Impact of the Cluster Topology of Autonomously Controlled Material Flow Networks on the Performance of a Logistic System

Vom Fachbereich Produktionstechnik

der

UNIVERSITÄT BREMEN

zur Erlangung des Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte

DISSERTATION

von

M.Sc. Darja Wagner-Kampik

Gutachter*innen: Prof. Dr. Till Becker (Hochschule Emden/Leer)
Prof. Dr. Michael Freitag

Tag der mündlichen Prüfung: 25.04.2024

Abstract

The planning and control of production processes is increasingly reliant on centralized approaches. This means that production schedules must be developed in advance. With changing conditions, such as the arrival of rush jobs, either partial or complete rescheduling is required, and these tasks are time-consuming and cost-intensive. A possible solution is the implementation of autonomous production systems. This alternative enables logistic objects (e.g. machines) to make autonomous decisions based on current system conditions, so that they can react flexibly and quickly to any changes. Autonomous control is thus expected to contribute positively to the achievement of logistic objectives. Recent research indicates that a hybrid approach – a mix of autonomous job sequencing and central work system allocation – is particularly promising.

Research on autonomous production systems can be subdivided into three fields: technology, algorithms and topology. The first two fields have been sufficiently researched. The relevance of the topology of a material flow system has been confirmed, but the extent to which such topology should be considered when introducing autonomous control has not been adequately addressed. A new research approach, adapted from network theory, examines how clustering algorithms can be used to identify units of logistic objects that share intensive material flows. Such units are referred to as clusters. The logistic objects in a cluster should act as autonomous units to make decisions regarding the sequencing of jobs on a given resource, based on the state of the current system.

The objective of this thesis is to investigate whether, or to what extent, the cluster topology (i.e. structural characteristics of a cluster) can be used as an additional source of information to support the introduction of autonomous control. First, the thesis explores how job sequencing could occur decentrally based on previously identified material flow clusters. It appears promising to select the logic for the sequencing using the characteristics of identified clusters. Next, a simulation study is conducted using synthetically generated material flow networks. The networks differ in cluster topology to yield the highest possible degree of variability. A benchmark is used, and the results show that considering the cluster topology significantly improved the achievement of logistic objectives in more than half of the studied cases. Additionally, the results indicate that the size and number of clusters in a material flow network are crucial parameters that represent a potential starting point for further research.

Zusammenfassung

Bisher wird bei der Planung und Steuerung von Produktionsprozessen vermehrt auf zentralisierte Ansätze gesetzt. Dabei werden Produktionspläne im Voraus erstellt. Bei veränderten Bedingungen (etwa der Ankunft von Eilaufträgen) wird entweder teilweise oder vollständig neu geplant. Dies gestaltet sich jedoch zeit- und kostenintensiv. Eine denkbare Lösung bietet die Etablierung autonomer Produktionssysteme. Entsprechende Systeme befähigen logistische Objekte, auf Basis aktueller Systemzustände autonom Entscheidungen zu treffen. Dies ermöglicht es ihnen, flexibel und schnell auf mögliche Veränderungen zu reagieren. Dadurch kann autonome Steuerung einen positiven Beitrag zur logistischen Zielerfüllung leisten. Jüngste Forschungsergebnisse deuten darauf hin, dass ein hybrider Ansatz, also ein Mix aus autonomer Reihenfolgebildung sowie zentraler Maschinenzuordnung, vielversprechend ist. Bisherige Forschungsaktivitäten auf dem Feld autonomer Produktionssysteme lassen sich in drei Bereiche einteilen: Technologie, Algorithmen und Topologie. Während die ersten beiden Bereiche als umfassend erforscht gelten, ist die Relevanz der Topologie eines Materialflusssystems bestätigt. Jedoch wurde bislang nicht ausreichend herausgearbeitet, in welchem Umfang sie bei der Einführung autonomer Steuerung Berücksichtigung finden sollte.

Anhand eines neuen netzwerktheoretischen Forschungsansatzes wird untersucht, wie mit Hilfe von Clustering-Algorithmen die Identifizierung eines Zusammenschlusses von logistischen Objekten erfolgen kann, die über einen intensiven Materialflussaustausch verfügen. Ein entsprechender Zusammenschluss wird als Cluster bezeichnet. Die Objekte innerhalb eines Clusters sollen schließlich als autonome Einheiten agieren, die aufgrund der aktuellen Systemzustände Entscheidungen über die Reihenfolge der als nächste abzuarbeitende Aufträge treffen. Das übergeordnete Ziel dieser Arbeit bestand daher darin zu untersuchen, ob und in welchem Ausmaß die Clustertopologie, also die strukturellen Eigenschaften eines Clusters, als zusätzliche Informationsquelle zur Unterstützung der Einführung autonomer Steuerung genutzt werden kann.

Zunächst wurde aufgezeigt, wie die Reihenfolgebildung in solchen autonomen Clustern umzusetzen ist. Dabei scheint es Erfolg zu versprechen, die Logik für die Reihenfolgebildung auf Basis der Eigenschaften der Cluster auszuwählen. Anschließend fand die Durchführung einer Simulationsstudie unter Verwendung synthetisch erzeugter Materialflussnetzwerke statt, die sich in der Clustertopologie voneinander unterscheiden, um eine möglichst hohe Variabilität zu erreichen. Durch die Verwendung eines Benchmarks konnte nachgewiesen werden, dass das Einbeziehen der Clustertopologie in mehr als der Hälfte aller Fälle zur signifikant besseren Zielerreichung führt. Darüber hinaus implizieren die Ergebnisse, dass die Größe und die Anzahl der Cluster eines Materialflussnetzwerks entscheidende Parameter sind sowie einen potenziellen Ansatzpunkt für weitere Forschungen darstellen.

Contents

List of Abbreviations

ASIM	Association for Simulation
c	number of clusters
ConWIP	constant work-in-process
CRC	Collaborative Research Centre
DFG	German Research Foundation
EDD	earliest due date
ERP	enterprise resource planning
FCFS	first come first served
FG	fast greedy
GA	genetic algorithm
GN graph	Girvan-Newman graph
ICT	information and communications technology
IM	Infomap
ID	identity
LCFS	last come first served
LDD	latest due date
LFR graph	Lancichinetti–Fortunato–Radicchi graph
LNOP	least number of operations
LP	label propagation
LPT	longest processing time
LRPT	longest remaining processing time
LV	Louvain
MES	manufacturing execution system
MNOP	most number of operations
NMI	normalized mutual information
NP-hard	non-deterministic polynomial-time hard
NP-complete	non-deterministic polynomial-time complete
OLTPR	one logistics target per rule
p	degree of interconnection
PPC	production planning and control

QLE	queue length estimator
RC	relaxed caveman
RFID	radio frequency identification
RQ	research question
s	cluster size
SPT	shortest processing time
SRPT	shortest remaining processing time
TLPT	total longest processing time
TSPT	total shortest processing time
TWK	total work content
U	shop utilization
WINQ	work in next queue first
WIP	work-in-process
WS	work system
WT	Walktrap

List of Tables

List of Figures

1 Introduction

This introductory chapter provides a brief review of today's challenges in manufacturing systems. The problem area of this thesis is identified, and the chapter outlines the main objectives, research questions (RQs) and the methods used. The organization of the thesis is then summarized.

1.1 Motivation and Problem Definition

Since the beginning of the 21st century, scientists have been pointing to increasing dynamics and complexity in modern today's manufacturing systems (Alkan, Vera, Ahmad, Ahmad, & Harrison, 2018). Heterogeneous markets with strong fluctuations in demand as well as short product life cycles have been mentioned as critical factors (e.g. Windt & Hülsmann, 2007). Additionally, high product variance and unforeseen changes in customer requirements increase the uncertainty faced by manufacturing companies (e.g. Nyhuis & Wiendahl, 2009).

One possibility to address these limitations is the development of autonomously acting production systems (e.g. Freitag, Herzog, & Scholz-Reiter, 2004). To enable such autonomy, recent technological developments such as cyber-physical systems (e.g. Monostori et al., 2016) are used, where logistic units are connected through modern information and communication technologies (ICT) to enable them to communicate with each other and make autonomous decisions. The development toward more autonomy and decentralization is ongoing in practice (Fottner et al., 2021). The terms autonomy and decentralization are often used synonymously and are used interchangeably in this thesis.

The above factors create both challenges and opportunities in production planning and control (PPC). The main challenge is the inability of existing approaches with a centralized planning and coordinating unit to meet the new requirements, as the effects of a dynamic and complex production environment may be insufficiently considered (Kurbel, 2016). The transformation from centralization to decentralization also creates new opportunities. Decentralized systems are able to react quickly and flexibly to changing customer needs or operating conditions, resulting in a better achievement of logistic objectives (Liaqait, Hamid, Warsi, & Khalid, 2021). Moreover, due to the increasing integration of modern ICT, new algorithms and methodologies have been developed to enable autonomous decision-making (Lasi, Fettke, Kemper, Feld, & Hoffmann, 2014).

The challenge for the future will remain the effective introduction and implementation of these new technological solutions and advanced methods. Many ideas and applications will follow as part of initiatives such as Industry 4.0 (Kagermann, Helbig, Hellinger, & Wahlster, 2013). Experts agree that the transition toward decentralization will be made in several stages (Fottner et al., 2021). Concrete approaches are thus necessary to achieve the paradigm shift from centralized to decentralized systems.

For example, at the operational level, the broad application possibilities of sensors enable the gathering of shop floor real-time data to support effective scheduling (Liaqait et al., 2021; Mehrsai, Figueira, Santos, Amorim, & Almada-Lobo, 2017; Parente, Figueira, Amorim, & Marques, 2020). Scheduling addresses two decision problems: (i) the allocation of resources (e.g. human or machine) and (ii) the sequencing of jobs on a given resource (Pinedo, 2014). To date, manufacturing scheduling techniques have operated efficiently in systems with relatively few dynamics (Mehrsai et al., 2017). The shift away from centralization toward decentralization allows manufacturers to schedule close to real-time, which means they can deal more effectively with new challenges. Scientists and practitioners are concerned with how these newly available real-time data can be used in scheduling to allow for increased autonomy (e.g. Liaqait et al., 2021; Parente et al., 2020).

Along with the advantages of autonomous systems, there exists a certain degree of autonomy in highly complex systems. Beyond that point, the achievement of logistic performance may decline again, since the lack of a coordinating unit can sometimes lead to confusion in the decision-making process (e.g. Philipp, Böse, & Windt, 2006). To solve this dilemma, hybrid approaches combine central production planning with autonomous control. The hybrid approach offers the advantages of reliable central production planning as well as the ability to respond quickly and flexibly to discrepancies through autonomous control (Trentesaux, 2009; Zambrano Rey, Bonte, Prabhu, & Trentesaux, 2014). In this approach, scheduling-related decisions can be split into central work system allocation and autonomous sequencing of jobs (see e.g. Mehrsai et al., 2017; Schukraft, Grundstein, Freitag, & Scholz-Reiter, 2015; Schukraft, Grundstein, Freitag, & Scholz-Reiter, 2016). The results of such a hybrid approach are promising. They offer an opportunity for practitioners to support the gradual integration of decentralized control, because only one scheduling task follows a decentralized strategy (Mehrsai et al., 2017). Because of its potential, this thesis focuses on the hybrid approach.

According to Mourtzis and Doukas (2012), to reduce complexity in autonomous systems, practitioners can merge the individual work systems into two or more modules that have their own decision-making competencies. For the coordination of such modules, a high level of interaction between the elements within the modules is necessary (Mourtzis & Doukas, 2012). The integration of ICT at the shop floor level enables collecting relevant data regarding, inter alia, the element-to-element interactions. Such data can be transferred into a network representation that offers an objective overview of the material flows between the individual work systems, which is referred to as material flow networks (Becker, Beber, Windt, & Hütt, 2012). This representation provides new methods from complex network theory to support the implementation of decentralized control, with consideration given to the topology of a material flow network (Becker, Weimer, & Pannek, 2015; Vrabič, Husejnagić, & Butala, 2012).

The modularization of a system proposed by Mourtzis and Doukas (2012) resembles the idea of clustering in network theory. In clustering, with the help of algorithms, several elements are merged to form individual clusters, so that the elements within a cluster have more mutual connections than they do to elements outside the cluster (Newman & Girvan, 2004). The clustering process thus involves the topology of a material flow network. Considering that most material and information flows remain within a cluster, Vrabič et al. (2012) evaluated the hypothesis that such clusters are generally suited to perform as units of autonomous work systems. They demonstrated that identified clusters were associated with specific work processes, so that in principle they could operate autonomously. Derived from this, it seems to be a promising approach, although there is a need for concrete practical suggestions to implement autonomy within such clusters. According to the hybrid approach, it has to be decided what decision-making competences should be assigned to the individual clusters. Given the importance and potential of autonomous sequencing, it seems promising to perform the decision-making situation-based and outsource the decision-making to individual clusters. In this way, a sequencing rule is not assigned to each machine individually, as was previously the case, but rather to an entire cluster. Research has shown that the overall topology of a material flow network has a measurable impact on a system's performance (Becker et al., 2012). However, studies have not sufficiently examined how existing sequencing rules generally perform in autonomous clusters, or how the rules may need to be adapted or modified depending on the given cluster topology of the network.

1.2 RQs and Methodology

The main objective of this thesis is to ascertain whether – or to what extent – the cluster topology of a material flow network affects the logistic performance achievement when the job sequencing is decentralized within previously identified material flow clusters. Based on this objective, the following five RQs were formulated:

RQ 1: How can autonomous job sequencing be implemented in previously identified material flow clusters?

The importance of topology has been highlighted in previous research (Becker et al., 2012) and there are concrete proposals for identifying autonomous clusters (Becker & Wagner, 2015; Vrabič et al., 2012). However, there are still no strategies concerning how the autonomous sequencing of jobs as part of a hybrid approach could be realized within such autonomous clusters (Becker et al., 2015). Such an approach should have a high degree of practicability and be easy to implement, and it should initially be defined independently of the given cluster topology. A literature review on existing PPC methods offers the answer to RQ 1. The result is a conceptual framework for a hybrid clustering-based approach.

RQ 2: How can synthetic material flow networks be created with a predefined cluster topology?

Despite the increasing digitalization in manufacturing systems, the volume of publicly available data is small (Nyhuis, Mayer, & Kuprat, 2014). The purpose of answering RQ 2 is to provide a data basis to study the influence of the cluster topology and demonstrate the practical applicability of the proposed hybrid clustering-based approach on different material flow systems with varying cluster topology. Certain models can be used to create synthetic networks (e.g. Lancichinetti & Fortunato, 2009b; Orman, Labatut, & Cherifi, 2011), but so far they have not addressed in sufficiently specific terms the emergence of material flow networks. Accordingly, to answer RQ 2, this thesis proposes a methodological approach to generate synthetic material flow networks with a predefined cluster topology.

RQ 3: Which global network information should be used to identify autonomous clusters from a network theory perspective?

Vrabič et al. (2012) and Becker and Wagner (2015) proposed concrete clustering algorithms to identify autonomous clusters. However, many other clustering algorithms are provided in literature to reveal hidden structures (Fortunato, 2010). These algorithms not only consider different clustering criteria but also enable including additional network information, such as the intensity or direction of the material flow. Considering or ignoring such information may have a major impact on the resulting clusters (Lancichinetti & Fortunato, 2009b) and thus the arrangement of the autonomous clusters. Therefore, a set of clustering algorithms was selected and applied that considered different kinds of network information. The evaluation of the various clustering algorithm outputs addresses RQ 3 from a network theory perspective.

RQ 4: How does the cluster topology of material flow networks relate to the logistics performance?

The general relationship between network topology and logistics performance has previously been documented (e.g. Becker et al., 2012; Vrabič, Škulj, & Butala, 2013). However, similar work has not been undertaken regarding the cluster topology in material flow networks. Understanding this relationship provides insights into the key challenges to overcome through appropriate shop floor management. To answer RQ 4, a simulation study was conducted to examine the relationship between the cluster topology and the logistics performance. Many synthetically generated material flow networks that differed from each other, particularly in their cluster topology, were employed.

RQ 5: Do systems having autonomous clusters that are formed according to the intensity of material flow improve the logistical performance?

This RQ is answered in three consecutive steps, with each step providing the input for the next steps. Thereby, the following sub-questions are addressed:

- **Sub-RQ 5.1:** For material flow systems that possess varying cluster topology, how effective are selected sequencing rules to reduce the mean tardiness of tardy jobs and enhance the schedule reliability?

  A set of sequencing rules within autonomous clusters was applied. First, their effectiveness in a range of scenarios – in particular regarding the cluster topology of different material flow networks – was tested in a simulation study. These tests provided an answer to sub-RQ 5.1. The results for the best sequencing rule for each scenario provided a benchmark to answer the next sub-questions.

- **Sub-RQ 5.2:** Which global network information should be used to identify autonomous clusters, from a logistics perspective?

  As stated above under RQ 3, considering or ignoring certain network information may have a major impact on the resulting clusters. Different algorithms can also result in different clustering outputs. However, when these outputs are used as autonomous clusters, the characteristics of individual clusters can be decisive because the selection of an appropriate sequencing rule may depend on them. To answer sub-RQ 5.2, the different outputs of the clustering algorithms generated to answer RQ 3 were treated as autonomous clusters. Hence, the sequencing rules were applied as proposed in the answer to RQ1. The evaluation from a logistics-based perspective was undertaken through a simulation. The individual simulation results were compared with the results from the respective benchmark sequencing rule.

- **Sub-RQ 5.3:** What is the efficiency of sequencing rules applied in autonomous clusters to reduce the mean tardiness of tardy jobs and enhance the schedule reliability in material flow systems with varying cluster topology?

  To make a final conclusion about whether the cluster topology of a given material flow network had a significant impact and thus should be taken into account in the gradual introduction of decentralized control, the performance of the proposed hybrid clustering-based approach was compared with the respective benchmark sequencing rule in a simulation study. This analysis yielded an answer to sub-RQ 5.3.

The above descriptions of the RQs indicate that it was necessary to use a combination of methods. In particular, the main research methods of scheduling, synthetic network, clustering and discrete-event simulation were used to address the above listed RQs. Additional methods are discussed in detail in the relevant sections of this thesis. Below is a description and justification of the four research methods mentioned.

Scheduling

The diverse contributions in the *Journal of Scheduling*[1] show the relevance of scheduling in both theory and practice and across many disciplines and applications. Scheduling problems are optimization problems and are usually known to be non-deterministic polynomial-time hard (NP-hard; Pinedo, 2014). To solve such problems, heuristics have proven to be practical, with solutions that are close to the optimum (Liaqait et al., 2021).

Given the ongoing technological progress in manufacturing systems, scheduling holds strong importance due to the availability of real-time data (see e.g. Parente et al., 2020). This data provides major potential for the effective application of existing heuristics and enables developing new heuristics (Lasi et al., 2014). The thesis proposes one approach to using such heuristics in autonomous clusters.

Synthetic Networks

In almost all disciplines, the availability of data is limited. This is not only due to technological limitations but is also because data are often not accessible to the public. Hence, new research activities arises in different disciplines that addresses the designing of synthetic networks (e.g. Palla, Lovász, & Vicsek, 2010). With the help of such synthetic networks, for example, new approaches and concepts can be tested for their effectiveness. The advantage is that the network generation process is relatively transparent and can typically be influenced by only a few parameters. Thus, the process can also be manipulated to generate various types of networks having specific properties (e.g. Lancichinetti & Fortunato, 2009b). This process is suited to generate material flow networks with a built-in cluster topology.

[1] https://www.springer.com/journal/10951, Retrieved October 14, 2022.

Clustering

In the theory of complex networks, the development and application of clustering algorithms have evolved into a specific research field (Emmons, Kobourov, Gallant, & Börner, 2016; Yang & Leskovec, 2015). In the simplest cases, little information is necessary and no prior knowledge is required (Fortunato, 2010). This extends the number of possible applications to reveal underlying relationships among the individual network elements (Lancichinetti, Fortunato, & Radicchi, 2008). Given this practicability, the use of clustering algorithms to identify autonomous clusters seems to be promising.

Discrete-Event Simulation

In manufacturing, new PPC approaches can be tested cost-efficiently and easily in different scenarios with the help of simulation, without affecting the actual production (Rabe, Spieckermann, & Wenzel, 2008). The importance of simulations becomes clear in particular through the numerous expert conferences on this subject (e.g. organized by the *Association for Simulation,* ASIM,[2] in the German-speaking region). Hence, the use of simulations is an essential part of this thesis.

1.3 Thesis Structure

This thesis comprises seven chapters. The structure of the thesis is illustrated in Figure 1.1, indicating in which sections the individual RQs are answered. Following this introduction, Chapter 2 summarizes the necessary theoretical knowledge of the principles and methods of PPC. The application of methods drawn from complex network theory on manufacturing systems is motivated. Given this knowledge, a hybrid clustering-based approach is derived. Chapter 3 provides a general understanding and the definitions of terms and concepts in complex network theory. Chapter 4 proposes a new methodological approach to generate material flow networks with varying cluster topology to evaluate the proposed hybrid clustering-based approach. Based on these material flow networks, the outputs of various clustering algorithms – which cover different network information – are evaluated and compared in Chapter 5. In Chapter 6, several studies are described, which can be divided into three subsections. First, the simulation setup and the assumptions made in this context are described. Then the relationship between cluster topology and logistic performance is examined. In the third subsection, the efficiency of the proposed hybrid clustering-based approach is evaluated, to demonstrate the impact of the cluster topology. Chapter 7 summarizes the main findings of the thesis and answers the five corresponding RQs. The chapter closes with an outlook regarding further research activities related to the findings.

[2] https://www.asim-gi.org/, Retrieved October 14, 2022.

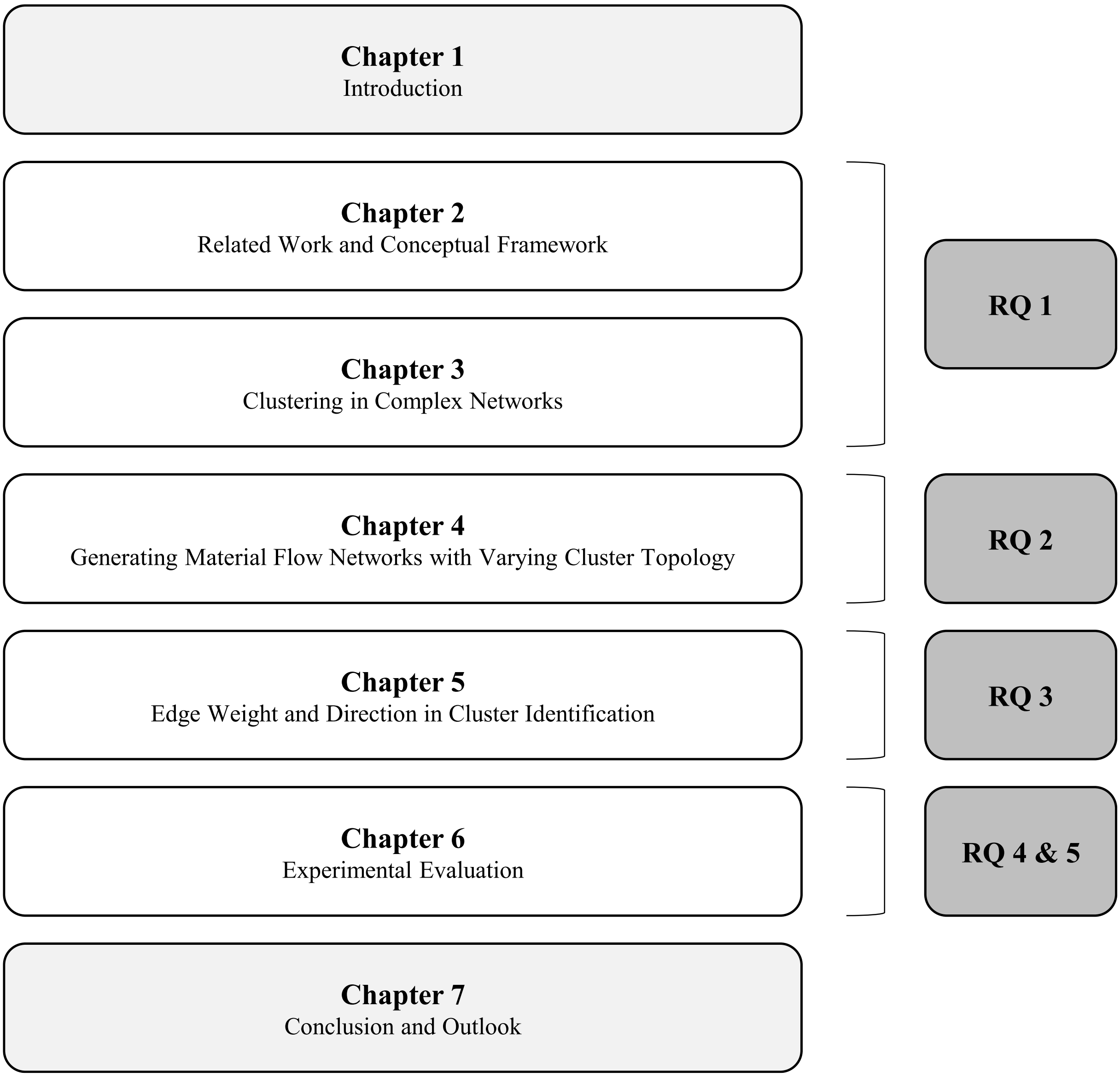

Chapter 1
Introduction
Chapter 2
Related Work and Conceptual Framework
Chapter 3
Clustering in Complex Networks
Chapter 4
Generating Material Flow Networks with Varying Cluster Topology
Chapter 5
Edge Weight and Direction in Cluster Identification
Chapter 6
Experimental Evaluation
Chapter 7
Conclusion and Outlook
RQ 1
RQ 2
RQ 3
RQ 4 & 5

Figure 1.1: Structure of thesis and relationship of chapters to RQs.

2 Related Work and Conceptual Framework

Given the many concepts and methods that are relevant to this thesis, the literature review was broad. Section 2.1 explains how production systems can be viewed from a system theoretical perspective, which is relevant for the classification of production systems as transformation systems with value-adding processes. Moreover, the four main logistic objectives are outlined in detail as they are essential for the thesis. In line with the overall objective of this thesis, key terms and basic concepts of PPC are introduced (Section 2.2). Next, various sources and thus also types of complexity and dynamics with which today's manufacturing companies are increasingly confronted are described (Section 2.3). Section 2.4 shows what attempts are made to deal with complexity and dynamics through centralized, decentralized and hybrid PPC approaches. In particular, the role of the topology of a material flow system in the previous research is highlighted. This review lays the theoretical foundation to propose a conceptual framework for the hybrid clustering-based approach. The chapter closes with a summary (Section 2.5).

2.1 A System Theoretical Perspective on Production Systems

A production system can be described using systems theory (Bellgran & Säfsten, 2010; Wiendahl, Reichardt, Nyhuis, & Rossi, 2015). Bellgran and Säfsten (2010) described a system as the sum of elements (e.g. machines and people) that are related to each other. The system's structure is thus determined by their reciprocal relations with each other. In general, a system is divided into several subsystems. The overall system and its components (i.e. individual elements and subsystems) are separated from the environment by the system's boundary. Depending on the level of interaction between a system and its environment, a distinction can be made between closed and open systems. As shown in Figure 2.1, a closed system is isolated from its environment. In contrast, an open system interacts dynamically with the environment and both influences it (through output) and is influenced by it (through input). According to Bellgran and Säfsten (2010), production systems are open systems and are thus an example of a transformation system. They are also known as an input-output model.

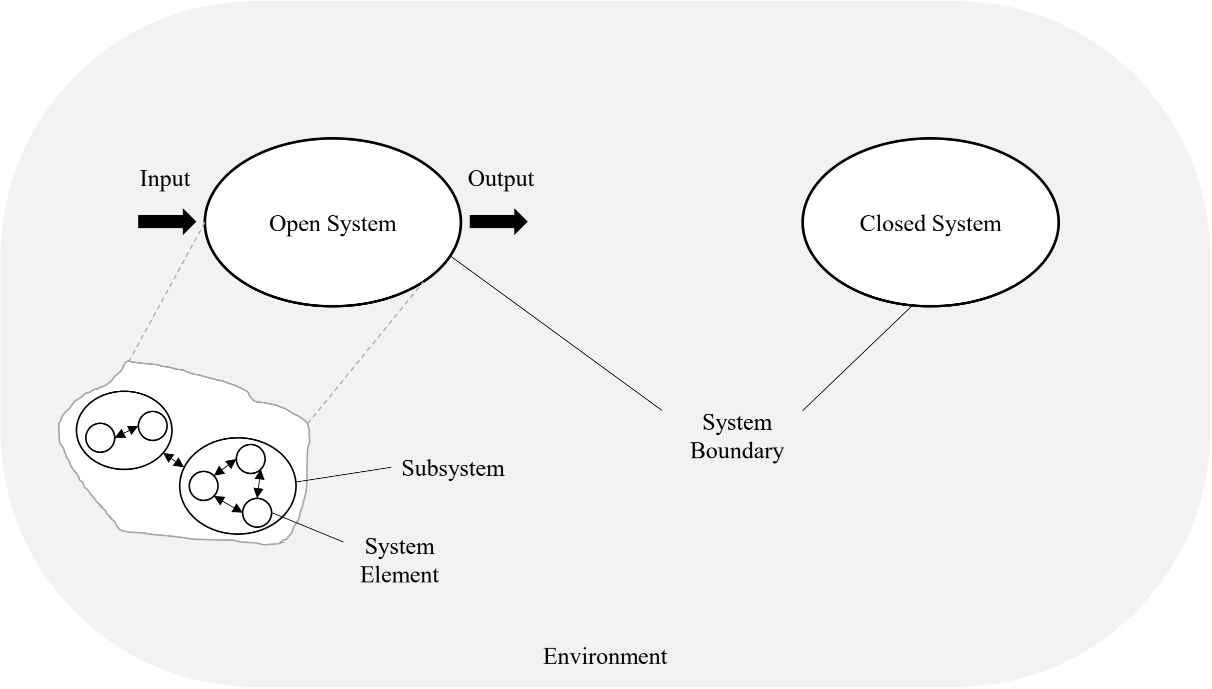


Figure 2.1: Characteristics of open and closed systems. Adapted from Wiendahl et al. (2015).

A typical transformation system comprises three components: inputs, transformation processes and outputs. As applied to production systems, this means that desired outputs – such as goods and services – are transformed through an activity or function from a set of inputs – such as land, labor, capital, raw materials and resources (Dangelmaier, 2009). The overall goal of this transformation process is to create and add value. The value of the outputs should thus exceed the value of the inputs (Reid & Sanders, 2005). The major challenge is to manage these complex processes successfully. All associated activities can be summarized under the term operations management. In operations management, a strong focus is placed on the efficiency and effectiveness of value-added processes to generate goods and services (Schönsleben, 2011). Additionally, according to Nyhuis and Wiendahl (2009), these objectives can be further specified and therefore they invented a target system. Regarding the objectives, both internal (company) and external (customers') viewpoints exist.

As illustrated in Figure 2.2, on the market side the logistic performance is mainly evaluated based on throughput times and schedule reliability. For this reason, the aims within a company are to achieve short throughput times and high schedule reliability. The company will attempt to keep its logistic costs as low as possible through high capacity utilization and low work-in-process (WIP) levels. Given their importance for the company's success as well as the evaluation of PPC methods (such as those proposed in this thesis), the logistic objectives are explained below in further detail. The overall objective is to improve efficiency, i.e. particularly to keep the unit costs as low as possible.

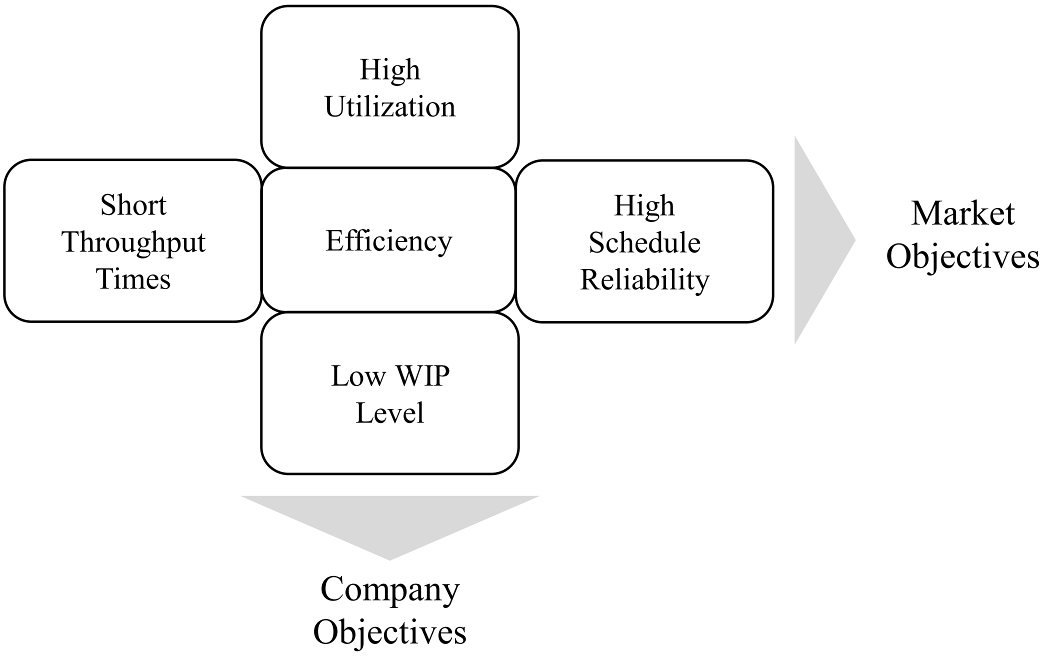


Figure 2.2: Four logistic objectives in the target system of Nyhuis and Wiendahl (2009).

Schedule Reliability

Schedule reliability is defined as the percentage of jobs that are completed within a predefined due date tolerance. High schedule reliability prevents companies from incurring extraordinary expenses for express deliveries and avoids large downtime costs for the customer (Nyhuis & Wiendahl, 2009).

Short Throughput Times

Throughput time is the period it takes to complete a job. This includes not only the completion process itself but also transportation and waiting times. Because a growing number of customers require quick deliveries, companies today aim for short throughput times (Nyhuis & Wiendahl, 2009).

High Utilization

Utilization refers to the probability that a resource will run empty due to a lack of jobs in the system. A highly utilized system is associated with increased profit for the company. This is because unit costs decrease as the total costs are absorbed across many products (Nyhuis & Wiendahl, 2009).

WIP Level

Broadly, the WIP level is defined as the number of jobs in the system at a specific point in time. A high level of WIP ties up capital and valuable storage space, which means that companies will aim to reduce their WIP (Nyhuis & Wiendahl, 2009).

In practice, it is inevitable that conflicts arise between the four objectives. For example, high-capacity utilization is generally achieved only by high WIP levels (Hopp & Spearman, 2011). However, this also leads to high throughput times and thus impairs the schedule reliability. For companies, the consequence is that there is a contradiction in the simultaneous pursuit of all logistic objectives. Therefore, prioritization is required. A possible solution to this problem comes in the form of logistic operating curves. A detailed description of this concept appears in Nyhuis and Wiendahl (2012).

In the past, the focus was mainly on high levels of shop utilization to avoid the high costs of resource downtime. Due to the transformation from a seller's market to a buyer's market, companies have become more focused on customers' demands. To enable these demands to be met, the focus has shifted toward short throughput times and high schedule reliability (Lödding, 2013; Wiendahl et al., 2015). The described transformation and the associated shift toward market-oriented objectives have led to increased requirements and challenges for the planning and control of production processes (Kurbel, 2016). Various PPC principles are described in detail in the following section.

2.2 Basic Principles of PPC

This section discusses the principles of planning and control in production systems. According to the length of the planning horizon, a distinction is made between strategic planning (about five years), tactical planning (one to five years) and operational planning (up to one year) for production. The primary task of strategic production planning is to define long-term strategies to promote the development of competitive production. The practical implementation of the strategies occurs during medium-term tactical production planning through restructuring the internal infrastructure. This concerns particularly the layout and material flow planning, product design and increased utilization of resource capacities. In line with the strategic and tactical production planning, the focus of operational production planning is to ensure the efficient use of existing resource capacities in the short term (Dangelmaier, 2009; Dyckhoff & Spengler, 2010). Operational production planning is also referred to as PPC (Dangelmaier, 2009; Kurbel, 2016) and is therefore further discussed. In the following text, the terms order and job are used synonymously, an approach also used by Lödding (2013).

One of the first definitions of PPC in the German literature was provided by Hackstein (1989). Illustrated in Figure 2.3, production planning is understood as a multistage process, which is subdivided into the following four tasks: production programme planning, quantity planning, due dates and capacity planning. Within production *programme planning*, the product type and quantity are determined. Based on these results, in the *quantity planning* the dependent requirements (such as raw materials, semi-finished products and sub-assemblies) are determined. In the final stage, *scheduling*, the due dates for jobs are determined, and it is ensured that all necessary resources are provided in the quantities demanded.

Once these tasks are complete, the production control is responsible for the release, control and monitoring of jobs. This stage is thus subdivided into order release and order monitoring. Finally, during data management an interface is created for communication between the planning and control aspects of production systems. As evident from the above description, the individual PPC tasks are processed in the order they are listed. Therefore, the results of one planning stage serve as input for the subsequent stage. The reference model by Hackstein is thus characterized by a successive, hierarchical character while the different tasks are performed.

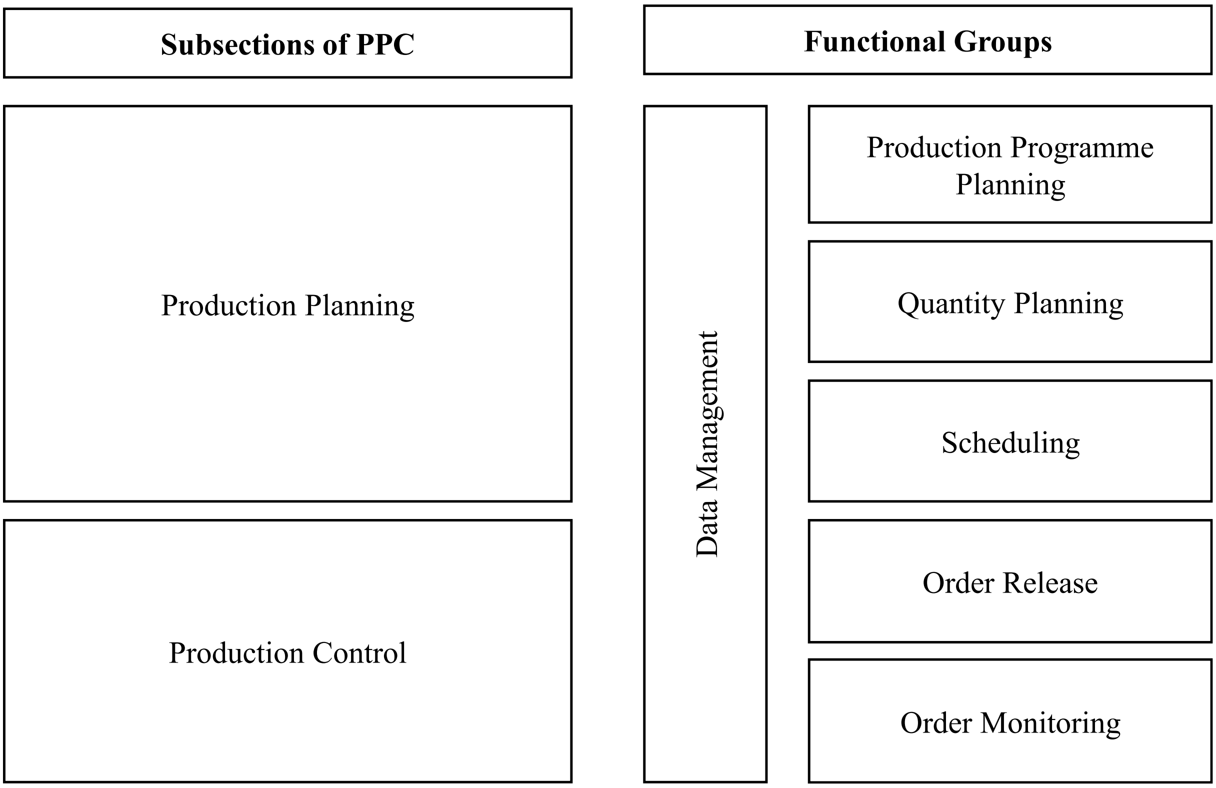


Figure 2.3: PPC model by Hackstein (1989).

According to Schuh and Stich (2012), the reference model of Hackstein does not fully reflect the logistic processes that occur in reality. For this reason, it was extended to the Aachen PPC model. The successive approach remains the same, but the hierarchical character is replaced by a cross-linked structure. A distinction is made between intercompany network tasks and key tasks or cross-functional tasks that occur within one company. Further explanation of the network tasks is not provided here, since the focus of this research project is an analysis of the material flow in a company. Further details can be found in Schuh and Stich (2012).

Similar to the previous PPC model, the first stage of the key tasks represents the *production programme planning* (see Figure 2.4), whereas the *production requirement planning* corresponds to the quantity planning. As shown in Figure 2.4, the main difference is the distinction between *external* and *in-house PPC*. A main reason for this are the different requirements for the planning tasks. The external PPC assumes tasks such as selecting suppliers or evaluating tenders. In contrast, the in-house PPC deals with internal procedures and thus allocates tasks such as fine-tuning, sequence scheduling, order release and order monitoring. Scheduling tasks and allocation to resources and the sequencing of jobs (see Pinedo 2014) fall under in-house PPC.

Cross-functional tasks enhance the integration and optimization of the PPC in all production areas. This includes order coordination, storage and PPC monitoring. Similar to the reference model by Hackstein, data management is viewed as an integrated solution to support all PPC activities.

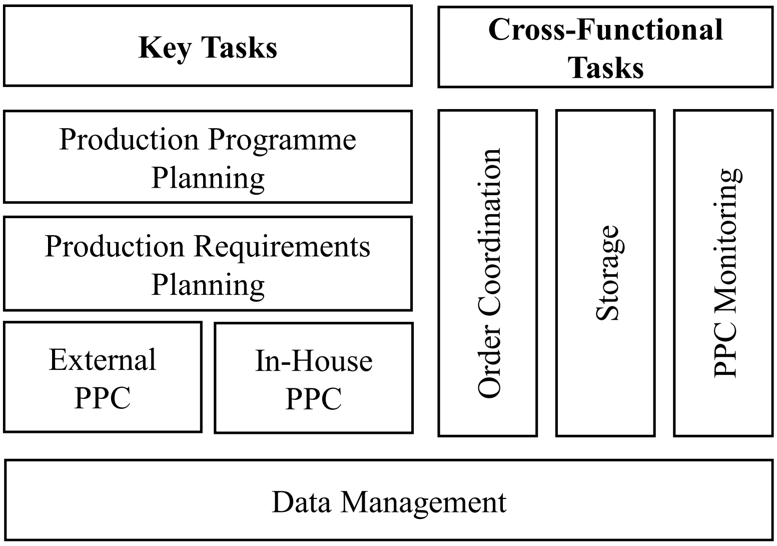


Figure 2.4: Aachen PPC model without network tasks. Adapted from Lödding (2013).

Lödding (2016) indicated that an emphasis must be placed on the in-house PPC in terms of achieving the desired logistic objectives. Because this was not the case in the Aachen PPC model, he developed a new model for production control, which he referred to as manufacturing control. As shown in Figure 2.5, there are four tasks in the new model to influence the logistic objectives: order generation, order release, order sequencing and capacity control.

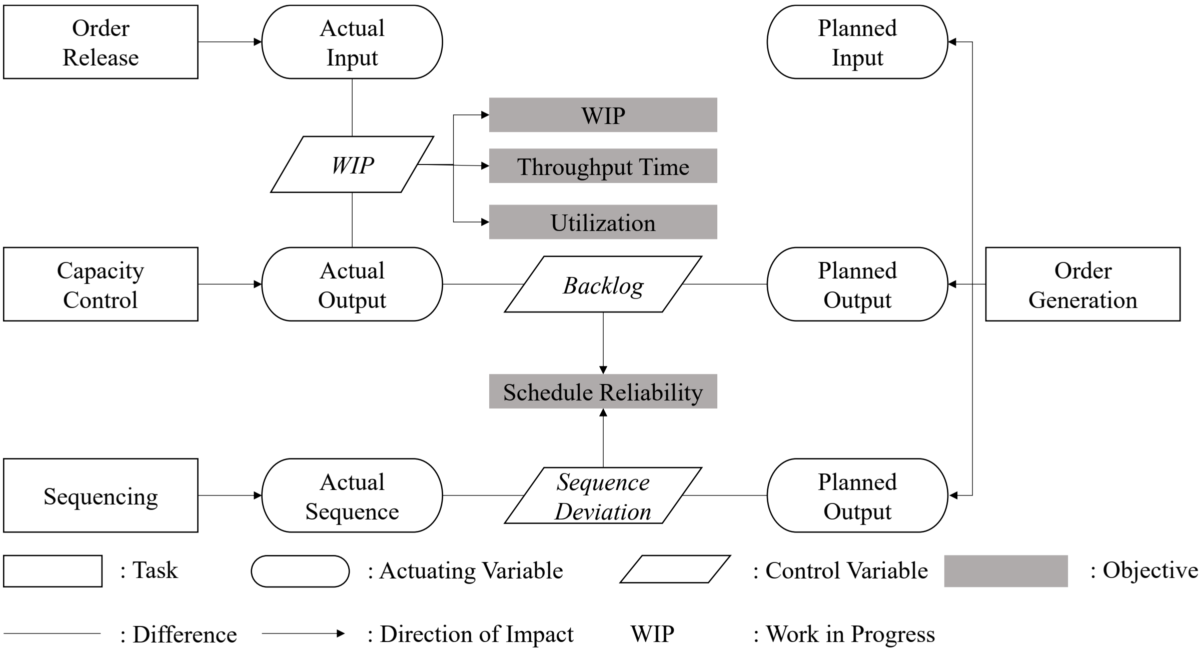


Figure 2.5: A manufacturing control model. The connections and dependencies between actuating variables, control variables and logistic objectives are illustrated. Adapted from Lödding (2013).

Strictly speaking, the *order generation* is part of production planning. It defines the planned input and output of jobs in the production system. Furthermore, the planned sequence of jobs is determined here, and the remaining tasks are assigned to the production control (Lödding, 2013).

The *order release* specifies the precise point in time at which the processing of a job order can begin. The main aim is to influence the WIP by releasing jobs once the WIP falls below a specified threshold. In this way, the system utilization and throughput times are impacted. A high WIP level increases the utilization and simultaneously results in faster throughput times. There exist several control policies to keep the WIP level low on the shop floor, with pull systems being distinguished from push systems. In many cases, pull systems require less WIP (Lödding, 2013) and can be more easily controlled than push systems (Hopp & Spearman, 2004). The two most widely known pull systems, Kanban and constant work-in-process (ConWIP), are briefly presented. (Further details, proposed modifications and alternative WIP control policies appear in Lödding, 2013). In Kanban, several card sets are used to control each pair of work systems, with a card being associated with (for example) a particular part type. After the completion, the card returns to the beginning and new cards are released. The process allows close control of the WIP. In this simplest form, a Kanban system is especially suited for production environments with high volumes and low variants (Hopp & Spearman, 2004). By contrast, the aim of ConWIP is to keep the WIP at a constant level anywhere in the system (Spearman, Woodruff, & Hopp, 2022). In ConWIP, only one set of cards is assumed, where the number of cards corresponds to the desired WIP level. This simplifies the implementation. Because the cards are not associated with a particular part type, ConWIP is suitable for production environments with low volumes and high variants (Hopp & Spearman, 2004).

The *order sequencing* determines which of the jobs in the waiting queue will be processed next on a particular resource. For this purpose, priority numbers are assigned to individual jobs according to the set logistic objective. The jobs with the highest priority are first in the queue and are processed first. The impact of order sequencing is significant in production systems with high WIP levels, since it leads to a high number of jobs in the queue and multiple sequence options. A match between the actual sequence and the planned sequence results in high schedule reliability (Lödding, 2013). The range of possible sequencing rules is explained in the following set of standards, which have been used for many years in both theory and practice (e.g. Blackstone, Phillips, & Hogg, 1982; Haupt, 1989; Sels, Gheysen, & Vanhoucke, 2012):

- First Come First Served (FCFS): select the job that is added first to the queue. Owing to its simplicity, this rule is fairly popular and is often used as a benchmark for other rules.
- Shortest Processing Time (SPT): select the job with the shortest processing time in the queue.
- Least Number of Operations (LNOP): select the job with the least remaining number of operations in the queue.
- Total Shortest Processing Time (TSPT): select the job with the least total processing time in the queue.
- Earliest Due Date (EDD): select the job with the earliest due date in the queue.
- Longest Remaining Processing Time (LRPT): select the job with the longest remaining processing time.

Some of these rules define the value of priority when jobs entering the system, such as EDD and TSPT. Others update the priority value after each processing step, such as LNOP, LRPT and TSPT. No one priority rule is effective for all cases. The effectiveness of rules essentially depends on the assumptions made regarding the given shop conditions and which specific performance indicators are considered (Holthaus, 1996).

As part of *capacity control*, the capacity of work systems (e.g. working hours for each resource) is adapted to the real-life conditions of the production system. The focus here is not only on determining what resources are necessary but also that they are reliably available at the required time (Lödding, 2013).

When selecting suitable production control methods, the planer must consider the structure of the production system and the jobs as well as company goals and customer requirements (Wiendahl et al., 2015). These characteristics are partly summarized under the terms complexity and dynamics and are explained in further detail in the next section.

2.3 Complexity and Dynamics in Production Systems

The presence of many influencing variables and strongly interconnected elements results in systems that have high complexity. Such complexity means that these systems tend to be difficult to describe and handle (Magee & de Weck, 2004). However, in the context of production systems, there is a more concise definition of complexity, which is used in English literature and is consistent with the system theoretical perspective. Depending on the area of application, a distinction can be made between functional and physical domain complexity (see Figure 2.6). Functional domain complexity is defined as “a measure of uncertainty in achieving the functional requirements” (Efthymiou, Mourtzis, Pagoropoulos, Papakostas, & Chryssolouris, 2016). The functional domain deals mostly with the design of production systems and is therefore not part of this thesis. More detailed descriptions of this type of complexity can be found in Efthymiou et al. (2016) and ElMaraghy, ElMaraghy, Tomiyama, & Monostori (2012).

The relevant definition of complexity in this thesis is from the physical domain, as it focuses on both the products and the production process. Figure 2.6 illustrates that a distinction is made between structural and operational complexity. *Static* or *structural complexity* depends on the number of product variants or the product mix, current processes and the structure of the respective system (ElMaraghy et al., 2012). Thus, structural complexity is time-independent as it includes all the static characteristics of a system. For example, the degree of structural complexity increases with an increasing number of system elements (e.g. machines and people) and/or the interconnections between those elements.

The number of interconnections between system elements, or in other words the material flow, is strongly related to the number of product variants (Lödding, 2013). According to Lödding (2013), an increasing number of variants can lead to production processes in which there are several predecessors and/or successors. In addition, an increasing number of variants can be associated with increasing material backflows. Both these aspects mean that the material flow is becoming more complex. In recent decades, the diversity of variants has risen due to the transformation from a seller’s to a buyer’s market and the related trend toward customer-specific products. With the high number of variants, the tasks of PPC are becoming notably complex and challenging, because even with a high number of variants, the aim is to maintain a low level of material flow complexity (Lödding, 2013).

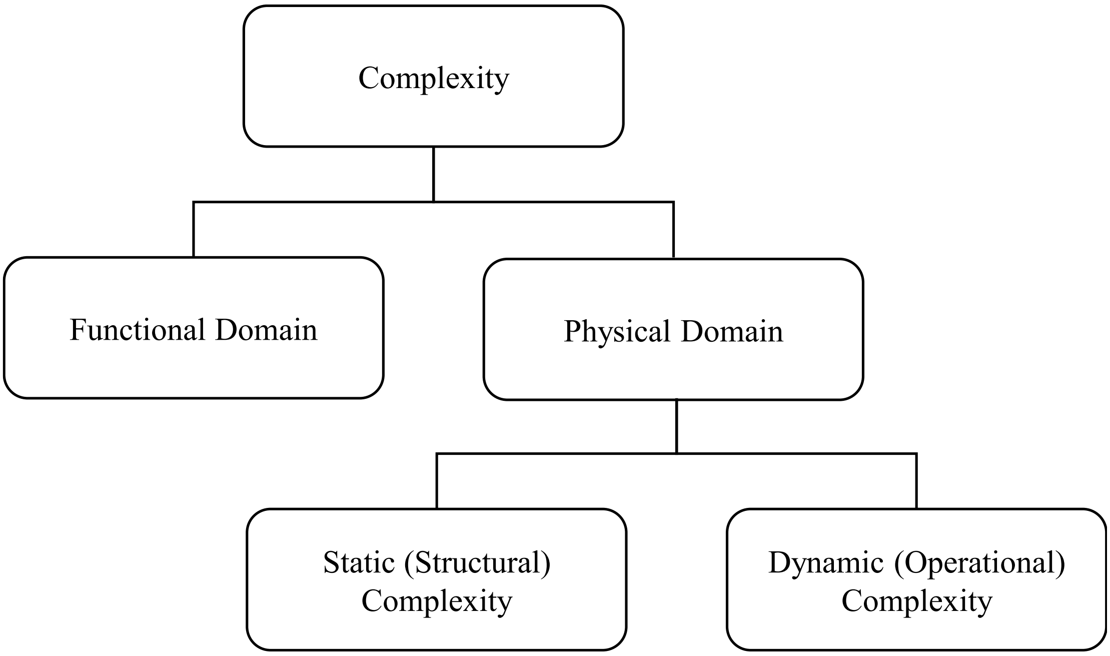

Complexity
Functional Domain
Physical Domain
Static (Structural) Complexity
Dynamic (Operational) Complexity

Figure 2.6: Different forms of complexity distinguished by domains. Adapted from ElMaraghy et al. (2012).

Dynamic complexity, also termed *operational complexity*, relates to the operational behavior of a system (Efthymiou et al., 2016). More precisely, the term refers to the frequency and intensity of system changes and environmental changes. Dynamic complexity is time-dependent (ElMaraghy et al., 2012), and unpredictable events – like machine breakdowns or variation in process times – increase the dynamic complexity. Equally, machine locations and process steps can change from time to time. The aim of PPC is to keep the dynamics of all predictable and unpredictable events under control (Efthymiou et al., 2016; Wiendahl & Scholtissek, 1994).

In this thesis, the term dynamic meets the definition of the dynamic complexity. It is irrelevant whether the root of dynamics is located inside or outside the system. Furthermore, the term complexity relates to static complexity. In the literature, there are numerous approaches to assess the complexity and dynamics of a system. This is achieved, for example, by entropy (Frizelle & Woodcock, 1995), individual indices (e.g. Schoettl, Paefgen, & Lindemann, 2014; Modrak & Soltysova, 2017) or even entire models (e.g. Philipp et al., 2006) that take account of several criteria. ElMaraghy et al. (2012) quantified the complexity using graph theory. Alkan et al. (2018) provided a literature review and reflections on the assessment of complexity in manufacturing systems.

According to ElMaraghy, Kuzgunkaya, and Urbanic (2005), an advantage of highly complex and dynamic production systems is that they can react quickly to changing requirements. This is possible because the interconnection patterns between the individual system elements are not static but can adjust to a new situation. In case of a machine breakdown, usually a similar machine can perform the requested process step. This is important in terms of increasing competitive pressure, because a system that has the ability to adjust to its environment by constantly generating new patterns enables a company remaining competitive. However, this flexibility is often related to higher costs. Isik (2011) described other performance-related disadvantages, such as customer dissatisfaction resulting from delays in deliveries or the rapid reduction of inventory levels that could result from a higher degree of dynamic and structural complexity. Hence, as previously mentioned, one of the challenges in PPC is to manage these complexity dimensions. There are several promising approaches, which are described in detail in Section 2.4.

In connection with the planning and control of production processes, there is a further type of complexity, namely computational complexity. Computational complexity is closely related to the evaluation of an algorithm for solving problems in various areas, such as scheduling problems (Deshmukh, Talavage, & Barash, 1998). Problems of this kind are normally NP-hard (Garey & Johnson, 1979; Hoogeveen, Lenstra, & Veltman, 1996). The fundamental difficulty can be traced to the fact that the computational time to find an optimal solution increases with an increasing problem size, more than any polynomial function. In the context of production, computational complexity mainly results from the number of system elements (e.g. individual machines, subsystems or jobs). With a high number of elements, an algorithm might no longer be able to solve the problem in a reasonable period (ElMaraghy et al., 2012). Computational complexity is thus closely related to structural complexity.

The most effective way to deal with computational complexity, according to ElMaraghy et al. (2012), is by reducing the number of elements. However, this is not always possible, since in some cases the availability of special resources is required for specific processing steps. Therefore, rather than removing individual system elements, it is possible to group several of them into manageable units based on the Divide & Conquer principle. In this approach, a problem is partitioned into small sub-problems that can be solved independently and in a reasonable computing time. To enable this method, the units must be autonomous and independent and must not interact with each other – more precisely, there must be no material flow between them (ElMaraghy et al., 2012).

A further possibility instead of searching for an optimal solution is to use heuristic algorithms. They offer solutions in a reasonable computing time, and which are close to the global optimum. There are numerous descriptions and examples, from simple to complex heuristic algorithms for solving scheduling problems. An extensive overview can be found in (Domschke, Scholl, & Voß, 1997). The following section describes how to deal with increasing complexity and dynamics in PPC.

2.4 Centralized, Decentralized and Hybrid PPC

In conventional production systems, software systems such as Enterprise Resource Planning (ERP) systems are often used to support PPC (Kurbel, 2016). The aim is to create an overall production plan for all areas. An often noted weak point of these software solutions is that they follow the successive planning approach, which was described earlier in connection with the various PPC models (Section 2.2). This means the results of a superior planning level are passed on to subsequent levels. It is possible, for example, that poor results of a particular planning level will be passed to the next level, or that short-term changes (like rush jobs) are not considered directly. These are all valid reasons for the creation of a new production plan. However, this is often omitted due to the time needed. Instead, an attempt is made to consider the dynamic changes at the shop floor level, for example by employees, or even in the subsequent planning period. Changes in a system are thus interpreted as disturbances, which are associated with economic losses or loss of quality. Assuming a more static production environment, a high degree of planning accuracy and reliability can be achieved through this centralized approach. Hence, from a theoretical perspective the centralized planning and control of production systems seems a reasonable approach. Nevertheless, in practical terms, this approach can only be used on a limited basis for real-world applications, as the plans often do not adequately reflect the actual production situation (Kim & Duffie, 2004; Spath, 2013; Windt, Philipp, Böse, & Becker, 2010b).

By contrast, decentralized systems can take into account the growing complexity and dynamics of current production systems, such as unforeseen changes or environmental influences. In autonomous systems, the responsibilities for decision-making are spread over several elements that are interconnected with each other over a real-time communication system (Freitag et al., 2004). This enables the elements to communicate and cooperate. A precondition for this approach is the use of ICT that enables real-time monitoring of shop floor processes. Changes can be detected directly, and based on this information, all possible alternatives can be identified and evaluated (Fottner et al., 2021). Thus, in contrast to the centralized approach, all activities related to information processing, decision-making and the execution of decisions are shifted to the shop floor level from the start (Bouhai & Saleh, 2017). Autonomous systems are characterized by short decision-making processes and low coordination effort, which means they are able to react quickly and flexibly (Thomas, Trentesaux, & Valckenaers, 2012).

Previous studies, especially within the Collaborative Research Centre 637 (CRC 637) "Autonomous Cooperating Logistic Processes – A Paradigm Shift and its Limitations" (the main findings are summarized in Schukraft, Teucke, Freitag, & Scholz-Reiter, 2021), support the assumption that in complex and dynamic production environments decentralized decision-making approaches provide as good or even better performance compared to selected centralized approaches (e.g. Scholz-Reiter, Rekersbrink, & Görges, 2010; Windt, Becker, Jeken, & Gelessus, 2010c; Windt, Becker, & Kolev, 2010a).

According to Becker et al. (2015), decentralized control systems are characterized by the following three elements (compare Figure 2.7), which are simultaneously the core research fields: technology, control algorithms and topology. *Technology* is important as it enables real-time control of the production processes. In addition to permanent data availability, data transparency and rapid data collection and processing are crucial. With the decision of a company to dispense with a central control unit, e.g. work systems, workpieces and products must become intelligent so that they are able to solve certain tasks autonomously (Kurbel, 2016). This is part of Industry 4.0 (Kagermann et al., 2013).

Cyber-physical systems are fundamental components of Industry 4.0 (Monostori et al., 2016). They are characterized by the combination of sensors, actors and intelligently embedded software systems. The integrated sensors are used to permanently collect data. The gathered data is evaluated and analyzed using the software. After this, the software sends control commands to the actuators. A detailed description of actuator and sensor technologies appears in Fottner et al. (2021). In generally, the research field of technology includes, in particular, the development of hardware and software to enable the communication between individual system elements and provide them with the ability to make decisions autonomously.

In autonomous systems, decisions are made according to the logic of *control algorithms*. The availability of real-time data gained from advanced technologies has enabled new control algorithms to be developed in recent years (Lasi et al., 2014). To make the right decisions, autonomous elements need reliable (real-time) information. Windt et al. (2010c) presented a selection of well-known control algorithms for autonomous decision-making. They identified seven dimensions to give a more specific classification of the algorithms. These included temporal data (referring to past or future data or both), the number of planning steps, the use of artificial values, the data scope and the type of data storage. According to Scholz-Reiter et al. (2010), the algorithms generally fall into three groups: bounded rational algorithms, rational algorithms or a combination of both. Many bounded rational algorithms are based on the adaptation of principles from biology, where all data needed for decision-making come from past events (e.g. ant algorithm proposed by Cicirello & Smith, 2001). Rational algorithms, by contrast, use data that represent potential future system states – such as queue length estimator (QLE) by Scholz-Reiter, Beer, Freitag, and Jagalski (2008). An example of a combination of bounded rational and rational algorithms is one logistics target per rule (OLTPR) mentioned in Windt et al. (2010c).

No autonomous algorithm is dominant in all cases. Rather, increases in performance depend on the particular assumptions made regarding the shop floor scenario or the considered evaluation criteria. To develop such a method, all available data (e.g. about the system state) would have to be provided to make decisions based on them. This requirement would contradict the idea of autonomous systems.

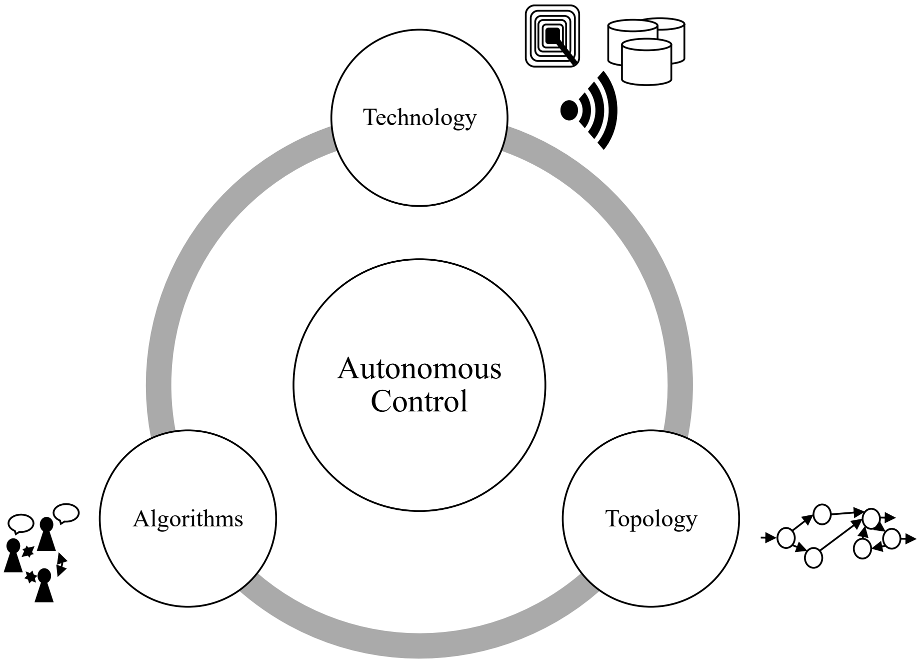


Figure 2.7: Three main elements of autonomous control according to Becker et al. (2015).

The term *topology* reflects the structure of the underlying material flow system, or more precisely the material flow pattern. At the shop floor level, topology describes the interconnection patterns of work systems that are linked to each other by material flows (Becker, Meyer, & Windt, 2014). Of all three research fields, the topology of a production system is the least researched (Becker et al., 2015), although Becker et al. (2012) showed there was a positive correlation between specific topological patterns and logistic performance. Vrabič et al. (2012) and Becker et al. (2015) indicated that if companies want to switch to autonomous control, they need to concentrate on whether the topology of the underlying material flow system enhances the practical implementation. Mourtzis and Doukas (2012) suggested that it seems reasonable to merge the individual system elements into two or more modules with their own decision-making competencies to reduce complexity. For better coordination, the elements within modules should interact with each other. However, it is not specified what criteria should be used for the modularization.

The idea to create autonomous units is not entirely new and is widely known as the concept of *Cellular Manufacturing*. In this approach, parts with a similar sequence of operations are grouped into part families. Subsequently, work systems that are required to process a specific part family are spatially grouped into manufacturing cells. The responsibility for technical and organizational interventions in these cells is assigned to the respective staff (Wiendahl et al., 2015). Decentralization is thus primarily defined by the operational decision-making framework of the staff. The advantages of such an organizational structure are (among other things) shorter order throughput times and setup times due to the spatial proximity of work systems required for a certain part family (Burbidge, 1975).

In Cellular Manufacturing, a static production environment is usually assumed (Saxena & Jain, 2011). However, today's production systems are increasingly exposed to dynamic influences, which are reflected in high fluctuations in demand and increasing numbers of variants. These changes require more frequent identification of new part families as well as a spatial regrouping of work systems. This is not only time-consuming and costly, but it can also happen that the regrouping required for new part families may not be feasible in practice (Hamedi, Esmaeilian, Ismail, & Ariffin, 2012). Additionally, it is possible for a specific machine type to be needed in several manufacturing cells simultaneously, so that the additional machine procurement leads to high investment requirement. To meet these difficulties, the concept was extended to include *Virtual Cellular Manufacturing*. In Virtual Cellular Manufacturing, work systems are summarized not spatially but logically, so that the procurement of redundant machines can be avoided (Khilwani, Ulutas, Islier, & Tiwari, 2011). This approach makes it possible to react more flexibly to dynamic influences, because the manufacturing cells are only virtual and thus can be regrouped easily.

In the context of Virtual Cellular Manufacturing, a distinction can be made between two different research areas (Hamedi et al., 2012). The focus of the first area is on comparing the performance of Virtual Cellular Manufacturing with other existing concepts. The second research area deals with the characterization of the actual Virtual Cellular Manufacturing. These include, for example, the formation of part families and manufacturing cells as well as the scheduling of jobs within these cells. The job processing order set this way will last until the next change in the production system (Tanchoco, 1999) and is therefore not adjusted after each operation, as would be the case using corresponding autonomous control algorithms. Therefore, this approach is a product-oriented solution that does not reflect the control-oriented perspective. Thus, while virtual cellular manufacturing is an interesting approach due to the logical combination of work systems, it is not sufficient for a dynamic production environment because of its insufficient utilization of autonomous decision-making.

The idea of virtual manufacturing cells was pursued and extended with aspects of autonomous control by Vrabič et al. (2012). They also demonstrated that topology is a crucial factor when applying autonomous control. In this context, they analyzed the intensive exchange of materials, and thus also of information, between the individual system entities. For this purpose, they transferred the material flow within a manufacturing system into a network representation (see Section 3.2 for details). According to Vrabič et al. (2012), this network representation has the decisive advantage of providing an objective evaluation of material and information flows between the individual work systems.

This new perspective differs from the concept of Cellular Manufacturing, in which work systems are grouped according to part families. Furthermore, the network representation offers a wide range of methods from complex network theory. A non-trivial topological characteristic of complex networks is the presence of so-called clusters (Fortunato, 2010). As described in

Section 3.4, there is more than one definition of clusters. However, as illustrated in Figure 2.8, all definitions have in common that the elements within a cluster have more connections within the cluster than they do with elements outside the cluster (Newman & Girvan, 2004). Vrabič et al. (2012) evaluated the hypothesis that such clusters are generally suited to perform as units of autonomous work systems, given that most material and information flows stay within a cluster. Essentially, the identification of such clusters comes close to the idea of Mourtzis and Doukas (2012) to merge the individual system elements based on their interaction into at least two or more modules with their own decision-making competencies. Indeed, Vrabič et al. (2012) showed that the identified clusters were associated with specific work processes, so that in principle they operated as autonomous units. Among other things, they demonstrated that work systems with similar functionality did not have to belong to the same cluster, as they might not exchange any material or information with each other (Vrabič et al., 2012).

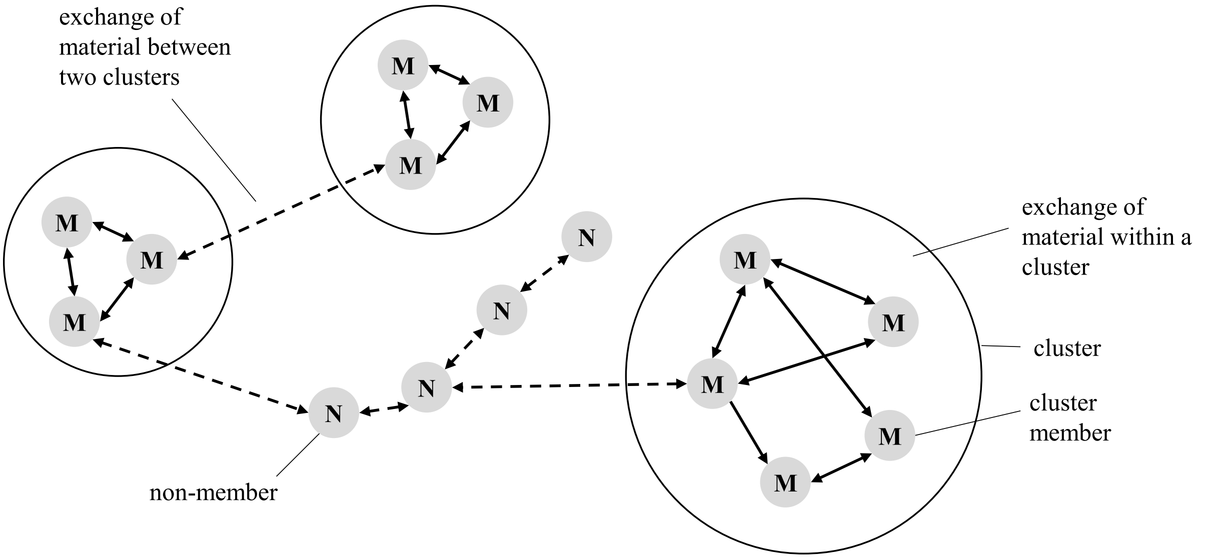

exchange of material between two clusters
M
M
M
M
M
M
N
N
N
N
exchange of material within a cluster
M
M
M
M
M
cluster
cluster member
non-member

Figure 2.8: Autonomously acting work systems merged into units with their own decision-making competencies. Vrabič et al. (2012) called such units clusters. Figure modified from Becker et al. (2015).

Despite the many advantages, in highly complex systems the use of autonomous control only makes sense up to a point; from then onward the achievement of logistic performance drops again (see Figure 2.9). The absence of a coordinating unit means that a high level of autonomy sometimes leads to confusion in the decision-making process (Philipp et al., 2006). Therefore, there exists an optimal degree of autonomous control. To support the implementation of autonomous control in manufacturing systems, Böse and Windt (2007) proposed a catalog of criteria to determine the degree of autonomy in a given manufacturing environment. The criteria can also be used to compare different systems. This approach does not necessarily support the specification of the optimal degree of autonomous control in advance, and there is generally a lack of concrete approaches to determine the optimal degree.

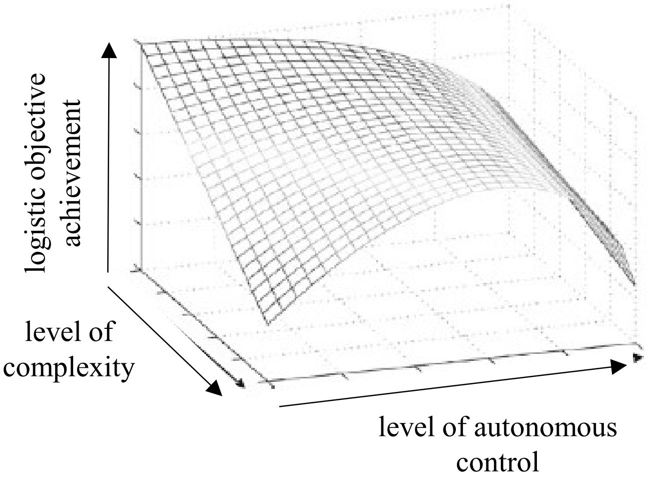


Figure 2.9: Limitations of autonomous control. The achievement of logistic objectives mainly depends on the degree of autonomy and the complexity level of a given production scenario. Adopted from Philipp et al. (2006).

Hybrid solutions combine the advantages of both approaches, namely, the planning reliability of central production planning and the ability of autonomous control to respond quickly and flexibly to discrepancies (Trentesaux, 2009). The more detailed the activities of the coordinating unit, the smaller the decision-making possibilities of the autonomous part. The challenge here is mainly to select suitable planning and control methods and combine them in a harmonized way to increase the overall performance (e.g. Blunck, Armbruster, Bendul, & Hütt, 2018). In the research project "Methods for the interlinking of central planning and autonomous control in production" (reference number SCHO 540/26-1) by the German Research Foundation (DFG) (see, for example, Grundstein, Schukraft, Freitag, & Scholz-Reiter, 2015; Schukraft, Grundstein, Freitag, & Scholz-Reiter, 2015; Schukraft, Grundstein, Freitag, & Scholz-Reiter, 2016) and in Mehrsai et al. (2017) a hybrid approach was proposed. The researcher split the scheduling-related decisions into central work system allocation and autonomous sequencing of jobs. This meant that work systems decided independently about the sequence of jobs based on real-time data from the shop floor. The results of this hybrid approach within a flexible job shop environment are promising. In previous works, the same sequencing rule was used for all work systems. Due to the local behavior of such rules, the possibilities for coordination between the individual system elements are limited.

By contrast, Miyashita (2000) merged work systems according to their individual utilization rate into several units and assigned different rules to them to improve the logistics performance. Such rule combination resulted in good performance. Although this was not done within the context of autonomous control, it still seems a promising approach. Pickardt, Hildebrandt, Branke, Heger, and Scholz-Reiter (2013) stated that merging could be challenging if the utilization of the individual work systems varies significantly. This problem could be avoided by applying the idea of Vrabič et al. (2012), where work systems are merged according to their exchange of material and information flow, to create a set of autonomous clusters with their own decision-making competencies. This scenario enables these autonomous clusters to individually select a rule from a set of sequencing rules to increase the overall performance.

Rules are chosen according to the material flow intensity within an autonomous cluster. Consequently, this thesis combines the hybrid approach and the clustering-based approach proposed by Vrabič et al. (2012). The conceptual framework of such a hybrid clustering-based approach is illustrated in Figure 2.10, and the approach is described further in Section 6.3.

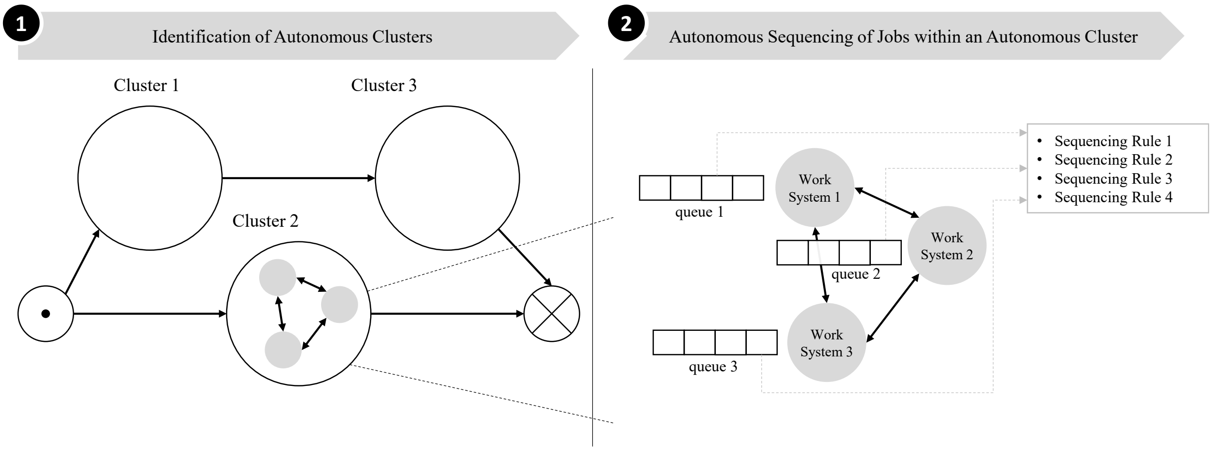


Figure 2.10: Conceptual framework of hybrid clustering-based approach. (1) The network representation of a material flow system provides global system information, which is used to merge the individual work systems to autonomous clusters. (2) Material is exchanged between clusters to some degree, but most material exchange occurs within a cluster. Sequencing rules use information that is local to the corresponding cluster.

2.5 Summary

To sum up, there is a need for new planning and control methods to deal with the changing manufacturing environment. In particular, the autonomous sequencing of jobs as part of a hybrid approach has proved promising. This approach can directly respond to newly occurring prioritizations without triggering a time-consuming adjustment. To reduce complexity, it seems promising to perform decision-making situation-based and outsource it to individual clusters. Such clusters are identified using the topology of a given material flow system. After the conceptual framework of hybrid clustering-based approach has been outlined as a basis for further steps, the next chapter presents the basics of complex networks, with a focus on identifying and evaluating clusters.

3 Clustering in Complex Networks

The theory of complex networks has gained attention in recent decades, mainly because of the technological progress in data collection and processing (Reichardt, 2009; Takemoto & Oosawa, 2012). A major benefit of network models is their representation of patterns of connections between the system elements. Networks are a powerful tool to simplify real-world complex systems and reduce them to the essentials for the user (Takemoto & Oosawa, 2012). Technological networks (e.g. the Internet, transportation networks and power grids), biological networks (e.g. metabolic systems, proteins and neural networks) and social networks (e.g. friendship networks or collaboration among scientists) represent concrete examples in which complex networks are used as a methodology for the description and modeling of complex systems (Newman, 2010).

Recently, complex networks have been used in manufacturing. An example is the study of structural characteristics of material flow systems to deduce appropriate consequences for decision-makers (e.g. Becker et al., 2014; Omar, Minoufekr, & Plapper, 2018; Vrabič et al., 2013).

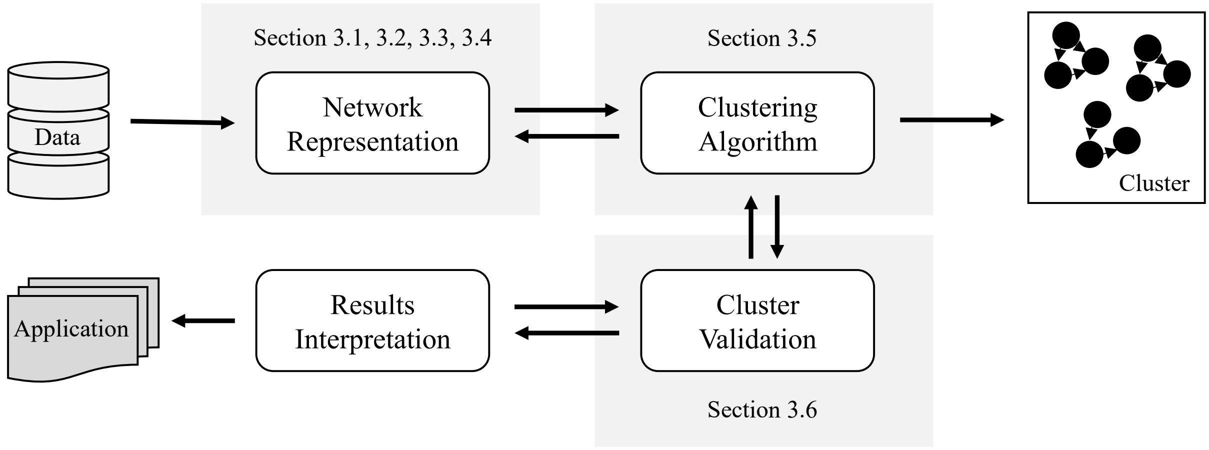


Figure 3.1: Four crucial steps in the clustering process. Modified from Xu & Wunsch (2005).

The structure of this chapter is based on the process shown in Figure 3.1. The first section (3.1) presents the relevant terms and concepts from complex network theory. In Section 3.2, the transformation of job routing data into a network representation is described in detail. The next section (3.3) introduces two structural properties that can be found in real-world networks. A detailed definition of the term cluster is given in Section 3.4. The last two sections provide an overview of classification options of well-known clustering algorithms (Section 3.5) and evaluation and validation possibilities for the resulting clusters (Section 3.6). This is followed by a brief summary in Section 3.7. This chapter is intended to give an impression of the manifold possibilities and should not be considered exhaustive.

3.1 Basic Notions in Graph Theory

Complex networks can be formally described using graph theory. Graphs and networks help scientists to understand and study complex systems, and the two terms are used interchangeably in this thesis. The necessary theoretical knowledge is discussed below. The explanations are limited to terms and concepts that are essential for this thesis. Detailed descriptions of graph theory and complex networks theory appear in Albert and Barabási (2002), Newman (2003), Boccaletti, Latora, Moreno, Chavez, and Hwang (2006) and Diestel (2017).

Graph Definitions Formally, a graph $G = (V, E)$ comprises a set $V$of $|V| = N$ vertices (also called nodes) and a set $E \subseteq V \times V$ of $|E| = M$ edges (also called links). It is possible to extend the definition of a graph regarding the edge direction and edge weight. Where edges have a direction, represented graphically by arrows (see Figure 3.2b), this is called a directed graph. The connection between two nodes of G can be characterized by two arrow directions:

- tuple of (i, j) specifies the arrow direction from i to j
- tuple of (j, i) describes the arrow direction from j to i.

For undirected graphs (see Figure 3.2a), $(i, j) = (j, i)$ applies. Regardless of whether it is a directed or undirected graph, assigning a real number to an edge $\omega_{ij} \rightarrow \mathbb{R}$ results in a weighted graph (see Figure 3.2c). The edge weight usually represents the intensity of a connection between any two nodes. Typical examples for edge weights are distances, costs or amount of flow.

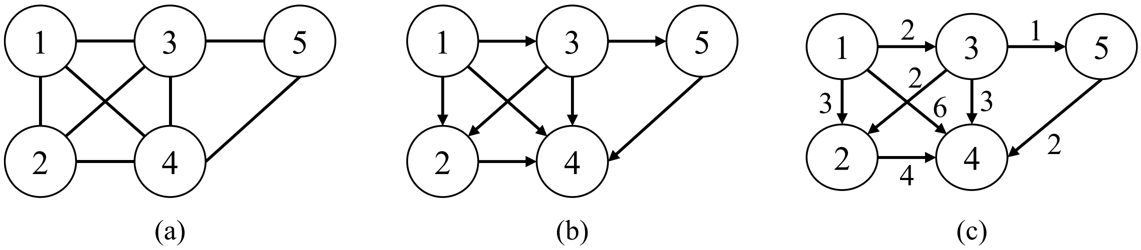


Figure 3.2: Graph definitions. (a) Nodes are illustrated as circles and edges as lines between nodes. (b) For the directed graph, the edge direction is indicated as an arrow. (c) For the weighted graph, the edge values represent the weights. A weighted graph can be both directed and undirected.

Adjacency Matrix Real systems typically comprise up to 1,000 nodes and are characterized by many connections between the nodes (Newman, 2003). Therefore, more complex calculations are necessary. For simplification, the mathematical representation as an adjacency matrix is widely used (Boccaletti et al., 2006). This matrix includes the relations of the individual nodes N of a graph G. The adjacency matrix A of an unweighted graph G is an $N \times N$ matrix, where the entries A_{ij} are defined as follows:

$$A_{ij} = \begin{cases} 1, \text{if } (i, j) \in E, \\ 0, \text{otherwise.} \end{cases} \tag{3.1}$$

Taking into account edge weights, the entries A_{ij} change to:

$$A_{ij} = \begin{cases} \omega_{ij}, \text{if } (i,j) \in E, \\ 0, \text{otherwise.} \end{cases} \tag{3.2}$$

For the graphs illustrated in Figure 3.2, the following adjacency matrices are obtained:

$$A(G_{(a)}) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}, A(G_{(b)}) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$
$$A(G_{(c)}) = \begin{pmatrix} 0 & 3 & 2 & 6 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 2 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}. \tag{3.3}$$

For undirected graphs, the adjacency matrix is symmetric because $A_{ij} = A_{ji}$.

Complete Graph In graph theory, the term complete graph describes the case where each node is connected to every other node of a graph. If only one edge is missing between any two nodes, this is called an incomplete graph (Reichardt, 2009).

Subgraph A graph G$_S$ = (V$_S$, E$_S$) is a subgraph of *G = (V, E)*, if $V_S \subseteq V$ and $E_S \subseteq E$. Hence, a subgraph is a subset of nodes in which the nodes are connected only by the edges that also occur in the original graph (Costa, Rodrigues, Travieso, & Villas Boas, 2007). Edge characteristics such as weights or directions are also adopted.

Clique A subgraph of an undirected graph *G* that is fully connected is referred to as a clique (Schaeffer, 2007). The size of a clique is defined by its number of nodes. A maximum clique is a clique of the largest possible size. If a clique comprises *k* nodes, it is referred to as a *k*-clique.

Node degree In general terms, the node degree k_i is defined as the number of edges a node *i* has with the remaining nodes (Newman, 2003). For directed graphs, a distinction is made between incoming and outgoing edges. The number of incoming edges indicates the in-degree k_i^{in} of a node, and the number of outgoing edges indicates the out-degree k_i^{out} of a node (Boccaletti et al., 2006).

In the case of weighted graphs, the edge weights must be considered. Hence, the node degree is referred to as weighted degree or node strength $s_i = \sum_{j=1}^{N} w_{ij}$ (Barrat, Barthélemy, Pastor-Satorras, & Vespignani, 2004). Accordingly, weighted in-degree s_i^{in} and weighted out-degree s_i^{out} can be calculated when analyzing directed graphs (Opsahl, Agneessens, & Skvoretz, 2010). A node without any edges represents a special case and is referred to as an isolated node (Albert & Barabási, 2002).

Path In an unweighted and undirected graph, a path is a node-to-node connection, where any two consecutive nodes are connected by an edge. The path length is determined by the required number of edges to traverse from one node to another (Albert & Barabási, 2002). There can be multiple paths between two nodes. The shortest path length d_{ij} specifies the least required number of edges to traverse from node *i* to node *j* (Boccaletti et al., 2006).

Density The ratio of actual edges to the number of theoretically possible edges is defined as density *D* (Albert & Barabási, 2002). Mathematically, for unweighted and undirected graphs, density is defined by the following equation:

$$D = \frac{2|E|}{|V|(|V|-1)}. \tag{3.4}$$

For directed, unweighted graphs, *D* is calculated as

$$D = \frac{|E|}{|V|(|V|-1)}. \tag{3.5}$$

If there are no connections between any nodes, meaning there are only isolated nodes, the density is 0. By contrast, *D* is 1 if the graph is complete. Therefore, the density value ranges from 0 to 1. For weighted graphs, the number of edges can be replaced by the sum of their weights (Arratia & Mirambell, 2021).

Clustering Coefficient For an unweighted and undirected graph, the clustering coefficient *C* of a given node *i* is defined as the ratio of existing edges e_i between the node's neighbors and potentially possible edges (Watts & Strogatz, 1998). If the neighbors of the observed node are fully connected, the clustering coefficient is 1. With only a few edges within its neighborhood, the value is close to 0. The equation for *C* is

$$C = \frac{2e_i}{k_i(k_i-1)}. \tag{3.6}$$

Random Walk A random walk generally represents a specific sequence of visited nodes, starting at node *i* and randomly selecting the next node *j* among its neighbors. A random walk is a stochastic process (Göbel & Jagers, 1974). The random walk picks the next step uniformly and at random from among all neighbors of a given node. The probability of the transition from node *i* to node *j* depends on the node degree k_i. The transition probabilities are summarized in a transition matrix *P*. The transition probability P_{ij} on an undirected graph is given by

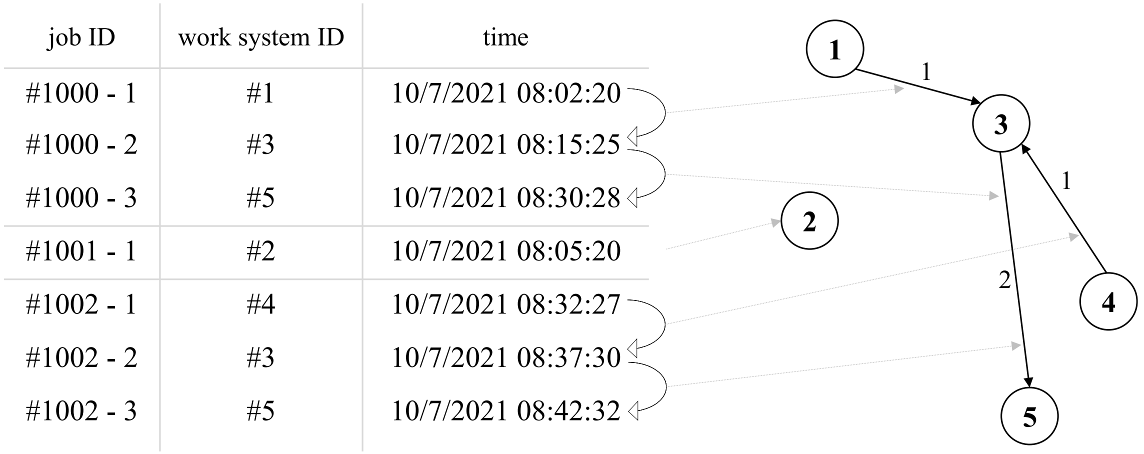
$$P_{ij} = \begin{cases} \frac{A_{ij}}{k_i}, \text{if } (i,j) \in E, \\ 0, \text{otherwise.} \end{cases} \tag{3.7}$$

The length *l* of a random walk is given by the number of steps required to traverse from node *i* to node *j* (Lovász, 1996).

3.2 Modeling Material Flow Systems as Complex Networks

The following explanations on the modeling of material flow networks as complex networks refer to Becker et al. (2014) and Wagner and Becker (2016). Within manufacturing systems, jobs are a form of routing between individual work systems. This routing creates material flows. A network model offers a clearly structured and easy-to-understand representation of the actual material flow. Such a network-based representation has the advantage that the focus is placed on the routing of jobs and the interactions between individual work systems, not on the underlying process or the manufacturing layout (Vrabič et al., 2012). The data required for the network representation can be obtained from the Manufacturing Execution System (MES).

Figure 3.3 shows an example of a data list of the internal material flows and how these data can be transferred into a network representation. Each line in the table contains all the information about a specific processing task. For each job, the job identity (ID) (first column), work system ID (second column) and time at which the job execution started (third column) are displayed. The table is sorted in ascending order, first by the job ID and then by the execution start time. As evident from the job ID, each job comprises one or more operations. Each work system represents a node in the network model. The material flow between any two work systems is represented by a directed edge. Depending on the required level of detail, an edge weight can optionally be added that corresponds to the aggregate material flow between any two work systems. The edge weight reflects the intensity and relevance of interactions between two work systems. A material flow system can thus be represented by a directed and weighted graph.



job ID	work system ID	time
#1000 - 1	#1	10/7/2021 08:02:20
#1000 - 2	#3	10/7/2021 08:15:25
#1000 - 3	#5	10/7/2021 08:30:28
#1001 - 1	#2	10/7/2021 08:05:20
#1002 - 1	#4	10/7/2021 08:32:27
#1002 - 2	#3	10/7/2021 08:37:30
#1002 - 3	#5	10/7/2021 08:42:32

Figure 3.3: Transformation of feedback data from Manufacturing Executions Systems (MES) into a network representation. Each work system is represented by a node, and the material flow between two systems is represented as edges. Edges have directions that show the orientation of flow. Edge weights are assigned to represent the amount of material flow between any two nodes.

This section has given a short overview of how job routing data can be used to create a network representation of a material flow system. The following section discusses the structural properties of real-world networks, some of which are also found in material flow networks.

3.3 Structural Properties of Real-World Networks

Access to real-world data enabled researchers precise analysis of networks in terms of their common structural properties. This information helped to illustrate how individual structural properties evolve and how they relate to each other. To this end, the properties of real-world networks were compared to those of random networks. In random networks, connections between any two nodes are distributed by chance (Boccaletti et al., 2006). By comparing real networks with random networks, scientists can determine whether the structural properties in real networks are significant or the result of random processes (Newman, 2010). Among the many properties that occur in real-world networks, scale-freeness and small-world-ness are considered the most important (Newman, Barabási, & Watts, 2006).

Scale-freeness means there are many nodes with only a few connections as well as some nodes with many connections to other nodes. The degree distribution of a scale-free network fits a power-law (Barabási, Albert, & Jeong, 2000). Barabási, Albert, and Jeong (1999) concluded that this structural property occurs because new nodes tend to attach to existing nodes that possess many edges to other nodes. This process is referred to as preferential attachment. Furthermore, scale-free networks are considered robust, at least when nodes fail randomly. However, if highly connected nodes are removed on purpose, the network quickly collapses into several sub-graphs (Boccaletti et al., 2006). These nodes, therefore, take a leading part regarding system's behavior (Barabási et al., 1999).

Vrabič et al. (2012) demonstrated the scale-freeness of material flow networks. Small-world networks are characterized by a short average path length between any two nodes and a high average clustering coefficient (Albert & Barabási, 2002; Travers & Milgram, 1969). This is reflected in a degree distribution where all nodes have nearly the same degree. Such networks are particularly well suited for disseminating information within the respective network (Boccaletti et al., 2006).

This section has introduced the important terms and concepts related to graph theory and used throughout this thesis. The next step is to describe how clusters can be defined, identified and evaluated.

3.4 Network-based Definitions of Clusters

The major benefit of finding clusters is that they support discovering and understanding the internal organization of a network. A cluster is generally defined as a group of elements with similar properties (Schaeffer, 2007). Therefore, it is assumed that a cluster reflects a special relationship between its nodes (Fortunato, 2010).

Clustering refers to a process where elements are assigned to individual clusters, with the aim of ensuring that elements within a cluster are more similar than they are to elements of other clusters. Similarity measures therefore play a major role in clustering. Depending on the particular application, a distinction can be made between different similarity measures. For example, the similarity of product groups can be described based on common characteristics (e.g. size, color) (Zhang, Priestley, DeMaio, Ni, & Tian, 2021).

In complex network theory, the degree of similarity is usually determined by the connectedness of nodes. This creates a cluster with strongly connected nodes. Moreover, these nodes have only a few connections to the nodes of other clusters (Fortunato, 2010; Schaeffer, 2007).

In cases where nodes within a cluster are fully connected, they are referred to as a clique (Palla, Derényi, Farkas, & Vicsek, 2005). Therefore, a clique is a special case of a cluster. The size of a clique is usually user-defined. Finding cliques of the maximum possible size is considered algorithmically difficult (NP-complete) (Schaeffer, 2007). However, this definition of clusters is rather restricted and a broader definition is used in this thesis. There is a need to modify this broader definition of clusters regarding the weights and directions of edges as relevant information about the intensity and orientation of material flows.

There is relatively little literature that considers the edge weights (Lancichinetti & Fortunato, 2009b). Fan, Li, Zhang, Wu, and Di (2007) showed the effects that arise from considering edge weights and confirmed that the cluster topology of a network changed significantly with varying edge weights. Usually this means that the broader cluster definition from above is extended to include edge weights. According to Brandes, Gaertler, and Wagner (2007), the sum of edge weights within a cluster should be higher than to rest of the network.

For directed graphs, two types of clusters can be distinguished, namely density-based clusters and pattern-based clusters (Malliaros & Vazirgiannis, 2013). The density-based definition of a cluster is essentially identical to the above definition for undirected cases. In the pattern-based definition of a cluster, the edge directions are used to find nodes with similar patterns of connections. It is assumed that these nodes are merged to one cluster. Malliaros and Vazirgiannis (2013) mentioned co-citation and flow as examples of potential patterns. Here, the flow-based definition of a cluster seems highly relevant, since the routing of jobs corresponds to a flow. Therefore, nodes are merged according to the flow pattern. By analogy to the density-based definition of a cluster, the flow within a cluster should be larger than to rest of the network. However, Malliaros and Vazirgiannis (2013) highlighted that density- and pattern-based clusters quite often arise within the same network.

To summarize, there is no clear and generally accepted definition of clusters in the literature. Moreover, it is important to determine whether edge weights and/or edge directions should be included in the definition of a cluster. Generally, all available information should be included for better clustering results (Fortunato, 2010). The definition of a cluster is important because it provides a necessary orientation when choosing an appropriate clustering algorithm. As described in Section 3.2, material flow systems can be modeled as directed and weighted networks. For this reason, edge directions and edge weights should be considered when choosing an appropriate clustering algorithm for identifying autonomous clusters. Since the routing of jobs corresponds to a kind of flow, the flow-based cluster definition seems reasonable. Furthermore, due to the consideration of edge weights, a clustering algorithm needs to be able to identify a mixed type of flow-based and density-based clusters. In literature, there are numerous clustering algorithms which allow edge weights and/or edge directions to be considered. The following section provides an overview of the different types of clustering algorithms and indicates the limitations of considering edge weights and directions.

3.5 Classification of Clustering Algorithms

In recent years, many clustering algorithms have been developed for different applications. Overviews of existing algorithms are provided by Schaeffer (2007), Fortunato (2010) and Dao, Bothorel, and Lenca (2020). However, there is no general agreement on how to classify the different clustering algorithms (Dao et al., 2020). In this thesis, clustering algorithms are classified according to the membership of nodes to each cluster. Most algorithms allow one node to belong to exactly one cluster (*disjoint clusters*). However, in recent years some extensions regarding the membership of a node were made, and clustering algorithms were developed where a node is allowed to belong to one or several clusters (*overlapping* and *fuzzy clusters*).

In the case that each node is assigned to exactly one cluster, clusters are referred to as *disjoint clusters*. Hierarchical and partitional clustering algorithms aim to identify such disjoint clusters. Hierarchical clustering can be further subdivided into agglomerative and divisive algorithms. Agglomerative algorithms start with each node representing its own cluster. Next, they are successively merged using a similarity measure until all nodes end up in a single cluster. To obtain the best clustering, the intermediate results from the merging are compared using quality criteria. With divisive algorithms, the procedure is the opposite, i.e. all nodes start in a single cluster (Scott, 2017). Then, this cluster is split into several clusters until each node represents its own cluster. Again, the intermediate results are evaluated. For both agglomerative and divisive algorithms, the same principle applies: if there is no improvement, the merging or splitting process is stopped.

In summary, hierarchical algorithms have the advantage that the number of clusters is variable, i.e. a predefinition of the number of clusters is unnecessary. However, according to Scott (2017), this aspect also has the disadvantage that it is up to the user to define a stop criterion for the merging or splitting process. This disadvantage is avoided when using partitioning algorithms because the number of clusters is defined from the beginning, although this requires a good knowledge of the network topology (Fortunato, 2010). Partitioning algorithms work by randomly assigning existing nodes to clusters, and successively changing the membership of nodes until a desired cluster quality is achieved. Because of the reassignment of node membership, partitioning algorithms are time-consuming (Fränti & Sieranoja, 2019).

Palla et al. (2005) stated that some networks intrinsically have *overlapping clusters*, i.e. nodes can be members of more than one cluster. The authors indicated that friendship networks provide a good example. A person makes friends with other people depending on their interests, and within a friendship network, one node (person) may be part of several clusters (friends with similar interests). In fact, networks with many overlapping clusters are more difficult to handle for users as they tend to become blurry (Lancichinetti, Radicchi, Ramasco, & Fortunato, 2011). Fortunato (2010) presented a review of clustering algorithms that provide overlapping clusters.

Another mechanism enables each node to belong to different clusters according to a certain degree of membership. These clusters are described as *fuzzy clusters*. In clustering algorithms that use the principle of fuzzy logic, each node is assigned a membership weight from the interval [0, 1] for each cluster found (Fortunato, 2010). A value of 0 indicates that the node does not belong to the cluster and a value of 1 indicates that the node definitely belongs to the cluster. Any value between 0 and 1 indicates the probability of a node belonging to the cluster.

With respect to material flow clusters, this classification of clustering algorithms according to the membership of nodes in each cluster has several implications. In overlapping and fuzzy clusters, work systems may belong to more than one cluster at the same time. This situation requires effective coordination among the clusters to check the current availability of work systems. Derived from the idea of the hybrid clustering-based approach, where no overall cluster coordination unit is intended, only algorithms that result in disjoint clusters are used in this thesis. In this approach, each work system belongs to exactly one cluster.

Most clustering algorithms focus on unweighted and undirected networks. Particularly, the simultaneous consideration of edge weights and edge directions is rare. Furthermore, the consideration of edge weights is easier than the consideration of edge directions (Lancichinetti et al., 2011). This is partly because directed graphs are described using asymmetrical matrices, which makes calculations particularly complicated (Fortunato, 2010). In their systematic review on clustering in directed networks, Malliaros and Vazirgiannis (2013) provided a summary of 27 popular clustering algorithms that can be extended to include edge directions. Lancichinetti and Fortunato (2009b) evaluated a set of seven clustering algorithms regarding their performance in directed and/or weighted networks. They created networks with different predefined cluster topologies according to various parameters, such as cluster sizes and the number of clusters, and thus covered a wide range of applications. Their findings suggest that in networks with well-defined clusters, the clusters can be identified equally well by several algorithms. Essentially, any algorithm is able to find clusters in any network, and thus it is crucial to evaluate the quality of the resulting clusters (Fortunato, 2010).

3.6 Cluster Validation for Weighted and Directed Networks

In the light of the fact that algorithms are able to find clusters in any given data set, evaluating the quality of the clustering results is essential. There are two ways to evaluate the quality of clustering: (i) quality measures can be used, especially where little is known about cluster topology, or (ii) the algorithm is applied to networks with a priori known cluster topology (also referred to as benchmark graphs; Fortunato, 2010). Both possibilities are described below.

Quality Measures

In literature, there is a variety of measures that can easily be adopted to evaluate the quality of clustering algorithms. One of the main advantages of using such quality measures is that no prior knowledge of the underlying cluster topology is required (Arratia & Mirambell, 2021). However, some of these measures have a dual function and are also used as objective functions in optimization clustering algorithms. The simultaneous use of the same measure as both an objective function and a quality measure positively influences the performance evaluation of a clustering algorithm (Schaeffer, 2007). Hence, this could mean that when the results of two different clustering algorithms are compared, one algorithm might be preferred. It is important to select quality measures that are objective and do not favor a specific clustering algorithm.

Several quality measures are provided for unweighted and undirected networks. An overview of quality measures for unweighted and undirected cases can be found in Brandes, Gaertler, and Wagner (2003), Schaeffer (2007) and Yang and Leskovec (2015). They focused mainly on quality measures for weighted and directed networks. Brandes et al. (2007) and Arratia and Mirambell (2021) adapted well-known quality measures for weighted networks. Brandes et al. (2007) indicated possible weaknesses in the quality measures used in their study and proposed using a combination of these measures to compensate for those weaknesses. Arratia and Mirambell (2021) indicated that some quality measures (e.g. internal density) are determined by the number of identified clusters. These measures penalize algorithms that find a few large clusters and reward those that find many small clusters. Weighted modularity, by contrast, seems to offer a measure that is not notably affected by this issue and is therefore better suited. Malliaros and Vazirgiannis (2013) surveyed directed graphs and stated that the directed version of modularity was the most widely used quality measure.

Based on these findings and the fact that modularity is one of the few measures that can integrate both edge weights and edge directions. Modularity is calculated as the difference between the total number of edges within a cluster and the total number of edges in a randomly generated network having same degrees of nodes. Modularity values range from 0 to 1. For a detailed mathematical description, the reader may refer to Newman and Girvan (2004) and Arenas, Duch, Fernández, and Gómez (2007).

It is a challenge to relate the network-theoretical perspective of cluster validation to the logistic-based perspective, since they can strongly differ. Just as there is no single correct quality measure, it must also be decided as the case arises what measure is appropriate in the context of autonomous clusters. Little experience is available. However, researchers agree that material flow clusters should be identified based on the interactions of their elements. The quality measures presented here can therefore be applied to the identified material flow clusters.

Synthetic Graphs

Alternatively, the performance of a clustering algorithm can be demonstrated based on networks with a priori known cluster topology. This approach requires real-world networks with known cluster topology, which are quite rare (Fortunato & Hric, 2016; Khan & Niazi, 2017). Hence, in recent years a number of models to synthetically create networks with a built-in cluster topology have been introduced. Such models allow a built-in cluster topology to be modified to meet specific application needs (Aldecoa & Marín, 2013).

The most well-known existing synthetic or benchmark graphs – these two terms are often used interchangeably – are based on the planted l-partition model (Fortunato, 2010). Hence, it was selected as a starting point. The planted l-partition graph comprises n nodes that are split into l clusters with equal size, where p indicates the probability that two nodes of the same cluster are connected, and r represents the probability that two nodes of different clusters are connected. The Girvan-Newman (GN) benchmark graph is a simple model that results in networks with same node degrees and clusters with same sizes (Fortunato & Hric, 2016; Girvan & Newman, 2002). However, in many real-world networks, node degrees and cluster sizes follow a power-law distribution (Clauset, Shalizi, & Newman, 2009). As a result, many clustering algorithms perform well on GN benchmark graphs (Danon, Díaz-Guilera, Duch, & Arenas, 2005).

The Lancichinetti–Fortunato–Radicchi (LFR) benchmark graph was proposed by Lancichinetti and Fortunato (2009a) to overcome the above problem. The following parameters (among others) are used: number of nodes n, exponent for the degree distribution, exponent for the cluster size distribution and mixing coefficient μ. The latter specifies how well-defined the clusters are. The larger the μ, the more blurred clusters the become and the harder it is to detect them (Orman, Labatut, & Cherifi, 2012). This model is characterized by a power-law distribution of node degrees and cluster sizes. This makes the network topology complex and the identification of clusters difficult, which means some clustering algorithms perform poorly (Lancichinetti & Fortunato, 2009b; Orman et al., 2011). LFR graphs are now widely used as a standard benchmark graph.

Another type of benchmark is based on relaxed caveman (RC) structures, where initially it is assumed that networks are created by isolated cliques. In the next step, the edges within these cliques are rewired with a probability p. The initial clique structures become more blurred by increasing the edge rewiring probability p (Schaeffer, 2007). In analogy to LFR graph, the identification of clusters becomes more difficult. Three parameters are required to create RC benchmark graphs: number of clusters c, cluster sizes s and edge rewiring probability p. Accordingly, it is a simple model that is easy to use. Although RC benchmark graphs do not reflect the characteristics of a real-world network, they – along with the LFR graph – are widely used as a standard benchmark graph (Aldecoa & Marín, 2013).

Regarding the consideration of edge weights and/or edge directions, two benchmark graphs have been notably modified. The GN benchmark was extended to include edge weights (Fan et al., 2007), and the LFR model was modified to take into account both edge weights and edge directions (Lancichinetti & Fortunato, 2009b). The modified LFR model also allows creating networks with overlapping clusters (i.e. nodes belonging to more than one cluster). However, these models do not consider any kind of flow within the generation process as it is required for material flow networks.

The overall conclusion is that there are numerous models to create benchmark graphs with a built-in cluster topology, but there remain significant structural differences between them and real-world networks (Orman et al., 2012). This fact must be considered when using the models to evaluate the performance of clustering algorithms.

Typically, to test the performance of a single clustering algorithm or to compare the clustering results of two or more clustering algorithms, the researcher generates a set of benchmark graphs with varying degrees of well separated clusters. In the next step, these clustering results are compared with the built-in cluster topology of the respective benchmark graph to quantify the similarity (Lancichinetti & Fortunato, 2009b). There are several similarity measures, of which the information-theoretical measures are most popular (e.g. Orman et al., 2011). They include the normalized mutual information (NMI) value. NMI is an entropy-based metric that quantifies the amount of information shared between two clustering results (Vinh, Epps, & Bailey, 2010). It can take values between 0 and 1. The higher the value, the more similar the clustering results of the two networks; 1 means that results are a perfect match, whereas values close to 0 indicate dissimilarity between the clusters. Normalized NMI can be used to compare the results of two clustering algorithms, where the number of identified clusters varies (van der Hoef & Warrens, 2019). Such similarity measures are also used to compare the clustering of different algorithms to ascertain whether the results are consistent (Malliaros & Vazirgiannis, 2013).

3.7 Summary

Clustering algorithms are typically developed to help to solve specific problems. Therefore, not every clustering algorithm is applicable for all problems and it requires validation before it can be used. Furthermore, edge weights and edge directions are still not widely considered. In particular, only a few models generate weighted and directed synthetic networks. Notably, most existing models do not allow for considering any kind of flow within the process. The next chapter introduces a method for the generation of networks with predefined cluster topology considering the material flow.

4 Generating Material Flow Networks with Varying Cluster Topology

Data relating to production processes are not widely available (Nyhuis et al., 2014) and there is a lack of models to generate weighted and directed networks that consider any kind of flow within the process (compare Section 3.6). The aim of this chapter is to develop and evaluate a methodological approach to generate material flow networks. The overall objective of this thesis is to ascertain whether (and to what extent) the cluster topology of a material flow network affects the achievement of logistic objectives when job sequencing occurs decentrally, within previously identified material flow clusters. The proposed methodological approach should be sufficiently flexible to allow generating networks with varying cluster topology with relatively little effort.

Although the term cluster was defined in detail in the previous chapter. Here, it must first be determined what type of clusters may coexist within a network to generate networks with varying cluster topology. Furthermore, an appropriate concept from network theory needs to be selected to imitate the material flow within a system (see Section 4.1). The next step is to evaluate the methodological approach (Section 4.2). To gain knowledge on how different parameters influence the cluster topology, the main results are given (Section 4.3). This is followed by a summary in Section 4.4. Parts of this chapter have been published in Wagner and Becker (2018).

4.1 Methodological Approach

According to the definition of a cluster given in Section 3.4, a rough distinction can be made between strong and weak clusters. Figure 4.1 shows examples of two networks with different cluster topologies. They differ particularly in the degree of interconnection (Schaeffer, 2007). In strong clusters the nodes are more connected to each other compared to the rest of the network. By contrast, in weak clusters the nodes have fewer connections between each other and therefore more edges connected to the rest of the network. For weak clusters, there is a risk that the structure will become too weak and the clusters will be no longer identifiable. Radicchi et al. (2004) offered a precise definition of strong and weak clusters. They assumed that in a strong cluster, any node has more connections within a certain cluster than to the rest of a network. Their definition of a weak cluster fits with the one of strong clusters. However, they highlighted that for some use cases, a slight variation of the definition is required. Regarding the effectiveness of the hybrid clustering-based approach, there is no evidence related to the influence of cluster quality; hence, a less strict definition seems sufficient for this thesis.

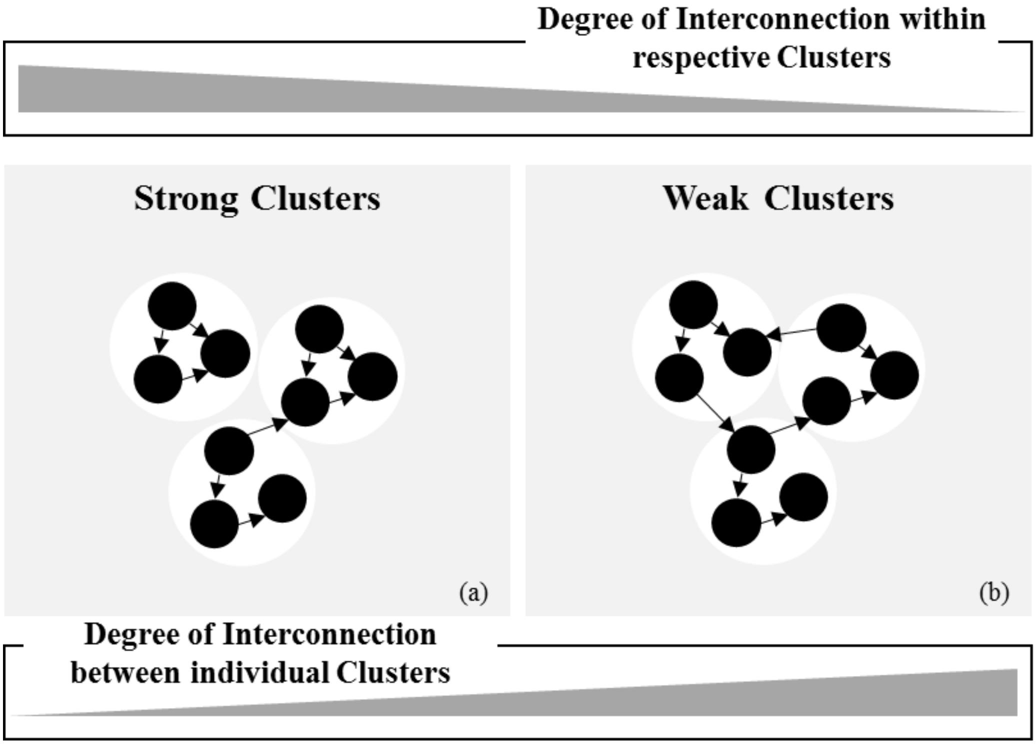

Degree of Interconnection within respective Clusters
Strong Clusters
Weak Clusters
(a)
(b)
Degree of Interconnection between individual Clusters

Figure 4.1: A distinction can be made between (a) strong and (b) weak clusters depending on the degree of interconnection. This partition can also be applied to weighted directed networks.

For creating networks with varying cluster topology, it seems insufficient to distinguish between strong and weak clusters, as a more global perspective is required. For example, networks can contain many or few clusters (Lancichinetti & Fortunato, 2009b). Another way to vary the cluster topology of networks is through defining the size of clusters (Lancichinetti & Fortunato, 2009b). For example, networks may display a few large or many small clusters (Figure 4.2).

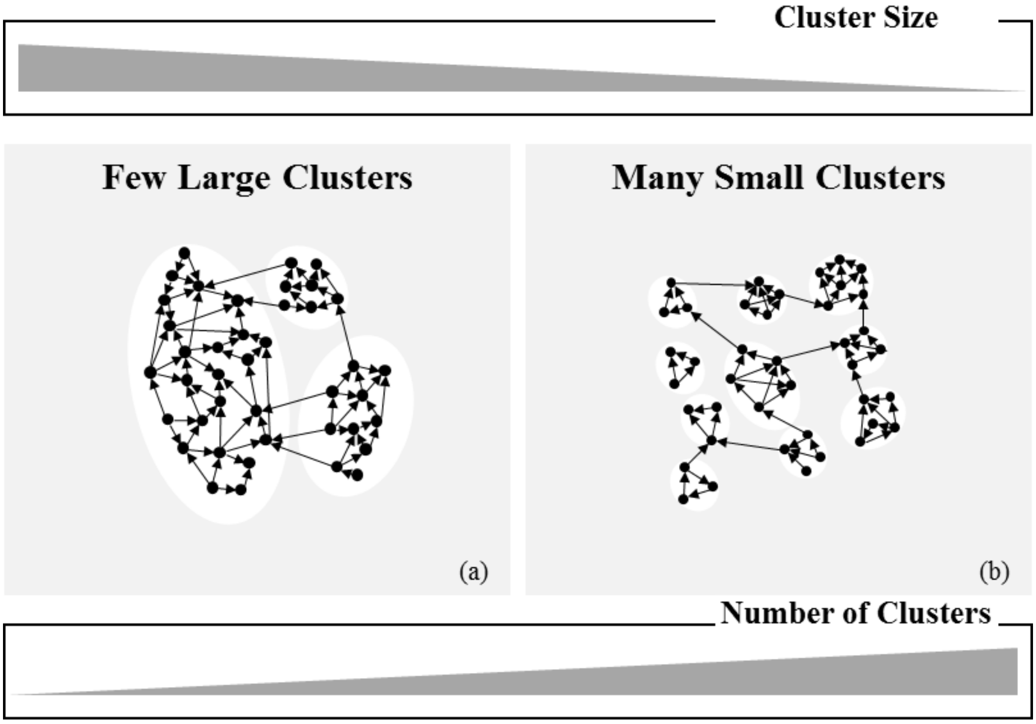


Figure 4.2: Networks with differing cluster topology, showing two extreme scenarios. (a) This network has a few large clusters. (b) A reduction in cluster sizes may lead to many small clusters. Other intermediate levels are possible; e.g. many large clusters with a few small clusters. The focus here is on the number and size of clusters and not on the correct representation of their degree of interconnection.

All the above descriptions assume disjoint clusters, i.e. each node is assigned to exactly one cluster. This is derived from the definition of decentralized control systems in Chapter 2. Thus, each cluster represents an individual control unit. In addition, all potential types of clusters can coexist in one network. Ideally, they can all be affected to achieve a sufficiently high degree of flexibility in the network design. To meet this requirement, numerous models have been proposed in literature; a short overview appears in Section 3.6. Some models, such as the LFR model, address edge directions and edge weights. However, material flow networks arise through job routing and processing; that is, jobs flow from one work system to the next in a certain order. A possible approach to simulate such dynamic processes is the random walk (Barrat, Barthélemy, & Vespignani, 2008). Essentially, a random walk is a stochastic process: the walker moves randomly within a network until the process is stopped (Bollobás, Riordan, Spencer, & Tusnády, 2006). The result is a node sequence, which is equivalent to the sequence of the operations of a job. A precise mathematical description of random walks appears in Lovász (1996).

Random walks are used in different areas and for different purposes (e.g. Lü & Zhou, 2011; Newman, 2005). Recently, new clustering algorithms have been developed based on the concept of random walks. The main idea is that once entered into a cluster, a random walk visits many of the cluster's nodes and therefore remains in the cluster for a while (Fortunato, 2010; Pons & Latapy, 2006). In this way, the different clusters can be identified. This concept completes the idea of using random walks to generate material flow networks with varying cluster topology.

The originality of the proposed methodological approach in this thesis concerns the integration of the desired cluster topology and a realistic manufacturing environment. Essentially, the approach builds on the general procedure for generating synthetic networks. First, information about the desired material flow network properties needs to be specified, including, for example, the number of nodes and some general characteristics of the cluster types. Furthermore, when defining a manufacturing environment for a simulation study, data on the relevant work systems and jobs need to be considered (Holthaus, 1996). Work system data are considered in the first step (Section 4.1.1), whereas job data are specified toward the definition of random walks in the second step (Section 4.1.2). Moreover, information from real data sets can be included or own assumptions can be made at any step. In the following section, the individual steps are described in further detail.

4.1.1 Definition of the Underlying Network

A suitable network topology must be selected, which forms the basis for the random walk process. Such a network is referred to as the "underlying network" (Cupertino, Carneiro, Zheng, Zhang, & Zhao, 2018; Rosvall, Axelsson, & Bergstrom, 2009). The definition of the underlying network is crucial, since the links between individual nodes represent potential routes for the random walks. Hence, random walks mainly affect the edge weights and directions but do not create any new edges.

Generally, any real-world manufacturing system can be used as an underlying network. However, if it is desirable to plant certain network properties – like cluster topology in this case – then existing networks models should be used. Through the use of predetermined model parameters, different network topologies can be created. Based on the above specification of various cluster types, the following parameters should take different values: specification of (i) the total number of nodes (i.e. number of work systems) in the network, (ii) the number of clusters, (iii) the cluster sizes and (iv) the degree of interconnection between individual clusters. If possible, to enhance the transparency and simplicity, the network model should only be adjustable using these four parameters.

A model that meets these requirements is the RC graph. As explained earlier, the RC graph is defined by three parameters (the number of clusters c, the cluster size s and the probability p that clusters are interconnected; Schaeffer, 2007). The number of nodes results from the product of c and s. Furthermore, the strictest definition of a cluster is assumed, which means that all nodes in a cluster are interconnected, with a low p-value. For higher p-values, clusters become more blurred and cannot be clearly identified, which leads to unequal cluster sizes (Aldecoa & Marín, 2013). Increasing the p-value has also the effect of reducing the number of clusters in a network. Regardless of the clustering algorithm used, there is effectively an upper limit for p, after which all nodes result in one giant cluster (Chin & Ratnavelu, 2016). The upper limit value depends on the respective use case and must be determined individually.

To summarize, the p-value has a substantial influence on the resulting cluster topology. Regarding the parameters c und s, they require further investigation. Since the underlying networks are considered crucial for the random walks, one aim is to show how these three network parameters correlate and affect the resulting cluster topology.

4.1.2 Definition of Random Walks

When jobs are created in a simulation study, time-related data and technological data are required to describe their characteristics (Holthaus, 1996). While time-related data (e.g. job processing time and job arrival time) become relevant when running the simulation study (see Chapter 6), technological data may affect the network topology. To use technological data during the process of creating material flow networks with varying cluster topology, equivalent parameters for random walks can be found in literature. Table 4.1 provides a summary of relevant parameters. However, random walk assumptions are constrained by the assumptions made for the underlying network.

Technological data	Random walks
Total number of jobs	Total number of random walks (Lü & Zhou, 2011)
Number of operations of an individual job	Number of steps of an individual random walk (Pons & Latapy, 2006)
Job routing under technological constraints	Start- and endpoint of a walker (Lü & Zhou, 2011)
	Selection of the next step of a walker (Pons & Latapy, 2006)

Table 4.1: Comparison of technological data (derived from Holthaus 1996) and equivalent random walk parameters to describe the characteristics of jobs.

In conducting a simulation study, it may be necessary to define how many jobs must be processed in a certain period. Usually, this is a fixed number and is described by the parameter *total number of random walks*. Depending on the assumptions related the underlying network, a sufficient number of random walks is necessary to adequately capture the hidden cluster topology. Furthermore, running many random walks increases the edge weights (Wagner & Becker, 2017). This point might be important when running a clustering algorithm.

Each job comprises a specified number of operations, expressed by the parameter *total number of steps of individual random walks*. This parameter is either fixed or variable. Depending on the cluster sizes, random walks with few steps generally perform more poorly in recognizing the hidden cluster topology than do random walks with more steps. Accordingly, from the network perspective, the values of this parameter should be variable.

Depending on the production type on hand, it might be that jobs have to be processed in a certain work system sequence. For example, in a flow shop, the sequence is the same for all jobs, and in a job shop it varies depending on the job type (Lödding, 2013). In this thesis, the assumptions made for random walks are based on the job shop characteristics because autonomous control methods are better suited for job shop environments (as described in Chapter 2). The parameters *start- and endpoint of a random walk* describe the first and last work system on which a certain job needs to be processed. Both start- and endpoint can be defined in advance. For a given endpoint, a walker must enable achieving this endpoint. However, because of the predefined number of steps of a random walk and the existing edges between the individual nodes, the endpoints are not always achieved, and thus start- and endpoints are determined randomly. Additionally, the parameter *selection of the next step* should be added. Derived from previous work, the next step of a walker can be individually set in advance or in the middle of the random walk process. Due to the fixed topology of the underlying network, it is difficult to establish predefined routes for the random walks. Given this restriction, it is assumed that the selection of the next step is purely random, with every neighboring node having an equal probability of being chosen. The fact that the selection of the next step is restricted can be seen as a technological constraint.

4.2 Experimental Setup and Evaluation

The proposed methodological approach was evaluated, and its performance was assessed on various underlying networks having different parameter configurations. The findings help to explain how the parameter values of the underlying network affect the desired cluster topology. For this purpose, real historical data derived from the ERP system of a medium-size tool and assembly manufacturer were used.

From the combination of a high number of variants and low volumes, it can be concluded that it is a single or small batch production in a job shop environment (Lödding, 2013). Because of the complexity of the routing processes, the use of autonomous control is appropriate, and this production environment is ideally suited for further simulation studies. However, when generating synthetic material flow networks with a certain cluster topology, it was not possible to consider all available information from the original data set. Where possible and appropriate, these dates were adopted in subsequent steps.

Based on the selected real data set, the resulting material flow network should include a total number of 50 nodes. Depending on the field of application, complex networks may comprise up to several thousand nodes (Albert & Barabási, 2002). However, real material flow networks are considerably smaller. For example, in Becker et al. (2014), networks with between 50 and 220 work systems were analyzed. Other material flow networks have included 99 nodes (Vrabič et al., 2013) or 199 nodes (Vrabič et al., 2012). By contrast, material flow simulations are performed with a wide range of work systems, such as from four to 223 (Freitag & Hildebrand, 2016; Holthaus, 1996; Vinod & Sridharan, 2010). The decisive factor here is that a network should be sufficiently large so that a sufficient number of clusters can be found. Based on the results of Becker and Wagner (2015) and Wagner and Becker (2016), the assumption that the resulting network comprises approximately 50 nodes, creates an appropriate basis for further analysis. Although a network and its elements are subject to changes due to dynamic processes (Wagner & Becker, 2016), this study assumed that the number of nodes would remain constant over the studied period.

When generating material flow networks, researchers face the challenging task of selecting the right configuration of the parameters c, s and p. Previous work on synthetic networks has indicated the effect of p on the resulting cluster topology. To date, it has not been fully assessed how the network topology changes for different combinations of c and s. Therefore, the value of p was varied for different combinations of c and s, as follows.

The values for the parameters c and s of the underlying network result from the total number of nodes, whereby the product of these two parameters remains 50. The value of 50 has the factors 1, 2, 5, 10, 25 and 50, with 1 and 50 being trivial. Table 4.2 shows the different combination options of these two parameters. For experiments in this work, 36 underlying networks of each possible configuration involving *c, s* ϵ {2, 5, 10, 25} and *p* ϵ {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} were generated.

Combination	***c***	***s***
1	2	25
2	5	10
3	10	5
4	25	2

Table 4.2: Possible combinations of number of clusters *c* and cluster size *s* to generate underlying networks with 50 nodes.

Derived from the real data set, the following assumptions were made regarding the random walks. Overall, there are 3,702 random walks, with the length of each random walk ranging from 1 to 31. As in the original data set, material backflows are allowed. This means a walker is allowed to go back to a node that it has visited before. Additionally, in the definition of an underlying network, it can be differentiated whether a manufacturing system contains one or several work systems of the same type. This mainly depends on the type of production (Holthaus, 1996; Lödding, 2013). Assuming that there are several work systems of the same type, it can happen that exactly these work systems result in one cluster. However, since a random walk selects its next step randomly, there is a high probability – especially for nodes with only a few neighboring nodes – that the walker will visit the same type of work system multiple times. This does not correspond to reality. Otherwise, one would have to restrict the random walks or force them to change the cluster. Therefore, it is assumed that there was only one unit of each type of work system. The remaining parameters (such as the start- and endpoint or the next step of a walker) were defined according to the general assumptions described in Section 4.1.2. Each of the 36 networks was generated 15 times with different random seeds.

The setting and effect of parameters of the underlying network and of the random walks were analyzed for the resulting cluster topology. It should be considered that the cluster results significantly depend on the clustering algorithm used. However, the results of different algorithms become more relatively similar if the clusters are relatively unique (Lancichinetti & Fortunato, 2009b). Derived from the characteristics of a material flow network (see Section 3.2) and the desired cluster properties described at the beginning of this section, a clustering algorithm should meet the following requirements:

- integration of edge weights and edge directions
- each node belongs to only one cluster (disjoint clusters)
- no specification of the number of clusters or the size of individual clusters in advance.

The challenge was that the integration of edge weights and edge directions makes a clustering algorithm more complex (Lancichinetti et al., 2011). It also limits the choice of available algorithms. The results of previous work on weighted and directed synthetic networks show that Infomap (IM) outperforms other algorithms (Lancichinetti & Fortunato, 2009a), and in unweighted and undirected networks it is one of the best-performing clustering algorithms (e.g. Aldecoa & Marín, 2013; Orman et al., 2011). This suggests that an algorithm that performs well for various synthetic networks is preferable to algorithms that work well only in a specific setting. Additionally, the IM algorithm is especially suited for networks where connections between individual nodes describe a movement-pattern of flow (Rosvall & Bergstrom, 2008). It is therefore well suited for material flow networks.

The IM algorithm is an information-theoretical approach, which analyzes the flow in networks. For modeling the network flow, random walks are also used here. The basic idea is to identify clusters in which the flow remains for a relatively long time. This is achieved through minimization of the map equation, which describes the average description length of a random walker. A detailed description of the IM algorithm can be found in Rosvall and Bergstrom (2008) and Rosvall et al. (2009).

To evaluate the effect of the underlying network on the cluster topology when applying random walks, the resulting clusters were compared with those of the underlying networks. To determine how closely the clustering results matched, NMI was used (for a definition, see Section 3.6). This is a widely used metric to measure the similarity of two clustering results (van der Hoef & Warrens, 2019).

In the next step, a measure to evaluate the quality of the individual clusters of the resulting network was used. According to the specifications mentioned at the start of this section, a distinction was made between strong and weak clusters. Modularity seemed to be a good indicator for this evaluation (for a definition, see Section 3.6). Higher values indicate that the nodes in a cluster have more connections to each other than to the rest of the network and thus represent a strong cluster. In addition, it is a widely used measure and is often used as a quality function in optimization algorithms (Fortunato, 2010). However, IM is based on information-theoretical principles. Overall, it seemed an appropriate measure for this use case as the assumptions on which the calculations were based varied.

4.3 Influence of Parameters *c*, *s* and *p* on Cluster Topology

The results of the NMI values are shown in Figure 4.3. As expected, depending on the selected configuration of the number of clusters *c*, cluster size *s* and probability *p* that clusters were interconnected, there were differences between the cluster results for the underlying networks versus the results after applying random walks.

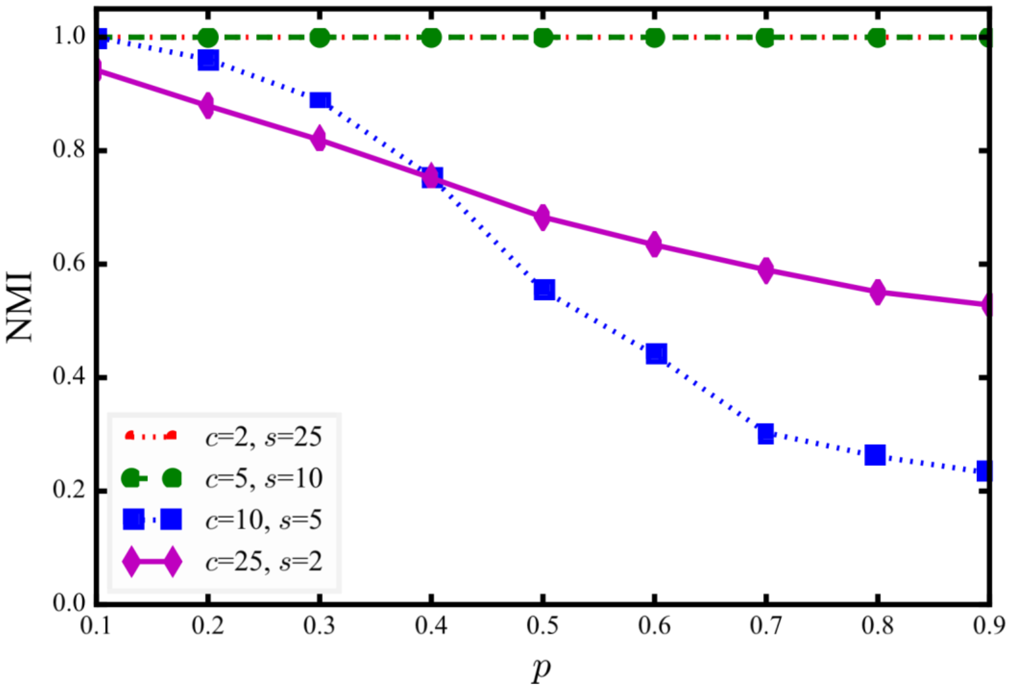


Figure 4.3: Similarity is expressed in NMI for different combinations of *c*, *s* and *p*. The IM algorithm was used to find clusters. Each point in the diagram corresponds to the average value over 15 material flow networks.

As described in Section 4.1.1, with increasing *p*, it becomes more difficult to identify clusters due to blurry boundaries between them. However, the results imply that the impact of *p* is determined by the selected combination of parameters *c* and *s*. If the *c*-value is smaller than the *s*-value, the cluster results of the underlying network match the results from applying the random walk approach, regardless of the *p*-value. However, if the *c*-value is larger than the *s*-value, the value of NMI drops with an increasing *p*-value. This means the clustering results differ strongly between individual networks. Thus, random walk's performance is particularly sensitive to combinations where *c*>*s*.

In other words, specifying an underlying network with few large clusters (*c*<*s*) leads to these clusters remaining almost unchanged after applying random walks. Given this scenario, once a random walk enters a cluster, it remains trapped in it for a while. This fact is not changed for different *p*-values. By contrast, specifying an underlying network with many small clusters (*c*>*s*) means these clusters become blurry. Hence, the random walk changes between the clusters more frequently, so the clusters of the resulting network no longer match those of the underlying network. This is reinforced by a higher *p*-value. This result could be connected to the performance of the IM algorithm. Previous work on synthetic networks indicates that the performance of the IM algorithm mainly depends on the average degree of nodes (e.g. Lancichinetti & Fortunato, 2009b; Orman et al., 2011).

As evident in Figure 4.4, the combination of the parameters *c* and *s* has a significant influence on the average degree of nodes. In general, the larger *c* is in comparison to *s*, the higher the resulting average degree. This principle can be explained by the fact that in networks with only a few large clusters (e.g. *c=2* and *s=25*), the nodes within these clusters have many edges between them. Thus, the individual nodes have a higher degree, which results in a higher average degree. By contrast, in networks with many small clusters (e.g. *c=25* and *s=2*), the individual nodes have a lower degree, and since random walks only use existing edges, this leads to a low average degree. The parameter *p* has little influence on the average degree for a fixed combination of *c* and *s* (see Figure 4.4). However, the degree distribution is affected by varying *p* (Orman et al., 2011). The different degree distributions are necessary so that different cluster topologies can result. These results are also observed when considering edge weights.

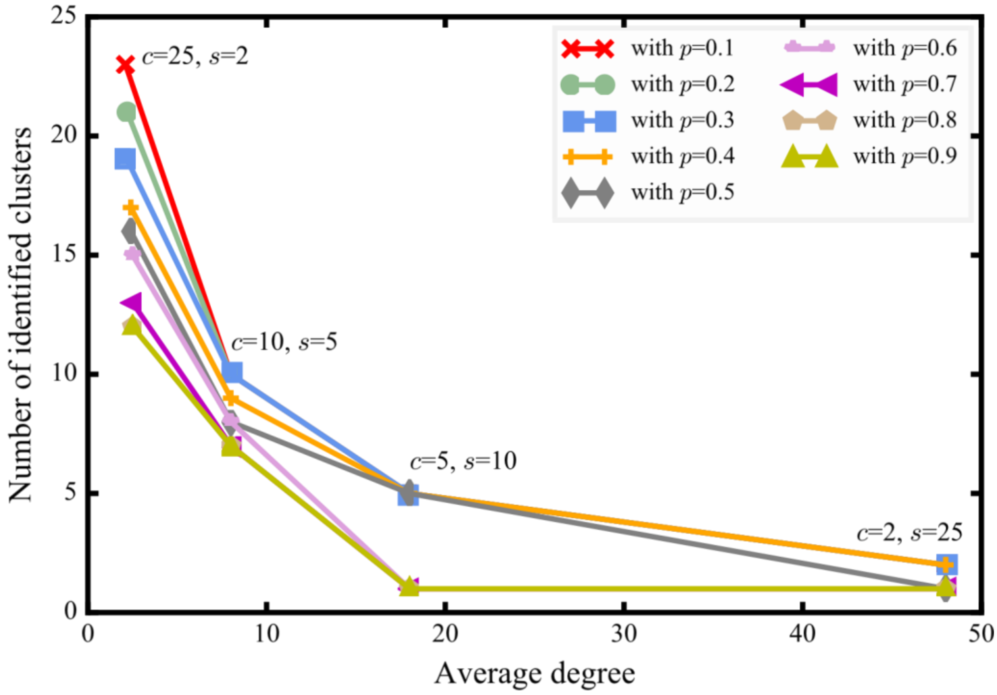


Figure 4.4: A small average degree leads to many clusters being found. By contrast, a high average degree means only a few clusters are found. Different values of *p* have relatively little effect on the average degree for a fixed combination of the parameters *c* and *s*.

A closer look at the correlation between the average degree and the number of identified clusters reveals that there seems to be an upper limit for *p*, above which all nodes end up in one giant cluster (see Figure 4.4). For example, this is the case where *c*<*s*, whereby the decisive factor is the exact combination of *c* and *s*. For *c*>*s*, as well as for larger *p*-values, several clusters can be identified. These results are also reflected when evaluating the cluster quality.

As expected, the modularity steadily decreased for all combinations of *c* and *s* as the *p*-value increased, as evident in Figure 4.5. This can be explained by the increasingly blurry boundaries between clusters. Furthermore, it can be seen that the combination of *c=25* and *s=2* was nearly the best performer regarding modularity. The choice of a relatively large cluster size (*c=2* and *s=25*) led to opposite results. Derived from the previous findings, a similar decrease in performance was observed for the modularity. For example, modularity fell from 0.36 at *p=0.3* to 0 at *p=0.4,* for *c=2* and *s=25.*

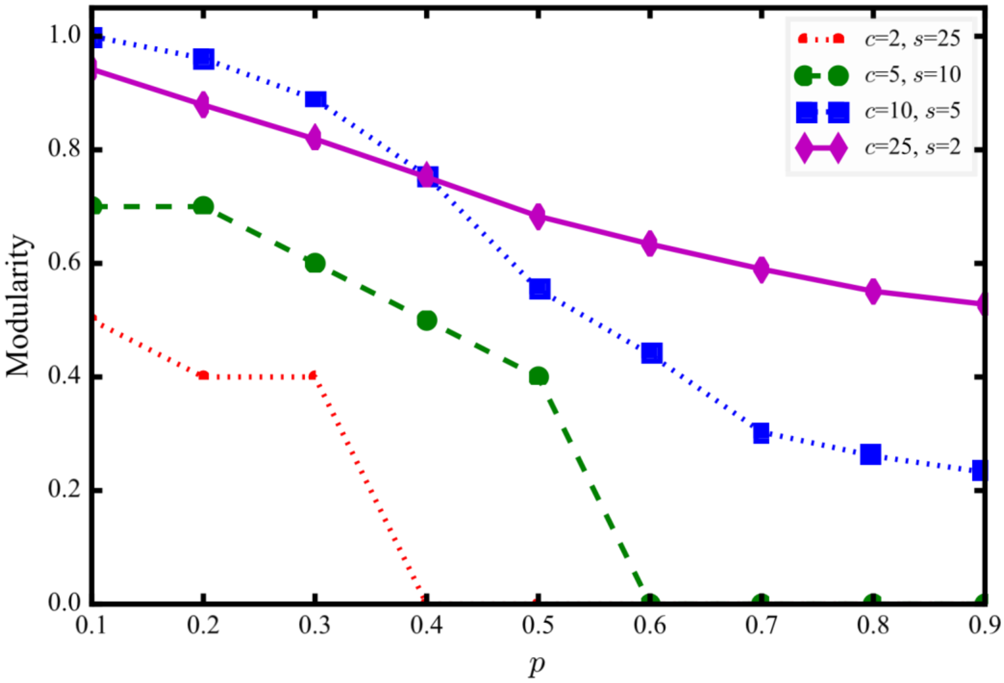


Figure 4.5: The measure modularity was used here to evaluate and compare the quality of clusters for different combinations of *c*, *s* and *p*.

When comparing the overall findings for different configurations of the parameters *c*, *s* and *p*, it is obvious that the parameter *p* has a major impact on the resulting cluster topology of a network, with the level of influence determined primarily by the combination of *c* and *s*. With this information, previous findings (e.g. Chin & Ratnavelu, 2016) can be updated regarding the influence of *p* on the resulting cluster output. Furthermore, the topology of the underlying network in this methodological approach has a considerable influence on the topology of the resulting material flow network. Thus, the creation of material flow networks with arbitrary properties is possible simply by using a different network model for the underlying network.

4.4 Summary

When developing this methodological approach, the main challenges faced were to determine what parameters should be used to define the cluster topology and what concept could be used to imitate the material flow resulting from job routing and processing. Initially, the following three parameters were identified to adequately describe the cluster topology: (i) the number of clusters *c*, (ii) the cluster sizes *s* and (iii) the degree of interconnection between individual clusters *p*. Next, random walks were identified as one possibility to simulate such dynamic job routing processes. Each random walk stands for an individual job. This involved adapting as many assumptions as possible from a given real data set. Initial evaluation results showed that by combining different values of the three parameters and simultaneously apply the random walk process, it was possible to achieve a wide range of synthetic material flow networks with varying cluster topology. Examples are networks with few large clusters ($c<s$) or many small clusters ($c>s$). Definitive statements about appropriate initial values for the parameters *c*, *s* and *p* can only be made if the particular use case is specified in further detail.

In summary, 36 different material flow network models, which varied primarily in their cluster topology, were generated and are available for further studies. Generally, with a few modifications this approach can be applied to create networks with arbitrary features. During the evaluation only the IM algorithm was used, but as stated in Chapter 3, other clustering algorithms may lead to different outputs. The next chapter demonstrates the differences using several clustering algorithms and shows how the consideration or neglect of edge weights and directions can influence the clustering results.

5 Edge Weight and Direction in Cluster Identification

As explained in Chapter 3, identifying clusters in material flow networks requires the user to select a specific algorithm from a variety of existing clustering algorithms. Algorithms differ in their underlying concepts (e.g. maximization or minimization of an objective function) to identify clusters. Some consider only the connections between nodes, while others include the edge weight in terms of material flow intensity and the material flow directions as relevant information (see Lancichinetti and Fortunato (2009b) for an overview). The challenge is that by considering such information, a clustering algorithm becomes increasingly complex, particularly in the case of directed networks, as described in Section 3.5. Therefore, users are restricted in their choice of an appropriate clustering algorithm. Nevertheless, the literature on complex networks recommends including both edge weights and edge directions, where these are available (Fortunato, 2010).

It has been insufficiently addressed to date whether the integration of edge weights and edge directions is a reasonable approach in all cases. This approach can affect the performance of a clustering algorithm and thus the output quality. This chapter is intended to show that the choice of an appropriate clustering algorithm mainly depends on the particular application and its requirements, and especially on the specific characteristics of a given network.

Using material flow networks (see Chapter 4), it is shown that from a network-theoretical perspective, a reduced representation of a network may be sufficient. A decisive benefit of using these material flow networks is that they clearly differ in their characteristics, namely their cluster topology. This allows for a wide-ranging evaluation. To assess the meaning and impact of additional information, the results of different clustering algorithms – which enable including the edge weights and/or edge directions, among other things – are evaluated and compared. The following sections describe the clustering algorithms applied (Section 5.1) and a presentation of the results (Section 5.2). This chapter closes with a brief summary (in Section 5.3).

5.1 Types of Clustering Algorithms

The literature presents several clustering algorithms that are able to consider edge weights and/or edge directions. The diverse combinations of the three parameters (number of clusters c, cluster size s and degree of interconnection between clusters p) lead to many use cases, of which only a few algorithms are applied here. Essential criteria for selection were that the consideration of edge weights and/or edge directions must be possible and that the underlying clustering mechanism must vary to cover a wide range of algorithms. Algorithms that take both edge weights and directions into account are rare. The clustering algorithms used in this work are, however, widely known, as they have been applied relatively often in relevant literature (e.g. Arratia & Mirambell, 2021; Lancichinetti & Fortunato, 2009b; Orman et al., 2012).

Table 5.1 provides an overview of the considered clustering algorithms. Fast Greedy (FG) and Louvain (LV) algorithms were created with the goal to maximize modularity. However, although they try to optimize the same objective function, the findings of Orman et al. (2012) and Dao et al. (2020) indicated that they obtain slightly different clusters. Furthermore, when selecting appropriate clustering algorithms to answer the corresponding RQs in this thesis, care was taken that only optional pre-settings were required for their application and that they would work on disconnected graphs. Therefore, the algorithms of Edge Betweenness (Girvan & Newman, 2002), Spinglass (Reichardt & Bornholdt, 2006) and Leading Eigenvector (Newman, 2006) were not included although they are extensively used as benchmark clustering algorithms.

Algorithm	**Concept**	**Directed edges**	**Weighted edges**
Fast Greedy (FG)	modularity-based	FALSE	TRUE
Louvain (LV)	modularity-based	FALSE	TRUE
Walktrap (WT)	random walk-based	FALSE	TRUE
Infomap (IM)	information theory-based	TRUE	TRUE
Label Propagation (LP)	diffusion-based	FALSE	TRUE

Table 5.1: Details of clustering algorithms used in this thesis, indicating whether edge weights and/or directions are considered.

Fast Greedy. The FG algorithm was originally proposed by Newman (2004) and was extended by Clauset, Newman, and Moore (2004) to make a faster version. It aims to optimize modularity. As its name implies, the greedy principle is applied. Based on the assumption that each node starts in its own cluster, the nodes are successively merged in further steps with the goal of optimizing the objective function. The clustering with the highest modularity value represents the best solution.

Louvain. The LV algorithm by Blondel, Guillaume, Lambiotte, and Lefebvre (2008) is another modularity-optimizing approach. As a first step, the greedy principle is used to identify initial clusters. In the second and last step, all nodes of each cluster are merged into a single node, and using these nodes, a new network is created. The greedy principle is again applied until no more modularity improvements are observed. By adding an extra step, LV is an improvement over the FG algorithm.

Infomap. The IM algorithm is explained in Section 4.2 in detail.

Walktrap. The WT algorithm was developed by Pons and Latapy (2006). Similar to the IM algorithm, the random walk process is applied. However, a series of random walks are used here to calculate the distance from neighboring nodes. It is generally expected that short random walks tend to stay in the same cluster.

Label Propagation. The LP algorithm was introduced by Raghavan, Albert, and Kumara (2007) and is based on the diffusion of information. First, each node is given its own label. Next, by an iterative process the label of each node is reassigned according to the labels of its neighboring nodes. The process ends when each node is given a label that the majority of its neighboring nodes have. All nodes with the same label are treated as a cluster.

5.2 Results

In this section, the evaluation results are presented and discussed. The objective is to understand how the clustering results differ from each other when edge weights and/or edge directions are considered. For this purpose, the NMI was selected as a similarity metric. To quantify the influence of edge weights, the quality of the resulting clusters was evaluated by assessing the modularity. This measure is applicable to weighted and unweighted networks, according to Arratia and Mirambell (2021). The results for NMI and modularity corresponded to the average over 15 material flow networks.

5.2.1 Edge Directions

First, the effects arising from edge directions were explored. The results of the IM algorithm for directed and undirected cases were evaluated and compared with each other. In Lancichinetti and Fortunato (2009b), the IM algorithm demonstrated superiority over other clustering algorithms when handling directed networks; hence, using IM as the only algorithm seemed sufficient to assess the significance of edge directions in this study. The following section further shows that the influence of edge directions depends largely on the consideration of edge weights.

	Directed, weighted vs. undirected, weighted networks				***Directed, unweighted vs. undirected, unweighted networks***			
c, s / ***p***	***2,25***	***5,10***	***10,5***	***25,2***	***2,25***	***5,10***	***10,5***	***25,2***
0.1	1	1	1	1	0.68	0.69	0.81	0.94
0.2	1	1	1	1	0.66	0.65	0.76	0.91
0.3	1	1	0.97	1	0.64	0.62	0.72	0.92
0.4	1	1	0.98	1	0.66	0.63	0.69	0.86
0.5	1	1	0.90	1	0.65	0.63	0.69	0.89
0.6	1	1	0.91	1	0.63	0.58	0.66	0.92
0.7	1	1	0.91	1	0.63	0.59	0.64	0.91
0.8	1	1	0.80	1	0.62	0.59	0.65	0.84
0.9	1	1	0.81	1	0.62	0.59	0.61	0.88

Table 5.2: Joint influence of edge weights and edge directions on the performance of IM algorithm using NMI as similarity metric. The influence of edge directions relies on considering the edge weights. Hence, in weighted networks, clusters are highly similar for undirected and directed cases. In the weighted case, the influence of p is almost non-existent (except for $c=10$ and $s=5$), whereas in the unweighted case the NMI values decrease with increasing p-value. The reason is the increasing blurriness of clusters when p increases. (Abbreviations: number of clusters, c; cluster size, s; degree of interconnection between individual clusters, p).

The results in Table 5.2 indicate that edge directions become more important when edge weights are removed, as evident in an NMI value deviating from 1. This can be explained by the fact that the IM algorithm detects substantially more clusters in the unweighted case than in the weighted case (see annexed Table A.1). For example, where *c=2* and *s=25* with *p=0.1*, the IM algorithm detects on average 25 clusters. With consideration of edge directions, the number of clusters is reduced to 14. However, with the results from Section 4.3 in mind, where the clusters of the underlying networks and the resulting material flow networks were compared, it can be reasonably assumed that the clusters from the unweighted networks do not correspond to the real clusters planted by the underlying network. Thus, edge weights appear to be important when identifying true clusters, at least when using IM algorithm. The next section provides further details. Furthermore, for unweighted networks, the combination of parameters *c* and *s* has considerable influence on the NMI values. In cases where *c>s* (i.e. many small clusters), the outputs become more similar, as evident in relatively high NMI values.

By comparison, with the same parameter setting (*c=2* and *s=25* with *p=0.1*), there were – as expected – only two clusters in the weighted network, irrespective of whether the edge directions were considered. However, when the parameters were set to *c=10* and *s=5*, there was a slight deviation from 1 for NMI values. Therefore, edge directions appear less important in this respect.

In summary, a distinction must be made between a weighted and an unweighted network. For the weighted case, edge directions can be neglected when using an IM algorithm. The influence of edge weights is examined in detail in the next section.

5.2.2 Edge Weights

This subsection examines how the clustering results of weighted networks differ when the edge weights are removed. The five clustering algorithms in Section 5.1 were performed to determine the influence of the edge weights.

As shown in Figure 5.1, LV was the only algorithm that was entirely insensitive to edge weights for all combinations of *c, s* and *p*. The FG algorithm and WT algorithm followed a similar pattern. As expected, the NMI value decreased with increasing *p*-values. This finding was due to the fact that clusters become blurrier for higher *p*-values and thus the results for the weighted and unweighted cases may differ significantly. Furthermore, Figure 5.1 indicates that although the FG and LV algorithms both had the optimization of modularity as a goal, they performed differently on weighted and unweighted networks.

The gap between IM and LP and the other algorithms widened as the *p*-value exceeded 0.4 and 0.6 for c<s. This is because from the point where the *p*-values are exceeded, all clusters end up in one giant cluster in the weighted networks. From that point onward, the NMI value drops to zero. For smaller *p*-values, the LP algorithm performed independently of whether edge weights were considered. The IM algorithm reacted very sensitively when edge weights were removed, for all values of *p*. For *c>s*, the trend of the curves of both algorithms were similar to the rest of the algorithms.

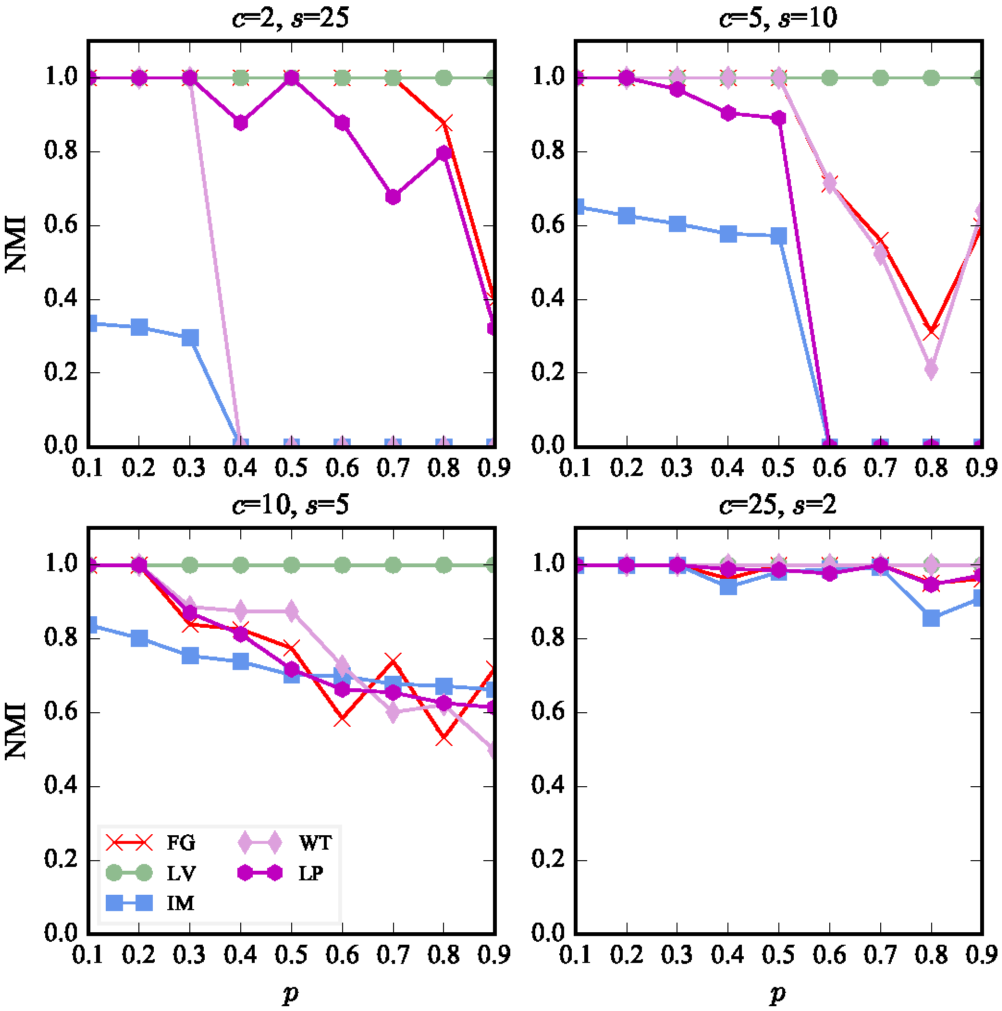


Figure 5.1: Comparison of the results for five algorithms. NMI was used to evaluate the similarity between the clustering results with and without considering the edge weights. Essentially, that the larger and fewer in number (c>s) the clusters, the more minor was the influence of edge weights. However, p had an increased impact on NMI values. IM, Infomap; WT, Walktrap; LP, Label Propagation; FG, Fast Greedy; LV, Louvain.

In summary, the influence of edge weights varied depending on what clustering algorithm was used. Furthermore, the development of NMI also depended on the setting of the parameters c, s and p. For a final evaluation, it was necessary to compare the quality of the resulting clusters from each clustering algorithm. Table 5.3 represents the results for modularity obtained by applying the five clustering algorithms on weighted and unweighted networks.

As evident in Table 5.3 and expected from the earlier findings, the IM algorithm reacted very sensitively to the removal of edge weights. In the unweighted case, the IM algorithm performed relatively poorly and the identified clusters showed a poor quality. Although the performance of IM improved with increasing p-values – which meant the clusters became blurry and thus the overall cluster quality decreased – existing edge weights should be considered when using IM. In the weighted case, particularly, IM performed as well as or better than the remaining algorithms.

	$p = 0.3$				$p = 0.7$			
	c=2,	c=5,	c=10,	c=25,	c=2,	c=5,	c=10,	c=25,
	s=25	s=10	s=5	s=2	s=25	s=10	s=5	s=2
FG	**0.38**	**0.66**	**0.66**	**0.94**	0.2	0.48	0.48	**0.91**
	(0.37)	(0.63)	(0.63)	(0.92)	(0.19)	(0.43)	(0.43)	(0.87)
LV	**0.38**	**0.66**	**0.66**	**0.94**	**0.21**	**0.51**	**0.51**	0.88
	(0.38)	(0.66)	(0.66)	(0.94)	(0.21)	(0.51)	(0.51)	(0.88)
IM	**0.38**	**0.66**	**0.66**	**0.94**	0.0	0.0	0.5	0.88
	(0.01)	(0.26)	(0.26)	(0.94)	(0.01)	(0.25)	(0.25)	(0.86)
WT	**0.38**	0.65	0.65	**0.94**	**0.21**	0.45	0.45	0.82
	(0.38)	(0.65)	(0.65)	(0.94)	(0.21)	(0.34)	(0.34)	(0.82)
LP	**0.38**	0.56	0.64	**0.94**	0.0	0.0	0.46	0.88
	(0.38)	(0.55)	(0.64)	(0.94)	(0.0)	(0.0)	(0.44)	(0.86)

Table 5.3: Comparison of modularity values from different algorithms. The larger the value of modularity, the better the cluster quality. For each network, the value for the unweighted case appears in brackets. The modularity value in bold indicates the best clustering algorithm in the weighted case.

In addition, Table 5.3 shows that taking edge weights into account when using FG and WT also improved the performance. Considerations of edge weights had no effect – or minimal effect – on the output performance of all other algorithms. Although the selected clustering algorithms reacted differently to the absence of edge weights, from a network theory perspective the consideration of edge weights seems reasonable.

As shown in Table 5.3, all algorithms performed well on weighted networks with well-defined clusters (low p-value). Only with increasing p, minor deviations arose in the performance, although no clustering algorithm was clearly identified as the best performing. One would expect the modularity-based algorithms to outperform the other algorithms. Indeed, the results indicated that LV and FG led to overall higher modularity values. However, the output of the IM algorithm was somewhat similar to these results. These conclusions were supported by the findings from the scenarios with the remaining p-values. A summary of all the modularity values is depicted in the appendix Table A.2.

To complete the performance comparison of the algorithms, the NMI matrices in Figures 5.2 to 5.5 provide interesting insights into how the results of the individual clustering algorithms differ from each other in the weighted case (weighted and directed in the case of IM). Based on the findings in Table 5.3, where the modularity values were highly similar for all algorithms in specific network models, it could be assumed that all clustering algorithms would yield similar results. The NMI matrices take all the same appearance. They indicate the level of similarity between different clustering algorithms for each network model. A value of 1 indicates that two clustering algorithms led to the same output. Each NMI matrix is symmetric around the diagonal.

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(a) *p*=0.1

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(b) *p*=0.2

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(c) *p*=0.3

	IM	FG	LV	WT	LP
IM	1	0	0	0	0
FG		1	0.9	0.9	0.9
LV			1	1	1
WT				1	1
LP					1

(d) *p*=0.4

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(e) *p*=0.5

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.9	0.9	0
LV			1	1	0
WT				1	0
LP					1

(f) *p*=0.6

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.8	0.8	0
LV			1	1	0
WT				1	0
LP					1

(g) *p*=0.7

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.7	0.7	0
LV			1	1	0
WT				1	0
LP					1

(h) *p*=0.8

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.4	0.4	0
LV			1	0.7	0
WT				1	0
LP					1

(i) *p*=0.9

Figure 5.2: NMI matrix for *c*=2 and *s*=25.

As evident from previous findings, the differences between the algorithms arose from the various combinations of *c*, *s* and *p*. For *c=2* and *s=25*, all algorithms showed the same results until the *p*-value exceeded *0.3* (compare Figure 5.2). Beyond that point, the IM algorithm could not identify any clusters. The same was true for LP for *p > 0.5*. The NMI values were 0. From *p > 0.4*, the FG also showed slight variations. The difference increased with increasing *p*-values. For *p=0.9*, there are no same clusters (apart from IM and LP as all clusters result in one giant cluster). Overall, the results in Figure 5.3 (*c=5* and *s=10*) corresponded to earlier results. This means that for *p > 0.5*, the clusters of FG and LP vary slightly. For *p=0.6*, all clusters end up in one giant cluster for IM and LP. The remaining algorithms led to different clusters. For *c=10* and *s=5*, the differences arose from *p=0.3* (see Figure 5.4). For smaller *p*-values, the resulting clusters showed only minor differences across the algorithms, with the differences increasing as the *p*-value increased.

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(a) *p*=0.1

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(b) *p*=0.2

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(c) *p*=0.3

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(d) *p*=0.4

	IM	FG	LV	WT	LP
IM	1	0.9	1	1	1
FG		1	0.9	0.9	0.9
LV			1	1	0.9
WT				1	0.9
LP					1

(e) *p*=0.5

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.6	0.5	0
LV			1	0.6	0
WT				1	0
LP					1

(f) *p*=0.6

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.5	0.7	0
LV			1	0.7	0
WT				1	0
LP					1

(g) *p*=0.7

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.4	0.5	0
LV			1	0.3	0
WT				1	0
LP					1

(h) *p*=0.8

	IM	FG	LV	WT	LP
IM	1	0	0	0	1
FG		1	0.4	0.5	0
LV			1	0.7	0
WT				1	0
LP					1

(i) *p*=0.9

Figure 5.3: NMI matrix for *c*=5 and *s*=10.

As shown in Figure 5.5, the most similar clusters were obtained for *c=25* and *s=2*. For *p=0.8*, the LV and WT algorithms yielded slightly different cluster results.

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(a) *p*=0.1

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(b) *p*=0.2

	IM	FG	LV	WT	LP
IM	1	0.9	0.9	0.9	0.9
FG		1	1	0.9	0.9
LV			1	0.9	0.9
WT				1	0.9
LP					1

(c) *p*=0.3

	IM	FG	LV	WT	LP
IM	1	0.7	0.8	0.9	1
FG		1	0.8	0.8	0.8
LV			1	0.8	0.9
WT				1	0.8
LP					1

(d) *p*=0.4

	IM	FG	LV	WT	LP
IM	1	0.8	0.8	0.8	0.8
FG		1	0.7	0.7	0.8
LV			1	0.7	0.8
WT				1	0.7
LP					1

(e) *p*=0.5

	IM	FG	LV	WT	LP
IM	1	0.8	0.7	0.8	0.7
FG		1	0.8	0.6	0.7
LV			1	0.5	0.7
WT				1	0.7
LP					0.7

(f) *p*=0.6

	IM	FG	LV	WT	LP
IM	1	0.7	0.9	0.8	0.8
FG		1	0.7	0.6	0.7
LV			1	0.7	0.8
WT				1	0.7
LP					1

(g) *p*=0.7

	IM	FG	LV	WT	LP
IM	1	0.6	0.8	0.7	0.7
FG		1	0.7	0.6	0.6
LV			1	0.6	0.7
WT				1	0.6
LP					1

(h) *p*=0.8

	IM	FG	LV	WT	LP
IM	1	0.7	0.8	0.8	0.7
FG		1	0.8	0.7	0.5
LV			1	0.7	0.5
WT				1	0.6
LP					1

(i) *p*=0.9

Figure 5.4: NMI matrix for *c*=10 and *s*=5.

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(a) p=0.1

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(b) p=0.2

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(c) p=0.3

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(d) p=0.4

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(e) p=0.5

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(f) p=0.6

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(g) p=0.7

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	0.9
WT				1	0.9
LP					1

(h) p=0.8

	IM	FG	LV	WT	LP
IM	1	1	1	1	1
FG		1	1	1	1
LV			1	1	1
WT				1	1
LP					1

(i) p=0.9

Figure 5.5: NMI matrix for c=25 and s=2.

In summary, the larger and fewer in number the clusters (c>s), the higher the dissimilarity across the algorithms. Consequently, in networks with many small clusters (c>s), the performance of the algorithms did not change considerably. In contrast, the impact of p was consistent for all combinations of c and s. As expected, with increasing p-values the dissimilarity between the clustering algorithms became more evident. Interestingly, although the modularity values deviated only slightly across the algorithms, the results (see Figures 5.2 to 5.5) showed that the resulting clusters differed depending on the respective combinations of c, s and p for the various algorithms. These results imply that it remains unknown which algorithm is superior. This uncertainty can be partly explained by the specific characteristics of the synthetically generated material flow networks. However, it is possible that clusters which can be identified by multiple algorithms are representative of the given network (Lancichinetti & Fortunato, 2009b). Hence, the results confirm that the choice of an appropriate clustering algorithm mainly depends on the particular application and its requirements, in addition to the characteristics of a given network.

5.3 Summary

It has been demonstrated that the edge directions have no or little impact on the resulting clusters when the IM algorithm is used with edge weights taken into consideration. Removing the edge weights strongly impacts the outcome of the IM algorithm and leads to clusters of poor quality. Although several authors have emphasized the importance of edge directions, especially in flow-based networks, the effect appeared negligible for the present material flow networks. Instead, edge weights must be viewed as the dominant edge characteristic – at least when using the IM algorithm.

In addition, the impact of edge weights was examined. Some clustering algorithms, such as IM, WT and FG, show different levels of sensitivity to the elimination of edge weights. For the remaining algorithms, the overall quality of the resulting clusters improved slightly or stayed the same when edge weights were included. From these results, it can be concluded that edge weights should be considered when available.

An unambiguous recommendation regarding the superiority of one algorithm in terms of cluster quality is not possible. Nonetheless, the results show that different algorithms lead to different clustering results depending on the network topology. However, overall, because of the characteristics of the given data sets, the differences between individual clustering algorithms were marginal. Because the findings in this chapter were based on the network theory perspective, the next step was to evaluate the clustering results through a simulation according to other logistic-based criteria. This analysis is described in Section 6.3.2.

6 Experimental Evaluation

This chapter is divided into several sections, with the simulation setup and the necessary assumptions made in this context described in Section 6.1. The simulation results presented in Section 6.2 illustrate the effects that arose given the varying cluster topology of material flow networks and how they were related to the logistic performance. Section 6.3 focuses on the evaluation of the hybrid clustering-based control approach. Using different material flow network models, which differed particularly in their cluster topology, the performance of several autonomous sequencing rules was compared (Section 6.3.1). These results serve as a benchmark for the hybrid clustering-based control approach proposed in this thesis (Section 6.3.2). In this context, the outputs of several clustering algorithms were evaluated from the logistic-based perspective. The individual simulation results were compared with results from the respective benchmark sequencing rule. This chapter concludes with a summary of all experimental results (Section 6.4).

6.1 Simulation Setup

For the subsequent simulation studies, the hypothetical manufacturing environment from Chapter 4 was used. Based on the assumptions described there, 34 different network models emerged, which differed primarily in their cluster topologies. More specifically, they were distinguished regarding their cluster size s, the number of clusters c and the degree of interconnection between individual clusters p. As a result of the varying cluster topologies, these network models were suitable for evaluating both the relationship between network topology and system performance and the effectiveness of the hybrid clustering-based approach. Wherever possible and reasonable, the assumptions were made according to the real data set. Due to inconsistency and lack of relevant data, it was necessary to make additional literature-based assumptions and simplifications. In the following paragraphs, basic assumptions are described that were necessary for the implementation of the simulation model. Additional modifications are listed in relevant sections of this chapter.

As a reminder: the examined manufacturing environment included 50 work systems, with each job having a random routing through the system. Overall, there were 3,702 jobs, each comprising a set of operations between 1 and 31. Both the number of work systems and the number of jobs significantly affect the computation time. Hence, the size of the system was smaller than in other research fields involving complex networks. As shown in Chapter 4, the network size was nonetheless sufficiently to create networks with varying cluster topologies and thus to achieve statistically significant results.

According to Law (2007), jobs were assumed to arrive randomly following a Poisson process to ensure the independent arrival of jobs, which is consistent with real systems. All arriving jobs entered a global buffer and were released according to their position in the buffer queue. This meant that a job in the first position in the buffer was released first. For the simulation experiments in Subsection 6.2.1, it was assumed that jobs were released directly after their arrival at the shop floor. To achieve a desired shop utilization level, a job release mechanism was applied in Subsection 6.2.2. The exact mechanism is explained in Section 2.2.

In all experiments, the processing time and due date of each job were determined after the job left the global buffer. Here, probability distributions were used to create a stochastic environment. In this way it was possible to evaluate the hybrid clustering-based approach in a dynamical environment arising from the use of probability distributions. The processing times were normally distributed with a mean of 60 minutes and standard deviation of 15 minutes. As is assumed in numerous studies, setup times and machine breakdowns were a part of the processing times (e.g. Holthaus, 1996). The due dates were assigned using the total work content (TWK) method, which is a simple and popular method (Baker, 1984; Kanet & Hayya, 1982). The due dates were created as a multiple of the processing times using a due date tightness factor. This factor determines how tight or loose the assigned due dates are (Lu & Liu, 2011); the larger the value of the factor, the looser the due date. By contrast, a low value of the factor means a tighter due date, which is more difficult to meet. As evident in literature, a due date tightness factor of 3 leads to tighter due dates than the value of 5, which leads to relatively loose due dates (e.g. Holthaus & Rajendran, 1997; Jayamohan & Rajendran, 2010). The due date tightness is also affected by the overall shop utilization rate (Baker, 1984). A high utilization rate leads to tight due dates even with a large due date factor. The corollary is that when the shop utilization rate is low, due dates tend to be loose even with a small due date factor. Derived from preliminary studies, a due date tightness factor of 3.5 was assumed for all experiments in this chapter.

So far, the assumptions described have related mainly to jobs. The assumptions regarding work systems must be complemented: There were no unforeseen interruptions, such as breakdowns or maintenance periods. Hence, all work systems were continuously available. Additional to the global buffer, a local buffer of unlimited size was added in front of each work system. This allowed jobs to build up queues in front of the corresponding resources and be processed according to their priority (Reményi & Staudacher, 2014).

As a rule, at the beginning of each simulation run the shop is empty. A shop thus reaches a steady state only after a set period (Robinson, 2014), which is referred to as the *warm-up period*. It was therefore appropriate to start with data collection at the end of a reasonable warm-up period to avoid the effects of the system warm-up that might otherwise bias the results. The warm-up period was set up for every network model individually. Furthermore, it was necessary to eliminate the *clean-up period* (Holthaus, 1996). During the clean-up, no new jobs enter the shop and the remaining jobs are completed. The time interval after the release of the last job is considered to be the clean-up period. Additionally, to reduce influences from the stochastic environment, this study employed several simulation runs (replications) with different seed values (Rabe et al., 2008). Twenty replications were performed.

6.2 Understanding the Topological Impact on System Performance

Modeling material flow systems as networks holds considerable potential to uncover hidden, non-trivial relationships between topology and logistic performance (Becker et al., 2012; Vrabič et al., 2013). The current research work is motivated in particular by Becker et al. (2012) and Liu, Li, Feng, and Rong (2013), who demonstrated the relationship between the overall efficiency of a logistics system and the various connectivity patterns among individual system entities. Becker et al. (2012) captured the topology of material flow networks using the average node degree of a network as single measure. Liu et al. (2013) described the connectivity pattern of supply networks considering the clustering topology.

The relationship between cluster topology and the resulting performance has not yet been considered in detail for material flow networks. An understanding of this relationship could help to identify the main challenges so as to address them through appropriate shop floor management. As described in Section 4.3, the average node degree significantly affects the output of the IM algorithm and thus the resulting clusters. Hence, the average node degree not only reveals much about the interaction between individual work systems but also allows conclusions to be drawn regarding the cluster topology. Specifically, a high average degree may be associated with few large clusters and a low average degree with many small clusters. However, the average degree does not allow any conclusions to be drawn about the cluster quality and the respective system behavior.

In the following sections, simulation experiments are reported, in which the cluster topology resulting from logistic processes and its relationship to performance were evaluated. In concrete terms, in the first step the objective is to show how the cluster topology of material flow networks and the resulting shop utilization level are related (6.2.1). The second step demonstrates how much average WIP is needed to achieve a desired shop utilization level according to the given cluster topology (6.2.2).

6.2.1 Cluster Topology and Utilization Rate

This subsection examines the relationship between the cluster topology of different material network models and the shop utilization rate. The utilization of a shop as a key measure was chosen because theory indicates it has a significant impact on logistic performance, e.g. on throughput time (Nyhuis & Wiendahl, 2012). A high utilization rate, for example, leads to long waiting times and thus to higher average throughput times of the jobs. In addition, the level of shop utilization and the associated average queue lengths influence the effectiveness of control algorithms (Holthaus, 1996). For example, low utilization levels may lead to short average queue lengths, which in turn may reduce the performance difference between the various control algorithms. A system should thus prevent overloading to keep the average queue length short; the system should also prevent underloading to yield sufficiently long queues and provide a choice between several jobs in the queue.

First, it is shown how the three parameters c, s and p (which are used in this thesis to determine the cluster topology) were related to the shop utilization. To study the influence of the topology accurately, the jobs in this study were processed on a FCFS basis. The simulation results are shown in Figure 6.1.

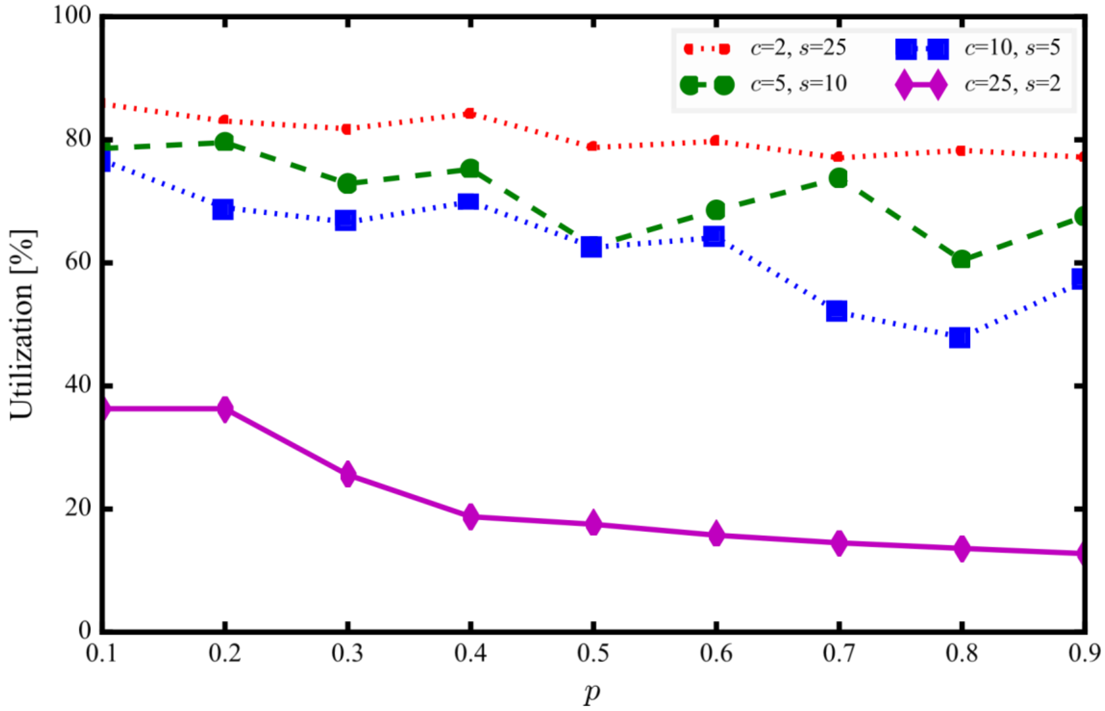


Figure 6.1: Mean shop utilization rate for different material flow network models using FCFS rule.

As shown in Figure 6.1, the mean shop utilization rate depended strongly on the parameters the number of clusters *c* and their sizes *s*. The highest mean utilization values were achieved for *c=2* and *s=25*, and the lowest mean utilization values were for *c=25* and *s=2*. Furthermore, the mean utilization dropped slightly as *p* increased (which meant clusters became blurrier and thus the overall cluster quality generally decreased). A slightly higher decline in mean utilization was evident for *c=25* and *s=2*.

The above point can be explained with reference to Figure 6.2. Figure 6.2 shows how the utilization of individual work systems varied for different combinations of *c* and *s* in dependency of *p*. The impact of the cluster topology is highlighted through two extreme scenarios (*p=0.1* and *p=0.9*; see appendix Figures A.1 to A.4 for the results of all other scenarios). For *c=2* and *s=25*, the work systems were more or less equally utilized; for *c=25* and *s=2,* it was evident that a few work systems played a leading role in the process – the term bottlenecks here. This was reinforced as the *p*-value increased. A reasonable explanation is that due to the bottlenecks there were not enough jobs to keep the remaining work systems busy. As a result, idle times might occur, which reduced the utilization rates. The result was that the mean utilization rate for *c=25* and *s=2* was by far the lowest. Hence, it appears that as the *p*-value increases, the role of individual work systems changes, and a kind of key work system emerges across all network models, which represents connecting parts between the individual clusters and thus processes most of the operations.

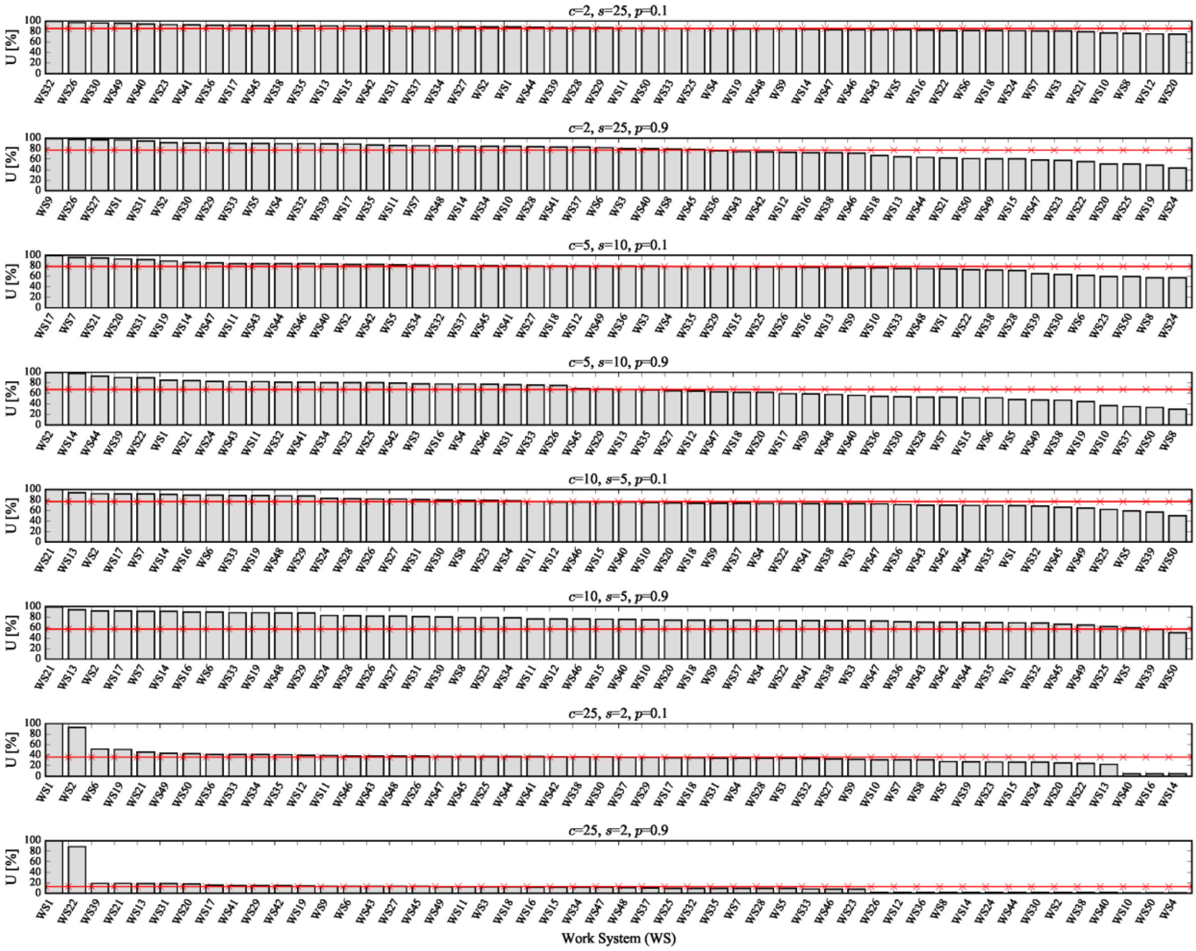


Figure 6.2: The impact of parameters *c*, *s* and *p* on the underutilization and overutilization of individual work systems, in descending order. It is assumed that if the utilization rate of a work system exceeds the overall mean utilization rate, this system is a bottleneck (as described in Raman, Talbot, & Rachamadugu, 1989). The horizontal red line represents the mean of the data.

In summary, there is a clear connection between network topology and shop utilization. Material flow networks with a few large clusters (*c=2* and *s=25*) indicate effective and consistent shop utilization. This is only slightly affected by increased blurriness of the cluster topology. By contrast, many small clusters (*c=25* and *s=2*) indicate that only a few work systems show a high level of utilization. This was reinforced by increased blurriness between clusters, leading to an overall low mean utilization rate of the shop.

6.2.2 Cluster Topology, Utilization and WIP

This section discusses the relationships between cluster topology, utilization rates and WIP. To quantify the effectiveness of the proposed hybrid clustering-based approach on different network models, the simulation studies assumed a fixed utilization level. The advantage of a fixed utilization is that it makes the results of the different network models comparable. Here, a fixed utilization level for the entire shop was set, with two levels of shop utilization assumed. The lower utilization level was determined mainly by the topological features of *c=2* and *s=25*. In this context, a theoretical utilization rate of 85 % is regarded as a moderate level and 90 % as a high level of utilization (Holthaus, 1996). Such utilization levels ensure that sufficiently long queues are created, which are necessary for the sequencing of jobs (Kiran & Smith, 1984).

To ensure that the planned shop utilization is achieved, ConWIP was used (for brief explanations see Section 2.2). This involved keeping the mean WIP, measured in hours, at a constant level for the duration of a simulation. In this way, all work systems were limited to the same time, and a job was released only when another job left the shop. The job release mechanism is especially suitable for small to medium series production with diverse variants and a high material flow complexity; hence, it was ideally suited to the production environment on hand. It is known that high shop utilization implies high average WIP (Hopp & Spearman, 2011). Therefore, the idea was to achieve the planned utilization levels with the minimum amount of WIP, to limit the throughput times.

To investigate what amount of WIP was necessary to reach the desired utilization levels of 85 % and 90 % for the individual network models, the normalized average WIP was introduced. The normalization was necessary to make the systems comparable (Schäfer, Chankov, & Bendul, 2016) and was achieved by

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{6.1}$$

where the observed value x is the current mean WIP, while x_{min} and x_{max} are the minimum or maximum mean WIP values among all network models. The normalized average WIP level of a corresponding network model is then scaled between 0 and 1. Figure 6.3 shows the development of the normalized average WIP level for different network models having different cluster blurriness (p). As expected from previous results, the lowest WIP level was required for *c=2* and *s=25*. With an increasing p-value, on average more WIP was needed. This was because the individual work systems were subject to varying degrees of demand, so that more jobs must be released to achieve a higher overall utilization. Furthermore, for *c=25* and *s=2*, the planned utilization of 90 % was not reached for all p-values, because (as noted in Section 6.2.1), only a few work systems were heavily utilized, whereas others remained idle. Due to the higher average WIP, the other work systems also became busier, but only for a short time, because there were still no long queues waiting for these work systems.

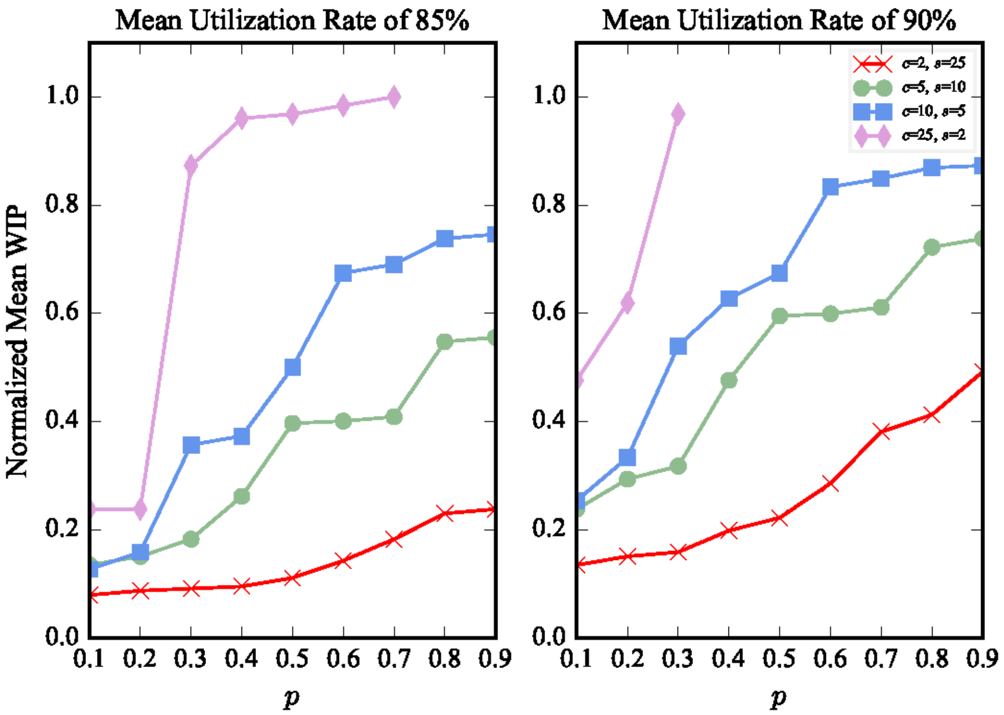


Figure 6.3: The normalized mean WIP values for various network models to achieve a desired shop utilization rate. For comparison of the network models, the normalized average WIP is used. There is a monotonous but nonlinear relationship between the normalized average WIP level and *p*.

In general, the results were highly similar to those in the previous section. There is a clear dependence of the cluster topology on the average WIP. The smaller and more numerous the clusters, the higher the average WIP. Furthermore, limitations regarding achievable utilization levels arose from the features of individual network models. Hence, in further simulation studies, a mean utilization rate of 85 % was assumed.

In Figure 6.4 the mean utilization levels of individual work systems are presented for *p=0.1* and *p=0.9* after applying ConWIP (see appendix, Figures A.4 to A.8 for details). In general, an improvement in the mean utilization of individual work systems is observed due to higher average WIP levels. The number of highly utilized work systems has increased for all combinations of *c* and *s* (for comparison see Figure 6.2). At first glance it comes as surprise that the number of highly utilized work systems increased with higher *p*-values. Conversely, it seemed reasonable that the number of low utilized work systems would increase with increasing *p*, as this effect was previously observed (see Figure 6.2). However, this development might be explained by the fact that the higher average WIP, which becomes necessary with a rising *p*-value to reach the desired mean shop utilization, implies that the work systems that were previously well-utilized increase in demand and simultaneously the number of work systems that most of the time remain idle is rising.

For *c=25* and *s=2*, the distribution showed an opposite trend. The combination of higher average WIP level and higher *p*-value led to a light smoothing of the utilization rates of individual work systems (Figure 6.4).

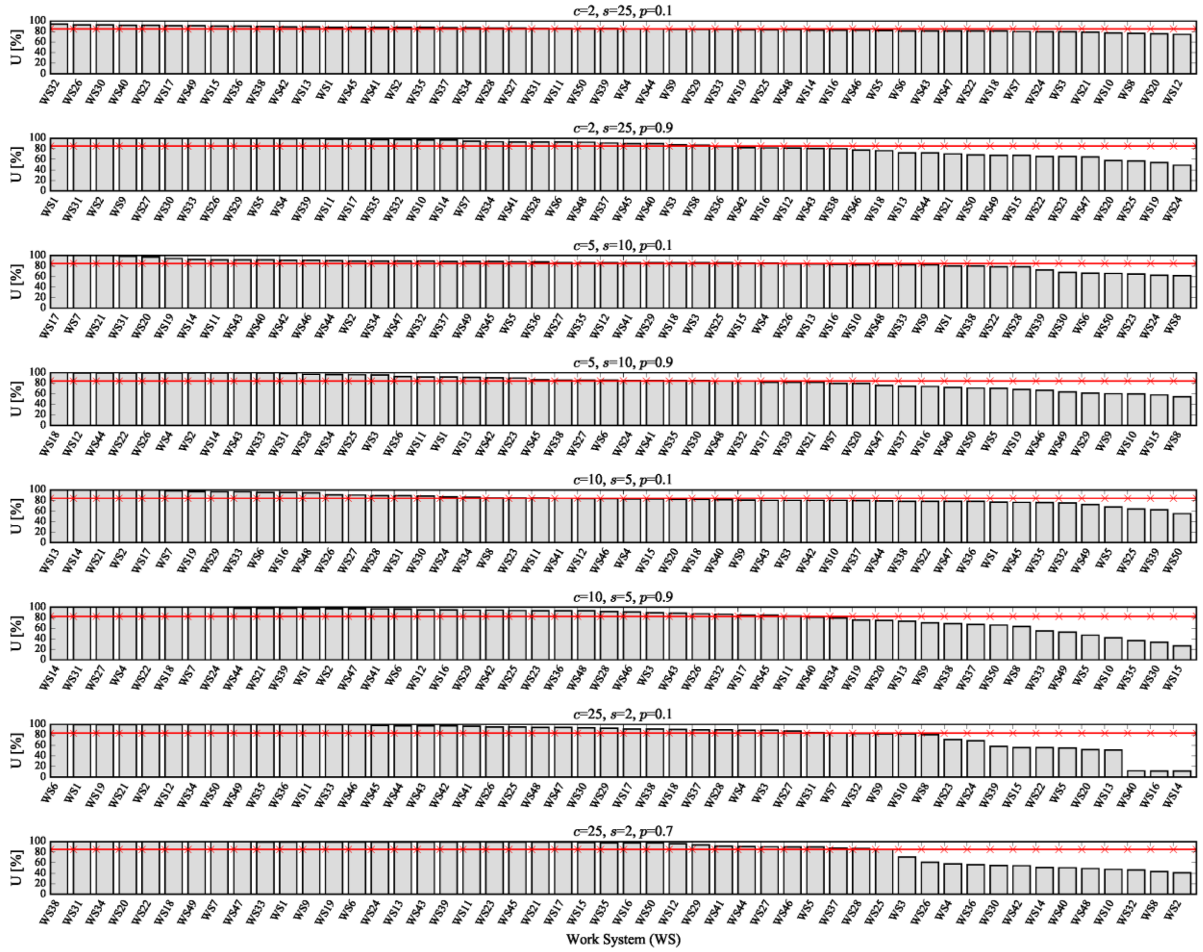


Figure 6.4: The impact of the parameters *c*, *s* and *p* on the mean utilization rate of individual work systems in descending order for *p=0.1* and *p=0.9* (or *p=0.7* for *c=25* and *s=2*) after applying CONWIP. The horizontal red line represents the mean of the given data.

The results of the simulation experiments indicate a clear relationship between the cluster topology of material flow networks and the resulting performance. The experiments confirmed that network models with a clear cluster topology (here, *c=2* and *s=25*) require less mean WIP to achieve the corresponding utilization levels. However, the smaller the clusters and the more plentiful they are (*c>s*), the larger the required WIP.

These results reflect those of Becker et al. (2012), who also found a relationship between network topology (represented by average degree) and logistics performance, inter alia, the corresponding WIP values. However, they found a non-monotonous relationship, whereas here monotonous, but nonlinear relationship between the cluster topology and the mean WIP values could be observed. This means that the blurrier the clusters become, the greater the required WIP. Liu et al. (2013) applied two network models with different clustering behavior (i.e. clusters were existent or non-existent). Their results showed that the network model with well-defined cluster topology positively affected the performance. Here, it was also shown that regardless of the corresponding combination of *c* and *s*, for lower *p*-values (where clusters are well defined) less mean WIP was required to achieve the target utilization level.

6.3 Evaluation of Hybrid Clustering-based Approach

This section aims to demonstrate the extent to which the hybrid clustering-based control approach performs better than the use of a single sequencing rule for the entire system. There is also a need for deeper knowledge about how networks with different cluster topologies specifically impact the ability to be managed using the hybrid clustering-based approach. This examination was achieved in a two-step procedure (see Figure 6.5).

First, preliminary simulation studies were conducted to examine the behavior of selected sequencing rules in a range of network models with varying cluster topology. In this thesis, job tardiness-based criteria were chosen due to their importance for the real-world manufacturing environment. The first objective was thus to test the effectiveness of the selected autonomous sequencing rules to reduce the mean tardiness of tardy jobs MT_T (Kiran, Alptekin, & Kaplan, 1991) and enhance the schedule reliability SR (Lödding, 2013). The mean tardiness of tardy jobs is calculated as follows:

$$MT_T = \frac{\sum_{i=1}^{n} {}^{T_i}/_{n}}{n_T}, \tag{6.2}$$

where

T_i tardiness of job i,
n number of all jobs,
n_T actual number of tardy jobs.

A job is considered tardy if the completion occurs after the due date (Lödding & Kuyumcu, 2015). From this definition, sequencing rules that result in only a few late jobs, but those jobs being notably late, should be penalized (Kiran et al., 1991).

The mean tardiness only considers the lateness of jobs, although completion before the due date is also undesirable (Lödding, 2013). Hence, schedule reliability was used as a second performance measure. It is defined as the percentage of jobs completed within a defined tolerance (Lödding & Kuyumcu, 2015). The relevant equation is

$$SR = \frac{\sum_{i=1}^{n} O_{i,tw}}{n} * 100\%, \tag{6.3}$$

where $O_{i,tw}$ is a job i completed within a given tolerance window, and n is the number of all jobs. Hence, some flexibility is provided, and the in-time completion of jobs is combined by penalizing both early and late completion. There is no general way to set the size of the tolerance window, which depends on many factors (e.g. due date tightness factor). Here, the size of the tolerance window was set to 15 %. For clear observation of the impact of the various autonomous sequencing rules and the hybrid clustering-based approach on the two performance measures, each measure was considered separately. To compare the results of the different network models with each other, the two performance measures were normalized in line with Equation 6.1.

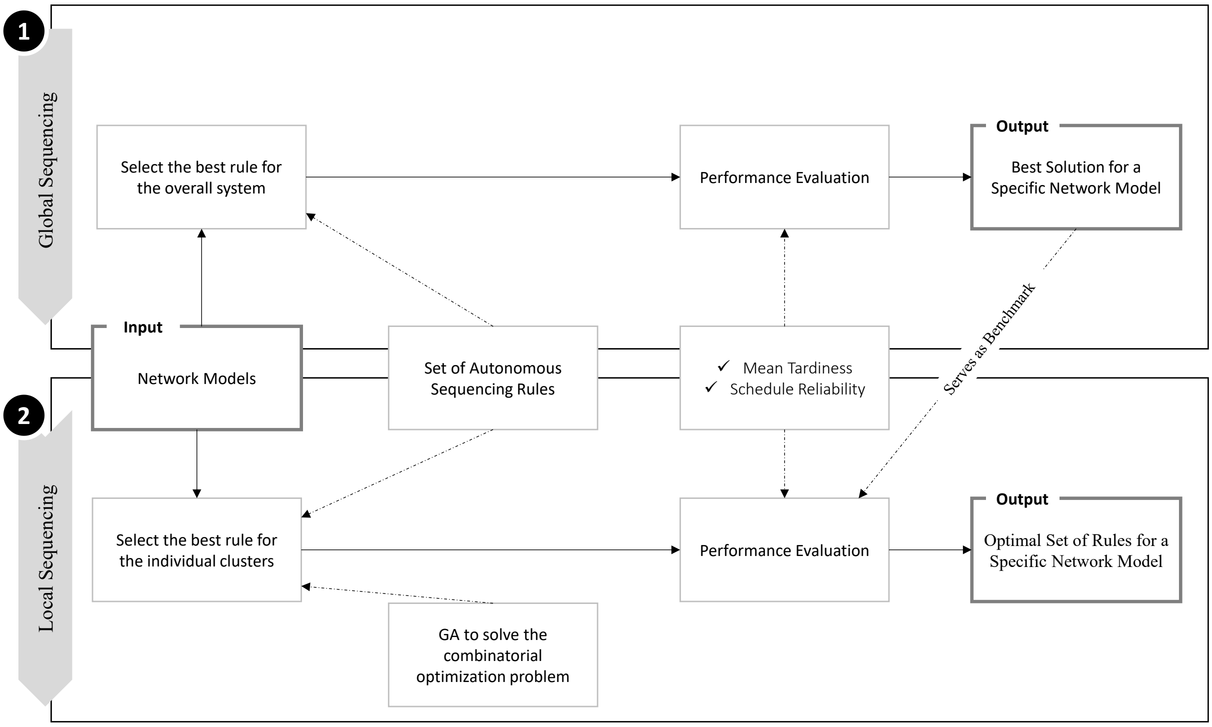

1
Global Sequencing
Select the best rule for the overall system
Performance Evaluation
Output
Best Solution for a Specific Network Model
Input
Network Models
Set of Autonomous Sequencing Rules
✓ Mean Tardiness
✓ Schedule Reliability
Serves as Benchmark
2
Local Sequencing
Select the best rule for the individual clusters
Performance Evaluation
Output
Optimal Set of Rules for a Specific Network Model
GA to solve the combinatorial optimization problem

Figure 6.5: Two-step procedure to evaluate the performance of the hybrid clustering-based approach. Global sequencing means that one autonomous sequencing rule is applied in the entire system. Local sequencing means one autonomous sequencing rule is applied in a cluster.

In this thesis, existing autonomous control algorithms – which were developed within CRC 637 – were not used because many of them assume there are several work systems of the same type in a manufacturing system. Hence, alternative routes arise. By contrast, this study assumed there was only one work system of each type, which means there are no alternative routes. Furthermore, many of these algorithms are look-ahead rules and use global system information for that. This is not necessary here, because the pre-clustering based on global system information means that autonomous clusters are created, which can make decisions based on local information. Therefore, sequencing rules are used here to specify an order for each work system for job completion.

The current simulation studies are limited to 12 frequently used autonomous sequencing rules. With each additional rule, the complexity increases, especially the computational effort. The first six rules (FCFS, LNOP, TSPT, SPT, SRPT and EDD) in Table 6.1 were chosen for due date–based objectives and their ability to consider different job-specific characteristics (e.g. due date, processing times and remaining number of operations). For practical implementation, these rules can be integrated with little effort and require no additional information regarding the overall system. The remaining six rules (LCFS, MNOP, TLPT, LPT, LRPT and LDD) in Table 6.1 do not correspond to the selection criteria mentioned above and are actually the preemptive version of the former rules. However, they are considered here for the sake of completeness.

Rule	Abbreviation	Description
First Come First Served	FCFS	select the job that is added first to the queue
Least Number of Operations	LNOP	select the job with the least remaining number of operations in the queue
Total Shortest Processing Time	TSPT	select the job with the least total processing time in the queue
Shortest Processing Time	SPT	select the job with the shortest processing time in the queue
Shortest Remaining Processing Time	SRPT	select the job with the shortest remaining processing time in the queue
Earliest Due Date	EDD	select the job with the earliest due date in the queue
Last Come First Served	LCFS	select the job that is added last to the queue
Most Number of Operations	MNOP	select the job with the most remaining number of operations in the queue
Total Longest Processing Time	TLPT	select the job with the longest total processing time in the queue
Longest Processing Time	LPT	select the job with the longest processing time in the queue
Longest Remaining Processing Time	LRPT	select the job with the longest remaining processing time in the queue
Latest Due Date	LDD	select the job with the latest due date in the queue

Table 6.1: Details of the sequencing rules used in this thesis. Other terminology may exist in the literature for the same rules listed here.

Usually, such heuristic approaches are compared with the optimal solution. Sequencing is in general NP-hard (Holthaus, 1996) and thus an optimal solution can only be determined for small problem instances. However, the given manufacturing environment with a high number of jobs and work systems is not considered a small problem instance. Hence, the individual heuristics were compared with each other. The results of the best-performing rule then served as a benchmark for studying the hybrid clustering-based approach.

To test the statistical significance of performance differences between the individual rules, a paired t-test was conducted (Law, 2007). A paired-t-test is performed to confirm the significance between two mean values. The values of interest here were derived from 1) two autonomous sequencing rules or 2) the benchmark sequencing rule and the hybrid clustering-based approach. For each material flow network model, the best sequencing rule was found, and all remaining rules were compared pairwise against this rule. For the tests a 95 % confidence interval was assumed.

Second, as evident in Figure 6.5, the second and last step consists of evaluating the hybrid clustering-based approach. To test which autonomous sequencing rule combinations led to improved performance for the given material flow model, all possible combinations were tested; the combination yielding the highest increase in performance was selected. With a small number of clusters (e.g. $c=2$ and $s=25$), it works well. For two clusters and 12 rules, there were 144 possible combinations. Increasing the number of clusters increased the number of possible combinations: for five clusters and 12 rules, there were 248,832 possible combinations. It was therefore no longer possible to calculate all possible combinations, given the high computational effort. Therefore, a genetic algorithm (GA) was used on the assignment of autonomous sequencing rules to individual clusters. GAs are particularly well suited where no optimal solution is available or many possible combinations arise because of a wide range of parameters (Shahsavar, Najafi, & Niaki, 2011). Both conditions occurred here. However, there is no guarantee that the solution is the best possible, without having tried all combinations beforehand (Kramer, 2017).

GAs are based on the reproduction process of living beings. A brief description of the principal function of a GA follows; for further details, the reader is referred to Eiben and Smith (2015). To find a near optimal solution, several iteration loops are required. First, an initial population is randomly created with possible solutions. Therefore, at the beginning, each cluster is randomly assigned a rule. To find the best solution for each iteration, a fitness score is calculated; in the current study, these were the two performance measures. To improve the solution, the population is newly created. This process is repeated until no more improvement is observed.

After having given all the necessary information for evaluation. The results of the simulation studies are presented and discussed in the next subsections.

6.3.1 Efficiency of Various Autonomous Sequencing Rules

This subsection demonstrates how the individual sequencing rules performed in material flow networks with varying cluster topologies, for a level of 85 % shop utilization. The results were evaluated regarding the mean tardiness of tardy jobs and schedule reliability, with a 15 % tolerance window. The values of the performance measures were determined by calculating the mean values over 20 replications.

Figures 6.6 and 6.7 show the results of the two performance measures for the best autonomous sequencing rule of a particular network model. In the result presentation, only the best rule of each network model is considered. This approach makes the result presentation clearer and focuses on the most interesting rules for further simulation studies.

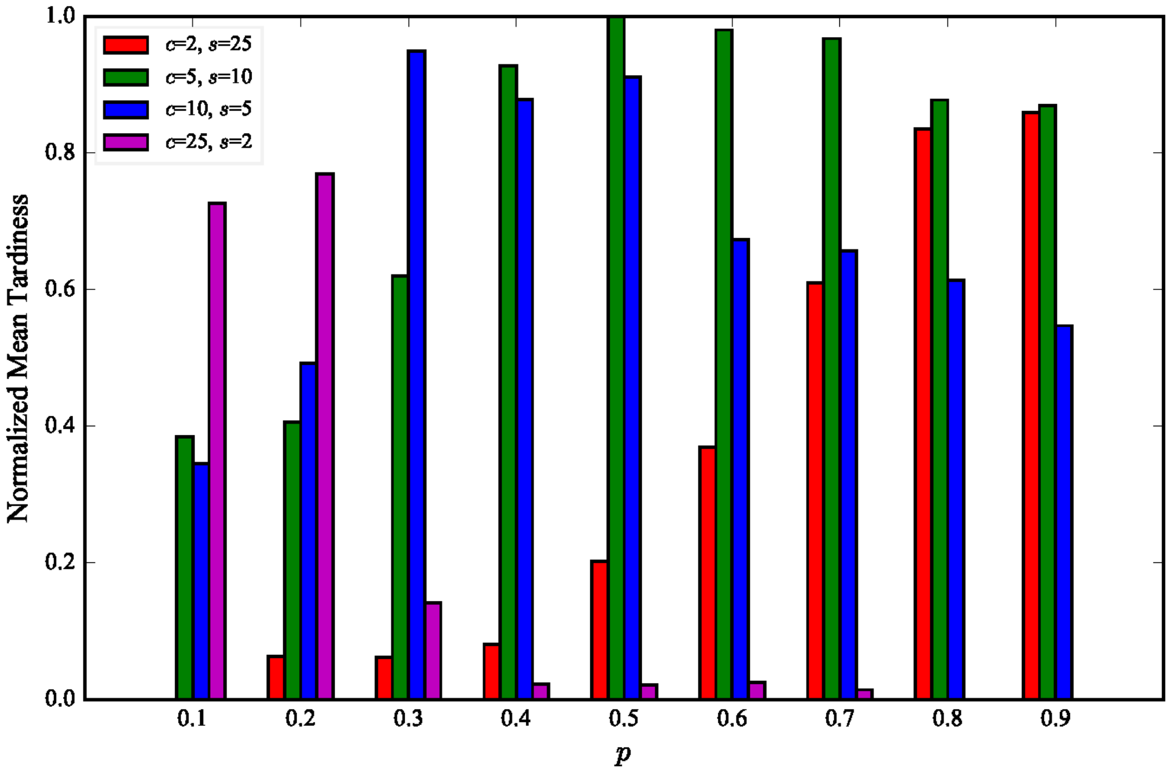


Figure 6.6: The mean normalized tardiness per tardy job for the best rule of each network model.

In moderate shop utilization (85 %), the influence of *p* is clearly observed for the different combinations of *c* and *s*. As expected from preliminary studies (see Section 6.2), for *c=2* and *s=25* and *p*=0.1, the normalized mean tardiness was by far the lowest value. With increasing *p*-values, the value increased rapidly. From the start – i.e. for low *p*-values – the normalized mean tardiness for the remaining combinations of *c* and *s* was much higher than for *c=2* and s=*25*. Furthermore, for *c=5* and *s=10* and *c=10* and *s=5,* the normalized mean tardiness increased with increasing *p*-values until it reached a maximum around *p=0.5* and *p=0.3*. Subsequently, the values gradually decreased again, i.e. the normalized mean tardiness was reduced. For *c=25* and *s=2*, the development of the normalized mean tardiness did not follow the general trend.

In short, the value of the normalized mean tardiness was highest for small *p*-values and fell abruptly at *p=0.3*. This means that increasing *p*-values were associated with lower normalized mean tardiness.

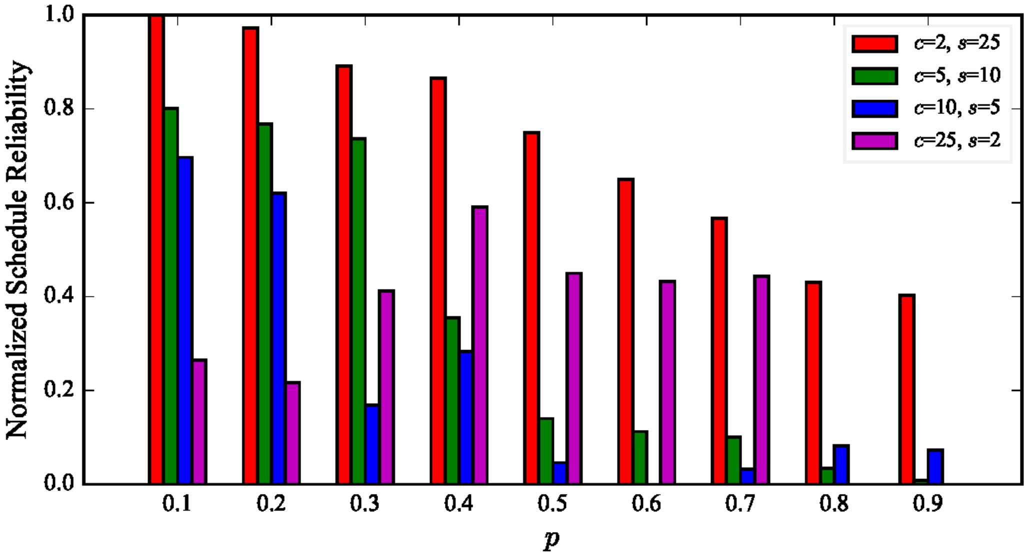


Figure 6.7: Normalized schedule reliability for the best rule of each network model.

Similar to the previous results regarding tardiness, the normalized schedule reliability was highest for *c=2* and *s=25* (see Figure 6.7). However, with increasing *p*-values, the smaller in size and larger in number were the clusters (*c* > *s*), the lower was the normalized schedule reliability. For *c=25* and *s=2*, again an opposing trend was observed: with increasing *p*-value, the normalized schedule reliability increased steadily, reaching a high at *p=0.4* and declining thereafter.

In summary, there were only minor differences between the results of the two performance measures. The characteristic patterns of the different combinations of *c* and *s* can be related to the results of preliminary studies in Section 6.2.2. The number of very high and very low utilized work systems strongly increased for all combinations of *c* and *s* (excluding *c=25* and *s=2*) for higher *p*-values. This means that in cases where the shop utilization is balanced – which is the case for low *p*-values with well-defined clusters – the best values are achieved for the performance measures. For this reason, the development of the performance measures shows an opposite trend for *c=25* and *s=2*, where higher *p*-values lead to a light smoothing of the utilization rates of individual work systems. Systems with highly utilized work systems have relatively long waiting queues, which creates the potential to select and process a suitable job first. However, the long waiting times have a negative effect on the performance measures. Hence, the outcomes for performance measures were guided by the cluster topology and underlying utilization of single work systems.

This section has presented the results of the best rule for each model network. The next step is to examine the frequency of an autonomous sequencing rule to be identified as the best rule. In Figures 6.8 and 6.9, the relative number of times a certain autonomous sequencing rule outperformed the others is summarized.

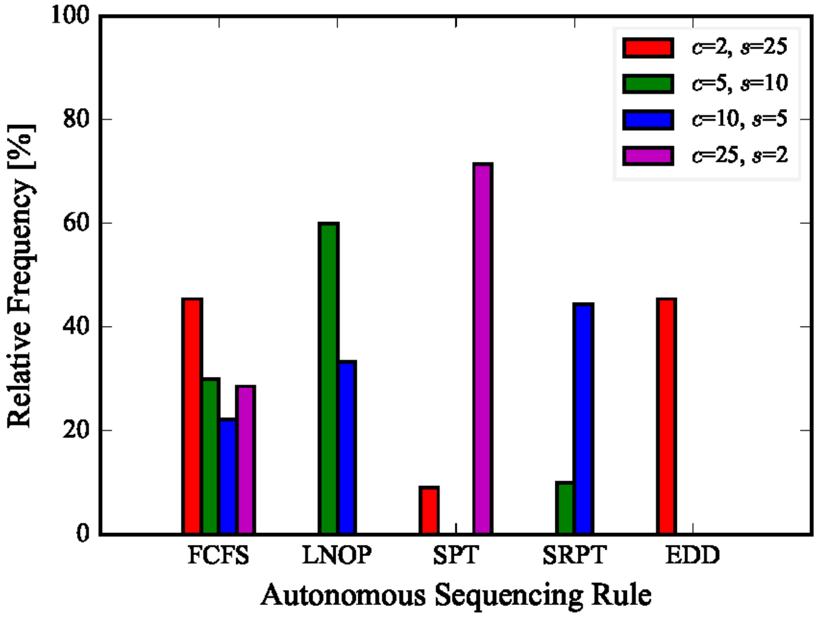


Figure 6.8: Relative frequency of autonomous sequencing rules that led to top ranking for mean tardiness.

In summary, the experimental results showed there were five autonomous sequencing rules (FCFS, LNOP, SPT, SRPT and EDD) that were ranked as a best rule (see Figure 6.8). For *c=25* and *s=2*, the rules SPT (71 %) and FCFS (29 %) were ranked first. For the remaining combinations of *c* and *s*, three rules were ranked first. For *c=2* and *s=25*, they were FCFS (45 %), EDD (45 %) and SPT (10 %). Furthermore, the representation of the relative frequencies in Figure 6.8 indicates that for *c=5* and *s=10*, the LNOP rule was ranked first in about 60 % of cases, FCFS in 30 % and SRPT in 10 % of cases. For *c=10* and *s=5*, SRPT outperformed all other rules in 45 % of cases. Moreover, LNOP was ranked first in 33 % and FCFS in 22 % of cases. The following conclusion can be drawn from the distribution of the relative frequencies: There is no single rule that outperforms all other rules. Interestingly, the FCFS rule was often ranked first across all combinations of *c* and *s*, as it essentially corresponds to a random selection of the next job. By contrast, in other cases it was one of the worst performing rules.

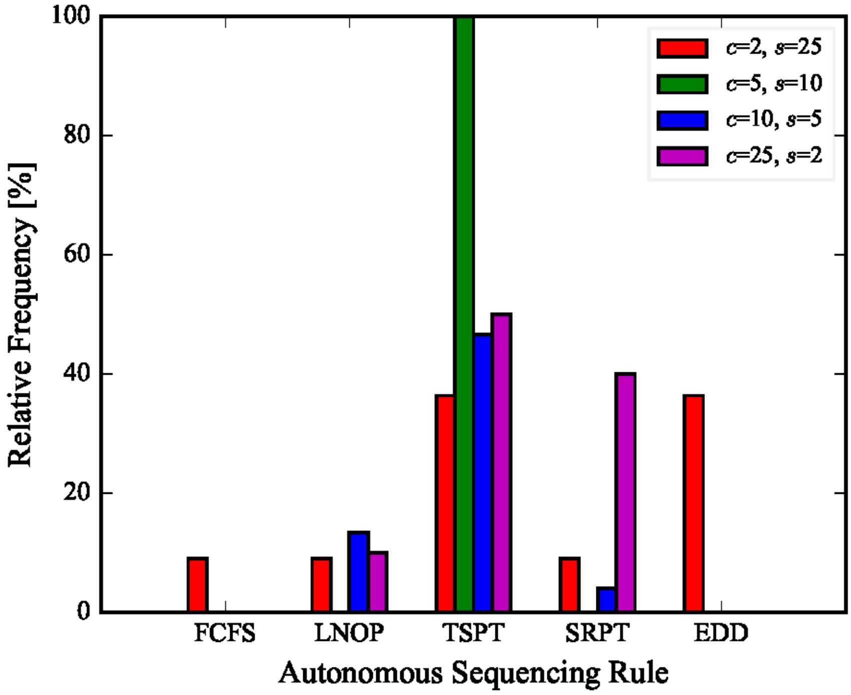


Figure 6.9: Relative frequency of an autonomous sequencing rule to be ranked first for schedule reliability.

The schedule reliability was evaluated, and the following five rules were ranked as the best rule: FCFS, LNOP, TSPT, SRPT and EDD (see Figure 6.9). This meant the SPT rule disappeared and the TSPT became more significant. At first glance, TSPT performed the best across all scenarios for *c=5* and *s=10*. For *c=10* and *s=5*, among all the autonomous sequencing rules, TSPT dominated for schedule reliability in 47 %, SRPT in 40 % and LNOP in 13 % of cases. For *c=25* and *s=2*, TSPT achieved the best the results in 50 %, SRPT in 40 % and LNOP in 10 % of cases. As shown in Figure 6.9, for *c=2* and *s=25*, all five autonomous sequencing rules appeared as best rules. The rules TSPT and EDD outperformed other rules in 36.5 %, respectively. For the remaining rules, they outperformed other rules in 36 % of cases.

To conclude, TSPT was the single rule that was ranked first most often across all combinations of *c* and *s*. Sometimes, multiple autonomous sequencing rules achieved the best value for a given network model and performance measure. This meant that several rules (here: 2 or 3) all achieved the top ranking. For mean tardiness it is in 8.82 % of cases and for schedule reliability in 26.47 % of cases.

These preliminary simulation studies determine the behavior of the selected autonomous sequencing rules in a range of situations, particularly regarding the cluster topology of different material flow networks. Apart from *c=5* and *s=10*, where TSPT was the dominant rule for schedule reliability, no rule outperformed all other rules in all cases. The remaining autonomous sequencing rules changed their behavior according to each network model and the observed performance measure. Hence, no specific recommendation can be made. The results of the best rule of each network model provide the benchmark for the following simulation studies.

6.3.2 Efficiency of the Hybrid Clustering-based Approach

Chapter 5 demonstrated the effects of additional information (namely edge weights and/or edge directions) on the resulting clusters when using different clustering algorithms. From a network theory perspective, edge weights should be considered when they are available. It has also been demonstrated that the edge directions have no or little impact on the resulting clusters when the IM algorithm is used with edge weights taken into consideration. Removing the edge weights has a strong impact on the outcome of the IM algorithm. From a production logistics perspective, edge weights are relevant, and considering them allows merging the work systems based on their interactions. Neglecting the edge weights leads to clusters where all edges have the same weight of 1 and are thus regarded as equal. This scenario does not reflect the true interactions. Hence, work systems would end up in the same cluster even if there was only single exchange of material between them. From a production logistics perspective, edge directions may be relevant, depending on what sequencing logic is used for a certain cluster. Emphasis has been given here to forward-looking sequencing rules, as it might be advantageous to place the subsequent work systems in one cluster. Therefore, the consideration of edge weights and edge directions must be carefully assessed, from both a network-theoretical and a production logistics perspective, as such considerations restrict the choice of available clustering algorithms. Moreover, no agreement could be reached on whether one algorithm was the best in terms of cluster goodness. Despite this, the results also showed that different algorithms led to slightly different clustering results, depending on the given network topology.

The findings were evaluated from a network theory perspective only. Hence, in this section, the clustering results are additionally evaluated for the ability to (i) provide an autonomous cluster to reduce the mean tardiness of tardy jobs and (ii) increase the schedule reliability, compared to the corresponding benchmark rule. The results of the best-performing clustering algorithm are then described in further detail. In this respect it is noted that the weighted versions of the clustering algorithms were used, and in the case of IM, the edge directions were also considered.

Analysis of Mean Tardiness over all Tardy Jobs

Figure 6.10 shows where the hybrid clustering-based approach performed better than the corresponding benchmark rule when using the clustering results of different algorithms. Overall, the clustering algorithms performed highly similarly. Most obvious were the differences for $c=5$ and $s=10$ and for $c=10$ and $s=5$. As mentioned earlier, the marginal differences between the individual clustering algorithms can be explained by the characteristics of the given data sets.

The findings for IM and LP need to be interpreted with care. For particular combinations of c and s, there seems to be an upper limit for p, above which all nodes end up in one giant cluster. It is evident that the LV algorithm performed slightly better. Therefore, the results for the LV algorithm are presented and discussed in detail below. The complete results appear the appendix (Table A.3).

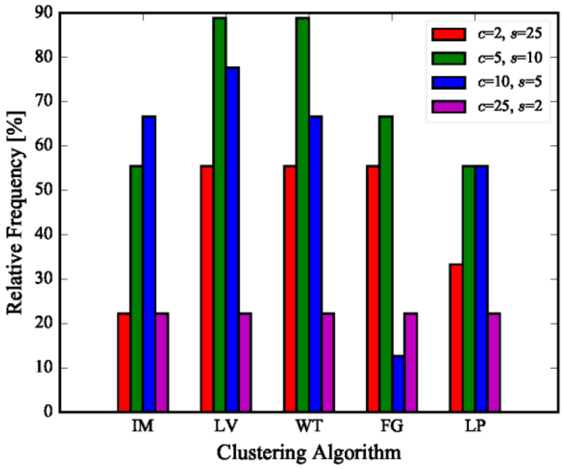


Figure 6.10: Relative frequency of the proposed hybrid clustering-based approach performing better than the benchmark rule. The variable of interest was the mean tardiness over all tardy jobs.

The results in Table 6.2 indicate that for *c=2* and *s=25*, in only three of nine cases did the best benchmark rule outperform the hybrid clustering-based approach. This was true especially for small and medium *p*-values. With the results from Figure 6.6 in mind, where it was described that mean tardiness is the lowest for *c=2* and *s=25* and small *p*-values, combined with the knowledge that this is a balanced manufacturing system (see Section 6.2.2), it seems reasonable that best autonomous sequencing rule dominates. When the hybrid clustering-based approach is applied in a system with balanced utilization, the clusters may be quite similar in their material flow pattern, which would mean a combination of several autonomous sequencing rules would not improve the performance. In 26.5 % of the cases in all scenarios, the hybrid clustering-based approach led to similar results for mean tardiness. By contrast, in 64.7 % of cases, the hybrid clustering-based approach outperformed the best rule.

Network Model (*c*, *s*, *p*)	**Best Rule**	**Hybrid Clustering-based Approach**	**Confidence Interval**	**Improvement [%]**
2, 25, 0.1	**571**	611	[-61.82; -18.58]	7.00
2, 25, 0.2	**846**	**861**	[-65.95; 35.55] (not significant)	1.80
2, 25, 0.3	**840**	891	[-69.52; -31.68]	6.10
2, 25, 0.4	**920**	953	[-53.12; -12.48]	3.60
2, 25, 0.5	3214	**2229**	[52.23; 253.77]	-30.60
2, 25, 0.6	2173	**1869**	[260.21; 348.59]	-14.00
2, 25, 0.7	3214	**2229**	[932.44; 1038.36]	-30.60
2, 25, 0.8	4292	**3404**	[529.05; 1048.95]	-20.70
2, 25, 0.9	4299	**3009**	[1185.26; 1395.54]	-30.00
5, 10, 0.1	2238	**1872**	[319.54; 412.06]	-16.40
5, 10, 0.2	2334	**1816**	[477.04; 559.76]	-28.50
5, 10, 0.3	3261	**2240**	[945.93; 1094.87]	-31.30
5, 10, 0.4	4592	**3759**	[734.62; 931.78]	-18.10
5, 10, 0.5	4907	**4309**	[352.12; 845.88]	-12.20
5, 10, 0.6	4821	**4610**	[-1538.29; -1158.91]	-4.40

Network Model (c, s, p)	**Best Rule**	**Hybrid Clustering-based Approach**	**Confidence Interval**	**Improvement [%]**
5, 10, 0.7	4766	**3867**	[774.82; 1023.58]	-18.90
5, 10, 0.8	**4377**	**4412**	[-137.73; 68.13] (not significant)	0.80
5, 10, 0.9	4343	**3966**	[206.17; 546.63]	-8.70
10, 5, 0.1	2070	**1737**	[227.56; 437.64]	-16.10
10, 5, 0.2	2706	**2159**	[481.81; 611.79]	-20.20
10, 5, 0.3	4686	**3724**	[858.54; 1064.26]	-20.50
10, 5, 0.4	4380	**3610**	[613.84; 928.16]	-17.60
10, 5, 0.5	4522	**4172**	[276.99; 423.41]	-7.70
10, 5, 0.6	3489	**3337**	[58.51; 246.29]	-4.40
10, 5, 0.7	3417	**3111**	[167.18; 442.82]	-9.00
10, 5, 0.8	**3231**	**3202**	[-39.8; 97.8] (not significant)	-0.90
10, 5, 0.9	**2944**	**2987**	[-150.81; 64.81] (not significant)	1.50
25, 2, 0.1	3721	**3583**	[64.06; 211.54]	-3.70
25, 2, 0.2	3909	**3637**	[184.53; 360.67]	-7.00
25, 2, 0.3	**1186**	**1226**	[-102.31; 22.31] (not significant)	3.37
25, 2, 0.4	**669**	**617**	[-5.06; 110.26] (not significant)	-7.80
25, 2, 0.5	**662**	**655**	[-89.91; 103.91] (not significant)	-1.06
25, 2, 0.6	**679**	**663**	[-34.51; 66.91] (not significant)	-2.40
25, 2, 0.7	**634**	**614**	[-9.82; 50.22] (not significant)	-3.20

Table 6.2: Computational results for mean tardiness [time units] of tardy jobs, using the LV algorithm. Values in bold indicate the approach that achieved the best performance. If there was no statistical difference between two approaches, both values are in bold. The non-overlapping confidence interval indicate that an approach was superior in the considered scenario. The improvement over the next-best autonomous sequencing rule is reported in the last column.

For *c=2* and *s=25*, five of the nine cases were dominated by the proposed approach, indicating significant improvement over the benchmark rule. For *c=5* and *s=10*, the hybrid clustering-based approach was superior in eight of nine cases. For *c=10* and *s=5*, in seven out of nine cases. Finally, for *c=25* and *s=2*, in four out of seven cases. However, it cannot be concluded from these results alone whether – and when – the hybrid clustering-based approach would lead to a significantly lower mean tardiness of tardy jobs. The results in Figure 6.11 should be useful for practitioners to make a choice.

Based on the above description of the results, it is surprising that for both *c=2, s=25* and *c=5, s=10*, the hybrid clustering-based approach achieved the greatest improvements, namely almost 30 % (see Figure 6.11). A possible explanation is that with increasing *p*, the shop becomes more unbalanced and following a distinction between the different clusters regarding their structural properties emerge, and hence a combination of several autonomous sequencing rules leads to significant better results, whereas for *c=25* and *s=2*, the improvement is the lowest. In conclusion, in network models having many small clusters (*c*>*s*), the potential for improvement decreases.

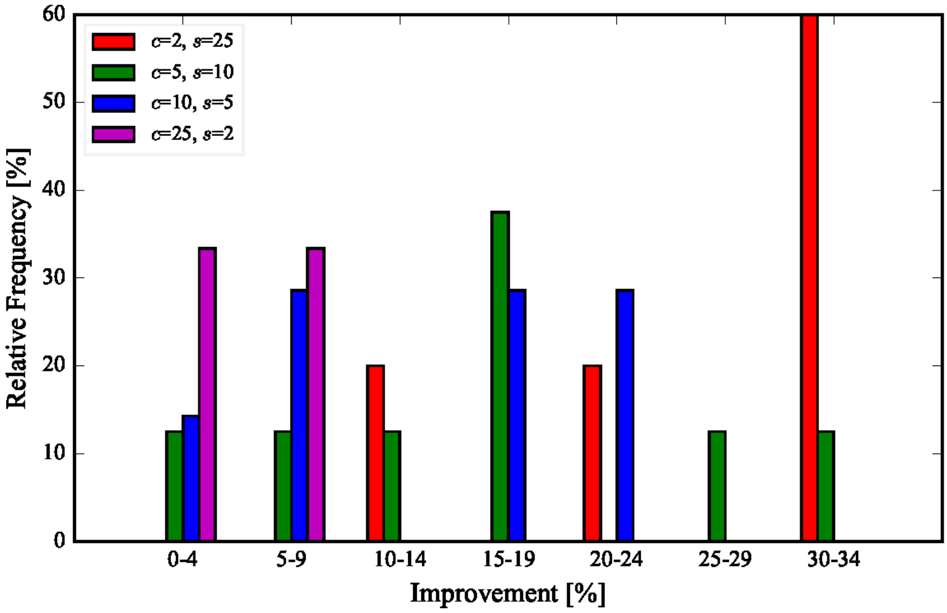


Figure 6.11: Relative frequency of improvement values. This representation considers only those cases where the hybrid clustering-based approach was superior.

Analysis of Schedule Reliability

The performance of all clustering algorithms is presented in Figure 6.12. Again, performance was measured by how often the hybrid clustering-based approach outperformed the corresponding benchmark rule. It is evident that the clustering algorithms perform very similarly. Again, the LV algorithm performed best for *c=10* and *s=5*. Hence, the results for LV are presented and discussed below. The complete results are depicted in the appendix (Table A.4). Because of the limited interpretation of the IM output in this thesis, the findings of Lancichinetti and Fortunato (2009a) were not confirmed. Those authors reported that IM surpassed all clustering algorithms they used.

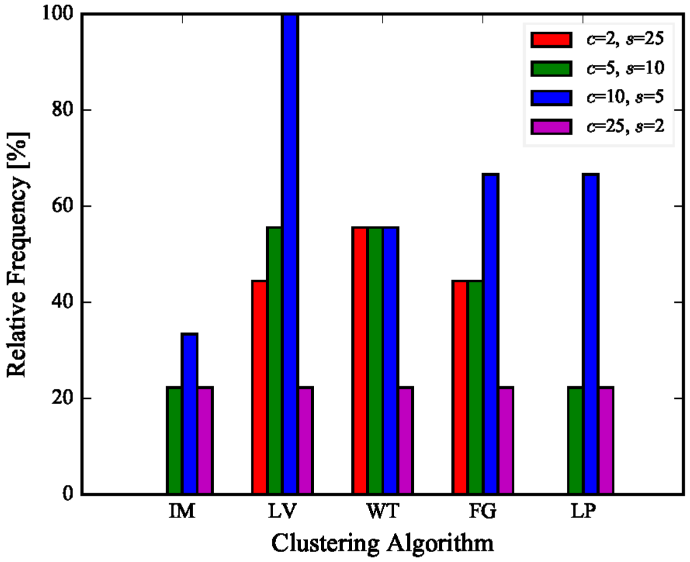


Figure 6.12: Relative frequency of the hybrid clustering-based approach outperforming the benchmark rule, for schedule reliability.

The computational results for schedule reliability are shown in Table 6.3. Again, the best-performing autonomous sequencing rule of the corresponding network model was used as a benchmark for evaluating the effectiveness of the hybrid clustering-based approach. The results (Table 6.3) indicate that the hybrid clustering-based approach outperformed the benchmark rule in many scenarios in terms of schedule reliability. For *c=2* and *s=25*, the results for small *p* were very similar to previous results. This meant the best sequencing rule dominated in two out of 34 cases. The difference was small but significant. In 12 of 34 cases, there was no statistical difference between the best rule and the proposed approach. For *c=10* and *s=5*, the proposed approach was superior in all cases. For *c=25* and *s=2*, it was superior in two of seven cases.

Overall, the superiority of the hybrid clustering-based approach was most notable for *c=10* and *s=5*. In summary, the proposed approach yielded better values than the best rule in 20 out of 34 cases. However, the dependence on *p* was less obvious.

Network Model (*c*, *s*, *p*)	**Best Rule**	**Hybrid Clustering-based Approach**	**Confidence Interval**	**Improvement [%]**
2, 25, 0.1	**97.95**	97.81	[0.0065; 0.2655]	-0.14
2, 25, 0.2	**97.20**	96.43	[0.5359; 1.0161]	-0.80
2, 25, 0.3	**95.02**	**95.18**	[-1.0144; 0.6984] (not significant)	0.17
2, 25, 0.4	**94.35**	**94.50**	[-0.6830; 0.3910] (not significant)	0.16
2, 25, 0.5	91.24	**92.16**	[-1.2485; -0.6035]	1.01
2, 25, 0.6	88.55	**89.32**	[-1.2915; -0.2325]	0.86
2, 25, 0.7	86.33	**87.82**	[-1.8517; -1.1083]	1.72
2, 25, 0.8	82.68	**84.37**	[-2.5423; -0.8377]	2.05
2, 25, 0.9	**81.95**	**82.43**	[-1.2583; 0.2863] (not significant)	0.59

Network Model (*c*, *s*, *p*)	**Best Rule**	**Hybrid Clustering-based Approach**	**Confidence Interval**	**Improvement [%]**
5, 10, 0.1	**92.60**	**92.60**	[-0.1641; 0.1881] (not significant)	0.00
5, 10, 0.2	**91.72**	**91.90**	[-0.6838; 0.3078] (not significant)	0.20
5, 10, 0.3	**90.87**	**90.76**	[-0.1054; 0.3374] (not significant)	-0.12
5, 10, 0.4	80.66	**81.77**	[1.3798; 0.8482]	1.38
5, 10, 0.5	74.89	**77.20**	[-3.0158; -1.6722] (not significant)	3.08
5, 10, 0.6	74.16	**76.88**	[-3.3595; -2.0765]	3.67
5, 10, 0.7	73.82	**74.99**	[-1.5442; -0.7838]	1.58
5, 10, 0.8	72.07	**74.88**	[-3.0609; -2.5551]	3.90
5, 10, 0.9	**71.37**	**71.15**	[-0.3208; 0.7488] (not significant)	-0.30
10, 5, 0.1	89.83	**90.59**	[-1.4994; -0.0166]	0.85
10, 5, 0.2	87.78	**88.22**	[-0.6082; -0.2518]	0.50
10, 5, 0.3	75.65	**80.93**	[-5.9460; -4.6020]	6.98
10, 5, 0.4	78.73	**80.71**	[-2.6385; -1.3175]	2.52
10, 5, 0.5	72.38	**78.51**	[-6.8529; -5.4151]	8.48
10, 5, 0.6	71.14	**75.40**	[-5.0031; -3.5929]	5.99
10, 5, 0.7	72.02	**72.78**	[-1.5111; -0.0049]	1.05
10, 5, 0.8	73.32	**75.50**	[-2.8953; -1.4647]	2.97
10, 5, 0.9	73.09	**73.95**	[-1.5518; -0.1682]	1.18
25, 2, 0.1	78.23	**81.86**	[-3.9352; -3.3208]	4.64
25, 2, 0.2	76.94	**80.52**	[-4.2099; -2.9461]	4.65
25, 2, 0.3	**82.18**	**82.26**	[-1.7332; 1.5772] (not significant)	0.10
25, 2, 0.4	**86.99**	**87.11**	[-0.9591; 0.7111] (not significant)	0.14
25, 2, 0.5	**83.20**	**83.03**	[-0.7202; 1.0602] (not significant)	-0.21
25, 2, 0.6	**82.73**	**83.33**	[-1.3800; 0.1800] (not significant)	0.72
25, 2, 0.7	**83.01**	**83.28**	[-1.2724; 0.7324] (not significant)	0.33

Table 6.3: Computational results for schedule reliability, with a tolerance window of 15 %, using the LV algorithm. Values in bold indicate the approach that achieved the best performance for mean tardiness. In cases where there was no statistical difference between two approaches, both values are in bold. The non-overlapping confidence interval indicate that an approach is superior in the considered scenario. The improvement over the next-best autonomous sequencing rule is reported in the last column.

As evident in Figure 6.13, the potential for improvement was highest for *c=10* and *s=5*, where the largest improvement was 8.48 % and the least 0.5 %. For *c=2* and *s=25*, the improvement was minimal. The smaller and more numerous the clusters become, the more the distribution shifts to the right. In summary, with many small clusters (*c*>*s*), the improvement increases. These findings are exactly the opposite of the previous findings.

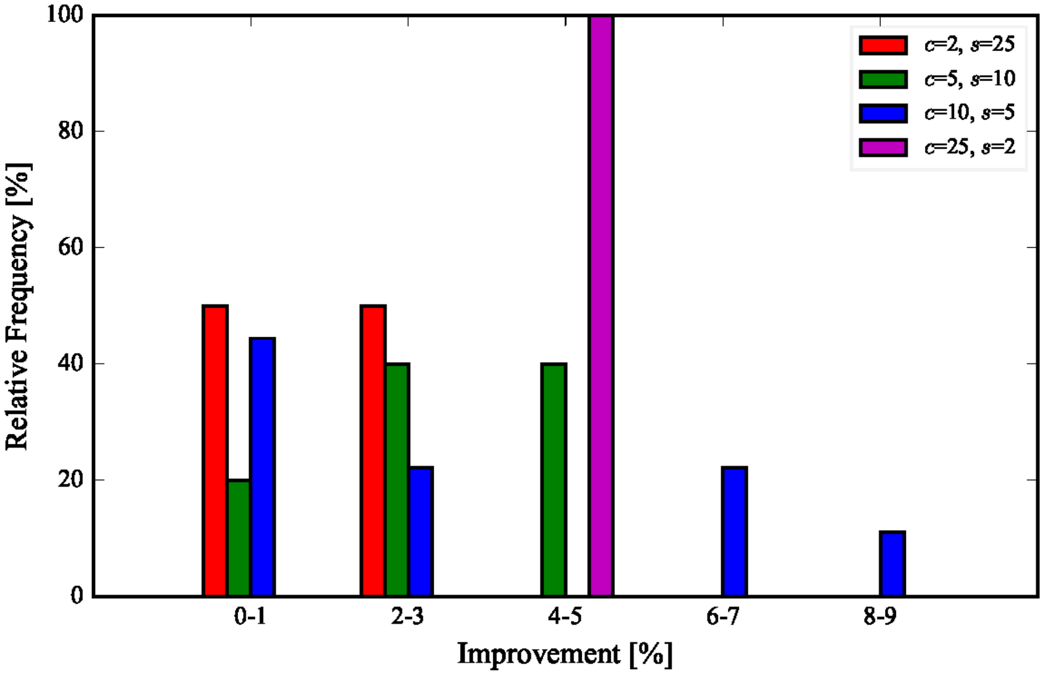


Figure 6.13: Relative frequency of improvement values. This representation only considers cases where the hybrid clustering-based approach is superior.

To conclude, consideration of the cluster topology of a material flow network through using the hybrid clustering-based approach is a promising approach. The results were better than those obtained using a single sequencing rule for the entire system. For the mean tardiness of tardy jobs, the proposed approach significantly outperformed others in 64.7 % of the cases and for schedule reliability in 58.8 % of cases. The improvement in schedule reliability was less noteworthy than for mean tardiness. For schedule reliability, the greatest improvement of 8.48 % over the best-performing rule was 22.82 % less than the improvement regarding mean tardiness. Overall, for mean tardiness, the hybrid clustering-based approach offers strong potential for improved results.

The simulation results show that considering the cluster topology of a material flow network can influence the achievement of logistic objectives, when the job sequencing is decentralized among identified material flow clusters. The effectiveness of the hybrid clustering-based approach depends mainly on the performance measure used. The substantial gains in performance must be considered carefully and should be ensured by adding further sequencing rules.

These findings broadly support the work of Vrabič et al. (2012) in this area, linking autonomous control with the topological characteristics of a material flow system. The results of

Becker et al., 2015) are also supported, where they considered topology in the context of autonomous control as a central element. However, the current findings do not enable direct conclusions to be drawn about the given cluster topology and the resulting performance. Rather, they lead to more general observations. For example, the potential improvement compared to a single sequencing rule for mean tardiness for a few large clusters (*c*<*s*) and for schedule reliability is higher for many small clusters (*c*>*s*).

For practitioners, these findings constitute a first indication of cases in which autonomous clusters would be advantageous. The influence of cluster blurriness *p* on the two observed performance measures was not clearly demonstrated, and further research is required for generalization. The findings give an initial indication of what parameters may benefit from further and deeper analysis. Overall, the findings are helpful for initial estimates of whether the proposed hybrid clustering-based approach has the potential to improve a system's performance. Furthermore, since there is no dominant sequencing rule, the hybrid clustering-based approach appears promising.

6.4 Summary

First, it has been demonstrated that there is a clear relationship between the cluster topology of a material flow network and the resulting performance. For this purpose, two performance measures were used: average shop utilization and average WIP to achieve a targeted utilization level. The findings show that material flow networks with a few large clusters (e.g. *c=2* and *s=25*) imply a high and consistent shop utilization of 86 %. This result remains generally consistent for higher *p*-values, which represent increasing blurriness of the cluster topology. By contrast, many small clusters (e.g. *c=25* and *s=2*) indicate that only a few work systems show a strong level of utilization. Thus, the average shop utilization is comparatively low at 36 %. This is reinforced with increasing blurriness between clusters. After considering the average WIP to achieve target utilization levels, the previous findings were confirmed. In summary, for *c=2* and *s=25*, it is less mean WIP required to achieve target utilization levels. However, when clusters are smaller and more plentiful (*c*>*s*), larger mean WIP is needed. Moreover, increasing blurriness of the cluster topology requires higher levels of WIP. These findings are consistent with those of earlier studies, where networks with well-defined cluster topology (low *p*-values) were identified as superior to networks with blurry clusters (high *p*-values).

Second, the effectiveness of the selected autonomous sequencing rules to reduce the mean tardiness of tardy jobs and enhance the schedule reliability was tested. The results showed a tendency for cluster topology to influence the effectiveness of individual sequencing rules and thus also the logistics performance. Similar to previous results, high *p*-values – indicating smaller and more plentiful clusters (*c*>*s*) – were associated with high mean tardiness and low schedule reliability. However, for *c=25* and *s=2*, an opposing trend was noted; i.e., with increasing *p*-value, the performance improved. There was no sequencing rule that was superior in all cases, regardless of the performance measure used. Thus, no specific recommendation can be made. Moreover, the results of the best sequencing rule of each network model provide the benchmark.

Third, the effectiveness of the hybrid clustering-based approach was evaluated using the clustering results from five algorithms (proposed in Chapter 5). The findings were evaluated from a production logistic perspective. The previous findings indicated that weights and directions should be considered. Therefore, the weighted versions of the clustering algorithms were used, and in the case of IM the edge directions were also considered. The performance was measured by how often the hybrid clustering-based approach outperformed the corresponding benchmark rule. There were no major performance differences among the clustering algorithms. Including the edge directions, as in the case of IM, did not lead to significant performance improvements. This confirms the previous results reported in Chapter 5. Since the results of the LV algorithm were marginally better, the output of that algorithm was used for autonomous clusters in further investigations.

Finally, the application of the hybrid clustering-based approach appears reasonable for almost all cases when the performance is evaluated in terms of mean tardiness or schedule reliability. However, for mean tardiness, the hybrid clustering-based approach offered the greatest potential for improvement. For example, the proposed approach achieved better results than the best rule in 64.7 % of cases. Reductions of up to 30 % were achieved for mean tardiness. Hence, cluster topology is a crucial factor to consider when introducing autonomous control. Although no general conclusions can be drawn concerning the relationship between cluster topology and logistic performance, in material flow networks, the potential for improvement compared to a single sequencing rule is higher for a few large clusters ($c<s$) for mean tardiness and for many small clusters ($c>s$) for schedule reliability.

7 Conclusion and Outlook

To answer the RQs introduced in Chapter 1, this chapter first summarizes the main findings of this thesis. The chapter also highlights the contributions of this research to theory and practice, discusses the limitations and offers an outlook for future research directions.

7.1 Summary of Results and Contributions

In the introductory part of this thesis, the complex and often unpredictable environment of today's manufacturing was described. One way to deal with the corresponding challenges is the transformation toward autonomous manufacturing systems. Apart from technology required to enable data acquisition and real-time control of manufacturing processes, along with algorithms used to set the logic behind decisions in autonomously acting systems, the topology of a given material flow network could prove to be an important factor. The topology of a material flow network can, for example, serve as a basis to identify work systems that are characterized by intensive interactions and merge those systems into autonomous clusters with their decision-making competencies. The main objective of this thesis was to demonstrate whether – or to what extent – the cluster topology of a material flow network affects the logistic performance achievement. Specifically, the study focused on situations where the job sequencing is decentralized within previously identified material flow clusters. The RQs and the relevant findings were as follows:

RQ 1: *How can autonomous job sequencing be implemented in previously identified material flow clusters?* The first step was to develop an approach to provide autonomous job sequencing within previously identified material flow clusters. In the following, this is referred to as the hybrid clustering-based approach. Besides the obvious possibility of using identical sequencing rules for all clusters, it is promising to enable autonomous clusters to individually select a rule from a set of sequencing rules. The combination of several sequencing rules seems promising to increase the overall performance. In addition, it is assumed that a sequencing rule is chosen according to the material flow pattern within an autonomous cluster. This permits it to react more easily to the conditions within a cluster because the autonomous sequencing rule is individually adapted to the cluster. Overall, the cluster topology is a central factor in the implementation process of decentralized control.

RQ 2: *How can synthetic material flow networks be created with a predefined cluster topology?* A methodological approach to generate synthetic material flow networks with a predefined cluster topology was proposed in Chapter 4. The challenges were to determine what parameters define the cluster topology and what concept can be used to imitate the material flow that results from job routing and processing. The following parameters were derived from the definition of a cluster in Chapter 3: (i) the number of clusters *c,* (ii) the cluster size *s* and (iii) the degree of interconnection between individual clusters *p*. Random walks were identified as one possibility to simulate such dynamic job routing processes. Initial evaluation results showed that by combining different values of the three parameters and simultaneously applying the random walk process, researchers can achieve a range of synthetic material flow networks with varying cluster topology. Examples are networks with a few large clusters or many small clusters. In summary, 36 material flow network models – which differed primarily in their cluster topology – were generated and are available for further study. Generally, with a few modifications, this methodological approach can be applied to create networks with arbitrary features.

RQ 3: *Which global network information should be used to identify autonomous clusters from a network theory perspective?* From a network theory perspective, the direction and intensity of material flow are represented by edge direction and edge weight, respectively. Existing clustering algorithms allow for the use of such information to some degree. To answer this RQ, the study employed five well-known clustering algorithms. The IM algorithm was applied to the 36 synthetic material flow networks, and the results showed that the edge directions had little or no impact on the resulting output when the edge weights were considered. Regarding edge weights, for some clustering algorithms – such as IM, WT and FG – the elimination of edge weights led to lower cluster quality. For the remaining algorithms, the overall quality of the resulting clusters improved slightly or remained the same when edge weights were considered. Overall, it was concluded that edge weights should be included when available. However, the edge direction may be ignored from a network theory perspective. There was no clear evidence for whether one algorithm was the best regarding cluster quality.

RQ 4: *How does the cluster topology of material flow networks relate to the logistics performance?* From the simulation results, it was concluded that a clear relationship exists between the cluster topology of given material flow networks and the resulting performance. Material flow networks with a few large clusters displayed the best and most consistent shop performance. This performance was only slightly affected by increasing blurriness of the cluster topology. By contrast, many small clusters indicated that the performance was negatively affected, which was reinforced by increased blurriness between clusters. These results support the importance of considering the cluster topology when implementing autonomous control.

RQ 5: *Do systems having autonomous clusters that are formed according to the intensity of material flow improve the logistical performance?*

Sub-RQ 5.1: *For material flow systems that possess varying cluster topology, how effective are selected sequencing rules to reduce the mean tardiness of tardy jobs and enhance the schedule reliability?* The performance of the 12 sequencing rules was tested to evaluate their potentials and weaknesses. It was assumed that one rule was applied to the entire system at once. The sequencing rule that outperformed all others for a given scenario then served as benchmark for the remaining simulation studies. The results showed that, depending on the given scenario and the observed performance measure (tardiness or schedule reliability), different rules led to the best results. No sequencing rule outperformed all other rules. However, only five or six (out of 12) sequencing rules, depending on the outcome measure, were ranked as the best rule. Overall, no specific recommendation can be made.

Sub-RQ 5.2: *Which global network information should be used to identify autonomous clusters, from a logistics perspective?* To evaluate the findings from a logistics-based perspective, this study examined the effectiveness of the hybrid clustering-based approach using the results of the five clustering algorithms proposed in Chapter 5. For this purpose and in addition to previous findings, the weighted versions of the clustering algorithms were used, and in the case of the IM algorithm, the edge directions were also considered. The aim was to determine the number of cases in which the hybrid clustering-based approach was better than the corresponding benchmark sequencing rule. It was concluded that there were no major performance differences between the different clustering algorithms. This confirms the findings from Chapter 5.

Sub-RQ 5.3: *What is the efficiency of sequencing rules applied in autonomous clusters to reduce the mean tardiness of tardy jobs and enhance the schedule reliability in material flow systems with varying cluster topology?* The potential improvement of the logistic objective achievement through the hybrid clustering-based approach was evaluated by simulation. The performance of the proposed approach was compared with the performance of the respective benchmark sequencing rule. It can be summarized that the application of the hybrid clustering-based approach is reasonable in almost all cases when evaluating the performance using mean tardiness of tardy jobs or schedule reliability. However, for mean tardiness, the hybrid clustering-based approach offers stronger potential for improvement. For example, the proposed approach achieves in 64.7 % better results than the best rule. Moreover, for the schedule reliability in 58.8 % of cases. Moreover, the findings imply that the potential for improvement, compared to a single sequencing rule, was highest for a few large clusters (c<s) for mean tardiness and was highest for many small clusters (c>s) for schedule reliability. These generalizations concerning the relationship between cluster topology and logistic performance are limited, and the topic should be further researched.

To conclude, the insight gained by answering the RQs is that the cluster topology has a significant impact on a system’s performance. Therefore, cluster topology is a crucial factor when introducing autonomous control. The findings of this thesis contribute to the theory by complementing previous work by Vrabič et al. (2012). This study has shown how methods, perspectives and resulting knowledge from network theory can be transferred to manufacturing. For instance, the idea of autonomy can be transferred to material flow clusters, and the performance of additional clustering algorithms has been evaluated. Moreover, the proposed hybrid clustering approach offers a high degree of flexibility regarding the sequencing rules used and the given cluster topology. Hence, transferability to similar problems is ensured.

These findings also contribute in practice toward a smooth transformation to autonomous manufacturing systems with a consideration of the topology of given material flow networks. Knowledge of a cluster topology of a complex material flow system could provide the production planner with a first indication (at an early stage) to decide whether the transition to decentralized control is worthwhile. If so, the expected degree of effort (e.g. computational effort and cost of technological devices) can be estimated. For the concrete practical implementation of the hybrid clustering-based approach, the availability of the data and tools for analyzing and processing these data must be ensured. For example, data obtained from the MES enable a network representation of a material flow system. For analytical purposes, *NetworkX*, a Python package (Hagberg, Swart, & Chult, 2008) and *igraphdata*, a R package (Csardi & Nepusz, 2006) offer sets of tools, and various clustering algorithms and quality measures are pre-implemented. In principle, the job sequencing in autonomous clusters is relatively easy to implement, assuming that many well-known sequencing rules are already realized within MES. Furthermore, technologies such as radio frequency identification (RFID) and sensors that provide job-specific information (e.g. processing times, due dates) can be saved locally on a product. This means control decisions can be decentralized, according to the underlying logic of a sequencing rule. For the preliminary simulation studies that evaluated the performance of a set of sequencing rules within autonomous clusters, there was no need to access existing systems. Additionally, companies have recently gained experience in simulating their manufacturing systems using software solutions like *Plant Simulation*[3] or *AnyLogic*.[4] However, the interfaces for data exchanges between all software solutions are crucial for the successful practical implementation and need to be addressed in the future.

7.2 Outlook and Future Research

Based on the results reported in this thesis, questions arise for future research. They can be divided into three distinct activities, described next.

Generalizability

In this thesis, the variety and number of scenarios considered was low. This kept the research at a manageable size, but it also meant that the generalizability of the findings is limited. Further research is necessary to test the findings and determine the effects of cluster topology in additional scenarios. Concerning the modeling of material flow networks, further parameter combinations with other sets of values – or even a different underlying network – are required. Additions could also be made regarding the hybrid clustering-based approach.

In this thesis, a small set of autonomous sequencing rules was chosen to keep the computational complexity low. The approach could be extended by integrating more or other sequencing rules. To further explore the effectiveness, researchers could also ascertain how the proposed approach performs on systems having additional uncertainties, such as machine breakdowns or rush jobs – which are associated with a complex and dynamic environment.

Computational Effort

The findings of this thesis demonstrate, inter alia, the potential of the hybrid clustering-based approach. However, the challenge in this approach is that the computational effort increases vastly with an increasing number of clusters and/or sequencing rules. The reason is that simulation must be performed for all the available sequencing rules to enable a proper decision; therefore, all possible combinations of sequencing rules must be computed. Regarding its practicability, future research could investigate a relationship between the cluster features (in the form of network measures to capture the cluster topology) and the corresponding sequencing rule. The aim would be to identify the most appropriate sequencing rule for a given cluster to avoid simulations where possible. Moreover, it might be possible to go a step further and study which individual clusters need to be addressed concretely. In concrete terms, further studies are necessary to explore whether there is a positive correlation between specific cluster features and the logistics performance. These insights could support the stepwise introduction of autonomous control in manufacturing systems.

[3] https://plant-simulation.de/, Retrieved October 14, 2022.

[4] https://www.anylogic.com/, Retrieved October 14, 2022.

Dynamics in Networks

This thesis has assumed that clusters do not change over time. This means the clustering algorithms were applied to the aggregated network representation over a certain period. Indeed, the material flow may increase or decrease between any two work systems, or work systems may remain idle for a long period (Wagner & Becker, 2016). Multiple changes would inevitably affect the resulting network clusters. This could have a considerable impact on the management of the shop floor, because the current clusters may no longer correspond to the autonomous clusters, as a result of the changing material flow network. Therefore, robust clusters are required that do not change significantly through certain events occurring.

A possible approach to this problem is to consider not only the aggregated form of the network representation but also several subsequent time steps. The drawback of this approach is that the resulting clusters can differ strongly across the individual time steps (Fortunato, 2010). It might therefore be appropriate to enable smoothing of the resulting clusters, as clusters do not usually change abruptly but develop gradually (Backstrom, Huttenlocher, Kleinberg, & Lan, 2006). Certain clustering algorithms (e.g. Bansal, Bhowmick, & Paymal, 2011; You, Hu, Kamigaito, Funakoshi, & Okumura, 2021) that compensate for random variations across the time steps could be used to find robust clusters based on historical data. Procedures such as that of Greene, Doyle, and Cunningham (2010) enable tracking the clusters over time and identifying superordinate clusters based on the observations. Further research on cluster changes over time is recommended.

References

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*(1), 47–97. doi: 10.1103/RevModPhys.74.47

Aldecoa, R., & Marín, I. (2013). Exploring the limits of community detection strategies in complex networks. *Scientific reports, 3*, Article 2216. doi: 10.1038/srep02216

Alkan, B., Vera, D. A., Ahmad, M., Ahmad, B., & Harrison, R. (2018). Complexity in manufacturing systems and its measures: A literature review. *European Journal of Industrial Engineering, 12*(1), 116–150. doi: 10.1504/EJIE.2018.089883

Arenas, A., Duch, J., Fernández, A., & Gómez, S. (2007). Size reduction of complex networks preserving modularity. *New Journal of Physics, 9*(6), Article 176. doi: 10.1088/1367-2630/9/6/176

Arratia, A., & Mirambell, M. R. (2021). Clustering assessment in weighted networks. *PeerJ. Computer science, 7*, Article e600. doi: 10.7717/peerj-cs.600

Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006). Group formation in large social networks. In L. Ungar, M. Craven, D. Gunopulos, & T. Eliassi-Rad (Eds.), *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '06)* (pp. 44–54). New York: ACM.

Baker, K. R. (1984). Sequencing rules and due-date assignments in a job shop. *Management Science, 30*(9), 1093–1104. doi: 10.1287/mnsc.30.9.1093

Bansal, S., Bhowmick, S., & Paymal, P. (2011). Fast community detection for dynamic complex networks. In L. F. Costa, A. Evuskoff, G. Mangioni, & R. Menezes (Eds.), *Complex Networks: Second International Workshop, CompleNet 2010, Rio de Janeiro, Brazil, October 13-15, 2010, Revised Selected Papers* (pp. 196–207). Berlin: Springer.

Barabási, A.-L., Albert, R., & Jeong, H. (1999). Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications, 272*(1–2), 173–187. doi: 10.1016/S0378-4371(99)00291-5

Barabási, A.-L., Albert, R., & Jeong, H. (2000). Scale-free characteristics of random networks: The topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications, 281*(1–4), 69–77. doi: 10.1016/S0378-4371(00)00018-2

Barrat, A., Barthélemy, M., Pastor-Satorras, R., & Vespignani, A. (2004). The architecture of complex weighted networks. *PNAS, 101*(11), 3747–3752. doi: 10.1073/pnas.0400087101

Barrat, A., Barthélemy, M., & Vespignani, A. (2008). *Dynamical processes on complex networks.* Cambridge: Cambridge University Press.

Becker, T., & Wagner, D. (2015). Stufenweise Einführung von dezentraler Steuerung. *Industrie 4.0 Management, 31*(2), 28–32.

Becker, T., Beber, M. E., Windt, K., & Hütt, M.-T. (2012). The impact of network connectivity on performance in production logistic networks. *CIRP Journal of Manufacturing Science and Technology, 5*(4), 309–318. doi: 10.1016/j.cirpj.2012.09.004

Becker, T., Meyer, M., & Windt, K. (2014). A manufacturing systems network model for the evaluation of complex manufacturing systems. *International Journal of Productivity and Performance Management, 63*(3), 324–340. doi: 10.1108/IJPPM-03-2013-0047

Becker, T., Weimer, D., & Pannek, J. (2015). Network structures and decentralized control in logistics: Topology, interfaces, and dynamics. *International Journal of Advanced Logistics, 4*(1), 1–8. doi: 10.1080/2287108X.2015.1012329

Bellgran, M., & Säfsten, K. (2010). *Production development: Design and operation of production systems.* London: Springer.

Blackstone, J. H., Phillips, D. T., & Hogg, G. L. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research, 20*(1), 27–45. doi: 10.1080/00207548208947745

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10), Article P10008. doi: 10.1088/1742-5468/2008/10/P10008

Blunck, H., Armbruster, D., Bendul, J., & Hütt, M.-T. (2018). The balance of autonomous and centralized control in scheduling problems. *Applied Network Science, 3*(1), Article 16. doi: 10.6084/m9.figshare.c.4159037.v1

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., & Hwang, D. (2006). Complex networks: Structure and dynamics. *Physics Reports, 424*(4–5), 175–308. doi: 10.1016/j.physrep.2005.10.009

Bollobás, B., Riordan, O., Spencer, J., & Tusnády, G. (2006). The degree sequence of a scale-free random graph process. In M. Newman, A.-L. Barabási, & D. J. Watts (Eds.), *The structure and dynamics of networks* (pp. 384–395). Princeton: Princeton University Press.

Böse, F., & Windt, K. (2007). Catalogue of criteria for autonomous control in logistics. In M. Hülsmann & K. Windt (Eds.), *Understanding autonomous cooperation and control in logistics* (pp. 57–72). Berlin: Springer.

Bouhai, N., & Saleh, I. (2017). *Internet of Things: Evolutions and innovations.* Hoboken: Wiley & Sons.

Brandes, U., Gaertler, M., & Wagner, D. (2003). Experiments on graph clustering algorithms. In G. Battista & U. Zwick (Eds.), *Algorithms – ESA 2003: 11th Annual European Symposium, Budapest, Hungary, September 16–19, 2003* (pp. 568–579). Berlin: Springer.

Brandes, U., Gaertler, M., & Wagner, D. (2007). Engineering graph clustering: Models and experimental evaluation. *ACM Journal of Experimental Algorithmics, 12*(1), Article 1.1. doi: 10.1145/1227161.1227162

Burbidge, J. L. (1975). *The introduction of group technology*. New York: John Wiley & Sons.

Chin, J. H., & Ratnavelu, K. (2016). Detecting community structure by using a constrained label propagation algorithm. *PloS one, 11*(5), Article e0155320. doi: 10.1371/journal.pone.0155320

Cicirello, V. A., & Smith, S. F. (2001). Ant colony control for autonomous decentralized shop floor routing. In *ISADS '01: Proceedings of the Fifth International Symposium on Autonomous Decentralized Systems* (pp. 383–390). Washington, DC: IEEE Computer Society.

Clauset, A., Newman, M. E. J., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E, 70*(6), Article 66111. doi: 10.1103/PhysRevE.70.066111

Clauset, A., Shalizi, C. R., & Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review, 51*(4), 661–703. doi: 10.1137/070710111

Costa, L. da F., Rodrigues, F. A., Travieso, G., & Villas Boas, P. R. (2007). Characterization of complex networks: A survey of measurements. *Advances in Physics, 56*(1), 167–242. doi: 10.1080/00018730601170527

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems, 1695*(5), 1–9.

Cupertino, T. H., Carneiro, M. G., Zheng, Q., Zhang, J., & Zhao, L. (2018). A scheme for high level data classification using random walk and network measures. *Expert Systems with Applications, 92*(7), 289–303. doi: 10.1016/j.eswa.2017.09.014

Dangelmaier, W. (2009). *Theorie der Produktionsplanung und -steuerung: Im Sommer keine Kirschpralinen?* Berlin: Springer.

Danon, L., Díaz-Guilera, A., Duch, J., & Arenas, A. (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment, 2005*(09), Article P09008. doi: 10.1088/1742-5468/2005/09/P09008

Dao, V.-L., Bothorel, C., & Lenca, P. (2020). Community structure: A comparative *evaluation* of community detection methods. *Network Science, 8*(1), 1–41. doi: 10.1017/nws.2019.59

Deshmukh, A. V., Talavage, J. J., & Barash, M. M. (1998). Complexity in manufacturing systems, Part 1: Analysis of static complexity. *IIE Transactions, 30*(7), 645–655. doi: 10.1080/07408179808966508

Diestel, R. (2017). *Graph theory* (5th ed.). Berlin: Springer.

Domschke, W., Scholl, A., & Voß, S. (1997). *Produktionsplanung: Ablauforganisatorische Aspekte* (2nd ed.). Berlin: Springer.

Dyckhoff, H., & Spengler, T. S. (2010). *Produktionswirtschaft: Eine Einführung* (3rd ed.). Berlin]: Springer.

Efthymiou, K., Mourtzis, D., Pagoropoulos, A., Papakostas, N., & Chryssolouris, G. (2016). Manufacturing systems complexity analysis methods review. *International Journal of Computer Integrated Manufacturing, 29*(9), 1025–1044. doi: 10.1080/0951192X.2015.1130245

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing* (2nd ed.). Berlin: Springer.

ElMaraghy, H. A., Kuzgunkaya, O., & Urbanic, R. J. (2005). Manufacturing systems configuration complexity. *CIRP Annals, 54*(1), 445–450. doi: 10.1016/S0007-8506(07)60141-3

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals, 61*(2), 793–814. doi: 10.1016/j.cirp.2012.05.001

Emmons, S., Kobourov, S., Gallant, M., & Börner, K. (2016). Analysis of network clustering algorithms and cluster quality metrics at scale. *PloS one, 11*(7), Article e0159161. doi: 10.1371/journal.pone.0159161

Fan, Y., Li, M., Zhang, P., Wu, J., & Di, Z. (2007). The effect of weight on community structure of networks. *Physica A: Statistical Mechanics and its Applications, 378*(2), 583–590. doi: 10.1016/j.physa.2006.12.021

Fortunato, S. (2010). Community detection in graphs. *Physics Reports, 486*(3–5), 75–174. doi: 10.1016/j.physrep.2009.11.002

Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports, 659*(12), 1–44. doi: 10.1016/j.physrep.2016.09.002

Fottner, J., Clauer, D., Hormes, F., Freitag, M., Beinke, T., Overmeyer, L., … Thomas, F. (2021). Autonomous systems in intralogistics: State of the art and future research challenges. *Logistics Research, 14*(2), 1–41. doi: 10.23773/2021_2

Fränti, P., & Sieranoja, S. (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition, 93*(2), 95–112. doi: 10.1016/j.patcog.2019.04.014

Freitag, M., Herzog, O., & Scholz-Reiter, B. (2004). Selbststeuerung logistischer Prozesse: Ein Paradigmenwechsel und seine Grenzen. *Industrie Management, 20*(1), 23–27. Retrieved from: http://www.sfb637.uni-bremen.de/pubdb/repository/SFB637-A5-04-002-IJ.pdf

Freitag, M., & Hildebrandt, T. (2016). Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. *CIRP Annals, 65*(1), 433–436. doi: 10.1016/j.cirp.2016.04.066

Frizelle, G., & Woodcock, E. (1995). Measuring complexity as an aid to developing operational strategy. *International Journal of Operations & Production Management, 15*(5), 26–39. doi: 10.1108/01443579510083640

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* San Francisco: Freeman.

Girvan, M., & Newman, M. E. J. (2002). Community structure in social and biological networks. *PNAS, 99*(12), 7821–7826. doi: 10.1073/pnas.12265379

Göbel, F., & Jagers, A. A. (1974). Random walks on graphs. *Stochastic Processes and their Applications, 2*(4), 311–336. doi: 10.1016/0304-4149(74)90001-5

Greene, D., Doyle, D., & Cunningham, P. (2010). Tracking the evolution of communities in dynamic social networks. In N. Memon & R. Alhajj (Eds.), *2010 International Conference on Advances in Social Networks Analysis and Mining* (pp. 176–183). New York: IEEE Press.

Grundstein, S., Schukraft, S., Freitag, M., & Scholz-Reiter, B. (2015). Planorientierte autonome Fertigungssteuerung. *wt Werkstattstechnik online, 105*(4), 220–224. Retrieved from https://www.researchgate.net/profile/Michael-Freitag/publication/319181935_Schedule-oriented_autonomous_production_control/links/59ca29f30f7e9bbfdc367482/Schedule-oriented-autonomous-production-control.pdf

Hackstein, R. (1989). *Produktionsplanung und -steuerung (PPS): Ein Handbuch für die Betriebspraxis* (2nd ed.). Düsseldorf: VDI.

Hagberg, A., Swart, P., & Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference (SciPy2008)* (pp. 11–16). Retrieved from: http://conference.scipy.org/proceedings/SciPy2008/paper_0

Hamedi, M., Esmaeilian, G. R., Ismail, N., & Ariffin, M. K. (2012). A survey on formation of virtual cellular manufacturing systems (VCMSs) and related issues. *Scientific Research and Essays, 7*(40), 3316–3328. doi: 10.5897/SRE11.217

Haupt, R. (1989). A survey of priority rule-based scheduling. *Operations-Research-Spektrum, 11*(1), 3–16. doi: 10.1007/BF01721162

Holthaus, O. (1996). *Ablaufplanung bei Werkstattfertigung: Simulationsgestützte Analyse von Steuerungs- und Koordinationsregeln.* Wiesbaden: Deutscher Universitätsverlag.

Holthaus, O., & Rajendran, C. (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics, 48*(1), 87–105. doi: 10.1016/S0925-5273(96)00068-0

Hoogeveen, J. A., Lenstra, J. K., & Veltman, B. (1996). Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *European Journal of Operational Research, 89*(1), 172–175. doi: 10.1016/S0377-2217(96)90070-3

Hopp, W. J., & Spearman, M. L. (2004). To pull or not to pull: What is the question? *Manufacturing & Service Operations Management, 6*(2), 133–148. doi: 10.1287/MSOM.1030.0028

Hopp, W. J., & Spearman, M. L. (2011). *Factory physics* (3rd ed.). Long Grove: Waveland Press.

Isik, F. (2011). Complexity in supply chains: A new approach to quantitative measurement of the supply-chain-complexity. In P. Li (Ed.), *Supply Chain Management* (pp. 417–432). London: InTechOpen.

Jayamohan, M. S., & Rajendran, C. (2010). A comparative analysis of two different approaches to scheduling in flexible flow shops. *Production Planning & Control, 11*(6), 572–580. doi: 10.1080/095372800414133

Kagermann, H., Helbig, J., Hellinger, A., & Wahlster, W. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 working group.* Berlin: Forschungsunion. Retrieved from: https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf

Kanet, J. J., & Hayya, J. C. (1982). Priority dispatching with operation due dates in a job shop. *Journal of Operations Management, 2*(3), 167–175. doi: 10.1016/0272-6963(82)90004-3

Khan, B. S., & Niazi, M. (2017). Network community detection: A review and visual survey. *arXiv, 1708.00977*. doi: 10.48550/arXiv.1708.00977

Khilwani, N., Ulutas, B. H., Islier, A. A., & Tiwari, M. K. (2011). A methodology to design virtual cellular manufacturing systems. *Journal of Intelligent Manufacturing, 22*(4), 533–544. doi: 10.1007/s10845-009-0314-6

Kim, J.-H., & Duffie, N. A. (2004). Backlog control design for a closed loop PPC system. *CIRP Annals, 53*(1), 357–360. doi: 10.1016/S0007-8506(07)60716-1

Kiran, A. S., Alptekin, S., & Kaplan, A. C. (1991). Tardiness heuristic for scheduling Flexible Manufacturing Systems. *Production Planning & Control, 2*(3), 228–241. doi: 10.1080/09537289108919351

Kiran, A. S., & Smith, M. L. (1984). Simulation studies in job shop sheduling—I a survey. *Computers & Industrial Engineering, 8*(2), 87–93. doi: 10.1016/0360-8352(84)90001-9

Kramer, O. (2017). *Genetic algorithm essentials.* Cham: Springer.

Kurbel, K. (2016). *Enterprise Resource Planning und Supply Chain Management in der Industrie: Von MRP bis Industrie 4.0* (8th ed.). Berlin: De Guyter Oldenbourg.

Lancichinetti, A., & Fortunato, S. (2009a). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical review E: Statistical, nonlinear, and soft matter physics, 80*(1 Pt 2), Article 16118. doi: 10.1103/PhysRevE.78.046110

Lancichinetti, A., & Fortunato, S. (2009b). Community detection algorithms: A comparative analysis. *Physical review E: Statistical, nonlinear, and soft matter physics, 80*(5 Pt 2), Article 56117. doi: 10.1103/PhysRevE.80.056117

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E: Statistical, nonlinear, and soft matter physics, 78*(4 Pt 2), Article 46110. doi: 10.1103/PhysRevE.78.046110

Lancichinetti, A., Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Finding statistically significant communities in networks. *PloS one, 6*(4), Article e18961. doi: 10.1371/journal.pone.0018961

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering, 6*(4), 239–242. doi: 10.1007/s12599-014-0334-4

Law, A. M. (2007). *Simulation modeling and analysis* (4th ed., internat. ed.). Boston: McGraw-Hill.

Liaqait, R. A., Hamid, S., Warsi, S. S., & Khalid, A. (2021). A critical analysis of job shop scheduling in context of Industry 4.0. *Sustainability, 13*(14), Article 7684. doi: 10.3390/su13147684

Liu, S.-Y., Li, C., Feng, Y.-P., & Rong, G. (2013). Clustering structure and logistics: A new framework for supply network analysis. *Chemical Engineering Research and Design, 91*(8), 1383–1389. doi: 10.1016/j.cherd.2013.02.021

Lödding, H. (2013). *Handbook of manufacturing control: Fundamentals, description, configuration.* Berlin: Springer.

Lödding, H. (2016). *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration* (3rd ed.). Berlin: Springer Vieweg.

Lödding, H., & Kuyumcu, A. (2015). Modelling schedule reliability. *International Journal of Production Research, 53*(9), 2871–2884. doi: 10.1080/00207543.2015.1008703

Lovász, L. (1996). Random walks on graphs: A survey. In D. Miklós, V. T. Sós, & T. Szőnyi (Eds.), *Combinatorics, Paul Erdős is eighty: Vol. 2* (pp. 353–397). Budapest: János Bolyai Mathematical Society.

Lu, M.-S., & Liu, Y.-J. (2011). Dynamic dispatching for a flexible manufacturing system based on fuzzy logic. *The International Journal of Advanced Manufacturing Technology, 54*(9-12), 1057–1065. doi: 10.1007/s00170-010-2993-8

Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications, 390*(6), 1150–1170. doi: 10.1016/j.physa.2010.11.027

Magee, C. L., & de Weck, O. L. (2004). Complex system classification. *INCOSE International Symposium, 14*(1), 471–488. doi: 10.1002/j.2334-5837.2004.tb00510.x

Malliaros, F. D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. *Physics Reports, 533*(4), 95–142. doi: 10.1016/j.physrep.2013.08.002

Mehrsai, A., Figueira, G., Santos, N., Amorim, P., & Almada-Lobo, B. (2017). Decentralized vs. centralized sequencing in a complex job-shop scheduling. In H. Lödding, R. Riedel, K.-D. Thoben, G. von Cieminski, & D. Kiritsis (Eds.), *Advances in Production Management System:. The Path to Intelligent, Collaborative and Sustainable Manufacturing* (pp. 467–474). Cham: Springer.

Miyashita, K. (2000). Job-shop scheduling with genetic programming. In D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, & H. G. Beyer (Eds.), *GECCO-2000. Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 505–512). San Francisco: Morgan Kaufmann.

Modrak, V., & Soltysova, Z. (2017). Novel complexity indicator of manufacturing process chains and its relations to indirect complexity indicators. *Complexity, 2017*(7), Article 9102824. doi: 10.1155/2017/9102824

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., … Ueda, K. (2016). Cyber-physical systems in manufacturing. *CIRP Annals, 65*(2), 621–641. doi: 10.1016/j.cirp.2016.06.005

Mourtzis, D., & Doukas, M. (2012). Decentralized manufacturing systems review: Challenges and outlook. *Logistics Research, 5*(3–4), 113–121. doi: 10.1007/s12159-012-0085-x

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review, 45*(2), 167–256. doi: 10.1137/S003614450342480

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E: Statistical, nonlinear, and soft matter physics, 69*(6), Article 66133. doi: 10.1103/PhysRevE.69.066133

Newman, M. E. J. (2006). Modularity and community structure in networks. *PNAS, 103*(23), 8577–8582. doi: 10.1073/pnas.0601602103

Newman, M. E. J., Barabási, A.-L., & Watts, D. J. (2006). *The structure and dynamics of networks.* Princeton: Princeton University Press.

Newman, M. E. J., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E: Statistical, nonlinear, and soft matter physics, 69*(2 Pt 2), Article 26113. doi: 10.1103/PhysRevE.69.026113

Newman, M. J.E. (2005). A measure of betweenness centrality based on random walks. *Social Networks, 27*(1), 39–54. doi: 10.1016/j.socnet.2004.11.009

Newman, M. E. J. (2010). *Networks: An introduction.* Oxford: Oxford University Press.

Nyhuis, P., Mayer, J., & Kuprat, T. (2014). Die Bedeutung von Industrie 4.0 als Enabler für logistische Modelle. In W. Kersten, H. Koller, & H. Lödding (Eds.), *Industrie 4.0: Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern* (pp. 79–100). Berlin: Gito.

Nyhuis, P., & Wiendahl, H.-P. (2009). *Fundamentals of production logistics: Theory, tools and applications.* Berlin: Springer.

Nyhuis, P., & Wiendahl, H.-P. (2012). *Logistische Kennlinien: Grundlagen, Werkzeuge und Anwendungen* (3rd ed.). Berlin: Springer.

Omar, Y. M., Minoufekr, M., & Plapper, P. (2018). Lessons from social network analysis to Industry 4.0. *Manufacturing Letters, 15*(2), 97–100. doi: 10.1016/j.mfglet.2017.12.006

Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks, 32*(3), 245–251. doi: 10.1016/j.socnet.2010.03.006

Orman, G. K., Labatut, V., & Cherifi, H. (2011). On accuracy of community structure discovery algorithms. *Journal of Convergence Information Technology, 6*(11), 283–292. doi: 10.4156/jcit.vol6.issue11.32

Orman, G. K., Labatut, V., & Cherifi, H. (2012). Comparative evaluation of community detection algorithms: A topological approach. *Journal of Statistical Mechanics: Theory and Experiment, 2012*(08), Article P08001. doi: 10.1088/1742-5468/2012/08/P08001

Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature, 435*(7043), 814–818. doi: 10.1038/nature03607

Palla, G., Lovász, L., & Vicsek, T. (2010). Multifractal network generator. *PNAS, 107*(17), 7640–7645. doi: /10.1073/pnas.0912983107

Parente, M., Figueira, G., Amorim, P., & Marques, A. (2020). Production scheduling in the context of Industry 4.0: Review and trends. *International Journal of Production Research, 58*(17), 5401–5431. doi: 10.1080/00207543.2020.1718794

Philipp, T., Böse, F., & Windt, K. (2006). Autonomously controlled processes-characterisation of complex production systems. In P. Cunha & P. Maropoulos (Eds.), *Proceedings of 3rd CIRP Conference in Digital Enterprise Technology, Setubal, Portugal*. Retrieved from: https://www.researchgate.net/publication/251249960_Autonomously_controlled_processes_characterisation_of_complex_production_systems

Pickardt, C. W., Hildebrandt, T., Branke, J., Heger, J., & Scholz-Reiter, B. (2013). Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems. *International Journal of Production Economics, 145*(1), 67–77. doi: 10.1016/j.ijpe.2012.10.016

Pinedo, M. (2014). *Scheduling: Theory, algorithms, and systems* (4th ed.). New York: Springer.

Pons, P., & Latapy, M. (2006). Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications, 10*(2), 191–218. Retrieved from https://jgaa.info/accepted/2006/PonsLatapy2006.10.2.pdf

Rabe, M., Spieckermann, S., & Wenzel, S. (2008). *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken.* Berlin: Springer.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *PNAS, 101*(9), 2658–2663. doi: 10.1073/pnas.0400054101

Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E: Statistical, nonlinear, and soft matter physics, 76*(3 Pt 2), 36106. doi: 10.1103/PhysRevE.76.036106

Raman, N., Talbot, F. B., & Rachamadugu, R. V. (1989). Due date based scheduling in a general flexible manufacturing system. *Journal of Operations Management, 8*(2), 115–132. doi: 10.1016/0272-6963(89)90015-6

Reichardt, J. (2009). *Structure in complex networks.* Berlin: Springer.

Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical review E: Statistical, nonlinear, and soft matter physics, 74*(1), Article 16110. doi: 10.1103/PhysRevE.74.016110

Reid, R. D., & Sanders, N. R. (2005). *Operations Management: An Integrated Approach* (2nd ed.). Hoboken: John Wiley & Sons.

Reményi, C., & Staudacher, S. (2014). Systematic simulation based approach for the identification and implementation of a scheduling rule in the aircraft engine maintenance. *International Journal of Production Economics, 147*(1), 94–107. doi: 10.1016/j.ijpe.2012.10.022

Robinson, S. (2014). *Simulation: The practice of model development and use* (2nd ed.). London: Red Globe Press.

Rosvall, M., Axelsson, D., & Bergstrom, C. T. (2009). The map equation. *The European Physical Journal Special Topics, 178*(1), 13–23. doi: 10.1140/epjst/e2010-01179-1

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *PNAS, 105*(4), 1118–1123. doi: 10.1073/pnas.0706851105

Saxena, L. K., & Jain, P. K. (2011). Dynamic cellular manufacturing systems design: A comprehensive model. *The International Journal of Advanced Manufacturing Technology, 53*(1), 11–34. doi: 10.1007/s00170-010-2842-9

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review, 1*(1), 27–64. doi: 10.1016/j.cosrev.2007.05.001

Schäfer, R., Chankov, S., & Bendul, J. (2016). What is really “on-time”?: A comparison of due date performance indicators in production. *Procedia CIRP, 52*(1), 124–129. doi: 10.1016/j.procir.2016.07.017

Schoettl, F., Paefgen, M.-C., & Lindemann, U. (2014). Approach for measuring change-induced complexity based on the production architecture. *Procedia CIRP, 17*, 172–177.8 doi: 10.1016/j.procir.2014.01.043

Scholz-Reiter, B., Beer, C. de, Freitag, M., & Jagalski, T. (2008). Bio-inspired and pheromone-based shop-floor control. *International Journal of Computer Integrated Manufacturing, 21*(2), 201–205. doi: 10.1080/09511920701607840

Scholz-Reiter, B., Rekersbrink, H., & Görges, M. (2010). Dynamic flexible flow shop problems: Scheduling heuristics vs. autonomous control. *CIRP Annals, 59*(1), 465–468. doi: 10.1016/j.cirp.2010.03.030

Schönsleben, P. (2011). *Integrales Logistikmanagement: Operations und Supply Chain Management innerhalb des Unternehmens und unternehmensübergreifend.* Berlin: Springer.

Schuh, G., & Stich, V. (2012). *Produktionsplanung und -steuerung 1: Grundlagen der PPS* (4th ed.). Berlin: Springer Vieweg.

Schukraft, S., Grundstein, S., Freitag, M., & Scholz-Reiter, B. (2015). Strategies for the coupling of autonomous control and central planning: Evaluation of strategies using logistic objectives achievement and planning adherence. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2015)* (pp. 121–126). New York: IEEE Press.

Schukraft, S., Grundstein, S., Freitag, M., & Scholz-Reiter, B. (2016). Evaluation of strategies for the coupling of central planning and autonomous control in dynamic job shop environments. *Advanced Materials Research, 1140*, 457–464. doi: 10.4028/www.scientific.net/AMR.1140.457

Schukraft, S., Teucke, M., Freitag, M., & Scholz-Reiter, B. (2021). Autonomous control of logistic processes: A retrospective. In M. Freitag, H. Kotzab, & N. Megow (Eds.), *Dynamics in Logistics: Twenty-Five Years of Interdisciplinary Logistics Research in Bremen, Germany* (pp. 3–34). Cham: Springer.

Scott, J. (2017). *Social network analysis* (4th ed.). London: Sage.

Sels, V., Gheysen, N., & Vanhoucke, M. (2012). A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. *International Journal of Production Research, 50*(15), 4255–4270. doi: 10.1080/00207543.2011.611539

Shahsavar, M., Najafi, A. A., & Niaki, S. T. A. (2011). Statistical design of genetic algorithms for combinatorial optimization problems. *Mathematical Problems in Engineering, 17*(3), 1–17. doi: 10.1155/2011/872415

Spath, D. (Ed.) (2013). *Produktionsarbeit der Zukunft: Industrie 4.0.* Stuttgart: Fraunhofer-Verlag.

Spearman, M. L., Woodruff, D. L., & Hopp, W. J. (2022). CONWIP Redux: Reflections on 30 years of development and implementation. *International Journal of Production Research, 60*(1), 381–387. doi: 10.1080/00207543.2021.1954713

Takemoto, K., & Oosawa, C. (2012). Introduction to complex networks: Measures, statistical properties, and models. In M. Dehmer & S. C. Basak (Eds.), *Statistical and Machine Learning approaches for network analysis* (pp. 45–75). New York: John Wiley & Sons.

Tanchoco, J. M. A. (1999). Cellular machine layout based on the Segmented Flow Topology. *International Journal of Production Research, 37*(5), 1041–1062. doi: 10.1080/002075499191391

Thomas, A., Trentesaux, D., & Valckenaers, P. (2012). Intelligent distributed production control. *Journal of Intelligent Manufacturing, 23*(6), 2507–2512. doi: 10.1007/s10845-011-0601-x

Travers, J., & Milgram, S. (1969). An experimental study of the small world problem. *Sociometry, 32*(4), 425–443. doi: 10.2307/2786545

Trentesaux, D. (2009). Distributed control of production systems. *Engineering Applications of Artificial Intelligence, 22*(7), 971–978. doi: 10.1016/j.engappai.2009.05.001

van der Hoef, H., & Warrens, M. J. (2019). Understanding information theoretic measures for comparing clusterings. *Behaviormetrika, 46*(2), 353–370. doi: 10.1007/s41237-018-0075-7

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research, 11*, 2837–2854. Retrieved from: https://jmlr.csail.mit.edu/papers/volume11/vinh10a/vinh10a.pdf

Vinod, V., & Sridharan, R. (2011). Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. *International Journal of Production Economics, 129*(1), 127–146. doi: 10.1016/j.ijpe.2010.08.017

Vrabič, R., Husejnagić, D., & Butala, P. (2012). Discovering autonomous structures within complex networks of work systems. *CIRP Annals, 61*(1), 423–426. doi: 10.1016/j.cirp.2012.03.083

Vrabič, R., Škulj, G., & Butala, P. (2013). Anomaly detection in shop floor material flow: A network theory approach. *CIRP Annals, 62*(1), 487–490. doi: 10.1016/j.cirp.2013.03.131

Wagner, D., & Becker, T. (2016). Dynamische Netzwerkanalyse: Ein Werkzeug zur Untersuchung struktureller Veränderungen in Materialflussnetzwerken. *Industrie 4.0 Management, 32*(6), 34–38.

Wagner, D., & Becker, T. (2017). Evaluation of the applicability of random walks for generation of material flow network models. *Procedia CIRP, 63*(1), 488–492. doi: 10.1016/j.procir.2017.03.164

Wagner, D., & Becker, T. (2018). Automated generation of material flow network models with cluster structures using random walks. *Procedia CIRP, 72*(1), 569–573. doi: 10.1016/j.procir.2018.03.249

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature, 393*(6684), 440–442. doi: 10.1038/30918

Wiendahl, H.-P., & Scholtissek, P. (1994). Management and control of complexity in manufacturing. *CIRP Annals, 43*(2), 533–540. doi: 10.1016/S0007-8506(07)60499-5

Wiendahl, H.-P., Reichardt, J., Nyhuis, P., & Rossi, R. (2015). *Handbook factory planning and design.* Berlin: Springer.

Windt, K., Becker, T., & Kolev, I. (2010a). A comparison of the logistics performance of autonomous control methods in production logistics. In P. Kuhlang & W. Sihn (Eds.), *Proceedings of the 43rd CIRP International Conference on Manufacturing Systems* (pp. 576–583). Vienna: NWV.

Windt, K., Philipp, T., Böse, F., & Becker, T. (2010b). Application of a three-component evaluation system for autonomous control in logistics. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 224*(8), 1267–1276. doi: 10.1243/09544054JEM16

Windt, K., Becker, T., Jeken, O., & Gelessus, A. (2010c). A classification pattern for autonomous control methods in logistics. *Logistics Research, 2*(2), 109–120. doi: 10.1007/s12159-010-0030-9

Windt, K., & Hülsmann, M. (2007). Changing paradigms in logistics: Understanding the shift from conventional control to autonomous cooperation and control. In M. Hülsmann & K. Windt (Eds.), *Understanding autonomous cooperation and control in logistics. The impact of autonomy on management, information, communication and material flow* (pp. 1–16). Berlin: Springer.

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE transactions on neural networks, 16*(3), 645–678. doi: 10.1109/TNN.2005.845141

Yang, J., & Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems, 42*(1), 181–213. doi: 10.1007/s10115-013-0693-z

You, J., Hu, C., Kamigaito, H., Funakoshi, K., & Okumura, M. (2021). Robust dynamic clustering for temporal networks. In G. Demartini & G. Zuccon (Eds.), *CIKM '21. Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 2424–2433). New York: ACM.

Zambrano Rey, G., Bonte, T., Prabhu, V., & Trentesaux, D. (2014). Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simulation Modelling Practice and Theory, 46*, 53–75. doi: 10.1016/j.simpat.2014.01.005

Zhang, L., Priestley, J., DeMaio, J., Ni, S., & Tian, X. (2021). Measuring customer similarity and identifying cross-selling products by community detection. *Big data, 9*(2), 132–143. doi: 10.1089/big.2020.0044

A Appendix

	Directed, unweighted networks				***Undirected, unweighted networks***			
c, s ***p***	***2,25***	***5,10***	***10,5***	***25,2***	***2,25***	***5,10***	***10,5***	***25,2***
0.1	14	14	12	33	25	23	22	22
0.2	14	12	12	32	24	23	21	19
0.3	13	11	10	33	24	23	21	20
0.4	14	12	10	27	24	23	21	17
0.5	13	12	9	26	24	22	20	17
0.6	13	10	9	26	24	22	20	17
0.7	12	10	10	30	24	22	20	18
0.8	13	11	10	15	24	22	20	16
0.9	12	10	8	22	23	21	19	18

Table A.1: Identified number of clusters using the IM algorithm. Each value corresponds to the average over 15 material flow networks.

	(a) p=0.1				(b) p=0.2			
	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2
FG	**0.45**	**0.70**	**0.83**	**0.94**	**0.43**	**0.65**	**0.76**	**0.93**
	(0.45)	(0.70)	(0.83)	(0.93)	(0.42)	(0.65)	(0.76)	(0.91)
LV	**0.45**	**0.70**	**0.83**	**0.94**	**0.43**	**0.65**	**0.76**	**0.93**
	(0.45)	(0.70)	(0.84)	(0.94)	(0.43)	(0.65)	(0.76)	(0.93)
IM	**0.45**	**0.70**	**0.83**	**0.94**	**0.43**	**0.65**	**0.76**	**0.93**
	(0.45)	(0.08)	(0.29)	(0.93)	(0.01)	(0.08)	(0.27)	(0.93)
WT	**0.45**	**0.70**	**0.83**	**0.94**	**0.43**	**0.65**	**0.76**	**0.93**
	(0.45)	(0.70)	(0.83)	(0.93)	(0.43)	(0.65)	(0.76)	(0.93)
LP	**0.45**	**0.70**	**0.83**	**0.94**	**0.43**	**0.65**	**0.76**	**0.93**
	(0.45)	(0.70)	(0.83)	(0.94)	(0.43)	(0.60)	(0.75)	(0.93)

	(c) p=0.3				(d) p=0.4			
	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2
FG	**0.38**	**0.66**	**0.66**	**0.94**	0.31	**0.52**	0.58	**0.84**
	(0.37)	(0.63)	(0.63)	(0.92)	(0.33)	(0.51)	(0.57)	(0.83)
LV	**0.38**	**0.66**	**0.66**	**0.94**	**0.32**	**0.52**	**0.59**	**0.84**
	(0.38)	(0.66)	(0.66)	(0.94)	(0.32)	(0.52)	(0.59)	(0.84)
IM	**0.38**	**0.66**	**0.66**	**0.94**	0.00	**0.52**	**0.59**	**0.84**
	(0.01)	(0.26)	(0.26)	(0.94)	0.01	(0.08)	(0.26)	(0.82)
WT	**0.38**	0.65	0.65	**0.94**	**0.32**	0.51	0.55	**0.84**
	(0.38)	(0.65)	(0.65)	(0.94)	(0.32)	(0.51)	(0.55)	(0.84)
LP	**0.38**	0.56	0.64	**0.94**	**0.32**	**0.52**	0.57	0.83
	(0.38)	(0.55)	(0.64)	(0.94)	(0.32)	(0.52)	(0.57)	(0.83)

	(e) p=0.5				(f) p=0.6			
	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2
FG	**0.30**	0.44	**0.55**	**0.89**	0.23	**0.31**	0.49	**0.90**
	(0.30)	(0.43)	(0.56)	(0.86)	(0.23)	(0.30)	(0.46)	(0.87)
LV	**0.30**	**0.46**	**0.55**	**0.89**	**0.24**	**0.31**	0.48	**0.90**
	(0.30)	(0.46)	(0.55)	(0.89)	(0.24)	(0.31)	(0.48)	(0.90)
IM	0.00	**0.46**	**0.55**	**0.89**	0.00	0.00	**0.51**	**0.90**
	(0.01)	(0.10)	(0.29)	(0.87)	(0.00)	(0.08)	(0.26)	(0.87)
WT	**0.30**	0.45	0.51	**0.89**	**0.24**	0.28	0.48	**0.90**
	(0.30)	(0.45)	(0.51)	(0.89)	(0.24)	(0.28)	(0.37)	(0.90)
LP	**0.30**	**0.46**	0.51	0.87	0.00	0.00	0.41	0.86
	(0.30)	(0.46)	(0.51)	(0.87)	(0.24)	(0.00)	(0.41)	(0.86)

	(g) p=0.7				(h) p=0.8			
	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2
FG	0.20	0.48	0.48	**0.91**	0.15	**0.27**	0.43	**0.75**
	(0.19)	(0.43)	(0.43)	(0.87)	(0.16)	(0.25)	(0.42)	(0.77)
LV	**0.21**	**0.51**	**0.51**	0.88	**0.16**	0.26	**0.44**	**0.75**
	(0.21)	(0.51)	(0.51)	(0.88)	(0.16)	(0.26)	(0.44)	(0.75)
IM	0.00	0.00	0.50	0.88	0.00	0.00	**0.44**	**0.75**
	(0.01)	(0.25)	(0.25)	(0.86)	(0.01)	(0.09)	(0.25)	(0.60)
WT	**0.21**	0.45	0.45	0.82	**0.16**	0.21	0.42	**0.75**
	(0.21)	(0.34)	(0.34)	(0.82)	(0.16)	(0.20)	(0.31)	(0.75)
LP	0.00	0.00	0.46	0.88	0.00	0.00	0.40	0.70
	(0.0)	(0.0)	(0.44)	(0.86)	(0.15)	(0.00)	(0.41)	(0.70)

(i) p=0.9

	c=2, s=25	c=5, s=10	c=10, s=5	c=25, s=2
FG	**0.12**	0.27	0.43	**0.84**
	(0.10)	(0.24)	(0.41)	(0.82)
LV	**0.12**	**0.28**	**0.45**	**0.84**
	(0.12)	(0.28)	(0.45)	(0.84)
IM	0.00	0.00	0.44	**0.84**
	(0.01)	(0.01)	(0.23)	(0.71)
WT	0.11	0.23	0.41	**0.84**
	(0.05)	(0.19)	(0.20)	(0.84)
LP	0.00	0.00	0.34	0.81
	(0.00)	(0.00)	(0.39)	(0.81)

Table A.2: Modularity resulting from different clustering algorithms (IM: Infomap, WT: Walktrap, LP: Label Propagation, FG: Fast Greedy, LV: Louvain). The values for unweighted networks appear in brackets. For the weighted case, the highest value of each column is shown in bold.

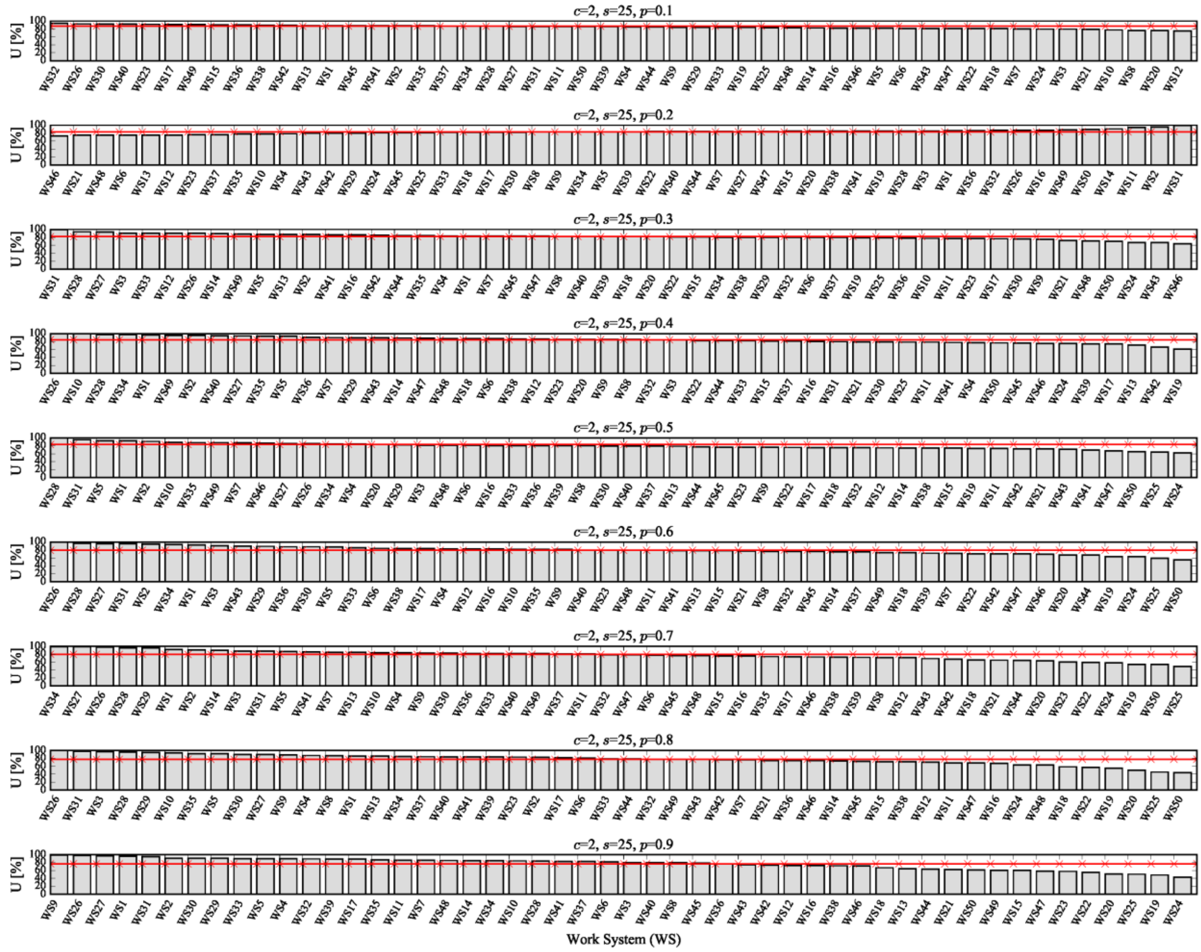


Figure A.1: Impact of parameter *p* on the mean utilization rate of individual work systems in descending order for *c*=2 and *s*=25. Jobs are released immediately after they have been entered into the system. The horizontal red line represents the mean of the given data.

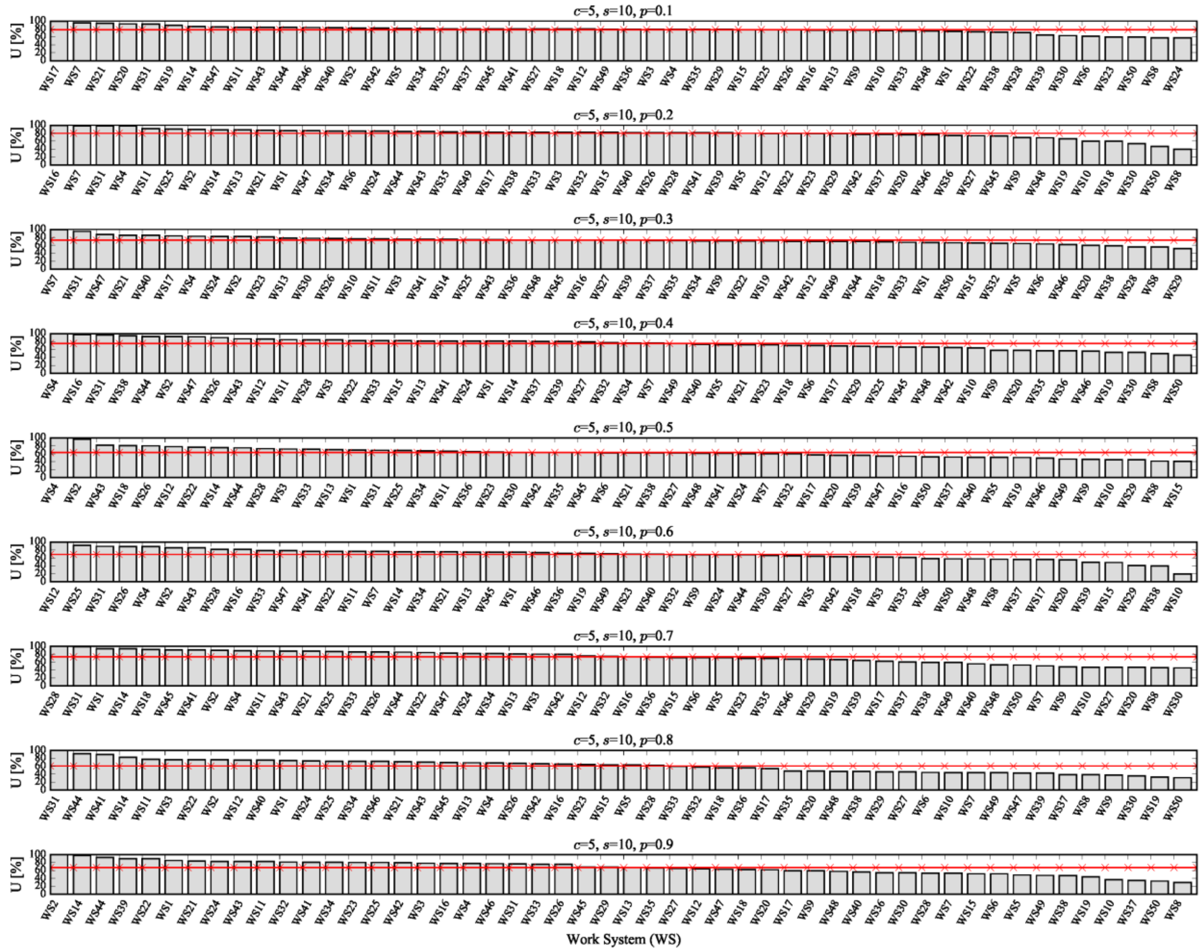


Figure A.2: Impact of the parameter *p* on the mean utilization rate of individual work systems in descending order for *c*=5 and *s*=10. Jobs are released immediately after they have been entered into the system. The horizontal red line represents the mean of the given data.

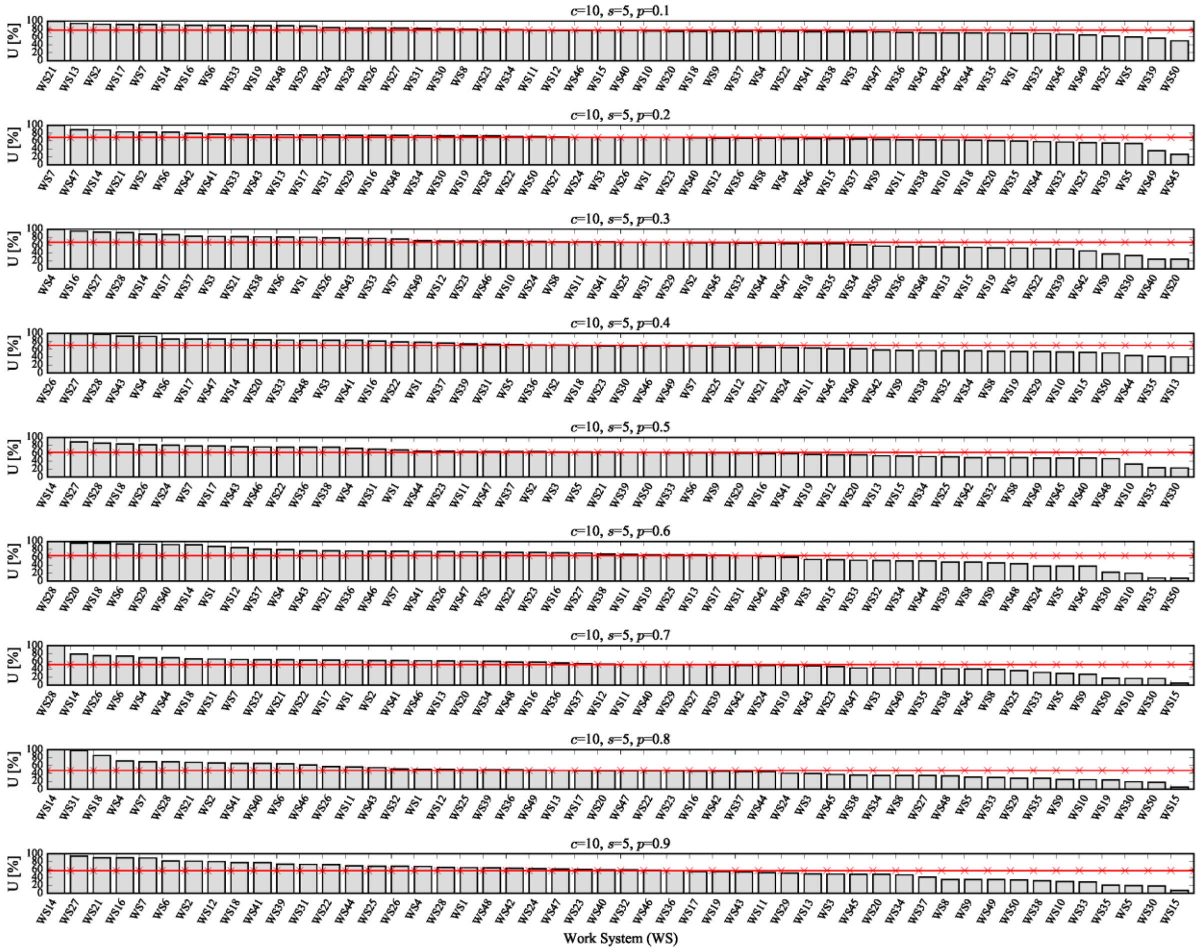


Figure A.3: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=10 and *s*=5. Jobs are released immediately after they have been entered into the system. The horizontal red line represents the mean of the given data.

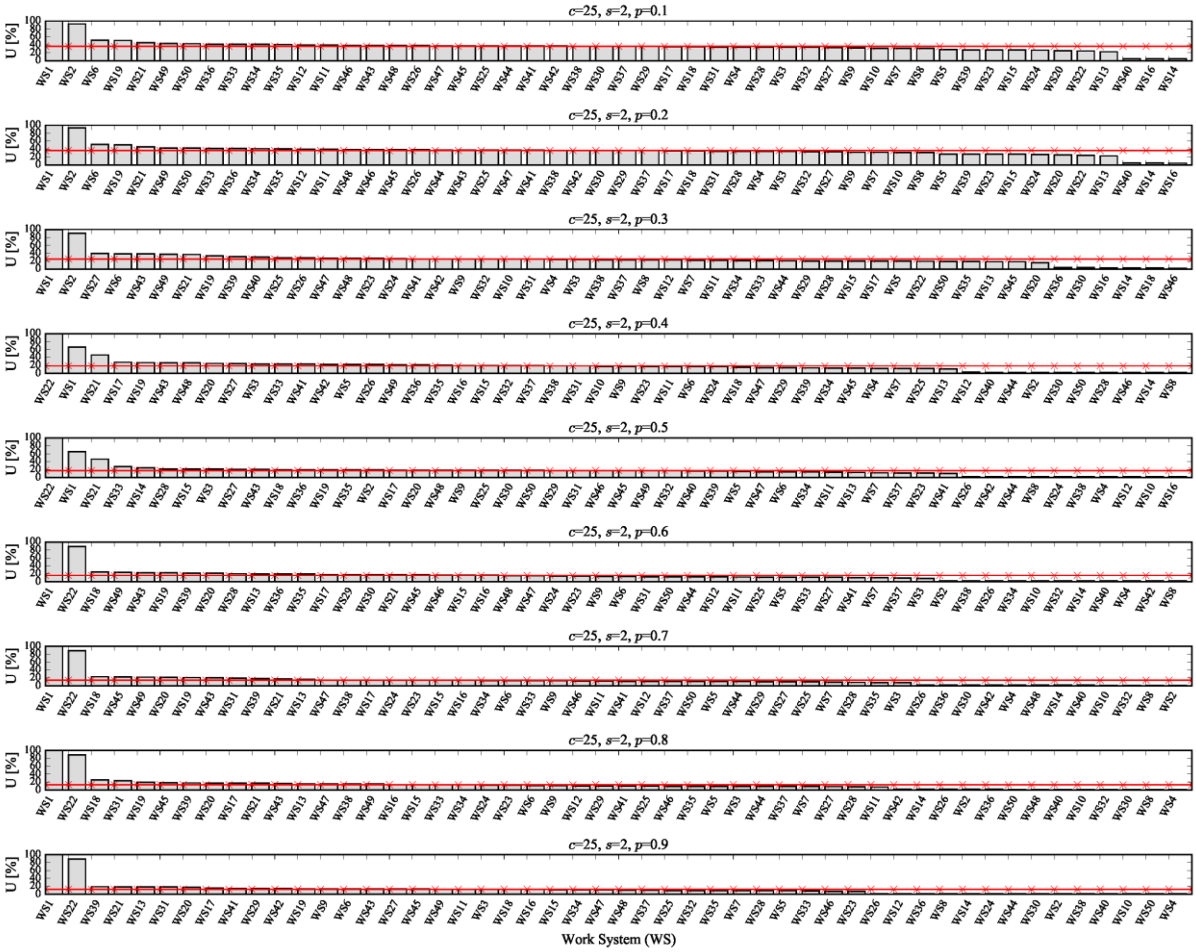


Figure A.4: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=25 and *s*=2. Jobs are released immediately after they have been entered into the system. The horizontal red line represents the mean of the given data.

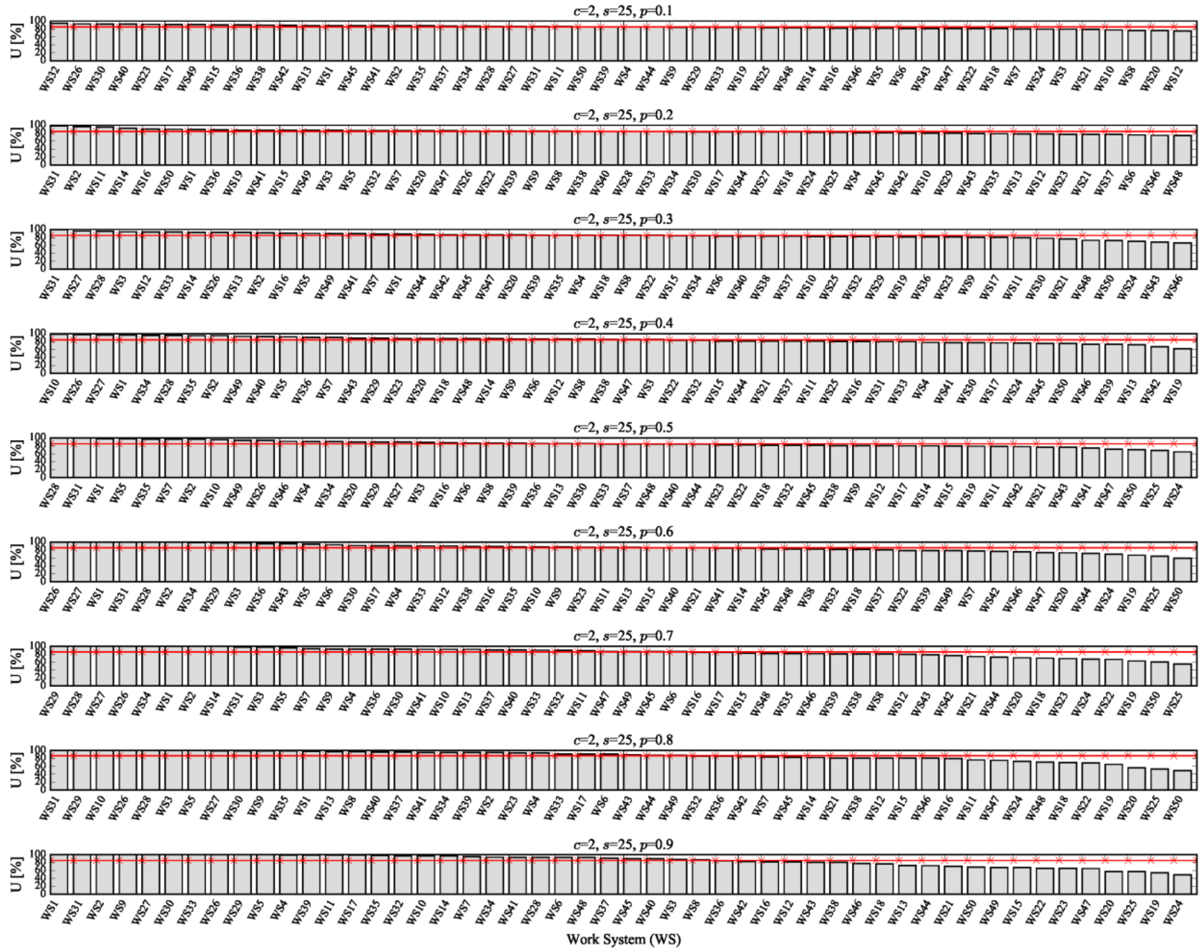


Figure A.5: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=2 and *s*=25 after applying ConWIP. The horizontal red line represents the mean of the given data.

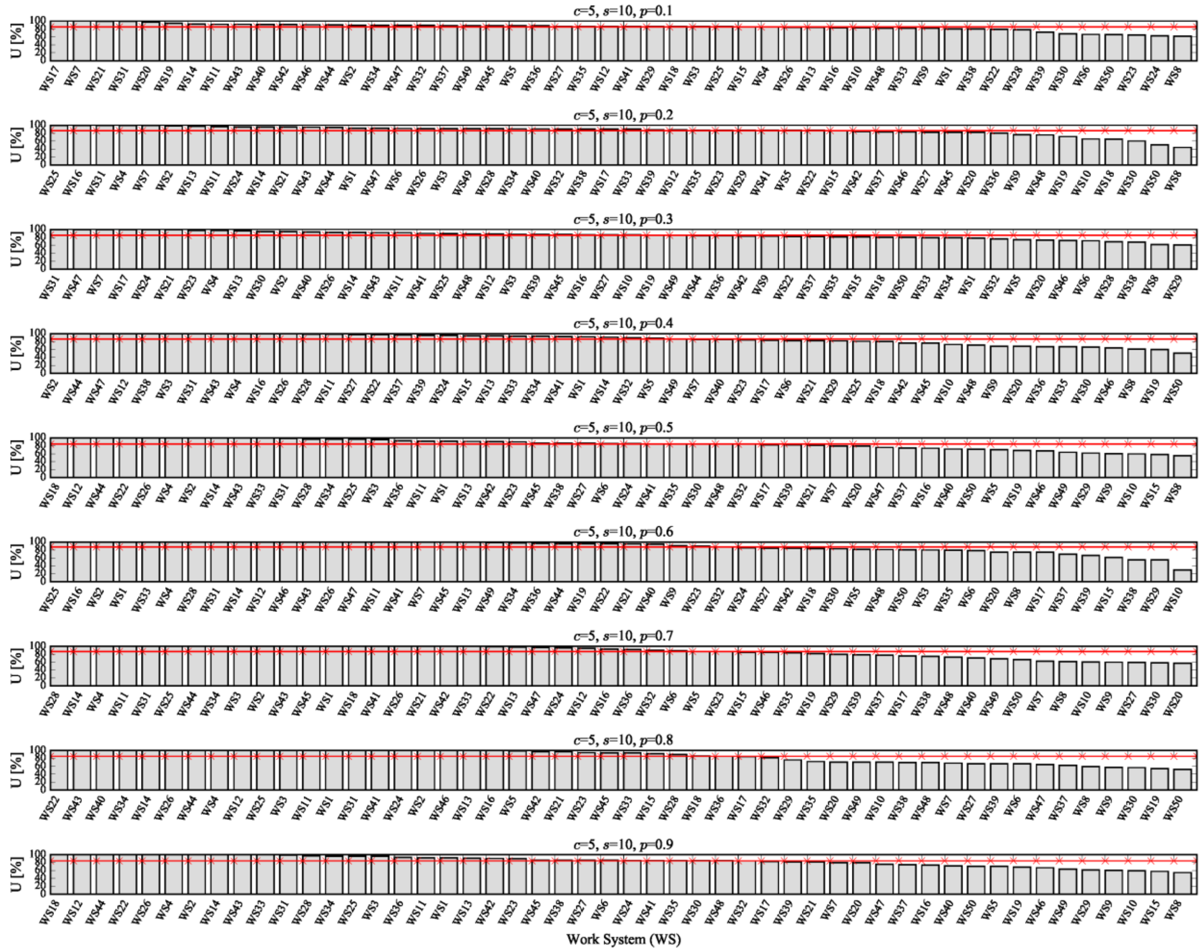


Figure A.6: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=5 and *s*=10 after applying ConWIP. The horizontal red line represents the mean of the given data.

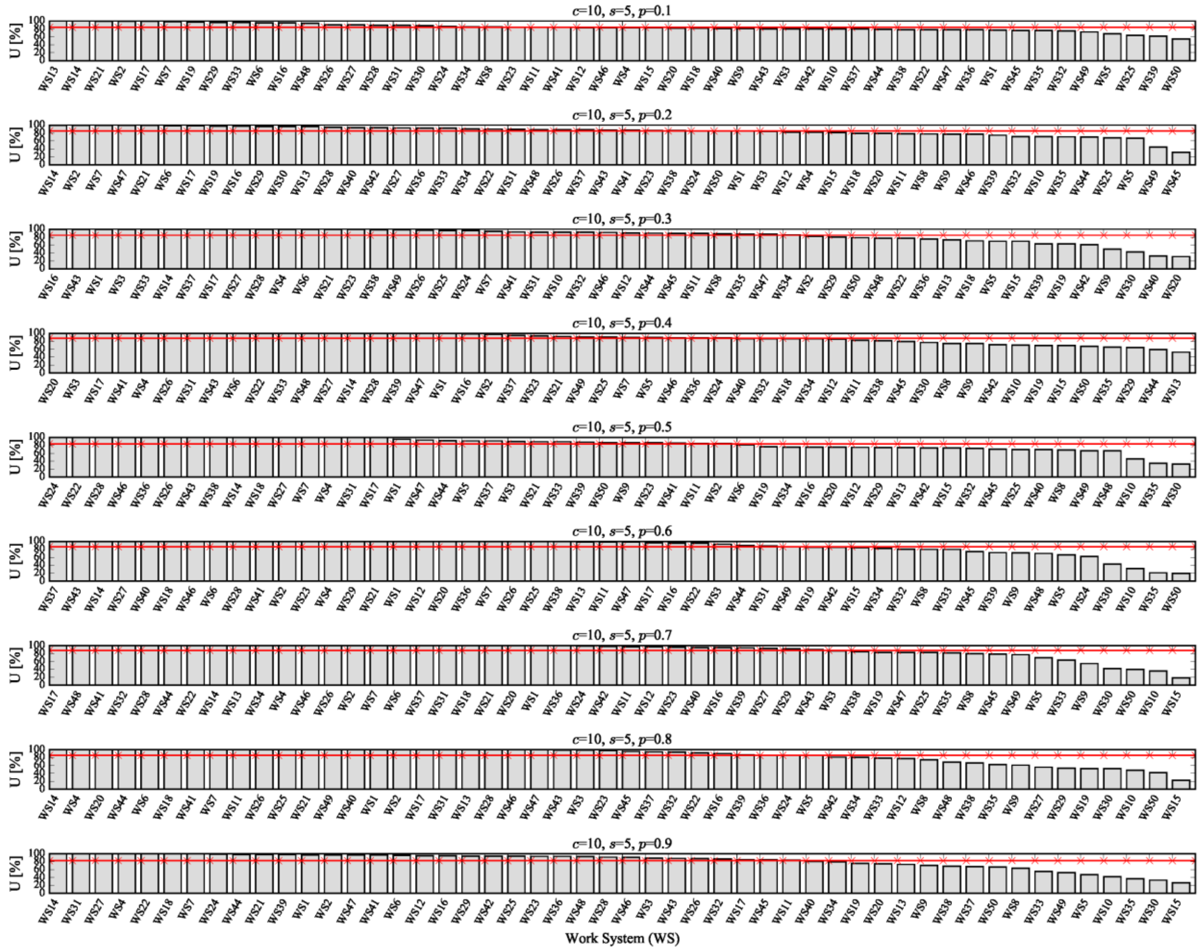


Figure A.7: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=10 and *s*=5 after applying ConWIP. The horizontal red line represents the mean of the given data.

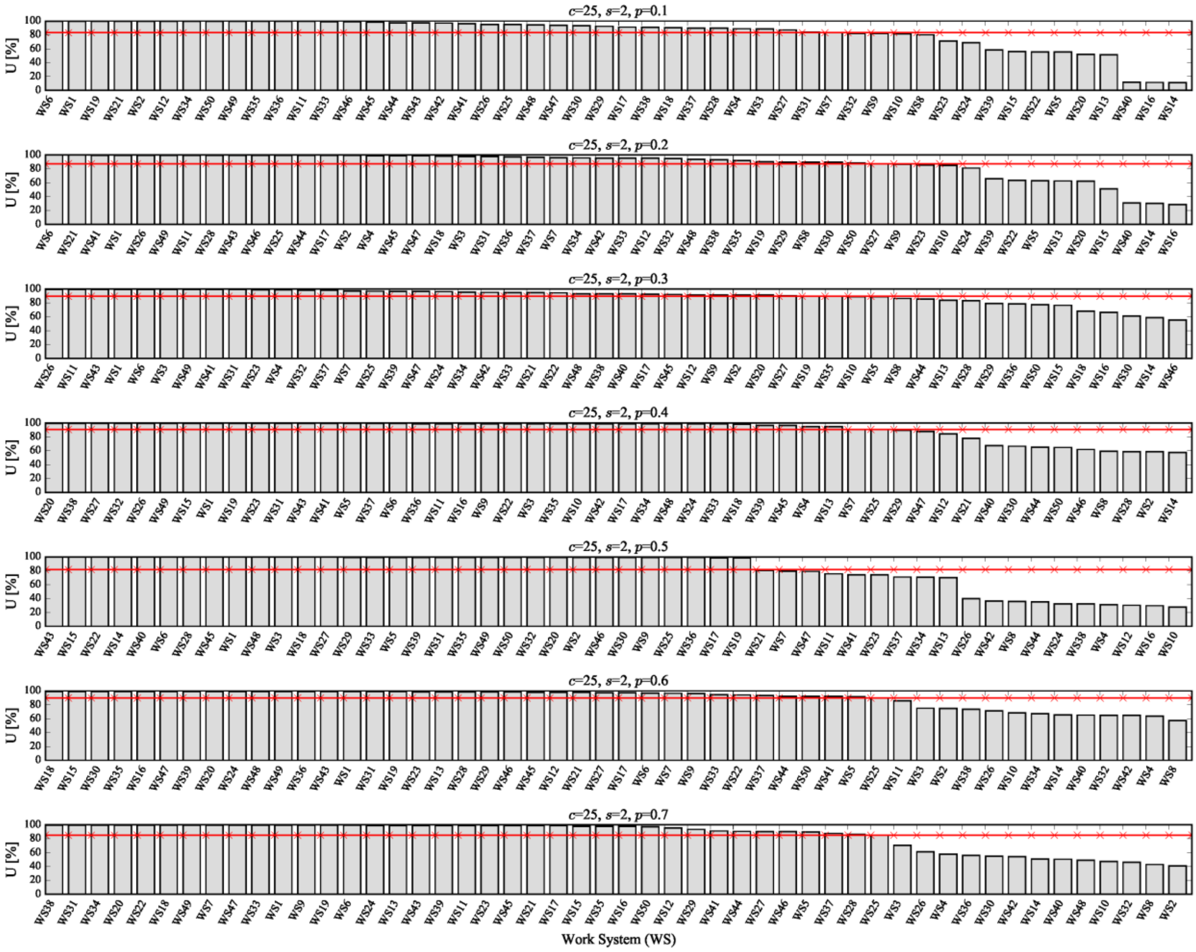


Figure A.8: Impact of the parameter *p* on the mean utilization rate of individual work systems, in descending order, for *c*=25 and *s*=2 after applying CONWIP. The horizontal red line represents the mean of the given data.

Network Model (c, s, p)	**Best Rule**	**Hybrid Clustering-based Approach with FG**	**Hybrid Clustering-based Approach with LV**	**Hybrid Clustering-based Approach with IM**	**Hybrid Clustering-based Approach with WT**	**Hybrid Clustering-based Approach with LP**
2, 25, 0.1	571	611	611	611	611	611
2, 25, 0.2	846	861	861	861	861	861
2, 25, 0.3	840	891	891	891	891	891
2, 25, 0.4	920	950	953	-	953	953
2, 25, 0.5	3214	**2229**	**2229**	-	**2229**	-
2, 25, 0.6	2173	**1892**	**1869**	-	**1869**	-
2, 25, 0.7	3214	**2307**	**2229**	-	**2229**	-
2, 25, 0.8	4292	**3297**	**3404**	-	**3404**	-
2, 25, 0.9	4299	**3486**	**3009**	-	**3211**	-
5, 10, 0.1	2238	**1872**	**1872**	**1872**	**1872**	**1872**
5, 10, 0.2	2334	**1816**	**1816**	**1816**	**1816**	**1816**
5, 10, 0.3	3261	**2240**	**2240**	**2240**	**2240**	**2240**
5, 10, 0.4	4592	**3759**	**3759**	**3759**	**3759**	**3759**
5, 10, 0.5	4907	**4568**	**4309**	**4309**	**4309**	**4626**
5, 10, 0.6	4821	4906	**4610**	-	4752	-
5, 10, 0.7	4766	**4615**	**3867**	-	**3923**	-
5, 10, 0.8	4377	4315	4412	-	**4044**	-
5, 10, 0.9	4343	4331	**3966**	-	**4083**	-
10, 5, 0.1	2070	**1737**	**1737**	**1737**	**1737**	**1737**
10, 5, 0.2	2706	**2159**	**2159**	**2159**	**2159**	**2159**
10, 5, 0.3	4686	**3724**	**3724**	**3811**	**4303**	**3820**
10, 5, 0.4	4380	**3845**	**3610**	**4127**	**4023**	**3848**
10, 5, 0.5	4522	**4258**	**4172**	4411	**4312**	4480
10, 5, 0.6	3489	3454	**3337**	**3371**	**3258**	3445
10, 5, 0.7	3417	3336	**3111**	**3232**	3412	**3244**
10, 5, 0.8	3231	3351	3202	3109	3163	3299
10, 5, 0.9	2944	3022	2987	3272	3022	2944
25, 2, 0.1	3721	**3583**	**3583**	**3583**	**3583**	**3583**
25, 2, 0.2	3909	**3637**	**3637**	**3637**	**3637**	**3637**
25, 2, 0.3	1186	1226	1226	1226	1226	1226
25, 2, 0.4	669	617	617	617	617	617
25, 2, 0.5	662	655	655	655	655	655
25, 2, 0.6	679	663	663	663	663	663
25, 2, 0.7	634	614	614	614	614	614

Table A.3: Results for mean tardiness [time units] over all tardy jobs for all clustering algorithms: (IM: Infomap, WT: Walktrap, LP: Label Propagation, FG: Fast Greedy, LV: Louvain). The missing values (-) result from the presence of only one giant cluster. Key: number of clusters, *c*; cluster size, *s*; and degree of interconnection among individual clusters, *p*. Values in bold indicate that the hybrid clustering-based approach led to better performance.

Network Model (c, s, p)	Best Rule	Hybrid Clustering-based Approach with FG	Hybrid Clustering-based Approach with LV	Hybrid Clustering-based Approach with IM	Hybrid Clustering-based Approach with WT	Hybrid Clustering-based Approach with LP
2, 25, 0.1	97.95	97.81	97.81	97.81	97.81	97.81
2, 25, 0.2	97.20	96.43	96.43	96.43	96.43	96.43
2, 25, 0.3	95.02	95.18	95.18	95.18	95.18	95.18
2, 25, 0.4	94.35	94.20	94.50	-	94.50	94.50
2, 25, 0.5	91.24	**92.16**	**92.16**	-	**92.16**	-
2, 25, 0.6	88.55	**89.15**	**89.32**	-	**89.32**	-
2, 25, 0.7	86.33	**87.70**	**87.82**	-	**87.82**	-
2, 25, 0.8	82.68	**84.38**	**84.37**	-	**84.37**	-
2, 25, 0.9	81.95	82.20	82.43	-	**83.40**	-
5, 10, 0.1	92.60	92.60	92.60	92.60	92.60	92.60
5, 10, 0.2	91.72	91.90	91.90	91.90	91.90	91.90
5, 10, 0.3	90.87	90.76	90.76	90.76	90.76	90.76
5, 10, 0.4	80.66	**81.77**	**81.77**	**81.77**	**81.77**	**81.77**
5, 10, 0.5	74.89	**77.70**	**77.20**	**77.20**	**77.20**	**76.50**
5, 10, 0.6	74.16	**74.70**	**76.88**	-	**75.20**	-
5, 10, 0.7	73.82	74.50	**74.99**	-	**78.70**	-
5, 10, 0.8	72.07	**73.00**	**74.88**	-	**75.70**	-
5, 10, 0.9	71.37	71.00	71.15	-	70.70	-
10, 5, 0.1	89.83	**90.59**	**90.59**	**90.59**	**90.59**	**90.59**
10, 5, 0.2	87.78	**88.22**	**88.22**	**88.22**	**88.22**	**88.22**
10, 5, 0.3	75.65	**80.93**	**80.93**	75.90	**77.90**	**81.30**
10, 5, 0.4	78.73	**81.80**	**80.71**	77.80	78.90	**79.70**
10, 5, 0.5	72.38	**77.60**	**78.51**	72.70	**73.70**	**73.90**
10, 5, 0.6	71.14	**72.30**	**75.40**	**75.00**	**74.80**	70.50
10, 5, 0.7	72.02	69.00	**72.78**	72.00	72.40	**73.70**
10, 5, 0.8	73.32	73.10	**75.50**	72.70	73.00	67.70
10, 5, 0.9	73.09	73.70	**73.95**	73.10	72.20	73.50
25, 2, 0.1	78.23	**81.86**	**81.86**	**81.86**	**81.86**	**81.86**
25, 2, 0.2	76.94	**80.52**	**80.52**	**80.52**	**80.52**	**80.52**
25, 2, 0.3	82.18	82.26	82.26	82.26	82.26	82.26
25, 2, 0.4	86.99	87.11	87.11	87.11	87.11	87.11
25, 2, 0.5	83.20	83.03	83.03	83.03	83.03	83.03
25, 2, 0.6	82.73	83.33	83.33	83.33	83.33	83.33
25, 2, 0.7	83.01	83.28	83.28	83.28	83.28	83.28

Table A.4: Results for schedule reliability [%] within a tolerance window of 15 % for all clustering algorithms: (IM: Infomap, WT: Walktrap, LP: Label Propagation, FG: Fast Greedy, LV: Louvain). The missing values (-) occurred because there was only one giant cluster. Key: number of clusters, *c*; cluster size, *s*; and degree of interconnection bet-ween individual clusters, *p*. Values in bold indicate that the hybrid clustering-based approach led to better performance.

Eidesstaatliche Erklärung

Hiermit erkläre ich, dass ich die beigefügte Dissertation selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel genutzt habe. Alle wörtlich oder inhaltlich übernommenen Stellen habe ich als solche gekennzeichnet. Ich versichere außerdem, dass ich die beigefügte Dissertation nur in diesem und keinem anderen Promotionsverfahren eingereicht habe. Die zu Prüfungszwecken beigelegte elektronische Version der Dissertation ist identisch mit der abgegebenen gedruckten Version.

Bremen, 1. Dezember 2022

Darja Wagner-Kampik