A REINFORCEMENT LEARNING APPROACH FOR ADAPTIVE DISPATCHING RULE ACQUISITION FOR AUTOMATED GUIDED VEHICLES

by

Yasemin Aylin Aktürk

B.S., Industrial Engineering, Istanbul Technical University, 2014

B.S., Mechanical Engineering, Istanbul Technical University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2019

to my grandparents…

ACKNOWLEDGEMENTS

It would have been impossible for me to complete this work without the help of the people who have supported me emotionally and morally. First, I would like to express my gratitude to my supervisor, Assoc. Prof. Gönenç Yücel for his efforts and supports throughout my thesis work. I would like to thank my research co-advisor Prof. Dr. Ümit Bilge for providing me invaluable guidance throughout this research. I offer my sincere appreciation for the learning opportunities provided by all the lecturers of the department.

To crown it all, I would want to show my gratitude to my primary mentors who are my parents and my sister for their endless support in everything I do in life. I will remain always indebted to you for everything. I must express my profound gratitude to my grandparents for providing me with unfailing support and continuous encouragement throughout life. This journey would not have been possible without them, and I dedicate this milestone to them. Finally, to my caring and supportive soon to be husband, my deepest gratitude. Your encouragement during difficult times is so much appreciated and noted.

My thanks go to all the people who have supported me to complete this work directly or indirectly. I realize that you all are irreplaceable to me. My heartfelt thanks.

ABSTRACT

A REINFORCEMENT LEARNING APPROACH FOR ADAPTIVE DISPATCHING RULE ACQUISITION FOR AUTOMATED GUIDED VEHICLES

There are continuous changes such as specification changes, equipment breakdowns, fluctuations in the customer orders that manufacturing systems of the 21st century have to deal with on a daily basis. Consequently, the need for self-adjustability to market requirements has elevated. One of the many methods tailored to the requirements of the production sector is holonic manufacturing systems (HMS).In this thesis, the practical implementation of HMS is described. A multi-agent based approach for distributed artificial intelligence is proposed to generate effective and adaptive control mechanisms for the management of dynamic processes in a realistic manufacturing testbed. Herein, intelligent agents within a manufacturing system such as products, machines, and automated guided vehicles (AGV) create a self-controlling network to manage the pickup-dispatching problem of multiple single-load AGVs. This mid-level shop floor problem is addressed with a reinforcement learning (RL) method. The application potential of Q-learning, a broadly used RL algorithm, to a pickup-dispatching problem is investigated. The aim of this study is twofold. First, the study is intended to determine if an AGV agent is able to learn the best dispatching rule regarding a system goal in various cases. This is experimentally investigated by an agent based simulation model. Second objective of this study is to demonstrate the feasibility of adaptive learning abilities of AGVs. These results principally show that the AGVs are able to learn to practice the best dispatching rule and are able to adopt another rule when the initially adopted rule starts to fail. The findings essentially demonstrate that the learning AGV agent is able to adapt itself to the changes in the environment and can learn to favor the application of the best action in a given state.

ÖZET

OTOMATİK YÖNLENDİRMELİ ARAÇLARA TAKVİYELİ ÖĞRENME YAKLAŞIMI İLE ADAPTİF DAĞITIM KURALLARI KAZANDIRILMASI

21. yüzyılın üretim sistemlerinin günlük olarak karşılaştığı şartname değişiklikleri, ekipman arızaları, müşteri siparişlerindeki dalgalanmalar gibi sürekli ve dinamik değişiklikler bulunmaktadır. Bu sebeple, üreticilerin pazar gereksinimlerine göre kendi kendini ayarlayabilme ihtiyacı artmıştır. Üretim sektörünün gereksinimlerine uygun birçok yöntemden biri de holonik üretim sistemleridir (HÜS). Bu tezde, HÜS'in bir uygulaması ajan bazlı simülasyon modeli kullanılarak anlatılmıştır. Gerçekçi bir üretim ortamındaki dinamik süreçlerin yönetimi için gerekli, etkili ve uyarlanabilir kontrol mekanizmaları, dağınık yapay zeka uygulanması içeren çok ajanlı bir yaklaşım ile ele alınmıştır. Ürünler, makineler ve otomatik yönlendirmeli araçlar (AGV) gibi bir üretim sistemi içindeki akıllı ajanlar ile, tek yük kapasiteli AGV sisteminin ürün alma sırasındaki eşleşme problemini (pick-up dispatching) yönetmek için kendi kendini kontrol eden bir ağ oluşturulmuştur. Bu çalışmanın amacı iki yönlüdür. İlk olarak, çalışmanın, bir AGV'nin her durumda bir sistem hedefi doğrultusunda en iyi ürün seçme kuralını öğrenip öğrenemediğini belirlemesi amaçlanmıştır. AGV ajanlarının amaç doğrultusundaki en iyi ürün seçme kuralını uygulamayı öğrenip öğrenemediğini tespit etmek için ajanlı bazlı bir simülasyon modeli kurulmuştur. Bu çalışmanın ikinci amacı, AGV'lerin öğrenme yetenekleri sayesinde sistemdeki değişikliklere adapte olabildiklerini göstermektir. Sonuçlar, AGV'lerin sistem performansını eniyileyecek ürün seçme kuralını uygulamayı öğrenebildiklerini ve sistemde bir değişiklik olduğunda ve ilk kabul edilen kural başarısız olmaya başladığında başka bir kurala geçebileceklerini göstermektedir. Bulgular, öğrenen bir AGV ajanının kendisini ortamdaki değişikliklere adapte edebildiğini ve sistem için en iyi eylemin uygulanmasını desteklemeyi öğrendiğini ortaya koymaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Through the centuries each advancement of manufacturing systems is characterized by its concerns: mass manufacturing was intended for low-cost products; for continuous quality improvement, lean manufacturing was proposed; flexible manufacturing was envisioned for products variety; and reconfigurable manufacturing was for the need of self-adjustability to market requirements (da Silva *et al*., 2016).

Throughout the last decade technology drastically advanced in countless fields, like robotics, artificial intelligence, and Internet of Things. The integration of information technology and physical entities where computational and physical objects are interconnected via networks and are able to communicate with each other in real-time is interpreted as a cyber-physical system (Wang & Haghighi, 2016). Cyber-physical systems have become functional solutions to the necessities of the manufacturing plants of this century. This prompt move in computer technology and manufacturing is referred to as the 4^{th} industrial revolution (Wang & Haghighi, 2016; Foit *et al*., 2017). Industry 4.0 is frequently described as the industrial revolution in which interdisciplinary technologies are utilized to integrate equipment and people to industrial factories to deliver services and products in an autonomous manner (da Silva *et al*., 2016). Industry 4.0 draws a picture where the cyber-physical system can manage itself through the interrelationships among its entities (Gräßler & Pöhler, 2017).

Industry 4.0 is often used in relation to the rapid evolution of industrial automation in the manufacturing sector. Universal access to the web has increased the competition between businesses in industrial environments, and has given an open door for establishing endless collaborative networks. The effect of these developments is reflected as the need for new solutions for modeling industrial environments. As the manufacturing sector got familiar with innovative technologies, businesses has paid significant attention to fresh methods of manufacturing process modelling and advanced shop floor control techniques (Foit *et al*., 2017).

Latest developments in manufacturing and computerization redefined the concept of the manufacturing systems. Consequently, some approaches utilized earlier in the field of computer science have been adjusted to the industries' demands. One of the many methods tailored to the requirements of the production sector is the agents-based and holon-based methodologies. The requirements of the new age promoted decentralized production architectures and particularly holonic manufacturing systems (Gräßler & Pöhler, 2017). Autonomous and cooperative building blocks called holons have been used to distribute decisional capabilities within the manufacturing system. The agent-based approach of modelling has been developed and adapted to the philosophy of holonic industrial automation.

There are continuous changes such as specification changes, equipment failures, fluctuations in the customer orders that manufacturing systems have to deal with on a daily basis. The need for decentralization, shorter product lifecycles and rapid reconfiguration are the main challenges of the 21st century that manufacturers face with (Wang & Haghighi, 2016). It is essential that the manufacturing system can promptly react to any abnormal situation. Researchers have offered the Holonic Manufacturing System (HMS) theory as a well-organized discipline for establishing an adaptive manufacturing system that can cope with such fluctuations, interruptions and uncertainties (Hsieh, 2002). The concepts of agents and holons conceptualized in the previous century can cope with the upcoming challenges of manufacturing plants of the future (Wang & Haghighi, 2016).

The holonic concept has been recommended as a system that has the ability to restructure the existing system resources to accomplish a pre-defined production goal by completing the assigned tasks through cooperation. Self-directed and cooperative resources of HMS such as machines, material handling vehicles and products have the ability of intelligent decision-making and negotiating with other individuals to fulfill the organization's goal (Wang & Haghighi, 2016). The proposed production system is able to configure itself regarding current tasks through collaboration among the resources (Gräßler & Pöhler, 2017). Holonic manufacturing systems are capable of dynamically reorganizing manageable entities to succeed a desired set of goals robustly while avoiding unwanted situations (Hsieh, 2002).

Paradigms like “Industry 4.0”, “Cyber-Physical Production Systems” and “Industrial Internet”, have moved the holonic manufacturing system experience to the next level (Gräßler & Pöhler, 2017). Nowadays, the holonic concept is widely used to design intelligent and adaptive cyber-physical systems (Wang & Haghighi, 2016). Academicians have accepted holonic manufacturing systems as the next wave of manufacturing revolution to handle dynamic changes in the industrial environment (Hsieh, 2002).

While tasks are distributed to resources by a global controller in traditional central shop floor control systems, this assignment practice is achieved through communication among the system components in decentralized shop floor controlling approaches. Such heterarchically structured manufacturing systems promote holonic production architecture for an entirely distributed shop floor control. In order to attain decentralization, production resources are enhanced with computational intelligence to generate a self-controlling network (Gräßler & Pöhler, 2017). Therefore, HMS methodology requires each resource to have its own ability of decision-making to improve performance (da Silva *et al*., 2016).

In this thesis, the heterarchical architecture is explained and its practical implementation is presented using a multi-agent simulation model. A multi-agent based approach for distributed artificial intelligence is proposed. An effective and adaptive control mechanism is proposed for the management of dynamic processes in a realistic manufacturing testbed. In accordance with this purpose, an adjusted holonic architecture is established where autonomous and intelligent agents cooperate with other agents. Herein, intelligent agents within a manufacturing system such as products, machines, and automated guided vehicles (AGV) create a self-controlling network to manage the pickup-dispatching problem of multiple AGVs. This mid-level shop floor problem is addressed with a reinforcement learning method where the coordination and distributed control of single-load AGVs are established with a RL algorithm implemented at the individual AGV level. The study examines the potential of the Q-learning procedure in the control of pick-up dispatching rule selection problem of multiple AGVs using an agent based simulation model.

2. PROBLEM BACKGROUND

2.1. Evolution of Manufacturing Control Architectures

In a manufacturing environment, a control architecture is characterized as the set of rules defined for managing the process flow (Bilge *et al*., 2016). Control frameworks, rule sets, and guidelines eventually develop into a control architecture to manage and operate the shop floor effectively. The manufacturing system's performance is highly associated with the shop floor control architecture (SFCA). SFCA has a direct impact on the implementation and the key performance indicators of the production plant since the organization of the control determines the capabilities of the shop floor to control the resources in an effective manner (Bilge *et al*., 2016). Various approaches have been proposed to cope with the industrial environments' physical problems and conceptual challenges.

Most of the SCFAs offered in the literature fall into one of the four basic control structures: centralized, fully hierarchical, semi-heterarchical (modified hierarchical) and full heterarchical control architectures. Figure 2.1 summarizes the different shop floor control architectures to allocate control decisions (Trentesaux, 2009). A basic comparison of the shop floor control architectures is presented in Table 2.1 (Leitão, 2009).

In centralized control architectures, a central decision making object allocates all of the tasks to resources and manages all of the activities of the system entities (Bilge *et al*., 2016). Conventional computer-integrated manufacturing systems implemented fully hierarchical shop floor control mechanisms (Wang & Haghighi, 2016). Predetermined layers of hierarchies, and a global controller entity with a parent-to-child stream of instructions, are the key attributes of hierarchically structured systems. In such cases, the activities of each entity is firmly determined by the central controller. The interactions of the system components are only restricted to their parents or children (Wang & Haghighi, 2016). One of the most recognized hierarchical SFCA in the literature is Advanced Manufacturing Research Facility (AMRF), which possess five levels of hierarchy: facility, shop, cell, workstation and equipment (Bilge *et al*., 2016).

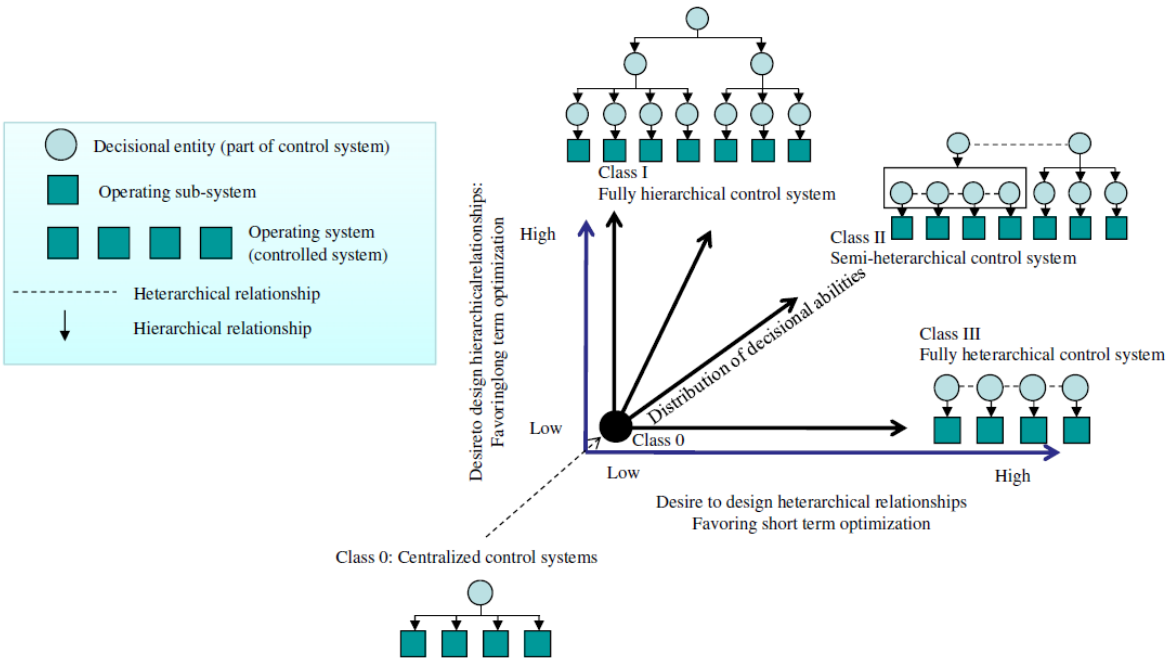

Figure 2.1. Shop floor control architectures (Trentesaux, 2009).

Despite the fact that hierarchical control architectures typically achieve global optimization due to their central decision-making unit (Trentesaux, 2009) they fail to achieve fault tolerance and robustness (Wang & Haghighi, 2016). Conventional manufacturing control mechanisms are not fully capable of being responsive, flexible, robust and re-configurable, due to the rigidity and centralization of their layered control architecture (Leitão, 2009). In case of even a single disturbance or failure, centralized SFCA typically ends up halting the entire system to shut down (Wang & Haghighi, 2016). They fail to handle failures effectively.

Many researchers criticize the hierarchical control systems as they cannot handle the requirements of the upcoming challenges of the dynamic manufacturing systems such as localized decision-making authority, cooperation, integration and reconfiguration flexibility characteristics, openness, interoperability, scalability, agility, fault tolerance, reconfiguration, etc. (Kanchanasevee *et al*., 1997; Trentesaux, 2009; Leitão, 2009; da Silva *et al*., 2016).

Table 2.1. A basic comparison of the shop floor control architectures (Leitão, 2009).

Traditional control solution	**Distributed and intelligent control solution**
Centralized solution for each individual control function	Distributed solution with cooperation between nodes and focusing on more than one control function
Rigid and static architecture	Flexible, programmable and dynamic architecture
Client–server relations	Holon–holon relations
Top-down approach	Bottom-up approach
Communications one for many (1-N)	Communications many for many (N–M)
Intelligence centered in the top levels	Intelligence distributed by the control levels
Efficiency through specialization	Efficiency through flexibility
Weak response to disturbances	Strong response to disturbances
More efficient for high volume and low variability	More adequate for high–low volume and medium–high variability

Industry needs have moved from static optimization to dynamic optimization. The need for a system that is considerably more reactive to disturbances on the shop floor and to fluctuations in product life cycle has aroused. It is becoming more and more important to build shop floor control architectures that are modifiable, extensible, reconfigurable, adaptable, and fault tolerant (Kanchanasevee *et al*., 1997). Businesses, nowadays, are in need of production systems that provide reactive, satisfactory, adjustable and robust solutions (Trentesaux, 2009). These requirements have moved the trend in shop floor control architecture towards more heterarchy. The drive to replace the hierarchical organizational structure of shop floor control with the heterarchical architecture arose with the revolution of distributed computing and computer networks in the 80s (Foit *et al*., 2017).

Factories of the future need to be flexible in many ways to serve custom-made products in small batches in a timely manner. The next wave of manufacturing facilities need to be dynamically reconfigurable and need to integrate distributed control to promptly react to any irregularities occurring in the system (Kanchanasevee *et al*., 1997).

Therefore, researchers and practitioners in the field have modified the hierarchical control architecture in various ways. Modified hierarchical control systems is the architecture where the rigid master-slave relationship gets relaxed, a semi-heterarchical system is structured, the responsibilities of lower level units increase and the entities are allowed to communicate peer-to-peer (Bilge *et al*., 2016). The idea behind this type of control architecture creates a model where the responsibility is distributed among decisional entities and certain group of agents interact to solve a problem collectively (Kanchanasevee *et al*., 1997). Such architectures are a mixture of hierarchical and heterarchical controls that, in theory, reap the benefits of both coordination mechanisms while avoiding their certain weaknesses (Wang & Haghighi, 2016). However, this relaxation in the hierarchical control could not meet the requirements of the dynamic manufacturing systems. Uniting a global vision with a comparatively quick response to unforeseen disturbances of the system was generating response time lags (Trentesaux, 2009; Bilge *et al*., 2016).

To manage the upcoming challenges within the manufacturing surroundings, fully heterarchical control architectures, or so-called distributed control systems, were proposed (Trentesaux, 2009). Academics in the field of Distributed Artificial Intelligence (DAI) have been developing control structures where a group of agents cooperates to handle tasks. Heterarchical control architecture is such a structure where a set of intelligent entities work self-sufficiently and share data to achieve the objectives of the production system in the absence of a central coordination mechanism (Kanchanasevee *et al*., 1997).

Distributing decisional capabilities to decisional entities leads to non-centralized control systems. In such a distributed control architecture, sub-systems are characterized as intelligent decision-making entities that perform tasks autonomously and collaborate to accomplish the overall objective of the system (Wang & Haghighi, 2016). In such control architectures, decisional responsibilities are assigned to local decision-making entities. The main idea in decentralized control architectures is to allow the intelligent entities to work together and respond rapidly as an alternative to requesting decisions from upper decisional level (Trentesaux, 2009). This distribution of all controlling tasks to intelligent decision makers is also the core vision of cyber-physical manufacturing systems (Gräßler & Pöhler, 2017).

Even though the heterarchical SFCA demonstrates advantages such as autonomy, robustness, agility, flexibility, collaboration, reactivity to the disturbance (Morariu *et al*., 2015; da Silva *et al*., 2016), it generally fails to obtain global optimization (Farid, 2004; Trentesaux, 2009; Leitão, 2009). Therefore, most hierarchical control systems define negotiation protocols as a group of rules that manage the interactions among the system entities to add a certain level of global aspect to the local decisions (Aziz, 2013).

Using an intelligent and distributed SFCA approach, multi-layered problems can be divided into a number of sub-problems and each one of the small problems can be allocated to a control unit. Intelligent agents such as robots, CNC machines, material handling devices in agile manufacturing systems can make self-sufficient decisions (Tripathi *et al*., 2005), have their own knowledge and objectives, and can function intellectually; however, none of them has an overall system view (Leitão, 2009). Controlling units are committed to handle their own decision process, which usually consists of a triggering event, problem formulation, problem solving and practice of the outcomes (Trentesaux, 2009).

A heterarchically controlled manufacturing system is able to satisfy the above requirements such as dynamic reconfiguration to cope with unusual events, and the introduction of new equipment and production methods (da Silva *et al*., 2016). The ultimate objective is to maintain effective and efficient system activities while minimizing downtime for reconfiguring, replanning, and rescheduling manufacturing tasks. Nonhierarchical SFCA with a high degree of machine intelligence may ease the control of the system in case of unexpected and unforeseen situations, increase maintainability and modifiability of the shop floor (Foit *et al*., 2017).

2.2. The Use of Holonic and Agent-Based Methods in Modelling Manufacturing Systems

A heterarchical manufacturing control system that fulfills the aforementioned challenging requirements functions in a completely different way when compared to the conventional centralized control architectures. Simultaneously, the need for the development of new methods for forming such control architectures has emerged.

The idea of heterarchical manufacturing control system stems from the emergence of entities that have the ability of making judgements, participating in teamwork and acting independently. In these circumstances, the challenge is to develop manufacturing control systems possessing the characteristics of heterarchical control such as adaptation, responsivity to disturbances, dynamic architecture and modularization that support small batches, high product variety, high quality and low costs of reconfiguration. The expectations are parallel to the ones in the case of the agent or holon based methods, which were developed shortly after the evolution of heterarchical SFCA (Foit *et al*., 2017). Therefore, multi-agent-based control and holonic manufacturing control have become two widely utilized models in the field of distributed and intelligent manufacturing control (Leitão, 2009). These approaches seem to be suitable to cope with the afore-mentioned requirements, because of their explicit advantages such as being scalable, resilient, flexible, adaptive and fully tolerant (Trentesaux, 2009; Leitão, 2009; Wang & Haghighi, 2016; da Silva *et al*., 2016). In addition, agent and holon-based methodologies embrace interactions between decisional entities to promote the emergence of a global behavior (Trentesaux, 2009).

In the field of manufacturing, studies on HMS and MAS are encouraged by flexible production plans, intellectual decision making processes and distributed manufacturing control structures. Farid (2004) gives a wide-ranging review of the holonic manufacturing systems literature. Leitão (2009) presents the state-of-the-art in intelligent heterarchical manufacturing control systems via fresh phenomena such as multi-agent systems and holonic manufacturing systems. Wang and Haghighi (2016) also report a literature review on the evolution of holons and explain the suitability of holons and agents for modelling of cyber-physical production systems.

2.2.1. Multi-Agent Systems (MAS)

A number of researchers using different approaches has defined an “Agent”. Wooldridge (1998) delineated an agent as “A software entity situated in some environment that is capable of autonomous action in this environment in order to meet its design objectives”. Franklin and Graesser (2005) state their own definition: “An autonomous agent

is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future". Unland (2015) defines an agent in terms of a computer system: "An agile and robust software entity that intelligently represents and manages the functionalities and capabilities of an industrial unit". Wang and Haghighi (2016) concludes as "An agent is an autonomous component that embodies physical or logical objects in the system, capable to act in order to achieve its goals, and being able to interact with other agents, when it does not possess knowledge and skills to reach alone its objectives".

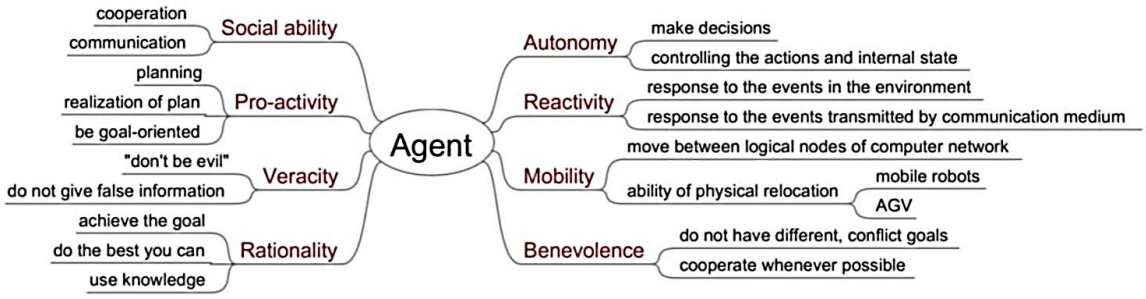

Figure 2.2. The attributes of an agent (Foit *et al*., 2017).

Some of the main characteristics of an agent have been visualized on Figure 2.2 (Foit *et al*., 2017). A number of authors have recognized the key attributes of an agent as follows:

- Reactive agents can sense the changes in their surroundings and are able to react (Leitão, 2009; Trentesaux, 2009; da Silva *et al*., 2011). Reactive agents have a pre-specified action for every potential sensory input (Farid, 2004).
- Agent are capable of being proactive towards achieving their assigned global and local objectives (Farid, 2004; Wang & Haghighi, 2016).
- Agents are autonomous. Agents possess enough knowledge to decide and function on their own (Wooldridge, 1998). They are able to act independently according to the requirements of specific situations (Dominici *et al*., 2010). Agents are the individual entities of a heterarchical structure and therefore have the high-level of autonomy. In such autonomous control structures, the absence of a global decision make makes the decision-making local (Trentesaux, 2009). However, Silva *et al*.,

(2011) show that this characteristic of the MAS allows a high performance against disturbances.

- Agents can act together cooperatively to achieve the global objective of the system (Trentesaux, 2009).
- Intelligent agents are recognized as adaptive decision making entities (Kanchanasevee *et al*., 1997; Hsieh, 2002; Leitão, 2009; Aziz, 2013; Zhang *et al*., 2012; da Silva *et al*., 2016). Agents can learn from their previous behaviors, past experiences and can apply their experiences to future scenarios (Wang & Haghighi, 2016). They have dynamic and adaptive responses to disturbances (Aziz, 2013).

The idea of autonomous and intelligent units stemmed from the DAI systems researches in the 90s (Wang & Haghighi, 2016). Agents are being promoted as a next generation model for representing complex distributed systems (Wooldridge, 1998; Jennings *et al*., 2001). To put decentralized control architecture into practice in a manufacturing facility, resources such as the machines and the material handling vehicles involved the manufacturing system and the production parts needed to be modeled as self-directed and self-acting agents (Kanchanasevee *et al*., 1997).

In non-hierarchical SFCAs, a central controller does not control decision makers (agents). Agents are autonomous and co-operating entities and achieve the system goal by co-operation and communication between agents (Trentesaux, 2009; Bilge *et al*., 2016). These agents usually use a heuristic set of rules to implement a bidding scheme that allows the parts to determine dynamically and cooperatively to take an action (Kanchanasevee *et al*., 1997). That is, agents are able to act without direct control of a parent instructor, but instead are capable of negotiating with other agents if necessary (Foit *et al*., 2017).

Leitão (2009) define a multi-agent system as a group of agents that represent the entities in an organization, where agents can cooperate to achieve their goals when they individually do not possess enough knowledge to achieve their particular objectives. MAS entities are liable for fulfilling their individual objectives, but they can also communicate with other agents to achieve an objective of the system. In other words, MAS represents individual and social behaviors in dispersed systems (Foit *et al*., 2017).

The studies indicate that in the field of industrial automation, agents are commonly used to represent physical resources, such as machines, material handling systems, products, and logical entities, such as the orders and schedulers. There is a wide choice of MAS engineering available in the literature. For instance, the suggested multi-agent system by Barenji *et al*. (2016) is designed as a network of software agents that interact with each other and with the system actors. The considered agents are shop management agents, agent managers, shop monitoring and command agents, station control agents, station monitoring agents, agent machine interfaces, and manufacturing resource agents. They built a simulation platform for multi agent based manufacturing control system based on the hybrid agent. Foit *et al*. (2017) have order agents, delivery agents, a supervisor agent, a manipulator agent, an assembly station agent. They construct a multi agent system where the assembly of two different types of toys take place. Gräßler and Pöhler (2017) suggested a remote controlled car manufacturing system with product agents, order agents, and resource agents such as warehouses, machines, transport vehicles, and employees. A recent study by Xue *et al*. (2018) applied a multi-agent method where each physical and logical entity was presented as an agent. Their system is designed in such a way that the machine agents take charge of job processing and automated guided vehicle agents decide the next transferring task.

2.2.2. Holonic Manufacturing Systems (HMS)

The word “holon” was first introduced by Koestler (1969). His studies focused on wholeness and singularity in living organisms and social systems. His philosophies originated from the idea that each entity is a part of a bigger whole while it is individually acknowledged as a whole for some other. The word “holon” holds this dichotomy of wholeness and partness. In Greek language, “holos” means “whole” and the suffix “on” denotes “part. The term implies the behaviors of actual agents that act independently but can work together to form obvious self-organizing systems.

Kanchanasevee *et al*. (1997) define a holon as “an autonomous and cooperative building block of a manufacturing system for transforming, transporting, storing and/or validating information, and physical objects.” Thus, holons are made up of an information processing part and frequently a physical processing part (Wang & Haghighi, 2016). This

particular construction of holons grants the incorporation of actual physical objects into the decisional control system.

Regarding decision making activities, an information processing part is inserted to the holon. The logical part is in authority of interactions and negotiations. The physical element of the holon characterizes the actual object in the manufacturing system such as machines, machine tools, and AGVs. On the other hand, there are non-physical holons with the logical part only such as customer orders.

Koestler used the term "holarchy" to define the holonic control architecture. Holarchy is known as a group of holons that are able to combine forces to accomplish a goal (Wang & Haghighi, 2016). The holarchy outlines the main rules for cooperation of the holons and thereby restricts their independence (Kanchanasevee *et al*., 1997). Koestler's ideas laid foundations for a new kind of control model for the industrial systems and consequently the holonic manufacturing system (HMS) archetype has emerged. Holonic manufacturing system is a holarchy that unites manufacturing activities (Farid, 2004). Being closely associated to the concept of MAS, a HMS is delineated as a set of actuators that unite the full range of production processes (Wang & Haghighi, 2016). In a manufacturing setup, holons make independent judgements on the use of manufacturing resources, e.g. machines, material, work force, energy, and time by negotiating with other holons in the holarchy.

Various architectures have been recommended for HMSs. Earliest studies as well as current work focus on the product-resource-order-staff (PROSA) architecture first introduced by Van Brussel *et al*. (1998) during their research on Intelligent Manufacturing Systems (IMS). PROSA design is built on three main types of holons: order holons, product holons, and resource holons. Basic building blocks of PROSA and their relations are shown on Figure 2.3 (Van Brussel *et al*., 1998). The resource holon represents the resource in the manufacturing system and holds a logical part that controls the resource. The product holon contains the process and product knowledge. The order holon symbolizes the tasks. The staff holons help and guide the other holons in the holarchy. In other words, each of these holon categories is in charge of one characteristic of the shop floor control mechanism. The holons are usually engineered using the object-oriented concepts.

The resulting structure makes it easier to integrate new components, enables easy self-configuration and extension. Therefore, it allows simple reconfiguration of the system (Van Brussel *et al.*, 1998).In his review of HMS, Leitão (2009) highlights the issue of self-configuration and easy extension for real industrial cases and notes that regarding this issue most researches stop at the prototype phase.

In this study, a slightly modified PROSA architecture, which originates from holonic manufacturing notion, is used.

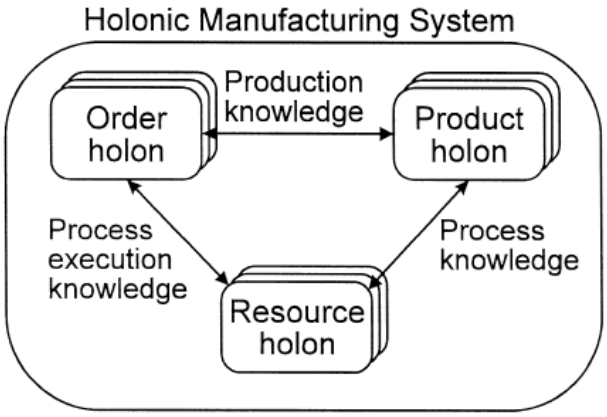

Figure 2.3. Basic building blocks of PROSA and their relations (Van Brussel *et al*., 1998).

2.2.3. Agents and Holons in Modeling of Manufacturing Systems

Manufacturing systems of the 21st century have to satisfy the following major requirements: enterprise integration, distributed organization, heterogeneous environments, interoperability, open and dynamic structure, cooperation, integration of humans with software and hardware, agility, scalability, and fault tolerance. Multi-agent-based control and holonic control are two appropriate approaches that address these expectations. A more comprehensive explanation can be found in Shen and Norrie (2013).

Implementing artificial intelligence techniques, increased the system's agile and correct response capabilities to disturbances, and improved its ability to adapt to changes without external supervision (Leitão, 2009). In this respect, agent-based and holonic systems give the impression of being major keywords in the immediate future.

Holons can create holarchies to establish cooperative groups for accomplishing a mutual goal (Koestler, 1969). Such short-term goal settings are compulsory as correcting actions if disturbances occur in the system (Trentesaux, 2009). Thus, in HMS, holons organize holarchies dynamically and then are able to dissemble after the task has been completed (Wang & Haghighi, 2016). Agents also can form various SFCAs such as hierarchies, heterarchies, or semi hierarchies. They can learn from previous circumstances, understandings and experiences and foresee the possible future scenarios (Wang & Usher, 2005; Chen *et al*., 2015; Xue *et al*., 2018; Ansari *et al*., 2018). The autonomy and learning characteristics of both holons and agents enables them to take initiatives, thus react in turbulent environments.

A series of recent studies has indicated that the holon is the special case of an agent. The authors also point out a few absolute dissimilarities between holons and agents. The most distinct characteristic of holons is that they have a distinct information processing and a physical processing part, and agents do not have this type of internal separation (Foit *et al*., 2017). A comparison of the capabilities and structures of agents and holons is given on Table 2.2 (Wang & Haghighi, 2016).

The literature review shows that resemblance of agents and holons has caused the researchers to combine both methods. Barbat *et al*. (2001) argued pros and cons for the agent- and holon-based approaches and decided to integrate both methods in order to attain better outcomes. They propose several theories specifying how the holonic paradigm overcomes the agent-based approaches' limitations in the field of establishing multi-level hierarchies in dynamic surroundings. This has also been explored by Unland (2015) who stated that the holonic approach extended by agent norms and policies, offers a foundation for multi-agent based models that can support hierarchical decision making as well. He also introduced the term "agentification of holon". This has been adopted by a great number of authors in the literature (Hsieh, 2002; Dominici *et al*., 2010; Junqueir *et al*., 2011; da Silva *et al*., 2016; Gräßler & Pöhler, 2017). On the other hand, there still are papers, where the writers practice pure agent-based method (Tripathi *et al*., 2005; Erol *et al*., 2012; Barenji *et al*., 2016), and some use pure holon-based approach (Kanchanasevee *et al*., 1997; Lind & Roulet-Dubonnet, 2011).

Both HMS and MAS have been practiced in the field of distributed IMS. HMS is a concept that is limited to the manufacturing systems. Holons are used to characterize both physical and non-physical entities particularly in manufacturing setups. However, multi agent systems have been widely employed in other fields of research, too.

In general, MAS is for modelling manufacturing resources such as operators, machine cells, machines, tools, fixtures, AGVs, products, parts, operations to facilitate manufacturing resource planning, scheduling and execution control (Shen & Norrie, 2013), while HMS is a conceptual model precisely recommended for the future developments in industrial organizations and their challenges (Wang & Usher, 2005). Therefore, MAS has been applied very successfully to the HMS concept. Monostori *et al*., (2006) outline the potential manufacturing applications of MAS through a comprehensive survey. Good discussions on agent-based technology for holonic manufacturing systems can be found in Barbat *et al*., (2001), Farid (2004), Trentesaux (2009), Leitão (2009), Wang and Haghighi (2016). In the course of this study, the terminologies of agents and holons have been used interchangeably.

In this section, some previous projects have been reviewed in a tabular format on Table 2.3. For each project, where applicable, its name, application domain, communication or negotiation modelling approach and main characteristics are given.

Table 2.2. General comparison of holons and agents (Wang & Haghighi, 2016).

	Holon	Agent
Nature	A conceptual unit that needs to be realised through implementation	A software component for modelling the behaviour of a system
Application	Currently in manufacturing systems	In different research areas
Characteristics	Autonomous, cooperative, open, proactive, reactive, learner, and recursive	Autonomous, proactive, reactive cooperative/ competitive, open, learner, non-deterministic, and mobile
Architecture	Explicit separation: information processing part and physical part (optional)	No explicit separation: one software unit
Organisation	Holarchies	Different organisational architectures such as hierarchies, heterarchies, etc.
Interfaces	Human – other holon	Humans – other agents
Main contributions	Architecture: integration of control systems, physical equipment and humans Performance: increasing the adaptability of shop floor to the introduced disturbances	Performance: providing software techniques for adaptive decision making

Table 2.3. Summary of studies using agent/holon technology.

Authors	**Title**	**Communication/ Negotiation modelling approach**	**Agents/Holons**	**Case**
Kanchanaseve e *et al*., (1997)	Contract-Net Based Scheduling for Holonic Manufacturing Systems	Contract-Net protocol	Holons: Product holon, Machine Holon, Scheduler holon, Computing Holon, Negotiation Holon	Induction motor production system.
Hsieh, (2002)	Multi-Agent Control Of Holonic Manufacturing Systems Based On Petri Net	Combination of Petri Nets and Contract Net protocol	Holons: Resource holons, Product holons, Order holons	Not available.
Junqueir *et al*., (2011)	Design Of Control Systems Based On Holon, Petri Net And Multi-Agent System	Architecture for normal and with fault tolerance scenario (schema) and Petri Nets	Holons: Product holon, Strategies holon, Supervisor holon, Operational holon	FMS-Cell.
Aziz, (2013)	A Review of the Negotiation Protocol for Agent based Manufacturing System Control	The Multi-Agent System Negotiation Model	Agents: Job and Resource agents, Plant agents, Group agents	Theoretical manufacturing scenario.

Table 2.3. Summary of studies using agent/holon technology (cont.).

Authors	**Title**	**Communication/ Negotiation modelling approach**	**Agents/Holons**	**Case**
Morariu *et al*., (2015)	Multicast dataset synchronization and agent negotiation in distributed manufacturing control systems	The negotiation protocol proposed here has two components: solution proposal and solution evaluation.	Agents: Order Agent, Work in Progress Agent, Resource Agent, Mediator Agent	A manufacturing scenario.
da Silva *et al*., (2016)	Control architecture and design method of reconfigurable manufacturing systems	Applying Petri Nets to model the workflow to solve unpredictable demands and to implement fault-tolerance behavior.	Holons: Product Holon, Task Holon, Supervisor Holon and Operation Holon.	Cylinder bodies manufacturing system and workstation representation.
Barenji *et al*., (2016)	Simulation Platform For Multi Agent Based Manufacturing Control System Based On The Hybrid Agent	-Colored Petri Nets -Sequence diagram for starting a new task in the system	Agents: Shop management agents, agent managers, shop monitoring and command agents, station monitoring agents, agent machine interfaces, and manufacturing resource agents	An example of the simulation platform is presented for a flexible manufacturing system.

Table 2.3. Summary of studies using agent/holon technology (cont.).

Authors	**Title**	**Communication/ Negotiation modelling approach**	**Agents/Holons**	**Case**
Bilge *et al*., (2016)	Implementation And Performance Based Comparison Of Shop Floor Control Architectures Using Distributed And Parallel Simulation	Contract-Net protocol	Holons: Product , Resource, Order	An example of the simulation platform is presented for Boğaziçi University Flexible Automation and Intelligent Manufacturing Systems (BUFAIM) Laboratory
Foit *et al*., (2017)	The comparison of the use of holonic and agent-based methods in modelling of manufacturing systems	Only the neighboring agents “talk” directly	Agents: Order agent, Delivery agent, Supervisor Agent, Manipulator agent, Assembly station agent. Holons: Order holon, Resource holon Product holon	Assembly of two different types of toys: a car and a dump truck.

Table 2.3. Summary of studies using agent/holon technology (cont.).

Authors	**Title**	**Communication/ Negotiation modelling approach**	**Agents/Holons**	**Case**
Gräßler & Pöhler, (2017)	Implementation of an adapted holonic production architecture	The part list is broken down and for each item; a negotiation procedure among suitable resources is performed.	Holons: Product, Order, Resource (warehouse, machine, transport, employee)	Manufacturing of four different elaborations of a remote controlled car.

2.2.4. Role of Communication in Holonic Manufacturing Systems

Both MAS and HMS represent a decentralized environment where there are multiple perspectives and/or the opposing interests. Furthermore, the agents in MAS have to interact somehow with each other to achieve their individual objectives or a system goal. As has been previously reported in the literature, these interactions can vary from simple data exchanges, to requests for specific actions to be executed and on to cooperation and coordination (Jennings *et al*., 2001). In this context, cooperation often stands for agents working together to accomplish a global goal, and coordination typically means arranging for activities to be performed in a rational manner. A discussion on frameworks for cooperation in distributed problem solving can be found in Davis (1981). A latter study of Andreadis *et al*. (2014) examine and categorize multi-agent systems based on their coordination technique.

Nevertheless, possibly the most widely utilized and the most powerful mechanism for managing agent interactions is negotiation. A number of authors have recognized negotiation as the process by which a group of agents come to a mutually acceptable agreement on some matter (Lander & Lesser, 1993; Jennings *et al*., 2001; Wong *et al*., 2006; Aziz, 2013).

Wong and Fang (2010) define negotiation protocols as a group of rules that manage the interactions among the system entities. They report that the rules, in this context, govern the types of the participants (e.g. the negotiators), the stages of the negotiation process, the occasions that changes the negotiation states and the possible legitimate actions of the negotiators in specific situations. In general, the majority of prior negotiation protocols draw a framework where the negotiators have to make proposals, exchange options, in some cases offer allowances, and eventually establish a mutually acceptable contract.

There exists a considerable body of literature on how negotiation protocols reinforce agents to cooperate and coordinate (Jennings *et al*., 2001; Aziz, 2013). The simplest scenario is where the participants can either accept or reject an offer requesting to perform a task. Next stage is where the bidders are allowed to change the values of the negotiation object; they know how to make counter-proposals to guarantee a better contract that fits their individual objectives. As a final point, negotiators might have the potential to alter continuously the structure of the issues in the negotiation by adding or removing issues.

A great number of authors in literature has discussed the communication issues. Communication languages, ontologies, interaction protocols and algorithms have been offered. The review of the main communication/negotiation modelling concepts is presented in Table 2.3.

Most studies have relied on the Contract-Net based approach, established by Smith (1980). The contract net protocol developed a set of rules for managing communication and control of entities in a decentralized problem (Smith, 1980). An agent sends a message to other agents saying that some task must be completed. The declaration holds a short task description and the criteria that must be satisfied by the agents involved in the auction. The format of the expected bid is prescribed. The bid submission deadline is also announced. The agents reply the announcement individually by sending bids. The proposal of each agent depends on the agent's capabilities, local measures and assessments. The bids are compared after the expiration time, the agent with the best bid wins the auction. Either an acceptance message or a refusal note acknowledges the agents. Once the task has been assigned to an agent, a contract is concluded.

2.3. Automated Guided Vehicle Systems

Material handling systems (MHS) using automated guided vehicles are frequently used in manufacturing plants, warehouses, and distribution centers. The transportation is made with battery-powered vehicles, equipped with manual or robotic pick-up and drop-off mechanisms in addition to the capability of automated obstacle-detection. Automated guided vehicle systems (AGVS) are computer-controlled material-handling systems usually used for repetitive tasks. A fleet of automated guided vehicles, a navigation network, a set of rules for dispatching and routing, plus traffic-management software is involved in an automated guided vehicle system.

AGV based material handling system installation require a number of decisions to be made. Mahadevan and Narendran (1990) address the key issues concerning the design and operational control of AGV-based material handling for flexible manufacturing systems. These issues include the number of vehicles required, the layout of the AGV tracks in the shop, the traffic flow pattern along the AGV tracks, decisions regarding traffic control, provision of control zones and type, number and capacity of buffer for the vehicles and AGV dispatching rules. Co and Tanchoco (1991) review the research on AGV vehicle management. Peters *et al*. (1996) formalize AGVs classification from a control perspective. Their outcomes are beneficial for understanding the impacts of the AGV design choices on the control system. The study of Le-Anh and De Koster (2006) presents a later review on design and control of AGVS. They report most important matters regarding guide-path design, estimating the fleet size, scheduling of the vehicles, positioning of the idle vehicles, battery management, routing, and conflict resolution. These questions are of central interest in a recent study conducted by Gaur and Pawar (2016). They review the various factors influencing the design and operation issues of AGVs relying on four main factors: throughput, unit load, flow path design and fleet size.

Below, the main issues in design and control of AGVs are summarized in tabular format on Table 2.4. The mentioned problems associate with different stages of the decision making process. According to Le-Anh and De Koster (2006), the issue of guide-path design is treated at the highest level, so called strategic level. At this level, decisions have strong impact on the choices at other levels. Mid-level, tactical issues take account of estimating

the number of vehicles, vehicle scheduling including dispatching, positioning idle vehicles and, handling battery charging. Decisions at the operational level involve the vehicle routing and conflict resolution problems.

This thesis focuses on tactical decisions, particularly on dispatching strategies. A typical product in a job (unit) type production system usually visits several machining centers before its machining requirements are satisfied. Products continuously circulates between the machines in the manufacturing environment. Egbelu and Tanchoco (1984) recognized this movement of parts as the foundation of the AGV dispatching problem. After the completion of the assigned delivery task, the vehicle is either assigned to another task or set idle. The AGV remains to be idle until another mission occurs in the system. If there is only one unattended job to transfer, then the task assignment problem becomes trivial. However, if there are a number of tasks waiting for the services of an AGV, then the task assignment problem becomes a serious operational decision. The problem may also occur in a different form. When there are several idle AGVs and a task to carry out, a selection criterion is needed to select an AGV to fulfill the transfer request. This problem is referred to as AGV dispatching problem.

Over time, an extensive literature has developed on issues of vehicle-task matching. Egbelu and Tanchoco (1984) categorize dispatching decisions into two groups. The first one considers the selection of an idle vehicle to assign a load to pick-up. This is usually a result of a request from a work center for an AGV. The second category involves the selection of one work center from multiple work centers requesting a vehicle. This scenario generally contains a single AGV and multiple machining centers.

The first class of dispatching problem is known as “Work center initiated task assignment (dispatching) problems”, while the second category is referred to as “Vehicle center initiated task assignment (dispatching) problems”. The decisions are to prioritize and select the highest priority AGVs or machining center, respectively. Egbelu and Tanchoco (1984) offer a few dispatching strategies for both cases:

- Work center initiated dispatching rules: random vehicle rule, nearest vehicle rule, farthest vehicle rule, longest idle vehicle rule, least utilized vehicle rule
- Vehicle initiated dispatching rules: random work center rule, shortest travel time/distance rule, longest travel time/distance rule, maximum outgoing queue size rule, minimum remaining outgoing queue space rule, modified first come-first serve rule, unit load shop arrival time rule

Le-Anh and De Koster (2006) classify AGV dispatching rules as single-attribute, multi-attribute, hierarchical, look-ahead and pre-emption dispatching rules. Single-attribute dispatching regard only one criterion to dispatch. They classify the single-attribute dispatching rules as follows: rules based on travel distance (distance-based), rules based on queue length (workload-based), rules based on waiting time (time-based), and rules based on vehicle availability. Multi-attribute dispatching rules use multiple parameters to make the decision. Hierarchical dispatching rules introduce hierarchical logic levels for prioritizing the parameters. Look-ahead dispatching rules use some advance information about upcoming requests.

Table 2.4. The main issues in design and control of AGVs (Le-Anh & De Koster, 2006).

Issues in design and control of AGVs	Guide-path design	Number of parallel lanes	Single lane Multiple lanes
		Flow topology	Conventional Single-loop Tandem
		Flow direction	Unidirectional flow Bidirectional flow
	Estimating the number of vehicles	Single-load capacity vehicles	
		Multi-load capacity vehicles	
	Managing automated guided vehicle systems	Vehicle scheduling system	Offline scheduling system Online scheduling system Vehicle dispatching system • Decentralized control system • Centralized control system
	Vehicle positioning strategy	Static vehicle positioning strategy	Central-zone positioning rule Circulatory-loop positioning rule Drop-off point positioning rule Distributed-positioning rule
		Dynamic vehicle positioning strategy	
	Battery management	Oppurtunity charging	
		Automatic charging	
		Combination system	
	Vehicle routing and conflict resolution	Vehicle routing	Aprroaches for tandem systems Approaches for conventional systems
		Conflict resolution	Balancing the system workload Forward sensing Control the traffic at intersections Zone planning

3. PROBLEM DEFINITION AND PROPOSED APPROACH

3.1. Problem Definition and Research Objectives

In the field of agent- and holon-based modeling there are many proposals for general approaches to building holonic manufacturing systems. As yet, however, there are comparatively few attempts which have taken manufacturing problems under a dynamic environment into consideration.

Literature review shows that (i) there are a few AGV related practical applications for agent based technologies, showing that there is still a long way to go to spread the holonic systems in this research area, (ii) usually there is no information about the use of a systematic method to structure negotiation mechanism between holons, and there is a small number of works that (iii) consider the learning ability and the adaptability characteristics of AGV agents. Furthermore, although there have been studies conducted by many authors presenting the practicality of reinforcement learning, its employment in larger manufacturing systems is still insufficiently explored.

If there are a number of tasks waiting for the services of an AGV, then the task assignment problem becomes a serious operational decision. A manufacturing system described in the following section with intelligent decisional agents such as products, machines, and automated guided vehicles is created to coordinate pickup-dispatching rule selection of multiple AGVs. The holonic methodology is adapted by a multi agent based simulation model. As a fresh approach to holonic systems, a reinforcement learning methodology is proposed for the distributed control problem of pick-up dispatching of single-load AGVs.

This study proposes a real-time distributed decision-making method based on reinforcement learning for assigning transfer tasks to AGVs considering turbulent environments. In this thesis, a popular reinforcement learning algorithm is applied to a multi-

AGV system's dispatching rule selection problem. This study explores the applicability of Q-learning to a pick-up dispatching rule selection problem.

3.2. The Proposed Approach

In unpredictable and dynamic environments, where it is challenging to foresee upcoming events, agents in distributed SFCA must learn to adapt their actions to those dynamic scenarios. The ability of learning integrates additional intelligence to an agent. Obtaining new knowledge and skills helps agents to take better decisions in the future (Leitão, 2009).

Perhaps, due to its simple algorithms and mathematical grounds, one of the most widely used learning methods in the area of artificial intelligence and machine learning is reinforcement learning (RL). Xue *et al*. (2018) define reinforcement learning as "a model in which an agent learns to select an optimal or near-optimal actions to achieve its long term goals through trial-and-error interactions with dynamic environment". In other words, the learning agent must learn by trial and error how to perform a task in order to get the most reward in the end.

In reinforcement learning methods, an active, exploring agent can perceive information about its environment and therefore can sense the current state of its surroundings. Afterwards, the learning agent selects an action to complete depending on the observed state. This action may transform the state of the environment into another state. In regard to the consequences of the action, the agent is either gets a reward or penalty. Through these interactive relationships amongst the agents and the surroundings, the agents learn a decision-making strategy that yields in the maximum overall reward.

Besides the agent and the environment, Sutton and Barto (1998) identify four main components of a reinforcement learning system: a policy, a reward function, a value function, and a model of the environment. A policy defines the way of action of the learning agent. A policy offers the learning agent particular actions in each state. The goal in a reinforcement learning problem is implied via the reward function. The reward function maps each state-action pair to a corresponding reward or punishment. In a way, the reward

function recompenses good and bad actions for the agent. The agent's objective is to maximize the total reward it collects in the long run. A value function specify which action is desirable in the end. The value of a state is the total amount of reward an agent can expect to collect over the future, starting from that state. While rewards are immediate consequence of actions, the values point out the desirability of states in the long run. The final element in a RL framework is the model of the environment. A model of environment mimics the behavior of the environment. It might predict the resultant state given the present state and a projected action.

Q-learning is a one of the most popular RL algorithms. Watkins introduced the first Q-learning pattern in 1989. The goal of Q-learning algorithm is to learn the state-action pair value, Q(s, a). Q(s, a) is the value function that denotes the expected reward for each state (s) -action (a) pair in the long run. The convergence of learned has been proven (Watkins & Dayan, 1992). When the learning episode is completed, the final Q-values signify the optimal strategy that the learning entity aims to learn.

Recently, because of its ability in finding optimal/near-optimal policies in dynamic environments, RL methods haven applied manufacturing problems. There have been numerous successful studies presenting the effectiveness of RL. Wang and Usher (2005) apply Q-learning to solve the problem dispatching rule selection of a single machine. Proper and Tadepalli (2006) apply reinforcement learning to a product delivery problem which integrates inventory control and vehicle routing aspects of a manufacturing system. Zhang *et al*. (2012) address an unrelated parallel machine-scheduling problem with reinforcement learning method in a dynamic environment. Chen *et al*. (2015) formulate the scheduling problem as a RL problem and develop a Q-learning algorithm based scheduling methodology. Xue *et al*. (2018) study a multi-AGV scheduling problem with a reinforcement learning method. They prove that the AGVs can learn optimal or near-optimal solution from the earlier understandings of the system. Herein, this approach is taken a step forward the learning ability and also the adaptability characteristics of AGV agents to dynamic scenarios is investigated.

4. SYSTEM DESCRIPTION

In the context of this thesis, a multi-agent based approach for distributed artificial intelligence is proposed to generate effective and adaptive control mechanisms for the management of dynamic processes in a realistic manufacturing testbed. Herein, intelligent agents within a manufacturing system such as products, machines, and automated guided vehicles create a self-controlling network to manage the pickup-dispatching problem of multiple single-load AGVs. This mid-level shop floor problem is addressed with a reinforcement learning method. The application potential of Q-learning, a broadly utilized reinforcement learning method, to a pickup-dispatching rule selection problem is investigated. A hypothetical manufacturing system is built to serve as an infrastructure for this specific research topic. A distributed manufacturing system that is in need of real-time flexibility, online decision-making capabilities and adaptation competencies is described in this section. The designed system fits well with the concepts of holons and agents.

There are machining centers in the manufacturing system that are combined with automated material handling systems and an Automated Storage and Retrieval System (AS/RS). Products enter and leave the system at the AS/RS. There is no constraint on the capacity of AS/RS input and output buffer.

When an order enters the manufacturing system, it is associated with a product. An AGV searching for a task to perform initiates an auction. Products requiring a transfer participate in the negotiation by providing a bid involving relevant information. AGV evaluates the bid by means of the pick-up dispatching rule and selects a product to pick-up. After evaluation the processing requirements of the product it is about to transfer, the AGV starts another auction. The AGV broadcasts the message for the request of bids from the machine agents. Machine reply with relevant information on their capabilities of processing. After evaluating the offers based on the parameters determined by the delivery-dispatching rule, the AGV makes a decision. It sends a confirmation message to the machine that is selected and it delivers the product. The interactions among different holons are shown in Figure 4.1. Sequential negations of the AGVs take place.

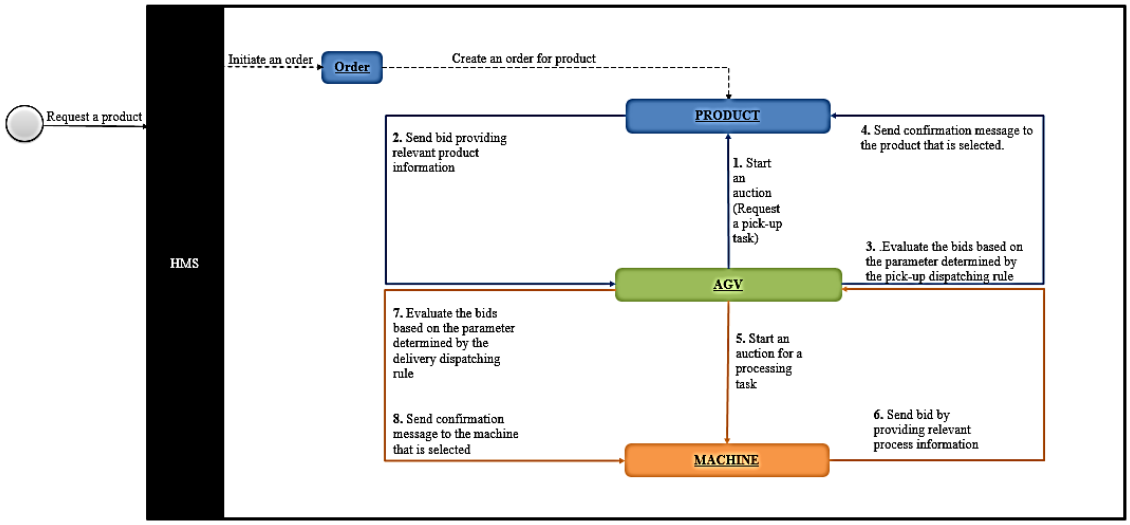

Figure 4.1. Interaction between agents to complete a job.

Products can be categorized into several types and each may have different processing requirements. Probability of the type of the product is given by a distribution. A system manager defines the operation requirement of each type of product. An operation can be performed by several machines. Operation times are fixed. It takes a certain “process time” to perform an operation on a certain machine and the operation time depends on the machine and the product type. There is no assembly in the system. There is nothing that comes into the system as a purchased product (semi-product). All of the products are produced in the system.

Several machines can perform the same operation but the cost (or the time) may be different. The products prefer to stay on the same machines to complete their route instead of travelling to another work center. The number of machines and their corresponding operation capabilities are changeable. Machine setup times are not considered. There are input and output buffers to avoid blocking and starvation. However, their capacity is limited. The products are transferred to the output buffer by a robot after processing. They wait for the AGV at the output buffer. If there is not enough place at the output buffer for a machine to unload, it is considered blocked. The time wasted due to blockage is tracked. The machine can unload the product only if an AGV transfers a product from its output buffer and a space becomes available. They are available to make a bid for the services of an AGV immediately after being finished at the machine. A product has to wait at the input buffer until the machine

is ready to process. The products do not question the availability of the robot, solely the availability of the machine is important.

Products flow one by one. Automated guided vehicles accomplish the material transfers among the workstations (including trips from/to AS/RS) with single load capacity. AGVs have a predefined path. Path network is predefined and the vehicles can move only in one direction. The velocity of the AGVs is fixed and constant.

On the path, first come the delivery points and then the pick-up points. AGVs transfer the products from the output buffer of a machine to the next machine's input buffer or directly to the AS/RS. A robot in each station performs AGV loading /unloading. The method of load transfer is off-track. Every machine has an off-track loading/unloading spot for AGVs. AGV initiates the matching of the work piece. There are several pick-up and delivery dispatching rules defined. Vehicles choose the shortest path to pick-up and deliver products. An AGV has the ability to sense its and others' location and avoid collisions.

There is a specific time requirement for loading and unloading. If there is no available space for an AGV to unload a product to a machine's input buffer, then the product either has to wait on the AGV for a spot to become available or can be delivered to another machine that can perform the same operation it requires and if there is no machine that is capable of performing that operation, the semi-product is transfer to the AS/RS. The later strategy prevents deadlock situations in front of machines' loading/unloading area. Nevertheless, handling of such cases is user-dependent. Every loop has a parking station in which idle AGVs can stay. If an AGV is idle, it must stay in the parking area unless it receives a bid from one or more products. There is no time lost due to recharging AGVs.

It is possible to keep track of the performance measures listed on Table 4.1. The statistics are updated in real-time.

Table 4.1. The key performance indicators of the system.

System	Machine	AGV
-Throughput -Cycle time -Manufacturing lead time -Manufacturing lead time of each type of product -Work in process -Number of product in negotiation -Number of products accumulating in ASRS -Travelling time of products -Total process time of a product -Average total process time (for all products) -Waiting time of products at input buffers -Waiting time of products at output buffers -The average waiting time of parts -The average number of jobs waiting for an AGV	-Idle time -Processing time -Machine utilization -Work load -Utilization balance -Number of products at input buffer / Input buffer capacity -Number of products at output buffer / Output buffer capacity -Average buffer utilization rate -Number of blockages -Time wasted due to blockages at output buffer	-Idle time -Empty travel time -Loaded travel time -Waiting time due to blockage at the arrival station -Utilization balance

Netlogo allows a relatively easy solution for prototyping the specified manufacturing system. The implementation of holonic manufacturing system on Netlogo is described in the following sections. Three basic holons, namely product, machine (including AS/RS) and AGV holons are defined for the model factory.

4.1. Product Holon

In the implemented architecture, product holon holds the product and process requirement knowledge. A product holon represents a specific product instance on the shop floor. Product holons have the necessary information about product characteristics such as part ID, type, operation requirements, location and all of the aforementioned product related measures.

The state of products in the system are defined by two main characteristics; the location of the product and the processing progress. The transition of the location of the products are illustrated on Figure 4.2. The transition diagram involves the functions triggering a state

change. The states are displayed by circles and the arrows indicate the state transition functions. A product enters the system from the input buffer of the AS/RS. The product is transfer between machines until all processing requirement are completed. The finished product is then transfer to the AS/RS.

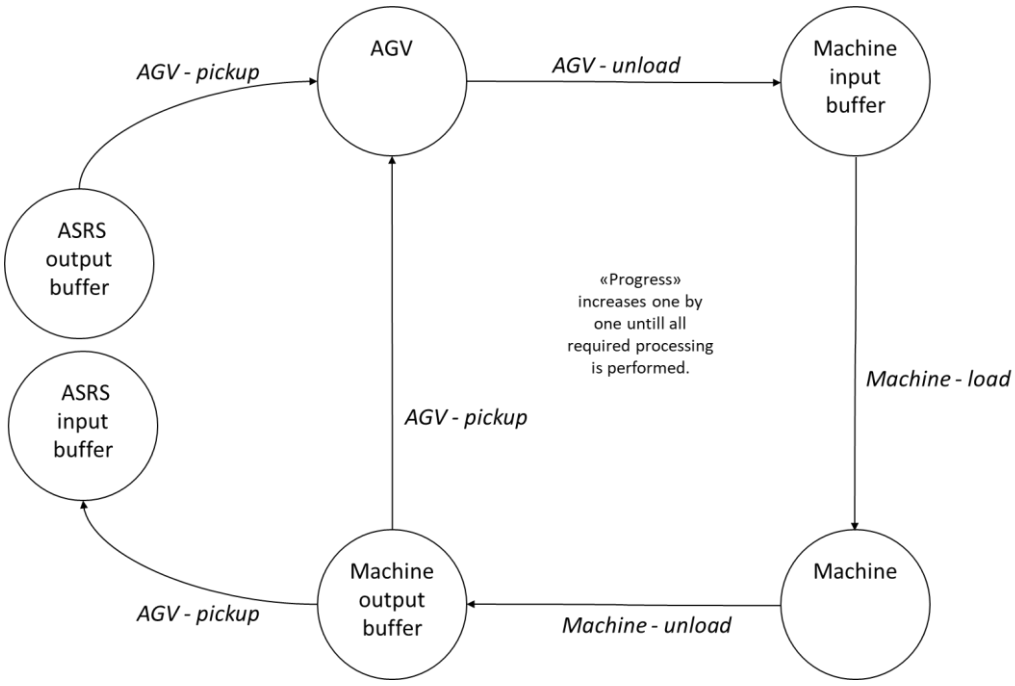

Figure 4.2. Product holon location transition diagram.

4.2. Machine Holon

A machine holon has knowledge about its capabilities and current state. The main responsibility of the machines is to contribute to production by means of processing products. Machine holons hold information about machine characteristics such as machine ID, location, status (processing, idle, blocked, etc.), processing capabilities and times, parts waiting in buffers, buffer capacity, expected machining time of products, and expected total waiting time of products in the input buffer.

The state of a machine holon depends on four criteria. The states and the corresponding operations the machine holon performs during those states are presented in a tabular format on Table 4.2.

A machine becomes blocked when its output buffer is full and cannot unload the product it currently finished processing. The machine remains blocked by the output buffer's capacity limitation until an AGV arrives, picks up a job and a spot at the output buffer becomes available. This blockage result in a waste of value-added time.

Table 4.2. The states and the corresponding operations of the machine holon.

	State determination criteria				
State-No	Is there at least one product at the input buffer?	Is there a product on the machine?	Is the machine processing?	Is the output buffer full?	Task of the machine holon
0	Yes	Yes	Yes	Yes	Process product
1	Yes	Yes	Yes	No	Process product
2	Yes	Yes	No	Yes	Blocked
3	Yes	Yes	No	No	Unload product
4	Yes	No	Yes	Yes	Dummy state
5	Yes	No	Yes	No	Dummy state
6	Yes	No	No	Yes	Load product
7	Yes	No	No	No	Load product
8	No	Yes	Yes	Yes	Process product
9	No	Yes	Yes	No	Process product
10	No	Yes	No	Yes	Blocked
11	No	Yes	No	No	Unload product
12	No	No	Yes	Yes	Dummy state
13	No	No	Yes	No	Dummy state
14	No	No	No	Yes	Idle
15	No	No	No	No	Idle

4.3. AGV Holon

AGV holons contribute to production by material handling. They possess information about their own capacity, velocity, location, loading/unloading time. In this implementation, AGV holons can be considered as information servers for the other holons. They manage negotiations with products and resources (machines).

The state transition diagram for an AGV holon is given on Figure 4.3. If an AGV becomes idle, it requires a task to perform and initiates a pick-up. It selects a proper product based on a pick-up dispatching rule, arrives to the target station, picks up the load and transfers the product to a machine selected by a delivery dispatch criterion.

	State determination criteria	
State-No	Is the AGV arrived to its target desitonation?	Is the AGV full or empty?
0	No	Full
1	Yes	Full
2	No	Empty
3	Yes	Empty

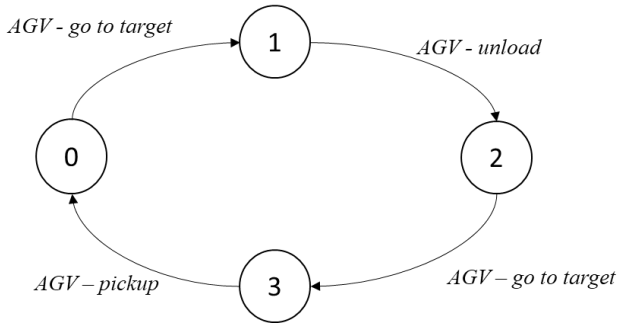

Figure 4.3. AGV holon state transition diagram.

The contracting mechanism between AGVs and products is shown on Figure 4.4. The mechanism does not allow for bargaining between AGVs and bidders. A single-load AGV uses one of the following pick-up dispatching rules while evaluating bids of products requesting a transfer. Fifteen vehicle-initiated pickup-dispatching rules are studied here. Some of them are taken from the literature. In addition, some newly proposed rule are integrated. Among these pickup-dispatching rules, some are distance-based (time-based) rules and some are workload-based rules. The followings introduce the pickup-dispatching rules integrated to the simulation model.

- Random: AGV picks a random load to pick-up.
- STTF: Every load sending a bid includes its location information. AGV calculates the distance from its current position and picks the load with the shortest distance. If there are multiple loads having the same distance (at the output buffer of the same machine), the AGV picks a random load among those.

Table 4.3. The pickup-dispatching rules integrated to the simulation model.

Pick-up dispacthing rules	Acronym
Random	Random
Shortest time to travel first	STTF
Shortest time to travel first - Shortest remaining process time	STTF-SRPT
Shortest time to travel first - Greatest waiting time	STTF-GWT
Shortest time to travel first - Longest time in system	STTF-LTS
First come first served	FCFS
Longest time in system	LTS
Longest Time to Travel First	LTTF
Longest remaining processing time	LRPT
Shortest remaining processing time	SRPT
Greatest queue length - Shortest remaining process time	GQL-SRPT
Greatest queue length - Greatest waiting time	GQL-GWT
Greatest queue length - Longest time in system	GQL-LTS
(Machine prioritizing) Greatest queue length - Shortest remaining process time	M-GQL-SRPT
(Machine prioritizing) Greatest queue length - Greatest waiting time	M-GQL-GWT
(Machine prioritizing) Greatest queue length - Longest time in system	M-GQL-LTS

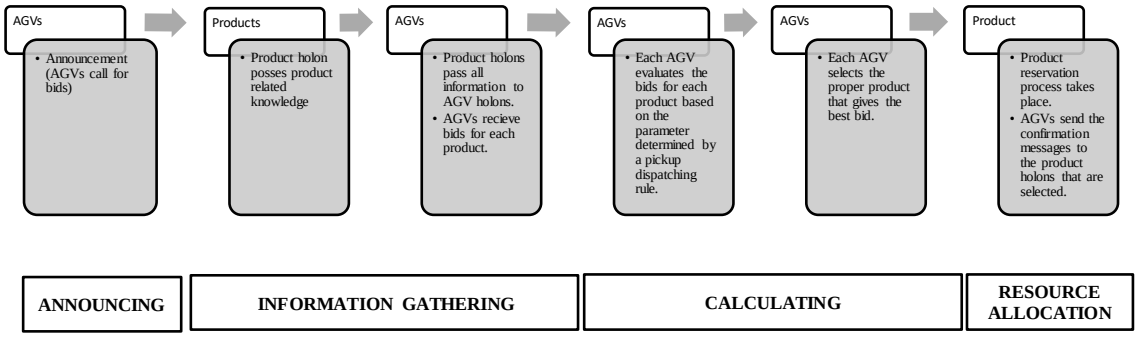

Figure 4.4. The contracting mechanism between AGVs and products.

- STTF-SRPT: Every load sending a bid includes its location information. For each product, AGV calculates the distance from its current position and picks the load with the shortest distance. If there are multiple loads having the same distance (at the output buffer of the same machine), the AGV picks the product with shortest remaining processing time.
- STTF-LTS: The bids involve the location information of the products. For each product, AGV calculates the distance from its current. It there are multiple products at the shortest distance, AGV sorts the products based on the total time they spent in the system. It selects the job, which has been in the system for the longest time among the products waiting at the closest destination.

- FCFS: The bids of the products include the exact moment when they requested an AGV. AGV picks the product, which requested a transfer first.
- LTS: AGV requests the time-in-system of every load currently sending a bid. AGV selects the load with the greatest time-in-system.
- LTTF: The bids involve the location information of the products. For every load, AGV calculates the distance from its current position and picks the load with the longest distance. If there are multiple loads having the same distance (at the output buffer of the same machine), the AGV picks a random load among those.
- LRPT: The bids involve the current progress of the products and all of the processing information. For every load, currently participating in the negotiation, AGV calculates its remaining processing time. It selects the load with the longest remaining processing time as the load to be picked up.
- SRPT: For every load, currently participating in the negotiation, AGV calculates its remaining processing time. It selects the load with the shortest remaining processing time as the load to be picked up.
- GQL-SRPT: For every pickup point that can be visited next (i.e., pickup points contain one or more loads that need to be picked up), AGV checks the number of products waiting at the output queue of pickup points. From all possible pickup points, AGV selects the product with the shortest remaining process time that waits at the pickup point that has the great number of loads waiting at its output queue.
- GQL-GWT: AGV selects the product with the greatest waiting time that waits at the pickup point that has the great number of loads waiting at its output queue.
- GQL-LTS: AGV acknowledges the pickup point whit the greatest output-queue size. AGV selects the product with the greatest time-in-system, which waits at that pickup point.

Machine prioritizing queue length based rules determine a pickup station based on the procedure described below.

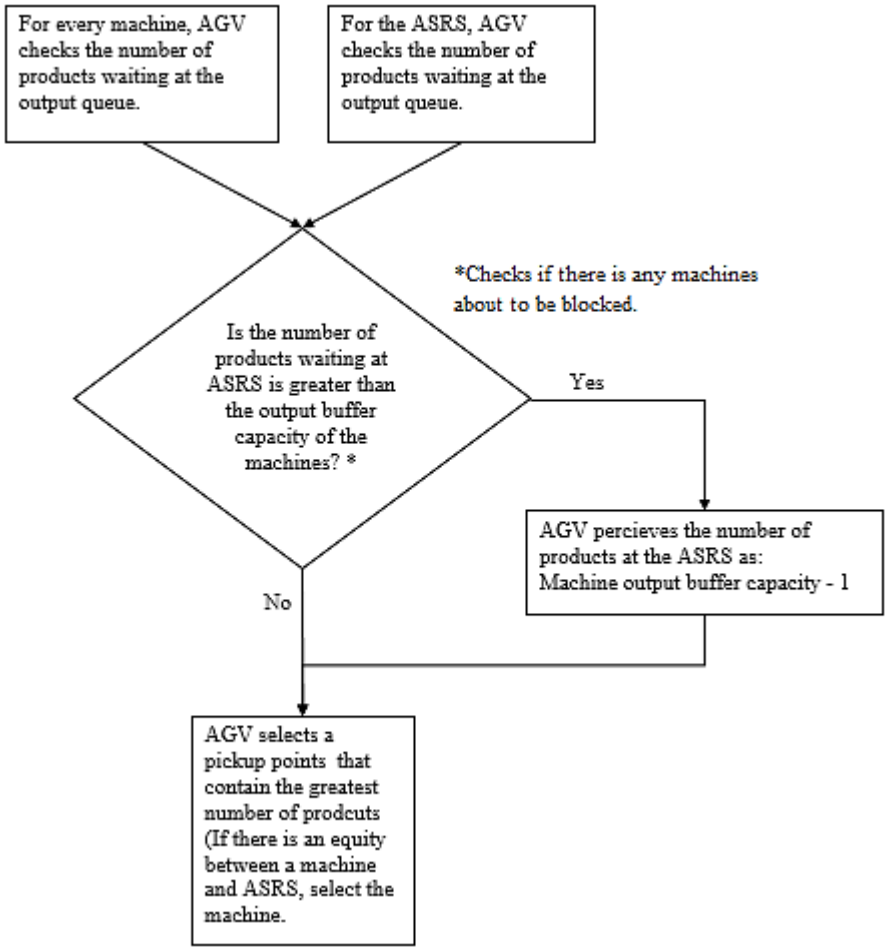

Figure 4.5. Procedure for machine prioritizing queue length based rules

- M-GQL-SRPT. From the selected pickup points, AGV selects the product with the shortest remaining process time.
- M-GQL-GWT: From the selected pickup points, AGV selects the product with the greatest waiting time.
- M-GQL-LTS: From the selected pickup points, AGV selects the product with the greatest time-in-system.

The delivery point of the selected load is determined by a delivery-dispatching rule. AGV is able to use one of the following delivery dispatching rules while evaluating bids of the machine offering its services.

Table 4.4. The delivery-dispatching rules integrated to the simulation model

Delivery dispacthing rules	Acronym
Longest elapsed time since last arrival	LET
Process time	PT
Process and travel time	PTT
Expected finish time	EFT
Smallest input queue	SIQ

- LET: AGV collects bids from machines that are capable of processing the next operation of the product currently on itself. The bids involve the amount of time that has elapsed since the last load arrived to each machine. From those delivery points, AGV selects the point, which has the longest time elapsed.
- PT: The collected bids involve solely the amount of time that is needed to process the product. From the possible delivery points, AGV selects the point that offers the smallest process time.
- PTT: The collected bids comprise of the amount of time that is needed to process the product and the travel time to reach the destination. The travel time is parameterized based on the layout. From the possible delivery points, AGV selects the point, which offers the smallest total process and travel time.
- EFT: Waiting time, process time and travel time are the three input variables of this rule. Waiting time refers to the product's waiting time in the input buffer of a candidate machine. The bids of the machines include that is needed to process the product, the travel time to reach the destination, and additionally, the time it has to wait in the input buffer. From those delivery points, AGV selects the point, which has shortest expected finish time.
- SIQ: AGV selects the point, which has the shortest input queue.

4.4. Implementation of Q-Learning

Due to its simple algorithm and mathematical grounds, Q-learning has been chosen as a suitable solution to solve the online AGV dispatching problem. A system transition epoch is constrained by the action of AGV unloading a job onto a machine's input buffer and it is ready for picking up another product. From this instant, the next act is chosen. The AGV

travels to the selected product. Epochs follow each other as a sequence. The learning episode is time terminated. The Q-learning algorithm followed by AGVs is given in Figure 4.6.

Xue *et al*. (2018) state that learning agents utilizing the same reward function ought to share the same Q-matrix. As a result, the Q values convergence rapidly. Otherwise, with limited learning experience, the Q-values may not reach convergence and the finest solution of the AGV dispatching problem may not be obtained. Thus, the experience gained by different AGVs are shared with each other in the course of this study.

In general, the difficulty in applying Q-learning procedure to manufacturing systems essentially lies in formulating manufacturing problems as RL problems, which involves specification of the system state space, the members of the action space, and the types of the reward functions. The defined system state space, members of action space, types of reward functions and Q-learning function are explained in the following sections.

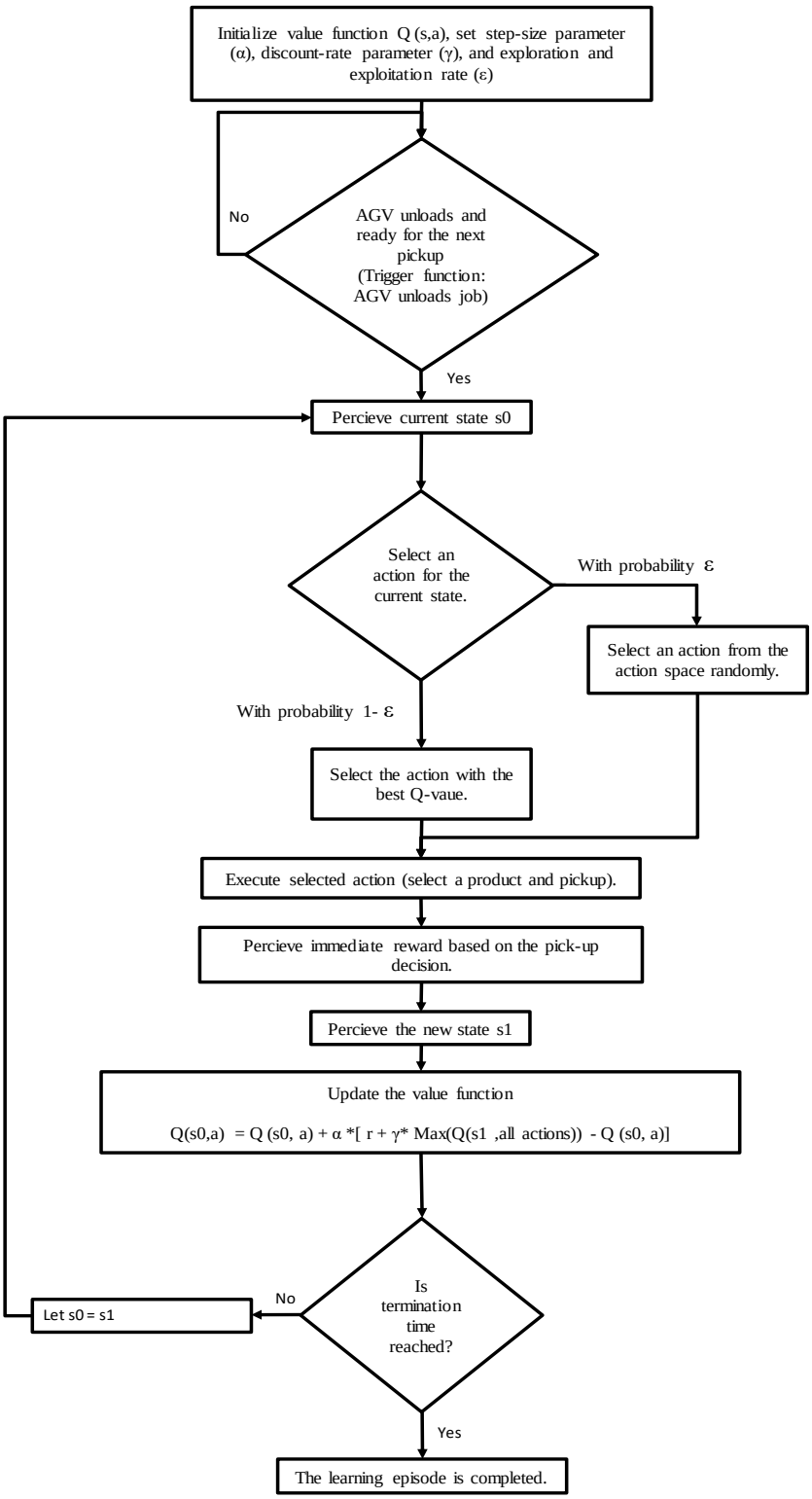

Figure 4.6. Q-learning algorithm followed by AGVs.

4.4.1. System State Space

Characterization of the state features and the dispatching actions is strongly linked to the dispatching problem by nature. State determination criteria help to represent the system state numerically. State features identify the concerned characteristics of the manufacturing environment. The system states are generalized and represented by the state features. Zhang *et al*. (2012) point out that state spaces should be kept brief and compressed for a more efficient learning environment. They also add that state determination criteria should be selected considering the convenience in computing.

The system state space is defined by the conditions of the output buffers of the machines. In the first state, there is at least one machine with a full output buffer. The other state is when every machine has at least one empty spot at their output buffer. In other words, the states correspond to either starvation or blockage in the system. The states determine whether the system needs to be discharged and be relieved from blockages or it needs to be fulfilled with products. Thus, the in-flow to the system and out-flow from the system depends on the system's state. The policy table used in this study is given on Table 4.5. The state features are appropriate for problems of different scales.

4.4.2. Action Space

An action is defined as the AGV selecting a product from a machine's output buffer for transportation. There is a group of alternative actions in the action space. The action selection made by an AGV agent is based on the present state of the environment.

Table 4.5. Policy table

		Action	
State-No	State determination criteria	a0	a1
0	No blocked workstations in the system.	Q(0,0)	Q(0,1)
1	There is at least one blocked workstation.	Q(1,0)	Q(1,1)

4.4.3. Exploration and Exploitation

Exploration enables an AGV agent to try something that has not been done before. On the other hand, exploitation is that an AGV must perform the rules that were selected before. Exploitation may promise a decent reward, on the other hand, exploration may deliver more chances to discover the maximum cumulative reward eventually (Wang & Usher, 2005).

One widespread approach to handle the trade-off between exploration and exploitation is the ε–greedy method. An AGV following the ε –greedy methodology selects a dispatching rule from the action space randomly with probability ε, otherwise, with probability 1- ε, the rule with the best Q-value. To enable exploration and exploitation, ε–greedy method is adopted in the course of this study.

4.4.4. Reward Function

The reward function states the goal of the AGVs. Rewards and penalties are used to determine the value of a pick-up dispatching rule depending on the perceived state. Since the AGV agents attempt to maximize the total reward in the long run, the reward function is essentially used for guiding the AGVs to accomplish the system objective.

In this study, the objectives are (1) minimizing the empty travel time of the AGVs, (2) the wasted machining time due to blockages. The rewards are connected with the empty travel time of the AGVs, and the wasted machining time due to blockages. An AGV's reward is calculated at each transition point; AGV gets the reward immediately after a pickup task. There are several different reward functions defined. They are described below.

In Case A, the travel time of an empty AGV until it reaches to the target product is defined as "empty travel time". The objective is to minimize the total empty travel time of the automated guided vehicle system. Each AGV records the empty travel time for each performed pick-up. Those records are then registered to a list where all the empty travel times of all AGVs are kept. The statistics are gathered continuously and online. When a pick-up is performed, the time of this specific empty travel is compared to the previously performed pick-ups. The average and the standard deviation of all the empty travel times up

to that exact time are calculated by the AGV. The reward is based on a value that shows how many standard deviations away from the average, this particular empty travel time is.

If there is not available spot at the output buffer for a machine to unload, it is considered blocked. The time wasted due to blockage is considered as "wasted machining time due to blockage". The study focuses on this machine-related performance measure in Case B. If an AGV solves a blockage by picking up a load from a blocked machines' output buffer, then it receives a "block-solving reward". If there is a machine that is blocked and the AGV disregards that situation and chooses not to serve that machine, then the AGV gets a penalty.

In case C, the above mentioned two objectives are combined. It is expected that the AGVs learn to minimize their empty travel time while figuring out how to handle machine blockages.

Table 4.6. Reward functions.

	Reward type -A	**Reward type -B**	**Reward type -C**
Reward type	**Total empty travel time of AGVs**	**Total wasted machining time due to blockage**	**Total empty travel time of AGVs & Total wasted machining time due to blockage**

Reward function	Reward = ((Average of all empty travel times) - (Empty travel time of this station)) / (Standard deviation of all empty travel times)	If (There is at least one blocked machine in the system and the AGV picks up a load from a blocked machine)
		Reward = 2 * *Block-solving reward (constant)*
		
		If (There is at least one blocked machine in the system and the AGV does not pick up a load from a blocked machine)
		Reward = -1 * *Block-solving reward*
		
		Otherwise, Reward = 0
			If (There is at least one blocked machine in the system and the AGV picks up a load from a blocked machine)
			Reward = ((Average of all empty travel times) - (Empty travel time of this station)) / (Standard deviation of all empty travel times) + *Block-solving reward*
			
			If (There is at least one blocked machine in the system and the AGV does not pick up a load from a blocked machine)
			Reward = ((Average of all empty travel times) - (Empty travel time of this station)) / (Standard deviation of all empty travel times) – Block-solving reward
			
			Otherwise,
			Reward = ((Average of all empty travel times) - (Empty travel time of this station)) / (Standard deviation of all empty travel times)

4.5. Assumptions

While building the simulation model the following assumptions are made for clarity and simplicity:

- The number of AGVs is known, and the layout of guide paths and the location of the machines has been determined.
- All guide paths are unidirectional.
- If there is no available space for the AGV to unload a product to the selected machine's input buffer, the AGV delivers the product to an alternative machine and if there is no machine that is capable of performing that operation, the semi-product is transfer to the AS/RS. This strategy prevents deadlock situations in front of machines' loading/unloading area.
- AGV and machine breakdown possibility is integrated into the model. However, during experimentation vehicles and machines operate continuously without any breakdowns.

5. EXPERIMENTATION

The aim of this study is twofold. First, the study is intended to determine if an AGV agent is able to learn the best pick-up dispatching rule in each occasion. This is experimentally investigated by a multi-agent simulation model to establish if the learning AGV agents are able to learn to employ the best pick-up dispatching rule under various circumstances. The second objective of this thesis is to demonstrate the feasibility of adaptive capabilities of learning AGVs by studying the agents' ability of switching to another rule when the initially adopted rule starts to fail.

To evaluate the performance of the proposed Q-learning method, experimentations are made on the built simulation model. First, the best pick-up dispatching rule that boosts each performance measure has been identified. Those results are taken as reference points for the later experiments. Second set of experimentations are conducted to assess the performance of the learning AGV agents using the Q-learning procedure. Tests are carried out to investigate if AGVs are capable to learn to practice the best rule as specified in the static pick-up dispatching problem. The final set of experiments are intended to demonstrate that the learning AGV can adapt itself to the changes in the environment and can learn to practice the best action in a given state.

The hypothetical manufacturing system is developed in a modular and extendible fashion. Problems of different scales are created to test the effect of pickup dispatching rules not only on smaller test beds, but on large-scale real-world cases, too. A study of today's flexible manufacturing systems points out that no more than 20 workstations capable of executing a variety of operations will be in the factories of the future (Mahalingam, 2014). To be able to reflect large-scale real-world manufacturing environments, a maximum of 20 machines were considered in the largest system. Simulation tests are conducted on 3 different scenarios (Table 5.1). The load of the system is determined by the inter-arrival times of the products. The inter-arrival times of products coming into the system follow an exponential distribution.

Table 5.1. Simulation scenarios.

Setup characteristics	Scenarios		
	Scenario-1	Scenario-2	Scenario-3
Number of loops	1	2	4
Number of AGVs	1	4	7
Number of machines	4	10	20
Inter-arrival time of products (Mean)	100 Ticks	100 Ticks	100 Ticks
%Product type#1	50%	20%	20%
%Product type#2	50%	20%	20%
%Product type#3		20%	20%
%Product type#4		20%	20%
%Product type#5		20%	20%

A representation of the full manufacturing environment is given on Figure 5.1. The model factory configured on Netlogo is given on Figure 5.2.

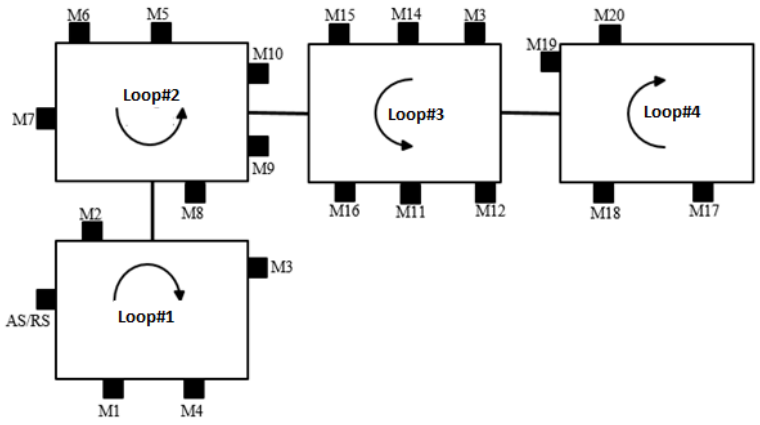

Figure 5.1. Representation of the manufacturing environment.

5 types of product are randomly generated. The required operations and the sequence of operations are randomly created. The required operations sequence for each type of product is presented on Table 5.2. Randomly generated process time multipliers are used to determine the product-specific machining times. The operation time of a product for a certain operation on a certain machine is calculated as follows: (Process time multiplier of the

product type) x (Machining time for that operation). The operation capabilities and machining times of the machine holons are given on Table 5.3.

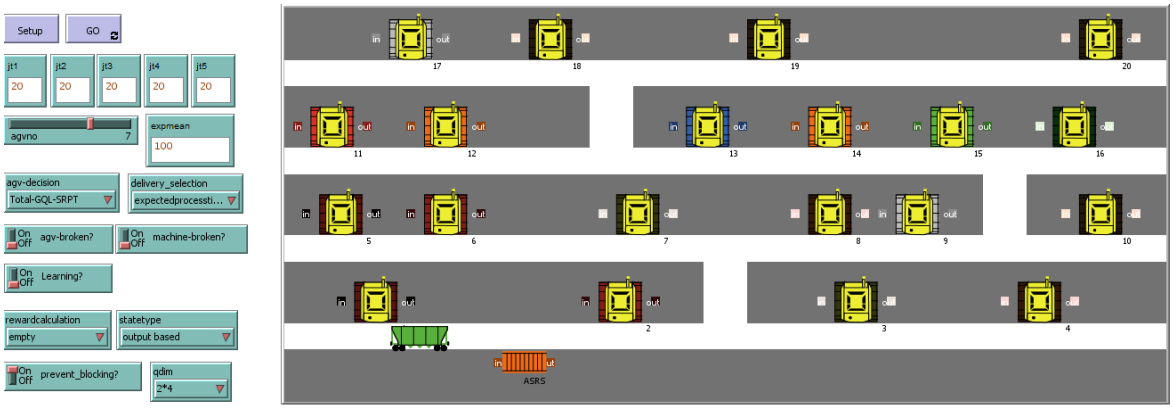

Figure 5.2. Pictorial representation of the model factory.

Table 5.2. Operation sequence and process time multipliers for each type of product.

Product type	Required operations					Process time multiplier				
Type#1	O8	O9	O1	O3	O6	0.7	1.2	0.8	1.1	0.9
Type#2	O6	O4	O5	O8	O9	0.8	1.3	0.9	0.7	1.2
Type#3	O7	O5	O6	O3	O1	0.9	1.1	1.1	1.3	0.9
Type#4	O7	O10	O2	O3	O5	0.7	1.2	0.8	1.1	0.9
Type#5	Q2	O9	O8	O10	O4	0.9	1.3	0.7	0.7	1.1

Through computational experiments, this study explores the capabilities of Q-learning algorithm in the context of a pick-up dispatching rule selection problem. It investigates the Q-learning applicability to determine if it can be used to enable automated guided vehicle agents to learn universally acknowledged pick-up dispatching rules for several reference instances.

Table 5.3. The operation capabilities and machining times of the machine holons.

Machine no	Location	Operation capabilities and process times (Ticks)									
	Lane no	O1	O2	O3	O4	O5	O6	O7	O8	O9	O10
M1	Loop#1	505								450	
M2	Loop#1		500				740				
M3	Loop#1			420							
M4	Loop#1				530			545			
M5	Loop#2		550				600				
M6	Loop#2			640			530	350			
M7	Loop#2		760		480			580			
M8	Loop#2	800								500	
M9	Loop#2					465	500	600			
M10	Loop#2								405		375
M11	Loop#3	500		600						1000	
M12	Loop#3		600			450			500		
M13	Loop#3			550							
M14	Loop#3										600
M15	Loop#3		635		650				570		
M16	Loop#3							450			
M17	Loop#4					460					
M18	Loop#4	565		485					520		
M19	Loop#4						350				
M20	Loop#4							360			
AS/RS	Loop#1	-	-	-	-	-	-	-	-	-	-

5.1. Evaluation of Pick-Up Dispatching Rules

The dispatching decisions on the operational level can have an impact on material flow, buffer storage requirements at the workstations, resource utilization and effectiveness (Egbelu & Tanchoco, 1984).

AGVS' performance is often evaluated by measuring both indicators concerning the AGVS operation and performance parameters representing the entire manufacturing system operation. However, production system measures are, in fact, also tied to other parameters influencing the manufacturing system. For instance, throughput is affected not only by AGVS operation parameters but also by order load and plant layout (Liu *et al*., 2008). This complexity makes the comparison between different applications problematic. Analysis and evaluation of AGVS would be lacking without a profound understanding of the inter-dependencies of the performance measures. Jain and Raj (2016) conduct a comprehensive literature review and identify fifteen performance measures, which are related to the performance of manufacturing systems. The developed model in their paper provides an opportunity to understand the dependent and the independent performance indicators. Their

analysis reveal that manufacturing lead-time, throughput time, and reduced work in process inventory have strong dependence on each other as well as drive power. Any alteration on such variables will have an effect on the others and also on themselves. Dependent performance parameters are unit manufacturing cost, unit labor cost, equipment utilization, ability of manufacturing of a variety of product, and capacity to handle new product. They tend to depend on one another strongly. Effect of tool life, rework percentage, set-up time, scrap percentage, automation, and use of automated material handling devices are identified as independent variables. Therefore, industry experts and the academia treat those parameters as the origins of all the measures.

Production measures used for AGVS assessment has been discussed by a great number of authors in literature and several studies have been conducted to assess the performance of dispatching rules. However, perhaps, one of the most significant results obtained is that in conventional AGVS, the dispatching policy does not have an influence on mean flow time, machine utilization, AGV utilization, or maximum input buffer queue length; it only affects the length of output buffer queues (Liu *et al*., 2008).

There is a wide choice of parameters by which AGVs are assigned to tasks. The selection of the parameters for AGV dispatching system is reasonably challenging, and the decision can radically affect the cost and performance of the system. Le-Anh and De Koster's review (2006) reveals that common objectives of the studies in this field are diminishing waiting time, maximizing throughput of the system, or reducing queue length. Each dispatching rule seems to promote a different performance measure. However, an increase in one asset may decrease another asset. Altogether, the ultimate objective is to sustain an effective system through minimizing inefficiencies.

Distance based rules are very common and frequently seen in literature. According to Shortest-Time-to-Travel-First, the vehicle is sent to the closest load to requesting an AGV. Amongst single attribute dispatching rules, STTF tends to have a good performance in many environments, mainly where capacities of the buffer queues are not limited (De Koster *et al*., 2004). STTF rule aims to diminish the time vehicles spend for empty travel. However, they note that this rule is highly sensitive to the layout of the manufacturing site. There might be

some pick-up points that may never qualify to receive a vehicle due to poor layout design. Since most of the delivery tasks will take place between the close work centers, the input queue may grow to its limit. Thus, other vehicles will then be blocked from making deliveries to that machining center (Egbelu & Tanchoco, 1984).

Workload-based online dispatching rules emphasize the limitations of queue sizes. In Egbelu and Tanchoco (1984), queue size based rules are introduced. The aim of such rules is to reduce the chance of the output buffer's queue overflowing and consequently to lessen the risk of workstation blocking.

In the light of these findings above, simulation tests are performed to assess the effects of employing different pick-up dispatching algorithms. Several distance-based rules (STTF, SD-SRPT, SD-LTS, SD-GWT, LTTF), a number of workload based rules (M-GQL-SRPT, M-GQL-GWT, M-GQL-LTS), a time-based rule (FCFS), and a random pick-up rule are compared over (1) the total empty travel time of the AGVs, and (2) the total wasted machining time due to blockages. As mentioned in previous sections, since the delivery-dispatching problem is not the focus of this study, the delivery-dispatching rule is fixed as EFT. Consequently, the best pick-up dispatching rule for each case is identified.

The simulation ran for 22500 ticks (universal Netlogo time) from which the first 2500 ticks is treated as a warm-up period. During experimentations, 1 tick has been considered as 0.5 minutes. Therefore, 20000 ticks of operation result in 7 full working days. For each case, 50 replications were executed.

Table 5.4, Table 5.5, and Table 5.6 provide the outcomes when each of the listed rules are used as the only rule employed for each of the two objectives. The results are obtained by taking the average of the 50 replications of each case. It can be observed that for all scenarios, the shortest distance based rule provide the best performance considering the total empty travel time of the AGVs. SD-SRPT performed best for the first two scenarios. The result in Table 5.6 is closely parallel to the results in Table 5.4 and Table 5.5. A t-test showed that SD-LTS and SD-SRPT are statistically indifferent in the 3rd scenario. M-GQL-GWT and M-GQL-LTS are also statistically indifferent in the last scenario. Random rule has an

average outcome concerning the first performance indicator since the AGVs' utilization is high for all cases. It is likely that AGVs visit the nearest or a considerably close station frequently. Eventually, the results are similar to the previous researches on this topic: Shortest distance based rules diminished the time vehicles spend for empty travel (Egbelu & Tanchoco, 1984).

Table 5.4. Results of practicing the individual pick-up dispatching rules (Scenario-1).

Pickup dispatching rule	Total empty travel time of AGVs (Ticks)	Total wasted machining time due to blockage (Ticks)
SD-SRPT	2619	11932
SD-LTS	2681	11874
SD-GWT	2715	11574
Shortest travel distance	2731	11542
Random	2998	46976
First-come-first-served	3660	48092
M-GQL-LTS	8783	636
M-GQL-GWT	8814	675
M-GQL-SRPT	8867	680
Longest travel distance	10983	12003

Table 5.5. Results of practicing the individual pick-up dispatching rules (Scenario-2).

Pickup dispatching rule	Total empty travel time of AGVs (Ticks)	Total wasted machining time due to blockage (Ticks)
SD-SRPT	7664	24660
SD-LTS	8178	23352
Shortest travel distance	8598	16154
SD-GWT	8738	14440
Random	19340	125231
First-come-first-served	21132	121599
M-GQL-LTS	31768	152
M-GQL-SRPT	32152	176
M-GQL-GWT	35210	397
Longest travel distance	37137	74368

Table 5.6. Results of practicing the individual pick-up dispatching rules (Scenario-3).

Pickup dispatching rule	Total empty travel time of AGVs (Ticks)	Total wasted machining time due to blockage (Ticks)
SD-LTS	20066	16822
SD-SRPT	20317	17442
SD-GWT	21287	15781
Shortest travel distance	21340	15984
Random	55974	179518
First-come-first-served	58509	175759
M-GQL-GWT	68463	2150
M-GQL-LTS	68498	2134
M-GQL-SRPT	68523	2139
Longest travel distance	71271	219029

The dispatching rules emphasizing the limitations of queue sizes resulted in better performance for the second performance indicator. Considering only the wasted total machining time due to blockages, M- GWL-LTS provided the best performance for all cases. Random rule has one of the worst outcomes concerning the second performance indicator. A speculative reason for this sort of result would be that the AGVs cannot discover the blocked station among many others purely by chance. A solely random methodology has a small likelihood of finding the blocked machine and resolving the blockage. Therefore, the wasted total machining time due to blockages accumulates vastly.

5.2. Reinforcement Learning Based Dispatching Approach

The parameters used during the implementation of the Q learning algorithm are designed as follows: step-size parameter, $\alpha = 0.09$, discount-rate parameter, $\gamma = 0.07$ and exploration and exploitation rate, $\varepsilon = 0.05$. In the entire study, these parameters remain permanent. All the actions in each state are supposed to be an equally valid option. Hence, all of the state-action values, Q (s, a), are initialized to zero.

The simulation ran for 22500 ticks from which the first 2500 ticks is treated as a warm-up period. Since the results of the previously introduced three scenarios are closely parallel regarding empty travel time minimization and blockage time minimization, the tests are conducted based on the largest setup (the 3rd scenario). For each case, 100 replications were executed.

In an attempt to train the AGVs to minimize empty travel time, Reward function A is used. Reward function B is used to guide the AGVs to minimizing the time wasted machining time due to blockage. It is expected that the AGVs learn to minimize their empty travel time while figuring out how to handle machine blockages by using reward function C.

SD-SRPT and M- GWL-LTS are the two rules performing best for the selected performance indicators. These two rules are integrated to the action space. The policy table is given on Table 5.7.

The best pick-up dispatching rule that boosts each performance measure has already been identified in the previous section. For estimating the performance of the learning AGV holons with the Q-learning algorithm, the number of occasions that each pickup-dispatching rule is executed throughout the simulation run is computed and the selection percentages of these rules are calculated. The results are provided on Table 5.8.

Table 5.7. Policy table used to train AGVs to minimize empty travel time and to minimize the wasted machining time.

		Action	
State No	State determination criteria	SD-SRPT (a0)	M-GQL-LTS (a1)
0	No blocked workstations in the system.	Q(0,0)	Q(0,1)
1	There is at least one blocked workstation.	Q(1,0)	Q(1,1)

Table 5.8. Selection percentages of the pick-up dispatching rules.

		Available dispatching rules	
Reward type		**SD-SRPT**	**M-GQL-LTS**
Reward function -A-	Total empty travel time of AGVs	93,0%	7,0%
Reward function -B-	Total wasted machining time due to blockage	49,6%	50,4%
Reward function -C-	Total empty travel time of AGVs & Total wasted machining time due to blockage	55,8%	44,2%

The learning agents finished choosing the SD-SRPT rule 93% of the time for the first case. This result essentially demonstrates that the learning AGV is able to learn to prefer the

execution of the superior rule. For the latter two reward types, the selection percentages of the SD-SRPT rule and the M-GQL-LTS rule are almost equal. The AGVs learn to pick-up the load blocking a machine by the M-GQL-LTS rule when a blockage occurs in the machines' output buffer. The final Q-values represent the optimal policy that the agent learns (Figure 5.3). The AGVs select the action giving the highest Q-value for each state. Similar behavior is observed for the third case (Figure 5.4). The agents learn to use the SD-SRPT rule if there are no blocked workstations in the system (s0). On the other hand, if there is at least one blocked workstation (s1), they apply the M-GQL-LTS rule to maximize their reward.

	Action	
State No	SD-SRPT (a0)	M-GQL-LTS (a1)
s0	-10.1604	-1.9371
s1	-24.0789	-8.1518

Figure 5.3. A sample Q-Matrix at the end of the learning period (Reward function B).

	Action	
State No	SD-SRPT (a0)	M-GQL-LTS (a1)
s0	-3.2487	-11.5543
s1	-24.0441	-6.9645

Figure 5.4. A sample Q-Matrix at the end of the learning period (Reward function C).

These results principally prove that the AGVs are capable of learning to practice the best action (as established in the static pick-up dispatching case). It can be concluded that the selection proportions of the pick-up dispatching rules reveal the relative strength of each action for different key performance indicators to some extent. Figure 5.5, Figure 5.6, and Figure 5.7 illustrate the change of the selection proportions rates of the dispatching rules throughout the learning phase of the AGVs for each case. The graphs' x-axes correspond to time (ticks) and the y-axes show the selection proportions rates of the dispatching rules (a0 and a1).

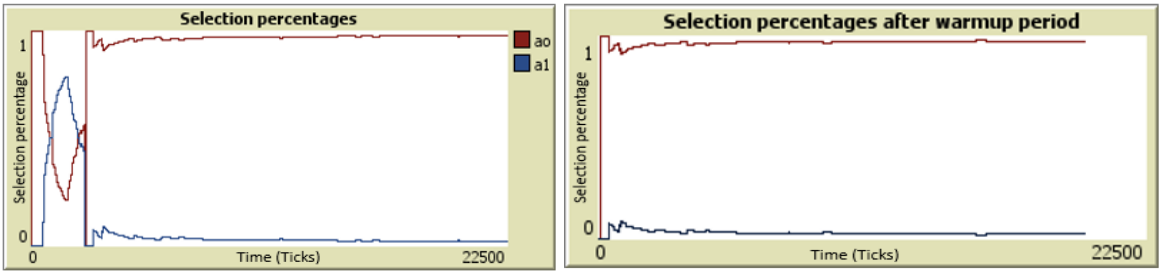

Figure 5.5. Rule selection rates for minimization of empty travel time of AGVs.

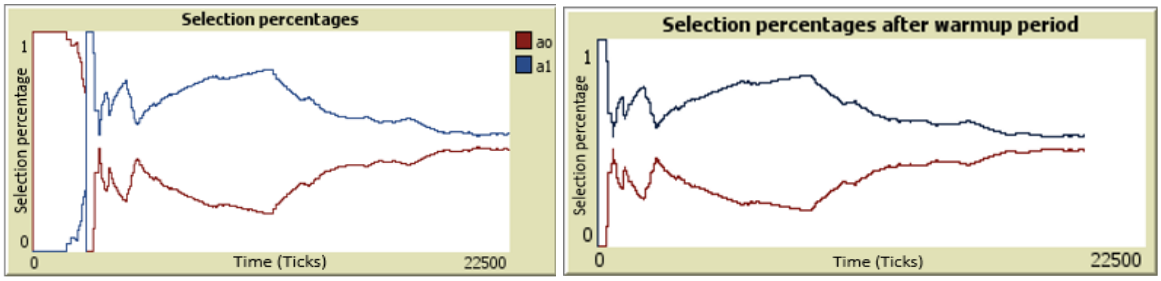

Figure 5.6. Rule selection rates for minimization of total wasted machining time due to blockage.

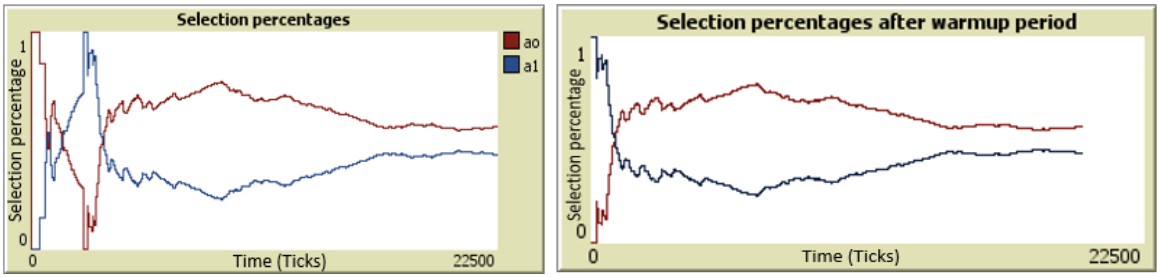

Figure 5.7. Rule selection rates for minimization of total empty travel time of AGVs and total wasted machining time due to blockage.

According to Proper and Tadepalli (2006), RL in practical domains suffers from the curse of dimensionality: explosions in state space and action spaces. Therefore, a sub study is performed with a larger action space where LTTF and FCFS rules are included to the existing action space. The selection percentages of each rule is calculated. Table 5.9 provides the results. In this case, action space complexity increases, state-action space is more complex and difficult to search. However, the learning agents are able to distinguish the best alternative favoring the preset system goal.

Table 5.9. Selection percentages of the pick-up dispatching rules.

Reward type		SD-SRPT	M-GQL-LTS	LTTF	FCFS
Reward function –A-	Total Empty Travel Time of AGVs	92,86%	3,48%	1,81%	3,15%

Up to this point, SD-SRPT has been assessed the best rule favoring the goal of reducing the empty travel time of vehicles. M-GQL-LTS has been proven to minimize the wasted machining time due to blockages. The performance of the static pick-up dispatching rules is compared to the case of learning AGVs on Figure 5.8. The results of both performance measures are given when the AGV's are trained with the previously mentioned 3 types of reward functions.

SD-SRPT works well for the minimization of the empty travel time of the AGVs but its performance is considerably bad regarding the minimization of the wasted machining time. Implementation of Q-learning with the reward considering the empty travel times of the AGVs results in a slightly worse performance in the first objective (compared to the SD-SRPT rule). This outcome could be explained by the nature of the Q-learning algorithm. According to the ε –greedy method, a learning AGV selects a dispatching rule from the action space randomly with a certain probability otherwise it prefers the action with the best Q-value. This exploration mechanism would not let the AGV select the SD-SRPT rule 100%. Therefore, the performance of the Q-learning methodology with reward type A drops slightly. However, implementation of Q-learning with the reward type A yields in a particularly better result in the second objective compared to SD-SRPT.

In the case of the M-GQL-LTS rule the exact opposite situation emerges. This rule yields in an outstandingly high empty travel time while minimizing the wasted machining time. Implementation of Q-learning with the reward type B results in a noticeably better result for the first objective and a slightly worse performance for the second objective (compared to the case of the M-GQL-LTS rule).

Learning agents with the reward type considering both the empty travel times of the AGVs and the wasted machining time due to blockages are able to improve the results of the M-GQL-LTS rule considering the first objective and enhance the performance of the SD-SRPT rule regarding the second objective steeply. Learning AGVs are able to switch

objectives based on the state of the environment in order to find an acceptable compromise between two extremes. In an environment where there are multiple perspectives and/or the opposing interests, employing a learning methodology results in better performance. Learning AGVs are able to find a common ground and hit a happy medium regarding two contradictory goals.

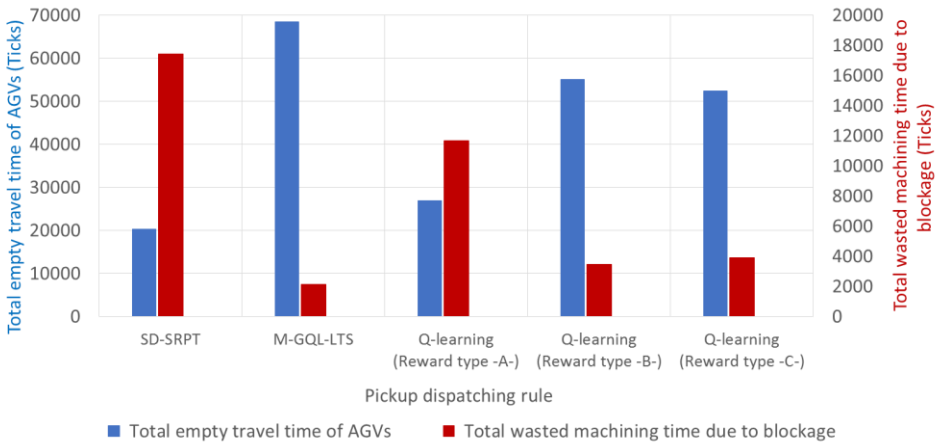

Figure 5.8. Performance of pickup dispatching rules.

5.3. Adaptive Learning Mechanism

In unpredictable and dynamic environments, where it is challenging to foresee upcoming events, agents must learn to adapt their actions to those dynamic scenarios. This study tries to demonstrate the feasibility of adaptive AGVS. It aims to show that learning agents are able to switch to another rule when the initially adopted rule starts to fail due to disturbances.

In this section, a theoretical scenario is implemented. Assume a situation where Action#1 is the best rule favoring a system goal. At a certain point, suddenly, a rapid change occurs in the system such as product specification changes, equipment breakdowns, fluctuations in the customer orders. After that moment, Action#2 performs better and Action#1 performs considerably worse. When the circumstances revert to the initial conditions, Action#1 become more favorable again. In such a case, the need for a system, which is reactive to disturbance on the shop floor, arouses.

Consider a system where the above-mentioned rapid change is represented with a flag. If the flag is 0, the system performs under regular conditions and Action#1 favors the system objective. In case of a disturbance, the flag is set to 1. Instantly, the Action#2 rule enhances the performance of the system. In the simulation model, the flag is initially set to 1, then when the 40^{th} product is finished, this event triggers the flag to set to 0, and finally when the 100^{th} product is finished and comes back to the AS/RS, the flag is set back to 1 again. The simulation ran until 160 products are completed. So in the ideal case, the AGVs would choose Action#1 in the earlier phase and then switch to Action#2 shortly after the 40^{th} product is finished, and finally favor Action#1 again when the 100^{th} product is finished. This hypothetical scenario is tested to see if the agents learn to adapt their actions.

The system objective is determined as minimizing the total empty travel time of the automated guided vehicle system. The reward function A is used. It has been previously proven that STTF outperforms LTTF considering this performance measure. Therefore, the better performing rule is defined as STTF.

Table 5.10. State and policy setup

Conditions	Action#1	Action#2
Flag = 1	STTF	LTTF
Flag = 0	LTTF	STTF
Flag = 1	STTF	LTTF

The state alternation procedure and the action space for this theoretical setup is given on Table 5.10. When the first flag change occurs, the initially adopted rule (Action#1) starts to fail. AGVs need to generate a self-controlling network to switch to the better rule (Action#2).

This experimentation aims to show that learning agents are able to switch from Action#1 to Action#2 and then back to Action#1 again. The selection percentages of Action#1 and Action#2 are calculated. At each change of state, the counters re-set to zero to capture the selection percentage of each rule solely during a certain state. The setup characteristics of Scenario-3 is used for experimentation.

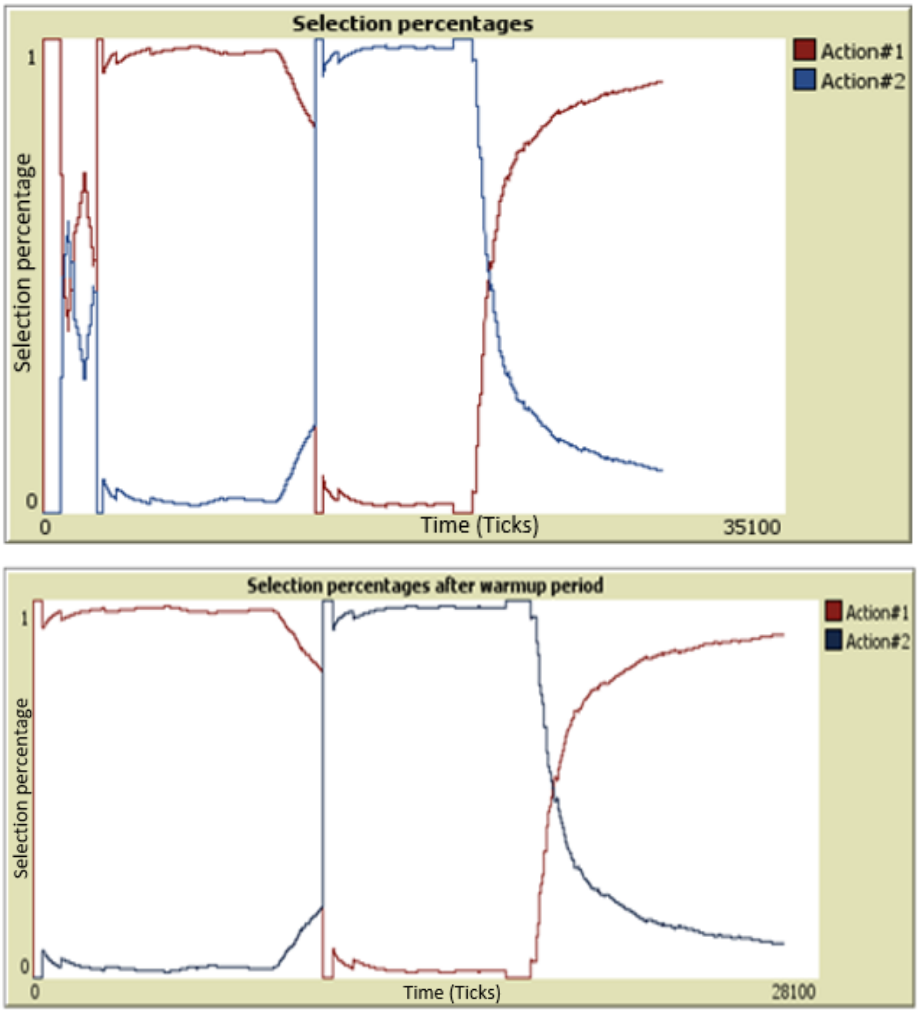

Figure 5.9. Selection percentages of actions.

The change of the selection proportions rates of the dispatching rules throughout the experimentation is calculated. A sample run's results are represented on Table 5.9. The results demonstrate the feasibility of adaptability of learning AGVs. The AGVs start to use Action#2 more intensely when they realize Action#1 starts to result in a bad performance. In addition, when Action#2 starts to fail, they switch back to applying Action#1. It can be concluded that learning agents are able to switch to another rule when the initially adopted rule starts to fail. The findings fundamentally demonstrate that the learning AGV holon are able to adapt themselves to the changes in the environment and can learn to execute the more favorable action in a given state.

6. CONCLUSION AND RECOMMENDATIONS

Reinforcement learning has become one of the most widely used learning methods in the area of artificial intelligence and machine learning. However, there are comparatively few attempts to relate RL to manufacturing systems. The practical applications of RL have not been thoroughly explored yet. Although, there have been studies conducted by many authors demonstrating the usefulness of reinforcement learning, there are a few AGV related practical applications demonstrating the feasibility of adaptability of learning AGVS. This issue has been previously assessed only to a very limited extent, showing that there is still a long way to go to spread the RL methodologies in this research area.

This thesis addresses a tactical level shop floor problem with a reinforcement learning method. Q-learning methodology is proposed for the distributed control problem of single-load AGVs. The consequences of implementing the Q-learning method to the pickup-dispatching rule selection problem of multiple AGVs is studied. In the course of this study, the holonic approach is adapted. In an effort to assess the performance of the learning AGV agents, a hypothetical multi agent manufacturing system is built to serve as an infrastructure for the specific research topic.

The objectives dealt with here is to train AGVs to minimize (1) the empty travel time of the AGVs, and (2) the wasted machining time due to blockages. The pickup-dispatching problem is formulated as a Q-learning problem by defining state determination criteria, actions space and 3 types of reward functions. AGVs are able to negotiate with products and machines and gather full information on each agents' instant state.

Simulation tests are carried out. Through computational experiments many conclusions are drawn. Simulation results principally proved that the AGVs are able to learn to practice the best dispatching rule as discovered in the static pick-up dispatching case. Results confirmed that AGV agents with the Q-learning algorithm are able to learn to favor the best action for various system goals. The learning agents are able to distinguish the best alternative favoring the preset system goal even when a larger action space is employed.

To evaluate the performance of the proposed approach, the performance of the proposed Q-learning approach is compared with that of static pickup dispatching rules. The simulation results indicated the learning AGVs are able to find a common ground between contradictory system goals. The learning agents are able to switch objectives in order to find an acceptable compromise between two extremes. In an environment where there are multiple perspectives and/or the opposing interests, employing a learning methodology results in better performance.

In the final tests, the proposed method is tested in a dynamic environment. The tests aimed to show that learning agents are able to switch to another dispatching rule when the initially adopted rule starts to fail. The findings most importantly establish that the learning AGV holon is able to adapt itself to the changes in the environment and can learn to practice the best action in a given state. The result proves that AGVs can learn gather experience from the past and update their actions based on the needs of the current system.

To determine a dispatching rule favoring a particular goal may not always be straightforward. It may require some specific resources and may take serious effort and time to find out which dispatching rule favors a certain system goal. Even simulation studies may not always be convenient and accessible. Especially when fluctuations and interruptions disrupt the natural behavior of the system and the system's response has to be performed in a timely manner. Presumably, the initially adopted rule starts to fail when a change occurs in the system. In these turbulent circumstances, there is no chance to determine and verify the best rule favoring a certain performance measure experimentally or practically. Instead, a holonic manufacturing system consisting of learning agents is able to sense state-action dependencies and agents are able to modify their behavior accordingly. AGVs can generate a self-controlling network to switch to the better rule. Heterarchical shop floor control architecture enables a modifiable, extensible, and adaptable control mechanism. It also promotes reactive, satisfactory, adjustable and robust solutions. The experience from this thesis provides a good foundation for forthcoming works into applying reinforcement learning methods to more complex agent/holon based manufacturing problems with larger state and action spaces in dynamic job shop environments.

REFERENCES

1. Andreadis, G., Klazoglou, P., Niotaki, K. & Bouzakis, K. D., 2014. *Classification and review of multi-agents systems in the manufacturing section.* s.l., s.n.

2. Ansari, F., Erol, S. & Sihn, W., 2018. *Rethinking Human-Machine Learning in Industry 4.0: How Does the Paradigm Shift Treat the Role of Human Learning?.* s.l., Elsevier B.V., pp. 117-122.

3. Aziz, F. A., 2013. A Review of the Negotiation Protocol for Agent based Manufacturing System Control. *International Journal of Information Technology & Computer Science* , 8(1).

4. Barbat , B., Candea , C. & Zamfirescu, C., 2001. *Holons and Agents in Robotic Teams A Synergistic approach.* Dubai, ENAIS'2001.

5. Barenji, A. V., Shaygan, A. & Barenji, R. V., 2016. *Simulation Platform for Multi Agent Based Manufacturing Control System Based on The Hybrid Agent.* s.l., s.n.

6. Berman, S., Schechtman, E. & Edan, Y., 2009. Evaluation of automatic guided vehicle systems. *Robotics and Computer-Integrated Manufacturing.*

7. Bilge, Ü., Beşikçi, U., Erbeyoğlu, G. & Şahinkoç, M., 2016. *Implementation and Performance Based Comparison of Shop Floor Control Architectures using Distributed and Parallel Simulation,* s.l.: s.n.

8. Bilge, Ü., Esenduran, G., Varol, N., Öztürk, Z., Aydin, B., & Alp, A., 2006. Multi-attribute responsive dispatching strategies for automated guided vehicles. *International Journal of Production Economics.*

9. Chen, C., Xia, B., Zhou, B.-h. & Xi, L., 2015. A reinforcement learning based approach for a multiple-load carrier scheduling problem. *Journal of Intelligent Manufacturing.*

10. Co, C. G. & Tanchoco, J. M., 1991. A review of research on AGVS vehicle management. *Engineering Costs and Production Economics.*

11. da Silva, R. M., Junqueira, F., Filho, D. J. & Miyagi, P. E., 2016. Control architecture and design method of reconfigurable manufacturing systems. *Control Engineering Practice,* 1 4, Volume 49, pp. 87-100.

12. da Silva, R. M., Miyagi, P. E. & Santos Filho, D. J., 2011. *Design of active holonic fault-tolerant control systems.* s.l., s.n.

13. Davis, R., 1981. Frameworks for Cooperation in Distributed Problem Solving. *IEEE Transactions on Systems, Man and Cybernetics.*

14. De Koster, R., Le-Anh, T. & Van Der Meer, J., 2004. Testing and classifying vehicle dispatching rules in three real-world settings. *Journal of Operations Management,* 22(4 SPEC. ISS.).

15. Dominici, G., Argoneto, P., Renna, P. & Cuccia, L., 2010. The Holonic Production System: A Multi Agent Simulation Approach. *iBusiness.*

16. Egbelu, P. J. & Tanchoco, J. M. A., 1984. Characterization of automatic guided vehicle dispatching rules. *International Journal of Production Research.*

17. Erol, R., Sahin, C., Baykasoglu, A. & Kaplanoglu, V., 2012. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. *Applied Soft Computing Journal.*

18. Farid, A. M., 2004. *A Review of Holonic Manufacturing Systems Literature,* s.l.: Center for Distributed Automation & Control Institute for Manufacturing Engineering Department University of Cambridge.

19. Foit, K., Banaś, W., Gwiazda, A. & Hryniewicz, P., 2017. *The comparison of the use of holonic and agent-based methods in modelling of manufacturing systems.* s.l., s.n.

20. Franklin, S. & Graesser, A., 2005. Is It an agent, or just a program?: A taxonomy for autonomous agents. In: s.l.:s.n.

21. Gaur, A. V. & Pawar, M. S., 2016. AGV Based Material Handling System : A Literature Review. *International Journal of Research and Scientific Innovation.*

22. Gräßler, I. & Pöhler, A., 2017. *Implementation of an Adapted Holonic Production Architecture.* s.l., Elsevier B.V., pp. 138-143.

23. Ho, Y. C. & Chien, S. H., 2006. A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs. *International Journal of Production Research,* 15 10, 44(20), pp. 4193-4222.

24. Ho, Y. C. & Liu, H. C., 2009. The performance of load-selection rules and pickup-dispatching rules for multiple-load AGVs. *Journal of Manufacturing Systems,* 1, 28(1), pp. 1-10.

25. Hsieh, F.-S., 2002. Multi-Agent Control Of Holonic Manufacturing Systems Based On Petri Net. *IFAC Proceedings Volumes.*

26. Jain, V. & Raj, T., 2016. Modeling and analysis of FMS performance variables by ISM, SEM and GTMA approach. *International Journal of Production Economics.*

27. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Wooldridge, M., & Sierra, C., 2001. Automated Negotiation: Prospects, Methods and Challenges. *Group Decision and Negotiation.*

28. Junqueir, F., da Silva, R. M., Filho, D. J. S. & Miyagi, P. E., 2011. *Design Of Control Systems Based On Holon, Petri Net And Multi-Agent System,* São João del-Rei - Brasil: X SBAI – Simpósio Brasileiro de Automação Inteligente.

29. Kanchanasevee, P., Biswas, G., Kawamura, K. & Tamura, S., 1997. *Architectures, Networks, and Intelligent Systems for Manufacturing Integration,* s.l.: s.n.

30. Klei, C. M. & Kim, J., 1996. AGV dispatching. *International Journal of Production Research,* 34(1), pp. 95-110.

31. Koestler, A., 1969. *The Ghost in the Machine.* London: Arkana Books.

32. Kumar, N., Tiwari, M. K. & Chan, F. T., 2008. Development of a hybrid negotiation scheme for multi-agent manufacturing systems. *International Journal of Production Research.*

33. Lander, S. E. & Lesser, V. R., 1993. *Understanding the Role of Negotiation in Distributed Search Among Hetergeneous Agents.* s.l., s.n.

34. Le-Anh, T. & De Koster, M. B., 2006. A review of design and control of automated guided vehicle systems. *European Journal of Operational Research.*

35. Leitão, P., 2009. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence,* 10, 22(7), pp. 979-991.

36. Lind, M. & Roulet-Dubonnet, O., 2011. Holonic shop-floor application for handling, feeding and transportation of workpieces. *International Journal of Production Research.*

37. Liu, S., Elmekkawy, T. Y., Fahmy, S. A. & Shalaby, M. A., 2008. *Design and operational analysis of tandem AGV systems.* s.l., s.n.

38. Mahadevan, B. & Narendran, T. T., 1990. Design of an automated guided vehicle-based material handling system for a flexible manufacturing system. *International Journal of Production Research.*

39. Mahalingam, V. K., 2014. *A Simulation Study of Flexible Manufacturing System Using Dynamic Scheduling Approach.* s.l., s.n.

40. Monostori, L., Váncza, J. & Kumara, S. R., 2006. Agent-based systems for manufacturing. *CIRP Annals - Manufacturing Technology.*

41. Morariu, O., Morariu, C., Borangiu, T. & Raileanu, S., 2015. *Multicast dataset synchronization and agent negotiation in distributed manufacturing control systems.* s.l., s.n., pp. 1087-1092.

42. Peters, B. A., Smith, J. S. & Venkatesh, S., 1996. A control classification of automated guided vehicle systems. *International Journal of Industrial Engineering*

: Theory Applications and Practice.

43. Proper, S. & Tadepalli, P., 2006. *LNAI 4212 - Scaling Model-Based Average-Reward Reinforcement Learning for Product Delivery,* s.l.: s.n.

44. Shen, W. & Norrie, D. H., 2013. Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey. *Knowledge and Information Systems.*

45. Smith, R. G., 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers.*

46. Srivastava, S. C., Choudhary, A. K., Kumar, S. & Tiwari, M. K., 2008. Development of an intelligent agent-based AGV controller for a flexible manufacturing system. *International Journal of Advanced Manufacturing Technology.*

47. Sutton, R. A. & Barto, A. G., 1998. *Reinforcement Learning.* s.l.:The MIT Press.

48. Trentesaux, D., 2009. Distributed control of production systems. *Engineering Applications of Artificial Intelligence.*

49. Tripathi, A. K., Tiwari, M. K. & Chan, F. T., 2005. Multi-agent-based approach to solve part selection and task allocation problem in flexible manufacturing systems. *International Journal of Production Research,* 1 4, 43(7), pp. 1313-1335.

50. Unland, R., 2015. Industrial Agents. *Industrial Agents,* 1 1.pp. 23-44.

51. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., & Peeters, P., 1998. *Reference architecture for holonic manufacturing systems: PROSA,* s.l.: s.n.

52. Wang, L. & Haghighi, A., 2016. Combined strength of holons, agents and function blocks in cyber-physical systems. *Journal of Manufacturing Systems,* 1 7, Volume 40, pp. 25-34.

53. Wang, Y. C. & Usher, J. M., 2005. Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence.*

54. Watkins, C. J. C. H. & Dayan, P., 1992. Q-learning. *Machine Learning,* 8(3-4), p. 279–292.

55. Wong, T. N. & Fang, F., 2010. A multi-agent protocol for multilateral negotiations in supply chain management. *International Journal of Production Research.*

56. Wong, T. N., Leung, C. W., Mak, K. L. & Fung, R. Y., 2006. An agent-based negotiation approach to integrate process planning and scheduling. *International Journal of Production Research.*

57. Wooldridge, M., 1998. Agent technology: foundations, applications, and markets. *Springer Science & Business Media.*

58. Xue, T., Zeng, P. & Yu, H., 2018. *A reinforcement learning method for multi-AGV scheduling in manufacturing.* s.l., s.n.

59. Zhang, Z., Zheng, L., Li, N., Wang, W., Zhong, S., & Hu, K., 2012. Minimizing mean weighted tardiness in unrelated parallel machine scheduling with reinforcement learning. *Computers and Operations Research.*