UNIVERSITÀ DEGLI STUDI DI MODENA E REGGIO EMILIA

INTERNATIONAL DOCTORATE SCHOOL IN
INFORMATION AND COMMUNICATION TECHNOLOGIES

DOCTORAL PROGRAMME IN COMPUTER SCIENCE

MODELS FOR THE PREDICTION AND MANAGEMENT OF COMPLEX SYSTEMS IN INDUSTRIAL AND DYNAMIC ENVIRONMENTS

Doctoral Dissertation of:
Emanuel Federico Alsina

Supervisor:
Prof. Giacomo Cabri

The Chair of the Doctoral Program:
Prof. Giorgio Matteo Vitetta

2016 – XXVIII

Abstract

THE world in which we live is becoming more and more complex. Modeling the reality means to create simplifications and abstractions of that, in order to figure out what is going on in this modern and complex world. Nowadays, models have become crucial to make better decisions. Models help us to be clearer thinkers, and to understand how to transform data in useful information. The number of available data is continuously increasing, models take these data and structure them into information, and then into knowledge. Two main topics are discussed in this work: (1) how to model complex systems, and (2) how to make predictions within complex systems, in industrial and dynamic environments. The purpose of this thesis is to present a series of models developed to support the decision makers in the complexity management. The first topic is addressed presenting some models concerning the balancing of assembly lines, machines' degradation in production lines, operations' schedule, and the positing of cranes in automated warehousing. In particular, concerning the assembly lines, two bio-inspired models which optimize the global picking time of the components considering their physical allocation are presented. Moreover, the use of a multi-agent model able to simultaneously consider different factors that affect machines in a production line is analyzed. This approach takes into account the aging and the degradation of the machines, the repairs, the replacement, and the preventive maintenance activities. Furthermore, in order to present how to manage the complexity intrinsic into the operations' scheduling, a model inspired by the behavior of an ant colony is showed. Finally, another multi-agent model is showed, which is able to find the optimal dwell point in automated storage retrieval systems exploiting an idea deriving from force-fields. After that, an entire chapter is dedicated to the prediction in complex systems. Prediction in industrial and dynamic environments is a challenge that professionals and academics have to face more and more. Some models able to capture non-linear relationships between temporal events are presented. These models are applied to different fields, from the reliability of mechanical and electrical components, to renewable energy. In the final analysis, models able to predict the users' behaviors within online social communities are introduced. In these cases, various machine learning approaches (such as artificial neural networks, logistic regressions, and random trees) are detailed. This thesis want to be an inspiration for those people which have to manage the complexity in industrial and dynamic environments, showing examples and results, in order to explain how to make this world a little more understandable.

Abstract

IL mondo in cui viviamo sta diventando sempre più complesso. Modellare la realtà significa creare sue semplificazioni e astrazioni, in modo da capire cosa succede in questo mondo moderno e complesso nel quale viviamo. Oggigiorno, i modelli sono diventati fondamentali per prendere decisioni migliori. I modelli ci aiutano a pensare con maggiore chiarezza e a capire come trasformare i dati in informazioni utili. Il numero di dati disponibili è in continuo aumento, i modelli prendono questi dati e li strutturano in informazioni, e successivamente in conoscenza. In questo lavoro sono affrontati due temi fondamentali: (1) come modellare sistemi complessi e (2) come fare previsioni all'interno di sistemi complessi, in ambienti industriali e dinamici. Questa tesi presenta una serie di modelli creati a supporto di chi si trova a dover gestire la complessità. Il primo tema è affrontato presentando alcuni modelli relativi a linee di assemblaggio, il degradamento delle macchine nelle linee di produzione, la programmazione delle operazioni e il posizionamento dei carrelli all'interno di magazzini automatizzati. In particolare, per le linee di assemblaggio, sono presentati due modelli bio-inspirati che ottimizzano il tempo totale di prelievo dei componenti, considerando la loro allocazione fisica. Inoltre, è analizzato l'uso di un modello multi-agente capace di considerare contemporaneamente diversi fattori che influenzano le macchine in una linea di produzione. Quest'approccio permette di tenere in considerazione l'età e la degradazione delle macchine, le riparazioni, le sostituzioni e le attività di manutenzione preventiva. In seguito, è mostrato un modello inspirato dal comportamento di una colonia di formiche, il quale è in grado di gestire la complessità intrinseca all'interno della programmazione delle operazioni. Infine, è presentato un altro modello multi-agente, questa volta capace di ottimizzare il punto di riposo di un sistema di stoccaggio e prelievo automatico, sfruttando un'idea derivata dai campi di forze. Dopo di che, un capitolo intero è dedicato alla previsione in ambienti complessi. La previsione in ambienti industriali e dinamici è una sfida che si trovano ad affrontare ogni giorno sempre più professionisti ed accademici. Sono presentati modelli capaci di catturare relazioni non lineari tra eventi temporali. Questi modelli sono applicati a diversi ambiti, dall'affidabilità di componenti meccanici ed elettrici alla previsione dell'irraggiamento solare. In ultima analisi sono introdotti alcuni modelli capaci di predire il comportamento degli utenti all'interno di comunità online. In questi casi, sono esposti in dettaglio alcuni modelli di machine learning, tra cui reti neurali artificiali, regressione logistica e random forest. Questa tesi vuole essere un'ispirazione per coloro che devono gestire la complessità in ambienti industriali e dinamici. Mostrando questi esempi e risultati si vuole umilmente spiegare come rendere questo mondo un po' più comprensibile.

Contents

CHAPTER 1

Introduction

Nowadays, *complexity* has become part of our ordinary vocabulary; it is used in everyday life and in different contexts. As recently as 20 years ago complexity theory became a major field of interdisciplinary research that has since then modified considerably the scientific landscape [152]. Actually, the theory of complexity goes back to the forties and for quite some time they remained confined to a large extent within communities of strong background in physical and mathematical science. But it is in the recent decades that complexity started to be studied in different disciplines, approaching a large body of phenomena of concern at the crossroads of engineering, environmental, life and human sciences from a unified point of view. This theory responded the scientific need to understand seemingly random phenomena (which were not random); the so called "noise" which appears in various types of physical systems when certain dynamic parameters exceed specific thresholds. Beyond these thresholds, science considered incomprehensible phenomenon itself and cataloged under the "chaos" voice. Chaos and complexity and their definitions are linked profoundly to time. Often, chaos is associated with the *butterfly effect* [98]. The famous saying goes that a butterfly flaps its wings in Brazil and a storm results in Texas. The principal message of this provocative sentence is that the behavior of the atmosphere in unstable with respect to perturbations of small amplitude. In fact, nonlinear systems address phenomena of: (1) self-organization in non-equilibrium systems; (2) pattern formation; (3) deterministic chaos; (4) explaining the success of unexpected structures; and (5) events generated from the laws of nature in systems involving interacting subunits when appropriate conditions are satisfied. In particular, nonlinear systems do not respond to the principle of superposition: the sum of the parts is not enough to understand the behavior of the total system. Their key feature is the importance of the interactions between the parties rather than the properties of its single components. On the other hand, linear systems are subject to the principle of superposition, i.e., it is possible to understand and analyze them by studying separately the parts that compose them.

Precisely because of its multidisciplinary nature, there is not an unambiguous definition of complexity. One of the most accepted definition is the one of Day and Mizrach [62], due to its “broad tent” nature. The authors claimed that a dynamical system is complex if it endogenously does not tend asymptotically to a fixed point, a limit cycle, or an explosion. Such systems can exhibit discontinuous behavior and can be described by sets of nonlinear differential or difference equations, possibly with stochastic elements. *Complex system* is then a wide term which can concern a different nature, and for this reason can be consider a sort of *lingua franca* among different disciplines. These systems are characterized by a certain number of interactions between the elements that are part of. These interactions are the centerpiece of the systems, and can generate a self-organized behavior of the system as a whole. Understanding and management of complex systems is difficult. In natural sciences, such as physics and chemistry, formulated theories can be tested directly in the laboratory, so researchers are able to verify the correctness of their hypotheses. These experiments allow to understand the factors that rise certain phenomena or to modify certain aspects to observe their consequences. These experiments can be extremely costly in industrial field for instance, or even unavailable in social sciences, as they do not have a real laboratory. Computer simulation offers to these disciplines a new and important research tool able to study some aspects of reality, such as processes and nonlinear relationships, through the creation of accurate computational models. Computer simulation, or computational modeling, involves representing a model as a computer program. Computer programs can be used to model either quantitative theories or qualitative ones. They are particularly good at modeling processes and although non-linear relationships can generate some methodological problems, there is no difficulty in representing them within a computer program [85]. Computer simulation can be considered as the “third way” which offers a good balance between the flexibility of descriptive models and the computational possibilities of the mathematical and statistical. The first ones are very flexible, but their suitability cannot be verified by calculation tools; while the latter are computable, but often need strong hypothesis which may cause a detachment from reality. It became clear that generic aspects of the complex behaviors observed across a wide spectrum of fields could be captured by minimal models governed by simple local rules. Some of them gave rise in their computer implementation to attractive visualizations and deep insights, from Monte Carlo simulations to multi-agent systems. These developments provide the tools and paved the way to an understanding, both qualitative and quantitative, of the complex systems encountered in nature, technology and everyday experience.

Understand models lead to know what is going on in this modern and complex world. Models are simplifications and abstractions, Box claimed that *essentially, all models are wrong, but some are useful* [34]. Models can be simplifications of the reality but they help understand how to do things in better ways. Models give the considerations under which something works. They identify relationships, explore alternatives and consequences, identify logical boundaries, and communicate. Models take data and structure them into information and then they structure information into knowledge. Since analytical methods have some limitations to handle the degree of complexity of real-world problems, simulation method is today recognized as a promising tool for detailed investigations and reliable for the problem solving of complex system [49]. Simulation is a numerical analysis method designed to estimate the true characteristics of complex systems in which some components behave stochastically. For this reason, this thesis aims to present how manufacturing systems could certainly benefit from simulation analysis, and how advanced models can be used to make predictions in complex and dynamic environments. This work presents several models which manage the complexity that influences the nonlinearity of industrial and dynamic environments, as well as social interactions. Practically, the thesis in divided into two main chapters. The first one concerns the introduction of some approaches to analyze complex problems in the manufacturing field. The second one, instead, presents some techniques to make predictions in complex systems, where predictive models have to be able to understand nonlinear dynamics.

The main purpose of chapter 2 is to introduce the concept of complexity and nonlinearity in environments that historically have been treated as linear. Within this chapter, section 2.4 presents a method that involves two bio-inspired models that optimize the global picking time of the components considering their physical allocation within assembly lines. The method is able to balance assembly lines in terms of time and space, hence optimizes the allocation of the components using an evolutionary approach: genetic algorithm and genetic programming. Section 2.2 introduces the use of a multi-agent model able to simultaneously consider different factors that affect the machines reliability in a production line. This model simulates the failure behavior of complex repairable manufacturing systems, managing the complexity deriving from the influence of the aging and the degradation of the machines, repairs, replacement, and preventive maintenance activities. Furthermore, section 2.3 presents a model able to manage the complexity intrinsic into the operations' scheduling inspired by the behavior of ant colonies. The flexibility of the model makes it useful to schedule the departures of buses in dynamic environments, such as developing countries. Finally, section 2.4 introduces the using of multi-agent models to simulate the behavior of the different parts of automated storage retrieval systems. Additionally, a model is presented which is able to find the optimal dwell point of the crane of different types of automated systems.

As introduced, complex systems are influenced by several complex factors that exhibit nonlinear patterns. Their prediction cannot depend on the assumptions of independence and linearity, but requires models which capture the complexities of the system behavior in a realistic way. The main purpose of chapter 3 is to introduce some of these models. Section 3.1 explores the accuracy of a series of artificial neural networks that predict the cumulative failure distribution of different components mechanical and electric components. The idea is to show that even simple networks can achieve good results, in order to encourage non-expert users to use them. After that, the performance of more advanced models of machine learning and soft computing are analyzed, finding how the random forest algorithm outperforms on average the other models. Section 3.2 analyzes the use of artificial neural networks to predict the average daily global solar radiation over several Italian places. In addition an agent-based model related to the use of web services to balance the exchanging of green energy in a hypothetically smart city is presented. Finally, section 3.3 presents how to apply research techniques to figure out some important patterns that influence the users' behavior within social media platforms. In particular, a huge network of question answering communities is analyzed, figuring out the influence of diversity in users' tenure in the ability of the community to produce worse or better answers. Moreover, the section explores how time constraints and cognitive capacity limit the effort that users spend on discovering and evaluating answers. Some evidences of simple cognitive heuristics are analyzed, affecting in this way the emergent collective behavior of askers and voters on the communities.

CHAPTER 2

Manufacturing systems

The numerous uncertainties of the manufacturing systems have attracted the studies of several research works. Manufacturing systems are complex systems, they are constituted by many elements which exchange information with each other. Production and assembly lines, for example, are multilevel systems with multiple targets formed by a number of standards units. Some examples include the modern high-tech manufacturing systems, such as in the electronics, semiconductor, aerospace, and automotive industries. The complexity is due by *external* and *internal* factors. The external complexity derives from the open and dynamic environment that is around the manufacturing systems, including organizational relationship, products complexity, and so on. The internal complexity derives from the relationship between the elements and the exchange of information between them, which include multiple part types made in the same line, numerous manufacturing steps (300-500 steps is not uncommon), very complex equipment that leads to high levels of maintenance and downtime, and multiple levels of sub–assemblies [77]. In addition, in a more and more competitive global market, it is easy to understand the increase of the competition between the manufacturing factories. Despite the final goods produced, the common denominator that factories at global level are dealing with is the incessant request to reduce production time and costs, maintaining a high level of service and quality of the goods offered. This challenge induces changes to the operations and configurations of manufacturing organizations, such as the increasing or decreasing of the production capacity, the introduction of new production technology and changes to the workforce [232]. All this complexity, combined with the high cost of setting up and maintaining such systems, leads to the need of system models, rather than just relying on experience or simple rules of thumb for performance evaluation and decision making. Therefore, there is a need for *modeling and simulation* for decision support in current and future manufacturing systems. Complex systems science concerns the transformation and evolution of systems over time and, therefore, it can provide a natural framework for study changes, consequences, and performance in manufacturing systems.

Models are intended to support management decisions about the system, but obviously a single model often will not be capable of supporting all decisions. Different decisions require different models which evaluate various aspects of the design and operation of the system. Literature is rich of studies concerning the *manufacturing planning* (i.e., the process of selecting and sequencing manufacturing processes and parameters so that they achieve goals and satisfy constraints), *manufacturing schedule* (i.e., the process of assigning manufacturing resources over time to the set of manufacturing processes in the process plan) [193], and *manufacturing control* (i.e., the process of managing and controlling the physical activities in the factory aiming to execute the manufacturing plans) [126]. It emerged that simulation models provide an accurate estimate of manufacturing system behavior, able to understand the dynamics of these complex systems, but usually at more computational cost. The strengths of the simulation models include: (1) time compression, they are able to simulate years of real system operation in a much shorter time; (2) component integration, they can integrate several complex system components to study their interactions; (3) risk avoidance, potentially dangerous or costly systems can be studied without the financial or physical risks involved; (4) physical scaling, larger or smaller versions of systems can easily studied; (5) repeatability, different systems in identical environments or the same system in different environments can be simulated; and (6) control, everything in a simulated environment can be precisely monitored [230].

This chapter describes how manufacturing systems can be considered as complex systems, and it presents some simulation models to manage this complexity in production and assembly lines, planning operations, and automatic warehouses. The chapter is organized as follow: Section 2.1 presents a method which involved two bio–inspired models that optimize the global picking time of the components of an assembly line. The method is able to balance assembly lines in terms of time and space, hence optimizes the physical allocation of the components using an evolutionary approach. The use of a multi-agent model able to simultaneously consider different factors that affect machines in a production line is presented in Section 2.2. This approach takes into account the aging and the degradation of the machines, the repairs, the replacement, and the preventive maintenance activities. Section 2.3 shows a model to manage the complexity intrinsic into the operations planning. The algorithm in this model was inspired by the behavior of an ant colony. And finally, a multi-agent model able to find the optimal dwell point in automated storage retrieval systems exploiting an idea deriving from force-fields is presented in Section 2.4.

2.1 Assembly lines

In the standardized industrial production world as well as in the production of customized products, assembly lines are flow oriented systems extremely diffused. An assembly line consists in a set of *work stations* arranged in series or parallel, creating a flow of assembling operation. The pieces of semi-assembled are consecutively moved from one station to the next, through a conveyor or a similar mechanical material handling equipment, supplying continuously the various stations [22]. The balancing of assembly lines, i.e. the assignment of the different tasks to the stations fulfilling certain restrictions, is one of the most studied industrial issues, both in academic and practical fields. The workable application of the solutions passes through a reliable simplification of the real-world assembly line systems. Time and space assembly line balancing problems consider a realistic versions of the assembly lines, involving the optimization of the entire line cycle time, the number of stations to install, and the area of these stations. In this section, the findings of the paper published in [11] are presented. This model balances the line in terms of time and space, optimizing the allocation of the components is presented. In particular, the model combines the bin packing problem with a genetic algorithm and a genetic programming, finding in a first step different solutions to the line balancing problem and then evolve they in order to optimize the allocation of the components in certain areas of the workstation.

2.1.1 Assembly line balancing

The assembling process is divided into a set of *tasks* which are cyclically performed. Each assembly task j requires a different *operation time* t_j for its execution. The assignment of the *workload* (i.e., a subset of tasks) to each station respecting some constraints or objectives, is one of the most usual and hard problems in the field, known as the *assembly line balancing* (ALB) problem [52, 72, 228]. Generally, each station has a fixed common time to complete its tasks, called *cycle time* CT, after which the semi-assembled piece is transported from one station to the subsequent. Each station has a *station workload time*, that is the cumulative operation times of the workload assigned to the station. If the cycle time is imposed, a line is considered balanced and feasible only if the workload time of each station does not exceed the CT. Performing a task does not require only a certain time, but also a series of other factors, i.e., the completion of previous tasks, equipment of machines, components, skills of workers, and so on. ALB problems consist in assign all the tasks to the various stations, respecting the constraints of time and further. In other words, the goal is to assign to each station a group of tasks that minimizes the inefficiency of the line (its downtime) and that respects all the constraints imposed on the tasks and on the stations [20]. Due to the large number of variables in the problem and the variety of possible assembly lines, this problem is considered a complex problem. For this reason, some previous research activities have focused only in one or more of those aspects and the fulfilling of certain restrictions in the stations.

The simplest family of problems considers the time and the precedence of some tasks respect other as the only constrains. The modeling and solving of these problems are called *simple assembly line balancing* (SALB) [189]. The purpose of this family of problems is principally to minimize the number of station (given a fixed cycle time), or minimize the cycle time (given a fixed number of stations). Or further, simultaneously minimize the cycle time and number of station achieving multi-objective cost and profit purposes. However, the assumptions made in the SALB problems are very restricting with respect to real-world assembly line systems. When other constraints and considerations are added to the SALB, the problems are called *general assembly line balancing* (GALB) [22]. This family of problems wants to sew up the gap between the academic discussion and practical applications. GALB problems can consider for instance equipment selection and cost [40], parallel stations [215], U-shaped line layout [84], among others [35]. Belonging to this family, Bautista and Pereira [20] defined a set of problems that they call time and space constrained assembly line (TSALB), where the spatial constraint of the components necessary to operate the tasks is considered. In fact, in the assembling of big components (i.e., automotive industry) the items normally are allocated in designated areas close to each station. Bukchin [41] considered that the components can be allocated in some traditional shelves or boxes, studying the TSALB problems focused on the dimensions of these containers for different components and their allocation along the line. It is important to keep the components as near to the workplace as possible, considering the space limitations. According to Chica et al. [54] these kinds of problems contain three conflicting objectives to be accomplished: the cycle time of the assembly line, the number of the stations, and the area of these stations. The modeling of the TSALB problem is based on the mathematical formulation of the SALB problem provided by Patterson and Albracht [162]. Declaring the following variables:

$j = 1,\ ...,\ J$ Tasks
$z = 1,\ ...,\ Z$ Components
$k = 1,\ ...,\ K$ Assembly stations
$w = 1,\ ...,\ W$ Allocation areas within each station

And the following decision variables:

$$X_{jk} = \begin{cases} 1, & \text{if task } j \text{ is assigned to station } k \\ 0, & \text{otherwise} \end{cases}$$

$$Y_{zw} = \begin{cases} 1, & \text{if component } z \text{ is stored in the area } w \\ 0, & \text{otherwise} \end{cases}$$

The following constraints were established to solve the TSALB problem:

$$\sum_{k=1}^{K} X_{jk} = 1 \quad \forall j \tag{2.1}$$

$$\sum_{w=1}^{W} Y_{zw} \geq A_{jz} \cdot X_{jk} \quad \forall j,\ z,\ k \tag{2.2}$$

$$\sum_{z=1}^{Z} A_z \cdot Y_{zw} \leq A_{wk}^{max} \quad \forall w,\ k \tag{2.3}$$

$$X_{jk},\ Y_{zw} \in \{0, 1\} \quad \forall j,\ z,\ w,\ k \tag{2.4}$$

Equality 2.1 ensures that each task is assigned to only one assembly station. Constraint 2.2, where A_{jz} is equal to 1 if the task j requires the component z, ensures that all the components necessary to a task will be allocated within the station. And finally, 2.3 is concerned with the physical area upper bound, where A_{wk}^{max} represents the limit of the allocation areas. 2.4 defines the domain of the decision variables. The variable Y will provide the assignment of the components to specific allocation areas, while the variable X will provide the assignment of the tasks to specific assembly stations. The objective is to minimize the the inefficiency of the line (i.e. the downtime of the stations); that in objective functions become:

$$\min(CT - \sum_{j=1}^{J}(t_j \cdot X_{jk})) \tag{2.5}$$

In addition to this formulation, another aspect which influences the balancing of assembly lines has to be considered: the picking time of the components necessary to complete the different tasks. These picking times can depend by their distance from the worker, the weight and manageability of the components themselves. Usually, racks for small components (screws, bolts, etc.) are arranged very close to the worker, in order to limit the number of travels to pick up a large number of small components. While bigger and often heavier components are arranged behind the worker much or less close to her. Solving a TSALB problem not considering the picking time of the components could culminate in a solution which assign to a single station several tasks with a lot of heavy components which increase more and more their time to be picked up. This because heavy components have limited space close to the worker. It is not just an issue of adding the pickup time to the task time, because the picking time can be different according to the area where the components are allocated. In order to consider also the picking time of the components, the following constraint has to be considered into the formulation:

$$\sum_{j=1}^{J}(t_j \cdot X_{jk} + \sum_{z=1}^{Z} A_{jz} \cdot \sum_{w=1}^{W} PT_{zw} \cdot Y_{zw}) \leq CT \quad \forall k \tag{2.6}$$

This inequality 2.6 ensure that each station workload time does not exceed the cycle time. In this constraint, PT_{zw} is the picking time of the component z from area w. And the objective function become:

$$\min(CT - \sum_{j=1}^{J}(t_j \cdot X_{jk} + \sum_{z=1}^{Z} A_{jz} \cdot \sum_{w=1}^{W} PT_{zw} \cdot Y_{zw})) \tag{2.7}$$

Taking into account the large number of constraints imposed by this TSALB problems, traditional search techniques for optimal solutions (exact, heuristic, and metaheuristics procedures, mainly based on branch and bound approaches) may not be the best approach. That can be true especially when the number of stations and tasks to take in consideration is particularly elevated. For this reason, an investigation towards the use of a bio-inspired approach can be an alternative artificial intelligence method for exploring these search spaces.

2.1.2 Bio-inspired approach

Bio-inspired artificial intelligence is a term that groups together several computational models and algorithms that are designed in order to mimic the efficiency of complex mechanisms that are observable in nature. Genetic algorithms (GA) [99, 146] are a branch of bio-inspired intelligence; such algorithms are population based in the sense that an initially randomized population of chromosomes represents the starting point for this mechanism. Different chromosomes represent a different solution for the same optimization problem, and each chromosome is encoded as a set of parameters needed for defining such solution. During the course of a generation, the population of solutions get through the evaluation phase in which each chromosome is tested in order to evaluate its fitness value (how well that specific solution is solving the problem); the subsequent phase of a standard genetic algorithm is the selection phase in which chromosomes featuring higher fitness values are more likely to survive and to be inserted into the next iteration (generation). Generation after generation, the selected individuals undergo a process of evolution and/or mutation, so to expand the search space in the vicinity of best performing chromosomes. In order to maintain diversity and maximize the exploration capabilities of the population, individuals that are not selected are usually replaced by randomized chromosomes. Rubinovitz and Levitin [180] applied a GA to solve a SALB, recommending the use of those algorithms when solutions diversity is more important than their accuracy. The authors noted how GAs perform much faster then traditional search techniques for problems with large number of stations. Several studies have been explored further, mainly to cope with the multiple objectives of an assembly line [53, 86, 112, 205].

Depending on the dimension and complexity of the optimization problem to solve, GAs may have the tendency to get stuck into local optima or may require an unfeasible number of generations, therefore the proposed method also investigates the use of genetic programming (GP) [117]. GP is a specialization of GA in the sense that the chromosomes are not encoded as set of parameters needed for the specific problem, instead GP uses a set of actions represented as nodes in tree structures. Nodes can be extrapolated from known heuristics so to combine sequences of actions with the purpose of evolving programs (instead of parameters) towards the optimal solution. Practically, GP allows the induction of computer programs to solve problems without explicitly programming them. Liu et al. [133] adopted a GP structure to solve a GALB problem which considers parallel assembly lines. Baykasouglu and Özbakır [21] tried to solve SALB problems using composite task assignment rules which are discovered through a GP structure. The purpose of the authors was to evolve different heuristics in order to optimize a predefined objective function. In this way the GP can adapt to solve different problem constraints and pursue different objectives. Their results argued the ability and potential of GP for solving combinatorial optimization problems. Some other few studies have been developed to generate heuristic decision rules in manufacturing system applications [66, 105, 161, 206].

In the next subsection two models which consider the physical allocation of the components are presented, highlighting the consequently optimizing of the global picking time to achieve them.

2.1.3 Method

The main idea of the proposed TSALB problem is that given a set of tasks with their temporal and spatial attributes, each task must be assigned to just one station providing that (1) there is not any station with a workload time greater than the cycle time, (2) there is not any station with a required area greater than the global available area A, and (3) the components are allocated along the line in order to optimize their global picking time. GAs and GP can be applied to almost any optimization problem [65], however in our specific case of GALB problems, it can be useful to investigate a pre-processing phase with the purpose of relaxing some constraints so to apply an evolutionary strategy (GA or GP) to a problem with a smaller dimension compared to the starting one. The well known Bin Packing problem [140] for instance, perfectly suits this need: having defined our problem to optimize an assembly line taking into account the cycle time CT, number of components Z, number of picking areas and respective picking times with an unknown number of assembly stations K, a 1D bin packing problem can be applied, in order to find the minimum number of stations in order to satisfy the CT constraint. Then it is possible to apply an evolutionary strategy to the remaining objectives.

This first phase is represented by solving a 1D *bin packing problem* (BPP) using time as the only dimension. BPP was already used in past to solve SALB problems [33, 188, 220]. In our case, each bin represents a station with its maximum time for processing being equal to CT: having a plurality of stations working in parallel ensures that cycle time constraint is respected. The items to pack are the tasks, with each task having its own processing time. So, simplifying the Equation 2.6 to consider only the execution times, the BPP trivially the following condition holds:

$$\sum_{j=1}^{J} T(t_j) \leq CT \tag{2.8}$$

With J the number of tasks within the same station/bin, $T(t_j)$ the time needed for complete task j. It is tested an implementation of BPP able to provide an exhaustive search that exploit the concept of branch and bound and propagation algorithms preprocessed with a First Fit Decreasing strategy [141]. The Java implementation used for the BPP is able to solve a moderately sized problem (26 unique tasks, 75 components, real world application data) in less than 10 ms using a Intel CORE i7, jre 1.8.40, 16 GB of RAM, therefore it is possible to conclude that it should be feasible to apply a 2D or 3D BPP for similarly sized problems, so to satisfy more constraints in this initial phase.

The BPP algorithm provides two important outputs in this first phase: (1) the minimum number of stations needed in order to satisfy CT, and (2) the list of tasks to be executed in each station particular station. The minimum number of stations K can be also mathematically verified, taking into account the execution time of each task, picking times, and cycle time of the line:

$$K = \left\lceil \frac{\sum_j \left(t_j + \sum_z \left(\min_x \left(PT_{zw} \right) \right) \right)}{CT} \right\rceil \tag{2.9}$$

In Equation 2.9, the number of stations is defined as the sum of the execution times of all the tasks and the lowest picking time of each component; the result is then divided by

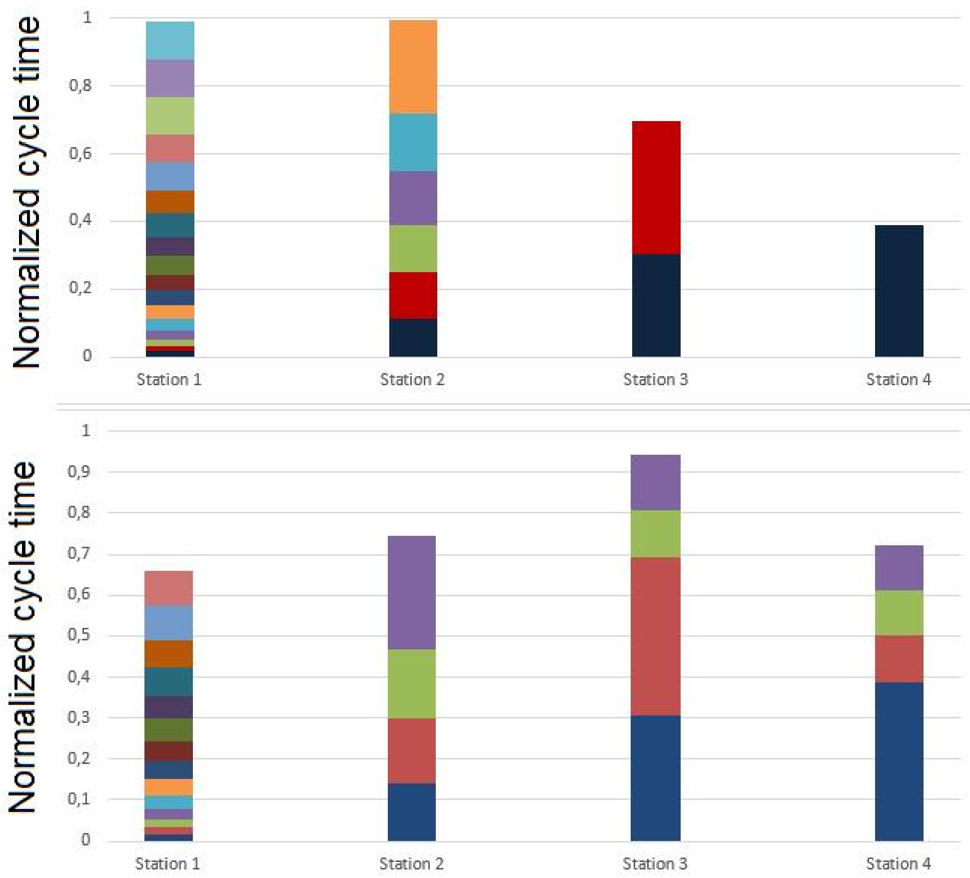


Figure 2.1: *Example of solution of BPP. Tasks with larger time requirements ended up in the rightmost bins (different colors indicates different tasks. On the Y axis, it is possible to see the proportional time occupancy normalized from 0 to 1). The second graph below is an example of equivalent solution, obtained from the original one by switching tasks with similar time requirements among different stations.*

the cycle time of the line. Concerning the tasks assigned to each station, BPP algorithm is biased to fill the first bins with smaller items, while the last bins are more likely to be filled with bigger items. More specifically to our case study, it is convenient to think that the stations are ideally located one next to the other on a straight line: the leftmost station is usually occupied with many tasks with small time requirements, while the rightmost station is most likely responsible of a small number of very time demanding tasks. This particular output configuration enables us to quickly find alternative and equivalent solutions: by exchanging tasks assigned to different stations that are characterized by a similar time requirement and/or by shifting low time demanding tasks from the leftmost bins to the rightmost bins (whenever it is possible), different starting conditions for the following phase of the algorithm are created (see Figure 2.1). It is trivial to verify that the CT constraint is easily satisfiable in each newly created solution; many solutions are created depending on the size of the problem and for future reference the set of solutions found with this mechanism are indicated as S_{BPP} . A genetic algorithm can be now applied in order to solve the picking time minimization objective. It is important to notice that up until now, the notions of components and picking areas are keept out of any equation, as at the beginning, only the CT constraint is considered. Now, in this second phase, this CT constraint can be ignored because already satisfied, and therefore the different tasks for each station will not be taken into account. Instead of thinking about tasks, the list of N components that are needed for the completion of every task is now analyzed. Starting by picking a specific solution $s \in S_{BPP}$, it is possible to model each station as a series of n tuples with n being the number of different picking areas, three in our case. Each picking area features a variable number of components z_x, and both the components and the picking areas influence the total picking time that the human operator has to spend in order to process a specific component. To summarize all of this, for each of the station k a chromosome can be encoded as seen in 2.10:

$$\begin{aligned} Station_1 \langle c_0, \ldots, c_j \rangle \, p_0 \\ \vdots \qquad \vdots \qquad \\ \vdots \quad \langle c_k, \ldots, c_n \rangle \, p_{n-1} \\ \vdots \qquad \qquad \qquad \\ Station_w \langle c'_0, \ldots, c'_j \rangle \, p_0 \\ \vdots \qquad \\ \langle c'_k, \ldots, c'_n \rangle \, p_{n-1} \end{aligned} \tag{2.10}$$

Random permutations of components within different picking areas represent the initial population of chromosomes (represented as in 2.10) that are the starting point for the genetic algorithm. Do note that permutations are only admissible within the same station. The search process begins with the first evaluation of every individual against the proposed objective function, hence for the selection of fittest individuals any selection operator normally used in literature is feasible. Similar conclusions can be drawn for mutation operators and elitism. The self-imposed constraint to contain mutations and generations of new individuals within the same station have pros and cons: the major advantage is represented by the fact that by restricting migration of single components only within different picking areas of the same station, the CT constraint is still satisfied. On the other side, the limitation of this approach is that by doing so the exploration space is limited as each station is optimized locally instead of moving towards a global optimal solution that concerns every station. On this latter remark, it has to be highlighted that such a genetic algorithm can be easily implemented to run in a parallel way w.r.t. every element $\in S_{BPP}$, so to minimize the possibility of being stuck in local optima.

This section speculates on how to use genetic programming as an alternative to genetic algorithms in order to evolve programs as set of actions able to solve the remaining constraints imposed by our objective function. Chromosomes here are therefore encoded as set of nodes in tree-like structures in which it is possible to randomly generate, select and evolve and so forth for a typically large number of generations. The set of possible actions to combine are divided into different layers, knowing that the first layer (symbolized by the choice of a station) will always be encoded into the root node of our evolutionary program. Constraining the subsequent layers to have implications: a specific layer l_a that implies layer l_b means that if an heuristic belonging to l_a

Table 2.1: *Layers, implications and proposed heuristics for GP.*

Layer	Example heuristics	Implies
Select station	Pick station having the largest/smallest number of tasks, Pick station having the largest/smallest number of components...	–
Move task	Pick task having the most/least time demanding task, pick task having the largest/smallest number of components...	Select station
Select component(s)	Pick component having the largest/smallest picking time...	Move to area
Move to area	Pick closest/2nd/3rd... closest area to human operator	Select component(s)

is chosen, the subsequent or precedent heuristic must belong to l_b: looking at Table 2.1 a summary of layers, some of their example heuristics and their respective implications is shown. Once the a first station is chosen, different chromosomes can be formed by randomly picking heuristics belonging to the other layers. If, for instance, *Move to Area* is selected after the root node, it means that the program will attempt to move to a specific component to a specific picking area within the same station, and those two artifacts (area and components) will be picked by following the heuristics indicated in the layers *Move to area* and *Select Component(s)*. Therefore it is possible to develop a program as two phase process: in the first phase a general structure of the program will be laid down, while in the second phase the generic structure will be translated into a set of actions, therefore shaping the terminal nodes of the tree-structure representing the program as an individual of the chromosomes population (see Figure2.2 for an example situation). Once an initial population of programs have been created, the following steps of the GP closely follows a standard genetic algorithm: each program is tested iterating over a maximum amount of steps or until the current action is found to be unfeasible due to breaking previously imposed constraints (such as the CT). The population is selected, evolved, mutated and re-initialized over each generation by following known genetic operators. As far as mutation operators are concerned, limits must be imposed on the maximum amount of actions that forms the program so to avoid excessively bloated solutions. It has to be pointed out that it is possible that consecutive sequences of actions may not bring any changes to the components/task distribution over the different picking areas/stations: choosing to erase such sequences while creating or evolving the programs may results in increasing computational complexity and therefore it is advisable not to detect those situations. Moreover, due to the fact that testing such sequences against the objective function will result in lower fitness values, programs that are largely constituted by these ineffective sequences are significantly less likely to survive for the subsequent generations.

2.1.4 Discussion

Time and space assembly line balancing problems model a close version to existing real-world situations of assembly lines. The existing approaches to solve TSALB prob-

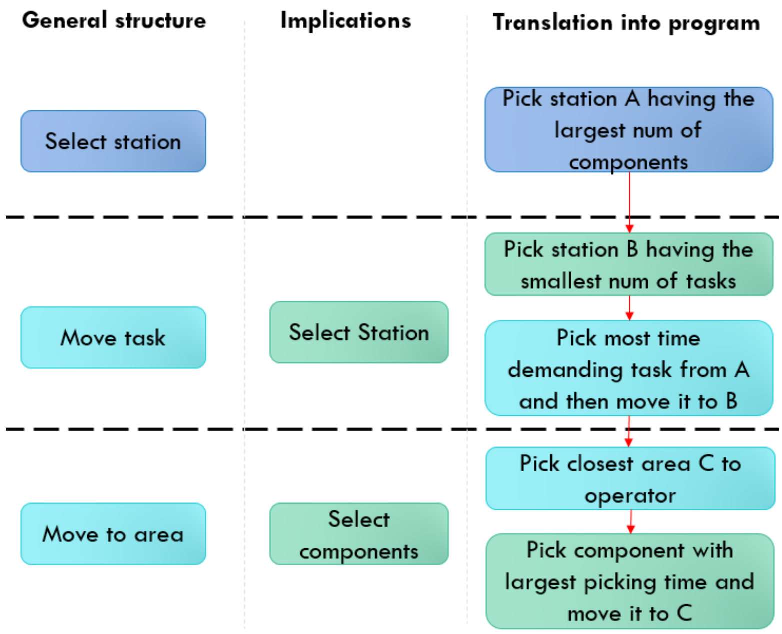


Figure 2.2: *Example situation of a randomly generated program. A first phase creates the program general structure that allows us to identify implications. From these implications it is possible to translate the general structure in a program as a sequence of actions.*

lems do not take into consideration an important real scenario of the physical allocation of the components used during the executions of the tasks. In this section, a method which addresses this issue was presented, balancing an assembly line trying to optimize the global picking time of its components. In particular, the proposed approach is able to produce solutions which respect of the time constraint and then evolve them in order to optimize the allocation of the components in different areas, complying the spatial constraint. The method was ideologically divided in two phases. The first phase is represented by solving a 1D Bin Packing Problem using time as the only dimension. Each bin represents a station with its maximum time for processing being equal to the cycle time of the entire line. The items to pack are therefore the operation times of the single tasks. In the second part, the optimization of the components' allocation and the conformity of the solutions with the spatial constraint are considered. At this point, two different evolutionary approaches were proposed: a genetic algorithm and a genetic programming. The genetic algorithm requires a variety of initial solutions given by the bin packing problem, in order to avoid a limited exploration of the solution space. The genetic programming approach, on the other hand, sets some nodes in tree-like structures in which a series of heuristics are randomly generate which solve the remaining constraints imposed by the objective function. Selecting, evolving, mutating and reinitializing the heuristics, following known genetic operators, the approach can be able to find sub-optimal solutions for the proposed TSALB problem. This presented method represents just a speculative preliminary investigation of solving methodologies for such a complex problem; however the suggested evolutionary approaches look promising as they have been already used for solving problems characterized by less constraints.

2.2 Production systems

A *production system* is the collection of people, equipment, and procedures organized to accomplish the manufacturing operations of a company (or other organization) [87]. A *production line* consists in a group of individual work cells: single production machine and worker assigned to that machine. In other words, a production line identifies a set of sequential machines collocated in series or in parallel between them, systematized to achieve the manufacturing process. Some examples of production systems include common systems such as bottling facilities, machine shops, textile plants, and so on. In the real world, all factory performance measures are about time. The *production rate* (also called throughput) is considered as the average number of parts produced in a time unit. Furthermore, *production capacity* is used to identify the maximum possible production rate of the entire production line. The traditional manufacturing systems were not designed to be responsible, flexible, and reconfigurable, since they were built upon rigid, centralized and hierarchical control structures that present good production optimization, but a weak response to change. Flexibility in formulating production strategies to overcome demand uncertainties is vital for the success of such industries [136]. This flexibility coupled with other disruptions such as equipment failures, workforce unavailability, quality misses, unavailability of the right raw materials at the right quantity, further compound the manner in which manufacturing systems are run and controlled [233]. In this dynamic situation, it is fundamental for the factories to pursue the objective of flexibility and optimization. Over time, indeed, the global market imposed the develop of collaborative and reconfigurable manufacturing systems that support efficiently small batches, product diversity, high quality and low costs, by introducing innovative characteristics of adaptation, agility and modularization [126]. In this section, the problem concerning the configuration (or re-configuration) of production lines is addressed. In particular, the findings of the paper published in [8] are presented, able to independently manage the variations of the production rates and the failure prone, caused by the degradation of the machines, repair actions, and replacements.

2.2.1 Production lines configuration

When a production line has to be configured or reconfigured, the simulation approach becomes of interest to predict the performance of the line designed. The major issues to address is to consider the failure-prone of the machines in the production line. In fact, the machines' installation depends on the reliability behavior of the machines. The *reliability* $R(t)$ of a machine is defined as the probability that a machine will function over a determinate period under specific operating and environmental conditions. In reliability studies, a function called *failure rate* $\lambda(t) = 1 - R(t)$ is often used. The failure rate function provides the probability that a machine breaks at the time t. Failure rate functions form is commonly called bathtub curve, as shown in Figure 2.3. When a machine starts to work, if it is placed incorrectly during assembly or poorly calibrated, it can fail in the short period. For this reason, the machines start their life cycle having a high failure rate (infant mortality), which then decreases sharply. The life cycle continues with a nearly constant failure rate period (useful life), and followed by an increasing failure rate (wear out). The area of greatest interest for reliability studies is

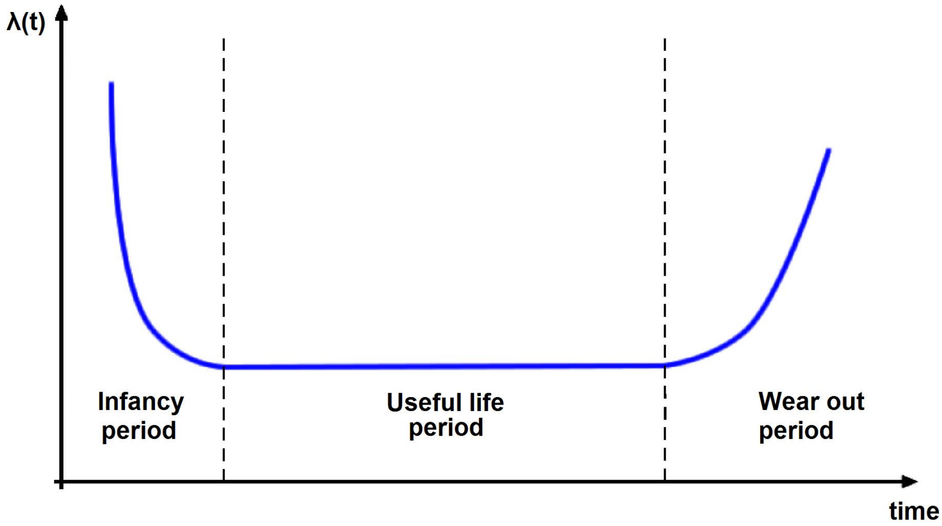


Figure 2.3: *Bathtub curve of the failure rate function.*

the "useful life" in which the breaks occur in a stochastic way [70]. Some machine are often installed in a parallel configuration due to their high failure rates, that increase the risk of blocking the entire production line.

An important concept in the manufacturing field is to maintain aligned the consumption of the machines, also if some machines could produce more than other machines in the parallel configuration. The surplus could be managed in different ways; one choice could be to produce all the necessary in one machine and leaving the other ones at rest, using it only in case of failure of the first. In this case, after some years, one machine will be consumed for a long time, while the other machine will be consumed only in the few periods when the first one was broken. For this reason, it is preferable to consume equally the machines in a parallel, producing at the same percentage of the nominal production rate.

In general, two general types of maintenance are distinguishable: *reactive* and *preventive*. Reactive maintenance is performed in response to unplanned machine downtime as a result of a failure. Preventive maintenance is scheduled downtime, usually periodical, in which inspections, repairs, cleanings, lubrications, adjustments and alignments can be performed. Particularly, preventive maintenance consists of performing a preventive intervention after tp hours of continuous machine operation. When failures occur and failure replacements are performed, the time clock is reset to zero and the planned preventive maintenance occurs when the component has been in use for the specified tp, Figure 2.4. Regardless of the type of maintenance, such as reliability, maintainability is defined as an activity characterized by a specific repair-time probability. When maintenance is performed, the repair (i.e., restoration) actions for a specific condition of a given failed component are stochastic [138]. In this stochastic repair process, a *repair rate function* $\mu(t)$ is defined as the probability that the repair is completed at time t. The forms of repair can be classified into three categories:

- *Perfect repair*, when the component functions like a new part and is called "as good as new;"

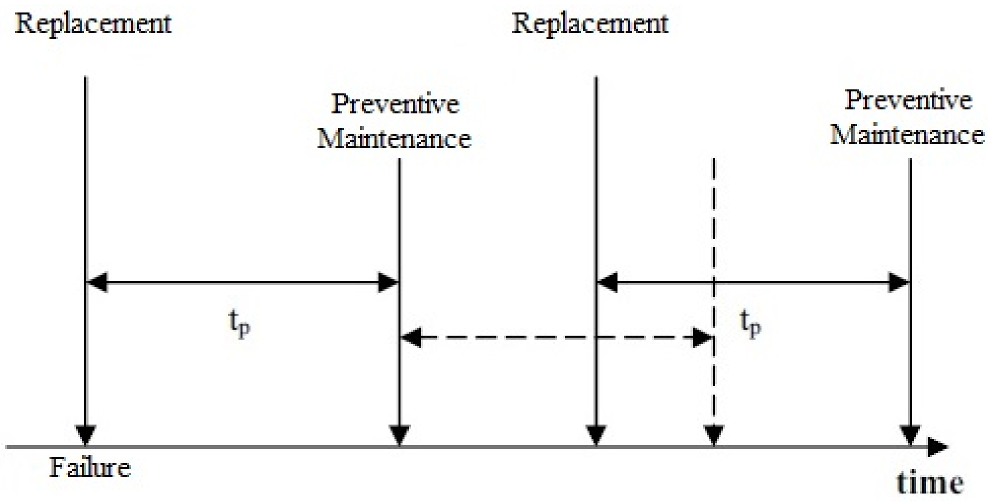


Figure 2.4: *Time-based preventive maintenance.*

- *Imperfect repair*, when the component is brought up to an intermediate stage of operation, called "better than old but worse than new;"
- *Minimal repair*, when the component returns to a state just before the break, called "as bad as old."

To understand the performance of the entire production line, considering the failure rate of its machine, maintenance and replacement activities is complex. The complexity is due to the interaction of all these elements together. To manage this complexity, different models have been created, including an autonomous agent approach which appears to be suitable to this purpose.

2.2.2 Autonomous agents approach

The failure-prone of manufacturing systems has been growing steadily as a result of the intensive search for increased productivity and better customer service [78]. This need has called for various approaches, such as holonic [159], fractal [184], biological [185], and multi-agent manufacturing systems [147]. Additionally, the maintenance activities raised research interest; due to the influence on equipment performance that the interactions between the maintenance and production functions have. Mathematical and simulation models are commonly adopted for analyzing these process interactions. As example, Soro et al. [199] evaluated the performance of multi-state degraded systems with minimal repairs and imperfect preventive maintenance. Nourelfath et al. [154] studied the consequences of an imperfect preventive maintenance in degraded systems with components connected in series and in parallel multi-state. Repairs actions, replacements, production planning, and the preventive maintenance of a deteriorating manufacturing system were studied in same mathematical model in the work of Dehayem Nodem et al. [153]. However, that model was limited to a small system consisting of one machine producing one part type, therefore with a little applicability in the real world. Furthermore, mathematical models are associated with several deficiencies that include: (1) the not adequate representation of the stochastic behavior of process equipment; and (2) the limited user understanding of the model algorithm thus use in manufacturing environment context. Consequently, simulation modeling emerged as a better approach, able to address the intrinsic limits of mathematical models related to modeling the performance of complex systems. Many articles have been written on the operational simulation of the production systems. Muchiri et al. [149] analyzed the performance behavior of manufacturing equipment subjected to various maintenance policies to counter the effect of deterioration and failure. Lavoie et al. [124] combined discrete and continuous simulation modeling to study the production rate control problem for a tandem manufacturing system with machines subject to random failures. An integration of optimization algorithms and simulation methods is proposed by Roux et al. [179]; in order to analyze maintenance strategies performances for manufacturing systems in which operating characteristics deteriorate with use and whose lifetime and repair duration are random. Smith et al. [196] have presented a panoramic view of the simulation modeling, in order to consider stochastic effects like breakdowns and interrelations between different work flows for making prognosis regarding the future system behavior.

All the above approaches are similar in that they assume network-like, dynamic, open and reconfigurable systems where decisions are made and production is carried out by more or less independent and cooperative partners. In particular, multi-agent systems seem to be suitable to face these requirements, since they present decentralization of control over distributed structures, modularity, scalability, autonomy and reusability, which are key factors for manufacturing success in the increasingly global market place. This approach can allow to represent a deteriorating production line whose failure and/or repair behaviors are not known and whose complexity is the result of the interactions based on the basic reliability and maintainability parameters of the single components in the system. Wooldridge and Jennings [222] provided one of the most common definitions of an agent as a computer system that is situated in some environment and that is capable of autonomous action in this environment to meet its design objectives. In the same way, the behavior of a single machine affects its environment, i.e., the production line. The simple behavior of each machine influences the production line, but a single agent does not have complete control over the entire environment just as a single machine does not have complete control over the entire production line. Simulating a complex production line as a multi-agent system may be beneficial because the states of the various machines can be modeled using real data and the behavior of the entire environment can emerge as the higher and well-adjusted productivity of the line. In fact, Bonabeau [28] explained how one of the most important aspects of an extended agent-based simulation is the possibility of detecting unknown (or unexpected) behavior in a complex system. Simulating a manufacturing system as a multi-agent system makes it easier to design the system; reduces the system's complexity; intensifies its recombination, expandability and reliability; and improves its flexibility, adaptability and dexterity [89]. The presented model is thought to be a good representation of the real productivity of the line, considering several different factors that affect it in the same model. A realistic simulation would enable manufacturing organizations to consider system reconfigurations and restructuring options to accommodate changing manufacturing situations, e.g., as a tool for testing a newly designed factory layout before construction [110]. The simulation is useful to verify the reliability behavior of the production line (in terms of the production rate that the line is able to achieve over time), which is unknown. The simulation was implemented using the AnyLogic software agent platform: as a general-purpose platform for simulations and trend plotting, it is widely used to model a variety of problems [29].

Hereinafter, a decentralized model that simulates distributed entities is proposed in this section. This model, presented in [8], simulates the variations of the production rates and the failure prone of production lines. These rates are influenced by several factors, such as the degradation and the failures of the machines (and their consequently repair or replace), and their maintenance.

2.2.3 Method

The presented multi-agent model takes into account the aging and the degradation of the machines, the repairs (with the time to repair related to the increase of the degradation and the number of previous repairs), the replacement, and the preventive maintenance activities. The replacement of components of the production line is also a preventive maintenance action in which the first decision addresses the definition of which critical parts must be preventively replaced. To facilitate the description of the model, an example of a production line of a pasta factory taken from the real world is presented. The scheme of this production line is composed by 9 fully automated machines, i.e. they have the capacity to operate for extended periods (longer than one work cycle) of time with no human attention. The flow of the production line follows the scheme in Figure 2.5. Row materials (RM) are introduced in the production line fueling two kneaders. The kneaders have identical proprieties and they are arranged in a fully redundant parallel configuration. Parallel stations are sometimes used to balance a production line. Their most obvious application is where a particular station has an unusually long task time, which would cause the production rate of the line to be less than that required to satisfy product demand [87]. After the kneaders, one moulder and one pasteurizer are in a series. Then the preform is moving to three bagging machines, they also in a fully redundant parallel configuration. At the end of the line, there are two palletizers which engender the final product. In order to manage the complexity, given by the number of the considered machines and the number of factors that affect the productivity of them, a multi-agent model was developed. In that model, every equipment unit in the production line was modeled as a single agent and they are well-organized in a group of agents, through their mutual coordination. The model allows a simulation that validate the performance of the manufacturing systems deteriorated with age and subject to stochastic failures, combining aligned production rates, repair/replacement, and the preventive maintenance activity.

The created multi-agent model is composed of three types of agents: *machine*, *contractNetProtocol* (CNP), and one *blackboard*. The machine agent represents the behavior of a machine and has working, failure and maintenance states. The CNP and blackboard agents serve to manage the production rate of the line. Conceptually, the blackboard sends a message to the machines asking for the maximum achievable productivity. The machines in a series configuration respond by sending a message directly to the blackboard, while the machines in a parallel configuration communicate with the blackboard through the CNP agent. Each parallel configuration (kneaders, bagging machines and palletizer in the pasta factory example) has one determinate CNP agent. This agent knows the maximum production rate that the machines in the parallel con-

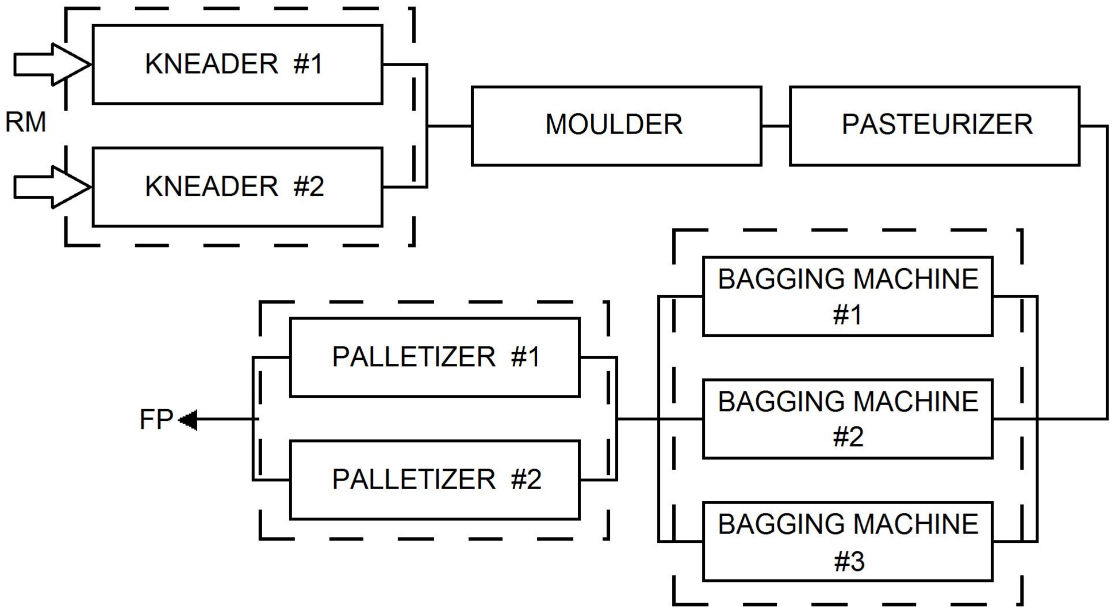


Figure 2.5: *Scheme of the flow of a modeled production line.*

figuration can achieve and it defines the production rate at which each single machine must work. When the blackboard agent receives all of the maximum production rates, it defines the production capacity of the entire line. If the maximum production rate of a machine changes as a result of degradation, failure, replacement or repair, the machine sends a message to the blackboard agent indicating the new maximum achievable production rate and the process of defining the line's production capacity restarts. If a machine in a parallel configuration changes the maximum production rate, it communicates with the CNP, which in turn tries to reconfigure the production rate of the other machines in the parallel configuration to satisfy the production capacity of the line. If that is not possible, the CNP agent sends a message to the blackboard agent to change the production capacity of the line.

Assumptions

The machines in the model are first characterized by periods of uptime when they are working correctly (in nominal conditions), and can then be characterized by periods of time when they are working but not at the expected conditions, and/or by periods when they stop working altogether owing to a failure. These changes affect the failure rate of the machines, and for this reason the presented model can be considered as a semi-Markov process. The assumptions adopted to model and manage the production line are:

- The generic machine can be "in order" (functioning perfectly), "out of order" (not working at all), and "degraded" (functioning but not at a specified nominal level);

- The generic machine may have many levels of degradation that reduce perfect functioning. This degradation occurs according to a degradation rate;

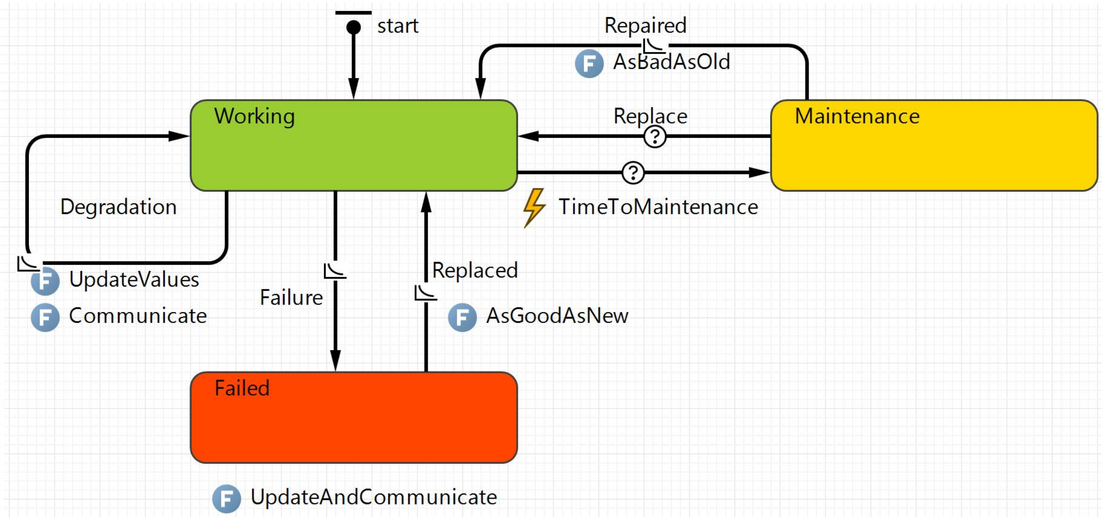


Figure 2.6: *State chart of the machine agent.*

- The generic machine may completely fail at random according to a failure rate, at which point it is replaced;
- The generic machine is considered to be repairable;
- The transition from one state to another is instantaneous;
- Periodic maintenance for the degraded machines is actuated, and could be a repair or a replacement depending on the severity of the degradation;
- The replacement starts immediately after a failure occurs and has a constant duration;
- The generic machine is assumed to be generally “as good as new” at the end of replacement and “as bad as old” at the end of repair activities.

To summarize, the machine is replaced when a failure occurs and can be either repaired or replaced when periodic maintenance occurs. The replacement action renews the machine “as good as new,” while a repair action leaves it “as bad as old.”

Machine Agent

This agent represents the comportment of the machines, particularly their failure behavior. Consequently, there are 9 replications of this type of agent in the pasta factory example, one for each machine in the model. Figure 2.6 shows the possible states of the machine agents. The failure rate captures the machine aging through the time and is used to model only two states: working and not working. In the presented model, the machines may continue to operate in a degraded mode. In this case, the machines continue to perform their functions but at a lower production rate. In reality, although the machine may be function, very often parts are damaged. This type of situation is different from that of a complete failure and occurs frequently in real-world machines. For example, a computer system may not be able to access all of its direct access storage devices, or a multi-engine aircraft may experience a problem in one of its engines.

Initially, the agent is in the working state. As time progresses, the machine can (1) degrade according to a degradation rate; (2) go to the failed state upon a sudden failure in accordance with a failure rate; or (3) go to the Maintenance state, a scheduled periodic maintenance event. We assume that the degradation rate is lower than the failure rate and that there is no relationship between these two rates: when degradation increases, the machine's production rate decreases, maintaining the same failure rate.

As previously mentioned, the machine performance changes from the nominal production rate when *degradation* occurs. The degradation rate $Y(t)$ used in this model is derived from the study conducted by Li and Pham [129], Eq. 2.11:

$$Y(t) = A + Bg(t) \tag{2.11}$$

where $A > 0$ and $B > 0$ are independent stochastic variables and $g(t)$ is an increasing time-dependent function. *A* measures the initial value of degradation and B represents the variations of the an increasing function $g(t)$. Degradation influences the machines' production rates. To model this influence, a random degradation factor less than one was introduced. When degradation occurs, the machine's production rate is multiplied by the degradation factor.

Simultaneously with degradation, a failure can completely halt the productivity of a machine. This event has been modeled using the transition from the Working state to the *failed state*, a transition that is triggered according to a failure rate. In the proposed model, the failure rate of each machine is composed of two parts: the first depends on the working time of the machine, and the second depends on the number of repairs that the machine has undergone. The working time indicates how long the machine is working and changes the type of failures to which the machine is subjected, and consequently the failure rate (Figure 2.3). The *Weibull distribution* is used to model a wide range of failure types, covering all of the different zones of Figure 2.3, simply by changing two parameters, the scale parameter α and the shape parameter β, which are fairly easy to determine [19]: infant failure, when $\beta < 1$, random failure when $\beta = 1$, and 'wear out' failure when $\beta > 1$, with complete predictable failure when $\beta > 3$. When a new machine is modeled, parameter α of the Weibull distribution is random between 20 and 50, while parameter β increases over time, varying between 0.1 and 3, to simulate all of the states of the bathtub. Differently, the existing machines in the pasta factory were simulated using provided real data. Starting from the historical failure time, it is possible to calculate the parameters of the Weibull distribution that best fit the real set of failure times. Using the method presented in [175], parameters α and β were chosen to model the failure rates of the machines in the example. Eq. 2.12 shows the failure rate used in the presented model:

$$\lambda(t) = W(t) \cdot Rep \tag{2.12}$$

where $W(t)$ is the Weibull distribution and Rep is the maximum value between 1 and the number of repairs divided by a smoothing factor greater than one. After a failure, the machine is replaced, considering the time to replace as constant. The replaced machine is considered "as good as new;" to model this situation, the degradation and numbers of repairs are restarted at 0 and the β parameter of the Weibull distribution restarts from 0.1.

As in the last analysis, time-based preventive maintenance was modeled. Maintenance is then due every tp periods; for this reason, it was important to retain the most recent maintenance and replacement times in two variables. The machine can transfer from the working state to the maintenance state according to an event called *TimeToMaintenance*, which works in parallel with the state chart to calculate when machine maintenance is due. In the maintenance state, the machine can be repaired or replaced according to a stochastic variable PoR. This variable linearly depends on the machine's repair history: the greater the number of repairs, the greater the probability that the machine must be replaced, as shown in Eq. 2.13:

$$PoR = Rp + NoR \cdot Ir \tag{2.13}$$

where Rp is a number chosen between 0 and 1 representing the tendency of a machine to be replaced; NoR is the number of repairs that the machine has undergone; and Ir is a [0, 1] value that represents the impact of the repairs on the probability of replacement. These values should be chosen based on the machine's replacement history and the experience of the maintainers of the line.

In the case of a replacement, the transition to the working state occurs "as good as new," as after a failure. Repair, on the other hand, represents cleanings, lubrications, alignments, and so on: minimal repairs that leave the machine "as bad as old." The duration of the repairs for a specific condition of a given degraded machine is stochastic, but in general, it increases as the machine ages according to a repair rate. In the presented model, the repair rate is a variable value that increases with aging and the number of failures. The model can harness several distributions that are commonly used to approximate the time to repair, i.e., the Normal, the Weibull and the Gamma. In the example, starting from the real times to repair provided by the pasta factory, the distributions that best fit the data for all machines were searched. In this case, all of the machines have the time to repair that best fits the Normal distribution. The repair rate $\mu(t)$ is shown in Eq. 2.14:

$$\mu(t) = f(t) + NoR \cdot Irep \tag{2.14}$$

where $f(t)$ is the function that best fits the times to repair and $Irep$ is a [0, 1] value that represents the impact of repairs on the repair rate. Thus, repair activities depend on the machine's repair history. The minimal repair action brings the machine to its previous operational state without affecting its failure rate. To model minimal repair, the number of repairs is not reset, although the degradation is reset because the maintenance activities settle the degradation. This means that the machine returns to produce at nominal production rate. Lastly, the β parameter of the Weibull distribution continues its variation to 3, continuing from the same value as before the maintenance.

ContractNetProtocol Agent

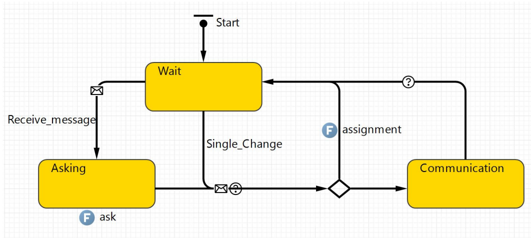
The previous subsection presents the behavior of single machines whose particular functional conditions and reliability are known. The performance of the set of machines that comprise the production line, however, results from how they interact with each other to fulfill the required tasks. The interactions between machines belonging to the same parallel configuration are modeled with the support of a contractNetProtocol (CNP) agent. The behavior of this agent is inspired by the contract net protocol originally proposed in [197], which is widely used in resource and task allocation problems in many different fields, including flexible manufacturing [155]. In the contract net protocol, each node on the network can at different times or for different tasks be an initiator, a participant or both. When a node receives a complex task, it announces the problem to the contract net, acting as an *initiator*. Bids are then received from potential *participants* and the winning participant(s) are awarded the job(s). In this model, the nodes in the contract net protocol are identified as the machines in the parallel configuration. To maintain the behavior of the machines and their coordination separately, a CNP agent has been created. When the production rate of a machine inside the parallel configuration changes, the machine agent recognizes the change and communicates with the correspondent CNP agent. At that moment, the CNP agent acts as an initiator, announcing the problem to the machine agents in the parallel configuration (i.e., the control net). Each machine agent in the parallel configuration bids its maximum achievable production rate. The CNP agent assigns the machines in the parallel configuration according to their capacities and with the goal of maintaining aligned consumption.

There is one CNP agent for every parallel configuration in the production line. Looking at the Figure 2.5, it is clear that there are 3 CNP agents in the pasta production line: one to manage the kneaders, one for the bagging machines, and one for the palletizers. All of the actions of the CNP agent presented below are referred to the group of machines that each single agent manages. The term *parallel configuration* will be used to refer to a group of machines in the same parallel configuration.

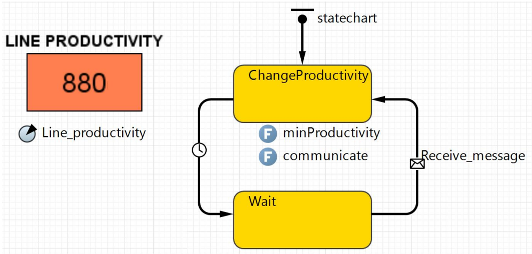
There are two different tasks that the CNP agent must do: (1) to know the maximum production rate that the parallel machines can achieve; and (2) to impose the production rate of each machine to guarantee the production capacity of the entire line. Concerning the maximum production rate guaranteed by the parallel configuration, the CNP agent sends a message to each machine in the parallel configuration, asking for the maximum productivity that each single machine can achieve. The maximum production rate of the parallel configuration is the sum of the single maximum productivities. In this way, if a machine is in the failed state, it sends a production rate equal to zero to the corresponding CNP agent, but the production rate of the parallel configuration may satisfy the line production capacity and it may not be necessary to stop the entire line. The other task concerns the enforcement of the production rate of each machine in the parallel configuration. The CNP agent ensures the equal consumption of the machines in parallel, using the formula expressed in Eq. 2.15:

$$PR_i = \frac{ProductionCapacity \cdot Availability_i}{\sum_j^M Availability_j} \tag{2.15}$$

where PR_i is the production rate of machine i; $ProductionCapacity$ is the production capacity of the entire line that is read in the blackboard, as explained below; M is the number of machines in the parallel configuration; and $Availability_i$ is the maximum production rate achievable by machine i. Initially it is equal to the nominal production rate but due to degradation, the availability of each machine can decrease. The state chart of the CNP agent is shown in Figure 2.7a. Each CNP agent begins in the Wait state, from which there are two transitions: *Receive_message* and *Single_Change*. Receive_message transitions when the CNP agent needs to know the maximum production rate of the parallel configuration. It occurs when the production rate of a machine in the parallel configuration changes, at which point the CNP agent needs to know the new maximum production rate of the parallel configuration. On the other hand, if the pro-



(a)



(b)

Figure 2.7: *State charts of the (a) CNP agent and (b) Blackboard agent.*

duction capacity of the entire line changes and the CNP agent already knows the maximum production rate achievable by the parallel configuration, it transitions directly to the "rhombus" of the state chart through the Single_Change. The rhombus in the state chart (Figure 2.7a) represents a crossroads: if the production capacity of the line is less than or equal to the production rate achievable by the parallel configuration, the CNP agent imposes the single production rates of the machines in the parallel configuration. Otherwise, if the parallel configuration is not able to guarantee the production capacity of the line, the CNP agent transitions in the *Communication* state. In this state, the CNP agent sends a message to the blackboard agent to communicate the highest achievable production rate by the parallel configuration. In either case, the CNP agent returns to the Wait state, where it waits until the production rates or capacity change.

Blackboard Agent

Blackboard systems are commonly used in the multi-agent systems research community to tackle problems related to the characteristics of uncertainty and non-deterministic behavior. The presented model uses a blackboard agent whose behavior is inspired by the blackboard architecture to model a system able to coordinate itself, regardless of the number or characteristics of the machines in the production line. Figure 2.7b shows the state chart of the blackboard agent. In the *ChangeProductivity* state, the blackboard agent asks the machine agents and CNP agents the maximum production rate achievable by the machines in the line. In the example of the pasta production line, the blackboard agent sends a message to the machine agents that represent the molder and the pasteurizer and to the three CNP agents for the kneaders, bagging machines and palletizers. When the blackboard agent receives the maximum production rates, it chooses the lowest as the production capacity of the entire line. The production rate at which the entire line must produce (the production capacity) is "written in a blackboard" so that the machine agents and CNP agents can read it. In this way, these agents can adapt their productivity depending on the objective to be achieved. The blackboard agent then transitions to the *Wait* state. When a machine or a parallel configuration changes its maximum production rate, the corresponding agent sends a message to the blackboard agent indicating that the productivity has been modified. This modification could be a decrease due to degradation or failure, or an increase due to maintenance activities or a replacement. When the blackboard agent receives a message, it transitions to the ChangeProductivity state, asking for the different maximum production

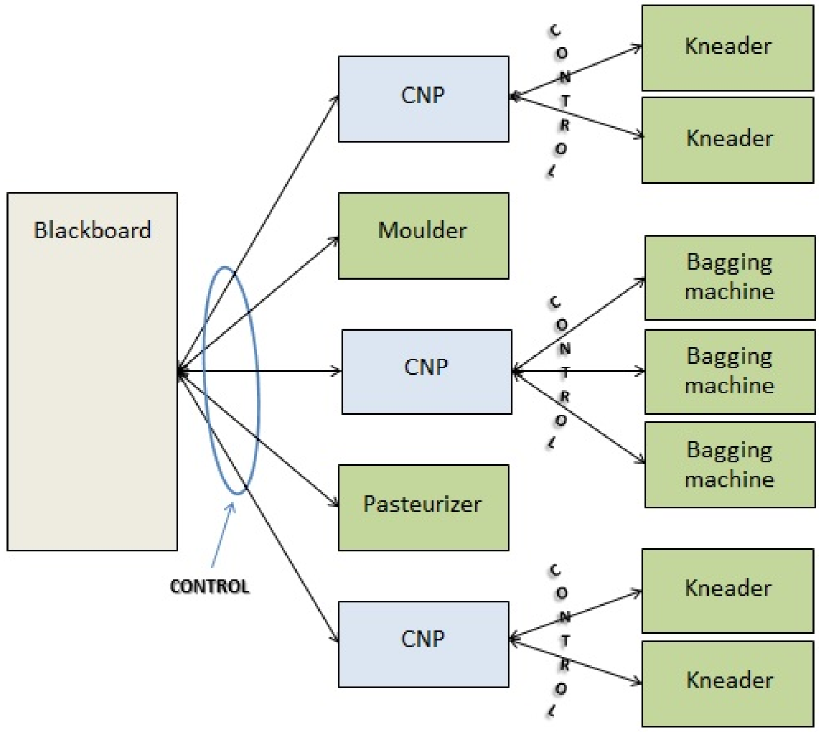


Figure 2.8: *Communication flow in the model.*

rates and restarting the process of establishing the line production capacity. In this way, the blackboard agent is able to define the production capacity of the production line and the communication between agents allows a simulation without a hierarchical process, as shown in Figure 2.8.

2.2.4 Discussion

This section presented an agent-based model to simulate repairable manufacturing systems. The agents approach has allowed to model two levels of complexity: the number of machines within the production line and the influence of several factors in the failure behavior of the machines. In fact, the presented model considers simultaneously the production at different rates, deterioration, repairs, replacements and preventive maintenance. The production rates were coordinated at different levels. The consumption of the machines in a parallel configuration is maintained and aligned so that machines of the same age will not have different consumptions. In addition, the production rate of the line is organized using a blackboard. In the model, the stochastic deterioration of the machines over time influences their production rates. The failure rate is different for each machine and depends on the age of the machine (the working time) and the number of repairs that the machine has undergone. After a complete failure, a replacement of the failed machine was modeled holding the replacement time constant. Preventive maintenance activities were also modeled to increase the production rate of the degraded machines. Maintenance activities can comprise replacement or repairs; the latter are actions that increase the production rate but do not influence the aging and number of repair that the machine has undergone.

2.3 Planning and scheduling

Planning and scheduling are forms of decision-making that are used on a regular basis in many manufacturing and service industries. They are the processes of determining the sequential order of activities, assigning planned duration, allocating resources, controlling and optimizing the advancement of these activities. These processes have to be done in such a way that the company optimizes its objectives and achieves its goals. Resources may be machines in a workshop, vehicles at a bus station, crews at a construction site, or processing units in a computing environment. Activities may be operations in a workshop, departures and arrivals at a bus station, stages in a construction project, or computer programs that have to be executed. Each activity may have a priority level, an earliest possible starting time and/or a due date. Objectives can take many different forms, such as minimizing the time to complete all activities, minimizing the number of activities that are completed after the committed due dates, and so on [166]. Often in a company, planning and scheduling functions rely on mathematical techniques and heuristic methods that allocate limited resources to the activities to be done.

Planning and scheduling in either a manufacturing or a service environment have to interact with many other functions. These interactions are typically an exchange of information between decision making functions of the analyzed system. Historically, planning was mainly related to the manufacturing sector, but today the concept of planning can be applied to very different fields [90]. In this section, an agent-based model able to automatically produce a scheduling which consists of three interrelated components: (1) creation of timetables; (2) scheduling vehicles to trips; and (3) assignment of drivers. This model is analyzed in deeply in [7], and due to its flexibility is particularly adapted to be used in dynamic environments, such as the planning of public transportation in *developing countries*. Planning in public transportation–generally buses in developing countries–has the objective to maintain the service accessibility and availability, considering social goals and company requirements. From the company point of view, operational planning span short-term decisions focus on minimizing the cost related with the usage of vehicles, fuel consumption, and drivers' wages [101]. The problem faced in this section can be defined as the creation of *bus schedules* under company requirements in terms of costs and offered service. This problem has been faced from a long time, earlier manually, then following models in operations research [223], and now by computer-based approaches. Due to their characteristics, typical solutions cannot be applied by the majority of companies of public transportation in developing countries; principally because of (1) high costs derived from the purchase of softwares, learning requirements of the planners, and long waiting time to define optimal solutions; and (2) high dynamism of the environments, e.g., the use of old vehicles, scarce state of the roads, and limited traffic regulations. The combination of these issues is a trigger to create a model that easily adapts to different situations, able to ensure reliable results even with incomplete inputs.

2.3.1 Operational planning in transportation

Public transportation is considered an important backbone of sustainable urban development, since it should allow more efficient movements across a city. Public transport has to provide a good level of service at an affordable cost for the users, guaranteeing a profitable system for the companies where costs of vehicles usage and drivers wages are low. Then, the efficiency of a transport system depends on several elements, such as available technology, governmental policies, the planning process, and control strategies. The interaction between these elements is quite complex, leading to intractable decision making problems. This section focuses on the planning process which spans every decision that should be taken before the operation of the system. Planning in transportation is commonly divided into tactical, strategical, and operational decisions [46]:

- **Transit network design**, defines the lines layouts and derived characteristics, e.g., rolling stock types and space between stops, in order to optimize specific objective functions;
- **Frequency setting**, determines the number of trips per hour needed to satisfy the passenger demand (morning and afternoon peaks, and so on);
- **Transit network timetabling**, defines departure and arrival times of buses in order to achieve different goals, such as to guarantee a given frequency, and minimize waiting times;
- **Vehicle scheduling**, determines the trips-vehicles assignment to cover all the planned trips such that operational costs based on vehicle usage are minimized;
- **Driver scheduling**, defines daily duties that cover all the scheduled trips and minimize the cost of driver wages. A solution of the DSP must satisfy specific labor regulations for drivers such as minimum/maximum work length, maximum working time without a rest, and daily rest for all drivers;
- **Driver rostering problem**, given a set of generic duties defined over a certain time horizon for the drivers assigned to a particular depot.

The interdependence between these sub-problems is shown in Figure 2.9, taken from [101].

Unfortunately, the traditional solutions for the operational planning problem are difficult to be applied in developing countries, because to find an optimal solutions is necessary a high level of *accuracy* in input data, and the solutions found are often too "hard" to be applied in an environment extremely dynamic. Developing countries indeed are even characterized by difficulties in obtaining reliable data. In addition, according to Iles [102], the factors that hamper the development of the automatic planning methods are:

- The poor roads conditions, "peculiar" driving style of the drivers, and overexploitation of buses affect the number of daily vehicles available;
- The scarce road regulations and continuous manifestations that interrupt the roads, affect even the vehicle routing that change almost every week.

In this context, it is necessary to rely in a model that can easily adapt to different situations that occur every day and capable to ensure good results even with incomplete inputs. A multi-agent model which focuses on finding a sub-optimal solution by modeling a relaxed environment can be useful in this case, due to its ability to ensure cost-effectiveness, speed and flexibility.

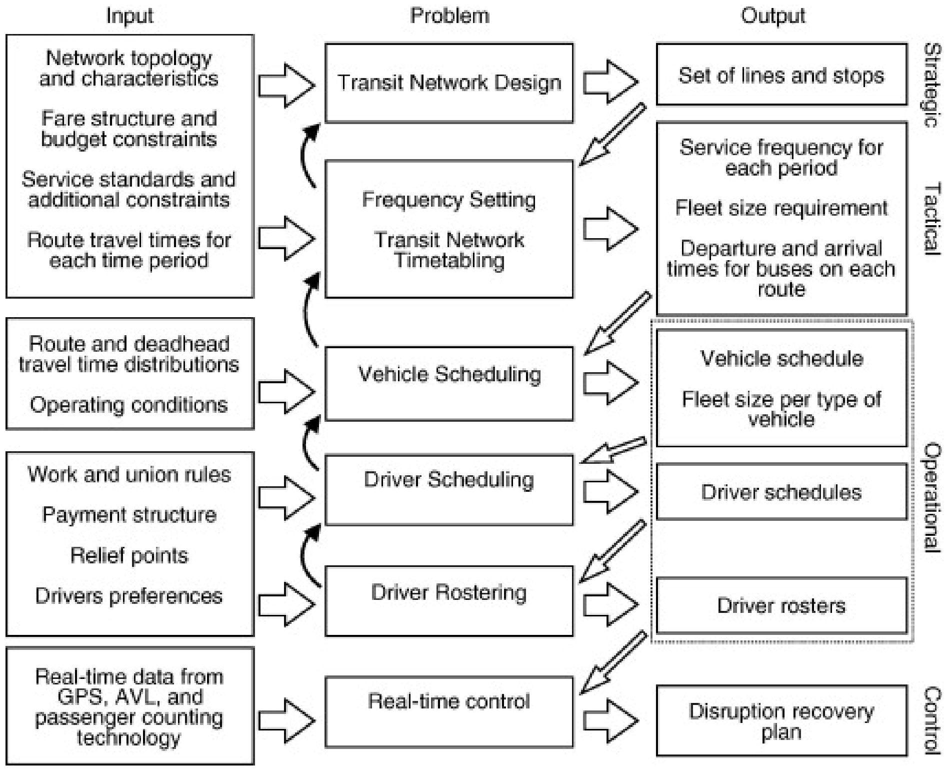


Figure 2.9: *Interaction between sub-problems of the planning process [101].*

2.3.2 Autonomous agents approach

According to a report from the University of Texas [209], in the sixties almost all the operational planning and the consequent scheduling were still done manually. Since the average age of human schedulers was increasing and few people desired to undertake the scheduling career, researchers were called to automate the planning process. The created methods, initially, tried to simulate the work of manual schedulers and were based on heuristics since the integer linear programming models were not able to solve models of realistic size. Unfortunately, due to the cost/performance ratio of the computers of the time and the inefficiency of the solutions, the introduction of information technology has not yielded the expected results. Over time, the computers' cost decreased, their performance increased, and the solutions improved, combining heuristics and mathematical programming. Nowadays, the largest bus companies rely on computers for their scheduling, with a significant improvement of the productivity and a more efficient operations. The dynamic nature of the problem and the wide applicability in the field of transportation, make the planning and scheduling an NP-hard problem with highly combinatorial characteristics. This means that the computational burden to solve these problems increases exponentially as the problem size increases. Different solution methods have been developed, such as heuristics [92], set covering formulations [195], genetic algorithms [224], branch-and-bound [91], and more recently "sharing-sweets-and-sour" [1], multiple depot vehicle scheduling [95], multi–objective genetic algorithm [238], clustering-based method [236], among others. These methods solve the vehicle scheduling problem, the driver scheduling problem, or both problems simultaneously.

As explained, developing countries are characteristics by an uncertain context, where solutions methods have to be adaptive and flexible. The agent abstraction is based upon the concepts of reactivity, autonomously, and proactivity that inhabit a dynamic environment. Several authors [59, 235] used multi-agent models to simulate and plan railway schedules. Problems are solved by the interaction of many agents in a cooperative way. Differently from traditional, centralized, hierarchical, sequential, batch scheduling systems, multi-agent systems are able to respond to any events very quickly and flexible by adjusting conflicts and rebuilding chain of links in scene, which represents formal specification of situation. In addition, they can be able to improve results proactively if there is enough time for results improvement [104]. Another important aspect of multi-agent systems is related to their adaptability to form schedules, that means that schedules are not necessarily created from the beginning every time but information can be updated and executed on a rolling basis of events. For these reasons, a multi-agent model which focuses on finding a sub-optimal solution by modeling a relaxed environment was developed, leaving to the human planners the eventual final adjustments.

2.3.3 Method

This section analyzes a case study of a real transportation company in the Autonomous City of Buenos Aires, Argentina. In this country, it is possible to find the characteristics above mentioned, especially the use of manual planning is extremely common. Also the information that may seem constants have to be considered as variables with high volatility, in a country like Argentina. The developed model wants to be a tool of automation and optimization for the integrated allocation of departures time and the respective vehicles and drivers. *Automation* to provide more efficient schedules and reduce costs for vehicles and operators; and *integration* to achieve a global cost reduction, considering aspects derived from different areas with different objectives. One important objective of the model is to grant flexibility independently of the presence or not of some inputs. With these objectives in mind, the developed model creates a series of agents-optimizers with their own schedules, which work in parallel in a bio-inspired way by trial and error method. In this way, the model finds different solutions, eliminating premature convergences and evaluating the solutions in order to eliminate those that do not achieve certain objectives. To do this, an ant colony algorithm [67] was adopted. This algorithm is an evolutionary optimization inspired by the behavior of ant colonies where ants are able to find the optimal route to reach the food tracing random paths and exploiting the releasing of some pheromones. At the end, the shortest path has the strongest smell of pheromone and is the most attractive for other ants as well.

Based on this theory, a model that seeks along various solutions creating various schedules in probabilistic way was developed. The created schedules respect the requested frequencies, work rules and optimize the drivers' costs. Creating several random schedules in terms of departure times, vehicles and drivers, the model evaluates the best solution in term of cost and supplied services. This solution is associated to the ant that finds food and leaves pheromone for the others. Iterating the creation process again, the departure times, vehicles and drivers are chosen randomly, but this choice will tend to follow the "pheromone", like in an ant colony. It means that the creation process will be affected by the choices occurred to arrive to the previously best solution. Iterating the process again and again, evaluating every step the created solutions, the best schedules will emerge. That solution will not be the optimal solution for the analyzed problem, but it will be an acceptable sub-optimal solution. To avoid convergences in local solutions not particularly suitable, the pheromone evaporates with time, affecting a limited number of subsequent choices. The best solution is chosen considering a trade-off between the schedules' adaptation and cost, according to some parameters defined by the human schedulers.

The model is composed of four type of agents: *allocators*, *evaluators*, *seeds*, and one *mother matrix*. Conceptually, the model generates a series of seed agents, i.e. generates times of departures scheduled randomly, under the frequencies imposed by the mother matrix agent. Each seed agent generates its own schedule of departure times which has to be satisfied. At this point, some allocator agents are associated to each seed agents. It means that the allocator agents associated to one seed agent have to assign vehicles and drivers to the departure times of that seed agent. Once drivers and vehicles were assigned for all departures, the schedule is considered *closed*. These created schedules are evaluated by the evaluators agents, considering different measures explained in following. There is one evaluator agent associated to one seed agent, therefore one evaluator agent compares the schedules of the allocator agents associated with the same seed agent. Then, evaluator agents define the best schedule for the allocator agents associated and leave the pheromone to the corresponding seed agent, and the process is iterated. From the second iteration, allocator agents assign vehicles and drivers in a probabilistic way but influenced by the assignments made for the best schedule. After a certain number of iteration, the best schedules of each evaluator agent is sent to the mother matrix agent, where they are all evaluates. The mother matrix agent selects one, the *best schedule* in absolute. The model uses some parameters to evaluate the generated solutions, for instance sometimes it is preferable to have a schedule with a lot of overtime and use few drivers, or in other cases it is preferable to have a schedule with limited used vehicles and a lot of driver working hours. At the beginning, these parameters can be chosen heuristically and then being optimized depending on the needs of the various lines, but this is out of the scope of this section. The flowchart of the presented model is shown in Figure 2.10.

Input

The principal input for the model are the desired frequencies for each hour of the day. In this way, the model can create the departure times to start the schedule. Some other preliminary data should be reported, in order to define the specific scenario:

- Number of available vehicles;
- Number of available drivers;
- Travel times for each hour of the day;
- Drivers information: dwell time of the drivers at depot, work and union rules, payment structure, and so on.

The first step of the model is to create a series of departures time that fulfill the frequencies in the mother matrix agent. Seed agents generate the departure times in a

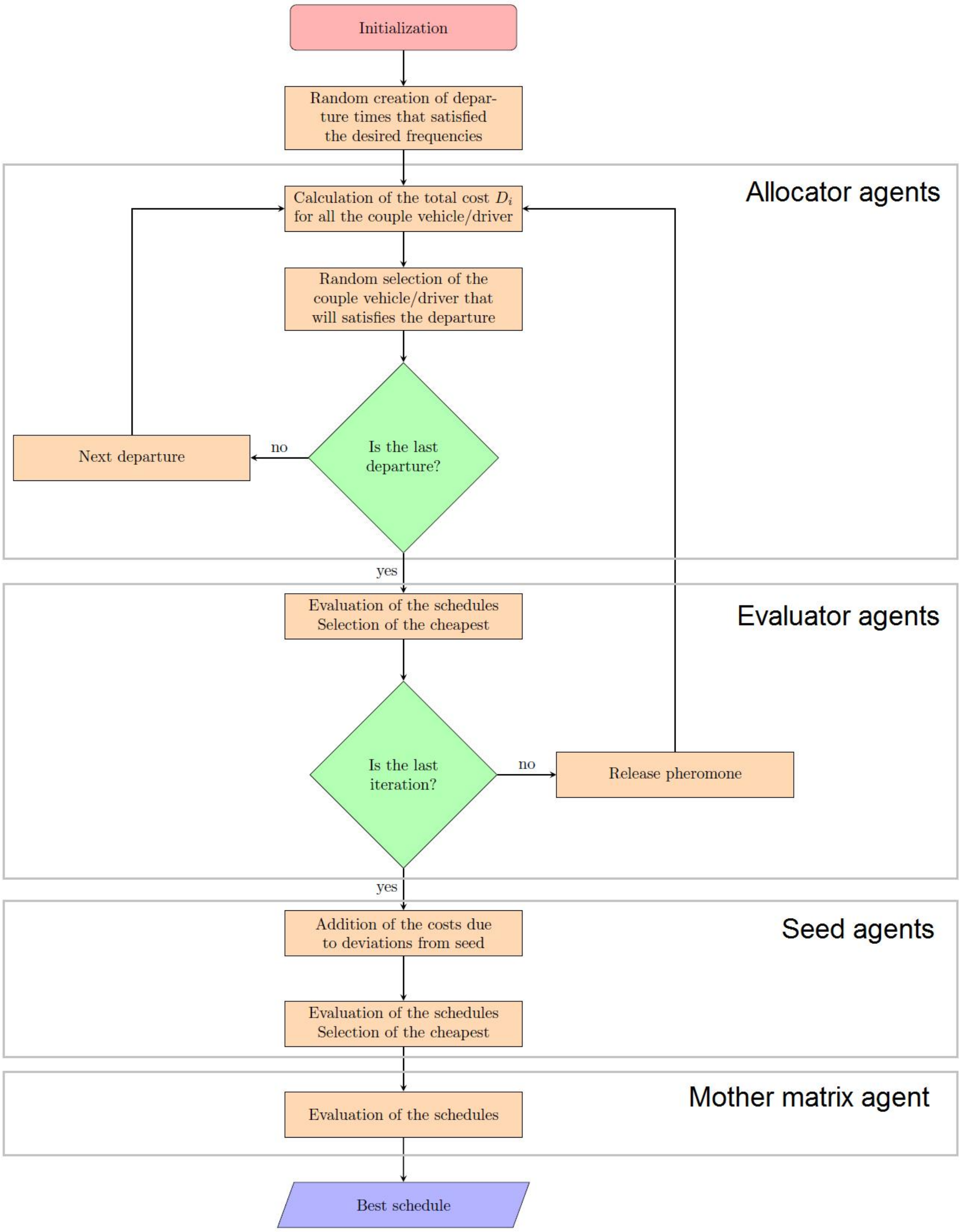


Figure 2.10: *Flowchart of the proposed model.*

probabilistic way, as explained in the following. Once the departure times are created, allocator agents allocate vehicles and drivers to each of these departures.

Allocator agents

Allocator agents are probably the most important agents in the model. This because their main task is to create the final schedules. The larger the number of allocator agents the higher the probability to find the optimal solution. On the other hand, the larger the number of allocator agents the higher the computational time of the model. The company where this study was carried out needed a model that produces results in about 10–15 minutes. In this case, the number of allocator agents can be decided depending on the execution time and less on the degree of optimization of the solution. Once the send agent send the departure times to the allocator agents, the allocation process considers one departure at a time. Initially the first departure to be considered is arbitrarily selected as the first in the morning. This choice is completely arbitrary, for instance it is also possible to start to ensure the departures at the hours with higher frequency and then assigns others consequently. For each departure, each allocator agent considers all vehicles and drivers available and chooses probabilistically who and which bus satisfied this departure. The allocator agent can also choose to delay the time of a departure, in order to allow the return of a vehicle in service. This delay has a cost and can change the hourly frequency of the departures, but it is an opportunity that the model can choose and evaluate. The probability to choose between the different available driver/vehicle couples depends on their cost: the greater the cost of the couple, the smaller the probability to choose them. The probability P_n for any possible couple n which will satisfy a departure is expressed in Eq. 2.16.

$$P_n = \frac{\frac{1}{D_n}}{\sum_{i=1}^{n} D_i} \times \frac{1}{(n-1)} \tag{2.16}$$

where D_i is the total cost of the couple i in case it is chosen to perform the departure. D_i is explicit in Eq. 2.17.

$$D_i = \sum_{c=1}^{5} \beta_c \times cost_c \tag{2.17}$$

where β_c are the weights of the following costs considered in the model:

1. *Taken cost* is the cost for sending a vehicle in a street if it has not exited the depot yet. This cost was introduced to avoid an early driver change before the approaching end of the shift. In this case, it was set equal to the hourly cost of a driver multiplied by the number of hours of the shift;

2. *Change shift cost* is the cost that is necessary to spend if the model chooses to change the driver of a vehicle. In this case, a driver must drive the same vehicle for all shift;

3. *Delay cost* quantifies the cost of delaying a departure from the established one in the seed;

4. *Overtime cost* is the cost for the hours worked out of the work shift;

5. *Idleness cost* quantifies the waiting time of a driver in the depot for the departure (being within his shift, it is regularly paid).

Due to a series of union agreements, in the Buenos Aires area the only *relief point* (i.e. a point along a route where a driver may leave or take over a bus) is the depot. Moreover in this area each vehicle is associated to only two drivers. This principally occurs for two important aspects: (1) buses in Argentina are really *folkloristic*: a vehicle is like an office for a driver who spends his own money to decorate the interior with lights, photos, and other stuff; (2) the kind of driving that drivers in South America have: if a vehicle is associated only to two drivers, they will take care of it driving better. In addition in this way the company can know the responsible in case of mechanical breakdowns.

The third cost to be incurred for each departure is the delay cost. This cost plays a decisive role at the time of choosing the couple driver/vehicle, since it quantifies the ability of the model to delay or not the departure times present in the seed. As mentioned above, when a couple driver/vehicle have to satisfy a specific departure, the allocator agent can wait for the arrival of a vehicles already in circulation, thus delaying the expected departure. The delay has a cost that varies depending on the frequency to maintain at that hour, e.g., delaying by 10 minutes a departure at 3AM with one vehicle in circulation should be cheaper than delaying of 10 minutes a departure at 8AM when the vehicles start from depot every 5 minutes. Moreover, delay cost should be of the same order of magnitude as the taken cost, because the choice of delaying will have to “compete” in the first instance with the possible departure of vehicles with drivers which have not started the work shift yet.

Pheromone is the last factor that influences the choice of the couple driver/vehicles. The presented costs influence the probability of each possible choice, but for a certain percentage the choice is affected by the choices made to achieve the schedule with the lowest cost in the previous iteration. At the end, when all the departures are satisfied the schedule is considered closed. Each allocator agent has produced in output a complete schedule, containing all relevant information to be evaluated: the departure times, vehicles and drivers assigned to those departures, but also the cost of each departure considering overtime hours and night working hours which can be calculated. Each allocator agent sends the own schedule to the corresponding evaluator agent, who will evaluate it.

Evaluator agents

Evaluator agents differ among themselves for the assignment of some parameters. The model uses weights, called βs, to give more or less importance to the five costs seen previously. βs of each evaluator agent are probabilistically chosen inside a defined range. These βs determine the importance of the different costs when the model chooses between different alternatives, e.g., if it is more or less important to consider the costs of changing working shift rather than the cost of overtime.

Evaluator agents have the task of evaluate the schedules in exit from the corresponding allocator agents. This evaluation is called $Assessment1$, which considers costs and desired frequencies, leaves the pheromone for next iterations, and at the end of the iterations selects the best schedules to send to the corresponding seeds agent. The words "leaving the pheromone" mean that the model saves the couples driver/vehicle chosen by the allocator agent that gave rise to the lowest cost scheduling. As mentioned before, for every departure each allocator agent can choose to choose this saved couple or another one in a random way. Eq. 2.18 shows the $Assessment1$ used to evaluate the schedules produced by the allocator agents.

$$\begin{aligned} Assessment1 = {} & \alpha_{costs} \times schedule\_total\_cost+ \\ & + \alpha_{frequencies} \times \Delta_{frequencies}+ \\ & + \alpha_{vehicles} \times taken\_cost \end{aligned} \tag{2.18}$$

This Eq. 2.18 can be seen as composed of three aspects:

1. *Calculation of cost*, the schedule cost derived by the allocator agents choices, such as overtime, hours at night, and so on;
2. *Calculation of deviation*, comparison of the hourly frequencies, it is the difference between the desired frequency in the mother matrix agent and the frequency generated by the allocator agents;
3. *Calculation of work cost*, the costs of drivers called to work.

The schedule with the lowest value of $Assessment1$ is considered the winner. In the next iteration the choices for selecting couples drivers/vehicles for each departure will influenced in some percent by the choices of this winner schedule. These iterations continue up to a maximum number of iterations or they stop if the winner is the same schedule for a certain number of iteration in a row, because in this case the model cannot improve the solution. The last task of the evaluator agents is to send the best schedule found to the corresponding seed agent.

Seed agents

Seed agents have two tasks: at the beginning they have to generate the departure times, and then, after the end of the iterations, they have to evaluate the solutions received from the evaluator agents. The creation of the seed agents is the first step of the model. After that, each seed agent creates the departure times, considering separately each hour of the day. So, it does not matter what time and how many buses departed the hour before or will do the hour after, because the desired frequency is defined hourly. This can be seen as a limitation of the model, however, human planners in our case study think in the same way even when they plan manually. Each hour is divided by the number of desired frequency, creating in this way a series of time slots where the departures will be designed. The first departure is designed in a probabilistic way inside the first time slot, according to a normal distribution of mean μ and sigma σ. In this way the model has a higher chance to design the departure at the same temporal distance, but it is not obviously a guarantee. Once the seed agent has chosen the first departure time, the next departure will be designed in a time slot that starts at the previous departure time and ends at the end of its predetermined time slot. Figure 2.11 shows graphically the process to choose the departure times where the desired frequency is equal to two. This

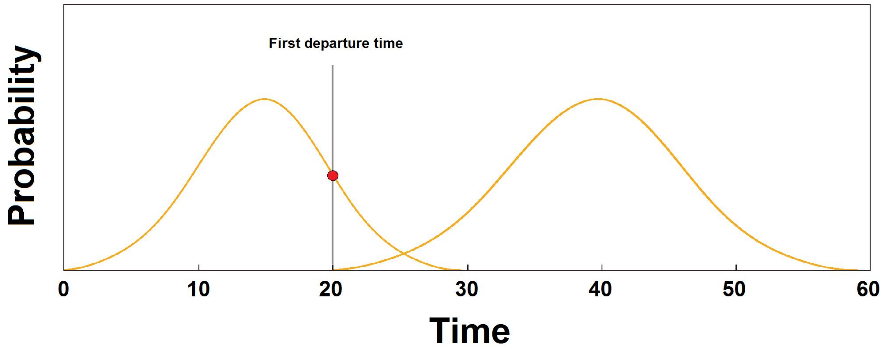


Figure 2.11: *Example of choice of departure time.*

approach may appear not sophisticated, but it is important to remember the context of the typical metropolis of developing countries, where the traffic conditions are not much regulated and extremely uncertain. In Buenos Aires for instance, it is very common to see two or three buses arrive at the bus stop at the same time. This is not only a planning problem, but it is also due to terrifying conditions of the streets. A perfect schedule is inapplicable in these roads because the conditions really change every hour of the day. This is the reason of the selected "hour by hour" approach, regardless of what happens in the hours before and after. The main idea of the model is to generate a large number of seeds, in order to increase the probability to find the optimal solution.

The second task of the seed agents is to evaluate the schedules in output from the evaluator agents. This evaluation considers the deviation of the created departure times respect those in the starting seed. The model defines a delay tolerance respecting the desired time, which is expressed in a ratio minutes/frequency. This is because minutes have different weight depending on the frequency, e.g., a delay of 10 minutes when it is necessary to guarantee a frequency of 12 vehicles per hour, equal to a departure every 5 minutes, is different respect a delay of 10 minutes when the desired frequency is one vehicle per hour. This tolerance allows to define the rigidity of the service level (i.e. frequencies or departures per hour) of the solutions. In this case, the tolerance was defined empirically together with the transportation company. To have a satisfactory schedule for the human planners, this tolerance was defined in the same order of magnitude as the total cost. If the difference between the scheduled and desired departure times is greater than the tolerance, the cost defined in Eq. 2.19 is added to the schedule.

$$Additional\_cost = \frac{total\_cost}{tolerance} \times (delay - tolerance) \tag{2.19}$$

where $total\_cost$ is the total cost of the schedule, i.e. the sum of costs of single departures; and $delay$ is the delay of the single departure expressed in minutes. This additional cost is used to adapt the schedules to the desired characteristics. Once the schedules are aggregated or not of additional costs, seed agents send the schedules total cost to the mother matrix agent. Then the mother matrix agent will evaluate the schedules and choose the schedule with the lowest cost as the best schedule.

Mother matrix agent

This agent reads the input data and create the other agents. Having the frequencies that should be guaranteed, the mother matrix agent evaluates the schedules received from the seed agents, considering a weight θ for the total cost of the schedule and the deviation from the desired frequencies. Based on these parameters, the schedules are evaluated finding the best one. This last evaluation is called $Assessment2$, where all the variations of frequency are considered in the same way, no matter they occurred in time of high or low frequency, as defined in Eq. 2.20.

$$Assessment2 = \theta_{frequencies} \times \delta_{frequencies} + \theta_{costs} \times schedule\_total\_cost \quad (2.20)$$

In Eq. 2.20 $\theta_{frequencies}$ and θ_{costs} are numbers in the range [0,1] and define the importance of the total cost and frequencies deviations; while $\delta_{frequencies}$ is the sum of the differences for all 24 hours of the created frequencies and desired frequencies.

Results

Different performance indicators were defined to assess the proposed model. Defining such indicators is not an easy operation because the performance of the model is closely linked to the setting of the parameters. The tradeoff is principally between the freedom of the model to differ from the desired frequencies or not. In any case, the follow indicators were used during the phases of model launching, setting the parameters to obtain the schedules with the desired flexibility of the company. The results of the model are compared to the actual results of the human planners.

1) Number of used vehicles. The model can be used only to define the number of vehicles needed to close the schedule, guaranteeing a certain level of satisfaction, intended as guaranteed frequencies. This information can help the decision making process of the human planners, which can do different strategic considerations, e.g., from customer satisfaction to maintenance optimization.

2) Respect of frequencies. This indicator concerns frequencies, Figure 2.12 shows the difference between the desired and modeled frequencies of a line taken as example. The results are related to the selected settings. However it is recommended to try various settings to find the one more suitable to the analyzed real case. Figure 2.12 shows how the model unlikely increases the service frequency, in the sense of increasing the frequency of offered services in the different time slot. The model starts to schedule from the beginning of the day, excellently planning the departures in the early hours, but it loses the peak of the six in the morning, where it is notable that the application does not reach the desired service.

3) Quantity of vehicles for every time. Figure 2.13 shows the number of vehicles on the street in each hour, number of vehicles in the depot, and desired frequencies. The results of this indicator are important because they show how the model slightly late to get the operating limits of the vehicles, in the middle of the day it is able to respect all the departures, exploiting also the entire fleet if necessary. One of the main problems in planning operations occurs when it is necessary to assign many departures in the same temporal slot, with a consequent excessive availability of resources in the following hours. Also the proposed model tends to ensure more night services than required, as shown in Figure 2.13.

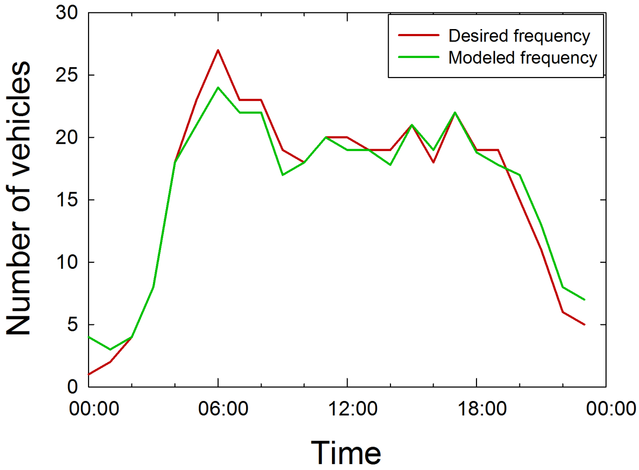


Figure 2.12: *Desired frequencies and modeled frequencies.*

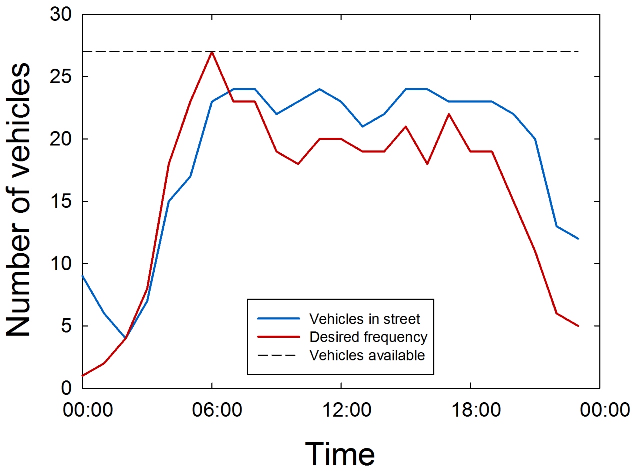


Figure 2.13: *Example of used vehicles for each hour.*

4) Working time. The average working time of drivers is 523 minutes with a standard deviation of 36 minutes, depending on the complexity of the line. These results are similar to those of the human planners. In particular, the model results have a variation from the average working time manually scheduled of 2%. These data are not official, but represent the average working day of the drivers according to the experience of the human planners.

5) Profiteer time. Before to consider this indicator, it is important to underline an important factor of the schedule quality. It is defined as the *number of the base turns*, the number of departures that a driver can achieve without going to overtime. For example, if a round of service lasts 74 minutes, including waiting time, a driver can ensure at the maximum 6 turns without going to overtime, as shown in Eq. 2.21

$$\left\lfloor \frac{480 \left[\frac{minutes}{shift}\right]}{74 \left[\frac{minutes}{turn}\right]} \right\rfloor = 6 \left[\frac{turns}{shift}\right] \text{ without overtime} \tag{2.21}$$

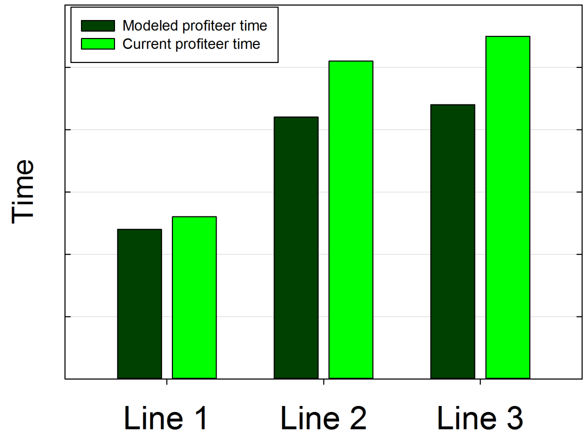


Figure 2.14: *Mean profiteer time.*

These 6 turns in a shift multiplied by the required time to traverse a turn (74 minutes in the example) are the time really worked by the driver. The difference between this time and the duration of the shift (480 minutes in the example) produces a quantity of time in a shift *profiteer time*. According to the experience of human planners, in developing countries it is preferable to minimize the profiteer time even at the cost of additional payment for overtime. This because the costs are very low and it is always possible to find a solution with the drivers by paying them. In developing countries is better one departure more than one less. In this case, an objective for the model is to minimize the profiteer time. For this reason a good indicator for a schedule is a low number of profiteer time. Figure 2.14 shows the average profiteer time for three lines taken as an example. The bars show the profiteer time obtained with the model and those from the charts currently in use. According to this figure, the model is able to reduce the mean profiteer time: for the first line, the simplest, the reduction is 8%, but as the lines become more complex the model is able to outperform the work of human planners almost of 20%.

6) Execution time. Using an Intel®Core™i5 CPU M 560 2.67 GHz 2.66 GHz and RAM 4 GB the average time of execution is 16 minutes. This time is definitely a good base to work on, since nowadays one or two planners are used full time with the only task of closing schedules.

2.3.4 Discussion

Planning and scheduling are common problems faced in many manufacturing and service industries. Developing countries can take advantage from the adoption of models that automate the planning process in public transportation. However, the common solutions must suit the peculiarities of those countries, in particular the very dynamic situation of roads and traffic. This section have presented a multi-agent model developed to solve operational planning problems, which was used in the Autonomous City of Buenos Aires, Argentina. The results show that the created schedules are very close to the desired ones, respecting different constraints. In particular, the presented model can improve the current situation of the company. The benefits of the automated solution can be summarized in (1) creation of more efficient schedules; (2) reduction of staff requirements for operational planning process; (3) reduction of costs for both vehicles and operators; and (4) introduction of enhanced flexibility and dynamism in the planning process. This work does not aim to be considered as a point of arrival, but as a starting point for the creation of models for operational planning within dynamic environments and especially useful in developing countries. The use of the model by qualified personnel with the necessary know-how of the planning tasks would lead to a continuous improvement of the model itself. This is encouraging since this process of continuous and iterative improvement, will lead to combining the power of automatic calculation of the model with the planning experience of the employees. This combination can be the key for the creation of a tool that not only reduces the working times, but that has the capacity to adapt more dynamically to unexpected events.

2.4 Automated storage and retrieval systems

Nowadays, logistics and distribution are crucial to guarantee a fast supplying that satisfies the high customer expectations. Automated storage/retrieval systems (AS/RS) are modern warehousing systems capable of automatically placing and retrieving loads from specific physical locations, allowing in this way several advantages. These storage systems are widely used in the logistics industry, in both distribution and production environments [177]. An AS/RS generally consists of stacker cranes running through aisles between racks, which are able to handling loads without the interference of an operator, thus the system is fully automated. The main advantages of AS/RS, compared to non-automated systems where workers guide forklifts, include high throughput, efficient use of space, costs reduction and improvement of safety [171]. In addition, AS/RS takes the maximum advantage of the cubic space within the racks, thereby favoring their study through the simulation. However, several factors, such as the high initial investments costs, inflexible layout, fixed storage capacity and necessary knowhow, impose the careful evaluation of system structures (e.g. the layout and size of the racks, S/R mechanism, the number of spans and levels) and operational policies (e.g. allocation of storage cells and scheduling of the tasks) [137].

The main components of an AS/RS are racks, cranes, aisles, input (I) and output (O) points, and pick stations. *Racks* are structures with locations where accommodate loads which have to be stored. *Cranes* are machines that can autonomously move, picking up and dropping off loads. *Aisles* are the empty spaces between the racks, where cranes are allowed to move. *Input point* is a location where incoming loads are picked up for storage, and *output point* is where retrieved loads are dropped off. These points can match, having thus a single location for I/O or not. Generally, each crane in an AS/RS has a vertical and a horizontal drive and typically one or two shuttle drives. The vertical drive raises and lowers the loads, while the horizontal one moves the loads back and forth along the aisle. The shuttle drives transfer the loads between the crane carriages and the storage cells in the rack. The vertical and horizontal drives are capable of simultaneous operations, allowing the crane to move obliquely. An example of an AS/RS is shown in Figure 2.15. The efficiency of an AS/RS principally depends on the design decisions (e.g., the storage rack dimensions, number of cranes and aisles) and the applied control policies (e.g., the dwell point position or the material clustering strategies).

Due to the numerous decisions involved at both design and operational levels, AS/RS have received considerable attention in the literature. Several authors discussed the expected results of different design or control strategies [63, 80, 148]. From a design perspective, the system configuration has to guarantee the handling of the current and future demand requirements. This phase is particularly critical because the physical layout is practically inflexible a posteriori. First of all, a selection of the AS/RS type is necessary, *system choice*. Then, the selected system has to be configured, *system configuration*. These choices are interrelated, and can be taken considering different factors, e.g., available space and budget, required throughput and storage space, and product characteristics. System configuration consists in select the number and length of aisles, height of racks, storage locations, number and location of I/O points, number of cranes per aisle, and number of order pickers per aisles. On the other hand, from

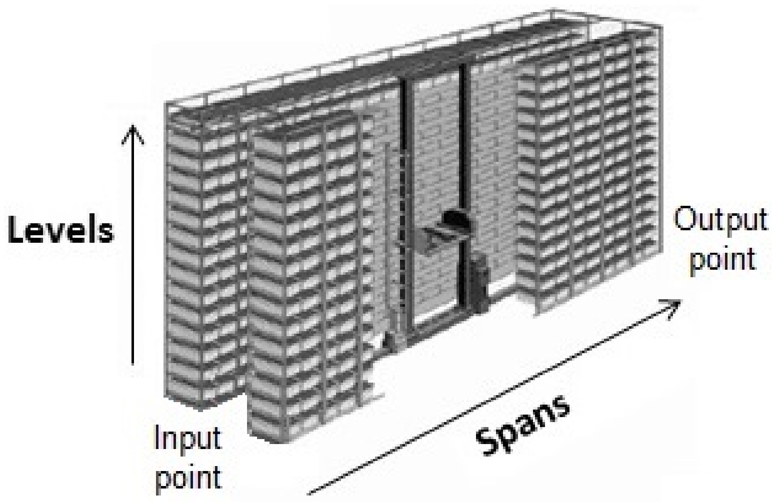


Figure 2.15: *Example of an AS/RS.*

the operational perspective, control decision must be addressed in order to achieve the maximum throughput capability for a given design [82]. In this case, problems related with the storage assignment [14], request sequencing [190], batching [132], and dwell positioning [143] must be optimized. Storage assignment refers to the configuration, and positioning of the storage space locations. Request sequencing addresses the ordering in which storage and retrieval operations have to be done by the cranes. Batching is related to the request sequencing, it concerns the possibility of combine several orders in a single tour of the crane.

2.4.1 Dwell point policy

The dwell point is the location where the stacker crane lies when it is inactive. An effective choice of the dwell point can reduce the travel time and the distance traveled, with a consequently reduction of the global costs [143]. For this reason, many dwell point strategies have been suggested and research problems investigating the optimal dwell point location have been formulated. Bozer and White [36] firstly understood the importance of finding a specific optimal point, within the crane aisle, where to place the crane before the execution of storage/retrieval requests. Egbelu and Wu [71] developed two dynamic models of linear programming that define the optimal dwell point coordinates, based on the relative likelihood that the next request was a storage or a retrieval. The first model had the objective of minimizing the expected response time, while the second minimized the maximum response time of the whole system. The authors compared six dwell point policies under dedicated and randomized storage policies using simulation, claiming that the solution from the minimum expected response time formulation performed better. Peters et al. [165] proposes several analytical models to determine the optimal dwell point based on the input and output points location. The authors derived a closed form solution procedure to obtain the optimal dwell point location in case of single server within a single aisle AS/RS. However, no discussion was included by the authors on the effectiveness of his strategy. Park [160] demonstrated that the input point is the optimal dwell point if the probability to receive a storage request is over 50%, regardless of the load allocation and rack typology. Furthermore, the author claimed that the input point is also the optimal dwell point for square-in-time racks and for loads allocated to dedicated positions. Van den Berg [213] developed an analytical expression for the optimal dwell point position such that the expected travel time to the first operation after an idle period is minimized. The author assumed that input and output points coincide into the lower left corner of the rack. Hale et al. [93] determined the optimal dwell point location with the scope to minimize the expected response times for incoming service requests. Ventura et al. [214] developed a polynomial time algorithm to find optimal dwell points when all request from a station are handled by a single automated guided vehicle, minimizing the response time. Regattieri et al. [176] evaluated the performance of different dwell policy found in literature. The performance of the AS/RS were analyzed in a parametric way, varying the number of spans and levels, the height of the input and output points, and the interval between requested missions. The results identify among the policies which one minimizes the travel time, distance traveled, and consequently warehousing costs by varying different parameters.

All these studies discussed dwell point strategies under a particular set of hypotheses. An agent-based approach can help the researchers to consider a wide range of variation of system configuration to study different control policy. In addition, an agent-based approach can help to simulate adaptive solutions. Indeed, different solutions can be the best under certain conditions. If the components of an AS/RS were modeled as agents, they could communicate each other, and autonomously find optimal dwell policies in different situations. This section presents an agent-based model able to find the optimal dwell point based on the previous travels. Storage and retrieval request act as forces that attract the dwell point, which will finish in an optimal point.

2.4.2 Autonomous agents approach

Roodbergen and Vis [177] analyzed the literature concerning AS/RS. The authors claimed that all the analyzed simulation models, only address some design aspects but their configurability of control policies is very limited. For this reason, only few control configurations and types of AS/RS have been tested, never guaranteeing that a optimal design has been found. According with these authors, Figure 2.16 resumes the typical decisions that have to be addressed to design and control an AS/RS, showing how they are connected each other. These interactions make these decision making processes very difficult, which open a door to AS/RS simulation studies, where the performance of different design and control combinations can be evaluated [81].

Modeling all the possible combinations of design and control decisions is costly and required a lot of effort on the part of the modeler, who often applies strongly simplified assumptions. The effects of different control policies in one physical design have been studied in [39, 50, 121, 213]. On the other hand, some studies focused on the effects on the design when one control policy was actuated [30, 125, 172, 178]. However, these models seem to be created ad-hoc for specific scenarios. Re-applying the same models when the scenario changes, or studying different control policies is really hard. Specific models, in fact, are poorly reproducible for other purposes, making their results difficult to generalize.

Some other studies analyzed the effect of control polices in different system con-

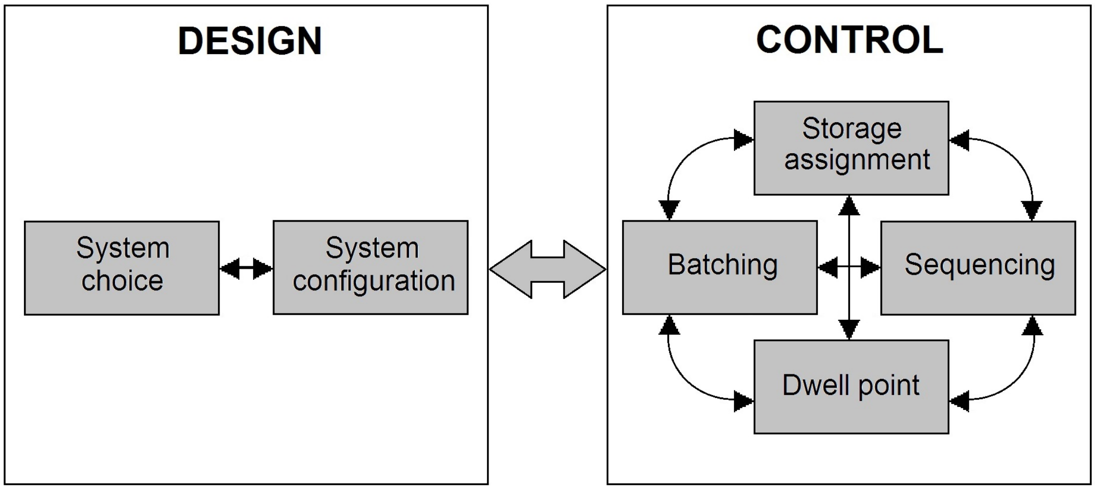


Figure 2.16: *Aspects involved in AS/RS designing and controlling processes.*

figurations. Taboun and Bhole [204] presented a discrete event simulation model to study the AS/RS performance, in terms of system throughput and utilization, changing system configuration and storage assignment. Their results indicated that systems with mixed pallet sizes holding several items yield better performance, independently by the size of the AS/RS. Randhawa and Shroff [173] simulated the effect of different sequencing rules on six layout configurations, varying I/O points, item distribution, and rack configuration and dimensions. Their results showed how locating a single I/O point at the middle of the aisle, improve the system throughput. Potrč et al. [167] explained an object–oriented model which considers different system configurations, determining a relationship between average the travel times and throughput capacity for different types of high storage racks and velocity profiles of storage retrieval machine. Azzi et al. [16] conducted a Monte Carlo simulation to analyze the performance of different batching policies changing system configurations in terms of cranes cinematic profiles and rack configurations. The authors suggested a new method to estimate the travel time of multi-shuttle systems. Bessenouci et al. [25] simulated the performance of two metaheuristic algorithms, which minimize the cranes' retrieval cycle time. The authors did not explain what kind of simulation they used, but they showed some results considering a random storage policy and varying the number of products in the system. Regattieri et al. [176] proposed an imperative model to evaluate the position of the dwell point considering a large set of conditions. The authors changed the AS/RS configuration and batching policies, concluding that the optimal dwell point depends on the size of the rack and the positioning of I/O points. Gagliardi et al. [81] presented a object oriented simulation modeling framework for AS/RS. Their model is able to study more than one scenario and it was used to study multi-aisle systems which were not studied before in literature. In addition, the authors explained the conceptual relationship between physical and decisional aspects.

Using imperative and discrete-event techniques, the control of autonomous components in a system is difficult to model. Hence, researchers have shifted their attention to the agent-based approach. This approach is able to decompose the design and control processes in small computational parts. In this way different combinations of design and control policies can be study, changing the decisions made within the single parts. Agent-based approach can be extremely useful for future studies, helping the researchers to model the different processes involved in an AS/RS. In fact, due to the elevated number of possible AS/RS system configurations and the relative control policies, AS/RS can be considered complex systems. One innovative way to model the AS/RS is represented by the complex adaptive systems, i.e., to model the activities of single entities and study the emerged global behavior. In this way, the interactions between the entities within an AS/RS and with the environment lead to the global behavior of the AS/RS. To implement and simulate complex adaptive systems, the agent-based approach was widely used in manufacturing and logistics [8,88,147,181], which represents a flexible framework to model autonomous systems and their interactions.

As already presented in this thesis, multi-agent systems are a common choice for simulating situations where different kinds of actors must be modeled according to the goal of their respective counterpart in the real environment [118]. When agents and their interactions are modeled, the analysis of the global system can be done considering two levels: *microscopic*, the study of the individual dynamics; and *macroscopic*, the observation of the collective behavior emerging by the agents interactions. For instance, Rabe and Clausen [169] analyzed the performance of a cellular transport system under various predefined scenarios, in order to support the design process, highlighting how the delays due to collisions between agents significantly impact the throughput capacity. Ito and Abadi [103] proposed an agent-based approach to model a traditional warehouse system. The authors decomposed the warehouse system in three "subsystems", an agent-based communication system, an agent–based material handling system, and an agent-based inventory planning and control system. Seven kinds of agents were designed: customer, supplier, order, inventory, product, supplier-order, and automatic-guided vehicle agents.

In this scenario, an agent-based approach can be able to compare numerous designs in combination with control policies, "simply" changing the parameters or the policies of single agents. As a result more information could be obtained exploring different designs. Separating the model into system configuration and control policies increases the flexibility and generality of the model. In this way, the interactions between the AS/RS subsystems can be modeled, and to simulate different interactions will be necessary to change only part of the subsystem and/or modify the number of agents. For instance, once a crane agent is created, to simulate more cranes, it is sufficient to increase the number of agents and to design how they interact. Agents can appear just like highly elaborated objects of the object-oriented programming. However, they derived from different conceptual ideas. An agent has a strong concept of autonomy, it can decide to do or not an action requested by another agent. Each agent has an autonomous control system, so they can be a good way to simulate AS/RS that are dynamic, flexible, and real-time adaptive to the variation of specific processes, products or the type of operation to perform. Each part of the physical configuration of the AS/RS can be modeled as an agent, but due to the almost unlimited possibilities of AS/RS configurations, only the components necessary for the analysis can be modeled. The physical configuration of the AS/RS is principally composed of: (1) storage racks, (2) storage locations, (3) items, (4) cranes, (5) aisles, and (6) I/O points. For instance, modeling the *rack* as an agent, as well as for other components, its characteristics become part of the agent itself. In order to model more than one rack with the same characteristics, it is enough to introduce within the environment different copies of the same agent that can exist independently. The rack in an AS/RS can be stationary or movable. The characteristics of the racks can be modeled as parameters of the agent. In case of stationary racks for instance, the rack can be considered as a continuous rectangular pick face, with parameters that define the length, height, and deep as single or double. In case of carousel racks, on the other hand, the agent rack contains also other parameters related with the vertical and horizontal rotating and can include intelligent behaviors. Through the reception of messages from other agents for example the rack can rotate when cranes are stacked in an aisle, or when there is a peak of operations related with a certain class of items.

This section introduces the abstraction of the agent modeling using a case study concerning the optimization of the dwell point positioning. The model proposes a new adaptive dwell point strategy which tries to optimize the traveling and waiting times of cranes, as presented in [10]. Different different AS/RS configurations in terms of output point position and racks' dimensions are considered.

2.4.3 Method

An effective choice of the crane dwell point minimizes the costs of an AS/RS, reducing the travel time and the distance traveled of the crane. For this reason, an agent-based model which studies the cranes' performance depending on the positioning of the dwell point was developed. In this presented study, agents were used to simplify the modeling. A further work is necessary to model an autonomous behavior of the agents. The presented model is composed of three type of agent: (1) one *crane agent* which executes storage and retrievals and can make single, dual or hybrid command; (2) one *dwell agent* that represents the dwell point defining its position; and (3) a series of *storage location agents* that attract the dwell agent depending on the number of times that being visited. As a result, each storage location agent has a location within the warehouse and a quite simple behavior, while the global behavior of the system allows the optimization of the dwell point positioning. The main idea is that storage location agents attract the dwell agent in sort of "mass" system where the dwell agent is located at the center of mass. The performance of the presented model are compared with the dwell policies found investigated Bozer and White [36], Linn and Wysk [130], and Regattieri et al. [176]:

1. *Proposed model* (M): the stacker crane returns to the dwell point found by the agents;

2. *Return to start policy* (IN): at the end of each cycle relative to a request (single or dual storage/retrieval command), the stacker crane returns to the input point;

3. *Last location policy* (LL): the stacker crane remains at the destination cell after the completion of a storage and retrieval operation, until another operation is required; and,

4. *Return to middle policy* (MID): the stacker crane moves to the midpoint location of the rack after the completion of any type of cycle.

Assumptions

In order to develop the model, some assumptions have been made:

- The rack is considered to be a continuous rectangular pick face where the input point is located at the lower left-hand corner, coordinates (0,0);
- The rack is single depth type;
- The storage locations are cubic spaces with a dimension known and equal;
- There is a stacker crane for each aisle (aisle captive AS/RS);
- Input/Output point can coincide or not;
- The stacker crane can make single command (SC) or dual command (DC) and the mission can be storage (S) or retrieval (R);
- If the cycle is dual command, the first mission must be storage and the second retrieval;
- The stacker crane travels simultaneously in the horizontal and vertical directions, with a known speed in both the directions;
- Each item is located in a cubic unit load with a known size;
- One unit load can be stored in one storage location;
- Load and unload times are equals and known;
- Randomize storage is used: the warehouse is initially filled for 80% of its capacity;
- Any operation (storage, retrieval, or both) has equal probability to be selected;
- The probability density function (PDF) of mission inter arrival time is a normal distribution, with a mean value μ and a standard deviation σ;
- The acceleration and deceleration of stacker crane are not considered.

The position of a generic unit load is defined by the duple (x,z), where x represents the number of spans, and z represents the number of levels. In this case study, the specific simulation dimensions are the follow:

Unit load dimensions	800×1,200×1,650 mm
Storage location dimension	2,800×1,300×1,700 mm
Number of spans	$8 \div 20$
Number of levels	$5 \div 10$
Crane horizontal speed v_x	2.0 m/sec
Crane vertical speed v_z	0.5 m/sec
Load/unload times	5 sec

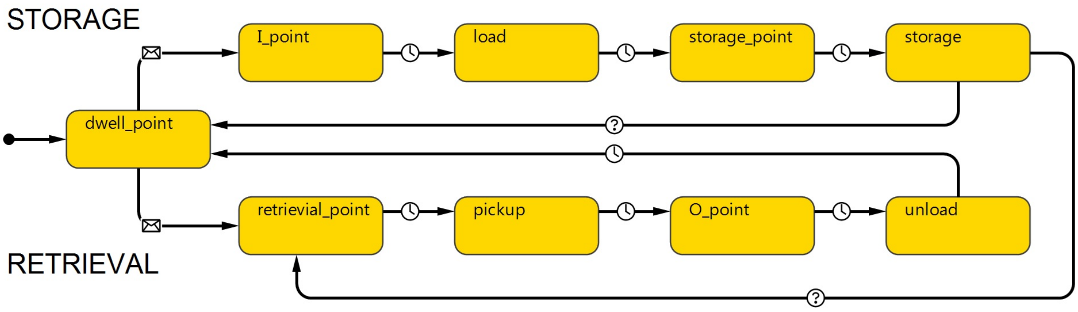


Figure 2.17: *An example of the crane agent behavior.*

Table 2.2: *Expressions used to model the travel time and travel distance.*

Request	Travel time	Travel distance
Storage	$\max\left\{\frac{\lvert x_c-x_I\rvert}{v_x},\frac{\lvert z_c-z_I\rvert}{v_z}\right\}+\max\left\{\frac{\lvert x_I-x_s\rvert}{v_x},\frac{\lvert z_I-z_s\rvert}{v_z}\right\}$	$\sqrt{(x_c-x_I)^2+(z_c-z_I)^2}+\sqrt{(x_I-x_s)^2+(z_I-z_s)^2}$
Retrieval	$\max\left\{\frac{\lvert x_c-x_r\rvert}{v_x},\frac{\lvert z_c-z_r\rvert}{v_z}\right\}+\max\left\{\frac{\lvert x_r-x_O\rvert}{v_x},\frac{\lvert z_r-z_O\rvert}{v_z}\right\}$	$\sqrt{(x_c-x_r)^2+(z_c-z_r)^2}+\sqrt{(x_r-x_O)^2+(z_r-z_O)^2}$

Agents

The crane agent can execute storage and retrieval operations, performing single, dual or hybrid command. A single command means that the crane performs a single operation, on the other hand, a dual command means that the crane performs one storage and then a retrieval operation in the same cycle. Figure 2.17 shows the behavior of the agent, where the yellow rectangles represent the status of the agents. At the beginning the agent is "resting" at the dwell point. The envelope represents a message, it means that the crane waits in the dwell point until it receive a message from the external that requires a storage or retrieval.

This model is based on a set of analytical expressions relating to the position of the input and output points, and the type of request. To simulate the movement of the crane, the crane agent waits in a status. The expected travel time and travel distance are calculated as shown in Table 2.2. In this table, I represents the input point, O the output point, c the crane, s the storage point, and r the retrieval point.

The storage location agent can be seen as a 3-dimensional area able to contain a certain volume of items. To simplify the modeling, items are modeled as unit loads, i.e., a cubic container that has to be moved within the AS/RS and stored in cubic storage locations. Each storage location agent has a position within the rack and a force that attracts the dwell agent. After each request, storage location agents close to point of request increase their own attraction force. The dwell agent calculates a mass balance and defines the center of mass as the best position to the dwell point. In case of a retrieval request, the best dwell point would be the retrieval point. In fact, if the stacker crane was exactly at the retrieval point, it would simply load and go to the output point, minimizing the travel time. For this reason, the main idea of the proposed model is that the storage location agent located at a retrieval point "attracts" the dwell agent. On the other hand, in case of a storage request, the best dwell point would be the input point, since, if the stacker crane was exactly at this point, it would load and go to the storage

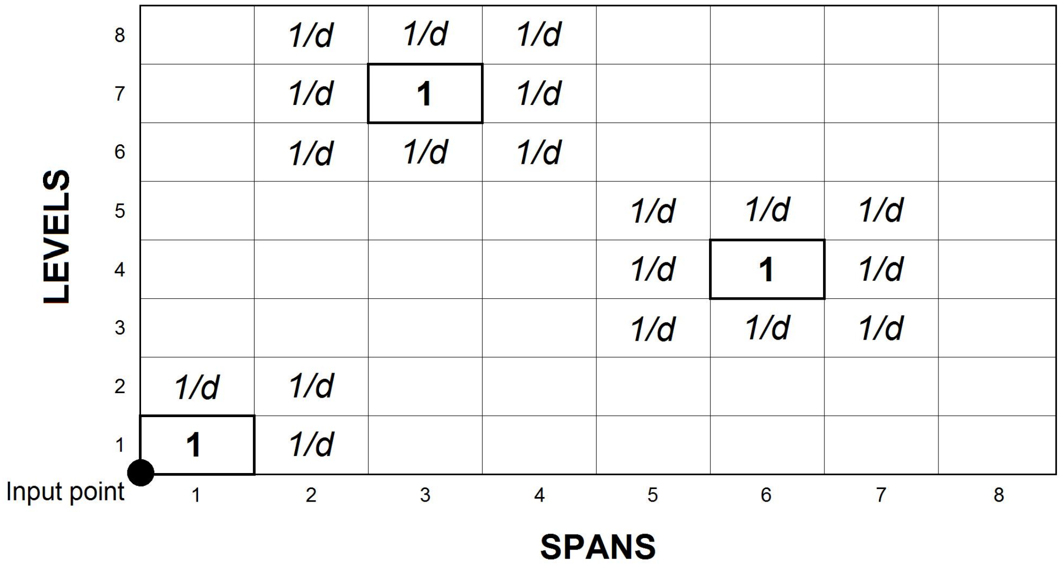


Figure 2.18: *Example of the attraction process.*

point, again minimizing the travel time. For this reason, the storage location agent located at the input point "attracts" the dwell agent after the storages. Figure 2.18 shows an example of the attraction process. In this example, each cell of the rack represents a storage location agent. The input point is located at the lower left corner, the origin (0,0). Suppose that two retrieval requests occur: one in the third span, seventh level; and one in the sixth span, fourth level. The "attraction forces" of the storage location agents located in these positions are called $W_{3,7}$ and $W_{6,4}$ respectively, and both of them can be defined by way of example equal to 1. Analogous to some physical forces (e.g. the force of gravity), moving away from the retrieval points that generate the force, there is a sort of reduction in the attracting force of the adjacent span agents compared to $W_{3,7}$ and $W_{6,4}$. In Figure 2.18, the attraction force of the storage location agents close to the retrieval points are defined equal to 1/d, where d is a smoothing factor greater than 1. Furthermore, suppose that a storage request also occurs, the storage location agent located at the input point attracts the dwell agent toward it, while the span agents close by also attract the dwell agent, although with a lower force. A storage location agent is influenced always more by retrieval or storage requests, summing the attraction force resulting from each one. For example, considering again Figure 2.18, if two retrieval requests occurred in the third span, seventh level, the attraction force $W_{3,7}$ of this storage location agent becomes 2, and the attraction forces of the storage location agents placed close to it become 2/d.

After each operation of storage or retrieval, the attraction force of the storage location agents changes. The dwell agent was modeled to communicate with the calculate the storage location agents and calculate the dwell position according to the following Equations 2.22 and 2.23:

$$X_{dwellAgent} = \frac{1}{\sum_{i=1}^{S} \sum_{j=1}^{L} W_{i,j} \sum_{i=1}^{S} \sum_{j=1}^{L} W_{i,j} \cdot x_{i,j}} \tag{2.22}$$

$$Z_{dwellAgent} = \frac{1}{\sum_{i=1}^{S}\sum_{j=1}^{L} W_{i,j} \sum_{i=1}^{S}\sum_{j=1}^{L} W_{i,j} \cdot z_{i,j}} \quad (2.23)$$

where S and L are respectively the number of spans and levels of the rack, $W_{i,j}$ is the attraction force of the storage location located at span i level j, $x_{i,j}$ is the horizontal coordinate of span i level j, and $z_{i,j}$ is the vertical coordinate of span i level j.

After that a certain number of storage and retrieval requests have been satisfied, the dwell position converges in an equilibrium point which can be considered an optimal point of dwell. A damping of attraction over time is fundamental for this convergence. After a certain number of operations, the fact that a storage location agent continues to attract the dwell agent loses its significance. Conceptually, if no operations are carried out in a certain area for a time, this area become not attractive for the dwell point. Therefore, after each request, storage location agent attraction $W_{s,l}$ decreases according to a force smoothing factor fs, in order to ensure the balance of the system (Equation 2.24).

$$W_{s,l} = W_{s,l} - fs \quad (2.24)$$

Set of simulated instances

The model considers a random storage allocation of the unit loads. Different storage configurations have been simulated to test the proposed model. Principally, different AS/RS configurations in terms of number of spans and levels have been considered. In particular, the simulation considers 322 AS/RS configurations: 23 different spans with 14 levels per span. In addition, for each rack configuration, the model considers six different PDFs of task inter–arrival times, defined as follows:

μ_1 = 120[sec] $\sigma_{1,1}$ = 30[sec] $\sigma_{1,2}$ = 60[sec] $\sigma_{1,3}$ = 100[sec]

μ_2 = 300[sec] $\sigma_{2,1}$ = 130[sec] $\sigma_{2,2}$ = 150[sec] $\sigma_{2,3}$ = 180 [sec]

For each rack configuration and PDF, two different locations of the output point were analyzed: output point coincident with the input point at the lower left-hand corner of the rack, and output point detached from input point, located at the middle of the rack length at the level 0. In total, 1,932 different instances have been defined and, for each of them, 1,000 unit load tasks has been performed. The simulation test bed has been implemented using the AnyLogic software agent platform. As already mentioned in this thesis, this simulation platform is widely used to model a very wide range of problems [29].

The purpose of this simulation is to verify the performance of the proposed model compared with the performance of the model found in literature. This simulation allows to understand if the models is able to adapt to different characteristics of the rack, in terms of number of spans, number of levels, output point position, and the rate of speed of the requests.

Results

An *Average Dwell Point* (ADP) is obtained considering all the tested instances for all warehousing configurations. Figure 2.19a shows the trend of the ADP coordinate z, respect the number of levels. According with the model, the vertical coordinate z of the optimal dwell point increases with the number of level. Figure 2.19b instead shows

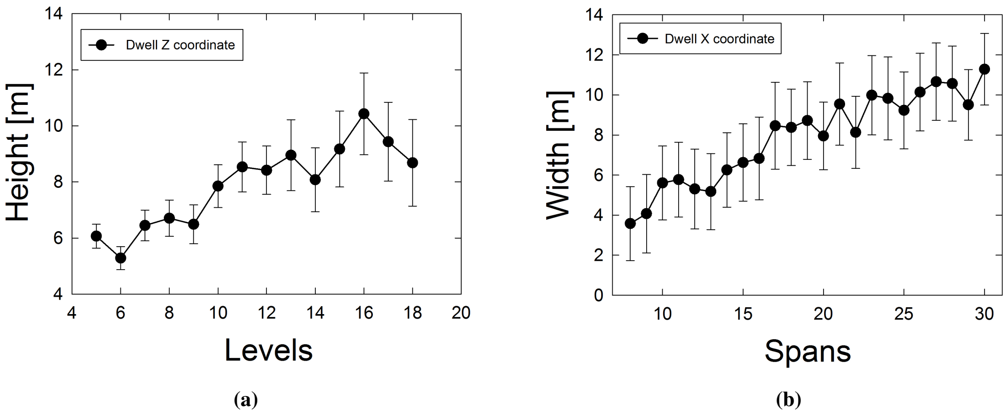


Figure 2.19: *Average dwell point coordinates for different warehousing configurations.*

the trend of the ADP coordinate x, respect the number of spans. Also the horizontal coordinate x of the optimal dwell point is influenced by the number of the spans. The results indicate that with a random storage allocation policy, input point located at the beginning of the rack, and the defined assumptions (in particular the horizontal speed higher than the vertical speed) the optimal dwell point tends to move between the input point and the middle of the rack.

To analyze the performance of the policy modeled, we have used the following parameters: (1) travel time, i.e., average time taken by the stacker crane to complete a task and possibly reach the dwell point; (2) wait time, i.e., the time that a unit load must wait to allow the stacker crane to reach the point of storage or retrieval. This time also includes the case in which the unit load must wait while the stacker crane is engaged in another request. An *average dwell point policy travel time* (ADPPTT) and an *average dwell point policy wait time* (ADPPWT) are obtained considering all travel times for all tested instances for all warehouse configurations. Figure 2.20a shows the trend of ADPPTT and Figure 2.20b shows the trend of ASPPWT for each dwell policy analyzed.

Considering all policies, the number of spans is more influential than the number of levels for both ADPPTT and ADPPWT. If the mean value μ of PDF of mission inter-arrival time is 300 seconds, the wait time is null, i.e., the loads do not have to wait to be positioned. If μ is equal to 120 seconds, the average wait time is in the order of 30 seconds for all policies, including the proposed one. Considering the travel time, the LL policy outperforms the other policies. However, the proposed model has on average a difference of travel time of 17% respect this policy. The model proposes an interesting dwell point policy in terms of wait time. The difference between the LL policy is 32% on average in this case. In conclusion, it is possible to affirm that the LL policy minimises the travel time, while the proposed model is able to minimise the wait time. It means that using the proposed approach, the unit loads are achieved first, thus increasing the time performance of the AS/RS.

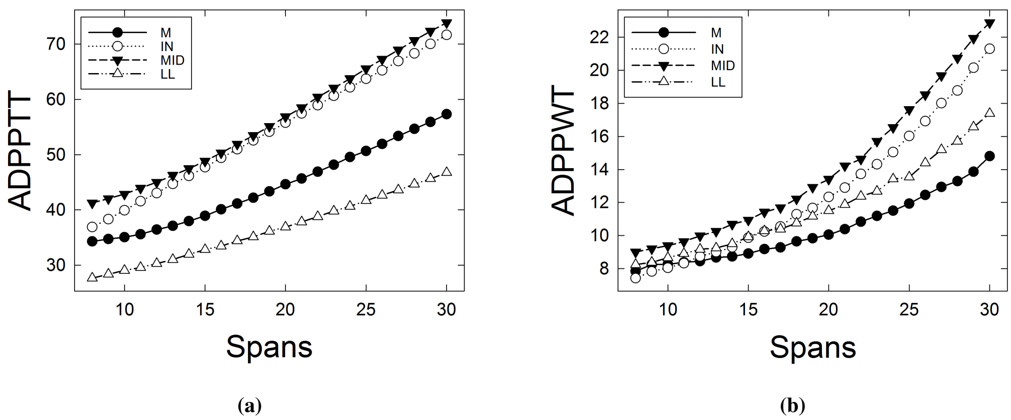


Figure 2.20: *Average dwell point policy travel time and average dwell point policy wait time for or the policies analyzed.*

2.4.4 Discussion

Automated storage/retrieval systems (AS/RS) are devices that allow intensive physical storage of materials. They can improve the performance of the supply chain, reduce labor costs, and ensure higher throughput of warehousing minimizing the system response time. Moreover, the automated control minimizes chances of product damage during movements, along with the errors likelihood in storage and retrieval. However, due to the high initial investments and fixed configurations, an a priori evaluation of system structures and operational policies is necessary. This section presented how to a multi-agent approach can be used to facilitate the modeling of different AS/RS dynamics and to improve the flexibility of simulated scenarios. These results are achievable, modeling separately the design and control decisions. The developed model investigates a new policy to find an optimal dwell point in AS/RS with a random allocation of unit loads. The model was tested considering a wide set of AS/RS configurations, simulating different storage dimensions (varying the number of spans and the number of levels simultaneously) and arrivals (varying the interval between requested tasks). 1,932 different instances have been investigated and, for each of them, 1,000 simulation runs have been executed. The model performance were compared in terms of travel time and waiting time with several policies found in literature. Results showed that the proposed model finds an interesting dwell point policy in terms of wait time. In particular, the model is able to minimize the wait time, i.e., the unit loads are achieved first, increasing the respond time of the AS/RS, making it leaner.

CHAPTER 3

Prediction in complex systems

Since ancient times, the prediction of future events has had relevant importance in the life of every culture. Nowadays, scientists predict a vision of the future developing theoretical models of fundamental processes, digitizing the nature in all its rich profusion and using powerful computers to analyzed enormous amount of data. Prediction runs from the mundane and individual–knowing it might rain today means we should bring an umbrella–to the anticipation of economic and cultural changes in countries or corporations. However, even the most sophisticated scientific prediction is plagued with uncertainties. Uncertainty, for instance, governs the prices of fuels, the demand and availability of electricity, and the reliability of our products. Decision-making under uncertainty is often further complicated by the presence of integer decision variables to model logical and other discrete decisions in a multi-period or multi-stage setting.

Classical approaches for the prediction in non–complex systems consist in carry out an analysis in a statistical perspective, in which the dynamical origin of the phenomenon and its deterministic aspects are not addressed. In the most common version it refers to stationary univariate data, where the location and spread of a time series is provided by the mean and standard deviation. The way in which some relevant variables vary in time is usually achieved by representing the data set in a mathematical model, involving a limited set of parameters. These parameters have to be fitted in such a way that the model statistical properties are identical to those determined by the data. In classical data analysis, the prototype of this kind of mathematical models are the *autoregressive models*. They can be regarded as "filters" converting an uncorrelated process to a correlated one. In other words, the regularities contained in a set of data are identified through the correlation function inferred from the data, and prediction is carry out by means of linear models fitted to reproduce this function. In this statistical and linear perspective the nature of the underlying dynamics is overlooked. Predicting the future evolution of complex systems is one of the main challenges in complexity science. The non–repetitiveness, a pronounced variability extending over several space and time scales, sensitivity to the initial conditions and to the parameters are some of the characteristics that make the prediction so challenging [152]. Indeed, complex systems can generate a whole variety of dependencies associated to the different states that are. Clearly the situation is far from a complicated system in which variability results primarily from contamination by a weak background noise. Complex systems are influenced by several complex factors that exhibit nonlinear patterns. Their forecast cannot depend on the assumptions of independence and linearity, but requires more and more sophisticated models which capture the complexities of the system behavior in a realistic way.

In the last years, machine learning models have attracted considerable interest due to their flexible capacity and ability to “understand” the complex nonlinear relationships between input and output patterns through an appropriate learning process. These models attracted the interest of the scientific community as the promising way to explore hypothesis and provide access to the inner workings of complex phenomena or to phenomena that are difficult to examine by other means [203]. Machine learning models allow research in several topics where exist cognitive, ethical, political, or practical barriers. The reliability of the prediction of machine learning models is based on their capacity of:

- Approximation of past events;
- Increased understanding of natural processes;
- Learning from experimental observations;
- Classification and generalization;
- Providing an avenue for communications.

Presenting different cases that explain how observing, analyzing, and predicting a system in a way that does not miss essential features of its complexity is the principal theme of this chapter. The chapter is organized as follow: Section 3.1 analyzes the performance of different machine learning approaches to predict the reliability parameters of mechanical and electric components. A first approach to these models can be complex, for this reason the section starts with the exploration of simple networks, usable even by non-experts. After that, the use of more advanced models (e.g., random forests, fuzzy systems, and support vector machine) is presented. Section 3.2 presents an approach to predict reliable climatological data. These data are used in this section as a key input for the installation of solar panels and for the optimization of the energy exchange in a smart city. Finally, Section 3.3 presents a basis above which the social interaction in an online community can be predicted. In particular, it is presented how the success of these communities are correlated with their diversity, intended as the variation of some characteristics of the users. And lastly, it is presented how a machine learning approach can be used to predict the choices of users within the community.

3.1 Reliability of components

The advancement of international markets and the consequently increase of global competition has led manufacturers to create more and more customizable products to cope with high customer expectations. Nowadays, manufacturers need to produce high-quality products in a lower and lower time. In this competitive environment, the interest of manufacturers is to focus more than ever before in the machines' reliability. This growth has been motivated by several factors which include the increasing of complexity and sophistication of the systems, insistence on product quality, warranty programs, safety laws, and profit considerations. Some of the latter factors are influenced by the high cost of failures, their repairs, and replacement [70].

A robust reliability analysis requires an a-priori effective failure process investigation, based on operating and failure times of a generic component. A *component* can be a part, device, piece of equipment, or a whole system individually considered. A failure process starts analyzing the failures of some components. Failure times, also called *times to failure* (TTF), are the times when the analyzed components have stopped working. Once TTF are available, statistical models are applied to obtain a meaningful estimation of the fundamental reliability parameters, especially the *cumulative failure distribution* $F(t)$ and the *reliability function* $R(t)$. Also some other technical condition or future failures can be predicted by using these models. These reliability models are often used under simplifying assumptions to enable analytical or suitable numerical treatment [61], that are difficult to validate. Reliability, indeed, is influenced by several complex factors that exhibit nonlinear patterns. Therefore, the prediction of the reliability requires sophisticated models which capture its complexities. The study of the reliability is important to make important decisions to improve various aspects of the manufacturing processes. Different purposes enable to apply many possible optimization policies, e.g., the production planning [145], an optimal mix of maintenance policies [3], fault detection [51], an effective spare parts management [174], warehouses optimization [116], with a corresponding improvement in safety and costs.

In the recent past, new models able to capture the complexity of the reliability parameters have emerged above the empirical techniques of failure data regression. In particular, the use of artificial neural networks (ANNs) [12,47] and support vector machines (SVM) [61,100] allows a feasibility and superiority of these models in reliability data analysis. Machine learning models, indeed, emerged as models able to approximate any nonlinear continuous function with good accuracy. Consequently, their use is emerging also for the systems reliability modeling, evaluation and prediction. This section is divided into two parts. The first one aims to demonstrate that also simple structures of ANNs may give interesting results to predict the reliability parameters. It presents the results of [9], where the performance of artificial neural networks for the prediction of the cumulative failure rate are analyzed. Approaching these networks can seem complex, but the study aims to explore the good prediction accuracy even of simple networks. In particular, the use of feed–forward neural networks with one hidden layer and back–propagation method of learning with different parameters (e.g., the learning rate and momentum) is analyzed. Furthermore, the second part of the section aims to test the performance of other three machine learning models to predict the reliability of mechanical and electrical components, in presence or absence of censored data. In particular, the main purpose is to understand if there is a single data-driven learning model able to outperform other models.

3.1.1 Context analysis

Reliability analysis can be classified depending on the prediction they make: prediction of the system's ability to operate, complete a mission, or perform at determinate circumstances. This section focuses on the *basic reliability analysis*, i.e. the prediction of the analyzed components to operate without maintenance and logistic support at certain times [119]. Therefore, the term reliability is used to identify the probability $R(t)$ that a component will perform without failing a required function at least for a given period of time t, under stated operating conditions and assuming that the component is new at time zero. Reliability is a decreasing function with the time t, assuming that it is equal to one at time zero, and zero at the infinity time [70]. On the other hand, the cumulative failure distribution $F(t) = 1 - R(t)$ describes the probability of failure (at least) up to and including time t.

The reliability analysis generally started with an empirical approach. The purpose of the *empirical approach* (or nonparametric approach) is to estimate directly the reliability parameters (reliability function $\widehat{R}(t)$ and cumulative failure rate function $\widehat{F}(t)$) from a set of failure times. Functions $\widehat{R}(t)$ and $\widehat{F}t$ correspond to the estimated reliability and cumulative failure rate functions using empirical approach. Once independent variables (TTF) and dependent variables ($\widehat{R}(TTF)$ or $\widehat{F}(TTF)$) are defined, different statistical distributions are tested to fit in the better way that set of couples, in order to define continuous functions $R(t)$ and $F(t)$. Using one distribution, the reliability parameters become available also for the times not presented in the range of collected data. The most common distributions used to fit these functions are the exponential, normal, and mostly the *Weibull distribution*. This fitting phase can be seen as a "pattern recognition" of the failure times distribution; field where machine learning models have been found to be extremely suitable for solving such problems [191].

The failure process investigation is costly because requires a lot of effort and time to collect information. One of the most difficult task in the industry is to obtain reliable information to get sufficient data available. For this reason, sometimes practitioners make use of hidden information to define the reliability of a component. These hidden information derive from the components which are still operating at the end of the tests for the failure investigation. These conditions are usually known as *censored data* situations and can influence the analysis of the components' reliability [138]. Considering an environment of n units, a *complete data* situation occurs when the failure times of all n units are available. However often data received from industry can include incomplete or missing information, but this is not necessary a monitoring tracking problem. Simply often, not all units fail during the tests, or several units fail between two data monitoring. In those cases, the failure time is unknown. In fact, censored time means precisely that the exact failure time is unknown, whatever the reason.

Practically, the starting point for an evaluation of the reliability parameters is a set of failure times. The available set is represented by t_1, t_2, ..., t_n, where t_i represents the time of failure of the i^{th} unit and n the number of available TTFs. These failure times have to be ordered in an increasing way, obtaining t_1, t_2, ..., t_n, where $t_i \leq t_{i+1}$. Then, depending on whether complete or censored data are present in the analyzed dataset,

	Direct Method	Improved Direct Method	Median Rank
$\widehat{R}(t_i)$	$\frac{n-i}{n}$	$\frac{n+1-i}{n+1}$	$\frac{n+0.7-i}{n+0.4}$
$\widehat{F}(t_i)$	$\frac{i}{n}$	$\frac{i}{n+1}$	$\frac{i-0.3}{n+0.4}$

(a) *Complete data*

	Product Limit Estimator	Kaplan-Meier Method
$\widehat{R}(t_i)$	$\left(\frac{n+1-i}{n+2-i}\right)^{\delta_i} \widehat{R}(t_{i-1})$	$\left(1-\frac{1}{n_i}\right)^{\delta_i} \widehat{R}(t_{i-1})$

(b) *Censored data*

Table 3.1: *Empirical methods to estimate the reliability parameters with (a) complete data and (b) censored data.*

different methods can be applied to estimate the empirical functions. The most common methods are the *direct* method (DM), *improved direct* method (IDM), and *median rank* (MRM) [138], for the complete data situations. On the other hand, in presence of censored data, the *product limit estimator* method (PLE), and its variation, know as *Kaplan & Meier* method (KM) [109] are most used. Table 3.1 shows the formulas of the various methods to estimate the empirical functions. In this study, censored data are used to predict only the reliability function.

Figure 3.1 summarizes the method used to find the reliability and cumulative failure rate functions, using complete and censored data, and artificial neural network as well as other machine learning models.

The index used to evaluate the goodness of the fitting is the *mean squared error* (MSE). For instance, in case of studying the reliability function $R(t)$, MSE is calculated as the sum of the squares of the differences between the value of the failure distribution in the i^{th} time in the output from the created models $R(t)$ and in the output from the empirical methods $\widehat{R}(t_i)$, Equation 3.1. Considering that MSE measures the deviation between the real and predicted values, the smaller the values of MSE, the closer are the predicted values to the real values.

$$MSE = \frac{1}{n} \cdot \sum_{i=0}^{n} (R(t_i) - \widehat{R}(t_i))^2 \tag{3.1}$$

Data sets

Data used in this study can be considered an exhaustive representation of the real industrial conditions. Indeed, the variety of the dataset has been achieved investigating the failure process of 19 different real world components, each of them with different available failure and censored times, as shown in Table 3.2. The column "Number of TTF" refers to the number of failure times of which there is a reliability function. It does not include the censored times and the repeated TTF. Different availability of TTF and censored times are typical characteristics of real conditions: monitoring and good maintenance policies, for example, can ensure the availability of a consistent number of TTF, but on the other hand, there are other cases where only few historical data

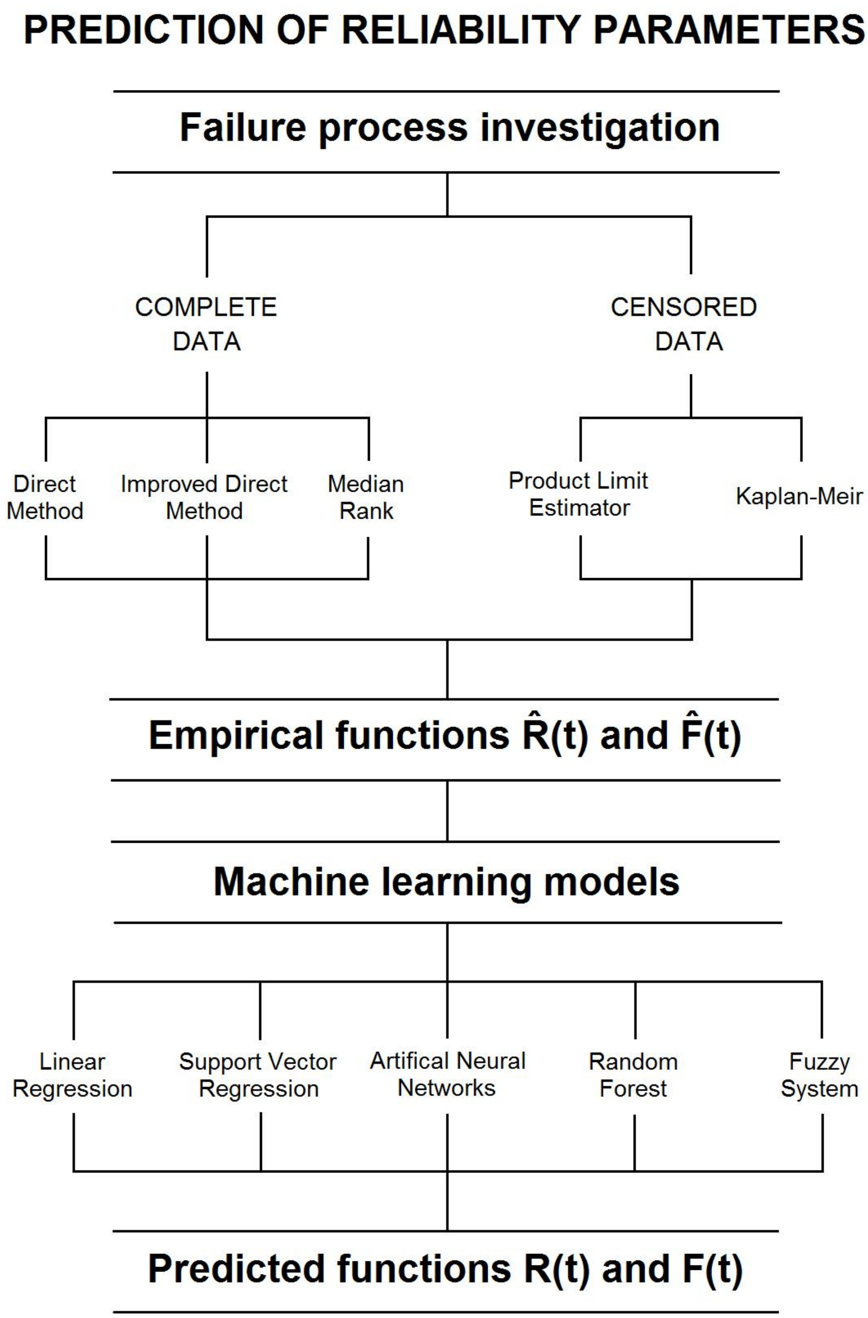


Figure 3.1: *Method to predict the reliability parameters using artificial neural networks and other machine learning models.*

	Component	Number of TTF	Number of censored data
1	cod A	73	10
2	cod B	96	10
3	cod C	20	-
4	cod D	25	10
5	cod E	21	6
6	cod F	37	-
7	cod H	22	-
8	cod I	23	4
9	cod L	47	5
10	cod M	32	4
11	cod N	71	7
12	cod O	34	-
13	cod P	11	-
14	cod Q	24	-
15	cod R	24	-
16	cod S	15	-
17	cod T	45	
18	cod U	607	18
19	cod V	282	10

Table 3.2: *Number of available TTF and censored data between them, within the data set of the different components.*

are available [75]. In particular, data come from the information system supporting the management of maintenance of three large companies located in northern Italy involved in different fields. The analyzed TTF are related to mechanical and electrical components of technical assets for the production of electricity, to equipment for the filling of beverages and are part of assembly lines of cars.

Table 3.3 shows the empirical reliability function of the cod E as an example for the sake of explanatory clarity. If a component failed more than once at the same time, the lowest reliability value is considered as reliability at that time. Cod E, for instance, has 21 TTFs in Table 3.2, but during the failure process investigation 29 TTFs were available: 6 of them were censored times and three times the component E failed after 2,202 unit time, as shown in Table 3.3. The estimated $\widehat{R}(t_i)$ derived from the above methods is used to evaluate the approximation of the machine learning models.

3.1.2 Artificial neural networks

During the last century, ANNs have attracted considerable interest in time series forecasting [234], principally due to their capacity for (1) approximation, (2) learning

Table 3.3: *An example of dataset used in this study. It can be used as complete or censored.*

i	TTF	Censored	Complete dataset $\widehat{R}(t)$ DM	IDM	MRM	Censored dataset $\widehat{R}(t)$ PLM	KM
1	376	+					
2	436	+					
3	614	+					
4	830	+					
5	1135		.957	.958	.970	.962	.960
6	1285	+					
7	1384		.913	.917	.927	.921	.918
8	1865		.870	.875	.885	.881	.877
9	1971		.826	.833	.842	.841	.835
10	2027		.783	.792	.799	.801	.793
11	2202						
12	2202						
13	2202		.652	.667	.671	.681	.668
14	2684		.609	.625	.628	.641	.626
15	2784	+					
16	2850		.565	.583	.585	.598	.581
17	3112		.522	.542	.543	.556	.537
18	3258		.478	.500	.500	.513	.492
19	3310		.435	.458	.457	.470	.447
20	3541		.391	.417	.415	.427	.402
21	3674		.348	.375	.372	.385	.358
22	3875		.304	.333	.329	.342	.313
23	4037		.261	.292	.286	.299	.268
24	4339		.217	.250	.244	.256	.224
25	4506		.174	.208	.201	.214	.179
26	5552		.130	.167	.158	.171	.134
27	5821		.087	.125	.115	.128	.089
28	6447		.043	.083	.073	.085	.045
29	8286		.000	.042	.030	.043	.000

from experimental observations, (3) classification, and (4) generalization. ANNs have been successfully applied to different sorts of reliability engineering problems. Liu et al. [131] applied neural networks to identify underlying failures distribution and to estimate the parameters. The potential of the neural network approach for the reliability and failures prediction of engine systems has been clearly demonstrated in the experimental results of Xu et al. [225]. The authors used an ANN to predict the occurrence of future failures. Bevilacqua et al. [26] adopted an ANN to weigh up the correlation existing among the failure rates and the several different operating conditions of several centrifugal pumps used in an oil refinery plant. Tong and Liang [208] combined neural networks and SARIMA model to forecast the reliability for repairable systems. In particular they used an ANN to predict the mean number of repairs. Yi-Hui [229] used an evolutionary neural network modeling approach to predict the reliability for repairable systems. In this study, genetic algorithms are used to globally optimize the neural network architecture. In the field of automotive vehicles, the studies of Marsaguerra et al. [139] demonstrate how an ANN system can be developed in order to predict accurately enough the reliability behavior. In the studies of Lolas and Olatunbosun [134], an ANN was trained to find the associations between the input parameters at 0 km and the target values of the reliability performance after a certain number of kilometers driven. The works of Chatterjee [47] and Pai and Lin [158] explained how to estimate reliability using forecasting failures with an ANN. Finally, Rajpal et al. [170] used a neural network approach to analyze the reliability, availability, and maintainability of a complex repairable system. An estimation of the reliability of components in petrochemical plants was studied by Dehghan and Hoseinnezhad [64]. Al-Garni and Jamal [5] used a series of ANNs to model the failure rates of the tires of five Boeing 737 airplanes.

On the one hand, all these studies have demonstrated that ANNs have a good ability to predict, even in complex situations, using their ability to learn the solution to a problem from a set of examples, prediction often better than traditional approaches. On the other hand, the creation of the networks may appear too expensive from the point of view of its greater computational burden, compared to the advantage that can be obtained this way. From the nature of these networks, the presence of an expert in their creation is necessary. That is because it is not possible to know a priori what is the best network topology (e.g., the network type, the hidden node size, the number of hidden layers, the learning parameters, the connections between the nodes, the types of the functions, and so on). Indeed, the main problem in using ANNs is the so-called *network fitting problem*, which usually is encountered upon a poor selection of the neural parameters mentioned above. The main challenge for any ANN user is to suitably choose the parameters that help the network fitting. In literature it is possible to find a certain amount of methods capable of selecting in an effective manner the parameters for network models [18, 156] that, however, are complex and time consuming. Despite the good results of these methods, the complete topology of the network to use for a single case is undefined a priori, and above all the methods can prove to be too specific to be applied by those that approach the neural networks for the first time or are not expert in the field. These factors create a certain resistance to the concrete application of ANNs in the maintenance field, where non-expert users could not exploit the benefits of the networks due to the complexity of the model.

This first part of the section explores the use of the most widely used type of net-

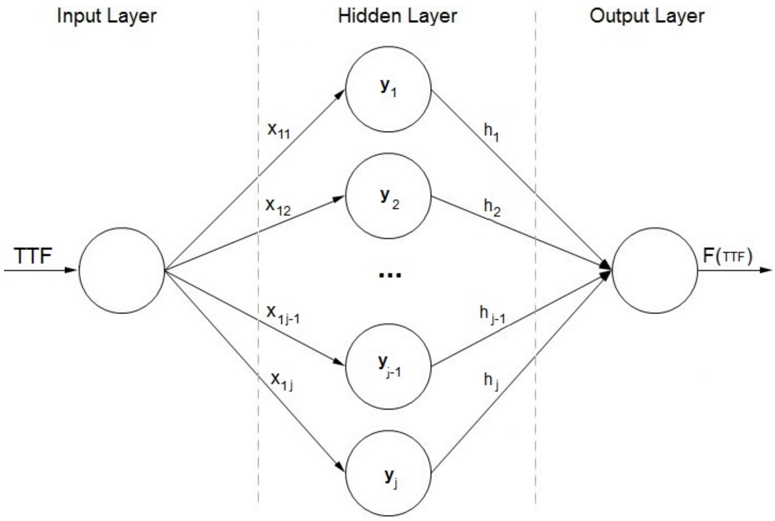


Figure 3.2: *Structure of the ANN model to predict the cumulative failure distribution.*

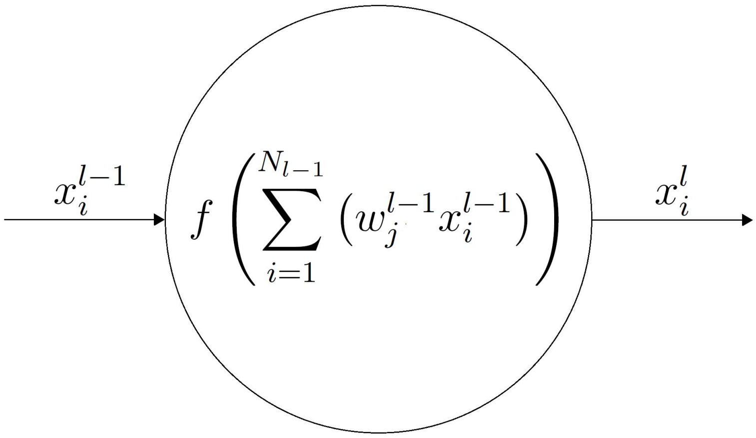


Figure 3.3: *Structure of a single node j.*

work, i.e., the multi–layer feed–forward network, which is considered the workhorse of ANNs [69]. In this part, the prediction of the cumulative failure distribution using a complete data situation is analyzed. The structure of a feed-forward neural network consists of an input layer, one or more hidden layers, and an output layer. In this case, the input layer is made up of a single neuron, because the purpose of the network is to find the relation existing between a single failure time TTF and the cumulative failure distribution. The output layer consists of one output neuron, producing the predicted value of $F(t)$. All the hidden neurons are fully connected to both the input and output neurons; therefore, a direct connection between input and output layers does not exist. The architecture of the three-layer neural network used is shown in Figure 3.2.

Every connection in the network has a weight w_j^l, and each neuron j in the hidden and output layer l integrates the signals x_i^l that it receives with the corresponding weights; introducing them in a function called activation function. The structure of each neuron is shown in Figure 3.3.

The activation functions widely used are the sigmoid or hyperbolic. The advantage of these functions is that their smoothness makes it easy to devise learning algorithms [142]. In our studies the basic continuous sigmoid function is used as activation function, shown in Eq. 3.2. This activation function gives output values in the range [0,1].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

The back propagation algorithm (BP) has been used in the learning phase of the ANN. One more time, this selection has been made for its popularity, in fact this algorithm is the most common procedure for training ANNs. The BP algorithm is a supervised iterative training method based on searching the global minimum in the error surface, i.e., the difference between the ANN output and the target, as a function of the ANN weights and biases, using the gradient descent approach. The difference (i.e., the error) between the ANN and the target outputs is used to adjust the interconnection weights, starting from the output layer, through the hidden layer, until the input layer is reached. Weights $w_(i, j)$ are initialized with random values and then changed during the training in a direction that will minimize the mean square difference between the real network output and the desired output. The weight adjustment at the iteration (called *epoch*) t is expressed as:

$$w_{ij}(t) = w_{ij}(t-1) + \delta w_{ij}(t) \tag{3.3}$$

where $\delta w_{ij}(t)$ is a correction term depending on the error committed by the network as a function of two constant parameters,$\eta \in$ [0,1] and $\mu \in$ [0,1], known as *learning rate* and *momentum* [15]. The learning rate, during each iteration, controls the size of weight and bias changes, while the momentum helps the search of the global minimum on the error surface, preventing the system from converging to a local minimum or saddle point. The iterative process of presenting an input-output pair and updating the weights continues until the error function reaches a pre–specified value or the weights no longer change. In that case the training phase is done and the network is ready to be tested with input that it never seen before. As a consequence, η and μ, are crucial for the network to learn properly and quickly.

Input values are another important aspect to consider. Approaching a neural network, it is possible to use either the original data or transformed data, such as standardized data between 0 and 1. Many researchers routinely use transformed data, sometimes due to the requirements of their algorithm, or sometimes for improved learning. Indeed, data standardization on neural networks principally influences the learning phase of the network. An a priori standardization data not only can reduce the time needed to complete the training process, but it may also reduce the network error estimation [198]. In this study, the empirical $\widehat{F}(t)$ derived from IDM is used to evaluate the effectiveness of the approximation of the ANNs and the Weibull distribution.

Results and discussion

Regardless of learning parameters, all the networks with a number of neurons in their hidden layer greater than four have a better performance than the Weibull distribution for all components. It means that MSEs using ANNs (MSE^{NN}) with more than four neurons in their hidden layer are always lower than MSEs adopting Weibull distribution (MSE^{Wei}). The percentage improvement of ANNs over the Weibull distribution is called

η	μ	N° of neurons in the hidden layer						
		5	**6**	**7**	**8**	**9**	**10**	**11**
0.3	0.25	68.04%	68.33%	66.91%	67.05%	67.36%	67.93%	67.06%
0.3	0.5	69.74%	70.85%	69.62%	71.10%	70.77%	70.82%	70.51%
0.3	0.75	71.73%	74.52%	74.96%	72.62%	73.52%	74.14%	75.15%
0.5	0.25	70.04%	69.38%	69.43%	69.85%	69.08%	69.37%	68.83%
0.5	0.5	74.51%	72.77%	72.91%	72.26%	72.38%	71.57%	72.55%
0.5	0.75	73.78%	74.81%	75.15%	74.49%	75.22%	74.94%	72.63%
0.7	0.25	71.08%	70.70%	71.42%	69.54%	69.29%	69.35%	69.23%
0.7	0.5	74.90%	74.71%	74.27%	72.68%	72.60%	72.72%	72.50%
0.7	0.75	73.28%	75.14%	74.75%	73.61%	74.82%	74.52%	72.57%

Table 3.4: *PI of the ANNs with more than four neurons in the hidden layer.*

Percentage Improvement (PI):

$$PI = \frac{\sum_{i=1}^{19}(1 - \frac{MSE_i^{NN}}{MSE_i^{Wei}})}{19} \cdot 100 \tag{3.4}$$

here, i is the number of study cases, MSE_i^{NN} is the MSE of the network for the i^{th} case, and MSE_i^{Wei} is the MSE of the Weibull distribution for the i^{th} case. PI tends to 100% for a good approximation, since it means that MSE^{NN} is near 0. A PI tending to 0% represents an approximation of the ANN approach similar to that obtained by the Weibull distribution. Finally, a PI<0 means that the ANN performed worse than the Weibull approach. Considering the 19 study cases, feed-forward neural networks with more than four neurons in their hidden layer have an average PI = 71.78%, taking into account the different values of the learning rates and momentum. Table 3.4 shows the PI of the 63 neural networks with more than four neurons in their hidden layer. The columns represent the number of neurons in the hidden layer.

ANNs with one or two neurons in their hidden layer are not reliable. Often they have a fitting problem and the approximation of $F(t)$ is poor if compared to the Weibull distribution. Considering the learning parameters, the neural networks with three or four neurons in their hidden layer can obtain a better approximation than the Weibull distribution only if they have a momentum smaller than 0.75. A momentum too high, combined with too few neurons in the hidden layer, increases the instability of the network and, apparently, does not allow the network to find adequate solutions. In any case, the neural networks with three or four neurons in their hidden layer with momentum smaller than 0.75 have an average value of PI equal to 68.86%. For the networks with more than five neurons in their hidden layer, this does not happen. Rather, selecting a high momentum (in our case equal to 0.75) these networks increase their performance: in fact, the average PI improves from 71.78% to 74.1% for these networks, and with respect to all the 63 networks with more than five neurons in their hidden layer, the

Cluster	Cod	N° of neurons in the hidden layer							Average	Cluster Average	Weibull	Cluster Weibull
		5	6	7	8	9	10	11				
A	cod U	9.06%	7.64%	8.95%	5.81%	7.27%	5.96%	6.71%	7.34%	6.01%	24.29%	20.83%
	cod V	5.22%	4.16%	4.78%	4.39%	4.60%	4.76%	4.83%	4.68%		17.37%	
B	cod A	5.54%	4.94%	5.18%	5.66%	5.10%	4.89%	5.37%	5.24%	5.12%	5.74%	9.92%
	cod B	4.58%	4.44%	4.52%	4.66%	4.62%	4.37%	4.31%	4.50%		15.87%	
	cod L	6.56%	6.60%	6.51%	6.70%	6.55%	6.58%	6.62%	6.59%		8.25%	
	cod N	3.98%	4.02%	4.10%	4.30%	4.09%	4.28%	4.20%	4.14%		9.81%	
C	cod T	6.33%	6.04%	6.43%	6.60%	6.47%	5.90%	6.18%	6.28%	7.54%	15.41%	13.71%
	cod O	7.58%	7.62%	7.62%	7.79%	7.83%	7.81%	7.72%	7.71%		10.36%	
	cod F	6.29%	6.29%	6.25%	6.37%	6.37%	6.53%	6.27%	6.34%		14.89%	
	cod M	9.52%	9.67%	9.99%	9.97%	9.53%	9.80%	10.25%	9.82%		14.17%	
D	cod Q	7.24%	7.16%	7.18%	7.17%	7.26%	7.20%	7.25%	7.21%	7.11%	23.28%	15.97%
	cod R	10.88%	10.73%	10.71%	11.11%	10.98%	11.34%	11.31%	11.01%		22.45%	
	cod I	6.77%	6.72%	6.76%	6.74%	6.73%	6.78%	6.75%	6.75%		7.08%	
	cod H	8.14%	8.47%	8.36%	8.25%	8.44%	8.46%	8.42%	8.36%		15.51%	
	cod D	4.60%	4.68%	4.64%	4.76%	4.73%	4.73%	4.72%	4.69%		8.41%	
	cod E	7.08%	6.93%	7.12%	7.00%	7.03%	7.32%	6.96%	7.06%		10.69%	
	cod C	7.29%	7.26%	7.16%	7.32%	7.21%	7.25%	7.24%	7.25%		12.38%	
	cod S	5.31%	5.44%	4.53%	4.73%	4.53%	3.96%	4.49%	4.71%		12.70%	
	cod P	6.87%	7.08%	6.70%	6.99%	6.87%	7.03%	7.23%	6.97%		31.25%	

Table 3.5: *The average percentage errors of the networks for the different clusters.*

networks with momentum equal to 0.75 produce the best approximation 83.46% of the time.

It is reasonable to assume that the larger the number of failure times available, the better the approximation produced by both ANNs and Weibull, because more data are available to align the curves. The Weibull distribution, in fact, needs a certain number of data points to adapt its parameters, and the neural networks need data for the learning phase. The number of available data then can play an important role on the effectiveness of the failure analysis. For this reason, Table 3.5 shows the results obtained dividing the data sets into clusters, based on the number of failure times available. In particular clusters contain components having each respectively: less than 25 failure times, from 25 to 49 failures times, from 50 to 99 failures times and more than 100 failures times. This subdivision into different clusters can be helpful for various necessities in the real world. Indeed, the life cycles of different components could be very different each other. For example, the life cycle of an elevator's gear motor is completely different from the life cycle of the car bulb.

According with the results, large values of learning rate are associated with a better approximation, but only for networks with more than four neurons in their hidden layer. In fact, for networks with only one or two neurons in their hidden layer, the opposite is true: the smaller the learning rate, the better the approximation. Networks with three or four neurons in their hidden layer better approximate with a learning rate close to 0.5. The momentum also influences the performance of the networks. For ANNs with one or two neurons in their hidden layer the best performances are obtained with a small momentum. For networks with three or four neurons in their hidden layer the best performances are obtained with a momentum equal to 0.5. Instead, considering the networks with more than four neurons in their hidden layer, ANNs with a high momentum achieve best performances.

The execution time plays a fundamental role when it is necessary to select an approach. Different values of the learning parameters minimally influence the execution

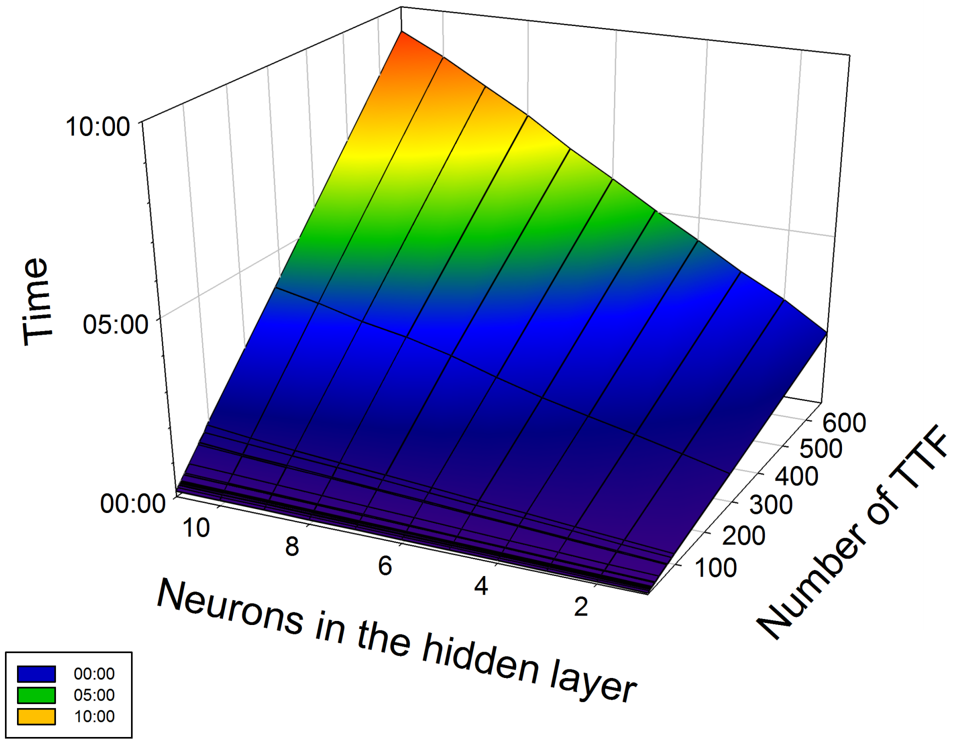


Figure 3.4: *The execution times of the learning process changing the networks topology.*

time, with a maximum variations range about 5%–6%. Regardeless the momentum and learning rate, the execution time depends mainly on the number of data in input and number of neurons in the hidden layer. In particular, the execution time varies linearly with the incrising of these two elements. Using an Intel®Core™i5 CPU M 560 2.67 GHz 2.66 GHz and RAM 4.00 GB, this requires six seconds for each failure in a data set in input, and five seconds for each neuron within the hidden layer to produce the results. In Figure 3.4, it is possible to see the variation of the execution time with the change in the number of neurons in the hidden layer and the number of failure times in the data set in the input.

Summarizing, considering the results presented, it is preferable to use 3-layer feed-forward neural networks with five or more neurons in their hidden layer to predict the cumulative failure rate of mechanical and electrical components. However, the execution time increases with the number of neurons in the hidden layer. Considering also that the learning parameters do not particularly influence the execution time of the learning process, the better performance occur with learning rates in [0.5–0.7] range, and a momentum around 0.75.

3.1.3 Machine learning models

This second part of the section focuses in the prediction of the reliability function of the same mechanical and electrical components. Beyond the ANNs and the linear regression, the performance of other three machine learning (ML) models were tested. These models are based on different paradigms and methodologies to be as diverse as possible, from classical regression to soft-computing. In this case, the study focuses also in the presence or absence of censored data. In particular, the main purpose is to understand if there is a single data-driven learning model able to outperform other models.

Artificial neural networks (MLP-CG)

Also in this case, a classical multilayer perceptron algorithm was used, i.e., it is a fully connected feed forward neural network, which is a directed graph constructed over multiple layers of nodes, where each layer is fully connected to the next one. A conjugate gradient method to learn the weights of a neural network was used. This method is an iterative search method that comes from the conventional numerical analysis. This approach uses a scaled conjugate gradient in order to speed up the learning process.

Linear regression (LMS–LR)

Linear regression is a technique for modeling the relationship between a scalar dependent variable (output) and one or more explanatory variables (input) [68]. The case of one explanatory variable is called *simple linear regression*. Instead, when there are more than one explanatory variable, the process is called multiple linear regression.

Least mean squares is a standard approach, very often used as a baseline, for the regression problems. The algorithm aims at fitting a straight line to the observed data. It is based on the idea of minimizing the sum of squares of the errors, obtained from the difference between an observed value of a point and the value provided by the fitted function [183].

Support vector regression (SVR)

In a support vector regression model the input examples are mapped into the high-dimensional feature space through a non-linear mapping (kernel) selected a priori. In this feature space SVR constructs a linear decision surface being a hyperplane in the original input feature space. Precisely, this hyperplane is a linear decision function with maximal margin between the examples of the different categories in order to separate them as much as possible. The maximal margin is determined by the support vectors, a small amount of the training data representing the decision boundary. Testing examples are mapped into the same feature space and predicted to belong to a category based on which side of the hyperplane they fall on. This method was firstly proposed for classification [57] and later on for regression [73].

Random forests (RF)

The ensemble learning systems are well-recognized machine learning tools capable of obtaining better performance than a single component model. They are able to deal with complex and high dimensional regression and classification problems [120]. They combine the output of the machine learning systems, in the literature called *weak learners*, from the group of learners in order to get smaller prediction errors (in regression) or lower error rates (in classification). Their performance strongly relies on diversity of the weak learners, in the way that ideally they make their errors on different parts of the problem space.

Random forests is a state-of-the-art ensemble algorithm for regression and classification, that constructs a multitude of diverse decision trees during the training phase and aggregates the results provided by all the component decision trees to compute the final output during the prediction phase [38]. For the classification problems it outputs the class that is the mode of the classes and for regression it return mean prediction of the individual trees. It is based on a bagging algorithm [37], which jointly with random trees playing a role of weak learners, is used to construct a forest.

Fuzzy system (EFS-SA)

Fuzzy systems, which are based on fuzzy logic, became popular in the research community, since they have ability to deal with complex, non-linear problems being too difficult for the classical methods [227]. Besides, its capability of knowledge extraction and representation allowed them to become human-comprehensible to some extent (more than classical black-box models) [6]. The lack of the automatic extraction of fuzzy systems have attracted the attention of the soft computing community to incorporate learning capabilities to these kinds of systems. In consequence, a hybridization of fuzzy systems and evolutionary algorithms has become one of the most popular approaches in this field [56,97]. In general, evolutionary fuzzy systems are fuzzy systems enhanced by a learning procedure coming from evolutionary computation, i.e. considering any evolutionary algorithm.

In particular, a fuzzy rule learning method based on genetic programming grammar operators and simulated annealing was used [187]. This hybrid approach is based on the fuzzy rule-based system with fuzzy if-then rules. It learns them and the fuzzy partitions by using a tree-based codification particular for genetic programming. The authors define specific genetic programming grammar operators for this kind of codification and then adapt them to the simulated annealing algorithm [114]. Thus, the learning mechanism is a hybrid method between simulated annealing and genetic programming.

The values of the selected parameters defining the used algorithms are presented in Table 3.6. All the experiments were ran on an Intel quadri-core i5-2400 3.1 GHz processor with 4 GBytes of memory, under the Linux operating system.

Results and discussion

The results obtained using ML approaches and the Weibull distribution based technique for the MRM (complete data) are analyzed. Table 3.7 presents the results obtained showing the MSE metric for the algorithms considered. The best result for a given dataset is presented in bold font. The average values over the datasets are reported at the bottom of the table.

Firstly, it is possible to observe that although the classical method for the prediction of the component reliability based on the Weibull distribution obtains good accuracy results on average (.0038), the ML approaches such as RF, MLP-CG and EFS-SA obtain better accuracy results on average (.0016, .0027 and .003, respectively). That however does not happen for SVR and LMS. As expected LMS, which is a simple regression algorithm, obtains the worst accuracy results on average. In contrast, RF confirms its high potential with these data obtaining the lowest accuracy results on average (.0016).

Table 3.6: *Parameters of the algorithms for the experiments.*

Algorithm	Parameters
LMS	-
SVR	- kernel type = radial - C = 0.5 - degree = 3 - gamma = 0.15 - coef = 0 - tolerance of termination criterion = 0.001 - epsilon = 0.1
RF	- nr of trees = 1000 - size of a node = 1 - nr of variables randomly sampled at each split = 1
MLP-CG	- nr of hidden nodes = 10 - nr of iterations (GD) = 10000
EFS-SA	- nr of labels = 3 - nr of rules = 8 - delta (SA) = 0.5 - nr of iterations (SA) = 10000 - mutation probability (SA) = 0.5 - mutation amplitude (SA) = 0.1 - initial probability for accepting (SA) = 0.5 - final probability for accepting (SA) = 0.5 - nr of individuals to be analyzed for each iteration (SA) = 10

Table 3.7: *Results for the median rank method.*

	LMS	SVR	RF	MLP-CG	EFS-SA	Weibull
COD A	.0015	.0006	**.0001**	.0006	.0012	.0003
COD B	.0006	.0011	**.0001**	.0006	.0009	.0028
COD C	**.0021**	.0090	.0026	.0023	.0017	.0036
COD D	**.0009**	.0032	.0010	.001	.0018	.0011
COD E	.0147	.0104	.0032	.0029	.0029	**.0028**
COD F	.0159	.0021	**.0006**	.0008	.0007	.0021
COD H	.0071	.0046	.0021	**.0015**	.0026	.0044
COD I	.0022	.0035	.0020	.0014	.0019	**.0010**
COD L	.0053	.0010	**.0005**	.0015	.0055	.0006
COD M	.0097	.0055	**.0012**	.0051	.0082	.0061
COD N	.0011	.0009	**.0002**	.0008	.0032	.0021
COD O	.0011	.0017	.0010	**.0008**	.0009	.0015
COD P	.0179	.0290	.0075	**.0027**	.0063	.0176
COD Q	.0279	.0093	**.0018**	.0179	.0068	.0059
COD R	.0113	.0089	**.0020**	.0031	.0027	.0085
COD S	.0096	.0105	.0035	**.0031**	.0017	.0039
COD T	.0183	.0082	**.0005**	.0042	.0048	.0051
COD U	.0002	.0003	**0**	.0001	.0019	.0023
COD V	.0002	.0003	**0**	.0001	.0004	.0014
Avg.	.0078	.0058	**.0016**	.0027	.0030	.0038

Moreover, considering individual component data sets RF outperforms the other approaches in 11 out of 19 cases.

Analysis of results with censored data

Regattieri et al. [175] showed that the neglecting of the censored information results in significant errors in the evaluation of the reliability performance of the components using the Weibull distribution. In the same way, this subsection analyzes the results obtained by the ML algorithms and the Weibull distribution using the censored data, the PLE method.

Table 3.8 presents the results obtained showing the MSE metric for the algorithms considered. The best result for a given dataset is presented in bold font. The average values over the datasets are reported at the bottom of the table. The results show very similar tendency as with the complete data (MRM method). Again, the Weibull approach obtains good accuracy results on average (.0024). Two ML approaches such as

Table 3.8: *Results for the PLE method.*

PLE	LMS	SVR	RF	MLP-CG	EFS-SA	Weibull
COD A	.0011	.0006	**.0001**	.0003	.0012	.0003
COD B	.0005	.0012	**.0001**	.0005	.0005	.0023
COD D	.0010	.0026	**.0009**	**.0009**	**.0009**	.0011
COD E	.0136	.0093	.0030	**.0025**	.0033	.0025
COD I	.0019	.0032	.0019	**.001**	.0017	.0009
COD L	.0056	.0010	**.0004**	.0008	.0051	.0007
COD M	.0095	.0053	**.0011**	.0036	.0094	.0061
COD T	.0206	.0099	**.0005**	.0022	.0061	.0070
COD U	.0002	.0003	**0**	.0001	.0006	.0022
COD V	.0002	.0003	**0**	.0002	.0004	.0013
Avg.	.0054	.0034	**.0008**	.0012	.0029	.0024

RF and MLP-CG obtain lower accuracy results than the classical Weibull distribution on average (.0008 and .0012, respectively). LMS turns out to be the worst performing approach on average (.0054). One more time RF turns out to be the best performing approach. Individually, RF outperforms the other approaches in 7 out of 10 cases (+1 tie) and globally RF obtains the lowest average results (.0008).

Table 3.9 presents the difference between the results obtained for PLE (censored) and MRM (uncensored) for each algorithm used. The negative values (in bold font) means that machine learning methods perform better for the censored data (obtain lower MSE). This happens for 36 out of 50 cases (in 6 cases the difference is equal to 0). In particular, the Weibull approach has no difference between the accuracy of censored and uncensored data on average. Four out of five ML approaches, all besides EFS-SA, obtain lower accuracy for the censored data on average. Namely, SVR obtains the lowest value on average (-.0007), just after there are LMS and MLP-CG (-.0006). RF is in the third place with slightly higher value (-.0005). It is clear from Table 3.9 that using censored time improves the performance of all the ML methods except EFS-SA.

Impact of datasets' clusters on the performance of the methods

As well as for the cumulative failure distribution, the behavior of ML methods is studied clustering the components based on their dataset size. In fact, all the ML methods improve their learning phase having more data available. Table 3.10 presents the results of the machine learning methods for the four clusters considering the MRM method. We can see how RF outperforms the other approaches for all the four clusters considering the MRM method.

Table 3.9: *Results showing the differences between the PLE (censored) and the MRM (uncensored) method.*

	LMS	**SVR**	**RF**	**MLP-CG**	**EFS-SA**	**Weibull**
COD A	**- .0005**	.0000	.0000	**- .0003**	.0000	.0000
COD B	**- .0001**	.0001	.0000	**- .0001**	**- .0004**	**- .0004**
COD D	**- .0011**	**- .0064**	**- .0017**	**- .0014**	**- .0008**	**- .0025**
COD E	.0127	.0061	.0020	.0015	.0015	.0014
COD I	**- .0128**	**- .0072**	**- .0012**	**- .0019**	**- .0012**	**- .0019**
COD L	**- .0103**	**- .0011**	**- .0002**	.0000	.0044	**- .0014**
COD M	.0023	.0007	**- .0010**	.0021	.0068	.0017
COD T	.0184	.0063	**- .0016**	.0008	.0042	.0061
COD U	**- .0051**	**- .0007**	**- .0005**	**- .0014**	**- .0049**	.0016
COD V	**- .0094**	**- .0053**	**- .0012**	**- .0049**	**- .0078**	**- .0048**
Avg.	**- .0006**	**- .0007**	**- .0005**	**- .0006**	.0002	.0000

3.1.4 Discussion

In this section, different approaches to predict the reliability parameters of mechanical and electric components have been analyzed. At the beginning, an exploration of the capacity of artificial neural networks was presented. A first approach to this model can be complex, for this reason the exploration concerned the performance of simple networks. Networks with different numbers of neurons in their hidden layer and with different values for their parameters (learning rate and momentum) were analyzed, in order to understand how these factors influence their performance and execution time needed for the learning phase. The results showed how simple networks can achieve satisfactory performance, even better than the traditional approach with the Weibull distribution, particularly in conditions of poor data. Then, this section analyzed how other machine learning models outperform the Weibull distribution. In particular, random forests emerged as the best model in average, even in the presence of few failure times. Additionally, the results showed that machine learning models can get better results in presence of censored data than when using a complete set. This research shows one more time the potential of the machine learning approach in the reliability analysis. The presented results can be used to define some threshold value or trend in order to predict the need of certain maintenance action, becoming a useful contribution in the field of maintenance engineering.

Cluster	Cod	LMS	SVR	RF	MLP-CG	EFS-SA	Weibull
A	cod U	.0002	.0003	**.0000**	.0001	.0004	.0023
	cod V	.0002	.0003	**.0000**	.0001	.0019	.0014
Avg.		.0085	.0036	**.0008**	.0022	.0037	.0028
B	cod A	.0015	.0006	**.0001**	.0006	.0012	.0003
	cod B	.0006	.0011	**.0001**	.0006	.0009	.0028
	cod L	.0053	.0010	**.0005**	.0015	.0055	.0006
	cod N	.0011	.0009	**.0002**	.0008	.0032	.0021
Avg.		.0021	.0009	**.0002**	.0009	.0027	.0015
C	cod T	.0053	.0010	**.0005**	.0015	.0055	.0006
	cod O	.0011	.0017	.0010	**.0008**	.0009	.0015
	cod F	.0159	.0021	**.0006**	.0008	.0007	.0021
	cod M	.0097	.0055	**.0012**	.0051	.0082	.0061
Avg.		.0113	.0044	**.0008**	.00027	.0037	.0037
D	cod Q	.0279	.0093	**.0018**	.0179	.0068	.0059
	cod R	.0113	.0089	**.0020**	.0031	.0027	.0085
	cod I	.0022	.0035	.0020	.0014	.0019	**.0010**
	cod H	.0071	.0046	.0021	**.0015**	.0026	.0044
	cod D	.0009	.0032	**.0010**	**.0010**	.0018	.0011
	cod E	.0147	.0104	.0032	.0029	.0029	**.0028**
	cod C	.0021	.0090	.0026	.0023	**.0017**	.0036
	cod S	.0096	.0105	.0035	.0031	**.0017**	.0039
	cod P	.0179	.0290	.0075	**.0027**	.0063	.0176
Avg.		.0104	.0098	**.0029**	.0040	.0032	.0054

Table 3.10: *Results of the components' clusters based on the dataset size for the MRM method.*

3.2 Renewable energy

The term renewable energy includes clean and abundant energy gathered from self-renewing resources such as solar radiation, wind, biomass (plant crops), ocean waves and rivers, geothermal heat, and other such continuing resources. Reliable energy supply is essential in all economies. In fact, these resources can be used to produce electricity for all economic sectors, fuels for transportation, and heat for buildings and industrial processes [211]. Virtually all regions of the world have renewable resources of one type or another.

The scientific community and practitioners often stress the crucial role of having reliable climatological data as a key input for a wide set of anthropic activities and operative sectors [60]. The prediction of some characteristic related with the production of renewable energy, e.g., the amount of solar radiation that will "hit" a specific area, or the amount of wind that will blow in another one, is crucial for the production and distribution of the energy. The availability of these climatological data over time and space is frequently required, but in several geographical contexts, direct measures present spatial and temporal lacks so that predictive approaches become the unique way to estimate such data. Their lack is the basic limit to the possibility of developing models and applications for the design of agricultural, industrial and energy systems.

This section presents two different approaches for the prediction of renewable energy. The first one concerns the use artificial neural networks to predict the global solar radiation over several Italian places. The use of these models for solar prediction are of global interest, and it was already used in literature. However, this study contributes to the intent of creating a panel of location dependent methods which cover almost all the most promising geographical regions, allowing in this way the comparisons and multi-site analysis. In fact, the relatively high initial investment costs for the installation of photovoltaic panels require a deep study concerning the amount of solar radiation available in the area.

The second part of the section presents the results of [42]. In this work, a model that considers some important factors concerning the electric energy distribution is presented by modeling context-aware agents able to identify the impact of these factors. Moreover, some tests are performed regarding the web service integration in which agents contracting energy will automatically retrieve data to be used in adaptive and collaborative aspects. An explicative example is represented by doing weather prediction that provides input on ongoing demand and data for the predicted availability, in case of photovoltaic and wind powered environments.

3.2.1 Solar radiation

Weather stations, located in several geographical sites under the control and management of the national institutions or international research centers, directly measure a major set of climatological data that are, consequently, immediately available to use. Databases storing climatological time series are even more diffuse and shared. Nevertheless, wide geographical areas are, still, partially or totally excluded from the so-called climatological atlas especially for uncommon climatological parameters, for which the direct measure is complex and expensive. *Global solar radiation*, and its components, is among the climatological parameters not systematically measured by all the weather stations [31]. If measured, the available time series may be incomplete or particularly limited in time and space to be of interest for the operative uses. To tackle such a lack, models and approaches, following multiple strategies and methods, are developed with the purpose of estimating the solar radiation levels starting from the data available for the considered locations. Multiple contributions are proposed in the past decades and the recent research trends are outlined in the literature reviews of the topic [24,111]. An useful classification criteria of the proposed contributions considered: (1) geographical context, i.e., mono versus multi-location models; (2) overall adopted dataset, i.e., regression models based on past time series versus correlation models based on climatological databases; and (3) adopted technique to estimate the solar radiation level. According to such a last criterion, mathematical linear and nonlinear correlation and regression models are introduced, together with fuzzy logic based methods, decomposition models and artificial intelligence techniques [201,207]. As already mentioned in this thesis, artificial neural networks (ANNs) are known as an effective predictive technique in several fields. Their application to the solar radiation estimation issues increases a lot in the recent past due to the promising results, so that several authors conclude about their performance superiority toward the other prediction methods [23, 168]. The application of ANNs to predict the solar radiation, using different climatological and geographical variables, was already explored. Both global solar radiation and its fractions, measured on horizontal and tilted terrestrial surfaces, are of interest and they constitute the output parameter to estimate [226].

The aim of this section is to present the adoption of ANNs to estimate the horizontal global radiation over the Italian territory. Italy belongs to the so-called *sun belt area* so that the interest in the availability of reliable solar radiation levels is particularly strong due to the economic and environmental potential of energy plants and applications based on such a source. Furthermore, Italian weather stations do not capillary measure solar radiation. Predictive approaches become of interest. Concerning the existing contributions on the solar energy prediction for Italy, they are particularly limited and no study is about the application of ANNs over Italy from a multi-location perspective. The existing models focus on a single location, only. On the contrary, the proposed study is based on monthly average data from 45 locations, enlarging the applicability of the obtained outcomes to the whole Italian peninsula. For each location, 14 parameters are included, i.e. 13 input parameters and 1 output parameter. The adopted dataset is made of 7560 values. A multi-layer feed forward perceptron is adopted to predict the monthly average global radiation. In addition, a sensitivity analysis is assessed to identify the input which most affect the accuracy of the solar radiation prediction. Results are compared through the most frequently adopted statistical key performance indexes (KPIs), as the MAPE between the predicted and the measured values. The key elements of innovation, of interest for the scientific community and the practitioners, deal with the considered geographical context, already not studied in depth and belonging to the sun belt area for which contributions are strongly expected. Furthermore, the multi-location perspective is able to develop a model valid for a wide area and the large input dataset, made of 14 parameters, allows pointing out the most relevant ones through a sensitivity analysis. The steps of this study were the following:

1. Selection of the variable to predict, e.g. global, diffuse or beam solar radiation on horizontal or tilted surface;

Table 3.11: *Geographical and climatological parameters used in this study.*

	Input	Unit
1	Average temperature	°C
2	Average daily cloudiness	Okta
3	Average TOA daily radiation	MJ/(m^2·day)
4	Clear sky days	days/month
5	Cumulated rainfall	mm/month
6	Rainy days	days/month
7	Latitude	°North
8	Longitude	°East
9	Altitude	m a.s.l.
10	Heating degrees day	°C
11	Average day duration	hour
12	Average sunshine duration	hour
13	Average wind speed	m/s
14	Period (month number)	-

	Output	Unit
1	Average daily solar radiation	MJ/(m^2·day)

2. Selection of the input parameters and data collection, e.g. climatological data, geographical coordinates, past time series, and so on;
3. Definition of the training and testing sets;
4. Development of the ANN model and training phase assessment;
5. Calculation of the prediction error, through statistical indexes, adopting the testing set;
6. Comparisons and final choice of the best ANN.

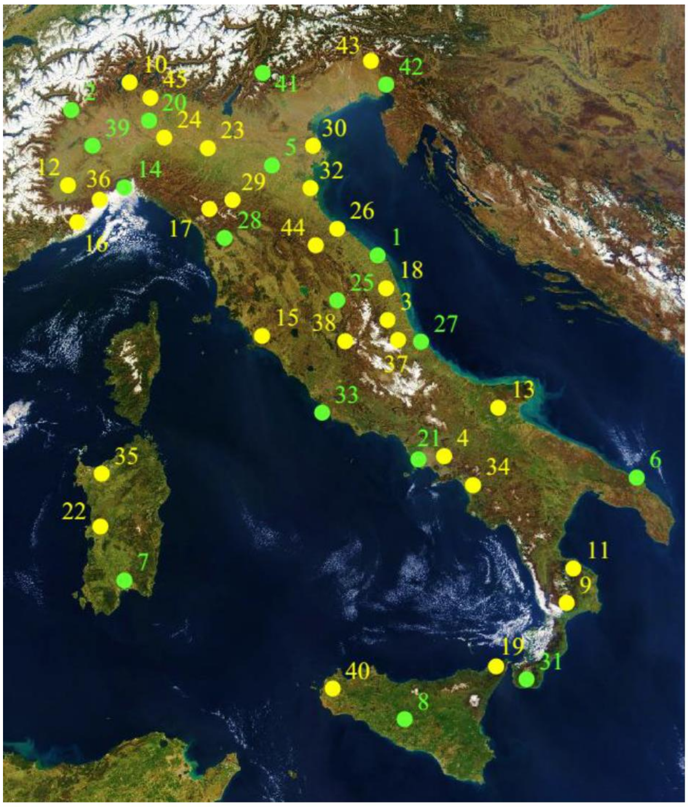
Input data

For each of the 45 Italian locations chosen for the ANN training and testing, a set of 14 parameters about both the geographical and climatological data was collected, Table 3.11. All the data are the monthly parameters referred to the so-called test reference year (TRY), so that no 196 biases due to the annual anomalies occur. The TRY is derived from the time series collected by the Italian national agency for new technologies, energy, and sustainable economic development (ENEA)[1].

Furthermore, the considered Italian locations are wide spread within the entire Italian country, from north to south. 540 (input, output) couples are provided. All the data are divided into two different subsets:

- Training set composed by 17 cities, about 38% of the whole set;
- Testing set including the remaining 28 cities, about 62% of the whole set.

[1]Climatic archive available online: http://clisun.casaccia.enea.it/Pagine/Index.htm. Accessed: January 2015.



Locations			
1	Ancona	24	Pavia
2	Aosta	25	Perugia
3	Ascoli Piceno	26	Pesaro
4	Avellino	27	Pescara
5	Bologna	28	Pisa
6	Brindisi	29	Pistoia
7	Cagliari	30	Ravenna
8	Caltanissetta	31	Reggio Calabria
9	Catanzaro	32	Rimini
10	Como	33	Roma
11	Crotone	34	Salerno
12	Cuneo	35	Sassari
13	Foggia	36	Savona
14	Genova	37	Teramo
15	Grosseto	38	Terni
16	Imperia	39	Torino
17	Lucca	40	Trapani
18	Macerata	41	Trento
19	Messina	42	Trieste
20	Milano	43	Udine
21	Napoli	44	Urbino
22	Oristano	45	Verbania
23	Parma		

Figure 3.5: *Locations for the ANN training (green) and testing (yellow).*

Considering a training set smaller than the testing set guarantees the reliability of the ANN predictions for the whole Italian territory, in case the testing network performances are good. The locations chosen for both the training and the testing phases are 209 uniformly distributed over Italy. Figure 3.5 synthetically depicts all meteorological stations located in the main Italian cities. The green bullets refer to the 17 locations selected for the ANN training, while the remaining 28 yellow bullets represent the testing locations. The latitude ranges from 37.08° North of Caltanissetta to 46.18° North of Trento, while the longitude is between 7.53° East (Aosta) and 17.95° East (Brindisi). Finally, the altitude ranges between 2m and 1933m a.s.l., so that both mountain and plain climates are included.

Because of the input variables are combined within the ANN, large input values together with small ones can undermine the training phase leading to local sub-optima. To avoid such a problem, before the training phase, all the input have 220 to be normalized [144]. Eq. 3.5 presents the normalization criterion.

$$y = y_{min} + \frac{x - x_{min}}{x_{max} - x_{min}}(y_{max} - y_{min}) \tag{3.5}$$

where $x \in [x_{min}, x_{max}]$ is the original data value and $y \in [y_{min}, y_{max}]$ is the corresponding normalized variable. The adopted normalization range is [0.1, 0.9].

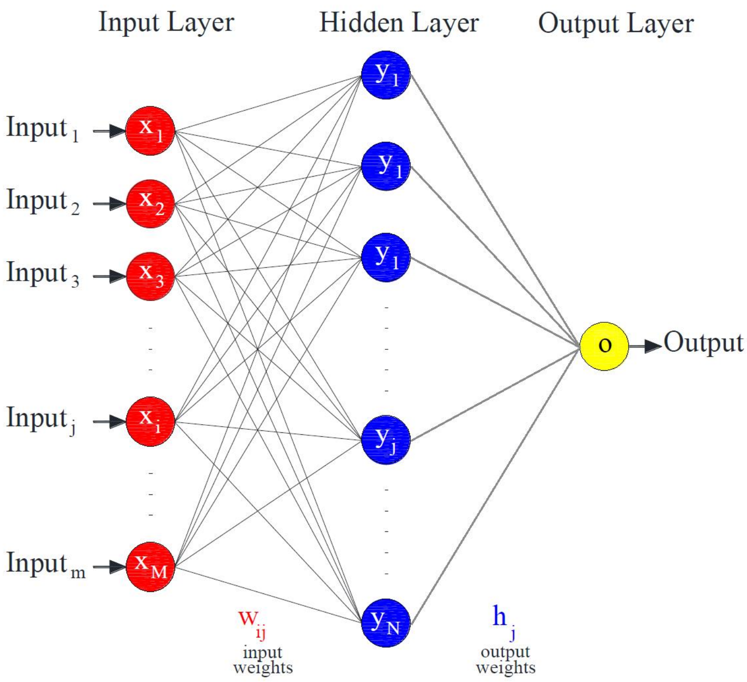


Figure 3.6: *Structure of the adopted multi-layer perceptron.*

ANN model

The ANN adopted in this work is the multi-layer perceptron (MLP) with one hidden layer. Such ANN is among the feed-forward ANNs and they are widely adopted in the climatological sector [212] due to their property of being able to fit with the measurable functions with good accuracy. Figure 3.6 depicts the structure of the adopted MLP, where x_i are the input of the first layer (the climatic components), connected to each neuron j of the hidden layer, and y_j are the responses of hidden layer neurons, computed in accordance with the log-sigmoid function in Eq. 3.6. The responses of the output layer neurons o is a linear combination of h_j as in Eq. 3.7, giving the predicted solar radiation. The parameters w_{ij} and h_j in these equations are the weights of the connections between the input-hidden and hidden-output layers.

$$y_j = f(x_1, x_2, ..., x_i, ..., x_M) = \frac{1}{1 + e^{-(\sum_{i=1}^{M} w_{ij} \cdot x_i + w_{oj})}} \tag{3.6}$$

$$o = f(y_1, y_2, ..., y_i, ..., y_N) = \frac{1}{1 + e^{-(\sum_{j=1}^{N} h_j \cdot y_j + h_o)}} \tag{3.7}$$

In this study, the back–propagation (BP) learning algorithm was adopted, and the mean absolute percentage error (MAPE) was used to measure the accuracy of the prediction. Starting from the study of the ANN structure, a free variable to fix is the best combination of neurons within the hidden layers. This value is related to each specific instance and to the particular dataset [212]. As a consequence, to evaluate the number of the hidden neurons leading to the best results, all configurations with a neuron number from 2 to 20 are compared. Figure 3.7 shows the MAPE trend for the considered instance

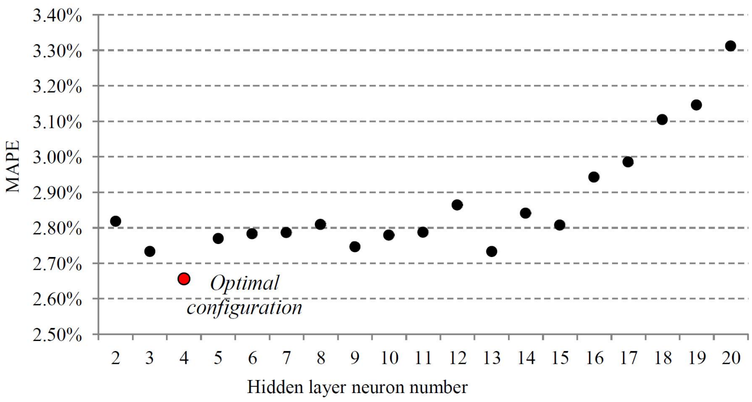


Figure 3.7: *Hidden layer neuron number optimization.*

City	MAPE	City	MAPE	City	MAPE	City	MAPE
Como	2.18%	Pistoia	3.22%	Ascoli Piceno	1.50%	Oristano	3.29%
Cuneo	3.72%	Ravenna	3.31%	Avellino	2.76%	Pesaro	2.43%
Grosseto	2.73%	Rimini	2.55%	Catanzaro	3.22%	Salerno	3.45%
Imperia	3.38%	Savona	3.81%	Crotone	4.16%	Sassari	3.34%
Lucca	2.19%	Terni	2.43%	Foggia	1.39%	Teramo	1.25%
Parma	2.44%	Udine	1.58%	Macerata	2.05%	Trapani	1.39%
Pavia	2.20%	Verbania	3.69%	Messina	2.73%	Urbino	2.00%
Avg. 2.66%							

Table 3.12: *MAPE for the testing locations adopting all the 14 input parameters.*

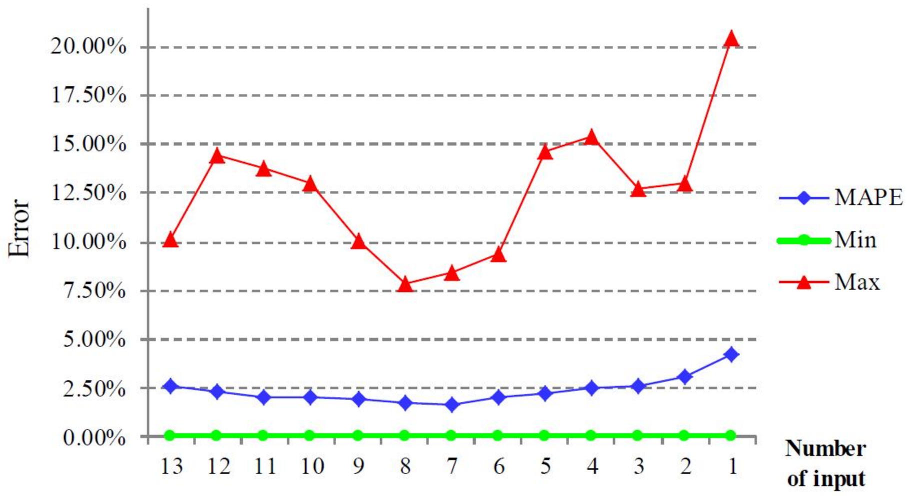
varying the number of neurons in the hidden layer. Evidences show that the ANN with 4 neurons in the hidden layer is the best configuration for this study.

Result analysis and discussion

The adoption of all the 14 available input for both the training and testing phases leads 246 to a MAPE value of about 2.66%. Table 3.12 lists the MAPE values for each location in the testing set.

To both improve the overall ANN performances and to understand 250 the relevance of each input parameter the automatic relevance determination (ARD) Bayesian method is adopted [122]. This approach, available in the Matlab©toolbox [151], creates an ordered list of the input by turning off one input at a time from the last to the first, sorting them by relevance [219]. Figure 3.8 shows the global solar radiation minimum, mean and maximum absolute percentage errors for all scenarios varying the input parameter number in accordance with the ARD method. For each scenario the considered input are marked.

The analysis shows that the increase of the input number does 259 not necessarily improve the performance of the ANN. The ARD method allows discarding irrelevant inputs (such as the longitude and the sunshine duration) that disturb the ANN training phase. The best results are for 7 input, i.e. TOA radiation, day duration, rainy days, altitude, rainfall, period, and latitude. This scenario leads to a MAPE of 1.67% and a

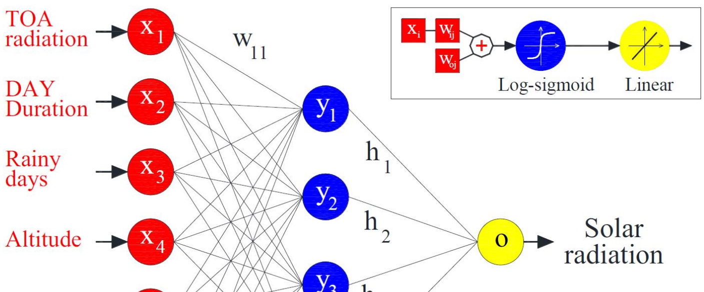


Input relevance	13	12	11	10	9	8	7	6	5	4	3	2	1
TOA radiation	X	X	X	X	X	X	X	X	X	X	X	X	X
Day duration	X	X	X	X	X	X	X	X	X	X	X	X	
Rainy days	X	X	X	X	X	X	X	X	X	X	X		
Altitude	X	X	X	X	X	X	X	X	X	X			
Rainfall	X	X	X	X	X	X	X	X	X				
Period	X	X	X	X	X	X	X	X					
Latitude	X	X	X	X	X	X	X						
Cloudiness	X	X	X	X	X	X							
Degrees day	X	X	X	X	X								
Clear sky days	X	X	X	X									
Average temperature	X	X	X										
Sunshine duration	X	X											
Longitude	X												

Best input scenario

Figure 3.8: *Performances of the testing phase varying the input number.*

Hidden layer weights			
W11	45.50644438	W13	3.37216327
W21	9.134000603	W23	0.082976493
W31	6.762185985	W33	-0.10903981
W41	-26.0090962	W43	-0.04410531
W51	-7.00181698	W53	0.105902685
W61	-7.00925978	W63	-0.06610606
W71	-16.7577145	W73	-0.17336334
W12	-13.0059768	W14	-7.88624175
W22	-3.34521693	W24	-0.49021099
W32	-0.53080478	W34	0.957948576
W42	4.807841869	W44	0.265951265
W52	-3.45569082	W54	0.232006662
W62	-1.10032843	W64	-1.24341316
W72	-1.42083641	W74	0.807948454
Biases			
W01	-44.0699144	W03	0.288562591
W02	-2.32719821	W04	7.841278368



Output layer weights	
h0	-2.34852
h1	34.42109
h2	-4.42676
h3	7.070229
h4	-3.88423
Bias h0	-2.34852

Figure 3.9: *ANN optimal configuration features.*

maximum absolute percentage error of 8.42%, against, respectively, 2.66% and 10.14% for the scenario with all the 14 input. By further reducing the number of input, the worsening of the MAPE is evident. The adoption of the most critic parameter, only, the TOA radiation, leads the MAPE to increase to 4.25% with a maximum absolute percentage error of about 20.40%. Moreover, the small input number makes this approach usable within almost all the Italian territory since it is necessary to field-measure the monthly cumulated rainy days and the rainfall, while the other 5 input data are immediately available (TOA radiation, day duration, altitude, latitude and period). The following Figure 3.9 details all the parameters necessary to implement the obtained best ANN. The weights and biases are presented together with the ANN structure. In such a configuration, the MAPE is equal to 1.67% with a narrow standard deviation of 1.58%, and a 279 maximum percentage error of 8.54%. Furthermore. the ANN predictions, when compared to the testing values, show a good fitting.

Figure 3.10 depicts the couples of the measured and the predicted data showing a very high correlation coefficient, $R^2 = .9984$. Furthermore, from the analysis of the results, the ANN performances are not affected by the site location. The ANN works reliably for both mountain and plain locations, from north to south of Italy.

3.2.2 Smart grids

Smart grids are a new generation of electric energy distribution that acts on the base of information about the behavior of all participants thus involving balancing between energy *consumers* and *producers* in a liberalized market. This is obtained by shaping a peer to peer network between consumers and producers of electricity in a short-term approach [17] for stipulating contracts between them. This allows multiple parties to participate in the energy negotiating and exchanging market, shaping a whole new scenario that features more actors compared to the traditional (i.e. centralized) approach. In particular, the novelty lies not only in the market opening to other traditional energy suppliers (mostly producing energy using fossil fuels, hydro power, nuclear or others), but also to smaller competitors (likely to be private domestic environments) that have

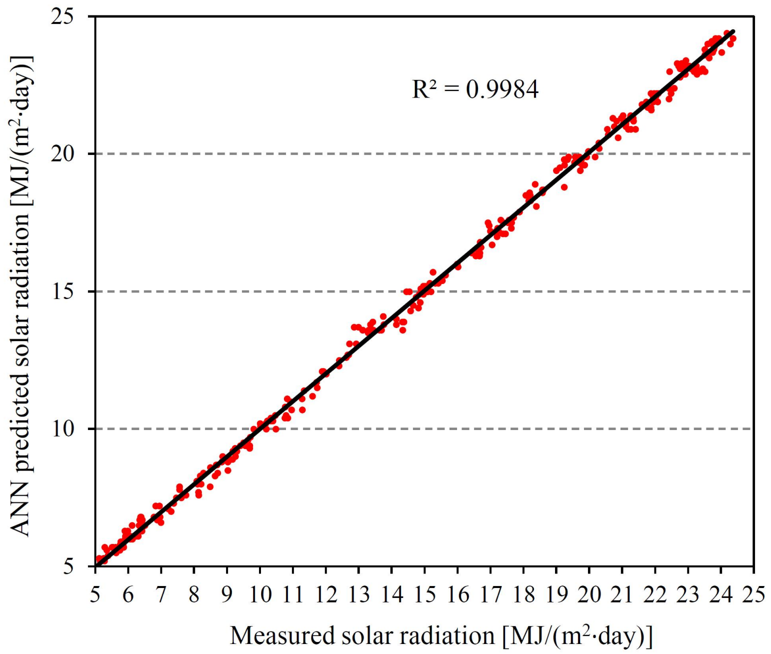


Figure 3.10: *Correlation between the measured and predicted values for the testing dataset.*

solar panels or wind turbines installed in their property, making them able to contract for incoming as well as outgoing electric energy. This scenario is leading to open challenges, and balancing is likely to be the most critical one: the total amount of energy supplied by producing peers should be ideally equal to the consumers' demand. However, in traditional approaches, the percentage of lost energy in distribution can reach important values. Failing to satisfy the balancing between demand and supply capacity will lead to (1) more expensive contract prices, (2) resource waste, and (3) environment pollution problems. In a strongly liberalized market with multiple sellers, the element of competitiveness is straightforward. However, producing more energy than it is needed is not convenient to anyone: hence, the concept of collaboration between producing units that can not adapt their production arbitrarily (for instance, intentionally limiting the production of a solar panel is unthinkable), and those who can indeed decide a production threshold by using the appropriate amount of traditional sources. The latter ones have to adapt their production according to the supposed photovoltaic or wind penetration in the market for instance. Failing to comply this aspect will lead to more expensive and to power for the grid likely to be unused.

In this scenario, the prediction can play an important role in the optimization of energy exchanges within smart grids. On the one hand, a good prediction of the future energy consumptions of the consumers is necessary to calculate the demand. On the other hand, a good weather forecast is necessary to provide important information of how much renewable penetration will be expected by the grid. Producers have to retrieve data from external sources and be aware of the design and installation aspects of her photovoltaic panels and/or wind turbines. This section presents a multi-agent system used for simulating new paradigms for collaboration and adaptation between these different actors: in particular, agents acting on behalf of producers and/or consumers could share the same goals, thus providing an effective simulation environment where agents representing different kinds of consumers and producers need to perform the tasks that this scenario requires. The short-term approach for stipulating contracts will force this calculation and data retrieval to be executed several times per day: this is complaint with the unpredictability of wind for wind turbines and solar radiation for photovoltaic panels. while meteorological stations can forecast temperature with several days in advance with acceptable errors, solar radiation and wind strength have to be analyzed with considerably less advance [210]. The short-term approach not only provides a higher granularity for this calculations, but it is also complaint to the actual possibilities of nowadays meteorological stations. For instance, a better accuracy of the wind forecast increases as narrowing down the forecast time. The proposed solution was presented in [42, 43], and it is able to deal with the following aspects of the proposed scenario:

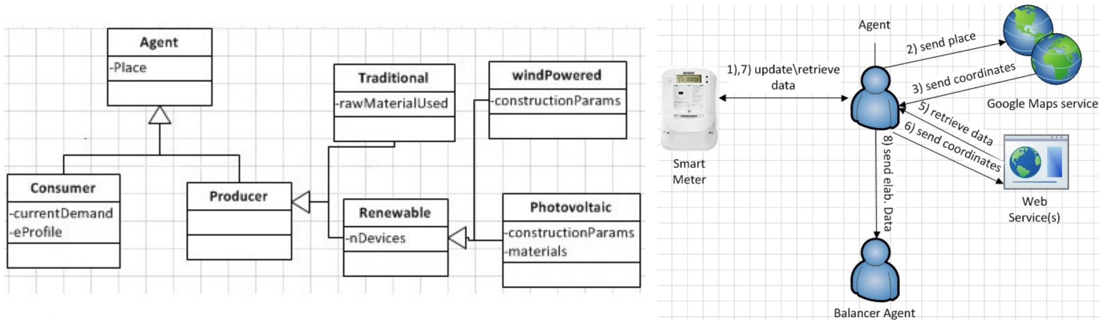
- *Awareness* of intrinsic characteristics of every single agent inside the system (smart meter communication and technical specifications of photovoltaic panels or wind turbines installed;

- *Collaboration* efforts for avoiding surplus of energy production (obtaining with the balancing with traditional suppliers and renewable penetration);

- *Adaptation* to the changing scenario (weather variations over time, location, and so on) and accurate planning of data retrieval for subsequent forecasting.

Autonomous agents approach

In the presented scenario, a multi-agent systems can be a common choice for modeling situations where different kinds of actors (producers and consumers) have different goals of their respective counterpart in the real environment [118]. Therefore, an agent represents a peer inside an electricity distribution net in which producers and consumers are the main components of the market. Previous work about smart grids and the energy distribution problems have been commonly addressed with agents [94]. Most of the previously works dealt with the negotiation step [44, 106] and for load balancing aspects [182]. More than taking into account the possible links overload, the proposed solution tries to establish a collaboration between different kinds of sellers for avoiding useless over production of electricity by balancing the correct penetration of photovoltaic and wind energy in the electricity market. Also, before any kind of attempt for balancing as well as negotiation, a correct prediction of the amount of electricity that a producer can provide is fundamental. Contacting already existing and appropriate web services provides the correct data to make good predictions. Indeed, web services and service oriented architecture (SOA) are not new in the field of smart grids or, more general smart computing. Warmer et al. [218] proposed an example on how to use and create web services to share common data in smart house environments, the energy exchanging problem *per se* can be seen as a SOA in which consumers and producers of electricity are registered and able to be exploited in a distributed architecture [157]. The proposed solution uses existing web services/sources of information, the retrieved data is then processed to correctly estimate the renewable energy penetration for the grid.



(a) (b)

Figure 3.11: *(a) Agent class diagram and (b) interaction scheme.*

Method

The most important actors in the presented scenario are the consumers and producers of electric energy. The formers present an energy demand that refers to the upcoming negotiating interval: according to the short-term approach for energy contract stipulation, in a deregulated market in which peers are connected to each others in the smart grid, there is no longer space for long lasting contracts. This will force buyers of electricity to contact the producers in an hourly basis. Producers that exploit devices producing renewable energy have to follow a similar, yet specular procedure: still referring to a particular time interval, each of them have to provide to the grid an estimation of the amount of energy that they are capable to produce according to the combination of internal and external factors. Internal factors are represented by the characterization of their wind/solar powered devices, while external factors are weather-related data like solar radiation, wind strength and direction. Figure 3.11a shows a class diagram with the created agents, summarizing the agent characterization so far introduced.

The *adaptive collaboration* strategy involves the formulation in Eq. 3.8:

$$\sum_{i=1}^{N_c} D_i(t+1) - \sum_{j=1}^{N_p} P_j(t+1) = E_D(t+1) \tag{3.8}$$

In Eq. 3.8, N_c (N_p) is the total number of consumers (producers), while D_i (P_j) is the demand (predicted availability) of the single i^{th} consumer (j^{th} producer). Subtracting the first two terms, the amount of production that remains to produce by traditional suppliers can be estimated. E_D is therefore the total energy demand not satisfied by renewable penetration: this information can be further used for setting in advance the production thresholds for traditional producers [44]. Every calculation shown in that equation is done every time interval according to the chosen granularity, therefore Eq. 3.8 is calculated at a generic time interval t, being $t + 1$ the following (therefore future) time interval for the forecast.

Another schematic for agents, services and entities involved in the model and their typical interaction scheme can be found in Figure 3.11b. In this figure, the generic agent is directly connected to its smart meter for retrieving information about incoming/outgoing electric energy, pricing and tariffs, present and past consumption measures. The versatility and easiness of interfacing a smart meter to software agents [45], adds the

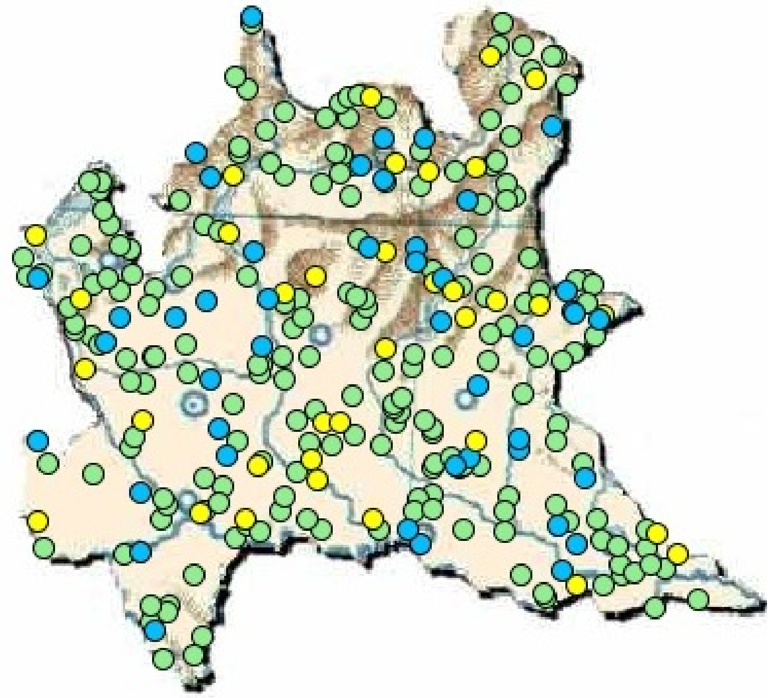


Figure 3.12: *Peer topology for the examined scenario: consumers are green, wind powered producers are blue, in yellow the photovoltaic suppliers.*

possibility to define other constants to be used for the proposed model: thinking about saving in the smart meter the construction and settings parameter of a generic renewable producer wind (or solar) device. Most of the existing web services for weather forecasting require an input expressed in absolute geographical coordinates (latitude, longitude): a generic agent, before contacting those web services, has to know those coordinates in advance. In case this agent has a position expressed in location (place) instead of coordinates, the conversion is used via an HTTP request to the Google Maps service[2]. This service will respond via a Keyhole Markup Language (KML) formatted response, from which the agent extracts the requested coordinates. The agent now can easily request the web service interaction, as explained following in detail. The elaborated data is then sent to a balancer agent, ables to calculate the Eq. 3.8, to then suggests the E_D value to the remaining traditional suppliers.

The proposed scenario involves 200 consumers and 80 small-scale renewable energy producers, with the latter ones able to produce energy either via photovoltaic or wind turbines. An investigation of a real working day (April 1, 2005) in all the 12 provinces of an northwestern Italian region is presented. A node topology in the chosen territory is shown in Figure 3.12. Time granularity for the examined day is 3 hours, starting from 00:00.

As previously mentioned, different kinds of agents need different kinds of *services*: in particular the consumer is defined by an energy profiling that provides the system with useful information about its future consumption. In this simulation test bed, the energy profiling for the demand forecast follows a trend that resembles the national

[2]http://maps.google.com/

Table 3.13: *Investigated web services.*

Service name	Service provided	Intervals	Input
Wunderground	Wind speed and direction; Temperature	Hourly	Location
HC3vX	Solar radiation	Hourly	Location or coordinates
NWX	Wind speed and direction; Temperature	Six hours	Coordinates

average, according with TERNA, the biggest Italian transmission system operator. This trend is then randomly adjusted to provide different needs for different consumers: in a real scenario those random parameters for trend changes are predicted thanks to energy profiles that refers to particular households [55]. However, external temperature is one of the important factor when dealing with demand forecast [74].

Table 3.13 shows the existing web services investigated for our case scenario (wunderground[3], HC3vX[4] and NWX[5]). Web services differ from type of input, services provided and granularity of data. All of them present some limitations in absence of registration (fees apply). Due to the scarce granularity of the NWX web service, only the data provided by the first two web services were used.

As far as the producers are concerned, they will need accurate forecasting of wind (in case of wind turbine owner) or solar radiation (in case of photovoltaic unit). In a hypothetical applied scenario, the external factors (obtained via web services) and the constructional characterization of producers' renewable devices can be soft coded in the smart meter or in a middle–ware device situated between the agent platform and the consumption reader. It does not really matter where these parameters are stored, provided that this information is known in advance and is complaint to any changes of the devices installation.

1) Wind power: The production of electrical power through wind turbines depends on the interaction between the rotor blades and wind speed; in particular, the electric power generated by a wind turbine P_e, expressed in watt, can be determined through the Eq. 3.9:

$$P_e = \eta_e \times \eta_m \times C_p \times \frac{1}{2} \times \rho \times A \times Ws^3 \tag{3.9}$$

where η_e is the efficiency of the electric generator, η_m represents the efficiency of mechanical components, C_p is the power coefficient, A the area swept by the rotor expressed in m^2, ρ is the air density in kg/m^2, and Ws is the wind speed expressed in m/s. $C_{p,max}$ is commonly called *Betz* limit, it is generally set equal to 0.59, and expresses the following basic idea: "The maximum power that can be theoretically extracted, considering an air current and an ideal wind turbine, may not exceed the 59% of the available power of the incident wind". In practice, there are three effects able to decrease the maximum power coefficient:

[3] www.wunderground.com/history/
[4] www.soda-is.com
[5] www.navlost.eu

- Rotation of wake behind the rotor;
- Finite number of blades;
- Aerodynamic resistance.

With modern turbines, however, reaching a $C_p \cong 0.5$ value represents a good approximation for the theoretical Betz limit. In particular, in this model, the C_p of different producers are randomly selected with values ranging from 0.3 to 0.5 in order to simulate the characteristics of the various wind turbines present in the market. Determined C_p does not express only the fraction of power that the wind transmits to the rotor, but, for sake of simplicity, also includes other factors that influence the operation of the wind turbine, e.g., the roughness of the ground or the installation height. The air density depends on the temperature and the altitude of the place of installation, so it is easily calculable. The performance parameters are constructional features of the turbine.

2) Solar power: Eq. 3.10 shows how to calculate the producible energy by the whole photovoltaic system, based on the data of average radiance in the considered time slot (3 hours). This equation refers to a single photovoltaic module, therefore in multi modules environments, it is necessary to sum the single contributions of the installed modules.

$$E_g = P_0 \times G \times K \times \eta_{PV} \times \eta_{INV} \tag{3.10}$$

Where P_o is the module peak power expressed in Wp, G the solar radiation in W/m^2, K represents the shading factor, η_{PV} is the photovoltaic generator efficiency in the range [0.70–0.86] according with the temperature, and η_{INV} is the inverter efficiency in the range [0.88–0.94]. K is a parameter smaller than 1 that considers the power reduction for aging, panels' inclination, shadowing and foliage due to nearby trees. In this simulation, K is chosen in a probabilistic way varying from 0.8 to 0.95, in order to better represent the analyzed scenario. Finally, P_o is a characteristic parameter of the modules. The variation of the electricity production depends on the weather conditions: solar radiation is one of the parameter retrieved from the web services.

Consumers demand trend

The consumption of electricity is affected by different factors such as workability (calendar effect), climatic variables, seasonality and economic activity. To predict the energy consumption in a short term, most of these factors can be considered known via user profiling. To validate the proposed model using a simulation, they are treated as random variables in a plausible range according to typical trends. This information is then integrated with our model for bounding consumption and temperature, creating a consumer model that takes into account both the energy profiling and the temperature impact. The influence of external temperature is crucial because of the sudden changes in temperature that will cause significant changes in consumption in the short term, while the other profile related factors are supposed to follow a more static trend. The relationship between temperature and consumption is non-linear and dynamic. In this model, the temperature versus consumption function is described by the followings:

- The relationship is a combination of linear functions, with knots (key values) in 8°C, 18°C, 22°C and 32°C. There is an interval of temperatures between 18°C and

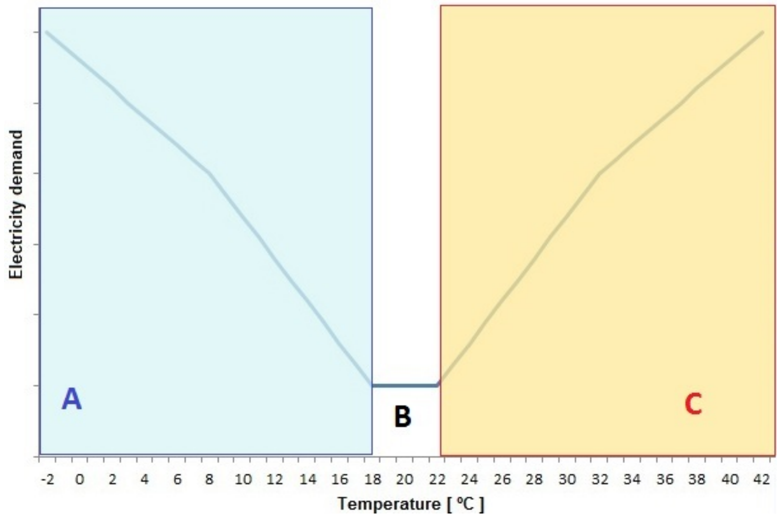


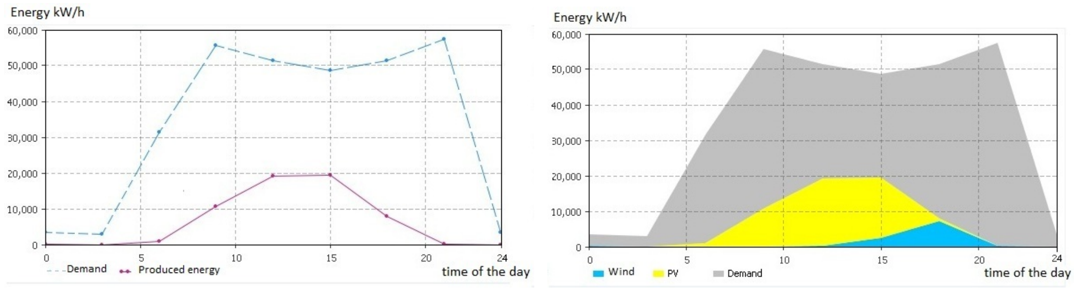
Figure 3.13: *Temperature vs consumption: (A) cold zone, (B) no effect on consumption, and (C) heat zone.*

22°C, where the temperature does not affect consumption; the temperatures below 20°C shape the cold zone and temperatures above 24°C the heat zone. These areas are shown in Figure 3.13 and they cause maximum consumption values;

- The cold zone can be approximated via two linear decreasing functions. The first one presents temperature values between 8°C and 18°C. To temperatures below 8°C, the slope of the function decreases in absolute value, still maintaining the linear trend. Further attempts to cool down the environment are supposed to be inefficient;

- In the heat zone there is a similar relationship. There is a linear response function between 22°C and 32°C; above 32°C the slope presents a smaller value, so that the marginal effect of temperatures above 32°C is lower than temperatures between 22°C and 32°C. As in the cold zone, this last slope changing is forced by the used electric heating device.

The simulation test bed was implemented using the AnyLogic software agent platform. The purpose of this simulation is to understand how a multitude of consumer agents can generate different demand values to be then satisfied by the producers. Producers can only satisfy a fraction of the total demand; the remaining demand is covered by traditional suppliers. The *adaptation* lies in all the aspect that are able to vary the consumers' demand according to their habits, but also how different devices react to the changing weather parameters previously obtained via the web service integration. The architecture should therefore prove to be *aware* of the context (energy profiling, weather, location, and so on). The *collaboration* aspect refers to the different kinds of producers. Assuming that renewable energy producers are able to sell their surplus energy directly to other consumers at a lower price compared to the bigger traditional energy companies. Figure 3.14 shows the results of the simulation. The demand follows the predicted trend, having peaks during certain hours of the day. Figure 3.14a shows the sum between the wind and solar penetration. Figure 3.14b shows the differ-



(a) (b)

Figure 3.14: *Results of the simulation for the examined scenario.*

ence between these two plots facilitating the reading of the two different contributions. During the central hours of the day, the renewable energy penetration reach its maximum values: that is a straightforward consequence of the presence of hour of sunlight. As for the wind speed, the peak is reached between 3PM and 6PM. In Figure 3.14b, the gray area represents the E_D value of Eq. 3.8, influencing the production requested for the traditional producers. The proposed model, therefore, makes the assumption that the totality of renewable energy produced will be sold during the appropriate time interval. This is justified by thinking that if a small producer does not manage to sell its production to a neighboring consumer, the energy will nonetheless be reintroduced in the grid. From a negotiating point of view, this means that this kind of energy without a direct buyer will be obtained by other traditional suppliers or transmission services.

3.2.3 Discussion

This section faces the problem of estimating the climatological data. At the beginning, the prediction of the average monthly global solar radiation over the Italian territory through ANNs was presented. This study contributes to the shared intent of setting up a common set of methods and approaches to estimate the solar radiation over the most promising geographical regions, e.g. the sun belt area. A multi-layer perceptron feed-forward network with one hidden layer was used. A wide and geographically distributed number of sites located in 45 different Italian cities composed the multi-location scenario, while for each of them, the input dataset was made of 14 geographical/climatological parameters. Evidences showed that by using all the available input, the ANN best configuration leads to a MAPE of about 2.66%. Furthermore, a relevance analysis of the input data generated several ANN configurations and stressed the most critic parameters. For Italy, a smaller number of parameters represents a good choice to further increase the ANN performances. Using 7 inputs only–in order of relevance, top of atmosphere radiation, day duration, rainy days, altitude, rainfall, time period and latitude–the ANN outperforms all the other scenarios achieving an overall MAPE equal to 1.67%.

Furthermore, a multi-agent system was presented. In this system, agents acted on behalf of energy producers and consumers, where the amount of energy exchanged was influenced by several factors related to the energy market context. After providing an exhausting list and explanation of all the factors involved, a web service integration was a useful source of information for providing input for the weather prediction. Therefore, a small set of existing web services were investigated and thus used for simulation purposes. By combining these external factors into different models which predict the future production of energy via renewable electricity, a realistic model of a North-Western region of Italy was constructed. A simulation showed that agents are able to adapt to different weather conditions and consumers' habits. The model presented an initial step for energy negotiating that is all about balancing between photovoltaic and wind penetration over the totality of the energy demand for the considered area.

3.3 Social interactions

This section turns to a class of complexity related problems in which the actors involved are groups of human beings. Groups of people provide, in fact, one of the most authentic prototypes of complexity. They also constitute a source of inspiration for raising number of new issues, stimulating in turn fundamental research in the area of complex systems. Here the new element that comes into play is the presence of such concepts as strategy, imitation, anticipation, risk assessment, information, history, and some other concepts from social science. The expectation would be that thanks to the rationality underlying these elements, the variability and unpredictability should be considerably reduced [152]. The variability inherent in the dynamics of the individuals is eventually controlled and channeled to yield some emergent patterns of groups behavior.

This section analyzes the users behaviors within social media, i.e., applications that engage users, allowing to create and exchange information inside different online communities. This new way of exchanging information via web created a massive amount of content generated by users: real human beings with real experience and expertise, communicating and sharing knowledge. The enormous amount of knowledge exchanged every day has intrigued the attention of many researchers. Probably, the most famous example of social knowledge repository is Wikipedia[6]. The slogan of Wikipedia website is “Wikipedia is the free encyclopedia that anyone can edit”. Some studies, among which [115, 192], provided empirical support for the potential of the crowdsourcing, demonstrating how the quality of the Wikipedia’s content is as good as the traditional encyclopedias written by experts. With the evolution of the so-called *Web 2.0*, other social media platforms have seen immense popularity. One of these platforms are the question and answer (Q&A) websites. These sites are platforms where users can ask, answer, and rate the created content, creating communities of users who interact to solve problems. Q&A communities are designed on mass collaboration and user participation. They are communities formed on the idea that everyone knows something, and through collaborative knowledge production users provide answers and create an archive of problems and solutions [76]. As well as other social media, the success of Q&A communities is based in part on the “wisdom of crowds” [202]. The wisdom of crowds effect allows groups to reach better decisions by leveraging their ability to aggregate diverse information, as compared to individual or small numbers of experts. In crowdsourcing applications, participants rely on the wisdom of crowds effect to reach better conclusions.

Studying a large repository of data from a popular network of Q&A communities, Stack Exchange, this section has a two fold purpose. Firstly, it presents the relationship between group’s performance and its diversity. This analysis aims to be a preliminary framework to evaluate success and diversity in Q&A communities. This framework can be the basis to develop some models able to predict the success of determined communities based on the diversity of their users. The second purpose of the section is to identify factors affecting which answers users vote for, as well as which answers askers choose to accept. This section provides evidence that users, likely motivated by a desire to limit the effort they expend on discovering and evaluating answers, select answers based on simple heuristics rather than their own judgment of answer quality,

[6] https://www.wikipedia.org/

especially as the number of available answers increases. These heuristics appear to affect the collective performance of a crowdsourcing system, providing evidence of crowd wisdom fragility due to an individual's information overload.

3.3.1 Stack Exchange

Stack Exchange[7] (SE) is a network of question answering communities that are created and run by experts and enthusiasts who are passionate about diverse topics. This network started in 2008 with the launch of their first Q&A site for computer programming questions: Stack Overflow. Over time, SE has added more and more sites based on the same model, but on different topics. Now, SE consists of more than one hundred communities covering extremely diverse topics:

- 49 *Technical* sites, for topics related to computer technologies. Some examples of technical sites are Programming, Server Faults, Information Security, Apple, Android and Ubuntu;
- 33 *Culture and recreation* sites, where themes addressed include English Language Learners, Bicycles, Videogamers Platforms, and Anime & Manga;
- 17 *Life and Arts* sites, for topics related to the everyday life, such as Cooking, Photography, DIYers, and Movies & TV;
- 16 *Science* sites, for topics related to science, e.g., Mathematics, Statistics, Biology, and Philosophy;
- 4 *Business* sites, which include questions and answers about Bitcoin, Project Management, and Finance.

In addition, for each community, there is a *meta* community where users discuss the workings and policies of the associated community, rather than discussing the topic itself. For example, in Meta Stack Overflow, users discuss the workings and policies of Stack Overflow rather than discussing computer programming itself.

All of these Q&A web sites are focused on question answering. As the SE website clarifies: the purpose of these boards is to provide good answers for each question, not to create discussion. Posts that are overly subjective, argumentative, or that are likely to generate discussion rather than answers, are removed from the website. This encourages users to post informative answers. Of course, not all questions are answered, but if a question is answered, it can receive multiple answers from multiple users. The *asker* (the person who asked the question) can mark one and only one answer as an *accepted answer*. Acceptance generally signifies that the asker is satisfied with the usefulness and clarity of the information provided by the answer. Regardless of acceptance, others can *vote* an answer up (or down) if they think that it provides good (or bad) information. Users can vote even if they did not ask or answer the question. The objective is that over time, the crowd will vote up the better answers. By upvoting the better answers, a community collectively curates the information in the answers to be useful for both askers and future users who are interested in the same topic. The difference between the up and down votes is the *score* of the answer. Answers with higher scores rise to the top of the list of answers to the question, so that they are easier to find. Answers with

[7] http://stackexchange.com/

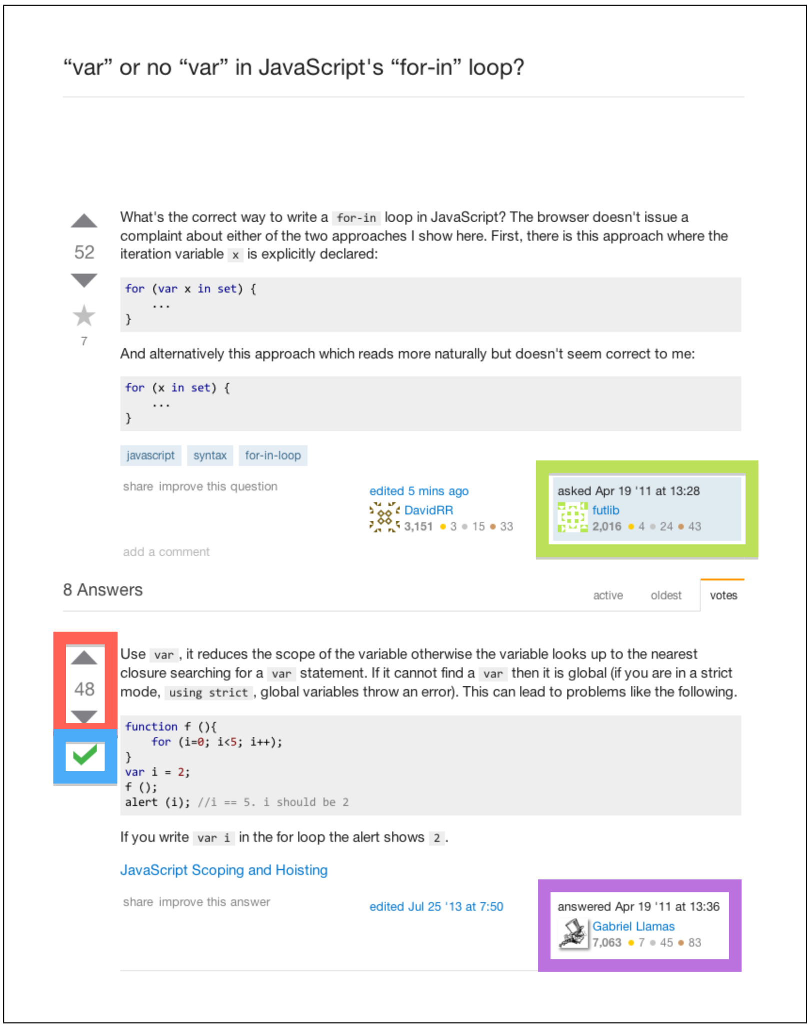
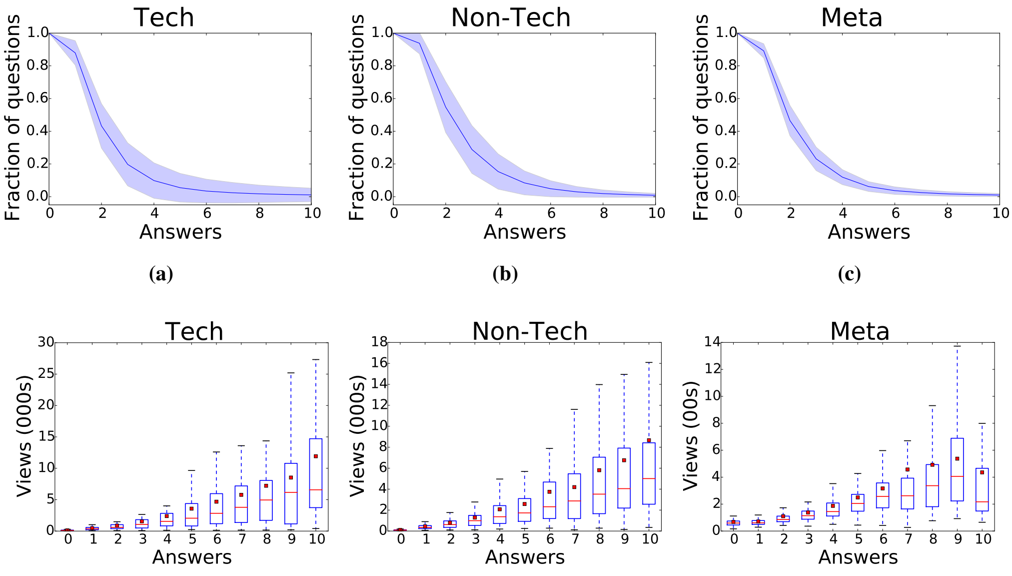


Figure 3.15: *A screenshot of a Stack Exchange web page, showing a question (at top) and answers listed below in default order. The score next to the answer (red box), is defined by upvotes minus downvotes, and the green check mark (blue box) denotes that the answer was accepted by the asker. The times the question was asked appears in the green box, and the time when answer was provided, as well as the answerer's reputation in the purple box.*

the same score are shown in random order, which may be different for different users viewing the same the question. The answers on the top of the page can be seen as a collectively curated collection of information, useful not only for the askers but also for future users interested in the same topic. In fact, questions and answers are saved on the site and often prominently ranked via search engines, more and more often therefore, people who may not even be a priori aware of the site can be directed to the information there [13]. Figure 3.15 shows an example thread, with a question, answers, accepted answer, and score for both answers and questions.

Data

SE makes available the anonymized data of all its communities, including questions, answers, and votes. The examined data consist in all user-contributed content from



(a) (b) (c)

(d) (e) (f)

Figure 3.16: *(Top row) Complementary cumulative distribution of the number of answers posted in reply to a question on (a) technical, (b) non-technical, and (c) meta sites. Shaded areas correspond to the standard deviation in the distributions. (Bottom row) Number of views per question as a function of the number of answers on (d) technical, (e) non-technical, and (f) meta sites. Boxes indicate 50% confidence intervals, with a red line to indicate the median view count, and a red dot to represent the mean viewcount.*

2009 until September 2014[8]. The data contain information about more than 27 millions posts (questions and answers) and 70 million votes. In particular, the data used come from 250 sites (48 technical, 76 non-technical, and 126 meta), including information related to the posts: the ID of the post, creation date, type of post (question or answer), ID of the relative question (in case of answers), the ID of the eventually accepted answer (in case of questions), and the content of the post; and related to the history of the votes made on each single post: the type of vote (up, down votes, or acceptance), the ID of the related post, and the time of assignment. In addition, other informations related to the users, such as the ID of the user, creation date (date of the sign up), and reputation were considered. Particularly, the reputation of the users at the moment they asked or answered a question, considering the rules of Stack was also calculated.

Each question on these sites receive almost three answers, on average. The "Programming Puzzles & Code Golf" site, where programming puzzle enthusiasts post questions, has the highest number of average answers per question at 8, while the "Magento" site has an average of only 1. About 10% of the questions are unanswered, with 42% receiving only one answer. Figure 3.16 shows the complementary cumulative distribution of the number of answers posted for each question on technical, non-technical, and meta sites. Stack Overflow has more than 13 million answers, almost 10 million of them with votes. The other sites have an average of 11k answers per site, although this varies significantly. Technical sites have on average twice as many answers as non-technical sites, and an order of magnitude more than meta sites, which are not broad

[8]https://archive.org/details/stackexchange

in appeal. Askers accepte an answer 59% of the time in technical sites, and 59% in non-technical sites, but, curiously, only 37% of the questions in meta sites are similarly accepted. Answers with votes consist of 86% (standard deviation 7%) of the total, but this too varies across site types. 78% of the answers on technical sites have votes, versus 86% in non-technical and 88% in meta sites (all differences are statistically significant with $p < 10^{-3}$ using t-tests). Therefore, the sites vary in terms of the number posted answers, but in general the percentage of voted answers is still high. From this point of view, users of the SE sites appear to be fairly involved in the social dynamics, understanding how the voting process is an important aspect for the success of the Q&A communities.

The median time to obtain the first answer is 166 minutes, and the eventually accepted answer, 268 minutes (this includes when the first answer was the accepted answer as well). There is no statistical difference between the median response time on technical, non-technical, and meta sites. As the site matures, the questions become more complex, which attracts the attention of users who may focus on different facets of the problem, posting multiple good answers for the same question.

3.3.2 Diversity and success in Q&A communities

The success of a Q&A community represents its capacity to provide good answers to questions. For this reason, the success of a community can be defined as the number of questions which were answered and how good these answers are on average. The ability of a community to provide good answers is evaluated according to the assessment of its users. In fact, the judgments of the users within a community can be considered reliable with a pretty high level of confidence [32]. Therefore, it is possible to hypothesize that users who did not respond in time or were unable to answer a question themselves are nevertheless able to recognize a correct solution if it is presented to them. According to Laughlin and Ellis [123], this is one of the characteristics that a group must have to perform better than individuals. In this study, the best answer for a question is that answer which received the highest score. Equations 3.11 and 3.12 show the measures used to calculate the success of a Q&A site.

$$\text{Answered questions} = \frac{AQ}{PQ} \tag{3.11}$$

$$\text{Mean highest score} = \frac{\sum_{q=1}^{AQ} max(\text{Score}_{aq})}{AQ} \tag{3.12}$$

where PQ is the number of posted questions, AQ is the number of questions answered, and Score_{aq} is the score of answer a posted to answer question q. In this way, the wisdom of crowds is used to obtain a collaborative assessment of the answers' quality. These measures can in a simple way capture user engagement (the more the users vote the more they are engaged), askers' satisfaction, and quality of the answers (the higher the average answer score, the higher the quality of the answers and the higher the askers' satisfaction). The percentage of answered questions is highly positive correlated with the ratio number of answers to number of questions (Pearson correlation index [164] = .744), as well as with the ratio number of answerers to number of askers (Pearson index .513). So, the greater the number of answers respect the questions and

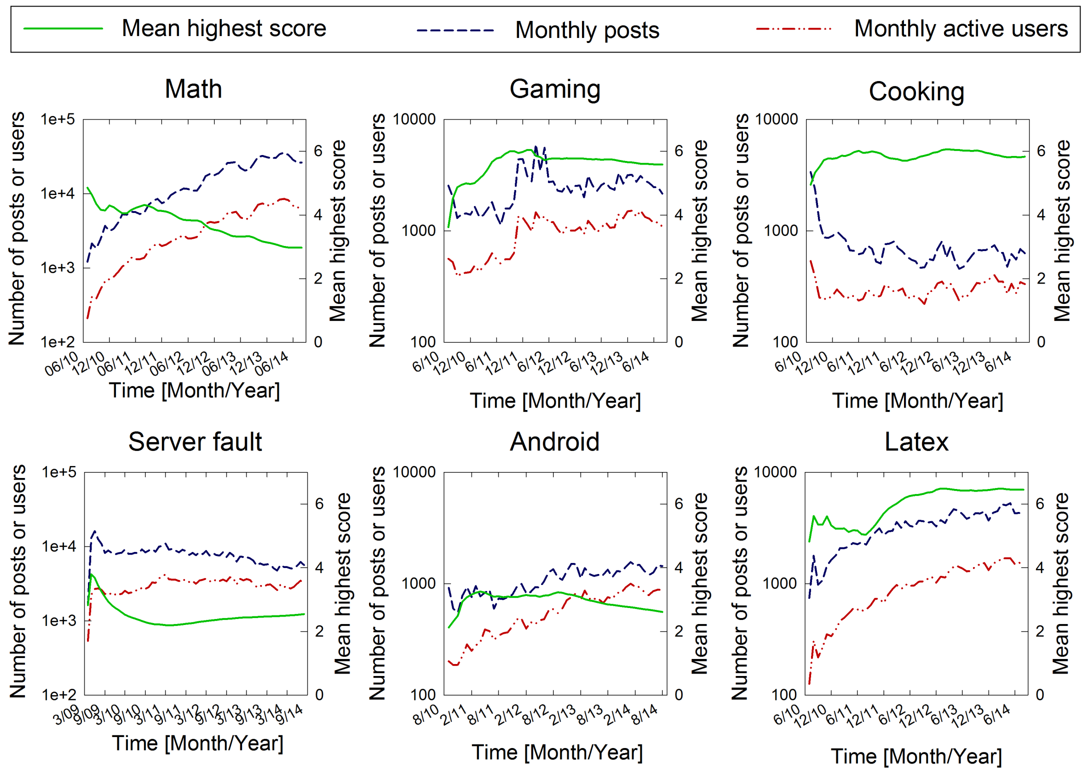


Figure 3.17: *How the mean highest score evolves across time for some example communities.*

the number of answerers respect the askers, the greater the success of the Q&A community, in terms of percentage of answered questions. On the other hand, the mean highest score is not related with the number of users or posts within the communities. In fact, the Pearson index between the mean highest score and the number of active users is -.193; and -.169 with the number of posts. That is important to be sure to consider a qualitative measure which does not depend on the number of users who can vote, but which is able to capture the assessment of the entire community. Some examples of how the success measure evolved during the time is shown in Figure 3.17. This figure shows how the qualitative measure has a trend independent of the number of active user or posts within the community.

Diversity, as success, can have different definitions in different contexts. Users in an online community can vary along many attributes, including how knowledgeable they are, how motivated they are to contribute, how much time they have to interact, and so on. This study focuses on user's tenure inside the community. A user's tenure, i.e., how long the user had been a member of the community, can be measured in different ways: as the difference between the last access date of the user and the profile creation date, or the difference between the last post creation date and the first post creation date. Since all these measures of diversity are highly correlated, and also correlated whether applied to active users or answerers, only one measure is presented for the purpose of this study. Specifically, diversity of a community is defined as the variance of the log of the difference (in days) between last access date and creation date for the answerers:

$$\text{Diversity} = \text{Variance}(\log(\text{last} - \text{creation})). \tag{3.13}$$

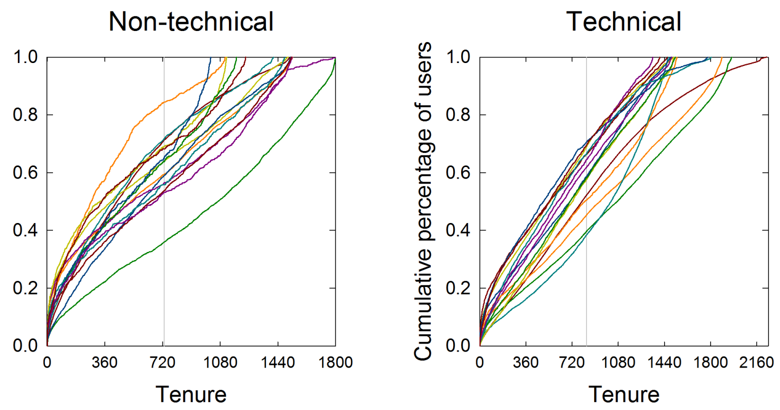


Figure 3.18: *Cumulative distribution function of the users tenure among the communities.*

The logarithm is used to reduce the spread between the last access date and the creation date. In Q&A communities, the distribution of the tenure fits the power law distribution. Figure 3.18 shows the cumulative distribution of the users tenure in technical and non-technical communities.

Veterans and novices

The success and diversity measures defined above are used to explore how much veterans, i.e., users with more experience within a community, contribute to the success of a community. This study explores if diverse mix of veterans and new users is important to the community's success, or if it is a more homogeneous community more successful. To analyze these aspects, 30 Stack Exchange communities were randomly chosen from among all of them with more than two thousand answers. The dataset used for this analysis contains information about more than 5 million users. Stack Overflow is the one with the larger number of subscribed users, counting with almost 3.5 million users. Other examined communities have on average approximately 59,000 subscribed users (with a standard deviation 64,000). Almost 26 million posts were considered, which includes both questions and answers. Stack Overflow has more than 21 million posts, the others have 140,000 posts on average with a median of 88,000. Therefore, the communities considered have a substantial difference in term of posts and active users, as shown in Figure 3.19.

Users' tenure distribution tends to follow a power law distribution. This means that users with a low tenure are frequent, users with a high tenure occur only a few times, and the distribution has a heavy tail. *Veterans* are defined as answerers with a normalized mean tenure greater than two thirds, and *novices* as answerers with a tenure smaller than three months. Veterans are on average 11% of the active users, and this average is valid for technical and non-technical communities. However, the percentage of veterans within technical communities has a standardad devition of 6%, meanwhile in non-technical communities the standard devition is only 3%. This is practically due to a particular community (*programmers*) in which 30% of active users are veterans. Novices instead represent a considerable slice of the active users in Q&A communities, on average 22% of active users are novices. There is a difference statistically signif-

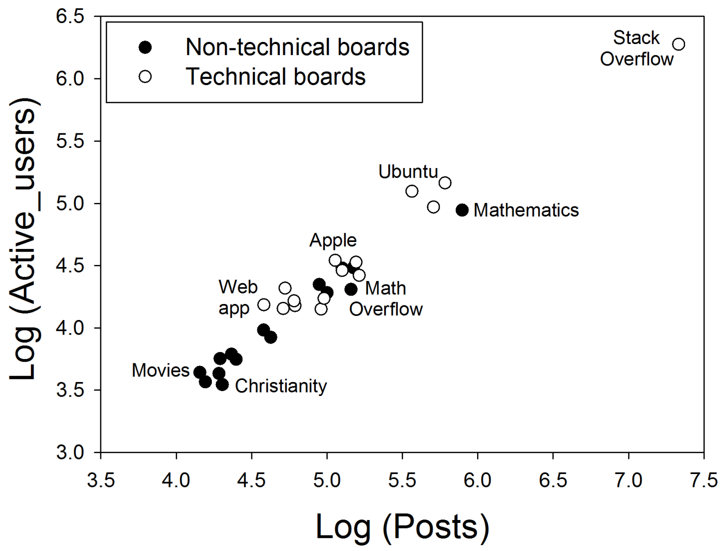


Figure 3.19: *Number of posts and active users inside the communities.*

icant between the percentage of novices in technical and non-technical communities. In technical communities 17% of the active users are novices, in contrast to 27% in non-technical communities. Curiously, veterans and novices have the same tendency to ask questions. Indeed, 9% of posts of veterans as well of novices are questions. This reflects the idea that generally users who tend to post answers, tend to ask just few questions, independently from their experience inside the group. In other words, commonly users show their tendency to be more answerers than askers already from the beginning of their entrance inside the group. And this is valid for technical and non-technical communities. As reminder, novices and veterans are identified between answerers, i.e., between users who posted at least one answer. Beyond the percentage of questions, surprisingly novices tend to post more answers than veterans. 57% of posts posted by novices are answers, while just 26% of posts posted by veterans are answers. At the same time, 34% of the posts posted by novices are comments, and 65% of posts posted by veterans are comments. This reflect a very common group dynamic: novices are new in the group, excited to be part of that, and want to participate in the community life. On the other hand, veterans are established members that comfortably participate in the community, trying to keep it running. It is important to notice that the number of comments posted can be affected by the fact that it is necessary to have a certain amount of reputation to post comments. This amount is not relatively high, but on average accross the communities just one third of novices can post comments.

The purpose of this part of the section is to analyze the performance of veterans and novices according with success measures presented previously, in order to understand the importance of these users within Q&A communities. The measure showed in Equation 3.11 depends on the number of answers posted. Obviously, veterans have had more time to post answers, so it is easy to foresee that they posted more answers than novices. Numbers in fact express exactly this: in technical communities, veterans provide on average 7 times the number of answers that novices provide, with a standard deviation of 6. Hence, there are some communities where this difference is more pronounced, e.g., Latex community where veterans posted 23 times the number of answers of the novices; and other communities where this difference is less pronounced, e.g., Android and Apple communities where veterans posted 2 times the number of answers of novices. On the other hand, in non-technical communities, veterans provide on average 5 times the number of answers respect the novices, with a standard deviation of 4. Also in this case there is a difference between NetMathOverflow community where veterans posted 15 times respect novices, and Gaming community where veterans posted 2 times respect the novices. Then, veterans posted more answers respect novices, but how many answers do they post per each time unit? Analyzing how many answers veterans posted in each month of 2013, that amount was compared with the number of answers that novices posted in same months. For each month, it was evaluated who was a novice (answerers who had joined the community for less than three months) and who was already a veteran (answerers with a tenure inside the community higher than two third of the community age at that time). The results in Figure 3.20 show how even though veterans post more answers in the long term, in the short period the situation can be different. The figure shows the number of novices and veterans who posted at least one answer per each month, and the sum of the answers posted. In Latex community for example, veterans posted always more answers than novices, posting on average 5 times the number of answers respect novices. But there are also other communities where this difference is more slight, e.g., Stack Overflow and Math. And interestingly, there are also some communities where novices posted more answers than veterans, e.g., Android and English. In these cases, also overall difference between answers posted by veterans and novices is less noticeable. In this monthly analysis, novices are those users whose tenure was less than three months at that precise time. For example, if the creation date of a users is January 1 and her last access date is June 1 of the same year, in this monthly analysis, her posts in February are considered made by a novice. The number of veterans is positive correlated with the percentage of answered questions, with a Pearson index equal to .487* and .602*[9] respectively for technical and non-technical communities. In non-technical communities also the number of novices is related with the percentage of answered questions (Pearson index .632*). In technical communities instead, this correlation is negative. In numbers, the correlation between percentage of novices and answered questions in technical communities is -.742**. It means that in technical communities, not necessarily, the greater the number of novices within the community, the greater the percentage of answered questions.

Considering the proposed success measure 3.12, veterans perform extremely well. The mean quality of the answers provided by a community depends in a large part on the quality of its veterans. On average, 57% of the times, the answer with the highest score of a questions was posted by a veteran. Only 8% of times on average, it was posted by a novices. The mean highest scores of veterans has a variation respect the overall mean highest scores of 0% on average. It means that there is no discrepancy between how veterans perform on average and how the overall community perform on average. Concerning novices, their mean highest score is on average lower than 60% with respect to the mean highest score of the communities. Obviously, it is impossible to know how the answers posted by novices would be voted, if the answers of veterans were not posted. But independently, it is possible to conclude that the mean quality

[9] * p-value ≤ 0.05, ** p-value ≤ 0.005

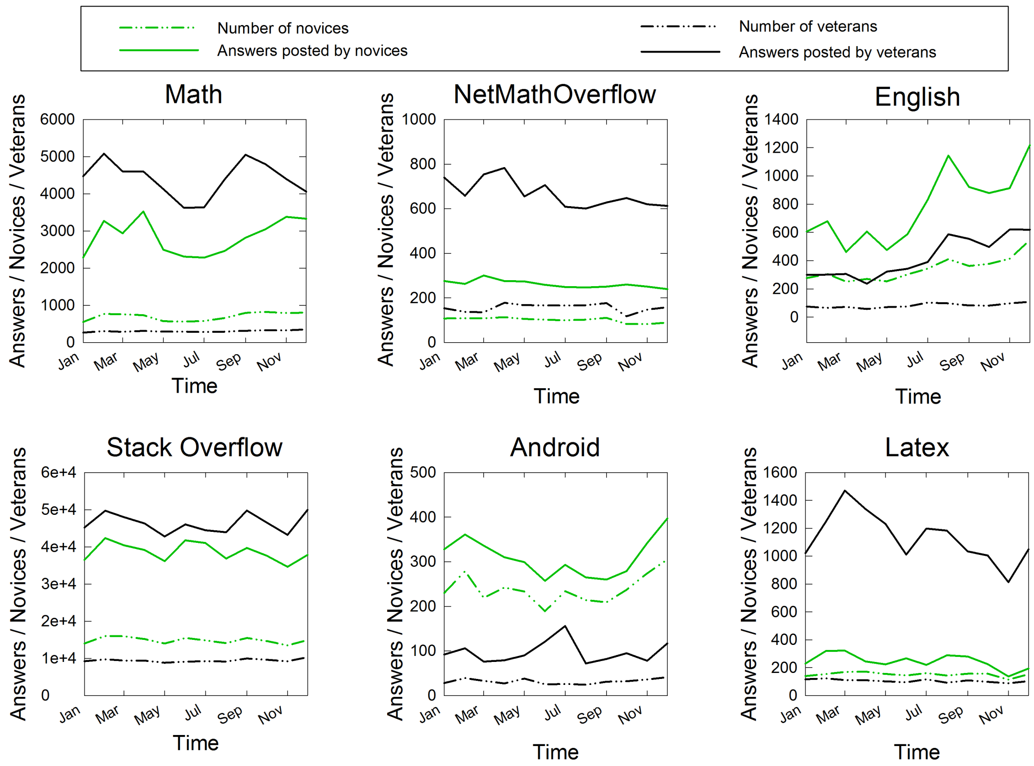


Figure 3.20: *Examples of number of answers provided by veterans and novices per each month.*

of the answers of veterans is higher than the mean quality of the novices' answers. Opposite correlations between the percentage of veterans and novices with the mean highest score of technical and non-technical communities were found. On the one hand, there is a highly positive correlation between the percentage of veterans and answer score for technical and non-technical communities (Pearson index respectively .722** and .848**). On the other hand, the number of novices is positive correlated with average answer score in non-technical communities (Pearson index .473*), and negative correlated in technical communities (Pearson index -.566*). It means that for technical and non-technical communities, the greater the percentage of veterans, the greater the mean quality of the answers. In non-technical communities, it also pretty true that the greater number of novices, the greater the mean quality of the answers. In technical communities instead, the greater the percentage of novices, the lower the mean quality of the answers.

Table 3.14 summarizes the results presented so far. A first conclusion is that veterans post a higher amount of answers in general, but considering just single periods, novices can provide the same amount of answers than veterans or even more. In addition novices post answers faster than veterans, but their quality is worse. So, both novices and veterans contribute to the overall success of Q&A communities, each in its own way. The next question to be addressed is: can be a mix of users with diverse tenure the key of the overall success of this kind of ad-groups?

	Technical		Non-technical	
	% veterans	% novices	% veterans	% novices
Answered questions	.487*	-.742*	.602*	.632*
Mean highest score	.722**	-.566*	.848**	.473*

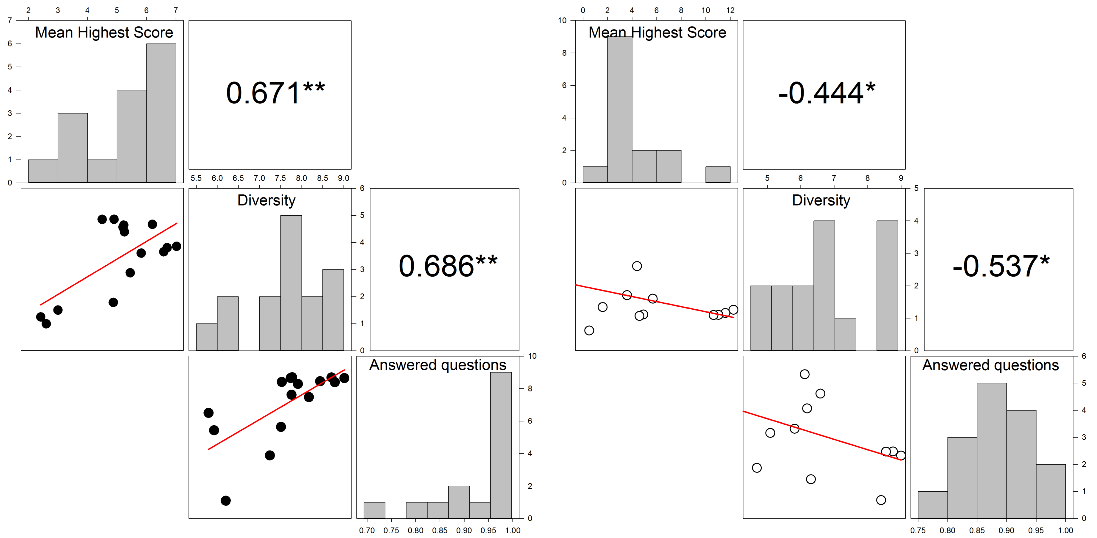
* p-value ≤ 0.05, ** p-value ≤ 0.005

Table 3.14: *Correlation between the success measures and the percentage of veterans and novices within the communities.*

Correlation between diversity and success

The Pearson index is used to understand whether Q&A communities with high variance of member tenure provide answers with higher scores on average. Considering the analyzed communities as a single group, the Pearson index between the percentage of answered questions and the diversity measure presented in Equation 3.13 is .142. Also considering the success score, the Pearson index is still lower, equal to .004. It reflects the fact that there is no correlation between the success and diversity. It means that having a mix of veterans and novices does not contribute to improve the success of a community. Consequently, technical and non-technical communities are considered separately, in order to understand if diversity can have different effects within communities which address different topics. In non-technical communities, the Pearson index between the quantitative success measure and diversity is .686**. The same index calculated between success and diversity is .671**. On the other hand, in technical communities, the Pearson index between the percentage of answered questions and the diversity measure is -.537*; and -.444* considering the mean highest score. So although diversity is not related to the success of Q&A communities in general, it does appear to be related in non-technical communities. Thus, non-technical communities composed by users with varied experience, i.e., both novices and veterans, increase its ability to obtain better answers. Figure 3.21a shows the correlation between the success and diversity measures in non-technical communities. On the other hand, the correlation indexes of technical communities revealed a negative correlation between success and diversity, Figure 3.21b. This means that within technical communities, the diversity of the users' tenure does not influence the overall success or, at least, the greater the diversity of users' tenure, the lower the success. It means that the interaction between diverse users is correlated with the success of Q&A sites, depending on the topic being addressed. In non-technical communities, groups of diverse users (novices and veterans) increase both qualitatively and quantitatively the answers; while in technical boards, diverse users negatively affect these measures of success.

Furthermore, the normalized mean tenure of users on technical communities is defined as the mean of the difference between the last access date and creation date, normalized considering the creation date of the community. For technical ones, the mean higher score and the normalized mean tenure are relatively highly correlated, with a Pearson index equal to .766**. Figure 3.22 shows that correlation, which indicates how the presence of veterans is the most important factor in the success of technical communities. In this figure, it is clear that the Programmers community leads the correlation;



(a) (b)

Figure 3.21: *Correlation between success and diversity measures in (a) non-technical and (b) technical communities.*

indeed not considering that board the correlation coefficient drops to .511. However, these two correlation coefficients are not significantly different from each other, so our conclusions can be still valid. Moreover, it is interesting to notice how in non-technical communities, the success measure and the normalized mean user tenure are not correlated at all, with a Pearson index equal to .062. This indicates that the success of non-technical communities can depend on the interaction between novices and veterans and not only on the experience of the veterans.

Veterans perform better than novices inside Q&A communities, and their continued contributions are important to their success. However, the presented results suggest that the interaction between veterans and novices is important for this success, especially for non-technical communities. Figure 3.23 shows the normalized average activity of users in their first three months inside a Q&A community, as measure of novices engagement level. At the beginning, users appear to be very active in answering questions, but already after the first two weeks their activity rapidly declines. This suggests that novices are important for the success of Q&A sites, because they are highly engaged and motivated to contribute. It is not clear why novices become less engaged so rapidly. The high engagement of novices is also demonstrated by their response time. Indeed, they provide faster answers with respect to veterans. The median time to respond of novices is just 7 minutes. On the other hand, veterans take on average 161 minutes to post an answers after that a question is posted. It means that novices are really active when they just joined the group, and they are motivated to provide answers as soon as they can. The median time to respond is not correlated with the percentage of novices or veterans, except for novices in technical communities. The Pearson index between the percentage of novices and the median time to respond on technical communities is .524*. It means that in technical communities, the time to respond is particularly influenced by novices. Technical and non-technical communities did not differ signifi-

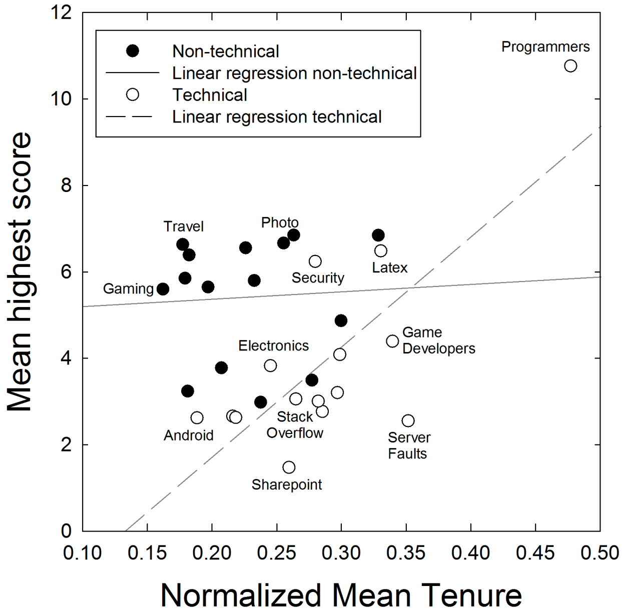


Figure 3.22: *Correlation between diversity and normalized mean tenure in non-technical and technical communities.*

cantly on this trend, but novices in technical communities struggle more to give useful answers initially. It is important for the community performance to maintain an influx of highly engaged novices, even if they quickly lose interest in the community.

3.3.3 Bounded rationality

Do crowds perform better than experts? Ever since Galton's pioneering paper [83], significant evidence has backed up this hypothesis [48,202,216]. For example, Surowiecki's popular book [202] gives many examples of how large groups outperform the vast majority of individuals at tasks, such as estimating the weight of an ox or the number of jellybeans in a jar. However, the author himself conceded that such wisdom of crowds requires independent, uncorrelated judgments. Among the recent works that have shown the limitations of crowd wisdom is one by Lorenz et al. [135], who found that social influence can undermine the performance of crowds on difficult tasks. This wisdom can be biased by individual cognitive heuristics, such as social influence [150] or rank-order [58], which creates inefficiencies in cultural markets and social recommendation systems due to lower quality content being chosen over higher quality content. For example, in a study of user biases [186], a set of songs were shown to individuals on the website MusicLab, ordered either at random or by the number of times previous users chose each song. In the latter case, the chance a song was subsequently chosen varied widely, and was only weakly correlated with its underlying quality, based on its popularity when ordered at random. In this part of the section, the limits of crowd performance are explored, asking how well crowds can choose the best answer to a question on a topic they presumably know well, given the amount and types of information individuals within the crowd receive.

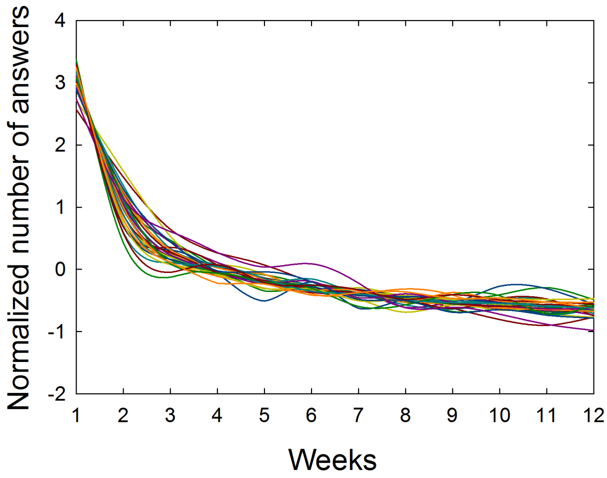


Figure 3.23: *Normalized number of contributions users make since the start of their tenure on the board. Each line is for a different board.*

Like other Q&A sites, such as Quora and Yahoo! Answers, SE has a number of features designed to promote user participation and enhance collaborative knowledge creation. User participation on the site includes content creation (asking and answering questions) and peer evaluation of existing content through two mechanisms: (1) users can *vote* for questions and answers, and (2) a user who asks a question can *accept* a specific answer. Prior research on Q&A sites has shown that upvoted answers can provide useful information on content quality [4,27,217]. For example, Kim and Oh [112] examined how users evaluate information in Yahoo! Answers forums, by examining the comments askers leave on answers. They found socioemotional-, content-, and utility-related criteria are dominant in the choice of the best answer, and found users evaluate information based not only upon the content, but also on cognitive and collaborative aspects. Adamic et al. [2] conducted a large-scale network analysis of Yahoo! Answers, trying to predict which answers would be judged best. They found, for both technical and non-technical sites, that answer length and the number of other answers the asker has to choose from are the most significant features to predict the future best answer, but a preference for longer answers diminishes with the number of answers. One limitation in these previous studies, however, is in assuming that the answer an asker chose was the "best" answer, and did not correct for asker biases when choosing any answer.

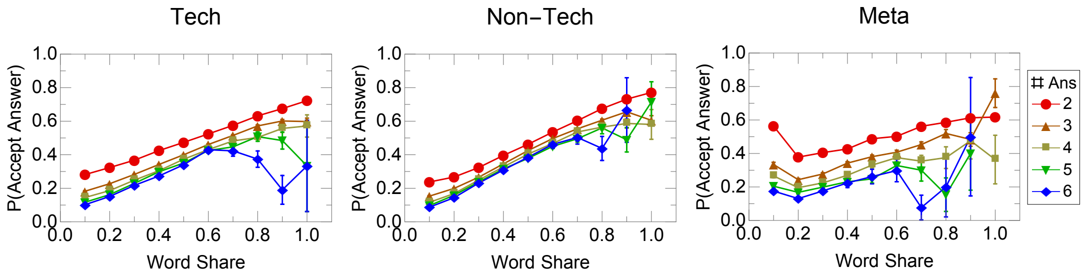
Crowdsourced peer evaluations are thought to provide a valuable signal about the quality of user contributions. By voting for stories on social news sites [200], favoriting photos [128], and retweeting messages [221], users can signal their opinions about the content. These signals, in aggregate, reflect how the community evaluates the material, enabling future users to identify higher quality answers to questions they have, without going through the effort of asking users the question themselves. Community feedback, however, can bias individual judgments, creating an "irrational herding" effect that can obscure the underlying quality of choices [186]. Moreover, the constraints of available time, motivation, and even cognitive resources, limit the effort users invest in processing and evaluating the relevant information. This phenomenon, known as "bounded rationality" [107, 194], profoundly affects the decisions people make, including their behavior online. Because people do not have the time nor capacity to process all information, they employ a variety cognitive heuristics to quickly (and unconsciously) decide what information to attend to [108]. These heuristics introduce predictable biases into their behavior. Social influence, *aka* "bandwagon effect", is one such heuristic: people pay attention to the choices of other people. Another important heuristic for online behavior is the "position bias" [163]: people pay more attention to items at the top of the list or the screen than those below. Heuristics may interact with how a web site presents information to alter collective performance. For example, when items are ranked by their popularity, the number of votes they receive is less reflective of their underlying quality than when they are ranked by time of latest vote [127].

This study presents evidences that cognitive heuristics affect the performance of Q&A communities. Instead of evaluating all answers, SE users appear to rely on simple heuristics for selecting the best answer, especially when there are many answers to choose from. Some of these heuristics, such as answer length, are features of the answer. Others arise from the contributions and judgments of the crowd, for example, whether an answer has received many votes. These attributes provide a signal of answer quality, but that signal may be obscured by herding dynamics. Also other attributes as potential proxies of answer quality are considered, such as how well the answer is readable, and the answerer's reputation. The main idea is to analyze the accepted answers, in order to show that users rely on simple heuristics as a proxy of answer quality, and these heuristics have a bigger impact on their choices than other proxies of quality.

Drivers of user behavior

Figure 3.24 shows that the probability of accepting an answer as a function of the answer's share of words across the different sites and when different numbers of answers are available to the asker. Here, word share is the fraction of words across all answers that a particular answer has. Similar trends exist for the probability to vote for an answer. Naïvely plotting acceptance probability, a few clear effects can be seen. Firstly, the probability of accepting an answer increases with word share, suggesting lengthy answers are preferred by askers. The effect becomes more pronounced when the asker has more answers from which to choose. Finally, users on different sites exhibit quantitative behavioral differences in this regard. Namely, the trends are less pronounced for meta sites, and that non-technical sites have a more linear relation between acceptance probability and word share compared to other sites.

Although this figure is suggestive, it gives an incomplete picture of the underlying dynamics. First, if a greater word share reflects answer quality (e.g., because askers could favor longer, more informative, answers) why is the effect different across types of sites, not monotonic, and, most importantly, dependent on the number of answers? This may suggest that askers employ a heuristic that uses word share as a proxy for quality, especially when faced with numerous options. For example, a larger number of answers requires greater cognitive effort to process, which individuals may want avoid. Efforts to minimize cognitive effort may also explain why the acceptance probability dips for answers with largest share of the total words. The purpose of this study is to separate answer quality from cognitive heuristics, and thus explore what can bias



(a) **(b)** **(c)**

Figure 3.24: *The probability of accepting an answer versus the word share for (a) technical, (b) non-technical, and (c) meta sites. Technical and non-technical site askers seem to behave similarly, while meta site askers have an acceptance probability that saturates at a significantly lower level than other sites, although meta site askers also seem to have a preference towards answers with a small word share.*

the choice of answers. To understand what drives voters and askers to pick a particular answer, it is necessary to examine how the probability of choosing an answer is affected by particular attributes, while controlling for other factors. These attributes include answer features that serve as quality proxies yet are unlikely to be used as heuristics (attributes 1–6), as well as heuristic candidates (attributes 7–13):

1. Answerer's reputation at the time the answer was created;
2. Mean rate reputation increases over time;
3. Answer's Flesch Reading Ease [113], or readability, score;
4. Answerer's tenure (i.e., time since joining the site) at the time of the answer;
5. Number of hyperlinks per answer;
6. Binary value denoting whether the answer was eventually accepted (for voting only);
7. Answer score before each vote;
8. Answer's score share (fraction of the total score of all answers on a given question);
9. Default web page order for an answer (i.e., its relative position);
10. Chronological order of an answer (whether it was first, second, third, etc.);
11. Time since an answer was created;
12. Number of words per answer;
13. Answer's word share.

These attributes are potential proxies of answer quality, and several are clearly visible on the web page by the users (Figure 3.15). Instead of evaluating all answers to select a good answer, a user could simply choose an answer based on one or more of these attributes, thereby using a decision heuristic. To study the impact of such heuristics on user choices, the probability of accepting (or voting for) an answer is modeled using a version of *logistic regression*. While logistic regression may not provide a complete explanation of the analyzed data, this method offers a principled approach to measure the relative importance of different factors. Specifically, binary variables (e.g., vote for an answer or not; accept an answer or do not) are chosen with a probability equal to:

$$F(x) = \frac{1}{1 + \exp[-(\beta_0 + \beta \cdot \mathbf{x})]} \tag{3.14}$$

where x is a vector of the answer attributes and β characterizes how much the probability depends on each attribute. Due to the correlation of some attributes with others and the multi-dimensionality of the data, the *LASSO regression* is used to fit data to the same function. The attribute dependence parameters, β_0, β_1, ... etc., are determined by maximizing the likelihood function with the addition of a penalty to avoid overfitting [96]. The package "glmnet" on R [79] is used, which allows for fast and accurate determination of β. Due to large variability in attribute values, all but three attributes are normalized by mapping them to their associated cumulative distribution function. The attributes "vote share", "word share", and "whether the answer was accepted" remain unnormalized because they are naturally between 0 and 1.

The regression coefficient β_i shows how a user's choice of an answer depends on the attribute i, independent of all other attributes. If β_i is positive, then larger (smaller) values of attribute i increase (decrease) the probability of voting for or accepting an answer. If β_i becomes larger (smaller) with the number of available answers, then the effect this attribute has on the vote probability becomes more (less) pronounced. Therefore, if β_i increases (decreases) with the number of answers, probability of acceptance or vote has a larger (smaller) dependence on attribute i. Figure 3.25 shows the relative importance of all the attributes, averaged over the first 20 answers. Error bars show the variance in these values as the number of available answers increases. The vote share and word share had the greatest effect, even after controlling for other potential quality proxies.

Figure 3.26 shows the score share versus the number of answers and notice that not only does the score share have a large effect, but that this effect increases with the number of answers. Not only is the social signal provided by score share an important factor in determining votes, but individuals appear increasingly rely on this signal as the number of available answers grows. The askers seem to be more strongly affected by score share than voters, i.e., they appear to use different heuristics than voters.

These results show that a rich-get-richer effect could partly explain vote share variation [231], because the probability of voting for answers depends strongly on the vote share (Figures 3.25 & 3.26). In other words, $p(v/n) \sim f(v/n)$, for a particular question where n is the sum of votes within a question, which is similar to a common rich-get-richer model where $p(v/n) \sim v/n$ [231]. However, this is not the only explanation for the distribution, because there does appear to be some dependence on, e.g., answer quality and word share, and, even with all the attributes included. More generally, it is not possible to completely rule out other covariates that may affect voting or choosing an answer, but the attributes that we believe signal quality and are unlikely to be used as heuristics (e.g., readability and the answerer's reputation) are very suggestive.

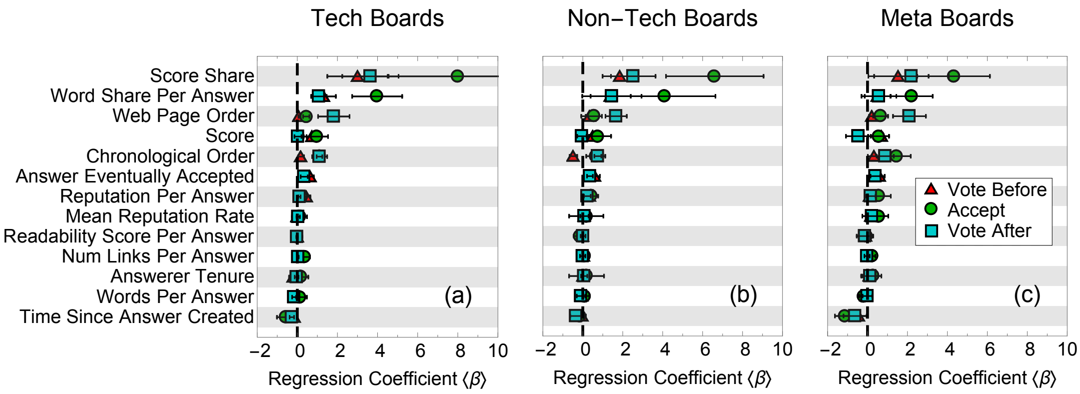
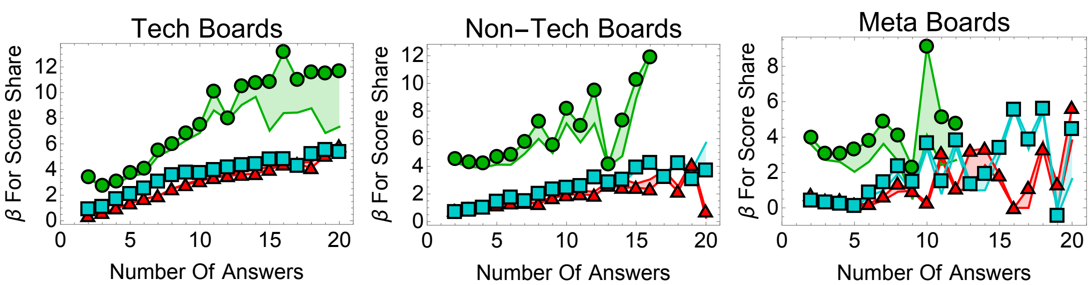


Figure 3.25: *Parameters for answerers to accept (green circles) and voters to vote for an answer both before (red triangles) and after (blue squares) an answer is accepted on (a) technical, (b) non-technical, and (c) meta sites, averaged over the number of available answers from 2-20. Higher values of regression coefficients indicate a stronger relationship between attributes and user behavior (voting or accepting an answer). Error bars indicate the variance of these values as the number of answers increases. The vote share, and word-share are the biggest factors overall, although this value changes significantly as the number of answers increases, creating a wide variance.*



(a) (b) (c)

Figure 3.26: *Score share regression coefficients for voting before (red triangles) and after (blue squares) an answer is accepted, as well as accepting an answer (green circles) for (a) technical, (b) non-technical, and (c) meta sites, with 2 to 20 answers. The score share has a large and increasingly important effect as the number of answers increases, and accepting an answer has an even stronger dependence on the vote share than voting for an answer does.*

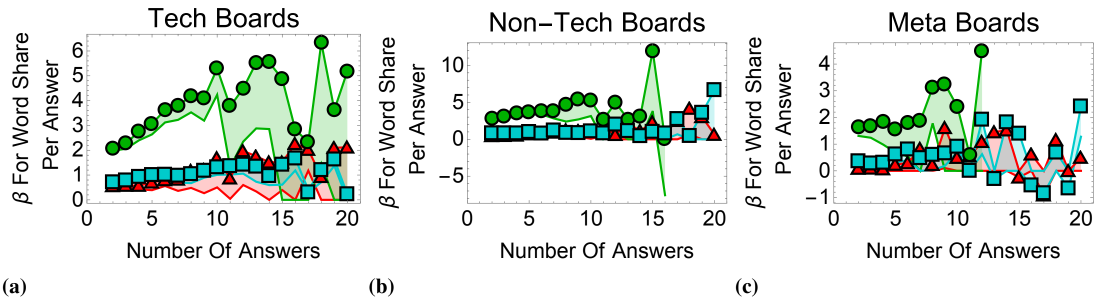
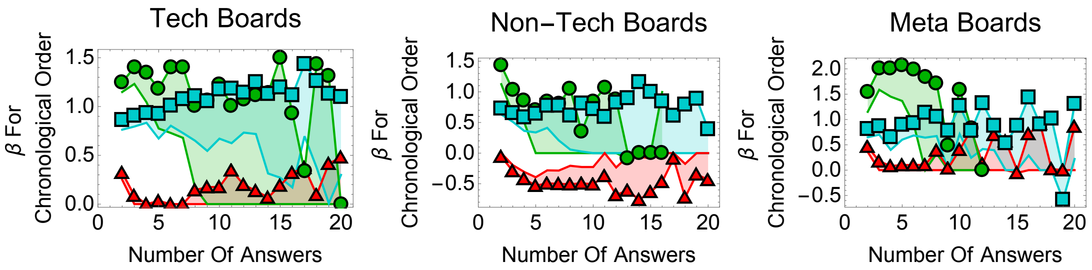


Figure 3.27: *Word share regression coefficients for voting before (red triangles) and after (blue squares) an answer is accepted, as well as accepting an answer (green circles) for (a) technical, (b) non-technical, and (c) meta sites, with 2 to 20 answers. Word share is second only to the vote share in likelihood to accept an answer. This discrepancy between voting and accepting an answer is plausibly because voters are much less likely to read a long answer, which may be of high quality, versus askers who want a thorough answer.*

The readability score (Figure 3.25) is not found to be significantly far from 0, while the reputation begins as a positive value but is not statistically significantly far from 0 when the total number of answers is between 10 and 15. Voters appear to change their behavior when an asker accepts an answer, based on the default answer order, and seem to be affected by the social signal of answer acceptance, an effect that increases with the number of answers visible. Coming back to the initial motivation, it is possible to see that the word share is a large and increasingly important factor that affects whether askers accept an answer (Figure 3.27). Its effect on voters, however, is much more modest, suggesting askers deliberately choose longer answers compared to voters.

Finally, quantitative and even qualitative differences appear between types of sites. This is most easily seen in Figure 3.28, where voters in non-technical sites are more likely to choose older rather than newer answers. In other types of sites, there is no real dependence on answer age, although there is apparently a strong positive dependence for votes after an answer is accepted, as well as for answerers accepting an answer. The reason older answers are preferred for voters in non-technical sites, however, is not altogether clear, especially when votes made after an answer is accepted appear similar across sites. Other differences are apparent in the Figure 3.25, where meta site users have a lower dependence on the score and word share than other sites, in agreement with Figure 3.24, possibly suggesting meta site users depend less on heuristics when reaching a decision.

Overall, there are clear differences between types of sites, and how answers are chosen that cannot be fully explained by differences in answer quality, as measured by the various proxies of quality. More intriguingly, it is notable that score share has a large and increasingly important effect on a user's behavior as the number of answers increases. Since the number of answers measures information load, it appears that users rely on social signals (e.g., score share) as heuristic of answer quality, especially as there are more answers to choose from. In contrast, other plausible proxies for quality, such as whether the answer is eventually accepted and answerer's reputation, have a smaller effect that decreases with the number of answers. Finally, voter behavior changes strongly just after an answer is accepted, suggesting that a social signal like



(a) **(b)** **(c)**

Figure 3.28: *Chronological answer order regression coefficients for voting before (red triangles) and after (blue squares) an answer is accepted, as well as accepting an answer (green circles) for (a) technical, (b) non-technical, and (c) meta sites, with 2 to 20 answers. There is not just quantitative but qualitative differences between site types using this attribute. It is notable that the largest site was removed from both meta and non-technical sites without any significant quantitative difference in the behavior (not shown). For tech sites, all coefficients are positive, although the coefficient for voting for an answer before it is accepted is near zero. For non-technical sites, however, the coefficient is strongly negative, suggesting users choose the oldest, rather than the newest answers, holding all things equal. Finally for meta sites, there is a much larger coefficient for accepting an answer than voting, suggesting askers choose much newer answers than in other sites.*

"this asker likes this answer" can stymie their better judgment, similar to lab experiments on choosing the "best" picture [237]. Furthermore, the answer found by users may not necessarily be the best answer because social signals are a much larger driver of behavior, as also seen in previous work [135].

3.3.4 Discussion

The growing success of Q&A communities depends principally on the goodwill of users to answer questions. In this section, the behavior of users within several active communities belonging to the Stack Exchange network was analyzed. At the beginning, the study focused on the correlation between success, performance of veterans and novices, and diversity in term of tenure, within technical and non-technical communities. Some measures of success in a quantitative and qualitative way were proposed, respectively using the percentage of answered questions and the mean highest score of the answers. At the same time, diversity was defined as the variance of the tenure of users inside the community. The results showed that veterans perform in an exceptional way, and that they drive the quality of answers inside communities. However, this success also depends on the contribution of novices. Diversity is not correlated with the overall success of Q&A communities. Indeed, the results presented show that the effect of diversity on success depends on the nature of the topic. This means that on non-technical topics, diversity in users' tenure increases the ability of the community to obtain better answers. On the other hand, on technical boards, expertise prevails over diversity, so it is more important to maintain a mature user community. This leads to hypothesize that within non-technical ad-hoc groups, a mix of veterans and novices can improve the success of the group, due to the high level of enthusiasm new users have at the beginning of their tenure. However, within technical topics, high level of expertise results in greatest success. Additional research is necessary to better understand the role of diversity in Q&A communities. In particular, though we have explored a variety of measures here, additional measures of success and diversity should be fully explored, in order to ascertain robustness of these results.

Furthermore, evidence that Stack Exchange users rely on cognitive heuristics were presented. Users use cognitive heuristics rather than their own judgments, to choose answers to vote for or accept. In particular, askers appear to rely more on heuristics than voters, when judging the best answer, while voters, although apparently less susceptible to heuristics, are in turn affected by the social signal that a particular answer is accepted, and change their behavior in order to further upvote that (somewhat arbitrarily) chosen answer or other top answers. Presumably, this is because askers are relatively naive about the topic and therefore cannot adequately judge the answer quality. The results showed that once an answer is accepted, voters significantly change their behavior to vote for the accepted answer, rather than evaluate other available answers.

CHAPTER 4

Conclusion

Complex systems are influenced by several complex factors that exhibit nonlinear patterns. This thesis presented a series of models able to capture the complexities of the behaviors of several systems in a realistic way, supporting in this way the decision makers in the complexity management. In particular, two main topics were discussed in this work: (1) how manufacturing systems can be modeled and how to analyze the complexity of their behaviors, and (2) how machine learning models can be used to make predictions in complex and dynamic environments.

The first topic was addressed by presenting models concerning the balancing of assembly lines, degradation of the machines in production lines, operation planning, and automated warehouses. Particularly, concerning the assembly lines, the existing approaches to solve the balancing problem do not take into consideration the physical allocation of the components used during the executions of the tasks. A method that addresses this issue was presented, balancing assembly lines trying to optimize the global picking time of its components. The proposed approach was able to produce solutions respecting the time constraint and then to evolve them in order to optimize the allocation of the components in different areas complying the spatial constraint. The method was ideologically divided into two phases: the first one solved a 1D Bin Packing Problem using time as the only dimension, then the second part optimized the components' allocation considering the spatial constraint. Two different evolutionary approaches were proposed: genetic algorithm and genetic programming. The genetic algorithm explored a variety of initial solutions given by the bin packing problem optimizing the allocation within the stations. The genetic programming approach, on the other hand, set some nodes in tree-like structures in which a series of heuristics were randomly generate in order to solve the remaining constraints imposed by the objective function. Moreover, the use of a multi-agent model able to simulate and analyze repairable manufacturing systems was presented. The agents approach has allowed to model the complexity derived from to the number of machines in a production line and the influence of several factors in the failure behavior of the machines. In the model, the stochastic deterioration, repairs, and replacements of the machines over time influenced their production rates. Furthermore, in order to present how to manage the complexity intrinsic into the operations' planning, a model inspired by the behavior of an ant colony was showed. The common solutions developed to face this problem do not suit the peculiarities of developing countries, such as the very dynamic situation of roads and traffic. A multi-agent model developed to solve operational planning problems was used in Buenos Aires, Argentina. The results showed that the created schedules respect the different constraints. In particular, the benefits of the automated solution can be summarized in the creation of more efficient schedules, reduction of staff requirements for operational planning process, reduction of costs for both vehicles and operators, and introduction of enhanced flexibility and dynamism in the planning process. Finally, it was presented how a multi-agent approach can be used to facilitate the modeling of different automated storage retrieval systems, improving the flexibility of simulated scenarios. It has been achievable modeling separately the design and control decisions. A concrete model was presented, able to investigate a new policy to find an optimal dwell point in automated warehouses with a random allocation of unit loads. The model was tested considering a wide set of system configurations, simulating different storage dimensions and arrivals. Exploiting an idea deriving from force-fields, the model found an interesting dwell point policy in terms of wait time. Results showed that the proposed model is able to minimize the wait time, i.e., the unit loads are achieved first, increasing the respond time of the warehouse, making it leaner.

The second topic was an exploration of several machine learning models able to capture the complexities of systems behaviors in a realistic way. Indeed, the prediction of complex system cannot depend on the assumptions of independence and linearity. In particular, prediction in industrial and dynamic environments is a challenge that professionals and academics have to face more and more. In this thesis, machine learning models were applied to different fields, from the reliability of mechanical and electrical components, to renewable energy. An easy approach to artificial neural networks was explained, in order to facilitate the use of these models to non expert users. Networks with different neurons in their hidden layer and with different parameters were analyzed, in order to understand how these factors influence their performance and execution time needed for the learning phase. The results showed that simple networks can achieve satisfactory performance, even better than one of the most used method, the Weibull distribution. Then, the performance of other machine learning models were analyzed, showing how random forests emerged as the best model in average, even in the presence of few failure times. Additionally, the results showed that machine learning models can get better results in presence of censored data than when using a complete set. This research showed one more time the potential of the machine learning approach in the reliability analysis. The performance of artificial neural networks were also studied to predict the average daily global solar radiation over several Italian places. An analysis of the relevance of the input data was conducted, showing that, for Italy, 7 inputs provide the best results. The inputs are in order of relevance, top of atmosphere radiation, day duration, rainy days, altitude, rainfall, time period and latitude. Using these inputs, the networks outperform all the other scenarios in 45 geographically distributed Italian cities. Furthermore, an agent-based model related to the use of web services to balance the exchanging of green energy in a hypothetical smart city was presented. In the model, agents acted on behalf of energy producers and consumers, and the amount of energy exchanged was influenced by several factors related to the energy market context. A web service integration was a useful source of information for providing input for the weather prediction. By combining several external factors into different models which predict the future production of energy via renewable electricity, a realistic model of a North-Western region of Italy was constructed. A simulation showed that agents are able to adapt to different weather conditions and consumers' habits. In the end, some models able to figure out some important patterns that influence the users' behavior within online social communities were introduced. In particular, a huge network of question answering communities was analyzed, figuring out the influence of diversity in users' tenure in the ability of the community to produce worse or better answers. The results claimed that within non-technical ad-hoc groups, a mix of veterans and novices can improve the success of the group, due to the high level of enthusiasm new users have at the beginning of their tenure. However, within technical topics, high level of expertise results in greatest success. Moreover, evidence that users within online communities rely on cognitive heuristics were found. Users used cognitive heuristics rather than their own judgments, to choose answers to vote for or accept. In particular, askers appeared to rely more on heuristics than voters, when judging the best answer, while voters, although apparently less susceptible to heuristics, are in turn affected by the social signal that a particular answer is accepted, and change their behavior in order to further upvote that (somewhat arbitrarily) chosen answer or other top answers.

In conclusion, presenting how manufacturing systems can benefit from modeling and how advanced models can be used to make predictions, this thesis aims to be an inspiration for those people which have to manage the complexity in industrial and dynamic environments. Showing examples and results, this work explains how to make this world a little more understandable.

Bibliography

[1] Erwin J.W. Abbink, Luís Albino, Twan Dollevoet, Dennis Huisman, Jorge Roussado, and Ricardo Lopes Saldanha. Solving large scale crew scheduling problems in practice. *Public Transport*, 3(2):149–164, 2011.

[2] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. Knowledge sharing and yahoo answers: everyone knows something. In *Proceedings of the 17th international conference on World Wide Web*, pages 665–674. ACM, 2008.

[3] El Houssaine Aghezzaf, Mohamed Anouar Jamali, and Daoud Ait-Kadi. An integrated production and preventive maintenance planning model. *European Journal of Operational Research*, 181(2):679–685, 2007.

[4] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183–194. ACM, 2008.

[5] Ahmed Z. Al-Garni and Ahmad Jamal. Artificial neural network application of modeling failure rate for boeing 737 tires. *Quality and Reliability Engineering International*, 27(2):209–219, 2011.

[6] José M. Alonso, Luis Magdalena, and Gil González-Rodríguez. Looking for a good fuzzy system interpretability index: An experimental approach. *International Journal of Approximate Reasoning*, 51(1):115–134, 2009.

[7] Emanuel Federico Alsina and Giacomo Cabri. An agent-based approach for transportation scheduling in developing countries. In *Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on*, pages 40–47. IEEE, 2013.

[8] Emanuel Federico Alsina, Giacomo Cabri, and Alberto Regattieri. An agent-based approach to simulate production, degradation, repair, replacement and preventive maintenance of manufacturing systems. In *Computational Intelligence in Production and Logistics Systems (CIPLS), 2014 IEEE Symposium on*, pages 24–31. IEEE, 2014.

[9] Emanuel Federico Alsina, Giacomo Cabri, and Alberto Regattieri. A neural network approach to find the cumulative failure distribution: Modeling and experimental evidence. *Quality and Reliability Engineering International*, 2015.

[10] Emanuel Federico Alsina, Giacomo Cabri, Alberto Regattieri, and Giulia Santarelli. An agent-based approach for finding the optimal dwell point in automated storage retrieval systems. In *Software, Knowledge, Information Management and Applications (SKIMA), 2013 International Conference on*. In press, 2013.

[11] Emanuel Federico Alsina, Nicola Capodieci, Giacomo Cabri, Alberto Regattieri, Mauro Gamberi, Francesco Pilati, and Maurizio Faccio. The influence of the picking times of the components in time and space assembly line balancing problems: an approach with evolutionary algorithms. In *Computational Intelligence in Production and Logistics Systems (CIPLS), 2015 IEEE Symposium on*, page In press. IEEE, In press, 2015.

[12] Nima Amjady and Mehdi Ehsan. Evaluation of power systems reliability by an artificial neural network. *Power Systems, IEEE Transactions on*, 14(1):287–292, 1999.

[13] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th*

ACM SIGKDD international conference on Knowledge discovery and data mining, pages 850–858. ACM, 2012.

[14] Ediz Atmaca and Ayla Ozturk. Defining order picking policy: a storage assignment model and a simulated annealing solution in as/rs systems. *Applied Mathematical Modelling*, 37(7):5069–5079, 2013.

[15] Nii O Attoh-Okine. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. *Advances in Engineering Software*, 30(4):291–302, 1999.

[16] Anna Azzi, Daria Battini, Maurizio Faccio, Alessandro Persona, and Fabio Sgarbossa. Innovative travel time model for dual-shuttle automated storage/retrieval systems. *Computers & Industrial Engineering*, 61(3):600–607, 2011.

[17] Rebecca J. Barthelmie, Frank Murray, and Sara C. Pryor. The economic benefit of short-term forecasting for wind energy in the uk electricity market. *Energy Policy*, 36(5):1687–1696, 2008.

[18] Mahdi Bashiri and A. Farshbaf Geranmayeh. Tuning the parameters of an artificial neural network using central composite design and genetic algorithm. *Scientia Iranica*, 18(6):1600–1608, 2011.

[19] Daria Battini, Maurizio Faccio, Alessandro Persona, and Alberto Regattieri. Buffer design for availability: a new simulative study in case of infant and random failures. *International Journal of Services and Operations Management*, 14(2):157–174, 2013.

[20] Joaquin Bautista and Jordi Pereira. Ant algorithms for a time and space constrained assembly line balancing problem. *European Journal of Operational Research*, 177(3):2016–2032, 2007.

[21] Adil Baykasoğlu and Lale Özbakır. Discovering task assignment rules for assembly line balancing via genetic programming. *The International Journal of Advanced Manufacturing Technology*, 76(1-4):417–434, 2015.

[22] Christian Becker and Armin Scholl. A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168(3):694–715, 2006.

[23] M.A. Behrang, E. Assareh, A. Ghanbarzadeh, and A. R. Noghrehabadi. The potential of different artificial neural network (ann) techniques in daily global solar radiation modeling based on meteorological data. *Solar Energy*, 84(8):1468–1480, 2010.

[24] Fariba Besharat, Ali A. Dehghan, and Ahmad R. Faghih. Empirical models for estimating global solar radiation: A review and case study. *Renewable and Sustainable Energy Reviews*, 21:798–821, 2013.

[25] Hakim Nadir Bessenouci, Zaki Sari, and Latéfa Ghomri. Metaheuristic based control of a flow rack automated storage retrieval system. *Journal of Intelligent Manufacturing*, 23(4):1157–1166, 2012.

[26] Maurizio Bevilacqua, Marcello Braglia, Marco Frosolini, and Roberto Montanari. Failure rate prediction with artificial neural networks. *Journal of Quality in Maintenance Engineering*, 11(3):279–294, 2005.

[27] Mohan John Blooma, Dion Hoe-Lian Goh, and Alton Yeow-Kuan Chua. Predictors of high-quality answers. *Online Information Review*, 36(3):383–400, 2012.

[28] Eric Bonabeau. Predicting the unpredictable. *Harvard Business Review*, 80(3):109–116, 2002.

[29] Andrei Borshchev and Alexei Filippov. From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. In *Proceedings of the 22nd international conference of the system dynamics society*, volume 22, 2004.

[30] Marco Bortolini, Riccardo Accorsi, Mauro Gamberi, Riccardo Manzini, and Alberto Regattieri. Optimal design of as/rs storage systems with three-class-based assignment strategy under single and dual command operations. *The International Journal of Advanced Manufacturing Technology*, pages 1–13, 2015.

[31] Marco Bortolini, Mauro Gamberi, Alessandro Graziani, Riccardo Manzini, and Cristina Mora. Multi-location model for the estimation of the horizontal daily diffuse fraction of solar radiation in europe. *Energy Conversion and Management*, 67:208–216, 2013.

[32] Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 866–874. ACM, 2008.

[33] Corinne Boutevin, Michel Gourgand, and Sylvie Norre. Bin packing extensions for solving an industrial line balancing problem. In *Assembly and Task Planning, 2003. Proceedings of the IEEE International Symposium on*, pages 115–121. IEEE, 2003.

[34] George E.P. Box and Norman Richard Draper. *Empirical model-building and response surfaces*, volume 424. Wiley New York, 1987.

[35] Nils Boysen, Malte Fliedner, and Armin Scholl. Assembly line balancing: Which model to use when? *International Journal of Production Economics*, 111(2):509–528, 2008.

[36] Yavuz A. Bozer and John A. White. Travel-time models for automated storage/retrieval systems. *IIE transactions*, 16(4):329–338, 1984.

[37] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[38] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[39] Simon Brezovnik, Janez Gotlih, Jože Balič, Karl Gotlih, and Miran Brezočnik. Optimization of an automated storage and retrieval systems by swarm intelligence. *Procedia Engineering*, 100:1309–1318, 2015.

[40] Joseph Bukchin and Michal Tzur. Design of flexible assembly line to minimize equipment cost. *Iie transactions*, 32(7):585–598, 2000.

[41] Yossi Bukchin and Russell D. Meller. A space allocation algorithm for assembly line components. *IIE Transactions*, 37(1):51–61, 2005.

[42] Nicola Capodieci, Emanuel Federico Alsina, and Giacomo Cabri. A context-aware agent-based approach for deregulated energy market. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on*, pages 16–21. IEEE, 2012.

[43] Nicola Capodieci, Emanuel Federico Alsina, and Giacomo Cabri. Context-awareness in the deregulated electric energy market: an agent-based approach. *Concurrency and Computation: Practice and Experience*, 2013.

[44] Nicola Capodieci, Giacomo Cabri, Giuliano Andrea Pagani, and Marco Aiello. Adaptive game-based agent negotiation in deregulated energy markets. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 300–307. IEEE, 2012.

[45] Nicola Capodieci, Giuliano Andrea Pagani, Giacomo Cabri, and Marco Aiello. Smart meter aware domestic energy trading agents. In *Proceedings of the 2011 workshop on E-energy market challenge*, pages 1–10. ACM, 2011.

[46] Avishai Ceder. *Public transit planning and operation: theory, modeling and practice*. Elsevier, Butterworth-Heinemann, 2007.

[47] Snehamoy Chatterjee and Sukumar Bandopadhyay. Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. *Expert Systems with Applications*, 39(12):10943 – 10951, 2012.

[48] Hailiang Chen, Prabuddha De, Yu Jeffrey Hu, and Byoung-Hyoun Hwang. Wisdom of crowds: The value of stock opinions transmitted through social media. *Review of Financial Studies*, 27(5):1367–1403, 2014.

[49] Xianshun Chen, Yew-Soon Ong, Puay-Siew Tan, Neng Sheng Zhang, and Zhengping Li. Agent-based modeling and simulation for supply chain risk management-a survey of the state-of-the-art. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 1294–1299. IEEE, 2013.

[50] Zhuxi Chen, Xiaoping Li, and Jatinder N.D. Gupta. Sequencing the storages and retrievals for flow-rack automated storage and retrieval systems with duration-of-stay storage policy. *International Journal of Production Research*, (ahead-of-print):1–15, 2015.

[51] Leo H Chiang, Richard D Braatz, and Evan L Russell. *Fault detection and diagnosis in industrial systems*. Springer, 2001.

[52] Manuel Chica, Joaquín Bautista, Óscar Cordón, and Sergio Damas. A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega*, 58:55–68, 2016.

[53] Manuel Chica, Oscar Cordon, and Sergio Damas. An advanced multiobjective genetic algorithm design for the time and space assembly line balancing problem. *Computers & Industrial Engineering*, 61(1):103–117, 2011.

[54] Manuel Chica, Oscar Cordón, Sergio Damas, and Joaquín Bautista. A new diversity induction mechanism for a multi-objective ant colony algorithm to solve a real-world time and space assembly line balancing problem. *Memetic computing*, 3(1):15–24, 2011.

[55] Gianfranco Chicco, Roberto Napoli, Petru Postolache, Mircea Scutariu, and Cornel Toader. Customer characterization options for improving the tariff offer. *Power Systems, IEEE Transactions on*, 18(1):381–387, 2003.

[56] Oscar Cordón, E. Herrera, E .and Gomide, E. Hoffman, and Luis Magdalena. Ten years of genetic fuzzy systems: current framework and new trends. In *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, volume 3, pages 1241–1246. IEEE, 2001.

[57] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[58] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94. ACM, 2008.

[59] Poulami Dalapati, Arambam James Singh, Arin Dutta, and Surya Bhattacharya. Multi agent based railway scheduling and optimization. In *TENCON 2014 IEEE Region 10 Conference*, pages 1–6. IEEE, 2014.

[60] Christopher Daly. Guidelines for assessing the suitability of spatial climate data sets. *International journal of climatology*, 26(6):707–721, 2006.

[61] Márcio das Chagas Moura, Enrico Zio, Isis Didier Lins, and Enrique Droguett. Failure and reliability prediction by support vector machines regression of time series data. *Reliability Engineering & System Safety*, 96(11):1527–1534, 2011.

[62] Richard H. Day and Bruce Mizrach. Complex economic dynamics. volume 1. an introduction to dynamical systems and market mechanisms. *Journal of Economic Literature*, 33(4):1976–1976, 1995.

[63] René B.M. De Koster, Tho Le-Duc, and Yu Yugang. Optimal storage rack design for a 3-dimensional compact as/rs. *International journal of production research*, 46(6):1495–1514, 2008.

[64] Masoud Dehghan and Reza Hoseinnezhad. Estimation of components reliability in petrochemical plants using a neural-weibull lifetime model. *Chemical Engineering Communications*, 196(8):917–931, 2009.

[65] Christos Dimopoulos and Ali Zalzala. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. *Evolutionary Computation, IEEE Transactions on*, 4(2):93–113, 2000.

[66] Christos Dimopoulos and Ali Zalzala. Investigating the use of genetic programming for a classic one-machine scheduling problem. *Advances in Engineering Software*, 32(6):489–498, 2001.

[67] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41, 1996.

[68] Norman R. Draper and Harry Smith. *Applied regression analysis*. John Wiley & Sons, 2014.

[69] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[70] Charles E. Ebeling. *An introduction to reliability and maintainability engineering*. Waveland Press, 2010.

[71] Pius J. Egbelu and Chungte T. Wu. A comparison of dwell point rules in an automated storage/retrieval system. *International Journal of Production Research*, 31(11):2515–2530, 1993.

[72] Maurizio Faccio, Mauro Gamberi, Alessandro Persona, Alberto Regattieri, and Fabio Sgarbossa. Design and simulation of assembly line feeding systems in the automotive sector using supermarket, kanbans and tow trains: a general framework. *Journal of Management Control*, 24(2):187–208, 2013.

[73] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using the second order information for training svm. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[74] Shu Fan and Luonan Chen. Short-term load forecasting based on an adaptive hybrid method. *Power Systems, IEEE Transactions on*, 21(1):392–401, 2006.

[75] Emilio Ferrari, Arrigo Pareschi, Alberto Regattieri, and Alessandro Persona. Tpm: situation and procedure for a soft introduction in italian factories. *The TQM Magazine*, 14(6):350–358, 2002.

[76] Pnina Fichman. A comparative assessment of answer quality on four question answering sites. *Journal of Information Science*, 37(5):476–486, 2011.

[77] John W. Fowler and Oliver Rose. Grand challenges in modeling and simulation of complex manufacturing systems. *Simulation*, 80(9):469–476, 2004.

[78] Kouedeu Annie Francie, Kenne Jean-Pierre, Dejax Pierre, Songmene Victor, and Polotski Vladimir. Stochastic optimal control of manufacturing systems under production-dependent failure rates. *International Journal of Production Economics*, 150:174–187, 2014.

[79] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[80] Jean-Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. On storage assignment policies for unit-load automated storage and retrieval systems. *International Journal of Production Research*, 50(3):879–892, 2012.

[81] Jean-Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. A simulation modeling framework for multiple-aisle automated storage and retrieval systems. *Journal of Intelligent Manufacturing*, 25(1):193–207, 2014.

[82] Jean-Philippe Gagliardi, Jacques Renaud, and Angel Ruiz. Sequencing approaches for multiple-aisle automated storage and retrieval systems. *International Journal of Production Research*, (ahead-of-print):1–11, 2015.

[83] Francis Galton. Vox populi (the wisdom of crowds). *Nature*, 75:450–51, 1907.

[84] Rita Gamberini, Andrea Grassi, Mauro Gamberi, Riccardo Manzini, and Alberto Regattieri. U-shaped assembly lines with stochastic tasks execution times: heuristic procedures for balancing and re-balancing problems. *Simulation Series*, 36(2):137, 2004.

[85] Nigel Gilbert and Pietro Terna. How to build and use agent-based models in social science. *Mind & Society*, 1(1):57–72, 2000.

[86] José Fernando Gonçalves and Jorge Raimundo De Almeida. A hybrid genetic algorithm for assembly line balancing. *Journal of Heuristics*, 8(6):629–642, 2002.

[87] Mikell P. Groover. *Automation, production systems, and computer-integrated manufacturing*. Prentice Hall Press, 2007.

[88] Janis Grundspenkis. Intelligent agents in logistics: Some trends and solutions. In *Proceedings of the 11th International Workshop on Harbor, Maritime & Multimodal Logistics Modeling & Simulation, Rende (CS), Italy*, pages 174–179, 2008.

[89] Qinglin Guo and Ming Zhang. A novel approach for multi-agent-based intelligent manufacturing system. *Information Sciences*, 179(18):3079–3090, 2009.

[90] Jatinder N. D. Gupta. An excursion in scheduling theory: An overview of scheduling research in the twentieth century. *Production Planning and Control*, 13(2):105–116, 2002.

[91] Knut Haase, Guy Desaulniers, and Jacques A. Desrosiers. Simultaneous vehicle and crew scheduling in urban mass transit systems. *The Institute for Operations Research and the Management Sciences*, 35(3):286–303, 2001.

[92] Ali Haghani and Mohamadreza Banihashemi. Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. *Transportation Research Part A: Policy and Practice*, 36(4):309–333, 2002.

[93] Trevor S. Hale, Faizul Huq, Jennifer V. Blackhurst, and Ken Cutright. Closed form models for dwell point locations for overhead bridge cranes. *International Journal of Operational Research*, 4(4):412–421, 2009.

[94] Rabab Hassan and Ghadir Radman. Survey on smart grid. In *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the*, pages 210–213. IEEE, 2010.

[95] Stephan Hassold and Avishai Avi Ceder. Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, 67:129–143, 2014.

[96] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[97] Francisco Herrera. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, 1(1):27–46, 2008.

[98] Robert C. Hilborn. Sea gulls, butterflies, and grasshoppers: A brief history of the butterfly effect in nonlinear dynamics. *American Journal of Physics*, 72(4):425–427, 2004.

[99] John Henry Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[100] Wei Chiang Hong and Ping Feng Pai. Predicting engine reliability by support vector machines. *The International Journal of Advanced Manufacturing Technology*, 28(1-2):154–161, 2006.

[101] Omar J. Ibarra-Rojas, Felipe Delgado, Ricardo E. Giesen, and Juan Carlos Muñoz. Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, 77:38–75, 2015.

[102] Richard Iles. *Public Transport in Developing Countries*. Elsevier, 2005.

[103] Teruaki Ito and S.M. Mousavi Jahan Abadi. Agent-based material handling and inventory planning in warehouse. *Journal of intelligent manufacturing*, 13(3):201–210, 2002.

[104] Anton Ivaschenko, Alexander Tsarev, Alina Vaysblat, and Petr Skobelev. Smart solutions multi-agent platform for dynamic transportation scheduling. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART)*, pages 372–375, 2011.

[105] Domagoj Jakobović and Leo Budin. Dynamic scheduling with genetic programming. In *Genetic Programming*, pages 73–84. Springer, 2006.

[106] Yuan Jia-hai, Yu Shun-kun, and Hu Zhao-guang. A multi-agent trading platform for electricity contract market. In *Power Engineering Conference, 2005. IPEC 2005. The 7th International*, pages 1024–1029. IEEE, 2005.

[107] Daniel Kahneman. Maps of bounded rationality: Psychology for behavioral economics. *The American economic review*, 93(5):1449–1475, 2003.

[108] Daniel Kahneman. *Thinking, fast and slow*. Macmillan, 2011.

[109] Edward L. Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[110] David W. Kelton and Averill M. Law. *Simulation modeling and analysis*. McGraw Hill Boston, 2000.

[111] Tamer Khatib, Azah Mohamed, and K. Sopian. A review of solar energy modeling techniques. *Renewable and Sustainable Energy Reviews*, 16(5):2864–2869, 2012.

[112] Yeo Keun Kim, Yong Ju Kim, and Yeongho Kim. Genetic algorithms for assembly line balancing with various objectives. *Computers & Industrial Engineering*, 30(3):397–409, 1996.

[113] Peter J. Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document, 1975.

[114] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[115] Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

[116] Davorin Kofjač, Miroljub Kljajić, and Valter Rejec. The anticipative concept in warehouse optimization using simulation in an uncertain environment. *European Journal of Operational Research*, 193(3):660–669, 2009.

[117] John R. Koza and Riccardo Poli. Genetic programming. In *Search Methodologies*, pages 127–164. Springer, 2005.

[118] Yoann Kubera, Philippe Mathieu, Sébastien Picault, et al. Interaction-oriented agent simulations: From theory to implementation. In *ECAI*, pages 383–387, 2008.

[119] Dinesh U. Kumar, John Crocker, Jezdimir Knezevic, and Mohamed El-Haram. *Reliability, Maintenance and Logistic Support: A Life Cycle Approach*. Springer, 2012.

[120] Ludmila I. Kuncheva. Combining classifiers: Soft computing solutions. World Scientific, 2001.

[121] Yiheng Kung, Yoshimasa Kobayashi, Toshimitsu Higashi, Masao Sugi, and Jun Ota. Order scheduling of multiple stacker cranes on common rails in an automated storage/retrieval system. *International Journal of Production Research*, 52(4):1171–1187, 2014.

[122] Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks–review and case studies. *Neural networks*, 14(3):257–274, 2001.

[123] Patrick R. Laughlin, Bryan L. Bonner, and Andrew G. Miner. Groups perform better than the best individuals on letters-to-numbers problems. *Organizational Behavior and Human Decision Processes*, 88(2):605–620, 2002.

[124] P Lavoie, J-P Kenné, and A Gharbi. Production control and combined discrete/continuous simulation modeling in failure-prone transfer lines. *International Journal of Production Research*, 45(24):5667–5685, 2007.

[125] Young Hae Lee, Moon Hwan Lee, and Sun Hur. Optimal design of rack structure with modular cell in as/rs. *International Journal of Production Economics*, 98(2):172–178, 2005.

[126] Paulo Leitão. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7):979–991, 2009.

[127] Kristina Lerman and Tad Hogg. Leveraging position bias to improve peer recommendation. *PloS one*, 9(6):e98914, 2014.

[128] Kristina Lerman and Laurie Jones. Social browsing on flickr. *Proceedings of 1st International Conference on Weblogs and Social Media (ICWSM-07)*, 2007.

[129] Wenjian Li and Hoang Pham. Reliability modeling of multi-state degraded systems with multi-competing failures and random shocks. *Reliability, IEEE Transactions on*, 54(2):297–303, 2005.

[130] Richard J. Linn and Richard A. Wysk. An analysis of control strategies for an automated storage/retrieval system. *INFOR*, 25(1):66–83, 1987.

[131] Ming C. Liu, Way Kuo, and Tep Sastri. An exploratory study of a neural network approach for reliability data analysis. *Quality and Reliability Engineering International*, 11(2):107–112, 1995.

[132] Tian Liu, Xianhao Xu, Hu Qin, and Andrew Lim. Travel time analysis of the dual command cycle in the split-platform as/rs with i/o dwell point policy. *Flexible Services and Manufacturing Journal*, pages 1–19, 2015.

[133] Ya-jie Liu, Tian-long Gu, Zhou-bo Xu, and Liang Chang. Parallel assembly sequence planning based on improved genetic programming. *Computer Integrated Manufacturing Systems*, 6:010, 2013.

[134] Stergios Lolas and Oluremi A. Olatunbosun. Prediction of vehicle reliability performance using artificial neural networks. *Expert Systems with Applications*, 34(4):2360–2369, 2008.

[135] Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.

[136] Ho-Yin Mak and Zuo-Jun Max Shen. Stochastic programming approach to process flexibility design. *Flexible services and manufacturing journal*, 21(3-4):75–91, 2009.

[137] Riccardo Manzini. *Warehousing in the global supply chain*. Springer, 2011.

[138] Riccardo Manzini, Alberto Regattieri, Hoang Pham, and Emilio Ferrari. *Maintenance for industrial systems*. Springer Science & Business Media, 2009.

[139] Marzio Marseguerra, Enrico Zio, M. Ammaturo, and Vincenzo Fontana. Predicting reliability via neural networks. In *Reliability and Maintainability Symposium, 2003. Annual*, pages 196–201. IEEE, 2003.

[140] Silvano Martello and Paolo Toth. *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990.

[141] Catherine McGeoch. *A Guide to Experimental Algorithmics*. Cambridge Univerisity Press, 2012.

[142] Kishan Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka. *Elements of artificial neural networks*. MIT press, 1997.

[143] Russell D. Meller and Anan Mungwattana. As/rs dwell-point strategy selection at high system utilization: A simulation study to investigate the magnitude of the benefit. *International Journal of Production Research*, 43(24):5217–5227, 2005.

[144] Adel Mellit and Alessandro Massi Pavan. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected pv plant at trieste, italy. *Solar Energy*, 84(5):807–821, 2010.

[145] Hubert Missbauer and Reha Uzsoy. Optimization models of production planning problems. *Planning Production and Inventories in the Extended Enterprise*, pages 437–507, 2011.

[146] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[147] László Monostori, József Váncza, and Soundar R.T. Kumara. Agent-based systems for manufacturing. *CIRP Annals-Manufacturing Technology*, 55(2):697–720, 2006.

[148] Geeju Moon, Gwang-Pil Kim, and William J. Moon. Improvement of as/rs performance using design and application of common zone. *International Journal of Production Research*, 47(5):1331–1341, 2009.

[149] Peter Nganga Muchiri, Liliane Pintelon, Harry Martin, and Peter Chemweno. Modelling maintenance effects on manufacturing equipment performance: results from simulation analysis. *International Journal of Production Research*, 52(11):3287–3302, 2014.

[150] Lev Muchnik, Sinan Aral, and Sean J. Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.

[151] Ian Nabney. *NETLAB: algorithms for pattern recognition*. Springer Science & Business Media, 2002.

[152] Gregoire Nicolis and Cathy Nicolis. *Foundations of Complex Systems: Emergence, Information and Predicition*. World Scientific, 2012.

[153] Dehayem F.I. Nodem, Jean Pierre Kenné, and Ali Gharbi. Simultaneous control of production, repair/replacement and preventive maintenance of deteriorating manufacturing systems. *International Journal of Production Economics*, 134(1):271–282, 2011.

[154] Mustapha Nourelfath, Eric Châtelet, and Nabil Nahas. Joint redundancy and imperfect preventive maintenance optimization for series–parallel multi-state degraded systems. *Reliability Engineering & System Safety*, 103:51–60, 2012.

[155] Djamila Ouelhadj, Peter Cowling, and Sanja Petrovic. Contract net protocol for cooperative optimisation and dynamic scheduling of steel production. In *Intelligent systems design and applications*, pages 469–478. Springer, 2003.

[156] Michael S. Packianather, Paul R. Drake, and Hefin Rowlands. Optimizing the parameters of multilayered feedforward neural networks through taguchi design of experiments. *Quality and Reliability Engineering International*, 16(6):461–473, 2000.

[157] Giuliano Andrea Pagani and Marco Aiello. Towards a service-oriented energy market: current state and trend. In *Service-Oriented Computing*, pages 203–209. Springer, 2011.

[158] Ping-Feng Pai and Kuo-Ping Lin. Application of hybrid learning neural fuzzy systems in reliability prediction. *Quality and Reliability Engineering International*, 22(2):199–211, 2006.

[159] Doru Panescu and Carlos Pascal. Holonic coordination obtained by joining the contract net protocol with constraint satisfaction. *Computers in Industry*, 2015.

[160] Byung Chun Park. An optimal dwell point policy for automated storage/retrieval systems with uniformly distributed, rectangular racks. *International Journal of Production Research*, 39(7):1469–1480, 2001.

[161] John Park, Su Nguyen, Mengjie Zhang, and Mark Johnston. Evolving ensembles of dispatching rules using genetic programming for job shop scheduling. In *Genetic Programming*, pages 92–104. Springer, 2015.

[162] James H. Patterson and Joseph J. Albracht. Technical note–assembly line balancing: Zero–one programming with fibonacci search. *Operations Research*, 23(1):166–172, 1975.

[163] Stanley Le Baron Payne. *The Art of Asking Questions: Studies in Public Opinion, 3*. Princeton University Press, 2014.

[164] Karl Pearson. Mathematical contributions to the theory of evolution.–on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proceedings of the royal society of london*, 60(359-367):489–498, 1896.

[165] Brett A. Peters, Jeremy S. Smith, and Trevor S. Hale. Closed form models for determining the optimal dwell point location in automated storage and retrieval systems. *International Journal of Production Research*, 34(6):1757–1771, 1996.

[166] Michael Pinedo. *Planning and scheduling in manufacturing and services*, volume 24. Springer, 2005.

[167] Iztok Potrč, Tone Lerher, Janez Kramberger, and Matjaž Šraml. Simulation model of multi-shuttle automated storage and retrieval systems. *Journal of Materials Processing Technology*, 157:236–244, 2004.

[168] Atika Qazi, H. Fayaz, A. Wadi, Ram Gopal Raj, N.A. Rahim, and Waleed Ahmed Khan. The artificial neural network for solar radiation prediction and designing solar systems: a systematic literature review. *Journal of Cleaner Production*, 2015.

[169] Markus Rabe and Uwe Clausen. Performance availability analysis of autonomous intralogistic systems: An agent-based simulation approac. In *ASIM Conference on simulation in production and logistics*, pages 207–2018, 2015.

[170] Paramjeet S. Rajpal, K.S. Shishodia, and Gurkirat S. Sekhon. An artificial neural network for modeling reliability, availability and maintainability of a repairable system. *Reliability Engineering & System Safety*, 91(7):809–819, 2006.

[171] R.K. Rajput. *A textbook of manufacturing technology: Manufacturing processes*. Firewall Media, 2007.

[172] Sabah U. Randhawa, Edward D. McDowell, and Wen-Tsao Wang. Evaluation of scheduling rules for single- and dual-dock automated storage/retrieval system. *Computers & industrial engineering*, 20(4):401–410, 1991.

[173] Sabah U. Randhawa and Raj Shroff. Simulation-based design evaluation of unit load automated storage/retrieval systems. *Computers & Industrial Engineering*, 28(1):71–79, 1995.

[174] Alberto Regattieri, Mauro Gamberi, Rita Gamberini, and Riccardo Manzini. Managing lumpy demand for aircraft spare parts. *Journal of Air Transport Management*, 11(6):426–431, 2005.

[175] Alberto Regattieri, R Manzini, and Daria Battini. Estimating reliability characteristics in the presence of censored data: A case study in a light commercial vehicle manufacturing system. *Reliability Engineering & System Safety*, 95(10):1093–1102, 2010.

[176] Alberto Regattieri, Giulio Santarelli, Riccardo Manzini, and Arrigo Pareschi. The impact of dwell point policy in an automated storage/retrieval system. *International Journal of Production Research*, 51(14):4336–4348, 2013.

[177] Kees Jan Roodbergen and Iris F.A. Vis. A survey of literature on automated storage and retrieval systems. *European journal of operational research*, 194(2):343–362, 2009.

[178] Meir J. Rosenblatt and Amit Eynan. Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Science*, 35(12):1519–1524, 1989.

[179] Olivier Roux, Mohamed Anouar Jamali, D Ait Kadi, and Eric Châtelet. Development of simulation and optimization platform to analyse maintenance policies performances for manufacturing systems. *International Journal of Computer Integrated Manufacturing*, 21(4):407–414, 2008.

[180] Jacob Rubinovitz and Gregory Levitin. Genetic algorithm for assembly line balancing. *International Journal of Production Economics*, 41(1):343–354, 1995.

[181] Nancy Ruiz, Adriana Giret, Vicente Botti, and Victor Feria. Agent-supported simulation environment for intelligent manufacturing and warehouse management systems. *International Journal of Production Research*, 49(5):1469–1482, 2011.

[182] Sébastien Rumley, E. Kagi, Hugh Rudnick, and Alain Germond. Multi-agent approach to electrical distribution networks control. In *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*, pages 575–580. IEEE, 2008.

[183] Jagdish S. Rustagi. *Optimization techniques in statistics*. Elsevier, 2014.

[184] Kwangyeol Ryu and Mooyoung Jung. Agent-based fractal architecture and modelling for developing distributed manufacturing systems. *International Journal of Production Research*, 41(17):4233–4255, 2003.

[185] Sameh M. Saad and Anna M. Lassila. Methodology for dynamic scheduling and control in biological manufacturing systems. *International Journal of Industrial and Systems Engineering*, 5(3):268–286, 2010.

[186] Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856, 2006.

[187] Luciano Sánchez, Inés Couso, and Jose A. Corrales. Combining gp operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 136(1):175–191, 2001.

[188] Pierre Schaus, Yves Deville, et al. A global constraint for bin-packing with precedences: Application to the assembly line balancing problem. volume 1, pages 369–374, 2008.

[189] Armin Scholl and Christian Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3):666–693, 2006.

[190] Leroy B. Schwarz, Stephen C. Graves, and Warren H. Hausman. Scheduling policies for automatic warehousing systems: simulation results. *AIIE transactions*, 10(3):260–270, 1978.

[191] Ishwar K. Sethi and Anil K. Jain. *Artificial neural networks and statistical pattern recognition: old and new connections*, volume 1. Elsevier, 2014.

[192] Pnina Shachaf. The paradox of expertise: is the wikipedia reference desk as good as your library? *Journal of Documentation*, 65(6):977–996, 2009.

[193] Weiming Shen, Lihui Wang, and Qi Hao. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(4):563–577, 2006.

[194] Herbert Alexander Simon. *Models of bounded rationality: Empirically grounded economic reason*, volume 3. MIT press, 1982.

[195] Barbara M. Smith and Anthony Wren. A bus crew scheduling system using a set covering formulation. *Transportation Research Part A: General*, 22(2):97–108, 1988.

[196] Jeffrey S. Smith, Erdal Sahin, and Levent Yilmaz. Agent-directed simulation for manufacturing system engineering. *Agent-Directed Simulation and Systems Engineering*, pages 451–473, 2009.

[197] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12):1104–1113, 1980.

[198] Josep M. Sola and Joaquin Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *Nuclear Science, IEEE Transactions on*, 44(3):1464–1468, 1997.

[199] Isaac W. Soro, Mustapha Nourelfath, and Daoud Aït-Kadi. Performance evaluation of multi-state degraded systems with minimal repairs and imperfect preventive maintenance. *Reliability Engineering & System Safety*, 95(2):65–69, 2010.

[200] Greg Stoddard. Popularity dynamics and intrinsic quality in reddit and hacker news. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[201] L. Suganthi, S. Iniyan, and Anand A. Samuel. Applications of fuzzy logic in renewable energy systems–a review. *Renewable and Sustainable Energy Reviews*, 48:585–607, 2015.

[202] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[203] John Symons and Fabio Boschetti. How computational models predict the behavior of complex systems. *Foundations of science*, 18(4):809–821, 2013.

[204] Salem M. Taboun and Sanjiwan D. Bhole. A simulator for an automated warehousing system. *Computers & industrial engineering*, 24(2):281–290, 1993.

[205] Seren Ozmehmet Tasan and Semra Tunali. A review of the current applications of genetic algorithms in assembly line balancing. *Journal of intelligent manufacturing*, 19(1):49–69, 2008.

[206] Joc Cing Tay and Nhu Binh Ho. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 54(3):453–473, 2008.

[207] Ahmet Teke, H Başak Yıldırım, and Özgür Çelik. Evaluation and performance comparison of different models for the estimation of solar radiation. *Renewable and Sustainable Energy Reviews*, 50:1097–1107, 2015.

[208] Lee-Ing Tong and Yi-Hui Liang. Forecasting field failure data for repairable systems using neural networks and sarima model. *International Journal of Quality & Reliability Management*, 22(4):410–420, 2005.

[209] Kristen Bernice Torrance, Ashley Raye Haire, and Randy B Machemehl. Vehicle and driver scheduling for public transit. Technical report, Southwest Region University Transportation Center, Center for Transportation Research, University of Texas at Austin, 2009.

[210] Jose Luis Torres, Almudena Garcia, Marian De Blas, and Adolfo De Francisco. Forecast of hourly average wind speed with arma models in navarre (spain). *Solar Energy*, 79(1):65–77, 2005.

[211] John Twidell and Tony Weir. *Renewable energy resources*. Routledge, 2015.

[212] Filippos S. Tymvios, Constantinos P. Jacovides, Silas C. Michaelides, and C. Scouteli. Comparative study of angstrom's and artificial neural networks' methodologies in estimating global solar radiation. *Solar energy*, 78(6):752–762, 2005.

[213] Jeroen P. Van den Berg. Analytic expressions for the optimal dwell point in an automated storage/retrieval system. *International Journal of Production Economics*, 76(1):13–25, 2002.

[214] José A. Ventura and Brian Q. Rieksts. Optimal location of dwell points in a single loop agv system with time restrictions on vehicle availability. *European Journal of Operational Research*, 192(1):93–104, 2009.

[215] Pedro M. Vilarinho and Ana Sofia Simaria. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations. *International Journal of Production Research*, 40(6):1405–1420, 2002.

[216] Christian Wagner, Sesia Zhao, Christoph Schneider, and Huaping Chen. The wisdom of reluctant crowds. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.

[217] Xin-Jing Wang, Xudong Tu, Dan Feng, and Lei Zhang. Ranking community answers by modeling question-answer relationships via analogical reasoning. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 179–186. ACM, 2009.

[218] Cor Warmer, Koen Kok, Stamatis Karnouskos, Anke Weidlich, David Nestle, Patrick Selzam, Jan Ringelstein, Aris Dimeas, and Stefan Drenkard. Web services for integration of smart houses in the smart grid. *Grid-Interop-The road to an interoperable grid, Denver, Colorado, USA*, pages 17–19, 2009.

[219] Andrew R Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.

[220] T. S. Wee and Michael J. Magazine. Assembly line balancing as generalized bin packing. *Operations Research Letters*, 1(2):56–58, 1982.

[221] Lillian Weng, Alessandro Flammini, Alessandro Vespignani, and Fillipo Menczer. Competition among memes in a world with limited attention. *Scientific reports*, 2, 2012.

[222] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[223] Anthony Wren. Heuristics ancient and modern: Transport scheduling through the ages. *Journal of Heuristics*, 4:87–100, 1998.

[224] Anthony Wren and David O. Wren. A genetic algorithm for public transport driver scheduling. *Computers & Operations Research*, 22(1):101–110, 1995.

[225] Kai Xu, Min Xie, Loon Ching Tang, and SL Ho. Application of neural networks in forecasting engine systems reliability. *Applied Soft Computing*, 2(4):255–268, 2003.

[226] Amit Kumar Yadav and S.S. Chandel. Solar radiation prediction using artificial neural network techniques: A review. *Renewable and Sustainable Energy Reviews*, 33:772–781, 2014.

[227] Ronald R. Yager and Dimitar P. Filev. Essentials of fuzzy modeling and control. *New York*, 1994.

[228] Candace Arai Yano and Ram Rachamadugu. Sequencing to minimize work overload in assembly lines with product options. *Management Science*, 37(5):572–586, 1991.

[229] L. Yi-Hui. Evolutionary neural network modeling for forecasting the field failure data of repairable systems. *Expert Systems with Applications*, 33(4):1090–1096, 2007.

[230] Enver Yücesan and John W Fowler. Simulation analysis of manufacturing and logistics systems. In *Encyclopedia of production and manufacturing management*, pages 687–697. Springer, 2000.

[231] Udny G. Yule. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213:21–87, 1925.

[232] David Z. Zhang. Towards theory building in agile manufacturing strategies–case studies of an agility taxonomy. *International Journal of Production Economics*, 131(1):303–312, 2011.

[233] David Z. Zhang and Anthony Ikechukwu Anosike. Modelling and simulation of dynamically integrated manufacturing systems. *Journal of Intelligent Manufacturing*, 23(6):2367–2382, 2012.

[234] G. Peter Zhang. Neural networks for time-series forecasting. In *Handbook of Natural Computing*, pages 461–477. Springer, 2012.

[235] Jiang Zhibin and Xie Chao. Multi-agent delay simulation model in mass rail transit system. In *Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on*, volume 3, pages 717–720. IEEE, 2009.

[236] Xu Zhoucong, He Peijia, Teng Jing, and Liu Liping. Transit vehicles intelligent scheduling optimization based on the division of characteristic periods. *Procedia-Social and Behavioral Sciences*, 96:1502–1512, 2013.

[237] Haiyi Zhu and Bernardo A Huberman. To switch or not to switch understanding social influence in online choices. *American Behavioral Scientist*, page 0002764214527089, 2014.

[238] Xingquan Zuo, Cheng Chen, Wei Tan, and MengChu Zhou. Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. *Intelligent Transportation Systems, IEEE Transactions on*, 16(2):1030–1041, 2015.