UNIVERSITY OF SALERNO

DEPARTEMENT OF BUSINESS SCIENCE, MANAGEMENT
AND INNOVATION SYSTEMS

Ph.D. Course in Big Data Management – XXXIII Cycle

SCATTERED MANUFACTURING DEVELOPING A CLOUD
MANUFACTURING FRAMEWORK BASED ON
AUTONOMOUS RESOURCES

Supervisor
Prof. Massimo de Falco

Ph.D. Candidate
Nicola Mastrandrea

Scientific Referees
Prof. Teresa Murino
Prof. Guido Guizzi

Ph.D. Program Coordinator
Prof. Valerio Antonelli

Abstract

Cloud Manufacturing is a resource-sharing paradigm that provides on- demand access to a pool of manufacturing resources and capabilities to utilize geographically scattered resources in a service-oriented model. These services are rapidly provisioned and released with minimal management effort via the Industrial Internet of Things and its underlying IT infrastructure, architecture models, and data and information exchange protocols and standards. In this context, the tradeoff between resources' autonomy and independence exigencies and platform needs for centralized control and coordination is a crucial enabler factor for implementing such vertically or horizontally integrated cyber-physical systems for intelligent manufacturing. The introduction of resources autonomy and network independence in a distributed cloud manufacturing system enables platforms with equal and open access to shared resources in a more sustainable way and potentially with higher scalability of manufacturing resources and capabilities.

This work aims to develop a framework to manage distributed operations in cloud manufacturing based on autonomous resources. This research investigates network architectures in the context of distributed Cloud Manufacturing systems with autonomous and independent resources to identify critical parameters that determine whether an efficient deployment is viable for a given scenario.

The framework includes: (i) a network architecture for a distributed Cloud Manufacturing platform based on autonomous nodes; (ii) a Multi-agent Systems architecture for managing communications and coordination issues in distributed operations; (iii) an implementation of the proposed network architecture in the context of large Additive Manufacturing networks; (iv) a unique optimization algorithm that combines scheduling and logistics issues inside such network. Additionally, an implementation of the Multi- Agent Systems architecture has been developed to offer practical guidance for implementing the framework into context closer to the industry and real life.

A literature review was conducted to analyze the research area to accomplish the goal and objectives of this work. Next, a framework was outlined to identify, assess, and control dynamics and issues inside the network. Two well-known and established approaches were used to implement the communication and coordination system and the optimization of the platform in this research: Multi-agent Systems to tackle the dynamic task arrival, the downtime of machines, the identification of the anomalous tasks; and Operation Research techniques to tackle

logistics and to schedule global optimization for a job order.

Results from this work are beneficial for both academia and industry in understanding aspects involving new varieties of cloud manufacturing networks. The principal contribution is a framework that offers new insights and outlines new issues on how to deal with autonomous and independent resources inside a Cloud Manufacturing platform and how to manage global optimization and long-term sustainability of such networks. Finally, this study also introduced a novel cloud manufacturing taxonomy, including a list of actors, a list of platform services and functionalities.

Contents

List of Figures

List of Tables

List of Publications & Awards

1. Journal Articles

   (a) D'Aniello, Giuseppe, Massimo De Falco, and Nicola Mastrandrea. "Designing a Multi-Agent System Architecture for Managing Distributed Operations within Cloud Manufacturing." Evolutionary Intelligence, April 1, 2020. https://doi.org/10.1007/s12065-020-00390-z

2. Book Chapters

   (a) de Falco, Massimo, Nicola Mastrandrea, Wathiq Mansoor, and Luigi Rarità. "Situation Awareness and Environmental Factors: The EVO Oil Production." In New Trends in Emerging Complex Real Life Problems, edited by Patrizia Daniele and Laura Scrimali, 1:209–17. AIRO Springer Series. Cham: Springer International Publishing, 2018. https://doi.org/10.1007/978-3-030-00473-6_23

3. Conference Papers

   (a) de Falco, Massimo, Nicola Mastrandrea, and Luigi Rarità. "Integrating Capacity and Logistics of Large Additive Manufacturing Networks." Procedia Manufacturing 39 (2019): 1421–27. https://doi.org/10.1016/j.promfg.2020.01.310

   (b) de Falco, Massimo, Nicola Mastrandrea, and Luigi Rarità. "A Queueing Networks-Based Model for Supply Systems." In Optimization and Decision Science: Methodologies and Applications, edited by Antonio Sforza and Claudio Sterle, 217:375–83. Springer Proceedings in Mathematics & Statistics. Cham: Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319- 67308-0_38

   (c) Best Paper Award Winner: De Falco, Massimo, Luigi Rarità, and Abdallah Asan Alalawin. "Negotiating and Sharing Capacities of Large Additive Manufacturing Networks." International Conference on Advances in Business, Management and Law (ICABML) 2017 1,

no. 1 (December 24, 2017): 440–66. https://doi.org/10.30585/icabml-cp.v1i1.37

Chapter 1

Introduction

1.1   Research Background

Decentralization and sustainable resource sharing are key drivers for success in today's globalized economy. From craftsmanship to Agile and Intelligent Manufacturing, production has become increasingly complex, depending upon new technological developments and advances in Information and Communication Technologies in response to changes in local and global markets [1]. Moreover, this context and market trends such as mass customization pose new challenges to industries and researchers. The process of sharing resources and assets efficiently on a global scale requires high interoperability, flexibility, and agility in manufacturing systems to respond to rapid changes. Therefore, the rapid evolution of markets and advances in key enabling technologies have introduced the distributed manufacturing paradigm. This paradigm aims to share geographically scattered manufacturing resources and capabilities and already profoundly impact current systems.

While the introduction of state-of-the-art technologies presents positive benefits for manufacturing enterprises over competitors, new issues in implementing these network technologies that affect production occur within the manufacturing industry. Most of these issues involve sharing manufacturing resources, where these resources, centralized into a central network, are not distributed efficiently through the platform due to a lack of global coordination in manufacturing services management in the network. And, secondly, the inability to access the independent manufacturing complex resources (equipment) in the manufacturing network due to complications in transferring hardware resources into the network [2][3].

Much of the shift towards new paradigms, indeed, is driven by the emergence of Big Data, and the issues connected to the ways by which industrial operations collect, manage and interpret their data remain prevalent[4]. Considerations about Big Data and the treatment of large datasets are an intrinsic challenge of each system operating in an Industry 4.0 scenario. Traditional statistical

processing methods are often useless due to the complexity and the sheer size of large datasets. Current implementations have demonstrated adaptive scheduling, real-time modelling of processes, and Decision Support Systems used to refine processes and component design[5]. For the optimization of issues within the context of production and logistics, a typical aim is gaining quantitative improvements, which also correspond to an increase in resource efficiency[6]. Sometimes new manufacturing models arise as such a situation leads to increasing adoption of new production technologies. The challenge with distributed production is to implement communication and integration technologies that reduce the coordination effort and provide a focused platform[7].

Building innovative models around the notion of being "globally virtual, locally physical" calls for a service-dominant logic of distributed resources in which reusable services models, shaped according to the concept of Manufacturing as a Service, represent homogeneous production processes [8]. Therefore, the ongoing servitization process in the manufacturing industry is progressively shifting the view of traditional resources as a set of services and solutions that supplement companies' traditional offerings consumed on an ad-hoc basis[9]. As a result, enterprises increase their capability to provide manufacturing services and offer more extensive and more complex jobs. Moreover, Cloud Manufacturing, with the proper implementation,presents the capability to transform and restructure manufacturing systems and move the entire industry from production-oriented manufacturing to service-oriented manufacturing[3]. Cloud Manufacturing can also be a significant factor to reduce costs, maximize productivity, reduce time to market, and increase business agility and innovation[10], as well as facilitating the whole life cycle of manufacturing, providing safe, reliable, high-quality, cheap, and on-demand manufacturing services[11].

Other potential benefits from the introduction of Cloud Manufacturing are the following [10]:

1. Virtual access to homogenous and interoperable manufacturing services over the cloud, reducing the need to invest, develop, maintain, and manage hardware and software manufacturing resources.

2. Higher utilization rates of manufacturing resources through the promotion of shared pools of resources.

3. Higher Scalability, encouraging Cloud Manufacturing users to control production capacity to balance the current demand dynamically.

4. The introduction of novel utility-based cost schemes that assigns costs based on user/provider resources consumptions.

5. An on-demand approach that endorses users to have ubiquitous access and natural human-computer interaction to manufacturing resources.

Main issues for enabling the transition to cloud manufacturing, as recent research efforts have summarized the main challenges for cloud manufacturing as follows:

1. Unclear principles for the protection of the end-user investment. The new business model that comes with cloud manufacturing requires fresh perspectives on the protection of rights.

2. Difficulty in communication and interaction between departments within the enterprise and among the stakeholders within the supply chain due to different systems with different focuses.

3. Limited collaboration and interaction between business partners within cloud manufacturing.

4. Absence of a readily available implementation framework for cloud manufacturing services. Each company has to implement this as a new system.

5. Difficulty in the deployment of physical resources, such as machines, monitors, and facilities. These issues are mainly due to the unpreparedness of a large portion of resources for the required connectivity.

This research attempts to answer some of these issues. In particular, an attempt to formalize the main founding principles that a Cloud Manufacturing platform should obey (see Chapter III Section 2). Moreover, a Multi-Agent Systems architecture for distributed operations is provided to identify the key process parameters for selecting communication approaches within service providers and service demanders. Finally, an implementation framework is depicted in the context of a large Additive Manufacturing Network scenario. The architectural model is used to simulate communications and operations in the scenario, while the implementation model is used to define an optimization algorithm to manage both scheduling and logistics problems using one cycle of negotiation.

1.2 Thesis Outline

This Thesis is divided into five chapters, as shown in Figure 1.1. Chapter I provides a background and general overview of the research project, followed by an introduction of the research motivation, research scope, research aim,and objectives. The first chapter also outlines the remaining chapters of the Thesis. Chapter II provides reviews of the literature on two main concepts: Cloud Manufacturing and Cloud Manufacturing Architecture. In phase one of the literature review, the focus was on cloud manufacturing and its types, characteristics, and attributes. In phase 2, the focus was on understanding architectures and exploring the role of autonomy and independence of resources in distributed manufacturing systems and their effects in the cloud environment. Phase 2 also identifies the research gap. Chapter III develops a framework to manage autonomous resources in cloud manufacturing. This chapter begins by introducing and explaining the phases of development of the framework. It then explores the process of identifying differences with frameworks available from literature in a detailed comparison. Then, it outlines in-depth platform

actors, roles, functionalities, and service management systems through an analysis of main platform factors, dynamics, and governance. Chapter IV presents two implementing models of the proposed architecture. In model 1, the focus was on developing a Multi-Agent Systems architecture for distributed operations in presence of autonomous service providers. In model 2, the focus was on developing an optimization model that combines localization, fragmentation, assignment, and picking issues for a specific job order in a large Additive Manufacturing Network. Chapter V summarizes the results, draws conclusions, and makes recommendations for future work. This chapter presents outcomes, including the research contribution to knowledge, research limitations, and future work. Also, it reveals answers to the research aim and objectives and presents the overall research conclusion.
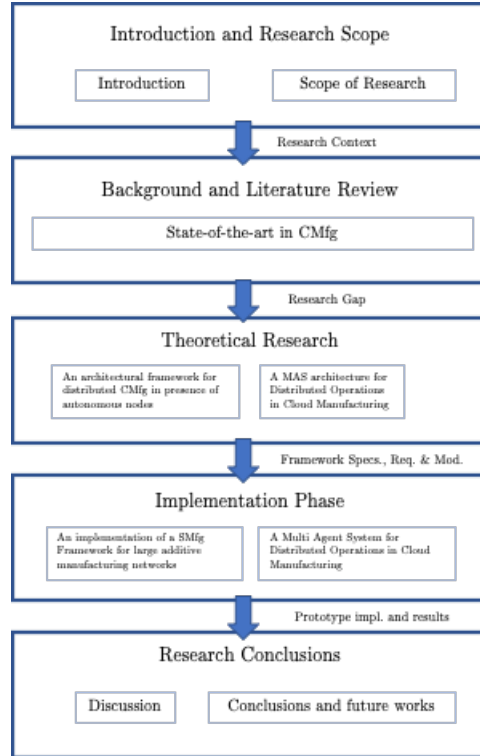
Figure 1.1: Thesis Outline

Chapter 2

The Scope of Research

This chapter outlines the aim of the research within its objectives and boundaries. The following sections define the boundaries of Cloud Manufacturing systems and explain the scope of research to identify the Author's perspective on applying distributed Cloud Manufacturing systems within an autonomous resources scenario.

2.1 Research Motivation and Gaps

This work is motivated by the need for practicable and applicable Cloud Manufacturing systems that can be temporary and dynamically created ad hoc to satisfy specific market demand in a sustainable way.

The transformation of existing manufacturing systems to new advanced and complex systems, such as Cloud Manufacturing, can be seen as a big challenge for any enterprise. This transformation poses new uncertainties in the new system that can impact every aspect of the operations lifecycle from design and engineering to the implementation final operations of the new manufacturing model. So, there is a need to understand and tackle uncertainties in cloud manufacturing networks derived from the introduction of resource autonomy and resource independence from a specific platform. To address these issues, steps needed to be followed, including understand and define key factors, main actors, and dynamics inside such networks; identify main issues that arise from the trade-off between decentralized governance and the need for centralized control in global scheduling, load balance and logistics optimization to provide long-term sustainability of the network; and develop a framework to implement such networks in a Cloud Manufacturing environment.

2.2 Aims and objectives of the research

The research aim is to develop a framework to manage operations in cloud manufacturing for autonomous resources. The framework comprises a taxonomy of the proposed architecture; a Multi-Agent System model to tackle coordination and communication issues; a detailed list of platform services, agents, and functionalities; a unique algorithm to determine local optimization in a job order combining logistics and distributed multi-task scheduling optimization; and the implementation process of a prototype with basic functionalities of the Multi-Agent System model.

Previous research has shown that most Cloud Manufacturing architectures require central governance and high investment for increasing efficiencies and capabilities across the product life cycle. This research aims to investigate the possibility for a Cloud Manufacturing platform constituted by independent and autonomous service providers and a set of clear founding principles to be deployable and viable for a homogenous manufacturing scenario.

The following objectives have been identified to track the progress of the research and ensure that the aim is achieved:

1. Identification and analysis of existing research gaps in the context of Cloud Manufacturing Architectures.

2. Development of a framework for a sustainable Cloud Manufacturing platform constituted by autonomous service providers

3. Realization of implementation models for critical areas inside the framework

4. Validation of the proposed models

2.3 Research Methodology

The following steps, as shown in Figure 2.1, will be undertaken to verify the validity of the proposed framework and achieve the research aim:

1. Review of the relevant literature on industry 4.0, cloud computing, cloud manufacturing, and smart manufacturing

    (a) Studies of Cloud Manufacturing: The state-of-the-art of Cloud Manufacturing will be reviewed to identify and demonstrate its impact.

    (b) Review of cloud manufacturing frameworks regarding governance, architecture layouts, scheduling methods, virtualization of manufacturing resources and capabilities.

2. Selecting a cloud manufacturing approach

(a) In this section, a review of cloud manufacturing frameworks will be conducted, and the results are analyzed based on functional requirements, business constraints, and technology constraints to adopt a suitable approach for system deployment.

3. Designing of a Cloud Manufacturing framework

(a) Based on research gaps identified in the previous steps and the outcomes of the last research step, a theoretical framework will be formulated to address cloud manufacturing system requirements. Additionally, a Cloud Manufacturing network will be implemented to form the baseline for analyzing the optimization problem and identifying critical parameters for a deploying approach.

4. Implementation and validation of the proposed architecture

(a) Development of implementation models of the proposed architecture focusing on specific critical areas.

(b) Validation through Multi Agents System simulation and numerical examples of the analytical optimization model.
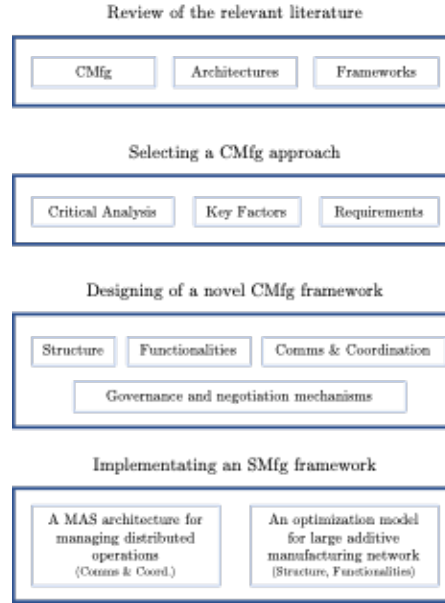
Figure 2.1: Research Steps

2.4   Scope of the Thesis

This research work focuses on identifying Cloud Manufacturing networks and their characteristics. It involved detecting and evaluating key factors at the architectural level and the implementation level within an autonomous manufacturing resources scenario. Due to the novel nature of the research that concerns a relatively new research field such as Cloud Manufacturing, the Author's approach was to apply well-known methods and theory inside this new context. Moreover, this research concentrated mainly on the architectural level and the related issues identified.
The overall research objectives are the following:

1. Identification and analysis of existing models and gaps presented in the literature.

2. Formulation of the operational context in the given scenario and its types, characteristics, and attributes.

3. Development of a novel architectural model based on autonomous resources.

4. Focusing on critical areas of the framework to implement the model.

Chapter 3

State of the art in Distributed Cloud Manufacturing: a Review

3.1 Introduction

This chapter presents a review of the state-of-the-art in Cloud Manufacturing and its related approaches and enabling key technologies. Techniques used for Cloud Manufacturing design are investigated, followed by a review of Cloud Manufacturing service management aspects. Furthermore, a review of Cloud Manufacturing architectures is provided. The result of this analysis is then used to identify gaps in the research field. This chapter aims to present, in a clear view, a unified picture regarding Cloud Manufacturing, its architectures, and applications. Hence, to provide a holistic view of the phenomena, prior research and frameworks presented in the field relevant to the research question have been analyzed. Therefore, the literature review focuses on two main concepts: cloud manufacturing and cloud manufacturing architectures. The search in academic database engines was limited to keywords related to the research topics.

Previous publications, research, and knowledge have been investigated to identify the need for the research and rationalize the research path. An alignment between the research goal and issues that have not been covered satisfyingly has been addressed during the process.

The search strings used in the research process are the following:

1. TITLE-ABS-KEY("Cloud Manufacturing")

2. (TITLE-ABS-KEY("Cloud Manufacturing") AND TITLE-ABS-KEY(Architecture) OR TITLE-ABS-KEY(Framework))

To define the inclusion criteria, mentioned search terms were considered, and

Options	Delimitation
Field	Title, Abstract, Keywords
Time	2010-2020
Document Type	Article or Review
Language	English

Table 3.1: Database search delimitations

Screening Stage	Stage Name	Description
1	Title Screening	Inclusion of search terms in title, abstract or keywords
2	Abstract Reading	Direct mention of cloud manufacturing context, aspects, implications, concept, algorithms, paradigms methods and/or models in the abstract
3	Full Text Screening	Relevance and contribution to the aim of the research and the research questions

Table 3.2: Publications selection process after each screening stage

based on them, a set of search terms were included in the search process. The database search was conducted from 2010 to 2020 since Cloud Manufacturing is an emerging and trending topic. Furthermore, only papers in the English language have been included. Table 1 represents the delimitations, inclusion, and exclusion criteria designed for the first screening stage.

The mentioned search terms were used for finding literature based on the inclusion of the search terms in the title, abstract, or keywords section of publications for the first screening stage. For the second screening stage, abstracts of all the selected literature were read to identify publications that might be used in the third stage that included reading through the publications. Table 2 represents the designed guideline for selecting publications after each screening stage in this literature review.

In the first part of the literature review, the focus was on Cloud Manufacturing and its types, characteristics, and attributes. The results from this phase are the following:

- Understand the cloud manufacturing concept by exploring various definitions of Cloud Manufacturing.

- Show latest Cloud Manufacturing frameworks.

- Identify Cloud Manufacturing key architectural factors.

- Detect Cloud Manufacturing research challenges and gaps.

3.2 Cloud Manufacturing

The development of new advanced manufacturing modes with the flexibility to suit the market is becoming one of the main trends of the manufacturing industry nowadays. A number of advanced manufacturing models, such as Agile Manufacturing [12], Virtual Manufacturing [13], and Networked Manufacturing, are flourishing in this context. Cloud Manufacturing was introduced in 2010 to overcome the impediments to applying Networked Manufacturing and solve more complex manufacturing problems and perform larger-scale collaborative manufacturing [9].
The evolution of key enabling technologies brought a growing unpredictability of the markets, and with increased competition, manufacturing systems boundaries are extended from a factory towards new kinds of network relationships. As a result, enterprises' mission and business strategy have also changed, e.g., from product competitive advantage towards collaborative added value, and the way enterprises perform business have been transformed [14]. Consequently, a wide range of different paradigms emerged, such as Lean Manufacturing, Agile Manufacturing, Flexible Manufacturing, reconfigurable manufacturing systems, distributed virtual manufacturing systems.
Agile Manufacturing systems are designed to respond to customer and market changes quickly. Although lean and agile manufacturing concepts sound similar, they have different approaches to manufacturing engineering systems. While Lean Manufacturing responds to competitive pressure with limited resources, agile Manufacturing represents the response to complexity brought about by constant change. Flexible manufacturing systems are manufacturing systems designed to rapidly adjust their production capacity and functionality in response to new circumstances by rearranging or changing their components. Networked Manufacturing systems combine advanced manufacturing technologies with network technology to introduce Distributed Manufacturing systems through the Internet. Networked Manufacturing models provide information and resource sharing among enterprises but lack direct access to physical resources, nor does it achieve the dynamic intelligent sharing and distribution of manufacturing resources. Intelligent manufacturing systems bring those features. These are manufacturing systems enhanced with human-like capabilities [14]. Cloud manufacturing is emerging as a manufacturing paradigm that combines most of the development from previous models and attempts to solve most of their drawbacks, attracting experts, scholars, and enterprises. Cloud Manufacturing is promising in transforming today's manufacturing industry towards service-oriented, highly collaborative, and innovative Manufacturing in the future [10]. Cloud Manufacturing is the result of adopting key enabling technologies (such

	Flexible manufacturing	Distributed (Network) manufacturing	Cloud manufacturing
System functions	Cooperation	Resource sharing/cooperation	Resource sharing/resource efficiency/cooperation
System openness	Many constraints, poor openness	Better openness	Highly open
Resource type	Organization, human, technology	Equipment, people, materials, network, information	Materials, equipment, software, hardware, logistics, human, knowledge
Resource usage	Customization	Dynamic configuration	On-demand dynamic configuration
Collaboration scope	Several companies	Companies in several industries	Companies in almost every industry

Table 3.3: Comparison of characteristics of three advanced manufacturing models, author's elaboration from [16]

as Industrial Internet of Things, Cloud Computing, Digital Twins, Big Data) by manufacturing enterprises to share resources and capabilities to enhance their response to market requirements and increase cost effectiveness[15]. The advantages of Cloud Manufacturing make it a new field of research.

In conclusion, the analysis of the state-of-the-art has highlighted three key trends in the evolution of manufacturing systems: (i) reconfigurability; (ii) lowering complexity; (iii) increase the need for autonomy. In addition, from the latest Smart Manufacturing techniques that mimic human-like capabilities, four interesting key factors are commonly presented in manufacturing systems:

- self-configuration: from low level (machine) to high level (plant), the system needs to be able to drastically adapt and change

- self-optimization: automated optimization methods to increase overall utilization

- self-protection: being able to anticipate possible threats and provide counteractions for the short and long term

3.3 Towards a common definition

The concept of Cloud Manufacturing was first proposed by Li Bo-Hu in China [9] and it is defined as a new networked manufacturing model that is able to

solve more complex manufacturing problems and perform larger- scale collaborative manufacturing through the introduction of key enabling technologies (such as cloud computing, cloud security, high-performance computing, Internet of things) in a new service-oriented model. In this model, scattered online manufacturing resources are structured in a platform where users can access eligible manufacturing services. While Cloud Manufacturing is a relatively new concept, a variety of definitions are present in the literature from scholars that have modified and enhanced it; a selection is listed below:

- "A customer-centric manufacturing model that exploits on-demand access to a shared collection of diversified and distributed manufacturing resources to form temporary, reconfigurable production lines which enhance efficiency, reduce product lifecycle costs and allow for optimal resource loading in response to variable- demand customer-generated tasking"[17].

- "A model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable manufacturing resources (e.g., manufacturing software tools, manufacturing equipment, and manufacturing capabilities) that can be rapidly provisioned and released with minimal management effort or service provider interactions"[3].

- "A new-generation service-oriented approach to supporting multiple companies to deploy and manage services for manufacturing operations over the Internet"[18].

- "A new networked manufacturing model which aims at achieving low- cost resource sharing and efficient coordination. It transforms all kinds of manufacturing, simulation, and computing resources and abilities into manufacturing services to form a huge manufacturing cloud and distributes them to users on-demand"[2].

Xu [3] expanded the original scope of "online manufacturing resources" from Li Bo-Hu [9] including manufacturing capabilities along with manufacturing resources. In order to access such manufacturing resources, [19] and [11] emphasized the importance of key enabling technologies in the definition of Cloud Manufacturing from ICT (such as Machine Learning, Big Data, 5G) and manufacturing technologies (such as Additive Manufacturing, Intelligent Robots, and Intelligent Manufacturing techniques). From an organizational point of view, an interesting addition to the Cloud Manufacturing definition is brought by Wu [17] where on-demand services are seen as a trigger to create instant, reconfigurable networks to respond to complex and variable task requirements from the market. Another important addition that widens the definition of Cloud Manufacturing comes from the work of Fisher [20] where the authors, after a detailed comparison of Cloud Manufacturing key characteristics and a deep analysis of the future of manufacturing systems, identify Cloud Manufacturing as a route to Sustainable Manufacturing. Finally, Tao [19] clarified the origin of Cloud Manufacturing. While this is a new service-oriented model, Cloud Manufacturing is an evolution from existing advanced manufacturing models presented in

the previous paragraphs (such as agile manufacturing, networked manufacturing, manufacturing grid). In other words, other research on this topic exists but presents slightly different viewpoints. Cloud Manufacturing can promote collaborative design techniques by sharing design information. Cloud Manufacturing, if correctly implemented,can also enhance resource sharing, rapid production of prototypes, and reduce costs. Distributed manufacturing can be developed as a result, although resource autonomy and system governance have not been addressed. Cloud Manufacturing can potentially reduce time-to-market, improve service, and enhance user experience, which advantageously impacts customer co-creation area [21]. While Adamson et al. [22] outlined that Cloud Manufacturing is not always a feasible solution for enterprises, mainly due to lack of competencies for its implementation, Wu et al. [23] identified the key economic benefits required for a comparative study that supports organizations in determining when traditional in-house design and manufacturing versus CBDM is most appropriate. The study explored key factors of a cost-benefit analysis through a cost breakdown and a price comparison with cloud computing pricing plans on different levels (e.g., IaaS, Paas, SaaS). Wu et al. [21], in another study, showed three sectors that could be affected by cloud manufacturing on long and short terms: (i) the engineering and design sector; (ii) the manufacturing sector; and (iii) the marketing and service sector. Explicitly, In the short term, Cloud Manufacturing can offer ubiquitous access to design information, improve efficiency, adequate computing resources for the engineering and design sector, thus producing a collaborative design approach in the long term.

In the manufacturing sector, the Cloud Manufacturing environment can potentially improve resource sharing, rapid prototyping, and reduction in costs, hence improving distributed manufacturing in the long term. As for the marketing and service sector, time to market can be reduced, service quality can be improved, and customer needs elicitation can potentially be enhanced. Consequently, cloud manufacturing can possibly provide a customer co-creation environment[24].

Throughout these insights, cloud manufacturing would thus play a significant role in the development and execution of product lifecycle processes, as in cloud manufacturing; product life cycle activities and functions can be supported by virtualized manufacturing resources and the manufacturing capabilities layer allocated within the cloud manufacturing system. Thus, this can allow more users to access these services, delegating the manufacturing enterprises (service provider) to carry out all activities (processes) involved in the entire life cycle of the product and to focus only on their core business and services [19].

3.4 Cloud Manufacturing Architectures

The architecture of Cloud Manufacturing is the system design planning for Cloud Manufacturing implementation and the basis for the development and application of a Cloud Manufacturing actual system; the supporting technologies of Cloud Manufacturing are the foundation for realizing the Cloud Manufacturing architectures and supporting the completion of Cloud Manufacturing

business; the phased application status analysis of the Cloud Manufacturing is the reference for finding the problems and deficiencies in the development of Cloud Manufacturing. Therefore, an effective exploration of the current research status in terms of architecture, supporting technologies, application status of Cloud Manufacturing plays a vital role in the innovation of its theory, technology, and application development [25]. Various models are used to describe the architecture of a Cloud Manufacturing platform. The most commonly used is based on a multi- layered architecture with a modular approach from He and Xu [26], where each layer presents a specific role that accomplishes the required functions. In this paragraph, a variety of Cloud Manufacturing architectures are depicted to embrace the similarities and contrast between them and further to be a baseline for the development of this research.

Ding [27] proposed a layered framework of collaborative manufacturing resources shared based on cloud services. The study designs an architecture with three main layers: (i) Cloud service demand layer; (ii) Cloud service center; (iii) Cloud service provider layer. Each layer is composed of more specific sub-layers. The cloud service demand layer is based on the Cloud user interface. Layer (i) is connected to layer (ii) through an application interface. Layer (ii) provides a variety of core services and function and is divided into two sub-layers: (a) Cloud service management, responsible for user management, task management (publication, aggregation, scheduling), service search; (b) Cloud service integration, provides integration and semantic interoperability of a wide range of manufacturing resources through a global and a local service integration model. A Cloud access interface is a gateway that allows multiple manufacturing resources from the Cloud service provider layer (iii) to work with the layer (ii).
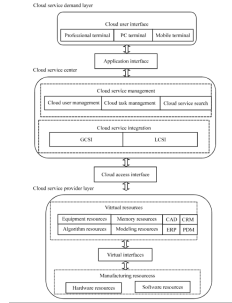
Figure 3.1: Ding [27] three-layer architecture for CMfg

Moreover, Jiang [28] introduced a five-layered structure based on collaborative agents (CAgents) with the following layers: (a) basement layer (b)access layer (c)functional layer (d)portal layer (e) application layer. The functional layer is responsible for controlling and coordinating the various service transactions within the cloud manufacturing system.

Wang [29] expands the role of the Master Cloud Agent within the smart cloud manager layer to analyze, optimize and control the Cloud Manufacturing
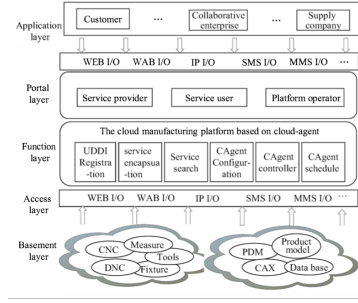
Figure 3.2: Jiang [28] cloud manufacturing integrated service platform based on CAgent

service interactions between the user layer and the manufacturing capability.
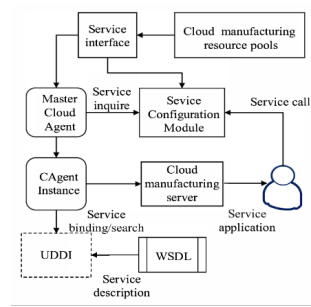
Figure 3.3: Wang [29] - The integrated manufacturing service mode based on cloud agents

Lv [30] proposed another typical four-layered hierarchy architecture. This architecture offers a more detailed mapping of resource entities into cloud services from physical resource layer to virtual resource layer, which highlights the core idea of an open cloud service architecture. The architecture is based on a multi-view model that integrates different views (function view, resource view, information view, and process view), with each view depicting a different aspect of the platform.

The function view lists the various tasks that a system can perform and comprises interlinked activities. The resource view enumerates the resources required to perform activities. The information view focuses on the required data for the activities, and the process view captures the sequence of the activities.

Moreover, a novel approach that is not mainly focused on technical aspects of the Cloud Manufacturing system comes from Škulj [31] that proposed a decentralized perspective for a cloud manufacturing architecture (CMdna) shown in Figure 8. One of the main contributions of this work derives from the introduction of the concept of a cloud manager component (layer) with the aim of

26
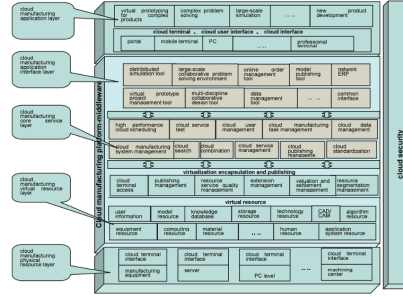
Figure 3.4: Lv [30] Cloud Manufacturing Architecture
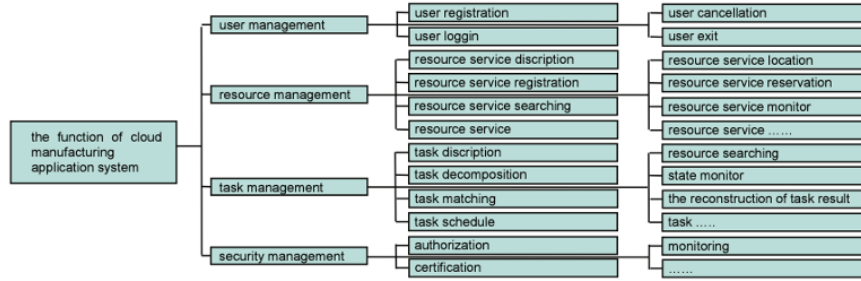
Figure 3.5: Lv [30] Cloud Manufacturing System function tree

creating a flexible connection between cloud service providers and service users through the utilization of autonomous work systems (AWS) that acts as numerous manufacturing clouds which vary depending on the requirements of both service users and service providers. Such an architecture would allow several clouds to bid for each stage of the required work to make the process as cheap as possible for the end-user.

Based on the proposed architectures and considering the similarities of the models presented in the literature, a novel architecture is proposed on Chapter III to overcome issues not tackled by the typical configuration of the cloud manufacturing systems as depicted on Chapter II Section 5.

/sectionCloud Manufacturing Service Managemen Services Management within Cloud Manufacturing is considered a critical issue. Indeed, an important goal of Cloud Manufacturing is to provide users with on-demand services for the manufacturing resources and capabilities they need. After these distributed and heterogeneous resources are virtualized, modelled, and transformed into services on the cloud, there is a solid need to effectively manage and coordinate these services in a centralized way to ensure the service performance, quality, security, and successful operation of manufacturing clouds [26]. Resources can interact into a public cloud or a private cloud based on the difference in service object [11]. In order to ensure service performance of Cloud Manufacturing,
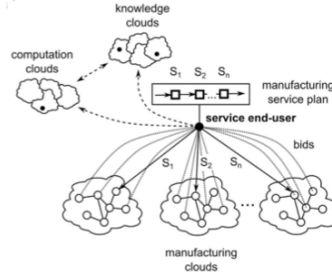
Figure 3.6: Škulj [31] - Decentralized Cloud Manufacturing Network

various methods have been proposed. Wang [1] developed a system based on an ontology of virtualized manufacturing resources. Liu [33] proposed three multi-agent systems architectures for different enterprise sizes. The three architecture are the following and mainly diversified by the role of the Master Agent:

- (a) the Facilitator Architecture: The facilitator is a special agent responsible for coordinating the communication among the agents. The facilitator provides a reliable communication layer, routes messages among agents based on the contents of the messages, and coordinates the control of the multi-agent activities. All the agents in a facilitator-centric architecture communicate with each other via the facilitator. As a result, the robustness of this architecture can be poor, and the overhead is relatively high.

- (b) The Mediator Architecture: As the facilitator, the mediator is a special agent with more functions than the facilitator. Besides coordinating the communication among the agents and the control of the multi-agent activities, the mediator is able to search for relevant agents according to the agents' requirements and assist in setting up communication among them. All the agents in a mediator-centric architecture communicate with each other through the mediator. However, the agents can also communicate with each other after the communication has been set up (indicated as dotted lines). In contrast to the facilitator-centric architecture, the overhead of the mediator-centric Multi-Agent System is reduced.

- (c) The Autonomous Agent Architecture: As the facilitator, the mediator is a special agent with more functions than the facilitator. Besides coordinating the communication among the agents and the control of the multi-agent activities, the mediator is able to search for relevant agents according to the agents' requirements and assist in setting up communication among them. All the agents in a mediator-centric architecture communicate with each other through the mediator. However, these agents can also communicate with each other after the communication has been set up (indicated as dotted lines). In contrast to the facilitator-centric architecture, the overhead of the mediator- centric MAS is reduced.
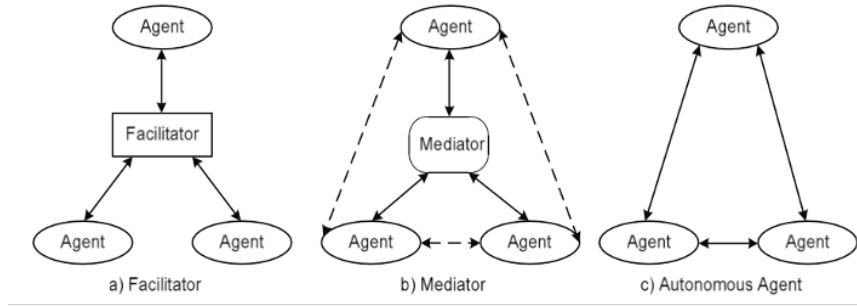
Figure 3.7: MAS Architecture proposed by Liu [33]

Several studies have examined service quality and composition in a Cloud Manufacturing platform. Lin [34] introduced an Ontology inference cloud service (OICS). An OICS is a knowledge-based cloud manufacturing system and is used to recommend machine tools and cutting tools based on the Ontology inference techniques for cloud services. The OICS comprises three core functional modules: The Ontology inference module, the VMT (Virtual Machine Tool) module, and the request filtering module. Modules are developed to allow multiple users to perform inference service and verify the recommended machine tools or cutting tools via VMT simulations. The proposed system provides the optimal number of machine tools for the acquired system based on the designed ontology data of the system and thus aims to improve the quality of the cloud manufacturing services.

Finally, Lu proposed a knowledge-based service composition and adaptive resource planning model in a cloud manufacturing environment in order to develop an integrated networked environment enabling the optimal allocation of resources based on given criteria. The model is deployed as a web service and is based on three critical stages: (a) collaborative business process modelling and verification of cloud workflow; (b) model instantiation with modelling and clustering of manufacturing services; (c) model execution, with the optimal matching of manufacturing service supplies and requirements.

3.5 Research Gap Analysis

Cloud Manufacturing can potentially present a strong impact on manufacturing systems. However, further investigation is still required to identify the communication and interaction protocols of the cooperative systems that enable the integration of service providers and service users. The most important gap identified by the author, however, is not in the constituent parts of the cloud (as many cyber physically enabled smart manufacturing components already exist), the protocols (as plenty of excellent work has been done in this area already), or the integration (as the researchers have proposed several approaches likely to succeed). Architecture designs that are presented in literature reflect the cognition and expectation of different researchers. While most architectures

found in the literature are characterized by functional views and resource-based views, articulated in a multi-layer structure, almost none presents a process and organizational view. While most architectures assume direct access and control of the scattered physical resources, only Škulj [31] proposes an architecture based on Autonomous Work Systems. Finally, while Cloud Manufacturing works presents multiple efforts on service optimization, almost none deals with the negotiation of service allocation with service providers. Services created by aggregating autonomous service providers represent a step forward in an architecture that fits actual enterprise characteristics (especially Small and Medium Enterprises) and better applicability in real-world cases.

The author believes that the main research gap in Cloud Manufacturing architectures is in the characteristics of Service Providers. The presence of autonomous and platform-independent manufacturing resources brings numerous issues derived from a distributed governance. Additionally, the literature shows that other gaps in Cloud Manufacturing research are present. Other research gaps identified include:

1. A lack of research directed towards the platform implementation: most scholars have concentrated only on Cloud Manufacturing architecture and its enabling technologies: there is a need to examine Cloud Manufacturing with real case studies to demonstrate the usability and successful implementation in a real-life context.

2. A lack of research work from the managerial point of view in cloud manufacturing: there are many studies regarding the technical issues around Cloud Manufacturing in the literature. These studies have typically overlooked how to manage cloud manufacturing from a management point of view. Issues that need to be addressed include stakeholders' interactions and their activities, the cloud's standards, services management, utility models, servitization technologies, and the role of clear and shared founding principles in a Cloud Manufacturing platform.

3. A lack of research regarding how to manage negotiation in cloud manufacturing: the literature reveals that there is not yet an understanding of negotiation mechanisms for cloud manufacturing. There is a need to identify, assess, and control interactions among service demanders and service providers inside the network.

Therefore, this research proposes an architecture of a distributed Cloud Manufacturing network comprised of autonomous service providers to manage operations and coordinate communications among manufacturing nodes and service providers. The aim is to offer new insights for industry and academia on how to deal with autonomous service providers at the adoption and implementation stages of the platform.

Chapter 4

Development of an Architectural Framework for a Distributed Cloud Manufacturing Platform in presence of autonomous nodes

4.1 Introduction

This chapter presents the theoretical framework of this research in the form of a typical cloud manufacturing platform to investigate and explore cloud- resource sharing and execution of manufacturing process plans for heterogeneous decentralized autonomous manufacturing resources. The limitations identified in the previous chapter were used to develop a set of requirements and founding principles for the manufacturing systems.

Due to increasing globalization, manufacturing activities often require complex dynamics. Consequent design activities of manufacturing networks are useful in order to guarantee suitable decisions that could endure competition among companies. For this reason, product and process varieties are key factors to address customers' need for personalization, as well as strategies for companies. Phenomena connected to customers determine new factors that represent a challenge for industries, always ready to perform various manufacturing tasks within mass customization contexts. Indeed, modern manufacturing networks consist of suppliers that obey a unique principle: delivering products to the final customers belonging to the market. In such a context, smart technologies are essential in

order to develop not- coupled and not-hierarchical heterogeneous systems, with the aim of satisfying constraints that, following needs of customizable products, market trends, and social media, allows directing expectations and desires, as well as demands of customers. In this direction, nowadays, an increasing necessity of personalized products is a growing necessity of international markets. This effort is the result of emerging mass customization that requires a fast and safe reconfiguration of various systems, especially of manufacturing type, as well as a competition that implies rapid changes in the customized production style. Over the last few years, there has been a remarkable growth in the research activities related to the Industry 4.0 paradigm [35]. The term collectively refers to a wide range of technological concepts that provide solutions and advancement to different needs of manufacturing systems. Many smart manufacturing concepts and architecture have been proposed to bring higher flexibility with enhanced productivity, customization, and shortening the time to market. Combining emerging technologies with advanced manufacturing models, Cloud Manufacturing is a new manufacturing paradigm that meets the needs of manufacturing systems [36].

In these models, resources are converted to independent and cooperative subsystems. These elements, connected to the physical environment through smart sensors, can work in Virtual Manufacturing Systems (VMS). A VMS is the aggregation and mapping of distributed physical elements. Each element may range across different levels of aggregation in the manufacturing processes from machine-level up to a whole production or logistics network.

Collectively seen, such new advances generate innovative technological possibilities potentially suitable to satisfy sophisticated customer demands, expectations, and desires. Building innovative models around the notion of being "globally virtual, locally physical" calls for a service-dominant logic of distributed resources in which reusable services models that represent homogeneous production processes are shaped according to the concept of Manufacturing as a Service [8].

A modern manufacturing network is composed of cooperating plants, suppliers, and dealers that produce and deliver final products to the market [37]. These systems are no longer hierarchical physical and logical capsulated systems but heterogeneous, loosely coupled, non-hierarchical structured, cyber-physical systems of systems with event-based communication, collaboration in unified networks [38]. The idea of non-hierarchical production networks consisting of autonomous enterprises has been present in the scientific community for more than 20 years. Although current models, especially in large enterprises, are organizationally centralized due to size, need for control, and lack of third-party trust. It seems that this idea waited for production systems to acquire proper information and communications technology (ICT) or new industrial platforms, like Industry 4.0 [39]. However, a strong effort towards Industry 4.0 is due to phenomena connected to Big-Data, also considering suitable ways by which industrial operations collect, manage and then interpret their own information [4]. This phenomenon is an obvious consequence of dynamics dealing with smart manufacturing systems, as they combine, mix and aggregate heterogeneous in-

formation sources located in different layers and/or domains. The possibility of achieving new status of associations, as well as finding patterns, is important within the context of Manufacturing for the reasons described as follows:

- Criteria generation to construct decision systems for supply networks and manufacturing activities.

- Data continuous monitoring of fluctuations, with consequent predictions of future streams and their optimization.

Remarks and/or details about Big Data, as well as the analysis of datasets, are a challenge of each system within the Industry 4.0 environment. In this sense, conventional statistical processing approaches are not often useful because of the complexity and the size of large datasets. Actual implementations deal with adaptive scheduling, as well as a real-time modelling of processes and Decision Support Systems, useful to refine processes and design of components [5]. As for the context of logistics and production, a possible optimization foresees growth of resource efficiency [6]. Such a situation allows the creation of different production technologies, with the consequent birth of new manufacturing models. The challenge with distributed production is to implement communication and integration technologies able to reduce the coordination effort and provide a focused factory [7].

4.2   Main Assumptions and Founding Principles

In the following scenario, the Cloud Manufacturing Network consists of nodes that utilize homogeneous technologies. These nodes are able to work in one or multiple distributed networks in an interoperable way. Each node inside the network can instance orders or buy production resources (slots) under the supervision of a coordinator that manages the negotiation and communication protocols among nodes.
Each network that works inside the platform follows three main principles: sustainability, transparency, and shared resources. In particular:

- Sustainability deals with either reducing resource demands or CO2 emissions over the entire product life cycle that transfers the production closer to the client and with cost-effective manufacturing. In general, sustainability has not a definition of its own. However, there is consensus toward a research of compromises among resource and service needs, intending to guarantee either the satisfaction of users or the health of ecosystems that allows obtaining the resources.

- Transparency according to the Circular Economy trend, these networks need to create a transparent, collaborative, open, and trusting environment with shared purposes and resources.
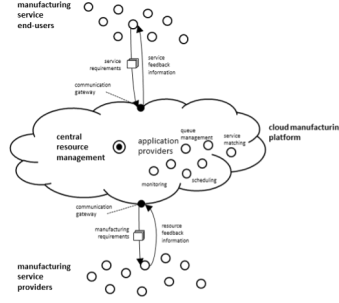
Figure 4.1: Centralized Cloud Manufacturing Architecture

- Shared resources refer to the possibility, for each node, to have equal unrestricted access to all possible resources inside the platform. Indeed, nodes can access other nodes' resources through an open bidding system.

4.3 Framework Components

4.3.1 Designing the architecture

Architectures enable systems to operate and evolve, providing services inside an environment with a predictable level of quality, quantity, and performance. In operations management, architectures may present different definitions and scopes. Still, the core characteristic is concerned with providing a bridge between multiple system functionalities and requirements for defining the attributes that the system has to meet.

Distributed Manufacturing architectures have been thoroughly analyzed in academics and business fields. As a result, engineers have proposed different ways to reconfigure these systems with a common aim to expand functionalities and fulfill a broad range of requirements.

While there are still multiple definitions and architectures of Cloud Manufacturing, as depicted on Chapter II Section 2 and Section 3, a common objective is to connect end-users with a ubiquitous network domain of manufacturing service providers to enable co-creation [40]. The platform comprises multiple application layers responsible for service matching, manufacturing scheduling, optimization, and execution of the manufacturing process. Platform management is usually designed to be automated with centralized governance to provide efficient service coordination. In this case, the application layer can directly connect with a specific manufacturing provider obtaining information and making decisions through remote control of the specific resource.

A fully decentralized network architecture for cloud manufacturing (CMdna) has been proposed in Škulj 2016 [31]. In this work, the authors introduce a two layers architecture composed of an end-user layer and a service provider layer. This architecture presents fixed boundaries among layers and is based on
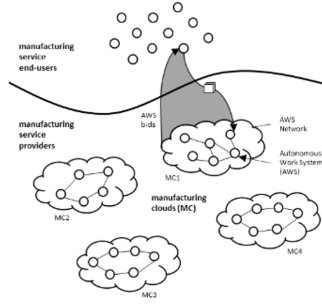
Figure 4.2: Decentralized Cloud Manufacturing Architecture

Autonomous Work System (AWS). Most platform's functionalities are decentralized to the AWS network distributing matching, scheduling, and decision-making mechanism among service providers.

4.3.2 Scattered Manufacturing Architecture

In a Scattered Manufacturing platform, resources, labeled as nodes, can be either a service demander or a service provider. The coordination and the resource allocation process inside the network require a multi-stage negotiation activity among nodes and a platform agent. A global coordinator, called System Orchestrator (SO), is responsible for keeping platform operations aligned with Scattered Manufacturing founding principles (sustainability, equally shared resources, transparency) over time. Keeping the platform in line with its objectives over time, as market and technologies evolve, is a fundamental characteristic to keep robustness and flexibility.

In an Scattered Manufacturing architecture, decentralization occurs through service instantiation. After receiving multiple orders and applying the localization, filtering, and clustering algorithm, the Orchestrator needs to fragment the order into a finite number of tasks assigned to a subset of network resource providers. Each sub-network is an autonomous Virtual Manufacturing System (VMS) where a platform agent, Service Manager,negotiates resources with candidate manufacturing nodes through an opening bidding system. Each node is fully independent and may sell its manufacturing capacities to multiples sub-networks simultaneously or use their capacity for themselves. For better clarity, VMS are labeled as Services.

4.3.3 Cloud Manufacturing Architectures: a comparison

Cloud Manufacturing architectures have similar advantages and disadvantages and reflect different needs from different physical systems. While Cloud Manufacturing's main strengths are efficiency and performance, it is also evident this platform can only be as flexible and robust as its centralized management.
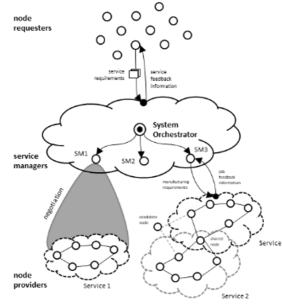
Figure 4.3: Scattered Manufacturing Architecture

Scattered Manufacturing and Cloud Manufacturing Distributed network architecture present more flexibility in adapting to context and environment through a negotiation process from an architectural perspective.

Another difference among architectures is in their scope and size. While Cloud Manufacturing should be more suitable for Large Manufacturing Companies due to their characteristics, Scattered Manufacturing should fit more for Small and Medium Enterprises and micro-manufacturing networks. A comparison of the main characteristics and requirements of the three architectures is shown in Table 4.

Finally, because of their structure, decentralized solutions may present drawbacks due to the higher degree of complexity and coordination needed:

- The platform must deal with the additional complexity and overheads from the granular nature of a distributed system based on autonomous resources.

- Current architectures are mainly oriented on building a monolithic Cloud Manufacturing platform with fully managed nodes to simplify the master planning and monitoring process.

- Monitoring Key Performance Indicators is a more complex process due to the need for monitoring globally distributed autonomous physical resources.

- Architects and engineers need to implement an intra-service formal negotiation mechanism and communication protocol.

- Communication protocols should also be able to support effective interactions among Service Agents and Node Managers.

- Implementing processes that span limited resources across multiple services without global coordination is challenging.

- The architecture requires careful coordination among services.

Characteristic	CMfg	CMdna	SMfg
No. of layers	users (consumers), application providers, physical resource providers	service users, service providers	system orchestrator, service managers, service nodes
Physical resource providers	Third-party or platform owned	AWSs	Associated autonomous nodes
End-users	External user	External user	Internal or External node/users
Resource Management	centralized within CM platform	decentralized to AWS level	decentralized to Node level
Service Matching	Application providers	End-Users	Service Agent
Resource allocation	Direct allocation	Asynchronously through propagation	Negotiation among a service manager and candidates nodes
Information flow	Vertical	Horizontal	Vertical and Horizontal
Scheduling	Centralized, developed by an application provider	Provided by the end-user	Developed by service managers coordinated by an Orchestrator
Optimization size	Global optimization	Local optimization, no virtual coordinator	Local optimization (SM)
Optimization scope	Short and medium term	Short-term	Global optimization (SO) Short-term (SM) Medium-long term (SO)

Table 4.1: A comparison among network architectures

- Deployment complexity. There is also a computational complexity of the software needed to deploy multiple agents and manage a system comprised of many different services simultaneously.

4.4   Building the platform model

This section defines the layout, agents, and functionalities for the Scattered Manufacturing platform. The three building blocks of the platform are (i) Architecture, (ii) Actors/Structure/Functions, (iii) Founding Principles. We can
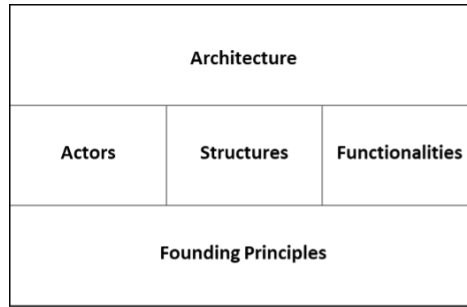
Figure 4.4: Platform Building Blocks

classify the main functionalities of this platform as follows:

1. Service Transactions Management: ordering, negotiation, contracting, delivering, payments.

2. Platform Operations Coordination: automated order processing, order decomposition, demand matching, load balance, job composition, task composition, platform, and service schedule.

3. Monitoring: concerning three different dimensions; (a) size (orders, job, task); (b) scope (global, service, local), (c) level of aggregation (i) Platform monitoring (ii) Service Monitoring (iii) Node Monitoring.

4. Evolution: adaptation of tactics and strategies used by the System Orchestrator, Service Coordinator, and Node Manager based on a continuous learning process.

The first two functionalities refer to the need for managing and operating a manufacturing and logistics network. A first assumption is that every node inside the network has been vetted with a preliminary registration process to parametrize different aspects of the process, such as orders generation, contracting, payment transactions, logistics, a messaging protocol for the open bidding system, and reporting and analytics protocol. Another assumption is that each order is composed of independent jobs that can be split and rearrange in tasks without logical dependencies. Other functional features inside the network are classified in Table 4.2:

Features	Description
Jobs Generation	Processing, filtering, and automated clustering of incoming orders based on selective and relevant features
Service instantiation	Instantiation of a new service to process and deliver a Job. A Service can be defined as a Virtual Manufacturing System (VMS). At the end of the negotiation process, the VMS negotiates capacity and prices and determines the schedule allocating a set of tasks to selected nodes. After concluding the job, the service and the related VMS terminates, and the nodes rearrange in new services bidding for new orders.
Service negotiation	Automated negotiation system based on software agents. representing the Service Coordinator, responsible for meeting jobs requirements and resource offerings, and the Node Managers, responsible for the machines scheduling and operations pursuing node objectives
Real-time scheduling and planning	Service Coordinators continuously send feedback about their jobs to the System Orchestrator. Based on the information received, the S.O. adapt the master planning, changing its strategy when instantiating new services

Table 4.2: Platform Functional Features

4.4.1 Platform Structure

Nodes inside the network can issue orders or sell production slots. An orchestrator determines the dynamics along with the network, managing communication activities among nodes via principles of equated shared resources, sustainability, and transparency. A unique approach is helpful to establish tradeoffs for the characterization of logistics and production costs in terms of resource allocation and show negotiation criteria among nodes. As for picking activities, the Author has proposed, on Chapter IV Section 3 Paragraph vi, a multi-stage algorithm that, once origins and destination are fixed, finds the route that permits to reach the destination in the shortest time.

Load balancing represents one of the key features to reach an equilibrium inside the network. Load Balance can be seen as the platform mechanism for self-regulation. Load Balance refers to the process of distributing a set of tasks over a set of resources to make their overall process more efficient. Load balancing can optimize the response time and avoid unevenly overloading some nodes while other resources are left idle. Different levels of load balance occur in the platform to reach an equilibrium in the overall network and in each service. A global load balancer should also implement a failover for those services that become non-responsive in allocating new jobs. This feature needs continuous monitoring through feedback communication systems from different levels of the network, such as Service Coordinators and Node Managers. In those cases, the System Orchestrator stops sending Jobs to that area of the network, instantiating new services in different regions or widening the size of the network that the service should query.

Another critical functionality to build a dynamic and stable open distributed network is the ability to instantiate new manufacturing networks when a service is needed. A Virtual Manufacturing System (VMS) is a key piece of a Scattered Manufacturing architecture. It can be defined as an ephemeral manufacturing system with variable dimensions and localization that dynamically changes its topology with time and scope. A VMS is deployed as a service each time the System Orchestrator needs to launch a new Job. Once the job is effectively delivered, the network and the relationships within its nodes are terminated. VMS introduces the ability of large-scale parallelism, and it is designed explicitly for a market-driven supply chain that requires carrying out lightweight network infrastructure and fast processing time in response to highly varying market needs. Indeed, the service can be seen as an operations function triggered when there is an actual market need. The networkless system can potentially become more adaptive, flexible, and efficient than traditional networks. This approach is location independent and, combined with fast, focused local area suppliers, could lead to better performance and scalability. Further, because every node inside the network can operate without the entry barrier of building and maintain a large supply chain infrastructure and working on its coordination, each node can focus more on the reliability and quality of the production.

4.4.2 Platform Functionalities

When the S.O. receives orders from multiple sources (external and internal associated nodes), it starts scanning the status of deployed services, and it assesses the overall platform load balance. Then it initiates the automated filtering and clustering algorithm that uses relevant features to decompose orders into jobs. After determining the size and localization of the service, the S.O. registers a new service deploying it inside the platform. Each service is initiated with Job characteristics and a first attempt of the network topology. This phase aims to reduce the manufacturing and logistics costs associated with a specific Job by searching and selecting candidate nodes. Once deployed, the S.O. leaves the service coordination to S.C.. Then, S.C. starts the multi-stage negotiation process with candidate nodes with a first attempt at balancing prices and workloads. At this stage, each node will deploy its strategies. In particular, it is worth mentioning that each node may not wish to subordinate their capacity to a specific service entirely, or they can probably also operate outside the platform. The presence of a service coordinator that needs to negotiate capacities and task prices with cooperative and competitive nodes ensures the balance is reached by the agents involved. Based on the response of the current negotiation iteration, S.C. has three possible actions to undertake:

1. In case of partial consensus, adapt the planning based on the feedback received from the nodes.

2. In case of total agreement, launch the task scheduler to initiate the service.

3. If the number of negotiation cycles is higher than a threshold iterations parameter, S.C. sends a denial of service message to the S.O.. In that case, the Job returns into the order pool and is managed by the S.O.. Based on the analysis of the states and actions previously taken, the System Orchestrator triggers a new deployment plan for the Job.
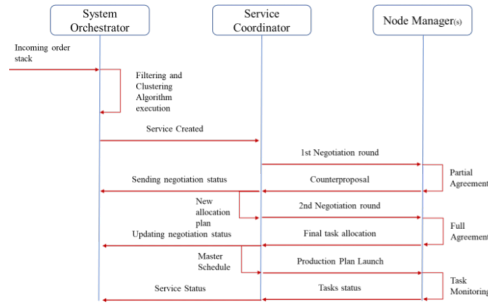

Figure 4.5: Operations mode of the SMfg platform

4.4.3 Platform Coordination and Negotiation Mechanisms

In such a distributed and opened system, coordination ensures that autonomous agents can act in a tightly coordinated manner to effectively reach their goals. This matter can be addressed, at least in part, by designing agents that communicate and cooperate through negotiation. The negotiation process is a sophisticated feature for introducing flexibility, efficiency, and achieving coordination in an open distributed manufacturing system.

Coordination mechanisms of actors involved should rely on decentralized governance to create an ecosystem-wide intelligence for adaptive control of platform operations. While centralized governance need of command-and- control poses potential issues in terms of the system's flexibility and scalability [41], decentralization of manufacturing system governance introduces structural complexity. The model requires to fully absorb the increased intricacies, variety of variables, and objectives of a modern manufacturing system. Therefore, a viable approach is the decomposition of decision-making tasks to improve the model's capability to understand and generalize complexity. In order to manage uncertainty and volatile dynamics, the model needs to introduce a certain degree of automation in decision-making and governance processes. Since it is impossible to model and rationalize each state and dynamic, advanced machine learning techniques are required. The model should be affected by the underlying system evolution and the decisions made by other autonomous elements, who are concurrently improving their policies through continuous learning. Continuous learning could be achieved with automated negotiation systems where software agents representing individuals or organizations are capable of reaching an agreement. This topic has seen a great deal of attention in the last decade from Multi-Agent Systems to Machine Learning and Artificial Intelligence researchers and represents a potential solution in simulating these systems [42].

Modeling mechanisms of coordination and dynamics in the behavior of each entity inside the network requires a good reasoning capacity about the long-term consequences of actions taken [43]. Configuring and managing strategies and tactics of each entity with an evolutionary approach is the main challenge for these systems. For example, as described in [44], a good job scheduler that should manage and interact within a cloud manufacturing sub-network should make decisions that are either reasonable for the immediate reward and good in the long term the sustainability of the network. Such an agent should sometimes forget short-term objectives in a shared effort of realizing greater long-term benefits. The scheduler agent should also adapt and react to variations in the underlying resource performance and scale in the presence of new or unseen workloads combined with large numbers of resources. Another fundamental requirement is model scalability and reconfigurability [41]. Indeed, the system should require a good generalization capacity, letting agents adapt to new environments, and the ability to decide in states of the environment that the model has not previously seen.

Chapter 5

Implementing Models and Algorithms for a Distributed Cloud Manufacturing Network based on autonomous resources

5.1   Introduction

This chapter outlines the implementation process of core functionalities in the Scattered Manufacturing framework, known as Scattered Manufacturing, by means of demonstrating the flow of the activities through a complete operations cycle. The first paragraph focuses on the implementation of a Multi-Agent System architecture for managing distributed operations. The second paragraph proposes an implementation of a scheduling and logistics optimization algorithm for a large Additive Manufacturing network.

5.2   A Multi-Agent System architecture for managing distributed operations

5.2.1   Introduction

The new generation of information technology dealing with cloud applications, big data, IoT has led to significant changes in manufacturing. The cloud appli-

cation service provided manufacturers with cloud-based software and collaboration by moving the processing and management of manufacturing information in the cloud and creating the phenomenon of Cloud Manufacturing [9][18]. Xu [3] defines Cloud Manufacturing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable manufacturing resources (e.g., manufacturing software tools, manufacturing equipment, and manufacturing capabilities) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Cloud Manufacturing aims at sharing and distributing in a collaborative manner large-scale manufacturing resources [45]. This is possible through a cloud manufacturing platform, which integrates distributed manufacturing resources, transforms them into manufacturing services, and manages them centrally [46] [3]. Cloud Manufacturing can handle multiple users' service requests, dealing with multiple manufacturing tasks (manufacturing lot) in parallel. Cloud Manufacturing can manage many distributed and idle manufacturing resources, providing a sustainable, cleaner production [47]. Anyway, there is no single standard for a Cloud Manufacturing implementation: there are several different Cloud Manufacturing architectures (e.g., see [9][45][48]). The shared resources in Cloud Manufacturing include the computing resource in cloud computing and other manufacturing resources. Such resources include hard manufacturing resources (e.g., machine tools), soft manufacturing resources (e.g., models and a massive amount of data), and manufacturing capabilities (design, production, and test capabilities). The on-demand supply method in cloud computing cannot be directly applied to cloud manufacturing because of some characteristics of manufacturing resources, such as heterogeneity, diversity, and dispersity, which cloud computing does not possess[26]. Hence, global scheduling is not always available [9]. In [10], a 3D printing service (3DPS) scheduling method in the context of Cloud Manufacturing is proposed to generate optimal service scheduling solutions; the method is based on a genetic algorithm. It is clear that one needs to select a suitable service because there may be multiple candidate services for a task. In [10], four attributes of the 3DPS, including size, material, accuracy, and cost, as the service matching rules, were considered in the scheduling problem. Anyway, in [10], the dynamic task arrival and downtime of 3D printers were not considered. Besides, the author did not consider anomalous tasks. In this section, the design of a Multi-Agent System for managing and monitoring 3DPS is proposed, addressing the issues above. Multi-Agent systems [49] represent a technology allowing modularity, flexibility, robustness, and adaptivity in complex systems, and they have been applied in many domains to solve complex problems [50][51][52]. Especially in industrial environments, where some requirements are needed depending on the application scenarios, the design is the first key factor to develop a suitable MAS [53].

In the following paragraphs, a Multi-Agent System scheme is proposed by analyzing it at the design stage. The analysis is supported by simulating some nodes through a small hardware system to check on communication issues.

5.2.2 Problem Formalization

In this paragraph, we briefly describe the problem and its context. Herein, we consider the Scattered Manufacturing Network [54], an adaptation of a Cloud Manufacturing network architecture described in the previous chapter. In a Scattered Manufacturing network, nodes are autonomous entities able to instance job orders or offer manufacturing services coordinated by an Orchestrator. The Orchestrator is responsible for the negotiation among nodes, ensuring the respect of network policy, and the overall optimization in the Supply Network. Scattered Manufacturing network policy obeys three main principles: sustainability, equally shared resources among nodes, and transparency.

Sustainability occurs in cost-effective manufacturing, reducing resource demands and related CO2 emissions over the entire product life cycle, transferring the production closer to the end-user. The Scattered Manufacturing network aims to create a collaborative, transparent, open,, and trusting environment with shared purposes and shared resources[54]. Cloud Manufacturing requires the interaction between three groups: the users, application providers, and physical resource providers [17]. In a Scattered Manufacturing network, actors are grouped and labeled as: Demanding nodes, Orchestrator, Manufacturing nodes. The Orchestrator coordinates resources and workloads matching orders from demanding nodes and local manufacturing available capacity.

At the first stage, demanding nodes submit their job orders with the required accuracies and admissible quantities, and cost ranges. The platform then localizes the order to define a subset of candidate manufacturing nodes. Potential resource providers are then filtered, considering technical constraints derived from job requirements.

Each service demanders have distinctive priorities in the optimization objective function [55]. A weight coefficient represents a priority ri according to the demander's latest product delivery time. Then we have a minimization problem, which is formulated as follows:

$$\min \frac{\sum_i r_i F_i}{\sum_i r_i} \tag{5.1}$$

where Fi is the product delivery time of a specific service demander Di, and it takes into account the start time of the task, the printing time, and logistics time.

The constraints are mostly inequality constraints, such as:

- model size, that is the maximum admissible size of the selected *kth* service S_k must not be smaller than the size of the 3DP model of task t_i

$$\min\left(u_i, v_i\right) \leq \min\left(U_k, V_k\right) \tag{5.2}$$

$$\max\left(u_i, v_i\right) \leq \max\left(U_k, V_k\right) \tag{5.3}$$

$$w_i \leq W_k \tag{5.4}$$

where u_i, v_i, w_i are the length, width, height of the 3D model associated with the task t_i, respectively, U_k, V_k, W_k are the maximum length, the maximum width, the maximum height of machine working area selected for S_k respectively:

- printing accuracy: the accuracy A_k of the selected 3DP service S_k should be smaller than the printing accuracy a_i of task t_i

$$A_k \leq a_i \tag{5.5}$$

- the cost: the acceptable maximum cost c_i of task t_i should be not higher than the practical task completion cost C_k with regard to the selected service S_k

$$c_i \leq C_k \tag{5.6}$$

and an equality constraint, that is:

- printing materials: since the printing material type M_k of the selected 3DP service S_k must be the same as the printing material type of $i_t h$ task m_i

$$m_i = M_k \tag{5.7}$$

The optimization problem can be solved using a genetic algorithm (GA) [55]. It is the case to point out that, after the localization and filtering stage, the Orchestrator needs to fragment the order into a finite number of tasks that will be assigned to the resource providers. The assignment phase requires negotiations and optimization steps to obtain an optimal solution. Further details about this topic, as well as numerical approaches, are discussed in the following paragraphs and have been detailed in [54] and [56]. In order to tackle some issues such as dynamic task arrival and downtime of 3D printers, as well as anomalous tasks, in the next paragraph, a Multi- Agent System scheme handling the optimization problem in a more general way is introduced.

5.2.3 The proposed Multi Agent System architecture

A Multi-Agent System is a system composed of interacting intelligent agents that are autonomous entities that can act and communicate with each other in a certain context, depending on the environment state [49].
For each agent a finite set A of actions are possible:

$$A = \{A_1, A_2, ..., A_n\} \tag{5.8}$$

Through actions, each agent interacts with the environment. As a consequence, the environment assumes a finite number of possible states:

$$X = \{X_1, X_2, ...\} \tag{5.9}$$

In the proposed model, the objects of monitoring are tasks, printers, scheduling, and the system's fitness. We consider a multi-agent system (MAS) model, with three types of agents: Task Agent (TA), Master Agent (MA), and Printer Agent (PA).

The task agent (TA) collects and processes tasks, then organizes them according to the user requirements and provider policy. The TA handles batches of n tasks as follows:

$$B = \{t_1, ..., t_n, r_1, ..., r_n, w_1, ..., w_n, o_1, ...o_n, a_1, ..., a_n, m_1, ..., m_n,$$
$$h_1, ..., h_n, c_1, ..., c_n, d_1, ..., d_n, \mu, \sigma\} \quad (5.10)$$

where:

- t_i, with i=1,...,n, are the tasks

- r_i, with i=1,...,n, the priority of the ith task

- w_i, with i=1,...,n, is the workload for the ith task

- o_i, with i=1,...,n, is the 3DP output size for the ith task

- a_i, with i=1,...,n, is the required accuracy for the ith task

- m_i, with i=1,...,n, is the demanded material for the ith task

- h_i, with i=1,...,n, hashes of tasks

- c_i, with i=1,...,n, the acceptable maximum cost for each task given the service demander

- d_i, with i=1,...,n, delivery location of ith task

- μ mean workload of all scheduled batches

- σ standard deviation of workload for each scheduled batch

The mean μ and standard deviation σ of the workloads are computed to compare the current workload to the ones of past tasks. This evaluation process allows checking if the workload of a task is below a certain threshold as follows:

$$|w_i - \mu| < \alpha * \sigma \quad (5.11)$$

where α is a tuning parameter to be determined. If the workload is over the threshold, tasks return to the service demander. This phase allows realizing a sort of global optimization to ensure a certain balance in the global network of printers to not overload a node or assign only small works to a given node.

The TA is also responsible for monitoring tasks by checking task features such as task size and task integrity to perform a local optimization. It is equipped with a classifier, e.g., an Artificial Neural Network (ANN) or a Functional Network [57] [58] (in case only small datasets are available for the learning), to detect

anomalously (not fitting to usual demand) tasks. Indeed, an anomalous task is a task that presents a set of features (e.g., quantity, accuracy, sizes) that have never been seen before. For this anomalous task, the classifier present in the TA agent will label the task as false. This false task will not be immediately rejected but sent back for human confirmation by an operator. As usual, the classifier works in two stages: an offline stage, which is the stage where the ANN learns the tasks from certain users; an online stage, where the training dataset is updated by adding new cases.

The training dataset contains a triplet of input attributes for the ith task, that is, workload w_i, output size o_i, required accuracy a_i, and a single desired output, which is a binary value, that is 0 or 1, representing the false or true task. During the online stage, each task detected as "false" is sent back to the user for additional confirmation. If the user confirms the task as a "true task", then it is added a new sampling pattern to the dataset.

Printer agents (PAs) monitor if a particular printer is under or overloaded. A PA records the downtime of the printer. Then, if the idle time is below a given threshold τ_0 it communicates to the master agent (MA) that the printer is overloaded and it needs less work to operate; if the idle time is above a threshold τ_v, then it communicates that it is underloaded and, in this case, the PA communicates its own cost for the task.

PAs are also in charge of checking task integrity before the execution. The task body is hashed, and this hash is then compared with the hash provided by MA. If the hashes are the same, the task is processed. If not, it means that the task was modified and in such a case the task is uploaded from MA again.

A master agent checks all basic system characteristics: it is responsible for generating times of starting task scheduling, as well as monitoring and supporting the genetic process of scheduling. When the schedule is ready, tasks are disposed to the printer units to be executed. During task execution, MA gathers the information from PA. Then it decides if the workload should be increased or decreased to obtain optimal printer utilization. This is measured by the assumed fitness function of the system. The fitness of the system depends on the printers' utilization. They may be idle or overloaded. If many printers are idle, then MA makes a decision about scheduling forcing, and dispatching a new portion of tasks. The decision is made on the basis of a social behavior model involving the PAs. We adopt a hybrid voting scheme.

If more than a threshold p of the PAs is reporting that less work is required, the batches are sent q% less frequently. If more than p of the PAs are reporting that more work is required and the total cost associated with such PAs is not higher than c, the batches are sent q% more frequently. The parameters p and c are set in a proper way. The actions of the agents may be described in pseudocode in what follows. The signature of each algorithm indicates the agent's name (e.g., TA: meansTask Agent:); followed by the name of the action with its parameters. Different agents may execute the same action but with different behavior. The basic behavior that emerges by the cooperation between the agents is the following: 1) the Task Agent (TA) checks the data received by the service demander (Listing 1). This input data represents the batch B of n tasks

in equation 5.10. If the information is correct, it sends the request to the Master Agent (MA) (Listing 2). The MA receives this batch B and asks for information to a set of PA regarding τ_0 and τ_v. Once received this information (PA sends the information using the action in Listing 4), MA starts the scheduling. The scheduling consists of creating a set of work queue Q_j, each containing a subset of the tasks of batch B, and assigning this queue to one of the PA. Therefore, using the scheduling results, MA will send a work queue Q_j, together with the hashes H_j of those tasks, to one of the identified PA_j (Listing 3) until all the tasks are assigned. In case one of the PA finds an anomaly or an error, it sends the task back to the MA (Listing 5). In this case, the MA proposes a new scheduling plan (Listing 6).

Algorithm 1 Task Agent checks the input data received by the service demander.

1: **procedure** TA: ACTION_CHECK(INPUT DATA)
2:     Receive data from service demander;
3:     **if** task is true **then return** call Action_Send(B)
4:     **else** send back to service demander;

Algorithm 2 Task agent sends the batch to the Master Agent.

1: **procedure** TA: ACTION_SEND(B)
2:     Create the batch B;
3:     **for** workload in B **do**
4:         **if** $workload > threshold$ **then return** remove task from B; send task to service demander;
5:     Send B to MA;

Algorithm 3 Master agent receives the information from PAs, perform the scheduling algorithm and sends the work queue Qj (containing the tasks assigned to PAj) and the hashes Hj of the tasks to each PAj identified by the scheduler.

1: **procedure** MA: ACTION_SEND(QJ,HJ)
2:     receive resource information from PAs;
3:     call Scheduling;
4:     send Q_j, H_j to PA_j;

5.2.4 Design analysis of the proposed MAS

Sequence diagrams at the design stage are a visualization tool to sketch inter-agent communications. Figure 15, Figure 16, and Figure 17 show the sequence diagrams referred to three different scenarios.

As shown in Figure 15, the process starts when TA sends the batch file B to

49

Algorithm 4 PA sends the information regarding τ_0 Agent MA.

1

---

   procedure PA: ACTION_CHECK(τ)

      if idle time τ ¡ τ_0 **then return** sent τ_0 to MA;

      else if idle time τ ¿ τ_v **then return** send τ_v to MA;

Algorithm 5 In case of anomalies or errors, PAi sends the task back to the Master Agent MA.

1: **procedure** PAI: ACTION_SEND(E)

2:    **if** $h_i^T A \neq h_i^P A$ **then**; **return** E=1;

3:    send E to MA;

MA. This behavior is implemented in Listing 1 and Listing 2. MA sends a flag R to the printer agents PA_i requesting information on resources (Listing 3). The printer agents PAi respond by sending the idle time τ and the costs C (Listing 4). According to this information, MA performs the scheduling by sending to PAi the work queue Qi and the hashes of tasks Hi. (Listing 3)

Figure 16 depicts the case when a new scheduling plan is generated after the printer agents send the parameter τ_u and the new costs C. In Figure 17, the case when a hash check failed on a PA. Then that PA sends a message error E to MA (Listing5), which sends again the work queue and the hashes of tasks (Listing 6).

As a first check on communication issues, an implementation on a small hardware system, in a certain sense motivated by the ideas inspiring the hardware-in-the-loop simulation, was conducted (e.g., see [59]). An Arduino UNO was used in order to simulate the exchange of information between the node MA and a PA node. The MA was simulated using Matlab, providing a suitable function to solve the optimization problem, which is handled by MA.

The PA was simulated using Arduino with an ATMEGA328 microcontroller linked to the PC through the serial port. Middleware in the form of a shared folder as a common workspace for Matlab and Arduino was provided. A Java code allowed the simulated nodes to create and to read .txt files. The file name contains the timestamp and the name of the node.

The process starts with a .txt file created by MA, simulated by Matlab. Arduino, which simulates PA, will elaborate the file's content by generating a new .txt file. The simulated MA will read the latter by checking it and, in the ab-

---

Algorithm 6 n case a node is not available, MA proposes a new scheduling plan.

1: **procedure** MA: ACTION_SEND($Q_j\prime, H_j\prime$)

2:    **if** $\frac{\sum_i \tau_v^i}{\tau_v} > p$ AND $\sum_i c^i < c$ **then**; **return** call Scheduling;

3:    send new vectors $Q_j\prime$, $H_j\prime$, to PA_j;

---

sence of reading errors, generating a new .txt file.

As a result, at the end of the simulation, we found a total delay equal to 0.1 s, mainly due to the elaboration, but not to the communication, with a 1% rate of reading errors.
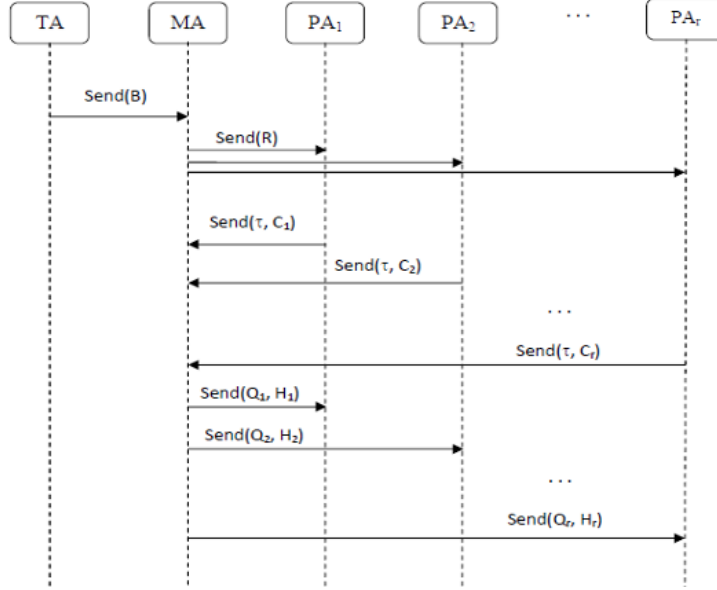
Figure 5.1: Sequence diagram, case 1 – starting the process (B = batch file; Q_i= work queue; H_i= hashes of tasks; R= flag for requesting information on resources)

Further work was conducted to outline a basic prototype of a Multi-Agent System for the Cloud Manufacturing network. A modular framework for building, analyzing, and visualizing agent-based models called MESA that uses the Python language has been selected [60]. More details about the implementation are in Appendix B: A MAS prototype of the CMfg platform.

5.3 An implementation of a Scattered Manufacturing Framework for large additive manufacturing networks

5.3.1 Introduction

In this section, an implementation of a Scattered Manufacturing network for large additive manufacturing scenario is presented. In the proposed scenario, the Scattered Manufacturing Network is constituted of Advanced Manufacturing Technologies nodes that can request or provide production resources (slots)
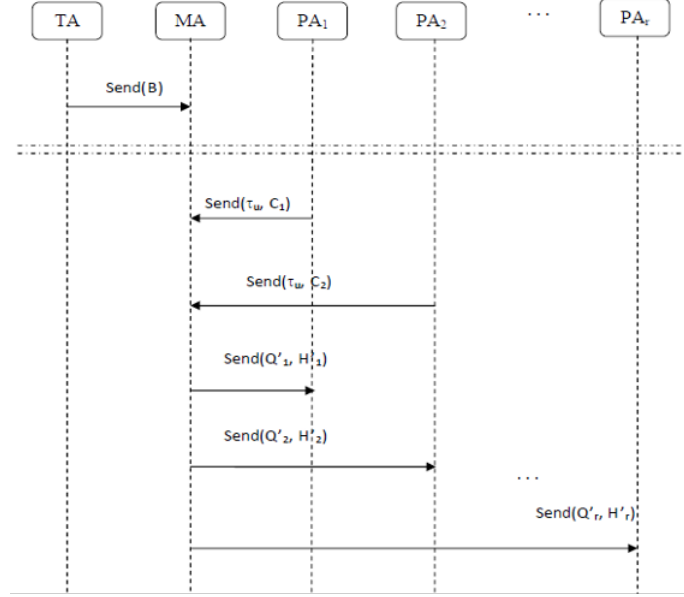
Figure 5.2: Sequence diagram, case 2 – new scheduling, according to the hybrid voting scheme (B = batch file; Q_i= work queue; H_i= hashes of tasks; τ_u= underloading parameter; C2= cost for the printer 2; $Q\prime_i$= new work queue ; $H\prime_i$= new hashes of tasks)

coordinated by an orchestrator, responsible for the communication along with the network, the negotiation among nodes, and the overall (production and logistics) optimization along with the supply chain structure.

In this scenario, nodes are either "productive" or "demanding". Productive nodes provide finished pieces realized by 3D printings. Demanding nodes, instead, orders finished pieces from the productive nodes. In this sense, demanding nodes formulate work orders that are satisfied by the productive ones. Considering complex dynamics inside large networks, one node can often be either productive or demanding in different times or situations. This context suggests that the communication and negotiation activities among nodes, managed by the orchestrator, are fundamental in order to share diverse resources and distribute them along with the network by satisfying principles of transparency and sustainability. In the following paragraphs, dynamics of networks with a unique demanding node and various productive ones are presented. In general, such assumption is not detrimental to the discussion of a general issue, but further details will be described in the last section.
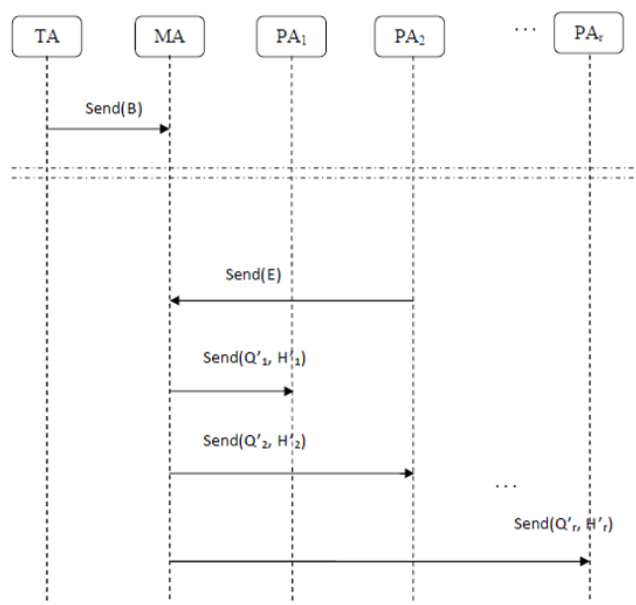
Figure 5.3: Sequence diagram, case 3 – Hash check failed on PA2 (B = batch file; Q_i= work queue ; H_i= hashes of tasks; τ_u= underloading parameter; C2= new cost for the printer 2; $Q\prime_i$= new work queue ; $H\prime_i$= new hashes of tasks; E=flag for failed check)

5.3.2 Founding Principles

The network observes three main principles: sustainability, shared resources, and transparency. Sustainability occurs in terms of cost-effective manufacturing, reductions of resource demands, and related CO2 emissions over the entire product life cycle, transferring the production closer to the client. According to the Circular Economy trend, the Scattered Manufacturing Network aims to create a collaborative, transparent, open, and trusting environment with shared purposes and shared resources. In fact, every node in the network can buy resources in the world with an open bidding system, while customers can send demands of products (orders) to the orchestrator. Hence, the orchestrator acts as an intermediate layer collecting orders from many customers.

5.3.3 Model description, requirements and network dynamics

To consider the variables and factors that affect a 3DPs network, a unique model is proposed by combining different approaches, which focus on such needs:

- Logistics issues related to the productive nodes that are near the demanding ones.

- Possibility of the division of demanding node's order into subparts and consequent assignment of each subpart to a productive node.

- Negotiation criteria between the demanding node and the productive ones in order to establish tradeoffs between different margin strategies.

Hence, the orchestrator has a primary role as it behaves like a control unit that applies a multilevel optimization that deals with the following exigencies:

- Localization: starting from the demanding node's geographical position, the orchestrator provides some neighboring nodes that define a "certified" sub-network to satisfy the users' requests.

- Fragmentation and assignment: the orchestrator establishes how to divide the work order into subparts, each assigned to different productive nodes to achieve the lowest overall purchase cost. Notice that this phenomenon requires a suitable negotiation between the demanding node, which asks for a predefined number of pieces, and the productive nodes, that have their quantities and pricing plans.

- Picking: the orchestrator defines a closed path that starts from the demanding node and returns to it, touching all the productive nodes once. Such a path, useful to collect the number of pieces from all the productive nodes, is obtained via an approach (see [61],[62],[63],[64],[65]) that minimizes the logistics costs.

Based on the just described requirements, the orchestrator set a run of iterations. As for the first one, the orchestrator:

- Indicates suitable productive nodes near the demanding one and assigns them the number of pieces to produce by satisfying constraints dealing with quantity/price plans.

- Defines a picking path at minimum logistics cost.

- Computes the weights that each productive node has inside the network. Precisely, for each node, the corresponding weight represents a tradeoff among logistics components, possible quantities of produced pieces, as well as reallocation of quantities by excluding the productive node from the network in consideration. Such last operation is necessary to discriminate among different productive nodes that can be far from the demanding one (hence requiring high logistics costs) but in turn useful due to their advantageous quantity/price plans.

As for the second iteration, the orchestrator works as follows. First, the productive node, whose weight indicates the highest decrement of the overall logistics and production costs, is excluded from the network. Then, the orchestrator redefines either the picking path or new fragmentations/assignments to the remaining productive nodes. This last phenomenon triggers a consequent

negotiation phase between the demanding node and the productive one, and the result is a tradeoff between different profits. Finally, the orchestrator recalculates the new weights of the remaining productive nodes, and the next iteration works as the second one. Iterations continue until the computation of weights indicates that further decrements of costs are not possible, hence reaching an equilibrium state.
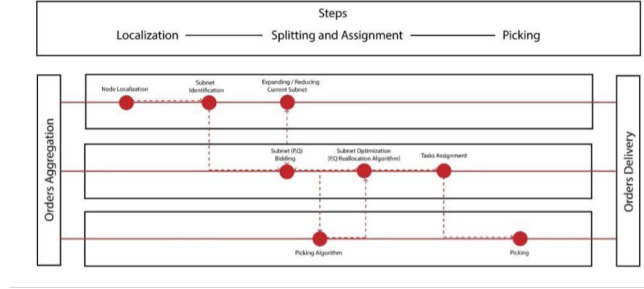
Figure 5.4: A visual representation of the steps involved in the proposed algorithm

Indeed, the originality and the contribution of the proposed approach foresees a complete balance among exigencies of different nodes. Starting from localization requests of the demanding node that needs a certified service network, a unique framework mixes approaches for picking paths and resource allocation problems that solve issues of fragmentation and assignment. Such aspects are dependent on each other, as they are strictly connected by the weights that the various productive nodes have inside the network. Indeed, the possible exclusion of productive nodes from the network determines a guideline to solve at the same time logistics issues, as well as reallocations by considering the overall quantity/price plans of each productive node. This last aspect, which clearly deals with the negotiation phases between the demanding node and the productive ones, represents the effective dynamics of the network at each iteration provided by the orchestrator.

The next paragraphs present a mathematical model followed by numerical examples. These examples show either expected features or unexpected ones. For instance, it is possible that the exclusion of a node during the iterations could provoke increments of logistics costs, as well as a suitable reduction of productive purchase. This situation implies the consequent need for new iterations. The tradeoff between logistics and production components indicates that the nodes to exclude do not obey a predefined and precise rule. Moreover, networks of medium dimensions might reach an equilibrium state in just one iteration. Such a phenomenon is important as it indicates that larger networks are sometimes easier to manage.

5.3.4 Modeling a 3DPs network

This paragraph briefly describes some features for a Scattered Manufacturing network within the context of 3DPs. The Scattered Manufacturing network is composed by a set of nodes $V = \{v_1, v_2, ..., v_n\}$ while the network is characterized by:

- e_{ij} is the arc that connects nodes v_i and v_j;

- c_{ij} is the cost for arc $e_i j$;

Notice that c_{ij} depends on various factors, such as the distance between v_i and v_j, the overall monetary cost for transports, the traveling time, as well as criteria of sustainability. Parameters are kept in a coefficient matrix:

$$X = \left(c_{ij}\right)_{i,j=1,...,N} \tag{5.12}$$

The SM network is assumed to be bidirectional, namely: two different nodes v_i and v_j are connected in the direction either "from v_i to v_j" or "from v_j to v_i". Obviously, e_{ij} - and e_{ji} - are the same arc while, in general, $c_{ij} \neq c_{ji}$. Each node provides services to the users in terms of finished pieces produced by 3DPs. Quantities Q_i and prices P_i of pieces for a generic node v_i obey a "law at three levels" of type:

$$P_i(Q_i) = \begin{cases} p_L^i & if 0 < Q_i \leq k_L^i \\ p_M^i & if k_L^i < Q_i \leq k_M^i \\ p_H^i & if k_M^i < Q_i \leq k_H^i \end{cases} \tag{5.13}$$

with $p_L^i < p_M^i, p_H^i$. The interpretation is the following: if the required quantity Q_i does not exceed k_L^i, the price P_i is the lowest p_L^i; otherwise, possible prices are p_M^i and p_H^i.
Notice that 5.13 represents a possible and realistic attempt to describer the evolution of pieces versus their possible prices. Indeed, future research activities aim at guaranteeing more suitable shapes for 5.13, with the aim of describing negotiation criteria among nodes.

5.3.5 Modeling a demanding node

This paragraph shortly describes a possible approach for optimizing demanding node's needs inside a 3DPs network. In particular, a unique model is described in which more approaches, often used individually, are used.
The demanding node aims to obtain a series of services from the Scattered Manufacturing network. In the specific case, in a preliminary phase, the orchestrator helps the demanding node referring to the following issues:

- *Localization*: the orchestrator makes the demanding node become the center of a circle with a radius of "economic" type. This means that, according to the demanding node's geographical position, only some production

nodes belonging to an area tracked by the orchestrator are able to offer services. Such a localization criterion has the advantage of defining the most known and neighboring nodes, thus creating a sort of "certified" sub-network. Therefore, the demanding node might have a higher level of trust. Each available node of the certified sub-network shows its offer in terms of prices/quantities plans.

- *Fragmentation and Assignment*: the orchestrator, considering the features of the sub-network, decides how to fragment the demanding node's work order and how to assign the various subparts to the production nodes in order to get the lowest purchase costs.

- *Picking*: the orchestrator chooses a closed path, which starts and returns to the demanding node through all the production nodes once. The path is defined via an approach described by the procedure shown as follows (see for details [61],[62],[63],[64],[65]).

5.3.6 The picking algorithm

For a Scattered Manufacturing network whose features are described in previous paragraphs, the following procedure is used for the picking activities. Assume that P is a possible closed path that crosses each node of V, starting from a source node $v_s \in V$ and coming back to it; $C(P)$ is the cost associated to P.

Algorithm 7 Picking algorithm (PA)

1: **procedure** INITIALIZATION
2:    $P := \emptyset, C(P) := \emptyset, v_s := v_j \in V.$
3: **procedure** STEPS
4:    1. From node v_j go to node $v_i \in V \setminus \{v_j\}$ such that $c_{ji} = \min \cup_{i=1, i \neq j}^{|V|} c_{ji}$
5:    2. $P \leftarrow P \cup e_{ji}, C(P) \leftarrow C(P)P + c_{ji}, V \leftarrow V \setminus \{v_j\}$
6:    3.
7:    **if** I **then**f $|V| = 1, P \leftarrow P \cup e_{ij}$ **return** end of the algorithm;
8:    **else**$v_j \leftarrow v_I$ and go to step 1

Considering a SM network with $V = \{v1, v2, v3, v4\}$ and matrix X:

$$X = \begin{pmatrix} 0 & 5 & 9 & 1 \\ 3 & 0 & 7 & 11 \\ 7 & 9 & 0 & 8 \\ 2 & 12 & 6 & 0 \end{pmatrix} \tag{5.14}$$

Preliminary, $P := \emptyset, C(P) := 0, v_s := v_1$. Table 6 shows the various iterations. Figure 19, where numbers indicate the nodes for simplicity, presents a graphical evolution of the path.

Iteration	P	C(P)	V
1	$\{e_{14}\}$	1	$\{2,3,4\}$
2	$\{e_{14}, e_{43}\}$	7	$\{2,3\}$
3	$\{e_{14}, e_{43}, e_{32}\}$	16	$\{2\}$
4	$\{e_{14}, e_{43}, e_{32}, e_{21}\}$	19	$\{2\}$

Table 5.1: Evolution of the iterations

5.3.7 Combining issues of Localization, Fragmentation, Assignment and Picking

Considering a demanding node that asks for services from a generic Scattered Manufacturing network, the aim is to decrease the overall cost. The cost function has two components, C_X and C_Y, that refer to the logistics (associated with the path) and the manufacturing costs, respectively. A suitable algorithm that considers all exigencies of the demanding role is presented in what follows. Consider a preliminary phase (iteration 0). The orchestrator, referring to a Scattered Manufacturing network with M nodes (Figure 20, up left), tracks an economic radius for the demanding node (in position T in Figure 20, upright), discriminates the unreliable nodes (in red, Figure 20, bottom left), defines a closed path P from T to T according to algorithm (PA), and computes the weights of each production node inside the network.

Notice that P has production nodes for which, respecting constraints of fragmentation and assignment, purchase costs occur. Hence, at the iteration 0, C_X and C_Y are as follows: $C_X^{(0)} = C(P)$ and $C_Y^{(0)} = \sum_{i=1}^{N} p_L^i$. For further iterations, the orchestrator works as follows in order to decrease the overall costs:

- Deleting nodes that can provoke the lowest costs in the next iteration.

- Reallocation of the sub-parts of the work orders (a new fragmentation and assignment phase), with consequent negotiation between the demanding node and the productive nodes. Notice that reallocation activities foresee a possible saturation of productive nodes.

- Computation of a new path for picking by using Listing 7: Picking algorithm (PA).

- Calculation of new weights (reallocation parameters) associated with the productive nodes.

Notice that removing nodes from the network implies an obvious natural variation of either logistic or purchase costs. In order to understand the entity of variations and compute the weights for the productive nodes, we define the following quantities:

- $\Delta C_X^{v_i}$, that represent the variation of the logistics costs when a node v_i is excluded from the network. Precisely, we have that $\Delta C_X^{v_i} = C(P) - C(P \setminus v_i)$.
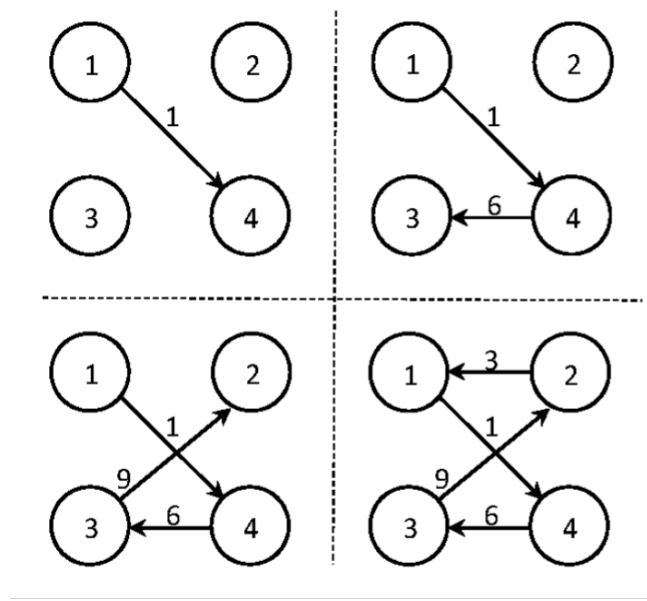
Figure 5.5: Graphical Evolution of the path
Iteration 1 (up, left): the first arc of the path P connects nodes v_1 and v_4. Iteration 2 (up, right): the path P connects nodes v_4 and v_3. Iteration 3 (bottom, left): the path P connects nodes $v3$ and $v2$. Iteration 4 (bottom, right): the path P connects nodes v_2 and v_1.

- $\Delta C_Y^{v_i}$ that indicates the dynamics of purchase costs when a node v_i is excluded from the network. In detail, we get that: $\Delta C_Y^{v_i} = -Q_i \left[P_i(Q_i) \right] + \sum_{j=1, i \neq j}^{|V|} Q\prime_j P_j(Qj + Q\prime_j)$, where $P_i(Q_i)$ follows 7 while $Q\prime_j$ is the amount of pieces redistributed on the network using the reallocation algorithm (RA), described above.

- $\Delta R^{v_i} = \Delta C_X^{v_i} + \Delta C_Y^{v_i}$ is the weight (reallocation parameter) associated to node v_i. Notice that, if $\Delta R^{v_i} < 0$, the exclusion of node v_i allows a decrement in the overall cost for the network.

5.3.8 Reallocation algorithm (RA)

Assume that L is the total amount of pieces, which the demanding nodes require from the network.
If node v_i is excluded from the network, Q_i is redistributed among nodes. The new quantity $Q\prime_j, j = 1, ..., |V|, j \neq i$, is defined as follows:

$$Q\prime_j = \left\lceil \frac{Q_i}{|V| - 1} \right\rceil \tag{5.15}$$
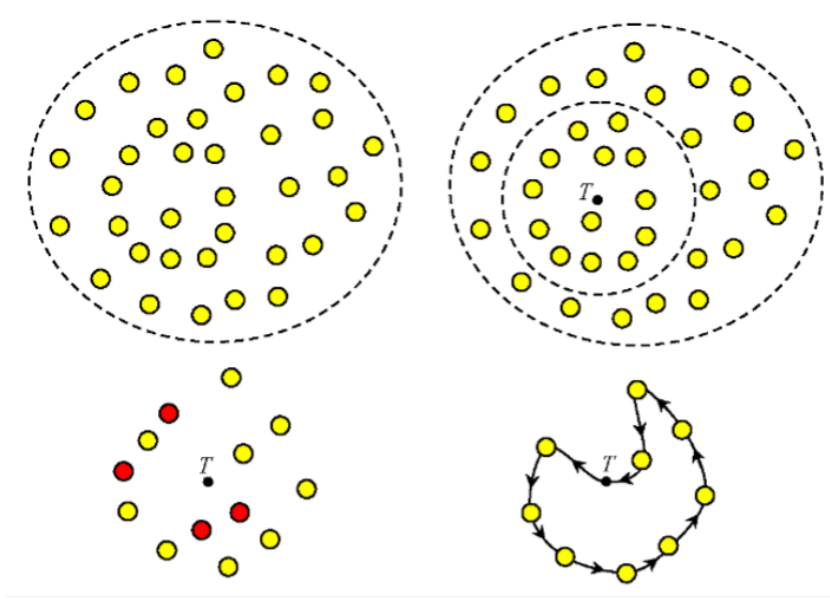
Figure 5.6: Preliminary phase (iteration 0)

such that:

$$\sum_{j=1,j\neq i}^{|V|} Q\prime_j = L \qquad (5.16)$$

Notice that Equation 5.15 has the following interpretation: as the productive nodes have all the same importance for the demanding node, the quantity Q_i is equally distributed among all the other remaining productive nodes. If $\frac{Q_i}{|V|-1}$ is not integer, then the whole upper part is taken.

Finally, the overall optimization algorithm, defined by the orchestrator's activities, works as follows, at the n-th iteration:

- Step 1: Erase the node j_n whose weights allows a reduction of the overall costs for the network in consideration.

- Step 2: Compute a new path for picking, with new cost $C_X^{(n)}$

- Step 3: Reallocate the quantities of pieces $Q_i, i = 1, i \neq j$ of each node according to the reallocation algorithm (RA).

- Step 4: Compute the new weights for productive nodes.

- Step 5: Come back to step 1 if there is at least one reallocation parameter ΔR^{v_i} is negative.

Figure 21 provides an intuitive idea of the optimization algorithm, considering the second, the third and then the n-th iteration.

60
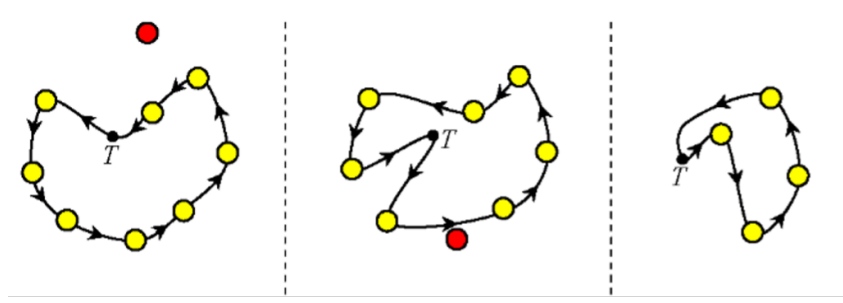
Figure 5.7: A visual representation of the optimization algorithm
Left: in the second iteration, a node (in red) is excluded, the picking path is recomputed, the logistics cost decreases while the overall purchase one can remain the same or decrease. Center: in the third iteration, another node is excluded, and the process continues. Right: in the n-th iteration, the overall logistics and purchase costs are highly decreased

5.3.9 Numerical examples of the proposed algorithm

The following example shows the numerous algorithm steps.

Example A

Consider a Scattered Manufacturing network with $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and matrix $X = (c_{ij})_{i,j=1,\ldots,6}$. A possible interpretation of the starting phase (iteration 0) is in Figure 22. The preliminary closed path P (Figure 22, up) involves the demanding node, v_i, and the productive nodes v_2, v_3, v_4, v_5 and v_6. For the productive nodes, the orchestrator determines purchase costs of p_L type (Figure 22, down), together with suitable assignments of finished pieces, as well as weights of each productive node inside the network.
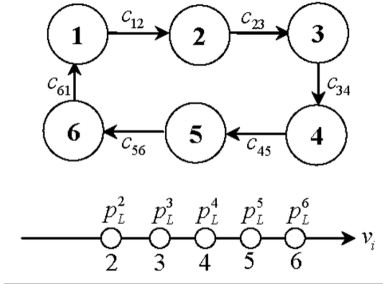
Figure 5.8: Example A, iteration 0

At iteration 1:

- Node v_3 is excluded because it has the lowest reallocation parameter.

- The closed path P is recomputed using algorithm (PA) considering the set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$.

- A cost $C_X^{(1)}$ is obtained

- A new fragmentation/assignment is made for the nodes of set V, see reallocation algorithm (RA).

- A cost $C_Y^{(1)}$ is obtained.

- New weights for productive nodes are computed.

Figure 23 sums up the iteration 1.
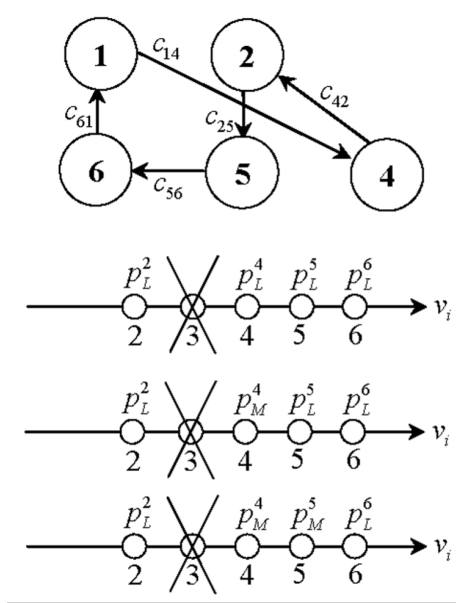
Figure 5.9: Example A, iteration 1

Precisely, Figure 23 (up) presents the new path, while Figure 23 (bottom) shows possible scenarios for $C_Y^{(1)}$:

- Scenario 1.1 (no variations): p_L^3 is erased and all other prices of p_L type remain the same.

- Scenario 1.2 (variations of just one cost): p_L^3 is erased, p_L^4 becomes p_M^4 and all other prices of p_L type remain the same.

- Scenario 1.3 (variations of more costs): p_L^3 is erased while, for instance, p_L^4 and p_L^5 become, respectively, p_M^4 and p_M^5.

Assuming that the scenario 1.2 occurs, at the iteration 2:

- Node v_2 is excluded due to its weight.

- The closed path P is obtained via algorithm (PA) for the new set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$.

- A cost $C_X^{(2)}$ is computed.

- A new fragmentation/assignment occurs for the nodes of V see reallocation algorithm (RA).

- A cost $C_Y^{(2)}$ is computed.

- New weights for productive nodes are established.

Figure 5.10 presents the iteration 2. Precisely, Figure 5.10 (up) considers
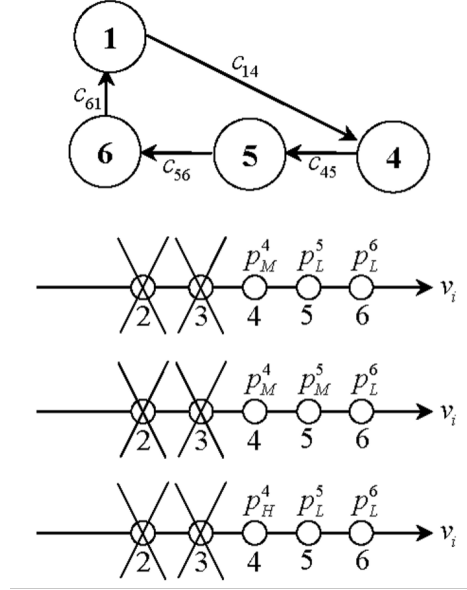
Figure 5.10: Example A, iteration 2

the new path, while Figure 5.10 (bottom) indicates possible new scenarios for $C_Y^{(2)}$:

- Scenario 2.1 (no variations): p_L^2 is erased while all other nodes have prices foreseen in scenario 1.2.

- Scenario 2.2 (variations of just one cost): p_L^2 is erased, p_L^5 becomes p_M^5 while all other nodes have prices described in the scenario 1.2.

- Scenario 2.3 (a high cost is achieved): p_L^2 is erased, p_M^4 becomes p_H^4 while all other nodes have prices described in scenario 1.2. In this case, the algorithm ends, as one term of p_H type is obtained, and all reallocation parameters become positive.

Assuming that the scenario 2.2 occurs, at the iteration 3:

- Node v_6 is excluded;

- A new closed path P is computed by picking algorithm (PA) for the new set $V = \{v_1, v_4, v_5\}$;

- A cost $C_X^{(3)}$ is obtained;

- A new fragmentation/assignment is made for the nodes of V, see reallocation algorithm (RA);

- A cost $C_Y^{(3)}$ is considered;

- Weights for productive nodes are updated.

Figure 5.11 presents the iteration 3.
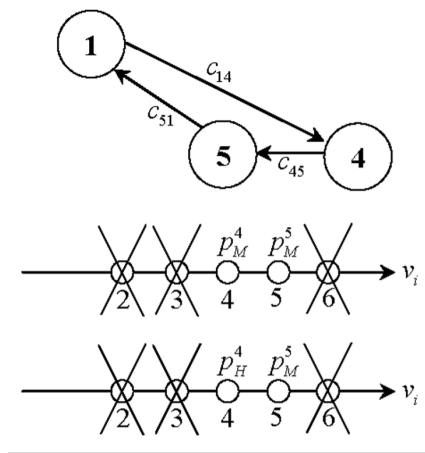
Figure 5.11: Example A, iteration 3

In particular, Figure 5.11 (up) shows the new path, while Figure 5.11 (bottom) presents possible various scenarios for $C_Y^{(3)}$:

- Scenario 3.1 (no variations): p_L^6 is erased while all other nodes have prices foreseen in the scenario 2.2.

- Scenario 3.2 (variations of just one cost): p_L^6 is erased, p_M^4 becomes p_H^4 while all other nodes have prices described in scenario 2.2. The algorithm ends, as one term of p_H type is obtained, and all reallocation parameters become positive.

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_1 4, e_4 3, e_3 1\}$	+10	-75	-65
v_3	$\{e_1 2, e_2 4, e_4 1\}$	-10	0	-10
v_4	$\{e_1 2, e_2 3, e_3 1\}$	-30	+375	-345

Table 5.2: Dynamics of logistics paths, costs and reallocations for the first iteration

5.3.10   Numerical tests

This paragraph is devoted to some numerical tests. In particular, each of them presents some features that are useful to provide a better idea of dynamics inside a Scattered Manufacturing network.

Test 1   Starting with a Scattered Manufacturing network with $V = \{v_1, v_2, v_3, v_4\}$ set of nodes and a matrix X:

$$ X = \begin{pmatrix} \infty & 10 & 40 & 30 \\ 10 & \infty & 20 & 50 \\ 40 & 20 & \infty & 40 \\ 30 & 50 & 40 & \infty \end{pmatrix} $$

Assume that $P := \emptyset, C(P) := 0, v_s := v_1$. Hence, the demanding nodes is v_1 while node $v_j, j = 2, 3, 4$ is of productive type. Price/quantity plans follow the formulation 5.13 with $p_L^2 = 25, p_L^3 = 20, p_L^4 = 15, p_M^2 = 30, p_M^3 = p_M^4 = 25; p_H^2 = 35, p_H^3 = p_H^4 = 45; k_l^2 = 40, k_L^3 = 50, k_L^4 = 70; k_M^2 = 50, k_M^3 = 90, k_M^4 = 95; k_H^2 = 60, k_M^3 = k_M^4 = 95$.
The total amount of pieces requested is $L = 100$ among the productive nodes $v_2, v_3 and v_4$. Preliminarily, the orchestrator indicates $Q_2 = 10, Q_3 = 30 and Q_4 = 50$. Such quantities are offered at prices p_L^2, p_L^3 and $p_L^4,$, respectively. The iterations run as follows. *First iteration:* From algorithm (PA), we have $P := 100 = C_X^{(1)}$. As $Q_2^{(1)} = Q_2 = 10, Q_3^{(1)} = Q_3 = 30$ and $Q_4^{(1)} = Q_4 = 50$,from the preliminary fragmentation we get $C_Y^{(1)} = Q_2^{(1)} p_L^2 + Q_3^{(1)} p_L^3 + Q_4^{(1)} p_L^4$. Therefore, $C_{TOT}^{(1)} = 1700$.
By considering the computation of weights for nodes, as well as the reallocation via algorithm (RA), Table 5.2 is obtained.

From Table 5.2, node 2 must be excluded in the next iteration.
Second iteration: We get $P := \{e_{14}, e_{43}, e_{31}\}, C := 110 = C_X^{(2)}$. Moreover, $Q_3^{(2)} = 35$ and $Q_4^{(2)} = 55$ at prices p_L^3 and p_L^4 and $C_Y^{(2)} = Q_3^{(2)} p_L^3 + Q_4^{(2)} p_L^4 = 1525$. Hence, $C_{TOT}^{(2)} = 1635$. Table 5.5 shows the posisble reallocations and variations of costs for the next iteration.

As Table 5.5 indicates that $\Delta R_{v_i} > 0, i = 3, 4$, the possible exclusion of nodes v_3 and v_4 does not imply a reduction of the overall cost. Hence, the iterations stop. From the following example, we consider two important phenomena. First, the exclusion of a node from the network does not necessarily

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_3	$\{e_{14}, e_{41},\}$	-50	+725	+675
v_4	$\{e_{13}, e_{31}\}$	-30	+275	+245

Table 5.3: Reallocation plan for the second iteration

imply a reduction of the logistics costs. This is due to the recalculation of the new picking path, which can be very different, also in terms of costs associated to arcs, from the ones of the previous iterations. Second, for the last iteration node v_2 is not considered. At a first sight, one expects that this could occur for node v_4 as $\Delta C_X^{v_4} < \Delta C_X^{v_2}$. Indeed, as $\Delta C_Y^{v_4} >> \Delta C_Y^{v_2}$ the possible exclusion of node v_4. implies the worst case for the overall cost, that increases considerably. Hence, although node v_2, unlike v_4, is the less advantageous in logistics terms for the demanding node v_1, it should be avoided due to an higher fluctuations of the production cost.

Test 2 Consider a SM network with $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and a matrix X:

$$X = \begin{pmatrix} \infty & 10 & 25 & 5 & 7 & 9 & 13 \\ 14 & \infty & 171 & 21 & 12 & 5 & 12 \\ 11 & 15 & \infty & 14 & 13 & 12 & 14 \\ 11 & 10 & 9 & \infty & 17 & 12 & 13 \\ 15 & 12 & 11 & 9 & \infty & 14 & 18 \\ 12 & 13 & 17 & 17 & 19 & \infty & 15 \\ 13 & 11 & 9 & 13 & 15 & 17 & \infty \end{pmatrix}$$

The demanding node is represented by v_i while node $v_j, j = 2, ..., 7$, is of productive type. For productive nodes, prince/quantity functions have levels shown in Table 5.4.

Nodes	p_L^i	p_M^i	p_H^i	k_L^i	k_M^i	k_H^i
v_2	25	35	45	15	25	35
v_3	35	45	55	20	40	50
v_4	30	40	60	25	35	50
v_5	20	30	45	10	25	40
v_6	30	45	60	15	25	35
v_7	30	40	50	25	40	55

Table 5.4: Network price/quantity plans

Assume that $P := \emptyset, C(P) := 0, v_s := v_1$. In this case, $L = 80$ pieces that should be scheduled among the productive nodes. At the beginning of the current iteration, the orchestrator provides $Q_2 = 10, Q_3 = 15, Q_4 = 20, Q_5 = 5, Q_6 = 10, Q_7 = 20$, at prices $p_L^i, i = 2, ..., 7$. The iterations run as follows. *First iteration:*

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{14}, e_{43}, e_{36}, e_{67}, e_{75}, e_{51}\}$	-11	+40	+29
v_3	$\{e_{14}, e_{42}, e_{26}, e_{67}, e_{75}, e_{51}\}$	-17	-120	-137
v_4	$\{e_{15}, e_{53}, e_{36}, e_{62}, e_{27}, e_{71}\}$	-14	+100	+86
v_5	$\{e_{14}, e_{43}, e_{36}, e_{62}, e_{27}, e_{71}\}$	-18	+50	+32
v_6	$\{e_{14}, e_{43}, e_{35}, e_{52}, e_{27}, e_{71}\}$	-18	+50	+32
v_7	$\{e_{14}, e_{43}, e_{36}, e_{62}, e_{25}, e_{51}\}$	-16	+50	+34

Table 5.5: Possible logistics paths, costs and reallocations for the first iteration

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{14}, e_{46}, e_{67}, e_{75}, e_{51}\}$	-3	+905	+902
v_4	$\{e_{15}, e_{52}, e_{26}, e_{67}, e_{71}\}$	-13	+800	+787
v_5	$\{e_{14}, e_{42}, e_{26}, e_{67}, e_{71}\}$	-17	+70	+53
v_6	$\{e_{14}, e_{42}, e_{25}, e_{57}, e_{71}\}$	-7	+925	+918
v_7	$\{e_{14}, e_{42}, e_{26}, e_{65}, e_{51}\}$	-11	+1010	+999

Table 5.6: Parameters variation for the second iteration

Picking algorithm (PA), we have $P := \{e_{14}, e_{43}, e_{36}, e_{62}, e_{25}, e_{57}, e_{71}\}$, $C(P) := 82 = C_X^{(1)}$. As $Q_i^{(1)} = Q_i, i = 2, ..., 7$ and the result $C_Y^{(1)} = \sum_{i=2}^{7} Q_i^1 p_L^i = 2375$, and $C_{TOT}^{(1)} = 2457$.
Computing the weights for nodes involved and applying the reallocation algorithm (RA), Table 5.5 is obtained.

Table 5.5 foresees that node v_3 must not be considered in the second iteration.

Second iteration:
In this case, the new path is $P := \{e_{14}, e_{42}, e_{26}, e_{67}, e_{75}, e_{51}\}$, $C(P) := 65 = C_X^{(2)}$. We get that $Q_2^{(2)} = 13, Q_4^{(2)} = 24, Q_5^{(2)} = 8, Q_6^{(2)} = 13, Q_7^{(2)} = 23$, while $C_Y^{(2)} = 2255$, and $C_{TOT}^{(2)} = 2320$.
As for the computation of reallocation parameters, we refer to Table 5.6.

The iteration stops because $\Delta R^{v_i} > 0, i = 2, 4, 5, 6, 7$. Moreover, the higher increments of terms $\Delta C_Y^{v_i}, i = 2, 4, 5, 6, 7$, are essentially due to the fact that prices become of type p_M. Such an event, indeed, does not always indicate very high discrepancies, as shown by $\Delta C_Y^{v_5}$.
Notice that the described example presents how a network of medium dimensions can reach an equilibrium situation in just one iteration. This suggests that suitable policies of choosing productive nodes could foresee to enlarge the economic radius in order to achieve higher advantages in terms of lower costs.

Test 3 Scattered Manufacturing network with $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, \}$ and matrix X:

$$
X = \begin{pmatrix}
\infty & 10 & 15 & 17 & 12 & 11 & 17 & 18 & 19 & 22 \\
14 & \infty & 11 & 18 & 17 & 24 & 14 & 22 & 18 & 20 \\
17 & 20 & \infty & 22 & 21 & 22 & 23 & 24 & 18 & 19 \\
17 & 16 & 15 & \infty & 19 & 21 & 24 & 23 & 22 & 21 \\
19 & 21 & 22 & 17 & \infty & 22 & 21 & 20 & 19 & 17 \\
20 & 21 & 22 & 23 & 24 & \infty & 24 & 22 & 21 & 19 \\
18 & 17 & 18 & 15 & 14 & 19 & \infty & 21 & 22 & 24 \\
15 & 18 & 21 & 24 & 27 & 22 & 21 & \infty & 18 & 20 \\
38 & 37 & 35 & 32 & 44 & 42 & 41 & 41 & \infty & 40 \\
15 & 18 & 18 & 17 & 16 & 17 & 18 & 19 & 21 & \infty
\end{pmatrix}
$$

Assuming v_1 as the demanding node, while node $v_j = 2, ..., 7$, are the productive ones. Levels of price/quantity plans are in Table 5.7

Nodes	p_L^i	p_M^i	p_H^i	k_L^i	k_M^i	k_H^i
v_2	10	12	15	20	25	35
v_3	15	18	20	25	30	40
v_4	10	11	12	25	35	50
v_5	15	17	19	35	45	50
v_6	25	28	30	35	55	
v_7	25	28	30	30	35	55
v_8	15	18	21	15	20	25
v_9	10	12	14	20	30	40
v_10	10	13	15	25	30	40

Table 5.7: Price/quantity plans for test n. 3

Preliminary, $P := \emptyset, C(P) := 0, v_s := v_1$. In this case, $L = 130$ .which have to be distributed among the nine productive nodes. When the iterations start, the orchestrator indicates the following division: $Q_2 = 10, Q_3 = 15, Q_4 = 15, Q_5 = 25, Q_6 = 20, Q_7 = 15, Q_8 = 5, Q_9 = 10$, and $Q_10 = 15$, at prices $p_L^i, i = 2, ..., 10$. The iterations are listed as follows.

First iteration:
Applying the picking algorithm (PA), the result is the following: $P := \{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{106}, e_{68}, e_{87}, e_{71}\}$, $C(P) := 185 = C_X^{(1)}$. As $Q_i^{(1)} = Q_i, i = 2, ..., 10$ and the result $C_Y^{(1)} = \sum_{i=2}^{10} Q_i^1 p_L^i = 1975$, and $C_{TOT}^{(1)} = 2160$.
Considering the various reallocations and variations of costs, we get Table 5.8 is obtained.
Table 5.8 shows that node v_6 has to be excluded in the second iteration.
Second iteration:
The new path becomes $P := \{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{107}, e_{78}, e_{81}\}$, $C(P) := 161 = C_X^{(2)}$. We get that $Q_2^{(2)} = 13, Q_3^{(2)} = 17, Q_4^{(2)} = 17, Q_5^{(2)} = 28, Q_7^{(2)} = $

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{16}, e_{610}, e_{105}, e_{54}, e_{43}, e_{39}, e_{97}, e_{78}, e_{81}, \}$	-12	+40	+28
v_3	$\{e_{12}, e_{27}, e_{75}, e_{54}, e_{46}, e_{610}, e_{108}, e_{89}, e_{91}\}$	-15	-20	-35
v_4	$\{e_{12}, e_{23}, e_{39}, e_{910}, e_{105}, e_{58}, e_{87}, e_{76}, e_{61}\}$	-10	+65	+55
v_5	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{46}, e_{610}, e_{107}, e_{78}, e_{81}\}$	-20	-30	-50
v_6	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{107}, e_{78}, e_{81}\}$	-24	-245	-269
v_7	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{106}, e_{68}, e_{81}\}$	-24	-95	-119
v_8	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{106}, e_{67}, e_{71}\}$	-19	-20	-39
v_9	$\{e_{12}, e_{23}, e_{310}, e_{105}, e_{54}, e_{46}, e_{68}, e_{87}, e_{71}\}$	-30	+40	+10
v_10	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{58}, e_{87}, e_{76}, e_{61}\}$	-15	+130	+115

Table 5.8: Logistics paths, variations of costs and reallocations for the first iteration test n.3

$17, Q_8^{(2)} = 7, Q_9^{(2)} = 13, Q_10^{(2)} = 18$, and the result $C_Y^{(2)} = 1730$, and $C_{TOT}^{(2)} = 1891$.

The computation of weights for nodes and possible variations of costs are presented in Table 5.9. From Table 5.9, it follows that the next iteration does not

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{15}, e_{54}, e_{43}, e_{39}, e_{910}, e_{107}, e_{78}, e_{81}, \}$	-5	+40	+35
v_3	$\{e_{12}, e_{27}, e_{75}, e_{54}, e_{410}, e_{108}, e_{89}, e_{91}\}$	-10	-45	-55
v_4	$\{e_{12}, e_{23}, e_{39}, e_{910}, e_{105}, e_{58}, e_{87}, e_{71}\}$	-7	+55	+48
v_5	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{410}, e_{107}, e_{78}, e_{81}\}$	-15	-60	-75
v_7	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{108}, e_{81}\}$	-20	-140	-160
v_8	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{107}, e_{71}\}$	-18	-85	-67
v_9	$\{e_{12}, e_{23}, e_{310}, e_{105}, e_{54}, e_{48}, e_{87}, e_{71}\}$	-26	+40	+14
v_10	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{58}, e_{87}, e_{71}\}$	-12	+70	+58

Table 5.9: Logistics paths, variations of costs and reallocations for the second iteration test n.3

foresee node v_7.

Third iteration:
In this case, the new path is $P := \{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{108}, e_{81}\}$, $C(P) := 141 = C_X^{(3)}$. We get that $Q_2^{(3)} = 15, Q_3^{(3)} = 19, Q_4^{(3)} = 20, Q_5^{(3)} = 30, Q_8^{(3)} = 9, Q_9^{(3)} = 16, Q_10^{(3)} = 20$, while $C_Y^{(3)} = 1590$, and $C_{TOT}^{(3)} = 1731$.
Weights for nodes and variations of costs are in Table 5.10. Table 5.10 shows that node v_3 must be excluded in the next iteration.

Fourth iteration:
In this case, the new path is $P := \{e_{12}, e_{25}, e_{54}, e_{410}, e_{108}, e_{89}, e_{91}\}$, $C(P) := 140 = C_X^{(4)}$. We get that $Q_2^{(4)} = 19, Q_4^{(4)} = 24, Q_5^{(4)} = 33, Q_8^{(4)} = 12, Q_9^{(4)} = 19, Q_10^{(4)} = 23$, while $C_Y^{(4)} = 1525$, and $C_{TOT}^{(4)} = 1665$.
Weights for nodes and variations of costs are in Table 5.11. Table 5.11 shows that node v_3 must be excluded in the next iteration.

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{15}, e_{54}, e_{43}, e_{39}, e_{910}, e_{108}, e_{81},\}$	-5	+35	+30
v_3	$\{e_{12}, e_{25}, e_{54}, e_{410}, e_{108}, e_{89}, e_{91}\}$	-1	-65	-65
v_4	$\{e_{12}, e_{23}, e_{39}, e_{910}, e_{105}, e_{58}, e_{81}\}$	-11	+45	+34
v_5	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{410}, e_{108}, e_{81}\}$	-15	-16	-31
v_8	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{510}, e_{101}\}$	-19	-25	-44
v_9	$\{e_{12}, e_{23}, e_{310}, e_{105}, e_{54}, e_{48}, e_{81}\}$	-30	+35	+5
$v_1 0$	$\{e_{12}, e_{23}, e_{39}, e_{94}, e_{45}, e_{58}, e_{81}\}$	-16	+45	+29

Table 5.10: Logistics paths, variations of costs and reallocations for the third iteration test n.3

Nodes	$P \setminus v_i$	$\Delta C_X^{v_i}$	$\Delta C_Y^{v_i}$	ΔR^{v_i}
v_2	$\{e_{15}, e_{54}, e_{410}, e_{108}, e_{89}, e_{91},\}$	-15	+315	+300
v_4	$\{e_{12}, e_{25}, e_{510}, e_{108}, e_{89}, e_{91}\}$	-21	+349	+328
v_5	$\{e_{12}, e_{24}, e_{410}, e_{108}, e_{89}, e_{91}\}$	-16	+299	+283
v_8	$\{e_{12}, e_{25}, e_{54}, e_{410}, e_{109}, e_{91}\}$	-16	+139	+123
v_9	$\{e_{12}, e_{25}, e_{54}, e_{410}, e_{108}, e_{81}\}$	-41	+240	+199
$v_1 0$	$\{e_{12}, e_{25}, e_{54}, e_{49}, e_{98}, e_{81}\}$	-125	+287	+162

Table 5.11: Logistics paths, variations of costs and reallocations for the fourth iteration test n.3

The Iterations stop as $\Delta R^{vi} > 0, i = 2, 3, 4, 5, 8, 9, 10$. The network has reached an equilibrium. Notice that further iterations could be possible if the demanding nodes and production nodes could negotiate about the price/plans. This is object of further research activities, dealing with possible variations of parameters k_L, k_M and k_H.