presents the simulation results. Finally, in Section 6.4, the results are discussed and considered with respect to Hypothesis 2.

6.1 The product marriage MAS

The MAS used for control of the production process consist of the following agents:

- **Machine agents:** Each machine in the production system is assigned to one agent. The machine agents are responsible for controlling the speed of its machine. It also manages the resources associated with its machine.
- **Quality agents:** The quality agents are responsible for checking and controlling the quality of the end product. There is one quality agent on each production line.

[1]It is assumed that it is important for the robustness of the next distribution cycle to replenish inventory.

- **Line agents:** Each production line in the system has a line agent. The line agent acts as a medium to increase or decrease load on the specific production line.
- **Schedule agent:** This agent is responsible for monitoring the production schedule. If the production system is sufficiently behind schedule the schedule agent will request production ramp-up to one of the line agents.
- **Storage agent:** The storage agent is responsible for managing inventory and distribution to costumers.

Detailed information on how each agent is designed can be found in Pedersen December 2010 [27]. There is however some extensions made to the MAS to facilitate following a schedule. The machine agents have a table indicating which product to produce at each period. The storage agent also has a table for the distribution schedule. These tables are communicated to the agents at the start-up of each simulation from the scheduler. The JAVA implementation can be seen in Appendix B.

In particular there are two main control features present in the MAS. These are:

1. **Self-organizing resource allocation:** Each machine has a set of resources assigned to it. These resources determine the maximum speed at which the machine can run. The MAS has a self-organizing resource allocation built into it to optimally distribute these resources such that the ”worst-case bottle neck” of the system is optimized. At each time instant each machine agent has control over some subset of the total workers in the factory. The following variables are now defined:

   R - The set of all resources for the whole factory - Global set
   $R_i \in R$ - The subset of resources which agent i controls - Agent set
   c_i - The resource efficiency for machine agent i - Agent variable

   Note that $|R_1| + \cdots + |R_n| = |R|$. Further, we define the speed constraint of each machine to be given as

   $$\bar{\xi_i} = c_i * |R_i| \tag{6.1.1}$$

   The resource efficiency c_i is changing with some probability P[2]. Recall also that synchronization is done such that all machines can run on the same speed. The system will therefore run at the speed of the ”worst case bottle neck” if some of the machines are constrained. Thus it is reasonable to always try to improve the worst bottle neck of the production system. The following three part algorithm for optimal resource distribution is now proposed:

[2] If the resource is a worker, this could for example be caused by events earlier in the production line that render manual labor more or less effective.

Algorithm for resource request
read virtual control u_v
set $\bar{\xi}_i = c_i * |R_i|$
IF $u_v > \bar{\xi}$ THEN
set value $= c_i * (|R_i|)$
send request to all other agents for a new resource with value attached
END IF

Algorithm for request handling
IF new request for resource THEN
IF value$<c_i * (|R_i| - 1)$
set value $= c_i * (|R_i| - 1)$
send offer for one resource with value
END IF
ELSE
refuse request
END IF

Algorithm for resource offer
wait for response from all agents
pick the offer with highest $c_i * (|R_i| - 1)$

The three way handshake in the algorithm ensures that the agent does not receive more resources if it does not improve the lowest bottle neck of the network. There may be race conditions with this algorithm, but it does however always ensure that the multi-agent system ***self-organizes*** to the optimal solution after a transient period. Note that this is not done by a central node, but rather from the collective interactions between the agents.

2. **On-line production rate control:** The production system can fall behind the schedule in two possible ways; trashing product and/or producing at a too low rate. The quality agent notifies the schedule agent each time it trashes a product. The product marriage agent measures the system output speed and compares this to the reference speed to keep the schedule. It integrates the possible deviation and communicates it to the scheduler agent at regular time intervals. The scheduler agent keeps a counter which it increases each time it receives notification from the quality agent or the product marriage agent. If the counter exceeds half a batch it will request a ramp up in production from the line with the lowest load. This feature ensures that redundancy in the system is activated as soon as the deviation from the production schedule is larger than half a batch.

6.2 Test setup

The simulation input parameters can be seen in Table 6.1. Each machine is initialized with 3 resources with a resource efficiency of 15, which makes the initial speed constraint for each machine 45 units/minute. It can be calculated that the throughput of the system needs to be 40 units/minute for one batch to be produced in one period. It was found through trial that a distribution cycle time of 30 minutes was ideal as any shorter simulation time overloaded the CPU. The physical part of the simulator is influenced by a disturbance in the resource efficiency. This disturbance is set up such that in some time intervals the resource efficiency is reduced. After some time the resource return to the initial value of 15. It is natural to assume that this corresponds with a real life situation as the machine is likely to have some default operating conditions.

Table 6.1: Input List

Input	**Value**
Number of retailers	12
Number of periods	12
Number of lines	2
Batch capacity	100
Setup cost	500
Holding cost	$(1, 5, 10, 20)$
Distribution cycle	30 minutes
Initial number of machine resources	3
Default resource efficiency	15
Demand	One random instance

With this input the system can in theory meet the production demand with the use of only one line. However, due to disturbances, the second production line may need to be activated in case the production falls too far behind schedule. As with the problem formulation in Chapter 5 it is assumed that one line is ”locked” to producing one type of product in each period. Thus, if the second line is activated, it will run for one period producing one batch minus the products lost due to disturbances.

The simulation is carried out on three different control schemes. These are:

1. **Open-loop control:** The production and production schedule is fed directly into the system. The production system tries to follow this schedule strictly even in the event of disturbances.

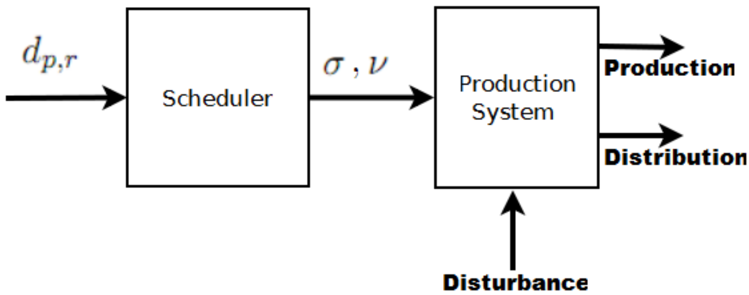


Figure 6.1: Block diagram for open-loop control.

2. **Multi-agent control without scheduling:** The MAS produces and delivers orders without any schedule. The orders are produced and delivered in a first-come-first-served fashion. The gain vector u is a collection of all the control signals given to to production system from the MAS. The measurement vector z is a collection of all measurement which are done by the MAS.

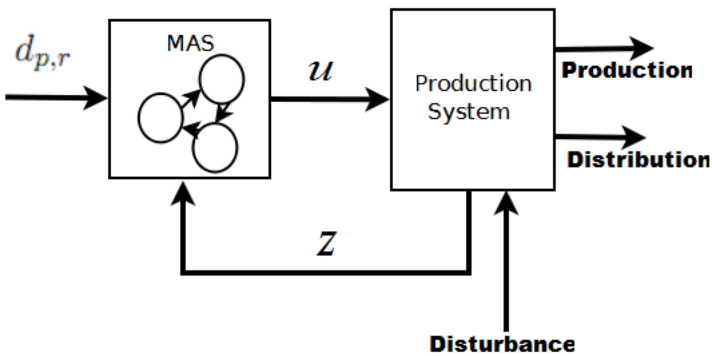


Figure 6.2: Block diagram for multi-agent control without scheduling.

3. **Multi-agent control with scheduling:** The schedule is fed into the MAS. Under normal operating conditions, the MAS follows the schedule strictly. However, in the event of disturbances, the multi-agent features explained in Section 6.1 will cause the system to deviate from the schedule. The gain vector u is a collection of all the control signals given to to production system from the MAS. The measurement vector z is a collection of all measurements done by the MAS.

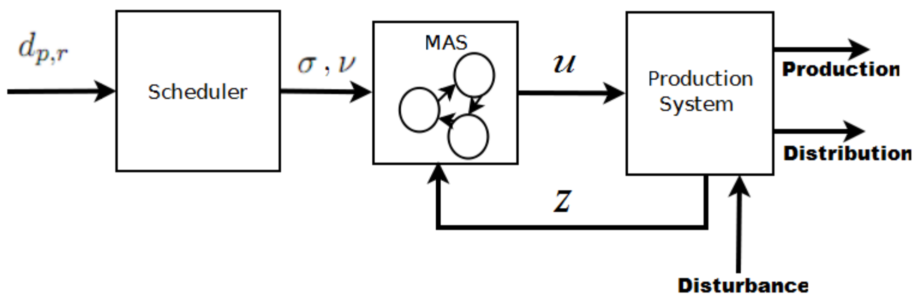


Figure 6.3: Block diagram for multi-agent control with scheduling.

For the two control schemes that utilizes a schedule the centralized optimization approach from Chapter 5 is used. Note also that all control approaches are applied the same disturbance, such that a comparison can be made under the same operating conditions.

6.3 Results

Figure 6.4, Figure 6.5 and Figure 6.6 shows the inventory for each control approach. The blue trajectory is the actual inventory from the simulation. The green dashed line is the theoretical inventory level calculated from the optimization model[3] (Chapter 5). The red dotted line shown in Figure 6.5 is the theoretical trajectory from a FIFO approach without any disturbances. Table 6.2 shows the deviation in holding cost from the theoretical optimal cost found from the optimization model[3]. The negative value given by the open-loop approach means that it gave a lower holding cost than that found from the optimization model[3]. The reason for this is that not enough product is produced to refill the inventory (see next section for discussion). The table also shows the number of extra setups used by each approach compared to the results from simulation on the optimization model[3].

Table 6.2: Simulation results

Control Scheme	**Δ Holding cost (%)**	**Additional start-ups**
Open-loop	96,4	6
MAS only	44,1	6
MAS w scheduling	-34,8	0

[3]Found with the centralized optimization model.

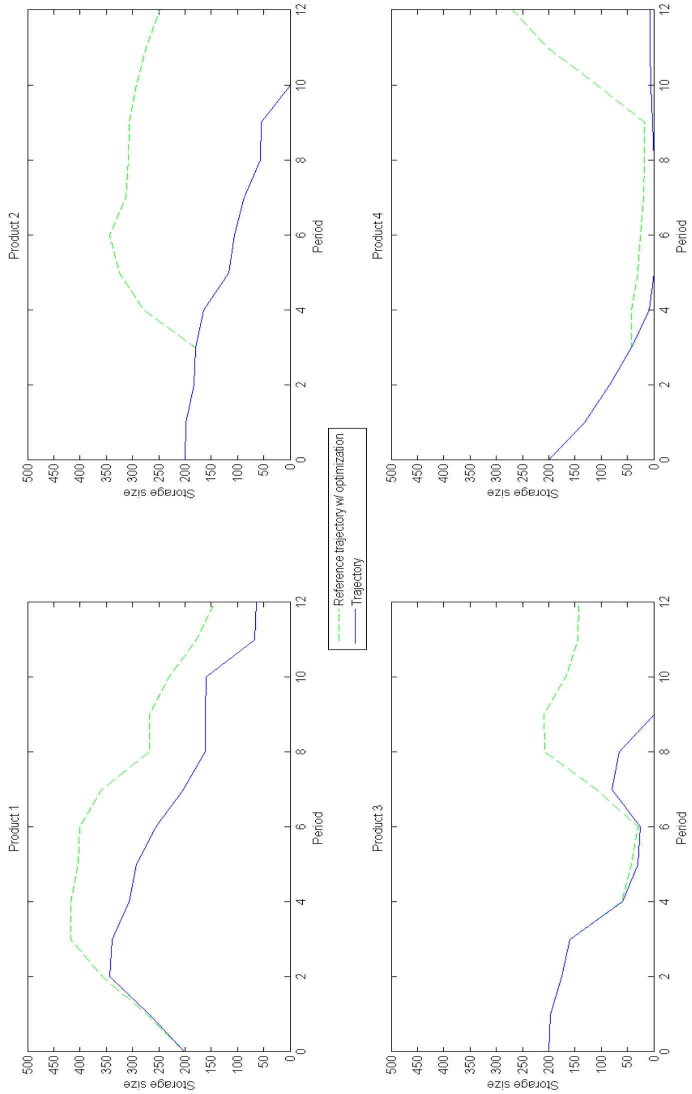


Figure 6.4: Simulation results for open-loop control.

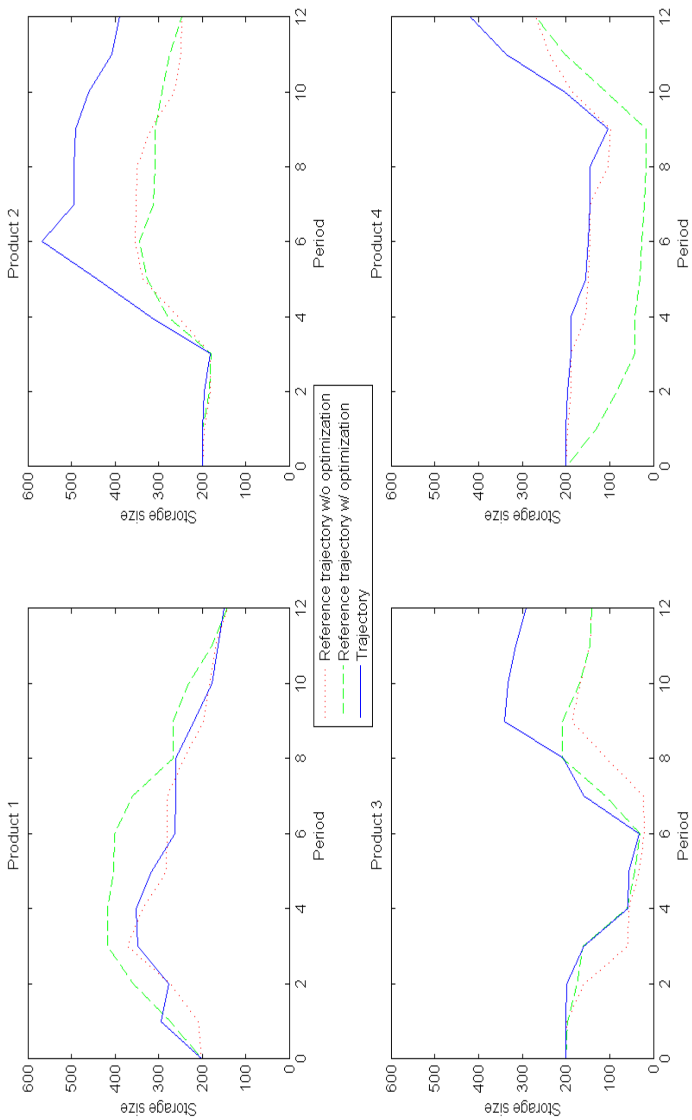


Figure 6.5: Simulation results for multi-agent control without scheduling.

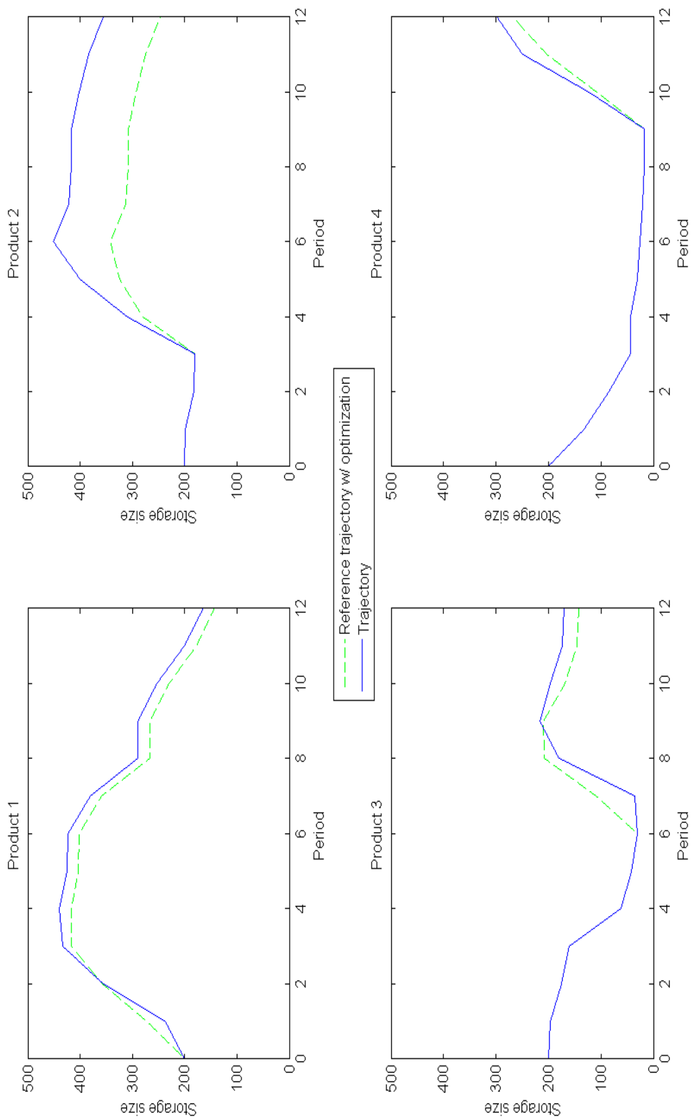


Figure 6.6: Simulation results for multi-agent control with scheduling.

6.4 Discussion

As seen from the inventory trajectory plots the open-loop control runs out of three products. This causes this control approach to miss orders because of insufficient inventory. As such, the open-loop control does not meet control objective 1 and 2. This should not come as a surprise, as this approach does not have any feedback. To be able to follow the reference it is dependent on the model being correct - which in this case means no disturbance.

On the other hand it can be seen that both the MAS approaches manages to keep all inventory levels above zero, such that all the orders are served. In fact, the on-line production rate control overshoots the reference trajectory. This is because the algorithms are set up in such a way that one whole batch is produced on the redundant line as soon as the production system is one *half* batch behind schedule. This design choice was made because of the belief that it is better to be sure that the inventory will be replenished rather than having the uncertainty that the inventory may end up lower than desired. Imagine, as an example, that second line was started up only when the production system was one whole batch behind production schedule. Then it could be possible that the production system ended up nearly one batch behind schedule without additional production being activated.

Although the efficiency of the on-line production rate control can certainly be discussed, it causes the system to meet the control objective 1 and 2. Other possible approaches includes, but are not limited to, re-optimization and single low layer controller (Figure 6.7 and Figure 6.8). Re-optimization is the idea that each time a disturbance occurs the state of the system is used as initial conditions to make a new schedule. The drawbacks of such an approach is that if the disturbance is not modeled, the system will continue to deviate from schedule even after the re-optimization is done. Also, when considering a real life production plant, the optimization model may become very large and time consuming to solve. As such it may not be practical or even possible to carry out the optimization with sufficient frequency. Non-model-based control can also be used instead of the MAS. As an example, a single PI[4] controller could be used to compensate for deviations from schedule. However, if such a controller does not involve some sort of heuristics, it will ignore the fact that there is an associated cost to start up the redundant line. As such, it could be tuned to follow the production tightly, but at the cost of many line start-ups which induces higher production costs. A choice has been made in this work to consider cases where these two approaches are not desirable or unfeasible.

[4]The MAS actually also has an integrating effect in that it does a counting of how many products the system is behind schedule.

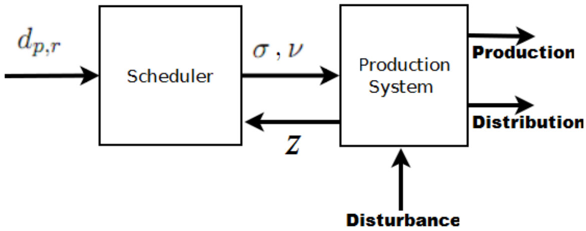


Figure 6.7: Re-optimization approach.

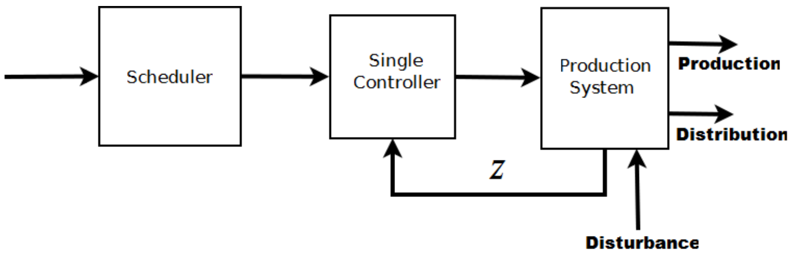


Figure 6.8: Single controller approach.

Further, it can be seen that multi-agent control with scheduling keeps the inventory of the more expensive products lower compared to the pure multi-agent approach. This is because the optimized schedule is design to minimize the holding cost. As a result, it can be seen in Table 6.2 that the holding cost deviates less from the theoretical optimal cost. Thus, the integration of an optimization layer in the control system reduces cost. This can be seen as an indication of the validness of Hypothesis 3.

Although the author realizes that this is a somewhat limited and constructed example, it gives quantifiable evidence in support of Hypothesis 2. That is, when induced with a significant disturbance that is presumed not possible to model, the MAS is superior to open loop control. If assumptions are made that other closed loop approaches are not suitable, MAS may very well be a favorable option.

It should also be noted that the disturbance in resource efficiency is not the only one effecting the system. The simulator is built in JAVA and counts over 4000 lines of code. As such, there are some non-modeled dynamics and uncertainties in the way the simulator behaves. There could, for example, be other processes running on the computer interfering slightly with the timing of some of the operations. This will of course also be the case in a real life production plants which is highly complex structure with a lot of different dynamics and dependencies.

Chapter 7

Discussion

The goal of this thesis has been to explore the field of MAS for manufacturing, from a high level to the low level implementation. Through the investigation it became clear to the author that the control theoretical foundation in this domain is very limited. It has also been the purpose of this thesis to spark the interest of readers with control theoretical background as well as those with a computer background. It is important for MAS developers to understand the control theoretical concepts of decentralized decision making. Also, it is equally important that control theorist understand the design and implementation issues associated with a particular control approach. Qualities like maintainability, scalability and cost of implementation are concepts that are sometimes overlooked. It is also the belief of the author that success of a particular control system is largely dependent on the industry being able to *understand* the implementation, which requires the system to be intuitive. One advantage with MAS is that they are very intuitive on an abstract level, much because of their modular nature and frequent use of heuristics. Having mutual understanding between these groups of scientists will thus be vital to the success of developing good multi-agent control systems in the future.

Chapter 2 emphasized the structural decision process of control system selection. This is important because its underlies the very reasons for using MAS control systems instead of the more traditional ones. This analysis step should always be done *before* getting into the details of implementation. The Chapter also shows that inspiration in this step can be drawn from slightly different fields like Skogestads work in process control design. More work in this area geared towards manufacturing is needed, as chemical and manufacturing application differ on certain points.

One of the benefits with the 'looseness' in the definitions of MAS is that multi-agent theory includes concepts that can be applied on many levels. This thesis shows that MAS can be used at both a high level, like in Chapter 5, and at lower levels, like in Chapter 6. Notice also that even humans fit within the definitions of an agent, and as such can be included into multi-agent networks for analysis.

Hypothesis 1 was exemplified in Chapter 5 which clearly shows that a optimality gap is induced in decentralized decision making. Readers with background from mathematical programming may not find this very surprising, but the idea that there is ”no free lunch” is important to keep in mind for multi-agent control designers. Design articles often does not emphasize this fact and it should certainly be given its fair weight in the measuring of pros and cons in the design process.

It is important to understand the two sides of Hypothesis 2. On one side, it states that a decentralized control approach can be justified in cases where there is no other choice. That is, if there are design constraints present that makes a centralized approach unfeasible you basically have not other choice. Examples that support this side of the Hypothesis were shown in Section 3.3 where the agents could only communicate with its nearest neighbor, leaving a centralized approach impossible. Notice also that as we increase the boundaries for what we define as 'the system', at some point a centralized approach is bound the be infeasible. No one controller can control the world, and at some point one has to look at a collection of entities. On the other side, the Hypothesis also states that a decentralized approach may be desirable even in cases where other control approaches *are* possible. The results from Chapter 6 show that the scheduler works better with a multi-agent system than without when the system is influenced by a disturbance. Of course, this test case is somewhat limited in complexity and the disturbance is very simple. Nevertheless, it is quantifiable evidence in support of Hypothesis 2. As most of the multi-agent design literature is based on qualitative analysis, this should serve as a motivation for more quantitative analysis of the conditions that favor multi-agent system[1]. Note also that Hypothesis 3 was supported in Chapter 6 as the use of the scheduling layer reduced the cost.

Literature review shows that only *one* multi-agent control system has been implemented to date in the industry[2]. This control system is no longer in operation because the designer left the company, which left the company unable to maintain it. For future implementation this should serve as a lesson for standardization of agent technology. Industry standards today show that companies in general favor well tested approaches. It should thus be of great importance for the success of multi-agent control systems that researchers include industry partners in their work. Hopefully, there will be some real test implementations of multi-agent technologies in the manufacturing industry, as this seems to be the next step to move forward.

[1]See Chapter 8 for future work suggestions.
[2]The DaimlerChrysler P2000+ production system. See Section 3.4 for more details.

Chapter 8

Conclusion and future work

8.1 Conclusion

This thesis has explored decentralized decision making as it relates to multi-agent control systems in the manufacturing industry. It has been shown how a manufacturing system can be decomposed into different domains for agent control. In this decomposition step inspiration was drawn from layered approaches like the one proposed by Skogestad. The control system selection process was further formalized and it was argued that under conditions where the system needs to exhibit a large degree of flexibility, multi-agent control approaches are favored over more specialized control structures.

A suitable test simulator for the product marriage process was developed in JAVA. This process is very commonly found in the manufacturing industry. A mathematical model for a scheduling problem on this simulator was further developed for the purpose of developing a top level optimizing control. Simulation results of the optimization problem show that there is an optimality gap introduced when distributing the optimization among two decision nodes when compared to centralized (fully cooperative) optimization.

The optimizing controller was implemented on the JAVA test bench. Simulations were carried out on three different control schemes; open-loop optimizing control, multi-agent control and multi-agent control with optimization. The results show that when the process is influenced by a disturbance in its operating efficiency that is not modeled, the multi-agent approaches are far superior to the open-loop control approach. Further, it is seen that the top level optimizing control improves the overall performance of the MAS by improving coordination, which in turn reduces cost. This supports the hypothesis that some degree of a layered approach can be beneficial for multi-agent control systems.

In summary, the simulations indicate that decentralized decision making may introduce optimality gaps, that multi-agent technology may be favorable in the presence of uncertain operating conditions, and shows the value of layered approaches in multi-agent control systems. Although the test case is simple and does not capture the full complexity of a real life production system, it gives quantifiable evidence to support the proposed hypotheses. This should serve as a motivation for performing more quantifiable research to identify which conditions favor the usage of multi-agent system, as opposed to purely qualitative publications.

8.2 Future work

Suggestions for future work includes, but are not limited to:

- More quantitative analysis is needed to identify more precisely the conditions that justifies multi-agent approaches. This could be done by developing more complex simulator software, or more preferably, a real test implementation.

- To date, there is no formal way of mathematically modeling a multi-agent system. Efforts should be made to formalize a way of modeling MAS that is general enough to encompass the wide variety of applications. Note that there has been efforts to develop such a formalized modeling in the field of cooperating autonomous vehicles, like in Bullo 2009 [3]. However, these models do not fit in with the dynamics present in the manufacturing industry.

- A problem with JADE is that Java is not developed for easy mathematical model implementation. It would thus be very beneficial to have MATLAB integrated with JADE to be able to perform agent simulations with more complex dynamics. A method for connecting these two software environments needs to be developed.

- More analysis on how the communication algorithms in a MAS effects convergence with respect to which information is shared is needed. As this is a wide problem formulation, it should be limited through a more specific MAS definition.

References

[1] Alonso and B. E. Ydstie. Stabilization of distributed systems using irreversible thermodynamics. *Automatic*, 37:1739–1755, 2001.

[2] H. Van Brussel, J. Wyns, P. Valckanaers, L. Bongaerts, and P. Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in Industry*, 37:255–274, 1998.

[3] Francesco Bullo, Jorge Cortès, and Sonia Martìnez. *Distributed Control of Robotic Networks*. Prinston University Press, 2009.

[4] S. Bussmann and K. Schild. An agent-based approach to the control of flexible production systems. In *8'th IEEE International Conferance on Emerging Technologies and Factory Automation*, 2001.

[5] G. Clair, E. Kaddoum, M. Gleizes, and G. Picard G. Self-regulation in self-organising multi-agent systems for adaptive and intelligent manufacturing control. In *Seond IEEE International Conference on Self-Adaptive and Self-Ogranizing Sytems*, 2008.

[6] M. Dawande, N. Geismar, N. Hall, and C. Sriskandarajah. Supply chain scheduling: Distribution systems. *Manufacturing & Services Operations Management*, 2004.

[7] M. Dorigo and A. Colorni. Distributed optimization by ant colonies. In *European Conference on Artificial Life*, 1992.

[8] L.A Doty. *Statistical Process Control*. Industrial Press Inc, 1996.

[9] J. Erickson. Lecture notes on np-hard problems, 2009. University of Illinois at Urbana-Champaign.

[10] FIPA. Fipa acl message structure specification. Technical report, Foundation for Intelligent Physical Agents, 2002.

[11] FIPA. Fipa agent message transport service specification. Technical report, Foundation for Intelligent Physical Agents, 2002.

[12] L. Gou, P.B Luh, and Y. Kyoya. Holonic manufacturing scheduling: architecture, cooperation mechanism and implementation. *Computers in Industry*, 37:213–231, 1998.

[13] L. Kercelli, A. Sezer, F. Öztoprak, P. Yolum, and S.I. Birbil. Mango: A multiagent environment for global optimization. In *Seventh Joint Coneference on Autonomous and Multi-Agent Systems*, 2008.

[14] R. Kota. *Self-Adapting Agent Organisations*. PhD thesis, University of Southampton, 2009.

[15] P. Leitao. *An Agile and Adpative Holonic Architecture for Manufacturing Control*. PhD thesis, Faculty of Engineering of University of Porto, 2004.

[16] P. Leitao. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22, 2009.

[17] P. Leitao and F. Restivo. A layered approach to distributed manufacturing. In *Proceedings of Advanced Summer Institute International Conference in Life Cycle Approaches to Production Systems: Management, Control and Supervision*, 1999.

[18] P. Leitao and F. Restivo. Adacor: a holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57:121–130, 2006.

[19] M. Luck, P. McBurney, O. Sheory, and S. Willmott. Agent technology roadmap (a roadmap for agent based computing), 2005.

[20] V. Marik and D. McFarlane. Intelligent manufacturing control. *IEEE Intelligent Systems*, 2005.

[21] L. Monostori, J. Vancza, and S. R. T Kumara. Agent-based systems for manufacturing. *CIRP ANNALS-MANUFACTURING TECHNOLOGY*, 55:697–720, 2006.

[22] M. Morari. Design of resilient processing plants -iii. *Chemical Engineering Science*, 38:1881–1891, 1983.

[23] J.F Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

[24] J. Oakland. *Statistical process control*. Butterworth-Heinemann, 2007.

[25] L. Panait and G. Mason. Cooperative multi-agent learning: The state of the art. *AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS*, 11:387–434, 2005.

[26] Sindre Pedersen. Multi-agent systems. SINTEF Report, August 2010.

[27] Sindre Pedersen. Steps towards the integration of control theory in multi-agent designs. TTK4550 Project report, December 2010.

[28] J.B. Rawlings and B.T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18:839–845, 2008.

[29] RT Rockafellar. Lagrange multipliers and optimality. *SIAM review*, 1993.

[30] J. F. Shapiro. Generalized lagrange multipliers in integer programming. *OPERATIONS RESEARCH*, 19:68–76, 1969.

[31] Y. Shi and Y.T. Hou. Distributed optimization for cooperative agents: application to formation flight. 2004.

[32] S. Skogestad. Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, 10:487–507, 2000.

[33] S. Skogestad. Near-optimal operation by self-optimizing control: From process control to marathon running and business systems. *Computers and Chemical Engieneering*, 29:127–137, 2004.

[34] M.; Harper J Townsend, C.; Begon. *Essentials of Ecology*. Blackwell, 2003.

[35] H. Van de Water and J Willems. The certainty equivalence property in stochastic control theory. *Automatic Control, IEEE Transactions on Automatic Control*, 26:1080 – 1087, 1981.

[36] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, 1998.

[37] M. Wooldridge. *An Introduction to Multi-Agent Systems*. John Wiley & Sons, 2002.

[38] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10, 1995.

[39] F. Zambonelli, N.R Jennings, and M. Wooldridge. Organisational abstractions for the analysis and design of multi-agent systems. *INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING*, 11:303–328, 2001.

Appendix A

The product marriage simulator v1.0

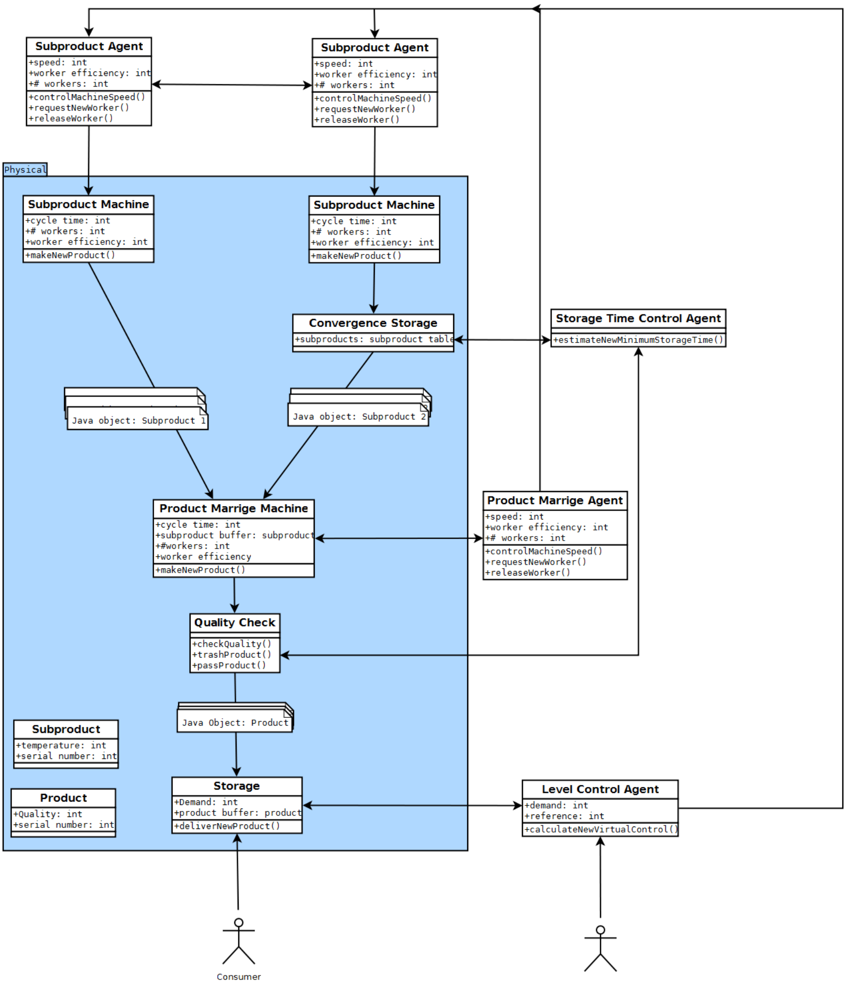


Figure A.1: The product marriage simulator as implemented in Pedersen December 2010 [27].

Appendix B

The product marriage simulator v2.0

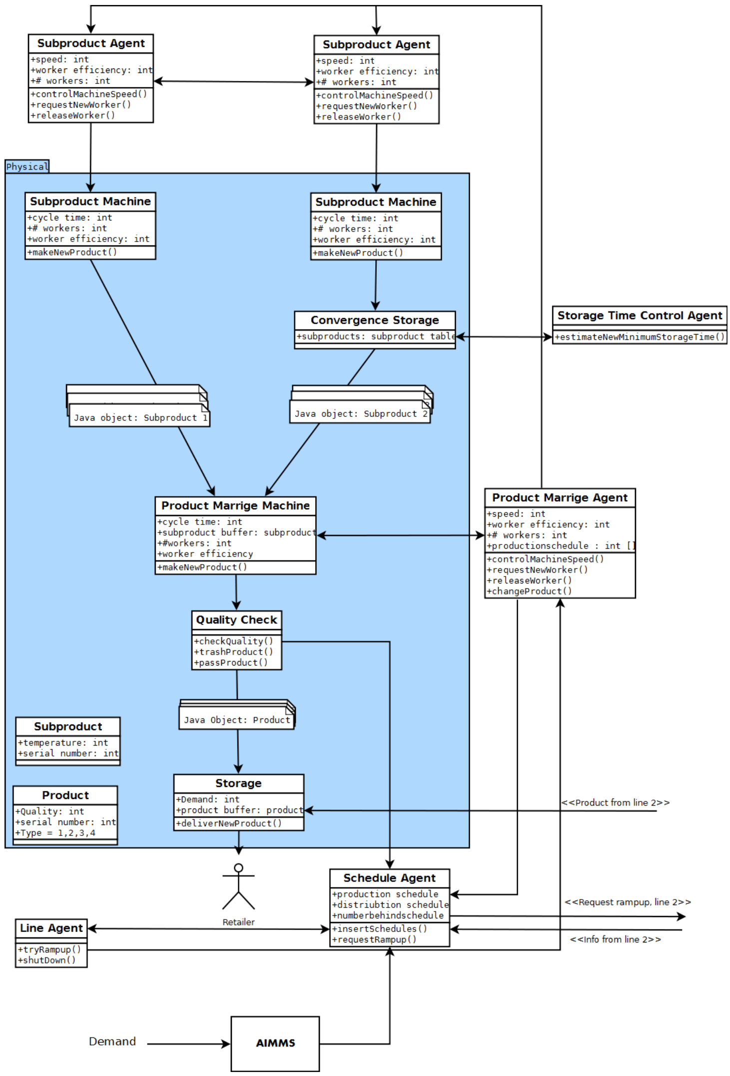


Figure B.1: The product marriage simulator as implemented in Chapter 6.

Appendix C

Conference article

The following article is derived from this master thesis and is aimed for submission to the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS) 2012 in Valencia. The AAMAS conference series was initiated in 2002 in Bologna, Italy as a joint event comprising the 6th International Conference on Autonomous Agents (AA), the 5th International Conference on Multiagent Systems (ICMAS), and the 9th International Workshop on Agent Theories, Architectures, and Languages (ATAL).

AAMAS is the largest and most influential conference in the area of agents and multiagent systems, the aim of the conference is to bring together researchers and practitioners in all areas of agent technology and to provide a single, high-profile, internationally renowned forum for research in the theory and practice of autonomous agents and multiagent systems.

AAMAS is the flagship conference of the non-profit International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

MAS for manufacturing control: A layered case study

Sindre Pedersen
Department of Engineering Cybernetics, Norwegian University of Science and Technology
NO-7491
Trondheim, Norway
sindrp@stud.ntnu.no

Johannes Tjønnås
Department of Applied Cybernetics, SINTEF Information and Communication Technology
NO-7491
Trondheim, Norway
johannes.tjonnas@sintef.no

Ingrid Schjølberg
Department of Applied Cybernetics, SINTEF Information and Communication Technology
NO-7491
Trondheim, Norway
ingrid.schjølberg@sintef.no

Bjarne Foss
Department of Engineering Cybernetics, Norwegian University of Science and Technology
NO-7491
Trondheim, Norway
bjarne.foss@itk.ntnu.no

ABSTRACT

Multi-agent systems presents a new paradigm for manufacturing control. Although there have been many publications in this field in recent years there is still an absence of industry adoption. It is argued that part of the reason for this could be that the field to date mostly presents *qualitative* arguments for its usage. In this paper we investigate the control of a simple manufacturing process using multi-agent systems, putting it in relation to classical control structures. As most current control systems are hierarchical, the analysis is done in a layered top-down fashion to facilitate for a more smooth transition into multi-agent control. Simulation results on the process presents, *quantitatively*, pros and cons with the introduction of multi-agent control into the manufacturing industry .

Categories and Subject Descriptors

J.7 [**Computer-aided Engineering**]: Computer-aided manufacturing

General Terms

Performance, Design, Theory, Algorithms, Experimentation

Keywords

Multi-agent systems, manufacturing control, optimization, layered control system

1. INTRODUCTION

The manufacturing industry in the western world is undergoing a paradigm shift from mass production to more specialized, customized production. In addition, the industry is experiencing increasingly diverse and volatile demands from the market [7][8]. The traditional control systems in the manufacturing industry are typically centralized and monolithic in structure [3]. Multi-agent manufacturing control is proposed as a new way of dealing with these challenges. Such control systems is said to have characteristics such as flexibility, agility and modularization which current rigid hierarchical control systems does not have. Some examples of such architectures can be found in [1] [2] [6].

In the field of control theory the notion of an agent is not very frequently used. However, MAS is an architecture that is decentralized in nature, and as such it puts restrictions on the possible control algorithms which can be implemented. Decomposition methods for mathematical optimization is an example of a field that has strong similarities with multi-agent systems in the sense that computational nodes may be distributed in space. It is well known that the interconnection of locally optimal objectives does not necessarily give a globally optimal objective. As an example, if the agents are greedy non-cooperative game theory states that the total system will converge to a Nash equilibrium which need not be the same as the globally best solution [9]. Rawlings and Stuart [11] show that a network of optimal controllers can be suboptimal and in fact also unstable if not special care is taken.

If measuring the performance of a control system with some objective function J (to be maximized), at an instant T a centralized control structure may thus be more optimal than a decentralized one, such that $J_c(T) \geq J_{dc}(T)$. If the centralized structure implements some globally optimal solution, the difference $J_c(T) - J_{dc}(T)$ is said to be a *optimality gap* [10]. With relation to optimality, the following statement is thus made:

STATEMENT 1. *Decentralized decision making can introduce an optimality gap when compared to fully integrated (centralized) solution.*

When considering a production plant, be it a chemical or

Cite as: MAS for manufacturing control: A layered case study, Sindre Pedersen, *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Yolum, Tumer, Stone and Sonenberg (eds.), June, 4–8, 2012, Valencia, Spain, pp. XXX–XXX.

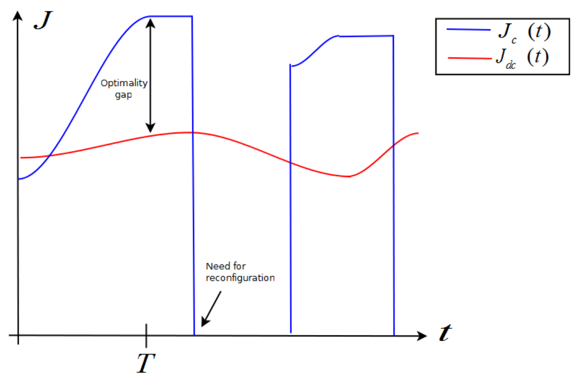


Figure 1: Optimality over time.

batch oriented one, it may have thousands of measurements and control loops. The issue of plantwide control considers control system design with emphasis on the structure of the overall plant [12]. It is in the realm of plantwide control that the justification for the usage of multi-agent systems is found. Multi-agent systems are architectures that implicates a decentralized control approach for plantwide control that aims to provide the system with a degree of robustness to variances. These variances can often be divided into operational variances, like rate of throughput, or external variances, like marked conditions. That is, the system should be able to function under the full range of operating conditions, internal and external, without the need for reconfiguration. This can be summarized in the following statement:

Statement 2. *Decentralized decision making can make a control system more flexible when compared to a fully integrated (centralized) implementation and can thus be more optimal* ***over time****.*

Although there may be a centralized control structure available that is specialized for the operating conditions *today*, it may be more beneficial to implement a decentralized structure that can also cope with the uncertainty of *tomorrow* with minimal need for expensive and time consuming reconfigurations. That is, over time the integral of the objective function may be larger for the decentralized control structure because it can handle a larger variety of operating conditions, such that $\int J_{dc}(t)dt \geq \int J_c(t)dt$. The idea is illustrated in Figure 1.

The multi-agent publications mentioned in the first paragraph all give excellent qualitative arguments for the use of multi-agent systems in manufacturing that follows in the lines of Statement 2. More control oriented literature, on the other hand, often emphasize Statement 1 and thus argues the usage of centralized control structures. It should be noted however that *both* the optimality gap $J_c(T) - J_{dc}(T)$ *and* the difference in accumulated difference in objective functions $\int J_{dc}(t)dt - \int J_c(t)dt$ should be weighted. This article thus also adds the following statement:

Statement 3. *A good control system gives a good balance between* ***optimality now*** *and* ***flexibility later***

It is the contribution of this article to compare agent-based approaches to more traditional ones in a *quantifiable* manner, exploring Statement 1 - 3, as this seems to be somewhat missing in the literature to date. Further, as most traditional control systems are hierarchical, examining a layered approach to multi-agent control can provide a more smooth transition into new multi-agent control systems. Thus, how agent-usage on different levels in a control system effects performance will be explored. This will be done trough a layered analysis of a simple manufacturing process.

2. PROCESS DESCRIPTION

We consider a simple manufacturing process where two parts are produced by separate machines and married into one final product. Each machine on the production line has a speed constraint given by system resources which can change in an unpredictable manner. That is, the maximum throughput of each machine is a variable that changes in a stochastic manner. Each of the two parts have a temperature $T_1 = T_{factory}$ and $T_2(t)$ respectively, where $T_{factory}$ is the temperature inside the factory compound. It is assumed that one of these parts is made from some materials that is stored outside the factory compound while the other part is made from materials inside the factory compound, such that $T_1 \neq T_2(0)$. The final product has an associated quality parameter which is a function of the difference between these two temperatures at the time of marriage. If the quality is out of a given bound, the product is deemed as trash before being put into the storage. This quality check is carried out by an intermediate quality station. Because of this, part 2 can be stored in a storage space before being sent to the marriage machine to converge (close enough) to the inside temperature.

The factory has two separate production lines which are terminated by four storage spaces designated for four distinct products. Products are being distributed from these storages to a number of retailers. Over a given distribution cycle each retailer has a demand for a combination of the four products. At each *period* in the distribution cycle *one* batch of one product can be produced from each production line and one truck can be filled with a combination of the products to serve *one* retailer. The number of retailers n is always greater or equal to the number of periods in the distribution cycle k. For each switch in production there is a cost μ. Also, there is an associated holding cost h_p for holding product p on storage for one period. For simplicity, it will be assumed that all other costs are constant. The process is shown in Figure 2.

The control goals for this system can be formulated as

1. Satisfy retailer demand
2. Avoid bottlenecks
3. Optimize the worst bottleneck
4. Minimize trash
5. Minimize the overall cost $\sum_{\forall h_p} h_p + \sum_{\forall \mu} \mu$

Goals 1-4 are all production specific goals while goal 5 is economic. Thus, an intuitive modularization would be to split the control system into two separate layers that each handles these two groups of goals. The idea is shown in Figure 3. The top layer scheduling controller makes coarse production and scheduling plans based on demand forecast.

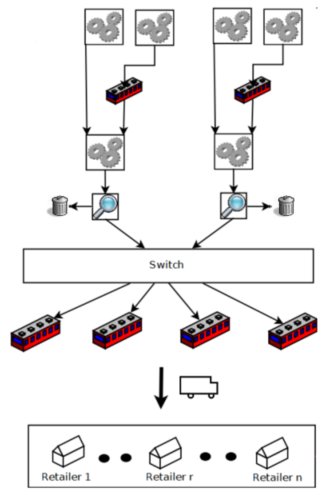


Figure 2: A simple manufacturing process with retailer distribution.

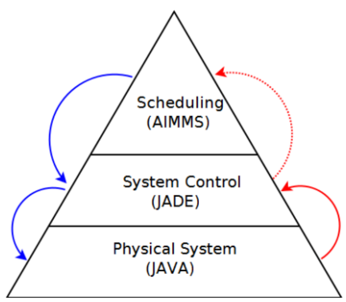


Figure 3: Layered control approach.

These coarse plans are passed down to system control which is responsible for real time production control. These control signals flowing down the layers are indicated as blue arrows in Figure 3. The red arrows indicate feedback from the layer below. As an example, the control layer receives measurements from the physical system through sensors. Notice that the feedback arrow from the control layer to the scheduling layer is not drawn in solid. This is because this feedback will not be implemented in this work, although it is possible in re-optimization schemes [4].

3. MULTI-AGENT CONTROL SYSTEM

The MAS used for process control includes the following agents.

- **Machine agents:** Each machine in the production system is assigned to one agent. The machine agents are responsible for controlling the speed of its machine. It also manages the resources associated with its machine.
- **Quality agents:** The quality agents are responsible for checking and controlling the quality of the end product. There is one quality agent on each production line.
- **Line agents:** Each production line in the system has a line agent. The line agent acts as a medium to increase or decrease load on the specific production line.
- **Schedule agent:** This agent is responsible for monitoring the production schedule. If the production system is sufficiently behind schedule the schedule agent will request production ramp-up to one of the line agents.
- **Storage agent:** The storage agent is responsible for managing inventory and distribution to the retailers.

The agents cooperate in order to achieve goals 1-4. Specifically, the MAS has a synchronization feature to coordinate the machine speeds to avoid product build up (bottle necks) at any point in the line. To minimize trash the quality agent and the machine agents cooperate to estimate minimum storage times before sending parts to product marriage. If the production falls behind schedule, the MAS has a mechanism for utilizing available capacity to catch up. The MAS also has a feature for self organization of machine resources to optimize the worst bottleneck. Self-organization is a vital element of multi-agent systems and this feature will thus now be explained in greater detail.

Self-organizing resource allocation

The MAS has a self-organizing resource allocation built into it to optimally distribute these resources such that the "worst-case bottle neck" of the system is optimized. At each time instant each machine agent has control over some subset of the total resources in the factory. The following variables are now defined

Table 1: Variables for self-organization

R	The set of all resources for the whole factory
$R_i \in R$	The subset of resources which agent i controls
c_i	The resource efficiency for machine agent i

Note that $|R_1| + \cdots + |R_n| = |R|$. Further, the speed constraint of each machine is given as

$$\bar{\xi_i} = c_i * |R_i| \tag{1}$$

The resource efficiency c_i is changing with some probability P.[1] Recall also that synchronization is done such that all machines can run on the same speed. The system will therefor run at the speed of the "worst bottle neck" if some of the machines are constrained. Thus it is reasonable to always try to improve the worst bottle neck of the production system. The following three part algorithm for optimal resource distribution is now proposed:

Algorithm for resource request
read virtual control u_v
set $\bar{\xi_i} = c_i * |R_i|$
IF $u_v > \bar{\xi}$ *THEN*
  set value $= c_i * (|R_i|)$
  send request to all other agents for a new resource with value attached

[1] If the resource is a worker, this could for example be caused by events earlier in the production line that render manual labor more or less effective.

END IF

Algorithm for request handling
IF new request for resource THEN
IF value$<c_i*(|R_i|-1)$
set value $= c_i*(|R_i|-1)$
send offer for one resource with value
END IF
ELSE
refuse request
END IF

Algorithm for resource offer
wait for response from all agents
pick the offer with highest $c_i*(|R_i|-1)$

The three way handshake in the algorithm ensures that the agent does not receive more resources if it does not improve the lowest bottle neck of the network. There may be race conditions with this algorithm, but it does however always ensure that the multi-agent system ***self-organizes*** to the optimal solution after a transient period. Note that this is not done by a central node, but rather from the collective interactions between the agents.

4. TEST SETUP

This section will present the setup used for simulation. Different setups for the top layer will be investigated, both single agent and multi-agent. Further, the low layer system control is implemented as the multi-agent system previously described. The layered setup (Figure 3) will be compared with two non-layered setups.

4.1 Scheduling

Both production and distribution are done in fixed batches of size C. The control goal for the scheduling layer is to create a production schedule σ and a distribution schedule ν that minimizes the total cost. This can be done by splitting the optimization problem up in two subproblems or it can be solved centralized. Two agents are introduced for solving the optimization problem in a distributed manner, namely the **Production Scheduling Agent (PSA)** and the **Distribution Scheduling Agent (DSA)**. Two non-cooperative setups between these two agents will be presented. In a centralized setup the optimization problem is solved by only one agent. Thus, three different structures can be distinguished in this control layer

- **Centralized approach:** In the centralized case there is one single agent which solves the total optimization problem. The corresponding outputs are the globally optimal production schedule σ^G and the globally optimal distribution schedule ν^G. The centralized model can be found in Appendix A.1

- **PSA domination:** In the case of producer domination the PSA finds a production schedule σ^P that minimizes the production cost. This implies that the cost function for the PSA is just a function of the production cost. This production schedule σ^P is then passed as a constraint to the DSA which finds a distribution schedule subjected to σ^P. The model used in this setup can be found in Appendix A.2.

- **DSA domination:** If the DSA is dominant it minimizes a cost function that is just a function of the distribution cost. The output will be a optimal production schedule σ^D and a optimal distribution schedule ν^D for the distribution agent. This production schedule is then passed to the PSA which creates a new, feasible production schedule where deviations from σ^D is penalized in the cost function. The model used in this setup can be found in Appendix A.3

All of the optimization models results in integer programs. In this work they are solved with the CPLEX[2] optimization software using AIMMS[3] as a modeling system.

4.2 Control system layering

To investigate different ways of modularizing the control system, two non-layered setups are proposed in addition to a layered setup. The following three setups where used for simulation:

- **Open-loop control:** The production and distribution schedule are applied directly to the simulator. The production system tries to follow this schedule strictly even in the event of disturbances. This approach is open-loop because there is no feedback from the simulator to the control system.

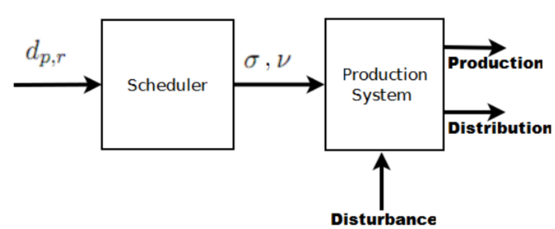


Figure 4: Open loop control.

- **Multi-agent control without scheduling:** The MAS produces and delivers orders without any scheduling layer. The orders are produced and delivered in a first-come-first-served fashion. The gain vector u is a collection of all the control signals given to the production system from the MAS. The measurement vector z is a collection of all measurement which are fed back to the MAS.

[2]IBM ILOG CPLEX Optimization Studio (http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/).
[3]Advanced Interactive Multidimensional Modeling System (http://www.aimmms.com).

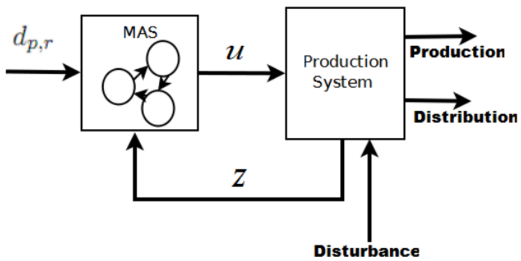


Figure 5: Multi-agent control.

- **Multi-agent control with scheduling:** The schedule is fed into the MAS. Under normal operating conditions, the MAS follows the schedule strictly. However, in the event of disturbances, the multi-agent features explained in Section 3 will cause the system to deviate from the schedule. The gain vector u is a collection of all the control signals given to to production system from the MAS. The measurement vector z is a collection of all measurement which are fed back to the MAS.

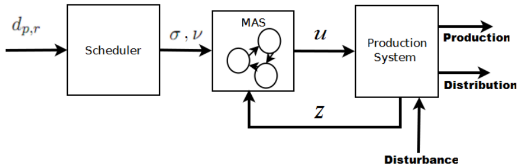


Figure 6: Multi-agent control with scheduling.

The centralized optimization approach is used for both open-loop control and multi-agent control with scheduling.

5. RESULTS

This section presents some simulation results from the scheduling layer setups and the different control system layerings presented in Section 4.

5.1 Scheduling results

The three models were solved using different input parameters. Two different setup cost $\mu \in \{500, 10000\}$ were used. For each setup cost, four different sets of holding costs $h \in \{(10,10,10,10),(8,9,10,13),(1,5,10,24)\}$ were used. For each of these instances a Excel macro generated 10 different demand matrices to be simulated. The demand was generated randomly such that for each retailer the total demand from each retailer is less than 150 units. For the same demand matrix the three models was solved. For each instance of (μ, h) the average PSA cost S_{av}, the average DSA cost T_{av} and the average total cost Ξ_{av} where recorded. Let S^*_{av}, T^*_{av} and Ξ^*_{av} denote the optimal values for the producer in producer domination, the distributor in distributor domination and the total cost in the centralized approach respectively. The average cost of conflict was calculated as $(S_{av} - S^*_{av})/S^*_{av}$ for the producer and $(T_{av} - T^*_{av})/T^*_{av}$ for the distributor in all three models. The distance from optimal solution for the total cost was calculated as $(\Xi_{av} - \Xi^*_{av})/\Xi^*_{av}$. The results for $\mu = 500$ is shown in Table 2. The results for $\mu = 10000$ is shown in Table 3.

Table 4: Simulation results

Control Scheme	Δ Holding cost (%)	Start-ups
MAS-only	96,4	6
MAS w. scheduling	44,1	6
Open-loop	-34,8	0

5.2 Control system layering results

In this simulation one demand matrix was used for all control schemes. The demand matrix was first given as a input to the centralized optimization model (Appendix A.1) with a corresponding outputs σ^G, ν^G, total distribution cost and total production cost. A holding cost of $h = (1, 5, 10, 24)$ was used. With the particular demand used the system can in theory meet the production demand with the use of only one line. However, the simulator is influenced by disturbances in resource efficiency and product quality. All the control schemes was influenced by the same disturbance such that a comparison can be made under the same operating conditions. This is not modeled in the optimization layer. Thus, the second production line can be activated in case the production falls to far behind schedule. As production is done in periods it is assumed that one line is "locked" to producing one type of product in each period. Thus, if the second line is activated, it will run for one period producing one batch minus the products lost due to trash and resource constraints.

Figure 5.2 shows the inventory in one storage for open-loop control. The non-solid green line is the theoretical trajectory found from solving the centralized optimization problem while the solid line is the actual inventory in the simulation. Notice the deviation.

Table 6.2 shows the deviation in holding cost from the theoretical optimal cost found from the centralized optimization model. Notice that the negative percentage of open-loop control means that it has a smaller holding cost than the theoretical optimal cost (explanation in Section 6). This table also shows the number of extra setups used by each approach in comparison with the centralized optimization model.

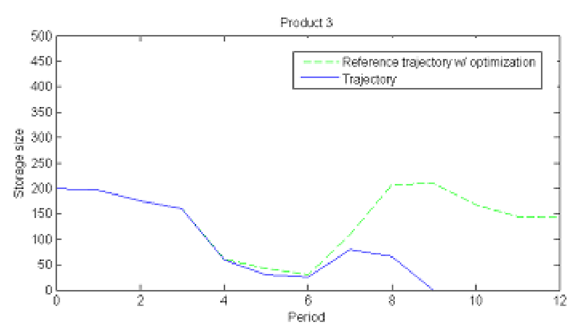


Figure 7: Inventory plot of one storage for open-loop control.

6. DISCUSSION

It can be seen from Table 3 that when $\mu >> \|h_p\|$ the optimal production schedule for the PSA is equal to the centralized optimal solution. However, when μ is closer to h_p the centralized solution is more close to the optimal sched-

Table 2: Results for $\mu = 500$

Scheme	**Cost of conflict** (%)		**Optimality gap**(%)
	Producer	**Distributor**	
$\boldsymbol{h_1 = h_2 = h_3 = h_4 = 10}$			
Producer dom.	0	156,9	126,4
Distributor dom.	114,3	0	3,9
Centralized	62,9	0,5	0
$\boldsymbol{h_1 = 8, h_2 = 9, h_3 = 10, h_4 = 13}$			
Producer dom.	0	161,2	123,6
Distributor dom.	139,4	0	5,7
Centralized	63,8	3,5	0
$\boldsymbol{h_1 = 1, h_2 = 5, h_3 = 10, h_4 = 24}$			
Producer dom.	0	237,6	168
Distributor dom.	127,6	0	5,7
Centralized	51,4	6,1	0

Table 3: Results for $\mu = 10000$

Scheme	**Cost of conflict** (%)		**Optimality gap**(%)
	Producer	**Distributor**	
$\boldsymbol{h_1 = h_2 = h_3 = h_4 = 10}$			
Producer dom.	0	98,4	28,9
Distributor dom.	171,4	0	102,8
Centralized	0	12,2	0
$\boldsymbol{h_1 = 8, h_2 = 9, h_3 = 10, h_4 = 13}$			
Producer dom.	0	167,5	36
Distributor dom.	169,4	0	114,5
Centralized	0	19	0
$\boldsymbol{h_1 = 1, h_2 = 5, h_3 = 10, h_4 = 24}$			
Producer dom.	0	191,8	31,6
Distributor dom.	152,4	0	113,9
Centralized	0	61,7	0

ule for the DSA. It can also be seen from both tables that when the spread between the holding costs for the products increases the cost of conflict for the DSA increases [4]. This is reasonable because when the spread increases it becomes more important for the distribution agent to get the more expensive products late in the distribution cycle. Thus it is more beneficial with respect to the total cost to use PSA domination in cases where $\mu >> \|h_p\|$. DSA dominance is good in the cases where μ is closer to $\|h_p\|$ and particularly if the spread in h_p is large.

Of course, both of the non-cooperative approaches does in general suffer a optimality gap when considering the total cost. This supports Statement 1 presented in the introduction. It may be natural to ask why a a centralized approach should not be used in every case. For one, when considering supply chains the 'agents' may very well be different organizations. If this is the case the agents are by nature greedy [5]. This is in general also the case even if the agents are within the same organization but belonging to different departments. There may also be information that the organizations considers confidential. As such, a fully centralized approach may not be possible to implement because all information needs to be processed centrally. Another argument for not using a centralized approach may be limitation in the available computational power. Also, breaking down the model in to smaller sub-models increases the maintainability and flexibility with respect to alternations or re-optimizations.

It can be seen from Table 4 that the layered approach (MAS with scheduling) has a lower holding cost than the MAS only approach. This is because the scheduling layer gives production and distribution plans which seek to minimize prolonged storing of more expensive products, while the MAS only approach produces and distributes in a FIFO fashion. Open-loop control gives a lower holding cost and no additional setups. This is because there is no MAS to handle the disturbances, and as a effect this control structure does not meet control goal 1. As seen from the plot in Figure 5.2 the inventory is empty in period 8, causing this control approach to serve incomplete orders to three retailers.

7. FUTURE CHALLENGES

Literature review shows that only *one* multi-agent system for manufacturing control has been implemented in the industry to date[6]. Industry standards today shows that companies in general favor well tested approaches. It is thus of great importance for the research community in this field to produce quantifiable evidence for the effectiveness of MAS. This includes exemplifying qualitative arguments presented in many publications like flexibility, responsiveness and re-configurability with simulation results. This work only presents a limited simulation of a manufacturing process. Work should be done to develop simulators capturing more of the complexity in a real manufacturing process.

As many publications start with the presumption that multi-agent control is superior to traditional hierarchical control structures, future work should also include quantitative results comparing the two approaches. As most manufacturing control systems *are* hierarchical, developing layered multi-agent control systems would provide the opportunity for a more smooth transition in implementation that can utilize the systems already in place. More work should also be done investigating possible performance benefits with such layered approaches, as this work shows it can in fact improve system performance when compared to pure multi-agent control.

8. CONCLUSIONS

This article presented a layered analysis for the control of a simple manufacturing process using multi-agent systems. It was shown that decentralized control can introduce an optimality gap when compared to a centralized solution. It was also shown that a layered multi-agent control structure gave performance benefits when compared to non-layered approaches when the process was influenced by a significant disturbance.

9. ACKNOWLEDGMENTS

This work was in part funded by the European Commissions Seventh Framework Programme for Research and Technological Development (FP7).

10. REFERENCES

[1] H. V. Brussel, J. Wyns, P. Valckanaers, L. Bongaerts, and P. Peeters. Reference architecture for holonic manufacturing systems: Prosa. *Computers in Industry*, 37:255–274, 1998.

[2] S. Bussmann and K. Schild. An agent-based approach to the control of flexible production systems. In *8'th IEEE International Conferance on Emerging Technologies and Factory Automation*, 2001.

[3] R. Colombo. An agent-based intelligent control platform for industrial holonic manufacturing systems. *IEEE Transactions on Industrial Electronics*, 53:322–337, 2006.

[4] J. Hoogeveen and A. P. A. Vestjens. Optimal on-line algorithms for single-machine scheduling. In *Proceedings of the Fifth Conference On Integer Programming and Combinatorial Optimization*, 1996.

[5] N. R. Jennings, S. Bussmann, and M. Wooldridge. *Multiagent Systems for Manufacturing Control*. Springer-Verlag Berlin And Heidelberg Gmbh & Co. K, 2010.

[6] P. Leitao and F. Restivo. Adacor: a holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*, 57:121–130, 2006.

[7] P. Leitão. Agent-based distributed manufacturing control: A state-of-the-art survey. *Artifical Intelligence*, 22:979–991, 2009.

[8] V. Marik and D. McFarlane. Industrial adoption of agent-based technologies. *IEEE Intelligent Systems*, 20:27–35, 2005.

[9] R. Myerson. *Game theory: analysis of conflict.* Harvard University Press, 1997.

[10] J. Nocedal and S. Wright. *Numerical Optimization.* Springer, 2000.

[11] J. Rawlings and B. Stewart. Coordinating multiple optimization-based controllers: New opportunities and

[4] Notice also that the L_1 norm for all the holding costs used are the same, such that this is not a variable.

[5] At least in a market economy

[6] The DaimlerChrysler P2000+ production system [5].

challenges. *Journal of Process Control*, 18:839–845, 2008.

[12] S. Skogestad. Plantwide control: the search for the self-optimizing control structure. *Journal of process control*, 10:487–507, 2000.

APPENDIX

A. OPTIMIZATION PROBLEM

This appendix will present the optimization models used to produce the results in Section 5. The symbols used in the models are given bellow.

Number of retailers	m
Number of periods	n
Number of lines	l
Batch capacity	C
Unit production time	t
Batch period	Ct
Distribution cycle	nCt
Production Schedule	σ
Distribution Schedule	ν
Products	P
Productionlines	L
Retailers	R
Periods	S
Setup cost	μ
Holding cost	h
Demand	d
Inventory level	I
Distribution variable	x
Production variable	y
Setup variable	g

A.1 Centralized model

$$\min_{x_{r,s},y_{l,p,s}} \quad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} + \mu \sum_{l=1}^{2}\sum_{p=1}^{4}\sum_{s=1}^{n} g_{l,p,s} + \Upsilon \|w\| \quad (2)$$

Subjected to

$$\sum_{r=1}^{m} x_{r,s} \le 1 \quad \forall s \in \{1,\dots,n\} \quad (3)$$

$$\sum_{s=1}^{n} x_{r,s} = 1 \quad \forall r \in \{1,\dots,m\} \quad (4)$$

$$I_{1,s} = I_{1,s-1} + C\sum_{l=1}^{2} y_{l,1,s} - \sum_{r=1}^{m} x_{r,s}d_{1,r} \quad \forall s \in \{1,\dots,n\} \quad (5)$$

$$I_{2,s} = I_{2,s-1} + C\sum_{l=1}^{2} y_{l,2,s} - \sum_{r=1}^{m} x_{r,s}d_{2,r} \quad \forall s \in \{1,\dots,n\} \quad (6)$$

$$I_{3,s} = I_{3,s-1} + C\sum_{l=1}^{2} y_{l,3,s} - \sum_{r=1}^{m} x_{r,s}d_{3,r} \quad \forall s \in \{1,\dots,n\} \quad (7)$$

$$I_{4,s} = I_{4,s-1} + C\sum_{l=1}^{2} y_{l,4,s} - \sum_{r=1}^{m} x_{r,s}d_{4,r} \quad \forall s \in \{1,\dots,n\} \quad (8)$$

$$I_{1,s} \ge 0 \quad \forall s \in \{1,\dots,n\} \quad (9)$$

$$I_{2,s} \ge 0 \quad \forall s \in \{1,\dots,n\} \quad (10)$$

$$I_{3,s} \ge 0 \quad \forall s \in \{1,\dots,n\} \quad (11)$$

$$I_{4,s} \ge 0 \quad \forall s \in \{1,\dots,n\} \quad (12)$$

$$x_{r,s} \in \{0,1\} \quad (13)$$

$$C\sum_{l=1}^{2}\sum_{s=1}^{n} y_{l,p,s} \ge k_p - w \quad \forall p \in \{1,2,3,4\} \quad (14)$$

$$\sum_{p=1}^{4} y_{l,p,s} \le 1 \quad \forall s \in \{1\dots,n\} \quad \forall l \in \{1,2,\} \quad (15)$$

$$g_{1,p,s} \ge y_{1,p,s} - y_{1,p,s-1} \quad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\dots,n\} \quad (16)$$

$$g_{1,p,s} \ge y_{1,p,s-1} - y_{1,p,s} \quad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\dots,n\} \quad (17)$$

$$g_{2,p,s} \ge y_{2,p,s} - y_{2,p,s-1} \quad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\dots,n\} \quad (18)$$

$$g_{2,p,s} \ge y_{2,p,s-1} - y_{2,p,s} \quad \forall p \in \{1,2,3,4\} \quad \forall s \in \{2\dots,n\} \quad (19)$$

$$w \in \mathbb{R}^4 \quad (20)$$

$$y_{l,p,s} \in \{0,1\} \quad (21)$$

$$(22)$$

$$g_{l,p,s} \in \{0,1\} \quad (23)$$

$$(24)$$

Constraint 3 is to ensure that no more than one retailer is being serviced at the end of each period. Constraint 4 is to ensure that each retailer is served once. Constraints 5 - 8 is to ensure that the inventory level at each period is being updated correctly. Constraints 9 - 12 is to ensure that the inventory level is always positive. Constraint 14 is to ensure that enough of product p is being produced to satisfy demand. Constraint 15 is to ensure that only one product is being produced on each line at each period. Constraints 16 - 19 is to assign the right value to the setup variable.

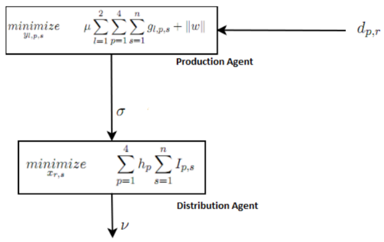


Figure 8: Producer dominance.

Note also that a slack variable w has been added to the objective function to ensure feasibility. This is due to the fact that the demand may be to large and in these cases it may not be feasible to refill the inventory.

A.2 PSA domination

The PSA solves

$$\underset{x_{r,s}}{minimize} \qquad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{25}$$

Subjected to

$$Constraints \quad 14-23$$

The DSA then solves

$$\underset{x_{r,s}}{minimize} \qquad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{26}$$

Subjected to

$$\begin{aligned} &Constraints \quad 3-13 \\ &x_{r,s} = \sigma^P \end{aligned}$$

The structure is shown in Figure 8.

A.3 DSA domination

The DSA solves

$$\underset{x_{r,s},y^*_{p,s}}{minimize} \qquad \sum_{p=1}^{4} h_p \sum_{s=1}^{n} I_{p,s} \tag{27}$$

Subjected to

$$\begin{aligned} &C\sum_{s=1}^{n} y^*_{p,s} \geq k_p - w \qquad \forall p \in \{1,2,3,4\} \\ &y^*_{p,s} \in \mathbb{R} \\ &Constraints \quad 3-13 \end{aligned}$$

The PSA then solves

$$\underset{y_{l,p,s}}{minimize} \qquad \mu \sum_{l=1}^{2}\sum_{p=1}^{4}\sum_{s=1}^{n} g_{l,p,s} + \Upsilon \|w\| + \Omega \left\| y^*_{p,s} - \sum_{l=1}^{2} y_{l,p,s} \right\| \tag{28}$$

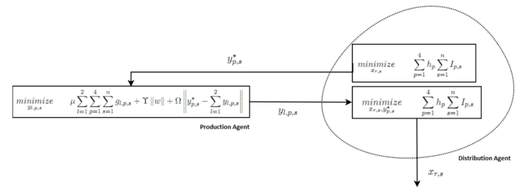


Figure 9: Distributor dominance.

Subjected to

$$\begin{aligned} &Constraints \quad 14-20 \\ &y^*_{p,s} = \sigma^D \end{aligned}$$

The structure is shown in figure 9