UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXVIII CICLO DEL DOTTORATO DI RICERCA IN

Ingegneria dell'Informazione

DECISION SUPPORT SYSTEM APPROACH FOR THE MANAGEMENT OF COMPLEX SYSTEMS IN TRANSPORTATION AND LOGISTICS

Settore scientifico-disciplinare: MAT/09, Ricerca Operativa

Dottoranda:

Monica CLEMENTE

Coordinatore:

Chiar.mo Prof. **Roberto VESCOVO**

Roberto Vescovo

Supervisore di Tesi:

Chiar.mo Prof. **Walter UKOVICH**

Chiar.ma Prof.ssa **Maria Pia FANTI**

Walter Ukovich

Maria Pia Fanti

Anno Accademico 2014-2015

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXVIII CICLO DEL DOTTORATO DI RICERCA IN

Ingegneria dell'Informazione

DECISION SUPPORT SYSTEM APPROACH FOR THE MANAGEMENT OF COMPLEX SYSTEMS IN TRANSPORTATION AND LOGISTICS

Settore scientifico-disciplinare: MAT/09, Ricerca Operativa

Dottoranda:

Monica CLEMENTE

Coordinatore:

Chiar.mo Prof. **Roberto VESCOVO**

Supervisore di Tesi:

Chiar.mo Prof. **Walter UKOVICH**

Chiar.ma Prof.ssa **Maria Pia FANTI**

Anno Accademico 2014-2015

Abstract

In recent years, the analysis and management of complex systems and their impacts in many aspects of the every-day life are topics that attract a lot of attention in the scientific literature. Consider for instance road and maritime transportation, modern healthcare systems, integrated supply chains, industrial processes or the new paradigm of smart cities: it is apparent that in all these contexts there is an increasing need of analysing and managing heterogeneous elements, networked together in order to reach a common goal otherwise not achievable. However, making decisions concerning such systems requires specific competences from many disciplines, leading to a very complex and often ineffective management process. Decision Support Systems (DSSs) can strengthen the capacity of predicting and controlling complex systems by integrating various sources of data and information, applying formal models typical of diverse and isolated disciplines and constantly interacting with the considered system.

The goal of this work is to define a general approach based on the DSS concept for the management of complex systems in transportation and logistics and to apply it to three problems of great interest nowadays: 1) the *user-based vehicle relocation problem* in Car Sharing services, 2) the *smart management of Electric Vehicles charging operations* and 3) the *container drayage problem*. In particular, the focus of the research is on the core of the DSS, i.e., on the part that directly supports the decision making process: optimization modules, simulation modules and their interactions. Different modelling, simulation and optimization approaches are ap-

plied, highlighting the generality of the considered approach regardless the specific context analysed.
Results show the ability of DSSs to enhance the effectiveness of the decision process, thus leading to an improvement of the considered systems performance. In particular: 1) the application of the DSS allows to optimize the set-points of an incentive policy designed to solve the vehicle relocation problem in Car Sharing services, guaranteeing an effective relocation and improving the system performance even in the case of nearly saturated offer; 2) the DSS allows the formalization of a leader-follower approach for the coordination of electric vehicles charging operations which takes into account simultaneously electric grid and drivers requirements; finally, 3) the DSS allows to improve the efficiency of drayage operations in container transportation, reducing total transportation costs.

Keywords: Decision Support Systems, Complex Systems, Car Sharing, Electric Vehicles, Container Drayage.

Riassunto espositivo

L'analisi e la gestione dei sistemi complessi e delle loro ripercussioni in diversi aspetti della vita quotidiana sono tematiche che continuano ad attrarre molta attenzione nella letteratura scientifica. Si considerino, ad esempio, il trasporto marittimo e su strada, i moderni sistemi di assistenza sanitaria, le catene di distribuzione integrate, i processi industriali o, ancora, il nuovo paradigma di città intelligente: è evidente come in tutti questi contesti vi sia sempre più la necessità di analizzare e gestire elementi eterogenei, collegati tra loro al fine di raggiungere un obiettivo comune altrimenti non realizzabile. Tuttavia, il processo decisionale in tali ambiti richiede competenze trasversali che abbracciano svariate discipline, rendendo la gestione di questi sistemi molto complessa e, spesso, inefficace. I Sistemi di Supporto alle Decisioni (DSS) ben si adattano alla previsione ed al controllo dei sistemi complessi grazie a: la loro capacità di integrare varie fonti di dati ed informazioni; l'applicazione di modelli formali tipici di diverse discipline; la possibilità di interagire costantemente con il sistema considerato.

L'obiettivo di questo lavoro di tesi è quello di definire un approccio generale basato sul concetto di DSS per la gestione di sistemi complessi nel settore dei trasporti e della logistica, e di applicare tale approccio a tre problemi di grande interesse oggigiorno: 1) il *problema della ricollocazione dei veicoli* nei servizi di car sharing, 2) la *gestione intelligente delle operazioni di carica dei veicoli elettrici* presso le infrastrutture pubbliche e 3) l'*ottimizzazione delle operazioni di drayage* nel trasporto container. In particolare, il focus della ricerca è rivolto al cuore del DSS, ovvero alla

parte che direttamente supporta il processo decisionale: i moduli di ottimizzazione e simulazione e le loro interazioni. Vengono considerati diversi approcci di modellazione, simulazione ed ottimizzazione, evidenziando il carattere totalmente generale dell' approccio considerato.
I risultati ottenuti nelle diverse applicazioni sottolineano l'efficacia dei DSS nel migliorare il processo decisionale, portando ad un miglioramento generale delle prestazioni dei sistemi in esame. In particolare: 1) l'applicazione del DSS permette di ottimizzare i set-point per l'introduzione di un sistema di incentivi economici atto a risolvere il problema di ricollocazione dei veicoli nei servizi di car sharing, garantendo un miglioramento delle prestazioni del sistema, anche in condizioni di quasi saturazione; 2) il DSS permette la formalizzazione di un approccio *leader-follower* per il coordinamento delle operazioni di ricarica di veicoli elettrici che tenga conto contemporaneamente sia dei requisiti dell'utente che quelli della rete elettrica; infine, 3) il DSS consente di migliorare l'efficienza delle operazioni di drayage nel trasporto containter, riducendo i costi di trasporto.

Parole chiave: Sistemi di Supporto alle Decisioni, Sistemi Complessi, Car Sharing, Veicoli Elettrici, Container Drayage.

Contents

*

List of Figures

List of Tables

Chapter 1

Introduction

The management of modern transportation and logistics systems requires addressing many challenging issues, among which:

1. the integration of heterogeneous and geographically distributed elements;
2. the interactions between different stakeholders, with different and competing objectives;
3. the need of applying methods and tools from several areas of expertise;
4. the necessity to deal with large amount of data made available in real time by the Information and Communication Technology (ICT) applications.

The cross-disciplinary character of planning problems in this context makes the decision making processes very complex, often straining decision makers capabilities and, thus, leading to several planning failures.

For these reasons, transportation and logistics systems are considered typical examples of complex systems, i.e., systems "comprised of a (usually large) number of (usually strongly) interacting elements, processes, or agents, the understanding of which requires the development, or the use of, new scientific tools, non-linear models, out-of equilibrium descriptions and computer simulation" (*Advance in Complex Systems* Journal).

An effective methodology for the study of complex systems should follow the principles of *system analysis*, i.e. [González et al., 2012]:

- identification of the main properties and parameters of the system;
- study of the interconnections among system components;
- study of the system interactions with the environment;
- system decomposition and partitioning;
- study of each system component by applying the approaches specific of the corresponding domain;
- composition of a whole model of the system.

However, additional issues have to be considered while dealing with complex systems, such as, the uncertainty of their nature, the emergent properties derived from the unpredictable interactions among their components, the heterogeneity of related information and the presence of humans as intelligent subsystems that form requirements and take decisions.

Therefore, the identification of new approaches for the analysis, prediction and control of complex systems is a topic that continues to attract a lot of attention, given also its repercussions in many aspects of every-day life.
Decision Support Systems (DSSs), i.e., "interactive computer-based systems, which help decision makers to utilize data and models to solve unstructured problems" [Gorry and Morton, 1971], represent a valuable solution in this context. Indeed, DSSs 1) allow to integrate, analyse and process huge amount of heterogeneous data; 2) apply mathematical models, simulations and optimization techniques to predict and

analyse the system behaviour, allowing to deal with the intrinsic uncertainty of the considered system; 3) can constantly interact with the system, guaranteeing responsive support to the decision makers.

This Thesis aims at identifying a general approach based on the DSS concept for the management of complex systems in transportation and logistics, and to apply it to three problems of great interest nowadays: 1) the *user-based vehicle relocation problem* in Car Sharing (CS) organizations, 2) the *smart management of electric vehicles (EVs) charging operations* and 3) the optimization of *container drayage operations*.

In particular, a model-based DSS with a modular structure, which simplifies the integration of the technologies typical of the specific subsystems involved, is considered. Three main components are taken into account: the *Data Component*, which handles all the data and information that the DSS needs to operate; the *Interface Component*, which constantly interacts with the real system; finally, the *Model Component*, which includes all the knowledge and the tools useful to provide support to the decision makers. All these components are composed by different separate sub-modules mutually interacting: this allows to deal with the complexity and the size of the considered systems.

Such DSS is applied to the three already mentioned transportation and logistics management problems, with particular focus on the core of the decision system, i.e., the Model Component and its sub-modules (the *Decision Module*, the *Simulation Module* and the *Optimization Module*). In particular:

1. the DSS is applied for handling the vehicle relocation problem in CS services. More in detail, a system of economic incentives ruled by a threshold policy is considered. To this aim, first the CS system is described in detail by Unified Modelling Language (UML) diagrams. Then, two implementations differen-

tiated by the assumptions made regarding the threshold determination are considered. First, an *a-priori set-point* strategy is considered (where the set-point is the minimum number of vehicle that should be available in a specific parking area) and a Timed Petri Net (TPN) framework is considered to analyse system behaviour in different operative scenarios. Then, the *optimized set-point* strategy is studied and the best threshold values for the incentives application are evaluated by discrete event simulation and Particle Swarm Optimization (PSO);

2. the requirements and specifications of the components and modules of a DSS devoted to handle the problem of EV charging operations management are analysed. Moreover, a leader-follower approach for the EV charging smart management with both drivers and electric grid requirements is formalized and a Mixed Integer Linear Programming (MILP) formulation for the vehicle-to-charging stations assignment problem is introduced;

3. a DSS to support truck managers in the assignment of container transportation orders to the available fleet of trucks is considered. More in detail, a MILP formulation for the multi-day drayage problem is proposed and a fast heuristic based on the rolling horizon approach is introduced.

The Thesis is organized as follows:

- Chapter 2 introduces complex systems and the advantages of applying a DSS approach in this context. Then, the general structure of the considered DSS for decision making in transportation and logistics is presented;

- Chapter 3 describes the application of the DSS approach to handle the user-based vehicle relocation problem in CS services;

- Chapter 4 deals with the smart management of EVs charging operations;

- Chapter 5 introduces the framework for the multi-day drayage operations optimization;
- finally, Conclusions summarize the main results and highlight the possible future research.

Chapter 2

Decision Support System for Complex Systems Management

In this Chapter, a general introduction about complex systems and their management challenges is presented, with particular attention to the logistic and transportation fields.

Then, the concept of DSS is introduced, together with the basic requirements that a computer tool has to meet in order to be classified as DSS.

Finally, the structure of the DSS considered in this Thesis is presented and the general approach for its application to the transportation and logistic problems addressed in this dissertation is outlined.

2.1 Complex Systems

A system is a collection of interacting elements, linked together into a unified whole by an internal structure.
Many systems surrounding us every day are inherently *complex* and their analysis requires competences and expertise from heterogeneous and sometimes isolated domains.

Even if there is no universally accepted definition of a complex system [Boccara, 2010], many authors agree on some common characteristics:

- they consist of a *large number* of interacting elements;
- their components are *heterogeneous* and span across different technological domains (structural complexity);
- their components are organized in a hierarchy of subsystems and strongly interact;
- they can exhibit *emergent properties*, i.e., self-organizing collective properties difficult to be predicted from the knowledge of the single components behaviours (dynamic complexity);
- they are permeated by *uncertainty*;
- *humans* can be part of the system, acting as intelligent subsystems that form requirements and take decisions.

Examples of complex systems can be find everywhere around us: air, road and maritime transportation, healthcare systems, integrated supply chains, financial systems, the world wide web, and so on.

Given the strategic relevance of such systems and their peculiar features, decision making processes in this context are becoming more and more challenging, imposing the need of new methods and tools.

Over the last decades, a particular type of complex system has gained a lot of attention in the scientific literature: the so-called "Systems of Systems" (SoSs), i.e., "large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal" [Jamshidi, 2011].
Maier [Maier, 1996] identifies five properties (also known as "Maier Criteria") characterizing SoSs:

1. *operational independence*, i.e., each subsystem is independent and it achieves its purpose by itself;
2. *managerial independence*, i.e., each subsystem is managed in large part for its own purposes than the purposes of the SoS;
3. *geographic distribution*, i.e., a SoS is distributed over a large geographic extent;
4. *emergent behaviour*, i.e., a SoS has capabilities and properties that do not reside in the component systems;
5. *evolutionary development*, i.e., a SoS evolves with time and experience.

SoSs find practical application in different fields of great interest nowadays, such as smart cities, automotive and aerospace applications, smart grids and health systems, but also traditional domains of research like transportation and logistics can benefit from the application of the SoS concept.

2.1.1 Transportation and Logistics as Complex Systems

The complexity of decision making processes in transportation and logistics has been historically widely recognized, and several examples of decision making failures in this context can be observed every day: contrasted transport infrastructures, ineffective road and parking pricing schemes, congestioned pubblic transport services, unrealistic business plans for freight transport companies, and so on [Cascetta et al., 2015].

This field of management encompasses a large number of decisions. For example, at the *strategic level* typical problems to be solved are the design of the logistic network, the determination of the facilities locations, the fleet sizing, the planning of the transit routes in order to meet passengers demand, and so on. Typical issues with which decision makers have to deal with at the *tactical level* are inventory policies, production and distribution planning, transportation modes selection, service schedule and timetabling, etc. Finally, classical issues of the *operational level* are operations scheduling, shipment and vehicles dispatching, crew scheduling and so on.

All these problems have been widely studied and treated separately in literature for decades [Barnhart and Laporte, 2006] and, in recent years, great attention has been paid to the development of integrated frameworks able to support management simultaneously at the strategic, tactical and operational levels [Manzini and Bindi, 2009], in order to identify global optimal policies, thus leading to big savings and important economic effects.
Moreover,

- the diffusion of alternative transport solutions in response to the pressing need of reducing pollutant emissions in urban areas and of alleviating the reliance of mobility on fossil fuels;
- the need to face the increasing demand of transport (both in passenger and freight transport);
- the need of responsive systems;
- the need of coordination between modes and transportation companies;
- the extensive application of ICT solutions, which makes available large amount of heterogeneous data in real-time;
- the need of coordinating different stakeholders, with different and competing objectives (e.g., private industries and governments)

impose the research of new quantitative analyses and mathematical tools, which allow the integration of expertise and competences from different domain, thus supporting participated decision making process and improving the quality of the management.

2.1.2 Complex Systems Management Challenges

The very essence of a complex system does not rely in its single components, but in the relationships and interactions between them. For this reason, decision making concerning complex systems often challenges human cognitive capabilities, leading to ineffective management processes.

In order to understand, control and predict the behaviour of a complex system, a decision maker should deal with:

- notions that cross several domains and areas of expertise;
- large amount of data and information made available in real-time;
- the non-linear and non predictable interactions between the system components;
- the need of guaranteeing the fulfilment of different, and often competing, objectives and priorities.

In recent years, many researchers have been addressing the problem of identifying new approaches for the analysis, prediction and control of complex systems and the following needs are generally identified [González et al., 2012]:

- development of *conceptual models* able to capture the most important processes and interactions between the different components of the system: in particular, such models have to be sufficiently *general* to encompass a wide range of possible scenarios, but, at the same time, sufficiently *structured* to catch the essential features of the system;
- development of *algorithmic and mathematical models*, in order to analyse each component of the system;
- identification and structuring of apt *simulation scenarios*, in order to check the behavioural adequacy of the developed models to the real system;

- extensive use of *expert knowledge and information* about the system, stored in data and information systems.

Many methods have been proposed in the last decades for managing and controlling complex systems, such as hierarchical control and optimization, decentralized control, perturbation-based techniques, artificial intelligence-based techniques, and so on [Filip, 2008]: most of them are based on the application of non-linear mathematical models, statistical methods and computer modelling approaches.

In particular, many authors underline the great importance of *computer simulations* in this context, since they can assist in either a static or a dynamic analysis of the considered system.

Although the advances in the implementation of totally automated systems, it is not always reasonable to completely exclude the human being from the decision process. Thus, the application of Decision Support Systems in the context of complex systems management represents a valuable solution [Elam et al., 1980], [Cats-Baril and Huber, 1987], [Mallach, 2000], [Charbonnier et al., 2005].

2.2 Decision Support Systems Background

The concept of *Decision Support System* was first introduced in the early 1970s as an evolution of the theoretical studies of organizational decision making done at the Carnegie Institute of Technology during the late 1950s and early 1960s and the technical work on interactive computer systems mainly carried out at the Massachusetts Institute of Technology in the 1960s [Keen and Morton, 1978]. According to [Sprague Jr and Watson, 1996], DSS became a field of study and practice during the 1980s and, since then, such concept has been growing and evolving.

As stated in [Turban et al., 2007], there is still no universally accepted definition of what a DSS is: with the ever-increasing advances in computer technology, different definitions of DSS have been given over the years.

[Gorry and Morton, 1971] states that DSSs are "interactive computer-based systems, which help decision makers to utilize data and models to solve unstructured problems".

According to [Keen and Morton, 1978], DSSs "couple the intellectual resources of individuals with the capabilities of the computer to improve the quality of decisions. A DSS is a computer-based support system for management decision makers who deal with semi-structured problems".

In [Mann and Watson, 1984], a DSS is defined as "an interactive system that provides the user with easy access to decision models and data in order to support semi-structured and unstructured decision-making task".

Bidgoli [Bidgoli, 1989] defines a DSS as "a computer-based information system consisting of hardware/software and the human element designed to assist any decision maker at any level".

Finally, for [Turban et al., 2007] DSS is "an umbrella term to describe any computerized system that supports decision making in an organization".

As there is no consensus about the definition of what a DSS is, there is no consensus on its standard characteristics and abilities [Turban et al., 2007]. Nevertheless, it is possible to highlight some essential features on which many authors agree ([Alter, 1980], [Keen and Morton, 1978], [Wallach, 1993], [Turban et al., 2007]). In particular, DSSs:

- are *computer-based* systems;
- are designed specifically to *facilitate* decision processes;
- should *support rather than automate* decision making, and are used *actively* by decision makers;
- should be able to *respond quickly* to the changing needs of decision makers;

- can support decision makers at *any level* of an organization;
- can support decision makers in *all* the phases of the *decision making process*;
- are intended for *repeated use*;
- are *flexible* and can be easily adapted to solve problems similar to the ones for which they have been originally designed;
- improve the *effectiveness* of decision making;
- can *support learning* and improving decision makers skills;
- include a body of *knowledge* that describes some aspect of the decision makers expertise.

2.2.1 Categories of DSSs

Various classifications for the different types of DSSs can be found in literature.

The *expanded DSS framework* developed in [Power, 2002] helps researchers and managers in understanding and categorizing DSSs by specifying a primary technology dimension and 3 secondary dimensions: in particular, the primary dimension is the *dominant technology component* that provides the functionality for decision support; the three secondary dimensions are the *targeted users* (internal or external), the *purpose* of the system (general or specific) and the *enabling technology* used to implement the DSS (client/server, web, stand-alone PC).
Five main categories can be identified by analysing the dominant technology component of the DSS: *communications-driven*, *data-driven*, *document-driven*, *knowledge-driven* and *model-driven* DSSs.

Communications-driven DSSs derive their functionalities from communications and information technologies and emphasis is given to collaboration and shared decision-making.

Data-driven DSSs emphasize access to and manipulation of a time-series of internal, and sometimes external, company data: in such a way, such tools allow users to extract useful information previously buried in large quantities of data. This category of DSS includes file-drawer and management reporting systems, data warehousing and analysis systems, Executive Information Systems (EIS) and data-driven Spatial DSS.

Document-driven DSSs are focused on the retrieval and management of unstructured documents and web pages. Almost all text-driven DSSs fall in this category.

Knowledge-driven DSSs make use of Artificial Intelligence (AI) and rules for automated decision making, and are also called *expert systems*. This category of DSS has specialized problem-solving expertise relevant to a specific task.

Finally, model-driven DSSs use models from various disciplines to provide decision support (e.g., algebraic models, decision analytic models, financial models, simulation and optimization models, and so on). The DSS structure core is therefore represented by the access and manipulation of models, rather than data.

2.2.2 General DSS Architecture

Different authors identify different components in a DSS.

In particular, [Sage, 1991] identifies three fundamental components:

1. the *database management system* (DBMS), which separates users from the physical aspects of the database, where the (large quantity) of data relevant to the class of problems for which the DSS is designed are stored;

2. the *model-base management system* (MBMS), whose primary purposes is to

transform the data from the DBMS into valuable *information* and to guarantee the independence between the models used in the DSS from the applications that use them;

3. the *dialog generation and management system* (DGMS), an easy-to-use interface which enhances the ability of the decision makers to utilize and interact with the DSS.

According to [Holsapple, 2008], a DSS has four essential components:

1. a *language system* (LS), which consists of all the messages that the DSS can accept;
2. a *presentation system* (PS), which consists of all the messages that the DSS can emit;
3. a *knowledge system* (KS), consisting of all knowledge the DSS has stored and retained;
4. a *problem-processing system* (PPS), the DSS core software engine, which tries to recognize and solve problems during the making of a decision.

[Power, 2002] identifies again four major components for a DSS: the *user interface*, the *database*, the *models and analytical tools* and the *DSS architecture and network*.

Other authors propose other general structures, but, summing up, it is possible to say that a DSS should include the following general components [Mallach, 1994]:

1. databases;
2. models;
3. software tools to allow users to access the databases and the models;

4. hardware and operating system platforms on which databases and models reside;

5. network and communication capabilities to connect the hardware platforms.

2.3 Decision Support System for Complex Systems in Transportation Logistics

A DSS could handle all the management challenges associated to decision making concerning complex systems by:

- **integrating**, **analysing** and **processing** the huge amount of heterogeneous data made available by the continuous monitoring and control of the system performed by several distributed computing elements;

- applying **mathematical models**, **simulations** and **optimization techniques** in order to obtain a powerful connection and synchronization among all the systems involved. In particular, simulation approaches are fundamental to deal with the emergent properties derived from the interactions among system components;

- **constantly interacting** with the system, stressing all the possible future critical situations and proposing effective solutions in a proactive approach.

This Section describes the general structure of the DSS considered in this dissertation, together with the approach followed for its application in different contexts of great interest nowadays.

2.3.1 DSS Structure

Given the necessity of formal modelling and simulation approaches enhanced in the previous Sections, a **model-based** DSS is considered [Clemente et al., 2016b], [Fanti et al., 2015].

In particular, such DSS should consist of three main components [Turban et al., 2007]: the *Data Component* handles all the data and information that the DSS needs to operate; the *Interface Component* interacts with the real system by means of a set of geographically distributed communication modules based on ICT and maintains the consistency between the models contained in the DSS and the real system; finally, the *Model Component* includes all the knowledge and the tools useful to provide support to the decision makers.
All these components are composed by different separate sub-modules mutually interacting: this deals with the complexity and the size of the considered systems.

2.3.1.1 DSS Model Component

Even if both the Data Component and the Interface Component are fundamental to guarantee the accuracy and the effectiveness of the DSS, the core of the support system is represented by the Model Component. Given that the specification of the number and the features of the sub-modules included in the Model Component depends on the particular case considered, all the possible modules belong to three main typologies: *Simulation Modules*, *Optimization Modules* and *Decision Modules*. In the following, the typologies of the Model Component sub-modules, together with the interactions among them, are described.

Simulation Modules. Simulation is useful for three different purposes in a DSS framework: as *data generator* for the Optimization Modules; as tools to perform *what-if* analyses to evaluate hypothetical scenarios; finally, to evaluate the values of particularly complex objective functions in a *simulation-optimization approach*. On the basis of the time scale at which the decisions have to be taken and the characteristics of the system that has to be represented, different types of simulation approaches can be considered. For example, in Chapter 3 both a Timed Petri Net framework and discrete event simulation are exploited to represent a generic CS system and solve the vehicle relocation problem.

Optimization Modules. The role of the Optimization Modules is to provide to the end-users of the DSS effective indications of behaviour (*how-to* analyses). In particular, on the basis of the current state of the system or the future conditions foreseen by the Simulation Modules, the optimizator has to identify the best operating rules for the component it is referred to (for example, in the case of a public transportation company, it has to determine the optimal fleet size, the routing of the vehicles, timetabling, crew scheduling, and so on).
A closed-loop approach is considered and each solution proposed by the Optimization Modules has to be validate, in terms of expected global system performances, by the Simulation Modules: indeed, the unpredictable emergent properties have to be assessed in order to obtain an effective coordination and integration among all the actors and subsystems involved.

Decision Modules. The Decision Modules can operate in two basic ways, determining thereby the operating mode of the entire DSS.
When the *on-line* application is considered, the Decision Modules receive the current system state from the Data Components and, on the basis of it, determine if it is necessary to trigger a new simulation-optimization procedure in order to optimize in real time the overall system performances.
On the other side, in the *off-line* approach, the performances evaluated by the Decision Modules do not depend on the real-time events occurring in the system, but on a set of hypothetical scenarios.

In order to evaluate the effectiveness of the proposed solutions or the necessity of a new simulation-optimization campaign, a set of Key Performance Indicators

(KPIs) are considered: such indexes have to be defined together with the decision makers and represent their knowledge of the particular problem considered.

2.3.2 DSS Approach

The particular applications considered in this dissertation make the necessary DSSs become very complicated. For this reason, the DSSs modelling requires a precise and structured approach.

In the following, the approach considered in this work, made up of three phases, is outlined.

- First, a detailed analysis of the problem has to be carried out, with the aim of identifying the structure and the evolution rules of the overall system, the role of each component, as well as the flow of information and the required data. With this objective, a top-down metamodeling technique based on the application of UML is considered [Booch et al., 1999], [Boschian et al., 2011]. The choice of using such a graphical and textual formalism is due to: the standard notation easily adaptable to a great variety of systems; the modularity, essential to clearly define the tasks and the interactions of each actor of the system; the versatility, i.e., the possibility of considering different levels of details for the different parts of the system using different types of diagrams; the readability and the compactness; finally, the possibility of a straightforward translation into several mathematical and simulation models. Moreover, first the structural (static) aspects of the considered system are modelled through *class* diagrams; hence, the evolution rules (behavioural description) are formalized through the *activity* diagrams, in which it is clearly pointed out which actor is responsible for which action.

- Second, on the basis of the systematic model obtained during the first phase, the simulation models have to be developed. In particular, in this dissertation

discrete-event simulation is considered, since it allows to represent large-scale systems in a efficient way.
Moreover, in this phase the KPIs necessary to evaluate the effects of each action have to be identified.

- Third, the suitable optimization procedures have to be defined on the basis of the particular problem to be addressed and the operative conditions in which the DSS is used (off-line or on-line).

In the next three Chapters, the described approach is applied to develop the DSSs for three different logistics and transportation problems of great interest nowadays.

In particular, in Chapter 3, all the three phases of the approach are described in detail in order to define a DSS for the user-based vehicle relocation problem in CS services. Chapter 4 and Chapter 5 detail the most significant phases regarding the specific context considered. More in detail, in Chapter 4, phases 2 and 3 are described to assess the EVs charging operations smart management problem, while in Chapter 5, phases 1 and 3 are detailed to deal with the container drayage optimization problem.

Chapter 3

User-Based Vehicle Relocation in Car Sharing Systems

In this Chapter, the user-based vehicle relocation problem in Car Sharing services is addressed.

3.1 Motivation

Over the past few decades, the environmental and socio-economic problems linked to the mobility in urban areas have underlined the need of reducing the massive use of private vehicles. In this context, systems in which a common fleet of vehicles is shared among multiple users (the so-called shared-use vehicle systems) have reached great popularity [Barth and Shaheen, 2002]. In particular, Car Sharing (CS) solutions are nowadays widely spread throughout the world: in such a kind of systems a car is used as a public transport means but individually, and every user can autonomously rent a car according to his needs and for a period that can be very short, unlike the traditional car rental.

The importance of CS in the current urban mobility could be strategic and two leading motivations to invest in its deployment can be pointed out. First of all, a CS system could induce a general improvement of urban transport efficiency, thanks to the decrease of the total number of vehicles required to meet the travel demand and to a more rational use of the mobility alternatives. At the same time, CS services represent an effective opportunity to support the diffusion of Electric Vehicles (EVs). EVs could give a concrete contribution to the decrease of the air pollution in urban areas, but high purchase prices and limited driving range severely hinder their popularity. The deployment of the EVs in the CS services fleets could be useful to overcome these drawbacks, since this solution allows to share the fixed costs of owning a vehicle and the typical distances travelled during a rental are compatible with the EVs driving ranges.

Nevertheless, in this kind of services it is fundamental to reach an overall level of efficiency as to make them effectively competitive with the ownership of a private vehicle. However, the continuous dynamic reconfiguration of the system during its operation and the coexistence of different and often competing objectives make the management of CS services a very complex process. In this context, the application of the modern ICT is essential [Barth et al., 2003].

As in other areas of applications, the decisions that the management of a CS service has to deal with can be classified into three hierarchical level. The *strategic level* includes parameters with a long-term impact on the system performances, such as: the *number* and the *location* of the *parking areas* [Nakayama et al., 2002]; the *number* of *EVs chargers* [George and Xia, 2011], in case of EVs ; the *optimal fleet size* [George and Xia, 2011]; the *fleet composition* (i.e., the number of EVs and the number of traditional vehicles composing the fleet). The *tactical level* is related to the mid term decisions, as: the *pricing policies*; the *service access rules* (only upon reservation or also on demand); the *rental rules* [Barth and Shaheen, 2002]. Finally, the *operational level* refers to the real time management of the system and, therefore, to aspects such as the *maintenance* of the vehicles, the *emergencies* management and the maintenance of the *coherent distribution of the vehicles* among the parking areas of the system.

In particular, rental rules play a central role in determining the attractiveness of a generic CS organization. If a so-called *two-way rental* system is deployed, only round trips are possible: therefore, the number of vehicles in each CS parking area is constant, but the flexibility of the customer travels is limited. On the other side, in a *one-way rental* system users are allowed to pick up and return the rented vehicle in different parking areas, but the distribution of the vehicles can become imbalanced during the day due to the non-uniform demand [Barth and Shaheen, 2002]. Therefore, in this case, vehicle relocation activities are necessary to satisfy users requests at any time. For this reason, an important CS management problem is the so-called *vehicle relocation problem* [Cepolina and Farina, 2012].

3.1.1 Vehicle Relocation Problem Background

Several approaches to the vehicle relocation problem are studied in the related literature, and it is possible to categorize them on the basis of four fundamental factors.

First, the main actors of the relocation activities are considered: *operator-based* and *user-based* strategies. In an operator-based strategy, system staff relocates the vehicles in the parking areas when needed, but in this case additional trips without customers are necessary [Barth and Todd, 1999], [Kek et al., 2009], [Jorge et al., 2014], [Boyacı et al., 2015], [Bruglieri et al., 2014], [Alfian et al., 2014], [Nourinejad and Roorda, 2014]. On the other hand, in a user-based relocation approach, users themselves ensure the rebalancing of the system with their travel behaviour, conveniently influenced through different types of incentives [Uesugi et al., 2007], [Bianchessi et al., 2013], [Di Febbraro et al., 2012]. From both an economic and an environmental point of view, the second solution is preferable.

Second, it is possible to characterise the relocation strategies on the basis of the approach used to determine the timing and the configuration of such activities. If an *off-line* method is considered, relocation activities are performed at a fixed time regardless of the actual system balance conditions (e. g., at the end of the working day). On the other hand, when a *real time* monitoring of the system is implemented, relocations are performed as soon as a critical situation occurs.

Third, if relocation events are triggered only when an established minimum (or maximum) threshold in a parking area is reached, then a *non-predictive* relocation method is considered. However, if relocations are based on the expected future demand, a *predictive* relocation approach is carried out [Barth and Todd, 1999].

Fourth, a strategic parameter of the relocation strategies is the desired number of vehicles in each parking area. Papers regarding operator-based techniques usually determine the optimal values of such *set-points*, as they can directly control the relocation operations [Nourinejad and Roorda, 2014], [Kek et al., 2009]. On the other hand, several papers addressing user-based techniques do not consider a methodology for determining the set-points, but take as reference the mean number of vehicles available in the system [Di Febbraro et al., 2012], [Bianchessi et al., 2013].

Table 3.1: Literature Review Classification.

Model / Reference	Relocation Modality		Relocation Time		Relocation Control		Set-point	
	Operator	User	Offline	Real time	Non-predictive	Predictive	A-priori	Optimized
[Barth and Todd, 1999]	*			*	*	*	*	
[Kek et al., 2009]	*			*	*			*
[Jorge et al., 2014]	*			*		*		*
[Boyacı et al., 2015]	*			*	*			*
[Bruglieri et al., 2014]	*			*	*		*	
[Uesugi et al., 2007]		*		*	*		*	
[Bianchessi et al., 2013]		*		*	*		*	
[Di Febbraro et al., 2012]		*		*	*		*	
[Nourinejad and Roorda, 2014]	*			*	*			*

Table 3.1 classifies the works related to the relocation problem approaches according to the four mentioned factors: relocation modality (*operator-based* vs. *user-based*), relocation time (*offline* vs. *real time*), relocation control (*non-predictive* vs. *predictive*) and type of set-point (*a-priori* vs. *optimized*). As Table 3.1 shows, few contributions deal with the user-based relocation approach and typically the authors apply such a strategy by using an a-priori determined set-point. Hence, investigating about methodologies to evaluate and implement the optimal values of the vehicle thresholds for the user-based relocation is an open problem.

Moreover, due to the complexity of the vehicle relocation problem, some authors propose a DSS approach, and a combination of optimization and simulation is applied: most of them focus on the operator-based relocation strategy. In particular, [Kek et al., 2009] introduces a 3-phase optimization-trend-simulation DSS to identify a set of near optimal operating parameters for the operator-based vehicle relocation problem. In [Nourinejad and Roorda, 2014] a dynamic optimization-simulation model for one-way CS organization with operator-based relocation is introduced and the optimization model is solved successively in a discrete event simulation. In both these works the simulation is used to perform *what-if* analyses after having optimized the system parameters, i.e., to evaluate the effectiveness of the optimal solutions already identified by the optimization models. In [Nourinejad and Roorda, 2015] two integer programming models are proposed for strategic and operational decision making in both two-way CS systems and one-way systems with operator-based relocation: a Monte Carlo simulation is set up in order to obtain the required input data for the optimization. It must be pointed out that [Kek et al., 2009], [Nourinejad and Roorda, 2014] and [Nourinejad and Roorda, 2015] do not consider the customers decision process in the proposed approaches, hence the objective functions of their optimization models do not strictly depend on the human behaviour. Conversely, the optimization of the set-points for the user-based relocation imposes to take into account the difficult tasks of considering the stochastic human behaviour and the urban and population models.

3.2 Objectives

As highlighted in the previous Section, the vehicle relocation problem has attracted the attention of several authors, but there are still open problems to be addressed, in particular regarding the user-based strategies and the determination of the optimal set-points to implement them.

This Chapter deals with the user-based vehicle relocation problem and presents a model-based DSS to solve it. The considered vehicle relocation strategy has the objective of encouraging the customer to return the car where and when it is mostly needed. With the aim of determining when the incentive has to be applied, a threshold policy similar to the one considered in the classical *stochastic inventory problem* [Porteus, 2002] is defined: if the number of vehicles in the parking area is minor of a threshold, then the incentive is applied for such a CS parking area. In this Chapter, the focus of the analysis is on the *type of set-point* considered for the application of the threshold policy. In particular, two different implementations are considered, thereby determining two different operative conditions for the DSS. First, an *a-priori set-point* strategy is considered: in this case, the Simulation Module of the DSS is exploited to perform *what-if* analyses in different operative scenarios and evaluate the effectiveness of the proposed relocation strategy. Then, the *optimized set-point* strategy is studied and the DSS is used to determine the minimum numbers of the vehicles necessary in each parking area on the basis of the state of the system (i.e., the number of vehicles and the customers waiting for an available vehicle): to this aim, a procedure based on discrete event simulation and Particle Swarm Optimization (PSO) is proposed.

The remainder of the Chapter is organized as follows: Section 3.3 presents the structure of the DSS according to the general scheme introduced in Chapter 2 and formalizes the structural and behavioural aspects of the CS problem through the UML framework. Section 3.4 studies the *a-priori set-point* case, while Section 3.5 analyses the *optimized set-point* case: in both these Sections, first the considered problem and assumptions are formalised, then the DSS modules are specified and, finally, different operative scenarios are studied in order to evaluate the effectiveness of the proposed solutions. Section 3.5 summarizes the remarks and the contributions of the present Chapter.

3.3 DSS for the CS Problem

3.3.1 DSS Formalization

Fig. 3.1 shows the proposed DSS components and modules and the two main actors with which it interacts: the *CS system* that includes the set of the parking areas and vehicles, and the *decision maker*, i.e., the park manager. The green arrows represent the information flow among the DSS components and modules, while the red arrows depict the information flow between the DSS, the real system and the park manager.

The inputs of the *decision module* are the data characterizing the system (e.g., the number of available vehicles in each parking area, the number of customers waiting for a vehicle, etc.) collected through the *interface component* and provided by the *data component*; the outputs are the objects of the decisions. Depending on the role considered for the DSS, the performances evaluated by the *decision module* depend on historical data and hypothetical scenarios (see the a-priori set point application) or on the real-time events occurring in the system.

The proposed DSS is designed following the three-phase approach described in Chapter 2.

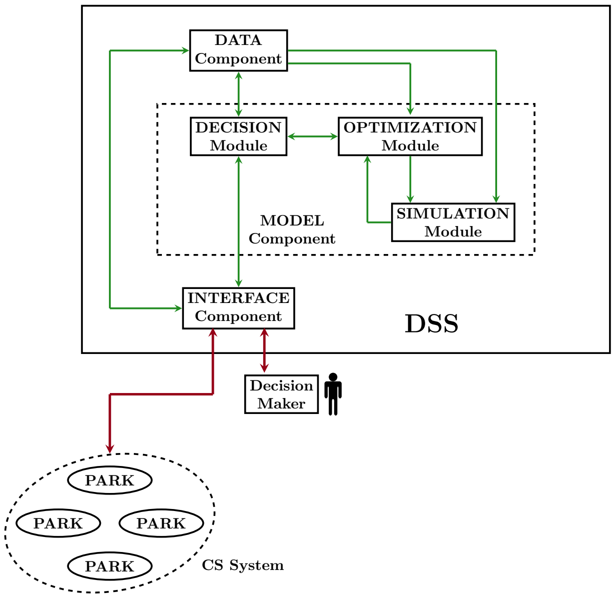

Figure 3.1: The Decision Support System architecture and the connections with the CS System and the Decision Maker.

- First, a detailed analysis of the problem is carried out: with the aim of identifying the structure and the behavior of the overall system, the role of each component, the flow of information as well as the required data, a top-down modeling technique based on the application of UML is considered [Booch et al., 1999]. Moreover, the structural and behavioural aspects of the considered system are modelled by *class* diagrams and *activity* diagrams, respectively.

- Second, on the basis of the UML description, the simulation and the decision modules of the DSS model component are developed.

- Third, the optimization module and its interactions with the simulation module are defined (in particular, for the optimized set-point problem).

In the following the first phase is described, while the second and the third phases are specified separately for the a-priori set-point problem and the optimized set-point problem.

3.3.2 Car Sharing System Description

As described in Chapter 2, the first phase of the development of the proposed DSS is described by considering two aspects: i) the structural description; ii) the behavioural description.

3.3.2.1 Structural description

In Fig. 3.2 the class diagram representing the structure of a generic CS service is depicted [Clemente et al., 2013a], [Clemente et al., 2013b], [Clemente et al., 2015], [Clemente et al., 2016b]: all the involved actor categories are represented with their main attributes, operations and relationships. The values of the attributes of these classes determine the specific CS organization. The structure highlighted by the class diagram determines the requirements of the *data component* of the DSS.

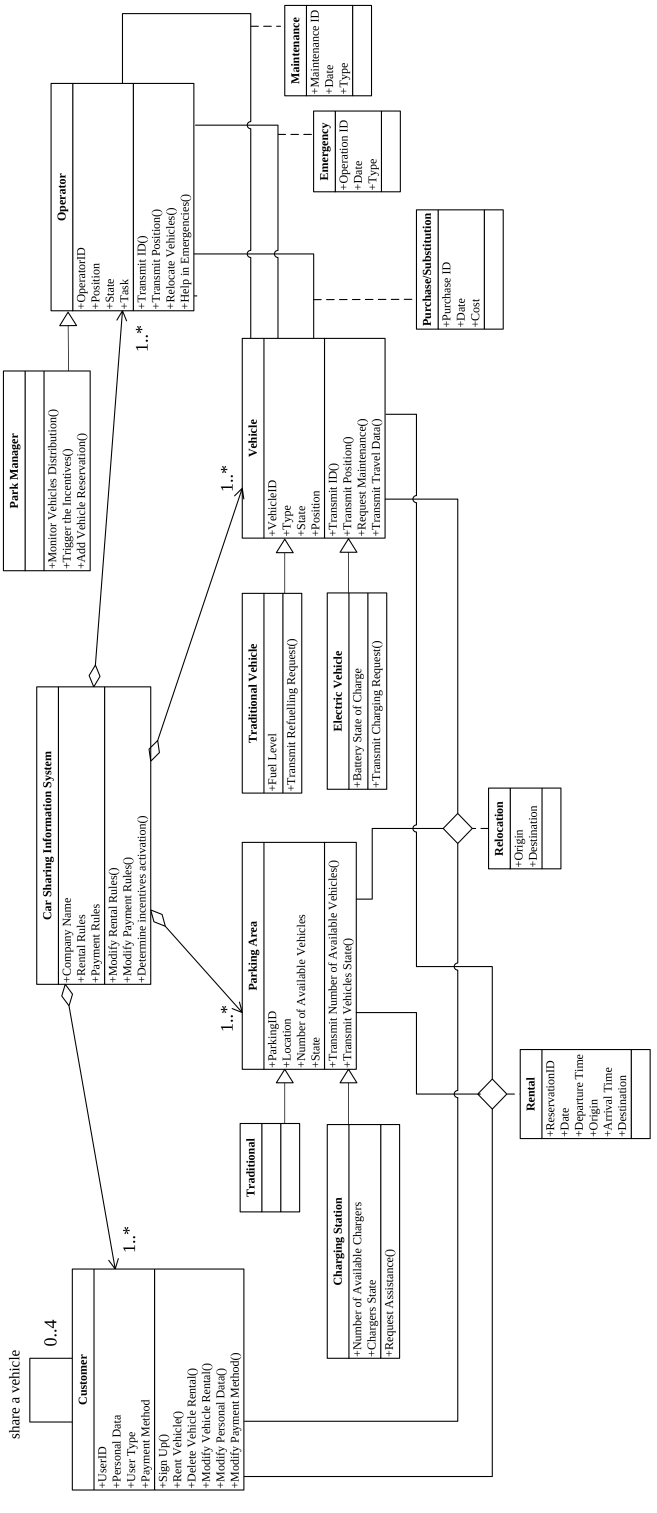

Figure 3.2: The CS service class diagram pointing out the main components of a generic CS system and their relationships: *Customer*, *Parking Area*, *Vehicle*, *Operator*, *CS Information System.*

3.3.2.2 Behavioural Description

Fig. 3.3 describes the vehicle rental process of a user after the introduction of the static real-time user-based relocation strategy [Clemente et al., 2013a], [Clemente et al., 2013b], [Clemente et al., 2015], [Clemente et al., 2016b]. Such diagram is the base of the implemented simulation module. Three actors are involved in this case: the "Customer" represents the generic service user; the "Vehicle" represents the generic vehicle of the CS fleet; and the "CS Information System" is the centralized information system of the CS. Six phases characterize this process:

1. the "vehicle request" phase, representing the user arrival, request of a vehicle and waiting. After a maximum waiting time, the user leaves the system without being served;

2. the "checking vehicle availability" phase, during which the CS information system checks the vehicles availability and, if there is a car not yet rented in the considered parking area, it grants the hire;

3. the "incentive determination" phase, during which the CS information system determines the state of activation of the incentives for the different parking areas and communicates it to the customer;

4. the "rental and use of the vehicle" phase, when the customer refines the rental of the vehicle and makes his trip. In particular, during this phase the customer chooses his destination and, if there are active incentives, the customer can accept or not the received incentive.

5. the "vehicle restitution" phase, during which the customer drops off the vehicle in one of the parking areas of the service and leaves the system;

6. the "maintenance" phase, which occurs only when the vehicle needs a repair service before being again available for rental or, in case of EV, if the vehicle

needs to be recharged. Only after this phase the vehicle is again available for rental.

As can be seen in Fig. 3.3, it is reasonably assumed that a user is willing to wait for a limited time interval for an available vehicle and, if she/he can not manage to rent a vehicle after such time interval, she/he leaves the system without being served.
Moreover, the activities “destination park selection” and “rental duration selection” have to be interpreted as simple schematization of the users decision process, and not as preventive declarations of behaviour made by customers to the CS information system.

3.4 A-priori Set-point Problem

3.4.1 Problem Statement

In this Section the following problem is considered [Clemente et al., 2013c], [Clemente et al., 2013a], [Clemente et al., 2013b], [Clemente et al., 2015].

1. **Class of user-based vehicle relocation problem**.

    - *relocation modality*: the relocation activities are performed only by the users during the service hours, while at the end of the working day system staff relocates the vehicles (*user-based*);
    - *relocation time*: the system status is monitored at regular intervals throughout the day (*real time*);
    - *relocation control*: the incentive for a parking area is triggered only when a minimum number of available vehicles is reached (*non-predictive*);
    - *set-point*: the minimum number of vehicles that should be available in each parking area is fixed (*a-priori set-point*).

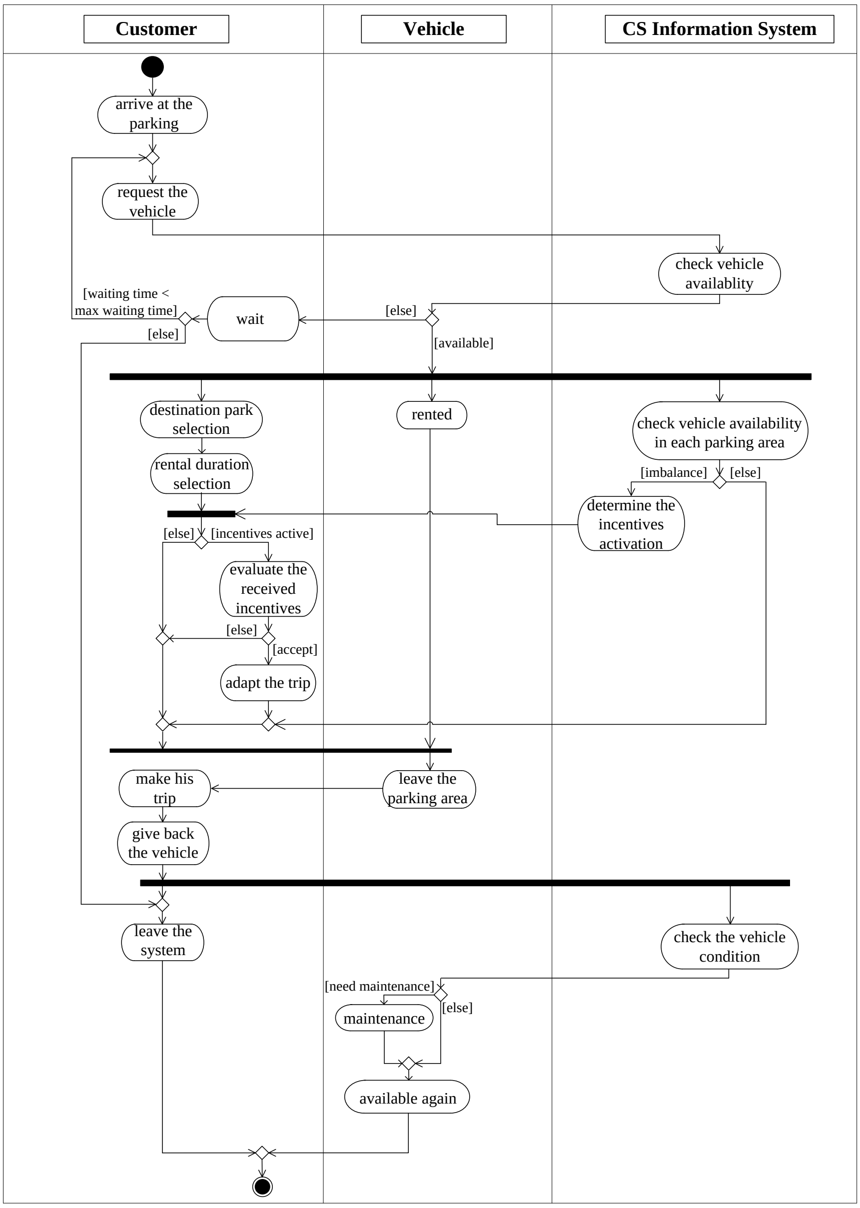

Figure 3.3: The vehicle rental process activity diagram.

2. **Relocation strategy rules**.

   - *incentive notification*: users receive the notification about possible incentives at the beginning of the rental period (before the effective usage of the car), therefore no particular equipment has to be installed on board the vehicles.
   - *types of incentive*: two main mutually exclusive types of incentive are considered, the *time-incentive* and the *destination-incentive*. The time-incentive is triggered when in *all* the parking areas of the CS system the number of available vehicles is less or equal the considered set-point: in this case, all the users are encouraged to return the rented vehicle as soon as possible. On the other side, if the time-incentive is not triggered, the destination-incentive is active for the parking areas with vehicles shortage, i.e., for the parking areas with a number of available vehicles lower than the defined set-point.

3. **Role of the DSS**.
   The DSS is used to provide suggestions about the implementation of the considered relocation strategy by evaluating different operative scenarios. Therefore, the model component modules considered in this application are the decision module and the simulation module.

3.4.2 DSS Modules Specification

3.4.2.1 Simulation Module Specification

In order to assess the impact of the considered relocation strategy, a model of a generic CS service is developed in a Timed Petri Net (TPN) framework [Clemente et al., 2013c]. Indeed, TPN allows concisely representing in an unified structure both static and dynamic aspects of the considered system, thanks to its twofold representation, graphical and mathematical. In particular, the graphical aspect enables a concise way to design and verify the model, while the mathematical description allows simulating the considered system in software environments, by considering different dynamic conditions. In the following, some basic definition on the TPN formalism are recalled.

Basics of Timed Petri Nets

- **Net Structure**

  A Petri net (PN) [Peterson, 1981] is a bipartite digraph described by the four-tuple $PN = (P, T, \mathbf{Pre}, \mathbf{Post})$, where P, T, $\mathbf{Pre}$, $\mathbf{Post}$ are defined as follows.

  1. P is a set of places with $|P| = m \in \mathbb{N}$.
  2. T is a set of transitions with $|T| = n \in \mathbb{N}$.
  3. $\mathbf{Pre} : P \times T \rightarrow \mathbb{N}^{m \times n}$ and $\mathbf{Post} : P \times T \rightarrow \mathbb{N}^{m \times n}$ are the *pre-* and the *post-*incidence matrices, respectively, that specify the arcs connecting places and transitions. More precisely, for each $p \in P$ and $t \in T$, element $\mathbf{Pre}(p, t)$ ($\mathbf{Post}(p, t)$) is a natural number indicating the arc multiplicity if an arc going from p to t (from t to p) exists, and it equals 0 otherwise.

  Note that $| \cdot |$ denotes the cardinality of the generic set $\cdot$, and $\mathbb{N}$ is the set of non-negative integer numbers.

  Matrix $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the $m \times n$ *incidence matrix* of the PN. For each place $p \in P$, the set $\bullet p = \{t \in T | \mathbf{Post}(p, t) > 0\}$ is the *pre-set* of p, i.e., the set of *input transitions* of place p; similarly, the set $p\bullet = \{t \in T | \mathbf{Pre}(p, t) > 0\}$ is the *post-set* of p (*output transitions* of place p). Analogously, for each transition $t \in T$, the set $\bullet t = \{p \in P | \mathbf{Pre}(p, t) > 0\}$ is the *pre-set* of t (*input places* of t), while the set $t\bullet = \{p \in P | \mathbf{Post}(p, t) > 0\}$ is the *post-set* of t (*output places* of t).

  A *marking* is a function $\mathbf{M} : P \rightarrow \mathbb{N}^m$ that assigns to each place of the net a non-negative number of tokens that represents the state of the system: $\mathbf{M}(p)$ denotes the marking of the place $p \in P$, while $|\mathbf{M}| = \sum_{p \in P} \mathbf{M}(P)$ denotes the total number of tokens of marking $\mathbf{M}$ summed over all places. A *PN system* $< PN, \mathbf{M}_0 >$ is a net PN with initial marking $\mathbf{M}_0$.

Classical PN do not convey any notion of time, but in order to represent systems with temporal constraints, TPN have been introduced: TPN are obtained from PN by associating a firing time to each transition of the net [Marsan et al., 1994]. In particular, there are three types of timed transitions: immediate transitions (represented with bars), stochastic transitions (represented with boxes) and deterministic transitions (represented with black boxes).

More formally, a TPN is a six-tuple $TPN = (P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{F}, \mathbf{RS})$, where P, T, **Pre**, **Post** have the same meaning as described above. Moreover, function $\mathbf{F} : T \to \mathbb{R}_0^+$ specifies the timing associated to each transition, where $\mathbb{R}_0^+$ is the set of non-negative real numbers. In particular, $\mathbf{F}(t_j) = \delta_j$ specifies the timing associated to the timed deterministic transitions, and $\mathbf{F}(t_j) = (\lambda_j)^{-1}$ is the average firing delay each stochastic transition, where λ_j is the average transition firing rate. Finally, $\mathbf{RS} : T \to \mathbb{R}^+$ is a function that associates a probability value called random switch to conflicting transitions: indeed, under the *no concurrency assumption*, only one transition may fire at a time.

- **Net Dynamics**

  A transition $t_j \in T$ is said to be *enabled* at a marking $\mathbf{M}$ if and only if (iff) for each $p \in \bullet t$, $M(p) \geq \mathbf{Pre}(p, t)$ and this is denoted by $\mathbf{M}[t >$. When t fires, the net reaches a new marking $\mathbf{M}'$ computed by the state equation $\mathbf{M}' = \mathbf{M} + \mathbf{C}(\cdot, t)$.

  Let $\sigma = t_1 t_2 \dots t_k$, with $t_j \in T$, be a sequence of transitions (or *firing sequence*). σ is enabled at $\mathbf{M}$ iff $\mathbf{M}[t_1 > \mathbf{M}_1[t_2 > \mathbf{M}_2 \dots \mathbf{M}_{k-1} > t_k$ and it is denoted by $\mathbf{M}[\sigma >$. A marking $\mathbf{M}$ is *reachable* from $< PN, \mathbf{M}_0 >$ iff there exists a firing

sequence σ such that $\mathbf{M}_0[\sigma > \mathbf{M}$. The set of all markings reachable from $\mathbf{M}_0$ defines the *reachability set* of $< N, \mathbf{M}_0 >$ and is denoted by $R(PN, \mathbf{M}_0) = \{\mathbf{M}| \quad \exists \quad \sigma : \mathbf{M}_0[\sigma > \mathbf{M}\}$. We denote with $\overline{\sigma}(t_j)$ the number of occurrences of transition t_j in the firing sequence σ and $\overline{\boldsymbol{\sigma}} = [\overline{\sigma}(t_1)\overline{\sigma}(t_2)\ldots\overline{\sigma}(t_n)]^T$ is the firing vector associated to the firing sequence σ. If a sequence contains a single transition t_j, its firing sequence is denoted as $\overline{t_j}$.
Moreover, the enabling degree of transition $t_j \in T$ at $\mathbf{M}$ is equal to $enab(\mathbf{M}, t_j)$ $= \max\{k \in \mathbb{N}|\mathbf{M} \geq k \cdot \mathbf{Pre}(\cdot, t_j)\}$.
If t_j is infinite-server semantics, a number of clocks that is equal to $enab(\mathbf{M}; t_j)$ [Marsan et al., 1994] is associated to it. Each clock is initialized to a value that is equal to the time delay of t_j, if t_j is deterministic, or to a random value depending on the distribution function of t_j, if t_j is stochastic.
On the contrary, if a discrete transition is k-server semantics, then the number of clocks that are associated to t_j is equal to $\min\{k, enab(\mathbf{M}; t_j)\}$. The values of clocks associated to t_j decrease linearly with time, and t_j fires when the value of one of its clocks is null (if k clocks reach simultaneously a null value, then t_j fires k times).

Note that in this Section *enabling memory policy* is considered and, therefore, if a transition enabling degree is reduced by the firing of a different transition, then the disabled clocks have no memory of this in future enabling [Marsan et al., 1994], [Alla and David, 1998].

TPN Model

A modular TPN model is developed: each parking area of the considered CS system is represented by the same structure, which reflects the main phases characterizing the vehicle rental process described by the activity diagram of Fig. 3.3. In particular, the activity diagram is translated into the TPN model by a resource oriented approach, using the same guidelines described in [Fanti et al., 2013].

Different sub-models are considered: a sub-model that represents users arrival and waiting for a vehicle, one for the travel time determination, one that models the destination determination and, finally, a sub-model that represents the evaluation of the need of maintenance. Moreover, the maintenance operations are modelled by a structure that can be identically repeated on the basis of the available number of car parks in the service.

Fig. 3.4 shows the TPN model of a CS service with two parking areas (Park_1 and Park_2), however it can be easily adapted with few modifications to a CS system of any size. Infinite server semantics and enabling memory policy are considered.

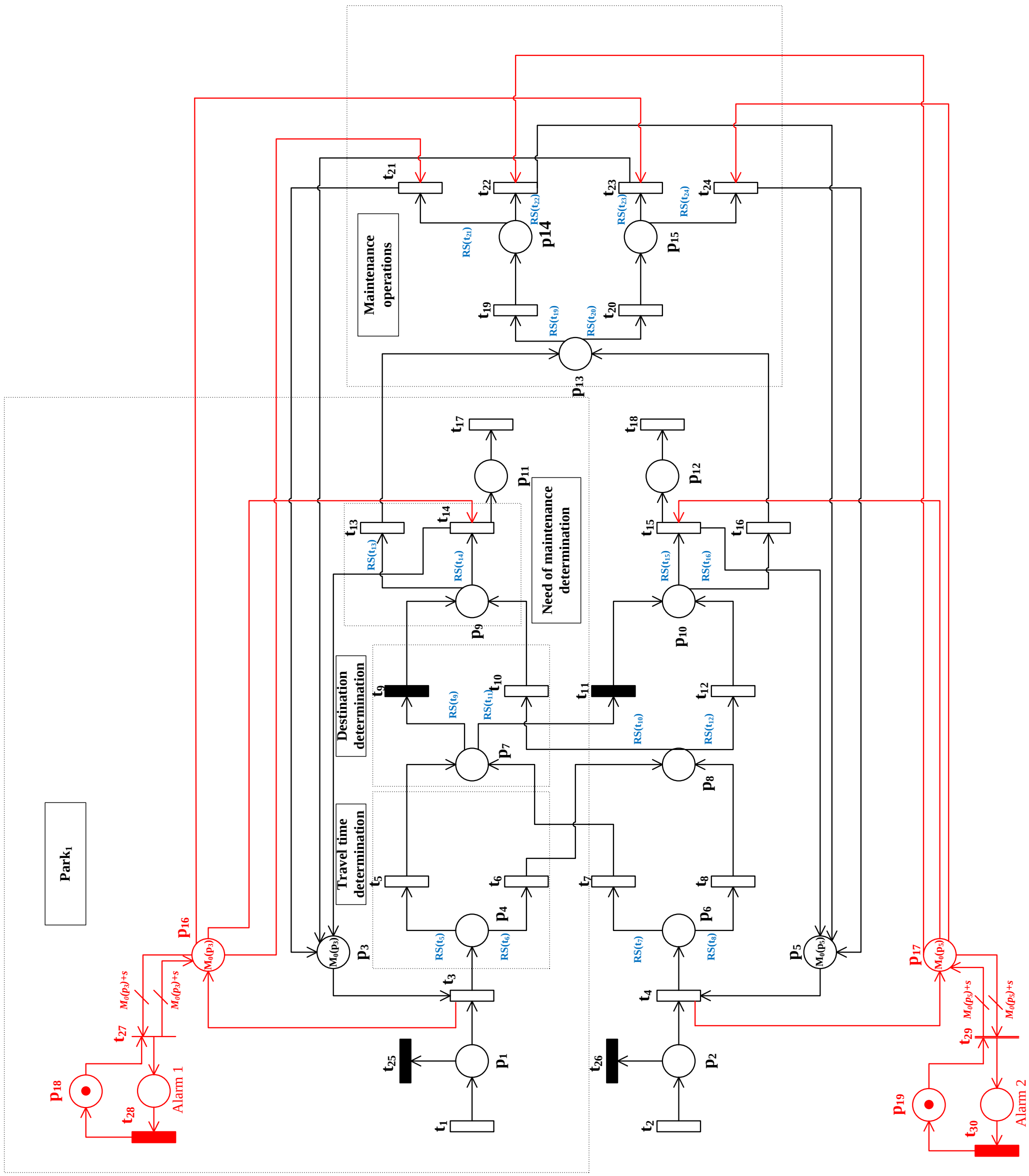

Figure 3.4: Complete TPN model of a CS system with 2 parking areas.

Tables 3.2, 3.3 and 3.4 show the meaning of places and transitions of the TPN.

More in detail, the user arrives in one of the two parking areas (transition t_1 and transition t_2). He/she waits for an available vehicle (places p_1 and p_2). If the waiting time is greater than a defined *maximum waiting time*, the user decides to not use the CS service (transitions t_{25} and t_{26}). Hence, the conflict between transition t_3 (t_4) and transition t_{25} (t_{26}) is resolved by the firing time.

If a vehicle is available, the user rents the vehicle in the parking area he/she arrived (transition t_3 or transition t_4). Now the users decides the rental length: two different possible travel times τ_1 time units (t.u.) and τ_2 t.u. are considered, with $\tau_1 < \tau_2$ (transition t_5 or transition t_7 for the parking area Park$_1$, transition t_6 or transition t_8 for Park$_2$). Note that the conflict between transitions t_5 (t_7) and t_6 (t_8) is now solved by the function **RS**, denoted in blue in Fig. 3.4.

Transitions t_9, t_{10}, t_{11} and t_{12} model the behaviour of the rented vehicle. For example, transition t_9 (t_{11}) represents the utilization for τ_1 t.u. of a vehicle that will be returned in Park$_1$ (Park$_2$), while transition t_{10} (t_{12}) represents the utilization for τ_2 t.u. of a vehicle that will be returned in Park$_1$ (Park$_2$). The conflicts between transitions t_9 and t_{11} and between t_{10} and t_{12} are solved by the function **RS**.

When the user returns the vehicle, two cases are possible. If the vehicle does not need maintenance, nor to be charged, in case of EV, it is again available in the parking area Park$_1$ or (Park$_2$) (transitions t_{14} and t_{15}, respectively); then, the user leaves the parking (transition t_{17} for parking area Park$_1$ and transition t_{18} for parking area Park$_2$).

If the vehicle need maintenance, transition t_{13} or transition t_{16} is enabled. Maintenance can be *short* (transition t_{19}) or *long* (transition t_{20}).

After the maintenance, the vehicle is dropped off to the parking area Park$_1$ or the parking area Park$_2$ (transitions t_{21}, t_{22}, t_{23} and t_{24}). All the conflicts are solved by the function **RS**, as shown in Fig. 3.4.

Note, in Fig. 3.4, the different modelling approaches for the two possible rental lengths: while travels of length τ_1 t.u. are represented by deterministic transitions (depicted as black rectangles), longer trips are modelled by stochastic transitions (white rectangles), since more delays and accidents may occur during the rental period. In particular, exponential distribution of parameter $\lambda[t.u]^{-1}$ is considered for the stochastic transitions.

Finally, the real time monitoring of the system necessary for the relocation strategy is represented in red in Fig. 3.4. In particular, if place *Alarm 1* (*Alarm 2*) is marked, then there is a vehicles shortage in parking area Park$_1$ (Park$_2$): therefore, the apt incentive is communicated to the users.

The impact of the real time suggestions on the customers behaviour is modelled with the variation of the **RS** of the involved conflicting transitions. The amount of such a variation varies with the specific parking area, considering that one parking area could be less attractive than the other. Note that the initial marking of place p_{16} (p_{17}) is equal to the initial marking of place p_3 (p_5), i.e., to the number of vehicles initially available at parking area Park$_1$ (Park$_2$). Moreover, denoted with $s \in \mathbb{N}$ is the a-priori defined set point for the incentive activation, it holds: $\mathbf{Pre}(p_{16}, t_{27}) = \mathbf{M}_0(p_3) + (s + 1)$ ($\mathbf{Pre}(p_{17}, t_{29}) = \mathbf{M}_0(p_5) + (s + 1)$).

The TPN model of Fig. 3.4 is simulated in the MATLAB®software environment. Such a matrix-based engineering software appears particularly appropriate to simulate the dynamics of TPN (matrix formulation of the marking update), as well as to describe and simulate PN systems with a large number of places and transitions.

3.4.2.2 Decision Module Specification

In order to assess the impact of the considered relocation strategy, the following KPIs are defined.

- *Level of Service* (*LOS*). This performance index is expressed in terms of average fraction of served users, and is a typical index suitable for evaluating the behaviour of this kind of systems [Nourinejad and Roorda, 2015], [Nourinejad

Table 3.2: Meaning of places of the TPN of Fig. 3.4.

Name	**Meaning**
p_1	user waiting for an available vehicle in Park_1
p_2	user waiting for an available vehicle in Park_2
p_3	parking Park_1 capacity
p_4	rented vehicle in Park_1
p_5	parking Park_2 capacity
p_6	rented vehicle in Park_2
p_7	selected rental length is τ_1 t.u.
p_8	selected rental length is τ_2 t.u.
p_9	selected destination is Park_1
p_{10}	selected destination is Park_2
p_{11}	vehicle again available in Park_1
p_{12}	vehicle again available in Park_2
p_{13}	vehicle maintenance required
p_{14}	short maintenance
p_{15}	long maintenance
p_{16}	monitored number of available vehicles in Park_1
p_{17}	monitored number of available vehicles in Park_2
Alarm 1	incentive mechanism active (Park_1)
Alarm 2	incentive mechanism active (Park_2)
p_{18}	*Alarm 1* capacity
p_{19}	*Alarm 1* capacity

Table 3.3: Meaning of the stochastic transitions of the TPN of Fig. 3.4.

Name	**Meaning**
t_1	user arrival at parking Park_1
t_2	user arrival at parking Park_2
t_3	vehicle rental at parking Park_1
t_4	vehicle rental at parking Park_2
t_5	user decision to rent the car for τ_1 t.u.
t_6	user decision to rent the car for τ_2 t.u.
t_7	user decision to rent the car for τ_1 t.u.
t_8	user decision to rent the car for τ_2 t.u.
t_{10}	destination determination (Park_1) (rent of τ_2 t.u.)
t_{12}	destination determination (Park_2) (rent of τ_2 t.u.)
t_{13}	vehicle picked up for maintenance
t_{14}	vehicle restitution at parking Park_1
t_{15}	vehicle restitution at parking Park_2
t_{16}	vehicle picked up for maintenance
t_{17}	user departure from parking Park_1
t_{18}	user departure from parking Park_2
t_{19}	short maintenance operations
t_{20}	long maintenance operations
t_{21}	vehicle dropped off in Park_1 after the maintenance
t_{22}	vehicle dropped off in Park_2 after the maintenance
t_{23}	vehicle dropped off in Park_1 after the maintenance
t_{24}	vehicle dropped off in Park_2 after the maintenance

Table 3.4: Meaning of the deterministic transitions of the TPN of Fig. 3.4.

Name	**Meaning**
t_9	destination parking determination (Park_1) during a rent of τ_1 t.u.
t_{11}	destination parking determination (Park_1) during a rent of τ_1 t.u.
t_{25}	user departure from Park_1 without being served
t_{26}	user departure from Park_2 without being served
t_{27}	incentive mechanism activation (Park_1)
t_{28}	incentive notification to user
t_{29}	incentive mechanism activation (Park_2)
t_{30}	incentive notification to user

and Roorda, 2014], [Alfian et al., 2014], [Bianchessi et al., 2013].

Formally, denoted with σ_{sim} the firing sequence associated to a simulation and with $\overline{\sigma}_{sim}(t_j)$ the number of occurrences of transition t_j in the firing sequence σ_{sim}, LOS is defined as follows:

$$LOS = \frac{\overline{\sigma}_{sim}(t_3) + \overline{\sigma}_{sim}(t_4)}{\overline{\sigma}_{sim}(t_1) + \overline{\sigma}_{sim}(t_2)} \tag{3.1}$$

Indeed, t_3 (t_4) fires if a user rents a vehicle in parking area Park$_1$ (Park$_2$), while t_1 and t_2 represents the total number of users arriving to the parking Park$_1$ and Park$_2$, respectively.

- *Company Revenue* (R). Such index is defined as the sum of the total travel cost supported by each user, considering the possibility of monetary incentives.

  In CS services, usually, each user has to pay both a *distance charge* and a *hourly charge*. In the presented context, only the hourly charge is taken into account, since it is the cost that is influenced by the incentive mechanism. Formally, the value of R is defined as follows:

$$\begin{aligned} R = &(\text{hourly charge}) \cdot (\delta_9 \cdot \overline{\sigma}_{sim}(t_9)+ \\ &+ \delta_{11} \cdot \overline{\sigma}_{sim}(t_{11})+ \\ &+ (\lambda_{10})^{-1} \cdot \overline{\sigma}_{sim}(t_{10}) + (\lambda_{12})^{-1} \cdot \overline{\sigma}_{sim}(t_{12}))\big) \end{aligned} \tag{3.2}$$

  The transitions t_9, t_{10}, t_{11}, t_{12} represent the decision of the user to rent the vehicle for τ_1 t.u. or τ_2 t.u., respectively. Moreover, δ_i is the firing delay of the deterministic transition $i \in T$, while λ_j is the average transition firing rate of the stochastic transition $j \in T$. Hence, $\delta_9 \cdot \overline{\sigma}_{sim}(t_9)$, $\delta_1 1 \cdot \overline{\sigma}_{sim}(t_1 1)$, $(\lambda_{10})^{-1} \cdot \overline{\sigma}_{sim}(t_{10})$ and $(\lambda_{12})^{-1} \cdot \overline{\sigma}_{sim}(t_{12}))$ represent the total time of utilization of the rented car. Moreover, in case of incentives application, *discount*, i.e., the amount of the discount received by each user that follows the provided suggestions of behaviour, has to be applied to the travels of length τ_1 (i.e., at firing of transitions t_9 and t_{11}).

- *Company Gain* (G). To evaluate this index, a *monetary penalty* P is introduced in order to quantify the damage that the users that leave the system without having been served represent for the company: indeed, the inability to satisfy users requests does not represent only a loss of earning for the company, but it has also repercussions on the image and the attractiveness of the service itself. Formally, P is defined as follows:

$$P = [penalty] \cdot (\overline{\sigma}_{sim}(t_{25}) + \overline{\sigma}_{sim}(t_{26})) \tag{3.3}$$

where $\overline{\sigma}_{sim}(t_{25})$ and $\overline{\sigma}_{sim}(t_{26})$ represent the number of disappointed users that leave the system without using the CS service in Park_1 and Park_2, respectively, while *penalty* is the monetary quantification of the damage caused by each user that has not been served.

Therefore, the company gain G can be straightforwardly calculated as follows:

$$G = R - P \tag{3.4}$$

where R is the company revenue defined in 3.2.

3.4.3 Case Study

3.4.3.1 CS service specification

As discussed above, in this Section the DSS is used to investigate the impacts of the considered a-priori set point relocation strategy. To this aim, a case study based on the real experience of the CS pilot service of Pordenone, a town of the North of Italy, is considered. Although it is a system of limited size, the results achieved in this context can be easily adapted and extended to any generic CS service. The considered system is characterized as follows.

- **Number of parking areas**. $N = 2$ parking areas are considered, named P_3 and P_5 hereafter.
- **Fleet size**. A total number of vehicles $V = 10$ is considered.
- **Fleet composition**. Both traditional vehicles and EVs are considered.
- **Service hours**. CS service is operative for 12 hours per day, 30 days per month.

Moreover, the following additional assumptions are considered.

- **Time unit**. The minute is considered as t.u.
- **Maximum waiting time**. The maximum waiting time is assumed equal to 10 minutes.
- **Maintenance**. Only one car park is available for the maintenance of the vehicles, with the capability to operate on a single vehicle at a time. It is assumed that the probability that a vehicle needs maintenance after the rental period is equal to 0.10 ($RS(t_{13}) = 0.10$, $RS(t_{16}) = 0.10$). Moreover, the 99% of the vehicles are available again in one of the two parking areas after a short maintenance ($RS(t_{19}) = 0.99$).
- **Initial distribution of vehicles**. It is assumed that, in the initial condition, the vehicles are equally distributed between the two parking areas ($\mathbf{M}_0(p_3) = \mathbf{M}_0(p_5) = 5$).
- **Monetary penalty**. A value of 5 € for each not served user is considered.
- **Hourly charge and discount**. The considered hourly charge is equal to 5€\hour, with a discount of 20% in case of incentive acceptance.

- **System monitoring**. The system balance conditions are monitored every 30 minutes during the working day.

- **Incentive strategy specification**. The a-priori set point considered for the incentive implementation is $s = 2$. Therefore, four types of incentives are possible:

    1. *type 0*: both in P_3 and in P_5 there are more than 2 available vehicles. The system is still balanced and no suggestions has to be provided to the customers.
    2. *type 1*: both in P_3 and in P_5 there are 2 or less available vehicles. Customers are encouraged to drop off the rented vehicle as soon as possible.
    3. *type 2*: in P_3 there are 2 or less available vehicles. Users are encouraged to drop off the vehicle in P_3.
    4. *type 3*: in P_5 there are 2 or less available vehicles. Users are encouraged to drop off the vehicle in P_5.

    Table 3.5 summarizes the types of incentive and their effects on the random switches of conflicting transitions. Note that it is assumed that P_5 is less attractive for users than P_3 and so customers are more unwilling to choose it as a travel destination, even with the promise of a discount.

Tables 3.6, 3.7 and 3.8 specify the TPN system of Fig. 3.4 (initial marking and transitions firing delays). Note that both for readability and coherence with equation 3.2, the average transitions firing rates of stochastic transitions and the firing rates of deterministic transitions are reported in hours.

3.4.3.2 Simulation specification

Three scenarios (A, B, C), characterized by different levels of congestion of the system, are considered: users inter-arrival times in *Scenario A*, *Scenario B* and

Table 3.5: Effects of the incentive on the RS of conflicting transitions

Incentive type	**Condition**	**Effect**
0	$\mathbf{M}(p_16) < 8 \wedge \mathbf{M}(p_17) < 8$	$RS(t_5) = RS(t_7) = 0.60$ $RS(t_6) = RS(t_8) = 0.40$ $RS(t_9) = RS(t_{10}) = 0.60$ $RS(t_11) = RS(t_{12}) = 0.40$
1	$\mathbf{M}(p_16) \geq 8 \wedge \mathbf{M}(p_17) \geq 8$	$RS(t_5) = RS(t_7) = 0.70$ $RS(t_6) = RS(t_8) = 0.30$
2	$\mathbf{M}(p_16) \geq 8$	$RS(t_9) = RS(t_{10}) = 0.70$ $RS(t_{11}) = RS(t_{12}) = 0.30$
3	$\mathbf{M}(p_17) \geq 8$	$RS(t_9) = RS(t_{10}) = 0.45$ $RS(t_{11}) = RS(t_{12}) = 0.55$

Table 3.6: Initial marking the TPN of Fig. 3.4.

Name	$\mathbf{M}_0(p_i)$	**Name**	$\mathbf{M}_0(p_i)$	**Name**	$\mathbf{M}_0(p_i)$
p_1	0	p_8	0	p_{15}	0
p_2	0	p_9	0	p_{16}	5
p_3	5	p_{10}	0	p_{17}	5
p_4	0	p_{11}	0	*Alarm 1*	0
p_5	5	p_{12}	0	*Alarm 2*	0
p_6	0	p_{13}	0	p_{18}	1
p_7	0	p_{14}	0	p_{19}	1

Table 3.7: Average transitions firing rates $\lambda\,[h]^{-1}$ of stochastic transitions of the TPN of Fig. 3.4.

Name	λ_s	**Name**	λ_s	**Name**	λ_s
t_1	2.5	t_{10}	1	t_{18}	15
t_2	2	t_{12}	1	t_{19}	1
t_3	20	t_{13}	0.333	t_{20}	0.125
t_4	20	t_{14}	15	t_{21}	3
t_5	40	t_{15}	15	t_{22}	3
t_6	40	t_{16}	0.333	t_{23}	3
t_7	40	t_{17}	15	t_{24}	3
t_8	40				

Table 3.8: Firing delays $\delta_d\,[h]$ of deterministic transitions of the TPN of Fig. 3.4.

Name	δ_d	**Name**	δ_d
t_9	0.333	t_{27}	0
t_{11}	0.333	t_{28}	0.5
t_{25}	0.167	t_{29}	0
t_{26}	0.167	t_{30}	0.5

Table 3.9: Users inter-arrival times $\lambda^{-1}(t_1)$ and $\lambda^{-1}(t_2)$ [min] in *Scenario A*, *B* and *C*.

Scenario	$\lambda^{-1}(t_1)$	$\lambda^{-1}(t_2)$
A	24	30
B	12	20
C	10	15

Scenario C, i.e., $\lambda^{-1}(t_1)$ [min] and $\lambda^{-1}(t_2)$ [min] are reported in Table 3.9. Note that the exponential distribution parameter $\lambda_s(t_1)$ and $\lambda_s(t_2)$ of Table 3.7 are referred to *Scenario A*.

Moreover, in order to study the outcomes of the considered relocation strategy, three different operative conditions are compared.

In the first one, no incentive system is taken into account and vehicles are relocated only at the end of the working day (*Case As Is*).

In the second case, users are always encouraged to drop off the rented vehicle as soon as possible, regardless of the system actual balance conditions (*Case To Be - Offline*).

In the third operative condition, the vehicles distribution among the parking areas is monitored at regular time intervals and, whenever the system is unbalanced, suitable travel suggestions are provided in real time to the users (*Case To Be - Online*).

The KPIs defined in 3.1, 3.2 and 3.4 are evaluated by a long simulation run of 21600 minutes (30 days, 12 hours per day), with a transient period of 30 minutes. In particular, the estimates of the performance indices are deduced by 50 independent replications, with a 95% confidence interval. Besides, the half width of the confidence interval is about 2.2% in the worst case, confirming the sufficient accuracy of the performance indices estimation. Finally, considering that the average CPU time for a simulation run is about 408 seconds on a PC equipped with a 1.73 GHz processor and 1 GB RAM, the presented modeling and simulation approach can be applied to large systems.

3.4.3.3 Simulation results

Simulation results are depicted in Fig. 3.5, Fig. 3.6 and Fig. 3.7.

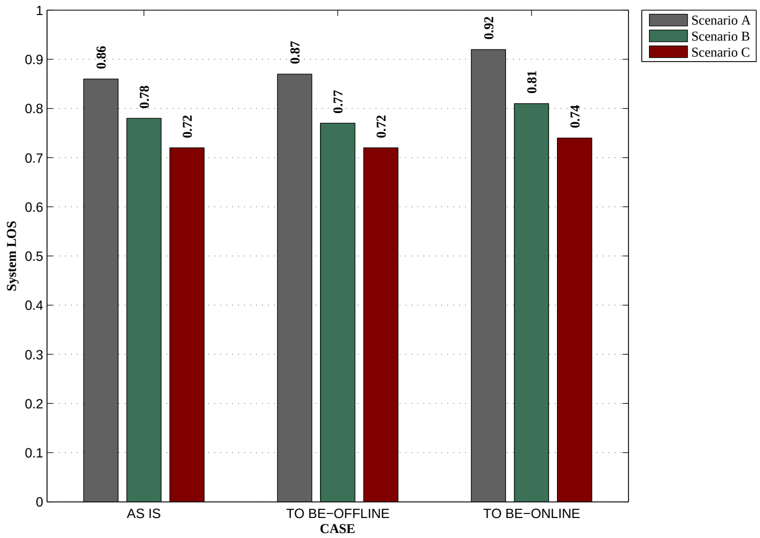

Figure 3.5: Average system LOS for each case in all the considered scenarios.

In Fig. 3.5 the system *LOS* is analysed. It is interesting to note that an incentive mechanism that does not consider the instantaneous vehicle balance conditions of the service (*Case To Be - Offline*) does not improve significantly the LOS. On the contrary, it results to be even counter-productive or irrelevant in *Scenario B* and in *Scenario C*, i.e., as the congestion level of the system increases.
At the same time, in the *Case To Be - Online* in *Scenario A* there is a *LOS* increase of about 7% with respect to the *Case As Is*, but the entity of this increase is reduced in *Scenario B* and in *Scenario C* (4% and 3%, respectively) : this means that the effectiveness of the proposed incentive mechanism decreases as the congestion of the system grows and such a solution is not able to guarantee evident benefits when the number of available vehicles is undersized compared to the mobility demand.

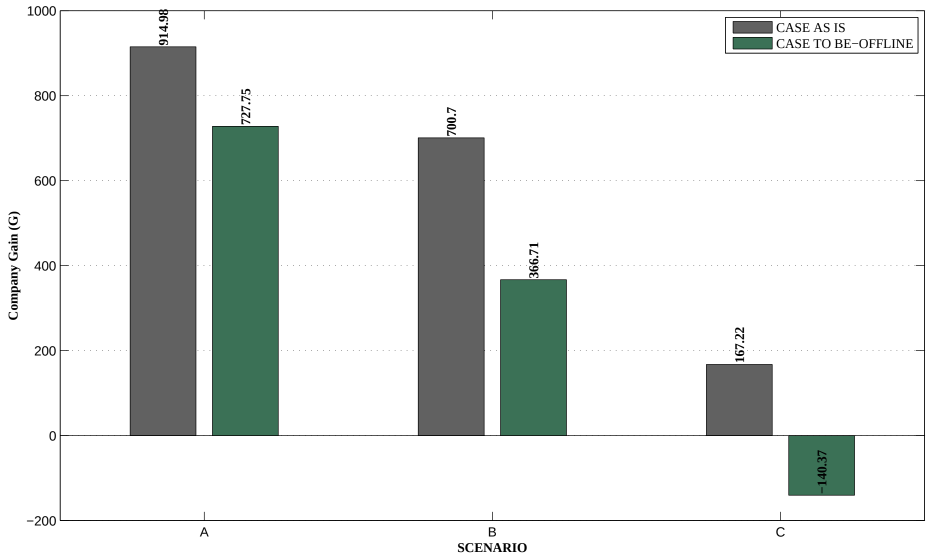

Figure 3.6: Average company gain comparison between *Case As Is* and *Case To Be-Offline*.

It is possible to observe the same behaviour also in the analysis of the impact of the proposed solution on the average CS company gain G, as shown in Fig. 3.6 and Fig. 3.7. In particular, Fig. 3.6 is referred to the *Case To Be - Offline*: as can be easily seen, the introduction of the incentive mechanism leads to a reduction of the company gain (with a peak of -49% in *Scenario B*) and to a real economic loss in *Scenario C*. On the other hand, when the typology of suggestion provided to the users is based on the current system conditions (Fig. 3.7), the company gain is higher in the first two scenarios, but the effects is *Scenario B* is less pronounced (+19% vs +7%; in *Scenario C*, on the contrary, there is a decrease of the company gain, therefore not even a real time monitoring of the system balance conditions is sufficient to ensure an enhancement of the system performance when it turns out to be too congested.

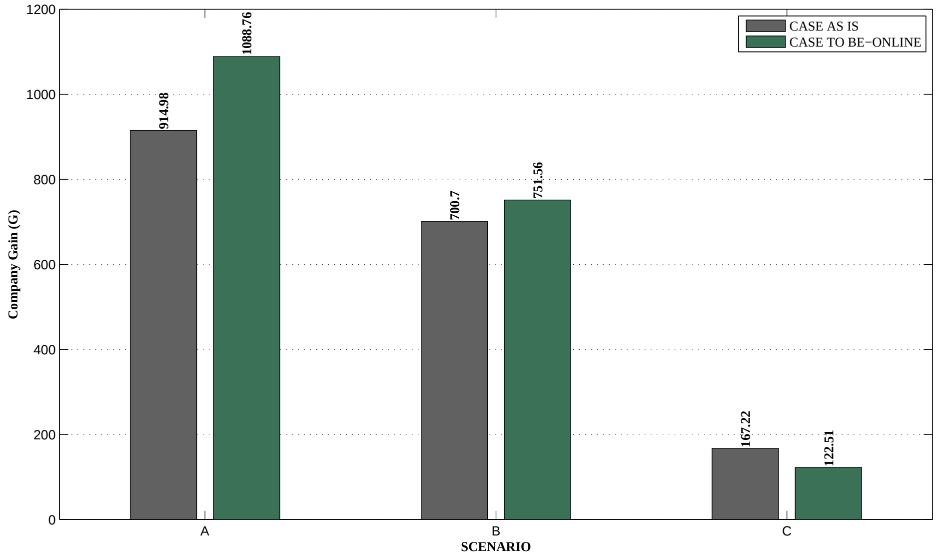

Figure 3.7: Average company gain comparison between *Case As Is* and *Case To Be-Online*.

3.4.3.3.1 Discussion about of the Proposed Solution

The introduction of an incentive mechanism based on a continuous monitoring of the vehicles distribution among the parking areas improves the *LOS* of a CS system with positive economical outcomes for the company itself, but it cannot disregard a prior and coherent sizing of the system. On the other hand, a mechanism which does not consider the actual balance conditions of the system does not turn out to be a concrete solution for the user-based vehicle relocation problem. Compared to the techniques described in other works previously mentioned, the considered approach does not take into account the possibility of *ridesharing* (users share a ride in a single vehicle) or *trip splitting* (multiple users that have to travel between the same origin and destination drive separate vehicles), and this represents an advantage for customers who can be unwilling to share the same vehicle with strangers or to travel separately from their acquaintances. However, the effectiveness of the proposed solution highly depends on users participation and, so, the percentage of discount on the total travel cost has to be strategically determined.

3.5 Optimized Set-point Problem

3.5.1 Problem Statement

In this Section the following problem is considered [Clemente et al., 2016b].

1. **Class of user-based vehicle relocation problem**.

    - *relocation modality*: the relocation activities are performed only by the users during the service hours, while at the end of the working day system staff relocates the vehicles (*user-based*);
    - *relocation time*: the system status is monitored at regular intervals throughout the day (*real time*);
    - *relocation control*: the incentive for a parking area is triggered only when a minimum number of available vehicles is reached (*non-predictive*);
    - *set-point*: the minimum number of vehicles that should be available in each parking area is determined on the basis of the system state knowledge (*optimized set-point*).

2. **Relocation strategy rules**.

    - *incentive notification*: users receive the notification about possible incentives at the beginning of the rental period (before the effective usage of the car);
    - *types of incentive*: only the *destination-incentive* is considered in this case, therefore users are encouraged to drop off the vehicles in suitable parking areas on the basis of the system balance conditions.

3. **Role of the DSS**.

   The DSS is used to solve the optimized set-point user-based vehicle relocation problem. In particular, the optimization module of the DSS implements a Particle Swarm Optimization algorithm whose fitness function in evaluated through the simulation module. Indeed, in order to the determine the optimal set-points for the incentive mechanism, it is necessary to evaluate their impact on the overall system performance, but this strictly depends on the customers reaction to the received incentives: the simulation is suitable to take into account the stochasticity of customers behaviour while identifying the optimal threshold vector.

In order to formally describe the considered problem, in the following a Discrete Event System (DES) model [Cassandras and Lafortune, 2008] for a generic CS system is described. Then, the proposed user-based relocation strategy is specified.

3.5.1.1 Discrete Event System Model

A CS system constituted by N parking areas is formally modelled as a DES described by the automaton $\mathcal{A}=\{\mathcal{E},\mathcal{X},\boldsymbol{f}\}$, where $\mathcal{E}$ is the event set, $\mathcal{X}$ is the state set, and $\boldsymbol{f}$ is the state transition function [Cassandras and Lafortune, 2008].

Denote by $\mathcal{P} = \{1, 2,\ldots, N\}$ the set of the N CS parking areas and by V the total number of vehicles composing the service fleet. In accordance to the activity diagram of Fig. 3.3, the following events are defined for the parking area $i \in \mathcal{P}$: a_i is the arrival of a customer; r_i is the quit of a customer without having rented a vehicle; p_i is the vehicle pick-up; d_i is the vehicle drop off; m_i is the maintenance operation for a vehicle.

Hence, the set of the *events* that determine the evolution of the CS system is the following:

$$\mathcal{E}=\{a_i, r_i, p_i, d_i, m_i : i \in \mathcal{P}\}. \tag{3.5}$$

.

Moreover, the state of the parking area $i \in \mathcal{P}$ is denoted by the following vector:

$$\mathbf{x_i} = \begin{bmatrix} q_i \\ v_i \end{bmatrix}, \tag{3.6}$$

where $q_i \in \mathbb{N}$ is the number of customers waiting to rent a vehicle at the parking area i, $v_i \in \mathbb{N}$ is the number of vehicles available at the parking area i and $\mathbb{N}$ is the set of natural numbers.

Hence, the system *state* is denoted by the following matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1 x_2} \ldots \mathbf{x_N} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix} \tag{3.7}$$

with $\mathbf{q} = [q_1 q_2 \ldots q_N]$ and $\mathbf{v} = [v_1 v_2 \ldots v_N]$.

Since it is reasonable to suppose that every user is willing to wait for a limited time interval before leaving without being served, the queue in each parking area can not increase indefinitely. Hence, assuming $Q \in \mathbb{N}^+$ a sufficiently large integer and $q_i \leq Q \; \forall i \in \mathcal{P}$, the set of the system states is the following:

$$\mathcal{X} = \left\{ \mathbf{X} \mid v_i = 0, 1, \ldots, V \quad q_i = 0, 1, ..., Q \quad i = 1, 2, ..., N \right\} \tag{3.8}$$

The system dynamics is described by the state equation vector $\mathbf{f} : \mathcal{X} \times \mathcal{E} \to \mathcal{X}$ defined as follows:

$$\mathbf{X}^k = \mathbf{f}(\mathbf{X}^{k-1}, e^k), \tag{3.9}$$

where $\mathbf{X}^k = [\mathbf{x_1^k x_2^k} \ldots \mathbf{x_n^k}]$, with $\mathbf{x_i^k} = [q_i^k \quad v_i^k + 1]^T$ is the state that the system reaches after the occurrence of event $e^k \in \mathcal{E}$, starting from state $\mathbf{X}^{k-1}$.

In particular, for each parking area $i \in \mathcal{P}$, the state transition function is the defined as follows:

$$\mathbf{x_i^k} = f_i(\mathbf{x_i^{k-1}}, e^k) = \begin{cases} [q_i^k \quad v_i^k + 1]^T & if \quad e^k = d_i \\ [q_i^k - 1 \quad v_i^k - 1]^T & if \quad e^k = p_i \\ [q_i^k - 1 \quad v_i^k]^T & if \quad e^k = r_i \\ [q_i^k + 1 \quad v_i^k]^T & if \quad e^k = a_i \\ [q_i^k \quad v_i^k - 1]^T & if \quad e^k = m_i \end{cases} . \tag{3.10}$$

Moreover, the occurrences of the events in $\mathcal{E}$ can be characterized as follows for each $i \in \mathcal{P}$:

- events a_i and m_i are the independent inputs of the system;
- events p_i may occur if $v_i > 0$, i.e., they are function of the system state;
- events r_i may occur if $v_i = 0$, i.e., they are function of the system state;
- events d_i are controlled events, i.e., the occurrences of such events are affected by the relocation strategy in order to guarantee a suitable number of available vehicles in each parking station.

In addition, regarding the state updating the following aspects are enlightened:

- events p_i, r_i and a_i with $i \in \mathcal{P}$ affect the number q_i of customers waiting to rent a vehicle in the parking area i;
- events d_i, p_i and m_i with $i \in \mathcal{P}$ affect the number v_i of vehicles in the parking area i.

3.5.1.2 User-Based Relocation Strategy

The relocation strategy is specified by introducing two matrices that allow describing the availability of a client to drive to an incentivized parking area, considering two important aspects: i) the willingness of the customer to drop off the rented car in a parking area different from his original destination; ii) the topographical relationships between two different parking areas by considering their distance and reciprocal positions.

- The *routing matrix* $\mathbf{R} \in \mathbb{R}^{N \times N}$. The element $r_{ij} \in [0, 1]$ is the probability that the customer drops off the car in the parking area j provided that the car is rent in the parking area i: such a value depends on the parking area locations, the time of the day, the day of the week and the type of the customers.

- The *affinity matrix* $\mathbf{A} \in \{0, 1, \ldots, N-1\}^{N \times N}$. Matrix $\mathbf{A}$ is introduced in order to model the attitude of typical customers to accept incentives to change their final destinations. Such attitude depends mainly on the specific pair of original and suggested destinations, but it takes into account also other factors, such as rush hours, day of the week, weather conditions, public transport alternatives and type of customer. Formally, $a_{ij} = 0$ $(= N - 1)$ means that park i has no affinity (maximum affinity) with respect to park j.

Now, the following variables are defined:

- $\mathbf{S}^{opt} \in \mathbb{N}^N$ is the *threshold vector suggested* by the DSS: $s_i^{opt} \in \mathbb{N}$ with $i \in \mathcal{P}$ denotes the minimum number of vehicles that should be available in the parking area i;

- $\mathbf{v}^* \in \mathbb{N}^N$ is the *threshold vector validated* by the decision maker;

- $\mathbf{u} \in \{0, 1\}^N$ is the *control vector*: $u_i = 1$ with $i \in \mathcal{P}$ if the incentive is activated for the parking area i and $u_i = 0$ otherwise.

The closed-loop control scheme to manage the user-based relocation problem is sketched in Fig. 3.8. The DSS receives vector $\mathbf{q}$ of the CS system state denoting the number of customers waiting for a vehicle and compares it with an expected value $\bar{\mathbf{q}}$, determined by the DSS on the basis of historical data. Denoted with $\mathbf{h} = \bar{\mathbf{q}} - \mathbf{q}$, if for some $i \in \mathcal{P}$ it holds $h_i < 0$, then the DSS triggers a new simulation-optimization campaign and determines the value $\mathbf{S}^{opt}$ of the vehicle thresholds in each parking area.

Vector $\mathbf{S}^{opt}$ is then checked by the decision maker and $\mathbf{v}^{*}$=$\mathbf{S}^{opt}$ is the new set-point for the successive control loop that manages the number of vehicles in each parking area.

Hence, the CS information system compares $\mathbf{v}^{*}$ with the system state and applies the following control law:
if $v_i \leq v_i^{*}$ then $u_i = 1$ for $i \in \mathcal{P}$, i.e., users are encouraged to drop off the vehicle in the parking area i.

Now, denote with $\nu(e_i)$ the number of occurrences of event $e_i \in \mathcal{E}$ during a working day. The proposed control strategy affects the event occurrences $\nu(d_i)$ of the described automaton $\mathcal{A} = \{\mathcal{E}, \mathcal{X}, \boldsymbol{f}\}$ and therefore the number of vehicles v_i available in each parking area.

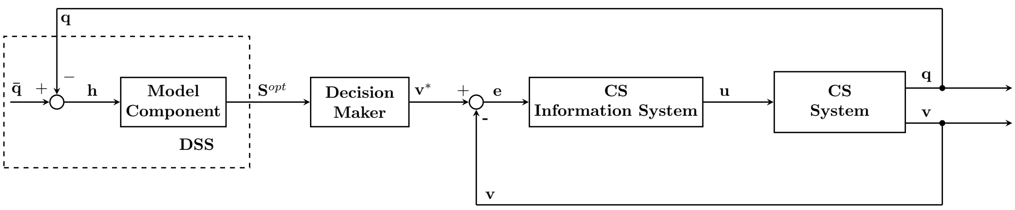

Figure 3.8: Complete control scheme resulting from the introduction of the proposed DSS.

3.5.2 DSS Modules Specifications

3.5.2.1 Simulation Module Specification

The DES model and the UML activity diagram of Fig. 3.3 can be translated in the Arena®environment, a discrete-event simulation software particularly suitable for dealing with large-scale and modular systems [Kelton et al., 2002].

Indeed, the Arena®simulation model can be straightforwardly implemented by the following three steps [Boschian et al., 2011].

- The Arena®modules are associated with the UML activity diagram elements, by establishing a kind of mapping between each Arena®module and the UML graphical element of the activity diagrams [Clemente et al., 2013a], [Clemente et al., 2015].

- The simulation parameters are included in the Arena®environment, i.e., the activity times, the process probabilities, the resource capacities, and the average input rates are assigned.

- The simulation runs of the experiments are singled out and the performance indices are determined and evaluated by means of suitable statistics functions.

In order to realistically evaluate the availability of a client to drive to an incentivized parking area, the following probabilities influenced by the control strategy and the affinity matrix $\mathbf{A}$ can be defined:

- $p_{select}(i|j,\mathbf{u})$ is the probability that the parking area i among the incentivized parking areas is selected instead of the original destination j. Such a probability can be determined as follows:

$$p_{select}(i|j,\mathbf{u}) = \frac{a_{ij}u_i}{\sum_{h=1}^{N} a_{hj}u_h} \tag{3.11}$$

  Note that $p_{select}(i|j,\mathbf{u}) = 0$ if area i is not incentivized ($u_i = 0$) or if there is not any affinity between i and j ($a_{ij} = 0$). Moreover, $p_{select}(i|j,\mathbf{u}) = 1$ if i is the only incentivized parking area and $a_{ij} > 0$.

- $p_{available}(ij)$ is the probability that the user accepts the selected parking area i, instead of the original destination j. Such a probability can be determined as follows by the affinity matrix:

$$p_{available}(ij) = \frac{a_{ij}}{\max_h a_{ih}} \cdot \vartheta, \tag{3.12}$$

where ϑ is the maximum value of the probability that an user accepts the new destination. Note that $p_{available}(ij) = \vartheta$ if $a_{ij} = \max_h a_{ih}$, i.e., the acceptance probability is maximum; $p_{available}(ij) < \vartheta$ in the other cases.

- $p_{accept}(i|j, \mathbf{u})$ is the probability that the customer accepts the incentive and returns the rented vehicle at the parking area i provided that the original chosen destination is area j. In particular, it turns out that

$$p_{accept}(i|j, \mathbf{u}) = p_{select}(i|j, \mathbf{u}) \cdot p_{available}(ij) \tag{3.13}$$

Remark that $p_0 = 1 - \sum_{h=1}^{N} p_{accept}(h|j, \mathbf{u})$ is the probability that the customer does not accept the incentives.

3.5.2.2 Decision Module Specification

Also in this case, the performance of the system is evaluated by studying the Level Of Service (LOS). In particular, according to the DES model, the LOS is defined as:

$$LOS = \frac{\sum_{i=1}^{N} \nu(p_i)}{\sum_{i=1}^{N} \nu(a_i)} \tag{3.14}$$

Fig. 3.9 sketches the interactions among the *decision*, *optimization* and *simulation modules*. On the basis of the value of $\mathbf{h}$, the decision module triggers a new optimization-simulation campaign: the decision variable is the threshold vector $\mathbf{S} \in \mathbb{N}^N$ and the objective function to be optimized is the system LOS.

The CS system dynamics is very complex and it is not possible to obtain an explicit formulation of the objective function. Therefore, the simulation module is exploited to evaluate the LOS and the PSO algorithm is used to optimize the objective function. The rationality of choosing the PSO algorithm with respect to other evolutionary methods is that the PSO is robust, efficient, suitable to handle non-linear problems and requires fewer number of function evaluations than genetic algorithms, while leading to better or the same quality of results [Perez and Behdinan, 2007], [Hassan et al., 2005].

The optimization module identifies the candidate values of $\mathbf{S}$ on the basis of the actual number $\mathbf{q}$ of customers waiting in the system. When the optimal value for the thresholds $\mathbf{S}_{new}^{opt}$ has been reached, the optimization module provides it to the decision module, that validates it and suggests to the decision maker the new candidate threshold vector $\mathbf{S}^{opt}$ through the interface component.

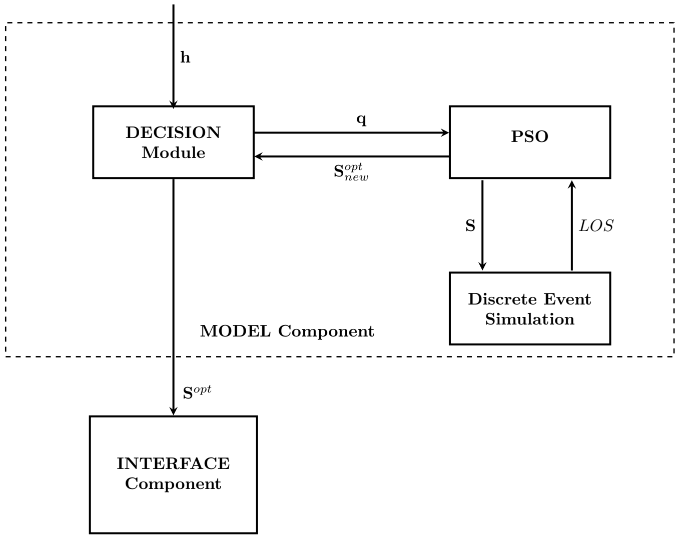

Figure 3.9: Interactions among the decision, optimization and simulation modules of the DSS model component.

3.5.2.3 Optimization Module Specification

3.5.2.3.1 Particle Swarm Optimization of the Thresholds

In the PSO algorithm, a number of components, called particles, are placed in the search space of the problem, and each of them evaluates the fitness (or objective) function at its current location. Each particle determines its movement through the search space by combining some aspects of the history of its own current and best positions with those of the nearest members of the swarm. The swarm as a whole moves close to an optimum of the fitness function. The swarm is composed by K particles, denoted by $\mathbf{p}_j$, $j = 1, .., K$, and each particle is composed of three

D-dimensional vectors (where D is the dimension of the search space) defined as follows:

$$\mathbf{p}_j = (\mathbf{ppos}_j, \mathbf{pbest}_j, \mathbf{pvel}_j) \tag{3.15}$$

where $\mathbf{ppos}_j$ is the current position of particle j, $\mathbf{pbest}_j$ is the best position reached so far by particle j, and $\mathbf{pvel}_j$ is the current velocity of particle j, which directs the movement of the particle.

The current position $\mathbf{ppos}_j$ is evaluated as a possible problem solution. If that position results to be better than the previous ones in terms of fitness function value, then its coordinates are stored in the vector $\mathbf{pbest}_j$. The position corresponding to the global best function obtained by any particle in the storm is stored in a variable called *global best*, denoted by $\mathbf{gbest}$. The objective of the algorithm is to move towards better positions and update $\mathbf{pbest}_j$ and $\mathbf{gbest}$ vectors. Moreover, the algorithm iteratively updates the velocity vector $\mathbf{pvel}_j$ and calculates new points by adding the $\mathbf{pvel}_j$ coordinates to $\mathbf{ppos}_j$.

In the present implementation, the current position $\mathbf{ppos}_j$ is the candidate incentives threshold vector $\mathbf{S}_j \in \mathbb{N}^N$. For each particle of the swarm, the simulation module is used to evaluate the *fitness* function LOS_j for the given value of $\mathbf{S}_j$.

The steps followed during the simulation-optimization campaign are summarised in Algorithm 1, that consists of five main phases.

1. **Initialize particles.** The PSO operates on K particles. Each particle has $D \cdot 3$ elements, where $D = N$, i.e., the number of parking areas. The K particles are initialized at a random generated values.
   The target value LOS^* to be reached is determined by the decision maker. If such value is not reached, the optimization process terminates after completing a maximum number of iterations ($MAXITER$).

2. **Calculate fitness values.** The fitness value LOS_j for each particle $\mathbf{p}_j$ is evaluated invoking the simulation module with $\mathbf{S}_j$ as input. Moreover, the current number of performed iterations ($numiter$) is updated.

Algorithm 1 Optimization-simulation procedure.

Phase 1 - PSO: Initialize particles

Set K, LOS^*, $MAXITER$ ▷ swarm size, minimum required LOS, maximum number of iterations

Set $LOS_{gbest} = 0$, $numiter = 0$ ▷ maximum value of LOS reached so far, current number of performed iteration

Set $\mathbf{K} = \{\mathbf{p}_j, j = 1, .., K\}$, where $\mathbf{p}_j = (\mathbf{S}_j, \mathbf{pbest}_j, \mathbf{pvel}_j)$ ▷ particle swarm, composed by $\mathbf{p}_j$ particles

for $j = 1 : K$ **do**

  Set randomly $\mathbf{S}_j$

  $\mathbf{p}_j = (\mathbf{S}_j, (0, 0, 0, 0, 0), (0, 0, 0, 0, 0))$

end for

Phase 2 - PSO: Calculate fitness values

$numiter = numiter + 1$

for $j = 1 : K$ **do**

  Simulate system behaviour

  $LOS_j = \text{getsol}(\text{simulation}(\mathbf{S}_j))$ ▷ system performance using as input the vector $\mathbf{S}_j$ of $\mathbf{p}_j$

end for

Phase 3 - PSO: Performances analysis

for $j = 1 : K$ **do** ▷ update $\mathbf{pbest}_j$

  if $LOS_j > LOS_{\mathbf{pbest}_j}$ **then**

    $\mathbf{pbest}_j = \mathbf{S}_j$

  end if

end for

$LOS_{\mathbf{gbest}} = \max\left(LOS_{\mathbf{gbest}}, \max_{j=1:K}\left(LOS_{\mathbf{pbest}_j}\right)\right)$ ▷ update **gbest**

$\mathbf{gbest} = \text{select}_{j=1:K}\ \{\mathbf{p}_j \text{ s.t. } LOS_j = LOS_{\mathbf{gbest}}\}$

Algorithm 1 Optimization-simulation procedure (continued).

Phase 4 - PSO: Stop criteria

```
if LOS_gbest ≥ LOS* or numiter >= MAXITER then                    ▷ stop criteria
    S^opt_new = S_gbest              ▷ optimal value of threshold S computed by PSO
    Return S^opt_new
    End of the optimization-simulation process                          ▷ EXIT
end if
```

Phase 5 - PSO: Particle swarm update

```
for j = 1 : K do
    pvel'_j = (φ1 · (pbest_j − S_j) + φ2 · (gbest − S_j))              ▷ new velocity
    S'_j = S_j + pvel'_j                                               ▷ new position
    p_j = ( S'_j, pbest_j, pvel'_j)      ▷ update position and velocity of particle j
end for
Go to Phase 2
```

3. **Performances analysis.** The fitness of each current position $\mathbf{S}_j$ is evaluated in order to determine how to move towards the optimum values. The best stored position $\mathbf{pbest}_j$ is updated for each particle. Moreover, the actual global best value of the LOS, $LOS_{\mathbf{gbest}}$, is computed and the corresponding particle position is stored in **gbest**.

4. **Stop criteria.** The optimization process is completed if one of the following stop criteria is reached: $LOS_{\mathbf{gbest}}$ is greater than LOS^* or $numiter = MAXITER$.

5. **Particle swarm update.** Each particle $\mathbf{p}_j$ is updated in order to reach potentially better fitness values: first the new velocity $\mathbf{pvel}'_j$ is computed, second the new position $\mathbf{S}'_j$ is obtained. The update of $\mathbf{pvel}_j$ uses two weights φ_1 and φ_2, with $\varphi_1 = c_1\mathbf{R}_1$ and $\varphi_2 = c_2\mathbf{R}_2$: c_1 and c_2 are *acceleration* coefficients, $\mathbf{R}_1$ and $\mathbf{R}_2$ are vectors of random values uniformly distributed in the interval $[0, 1]$. The expression $\varphi_1 \cdot \left(\mathbf{pbest}_j - \mathbf{S}_j\right)$ is called *cognitive component* and

represents the tendency of the particles to move towards its best position, while the expression $\varphi_2 \cdot (\mathbf{gbest} - \mathbf{S}_j)$ is the *social component* and represents the attraction of the particle towards the position associated to the global best value [Kennedy and Eberhart, 1995].

3.5.3 Case Study

This Section describes the application of the DSS for solving the relocation problem of a CS system designed for Trieste, a city in the north of Italy. Considering the dimension of the town and the necessary services, five parking areas are proposed and positioned in strategic locations. In particular, in the following the CS system and the parking areas are specified for the simulation model.

3.5.3.1 CS Service Specification

- **Time unit**: the minute is considered as t.u..
- **Number of parking areas**: $N = 5$.
- **Fleet size**. A total number of vehicles $V = 20$ is considered.
- **Fleet composition**. Both traditional vehicles and EVs are considered.
- **Service hours**. CS service is operative for 16 hours per day, 30 days per month.
- **Daily customer demand**. Three levels of demand characterized by different inter-arrival times are considered, corresponding to different levels of demand during a typical day: *high* (μ_h minutes), *medium* (μ_m minutes) and *low* (μ_l minutes).
- **Routing**. The following matrix is determined by considering the proposed locations for the five parking areas:

$$\mathbf{R} = \begin{bmatrix} 0.08 & 0.20 & 0.32 & 0.10 & 0.30 \\ 0.15 & 0.05 & 0.35 & 0.25 & 0.20 \\ 0.23 & 0.23 & 0.08 & 0.23 & 0.23 \\ 0.18 & 0.25 & 0.20 & 0.02 & 0.35 \\ 0.15 & 0.30 & 0.20 & 0.30 & 0.05 \end{bmatrix}$$

In particular, $r_{ij} \ll 1$ means that there is a low probability that a car rented in the parking area i is dropped off in the parking area j. On the other hand, if $r_{ij} \cong 1$ then there is a high probability that a car rented in the parking area i is dropped off in the parking area j.

- **Maximum waiting time**. It is assumed that, if a user can not rent a vehicle within 10 minutes from his arrival to a parking area, he will leave the system without being served.

- **Vehicles' maintenance and EVs charging operations**. The 10% of the total number of rented vehicles needs maintenance operations after the rental period. Moreover, among them, the 99% are available again at the parking area after 1 hour, while the remaining need an 8-hour service. In case of EVs, such maintenance operations represent the necessary charging operations.

- **Service and rental times**. The times associated to the vehicle rental operations, the maintenances and the charging operations, as well as the length of the rental period, have triangular distribution. Indeed, it is reasonable to consider times centered around a most likely value, avoiding extreme and unrealistic values.

- **User acceptance probability**. The maximum value of the probability that a user accepts the new destination is assumed equal to $\vartheta = 0.50$.

- **Degree of Affinity**. The affinity matrix $\mathbf{A}$ associated to the considered parking areas is the following:

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 & 2 & 1 \\ 2 & 0 & 1 & 3 & 4 \\ 3 & 2 & 0 & 3 & 2 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 1 & 2 & 0 \end{bmatrix}$$

The elements a_{ij} of $\mathbf{A}$ are determined considering two aspects: the distance between stations i and j and the possibility of using quick and reliable public transport means between the two stations.

- **System monitoring**. The status of the system is monitored every 10 minutes in order to determine the incentive activation status.

3.5.3.2 Simulation Specification

With the aim of assessing the effectiveness of the proposed approach, a set of scenarios is considered: in each test, the estimates of the service performances are deduced by a simulation campaign of 100 independent replications, with a 95% confidence interval, whose half width is about 1.4% in the worst case. The length of each replication is 960 minutes (i.e., a complete working day is simulated), with a transient period of 30 minutes.

In order to identify the best number of particles for the PSO implementation, a set of different sizes of the swarm have been tested. Such tests showed that, in the proposed case study, the results do not improve using a swarm of size greater than 10 particles. Therefore, in the considered test case, the PSO algorithm runs with $K = 10$ particles. Moreover, c_1 and c_2 are both set to 2, as suggested in the related literature [Kennedy and Eberhart, 1995].

Two different models to describe the user demand are considered: deterministic and stochastic interval times between customer arrivals. The deterministic model is proposed as a benchmark for the incentive approach, and can be used to estimate a typical value for the thresholds based on historical data. On the other hand, the stochastic scenarios take into account more realistic demand behaviour and random variation among the user inter arrival times.

3.5.3.3 Simulation Results

In the following, the results of different simulation campaigns are presented.

3.5.3.3.1 Effects of the Incentive Mechanism

In order to assess the impact of the proposed incentive mechanism, five scenarios (denoted by A,B,C,D and E), characterized by different service fleet sizes and different inter-arrival times μ_h, μ_m, and μ_l, are considered. Each scenario is studied in two cases: deterministic and stochastic inter-arrival times.

Table 3.10 reports the inter-arrival times expressed in minutes: in the case of deterministic demand the values are the deterministic inter-arrival times; in the case of stochastic demand the average values of the exponential distribution of the inter arrival times are reported.

The values of the LOS are determined by the simulation in three Operative Conditions (OC):

1. the incentives are not applied (LOS_{ni});
2. the incentives are applied with the thresholds $\mathbf{S}^{av}$ equal to the average number of vehicles available in the system (LOS_{av}) (a-priori set-point);
3. the incentives are applied with optimized thresholds $\mathbf{S}^{PSO}$ obtained by Algorithm 1 (LOS_{PSO}).

Table 3.11 reports the 5-elements vectors $\mathbf{S}^{av}$ and $\mathbf{S}^{PSO}$ for the OC 2) and 3). Moreover, Table 3.11 shows the values of the LOS obtained in the three simulated

Table 3.10: Scenarios.

Scenario	**Fleet size**	**Demand**		
		μ_h	μ_m	μ_l
A	20	12	20	60
B	40	6	10	30
C	60	4	6	15
D	80	2.5	5	10
E	100	2	2.5	5

Table 3.11: Tests for Incentive Mechanism Evaluation.

Scenario		**System LOS**			**S**	
		LOS_{ni}	LOS_{av}	LOS_{PSO}	$\mathbf{S}^{av}$	$\mathbf{S}^{PSO}$
A	deterministic demand	0.65	0.72	0.76	$[4\,4\,4\,4\,4]^T$	$[1\,4\,2\,2\,1]^T$
	stochastic demand	0.61	0.68	0.70		$[1\,1\,2\,1\,1]^T$
B	deterministic demand	0.65	0.73	0.76	$[8\,8\,8\,8\,8]^T$	$[4\,4\,1\,3\,3]^T$
	stochastic demand	0.63	0.71	0.73		$[1\,3\,4\,4\,3]^T$
C	deterministic demand	0.68	0.75	0.80	$[12\,12\,12\,12\,12]^T$	$[3\,9\,2\,3\,4]^T$
	stochastic demand	0.67	0.74	0.77		$[7\,6\,10\,3\,7]^T$
D	deterministic demand	0.63	0.70	0.74	$[16\,16\,16\,16\,16]^T$	$[5\,7\,3\,1\,2]^T$
	stochastic demand	0.62	0.68	0.72		$[3\,6\,2\,0\,2]^T$
E	deterministic demand	0.69	0.73	0.76	$[20\,20\,20\,20\,20]^T$	$[1\,4\,2\,2\,1]^T$
	stochastic demand	0.68	0.72	0.75		$[1\,2\,2\,2\,1]^T$

operative conditions and in the five scenarios with stochastic and deterministic interval times: the LOS is low when no control is applied; the LOS increases if a control rule based on the incentives is applied; the application of the optimization-simulation procedure leads to a LOS increase of about 5% compared to the case without optimized thresholds.

What is worth noting is that in each scenario the values of the thresholds determined by the PSO are significantly lower than the mean number of available vehicles: this is due to the fact that in the OC 3) the thresholds are not determined a-priori but on the basis of the customers' preferences and the relative locations of the parking areas.

Table 3.12: Average Fraction of Time during which the Incentives are active.

Scenario		Incentive activation		
		t^{av}	t^{PSO}	δ
A	deterministic demand	0.67	0.49	27%
B	deterministic demand	0.77	0.56	27%
C	deterministic demand	0.87	0.65	25%
D	deterministic demand	0.84	0.66	21%
E	deterministic demand	0.88	0.74	16%

In order to enlighten the consequences of these results, the following additional performance indexes are determined and compared in Table 3.12:

$$t^{av} = \frac{\text{average time during which the incentives are active in OC 2)}}{\text{working day duration}}, \tag{3.16}$$

$$t^{PSO} = \frac{\text{average time during which the incentives are active in OC 3)}}{\text{working day duration}}, \tag{3.17}$$

$$\delta = (1 - \frac{t^{PSO}}{t^{av}})100. \tag{3.18}$$

It is apparent that the period of activation of the incentives is significantly reduced, and this leads to economic benefits for the CS company, which obtains a better LOS while incentivizing a lower number of customers.

As Fig. 3.10 highlights, the application of the incentive mechanism with the threshold determined by the simulation-optimization procedure leads to a LOS increase of about 16% in all the cases and the stochastic demand does not affect the effectiveness of the solution. The observed LOS increase is coherent with the values typically observed in the related literature, both for user-based and operator-based policies [Nourinejad and Roorda, 2015], [Nourinejad and Roorda, 2014], [Alfian et al., 2014], [Bianchessi et al., 2013], [Bruglieri et al., 2014].

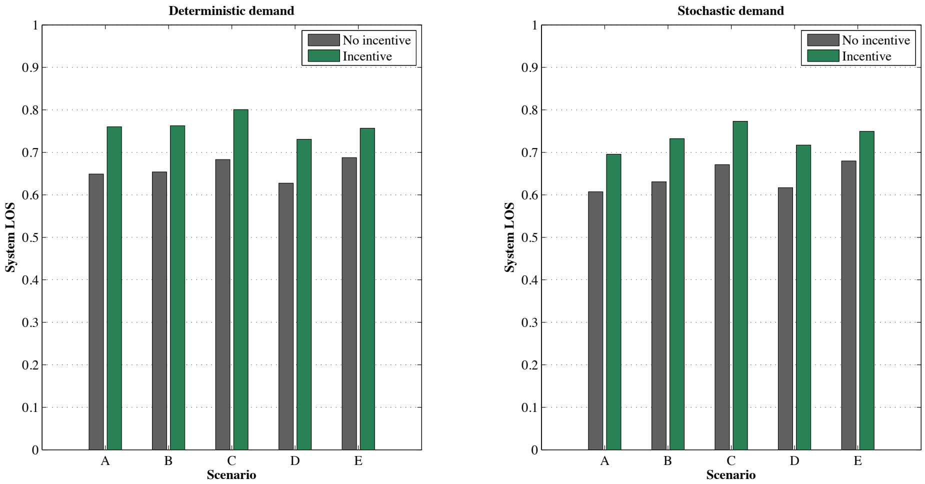

Figure 3.10: System LOS before and after the application of the incentive with optimized thresholds.

3.5.3.3.2 Sensitivity Analysis about Acceptance Variation

In order to assess the robustness of the proposed solution to the customers' acceptance variation, the optimal incentive configuration identified by the PSO for scenario A (both in deterministic and stochastic cases) is considered with $0.30 \leq \vartheta \leq 0.80$. Fig. 3.11 points out that, even in the worst case, i.e., for $\vartheta = 0.30$, there is a LOS increase of about 8% under both deterministic and stochastic demand assumptions.

3.5.3.3.3 Discussion about of the Proposed Solution

The effectiveness of the proposed solution relies on the fleet size in relation with the demand. In order to highlight such a behavior, a fleet of 20 vehicles, as in the Scenario A, is considered and the demand is gradually increased as described in Tab. 3.13. Fig. 3.12 points out that the incentive mechanism is very effective if the fleet size is adequate with the demand and, obviously, the benefit decreases if the demand increases too much.

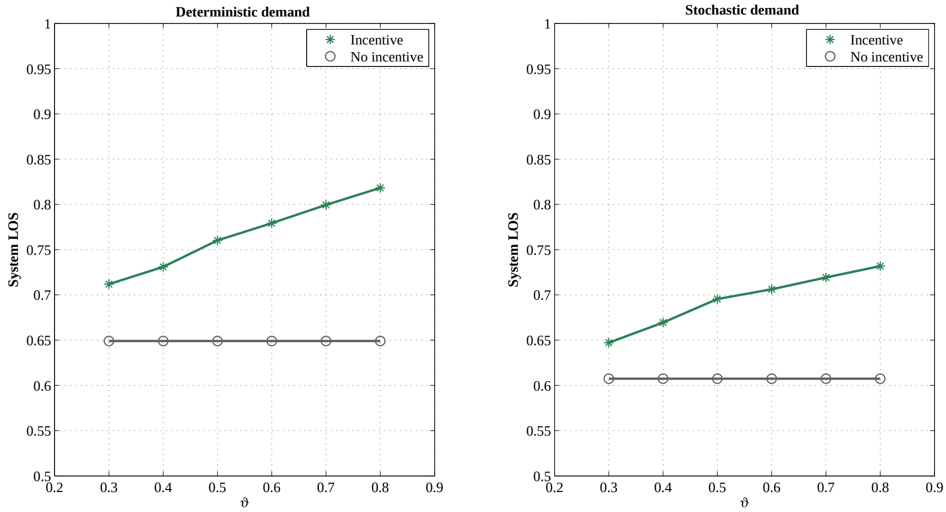

Figure 3.11: Sensitivity analysis about the acceptance variation.

Table 3.13: Scenarios with a fleet size of 20 vehicles.

Scenario	**Demand**		
	μ_h	μ_m	μ_l
AA	12	20	60
AB	10	15	30
AC	6	10	15
AD	3	6	10
AE	2.5	3	6
AF	2	3	4

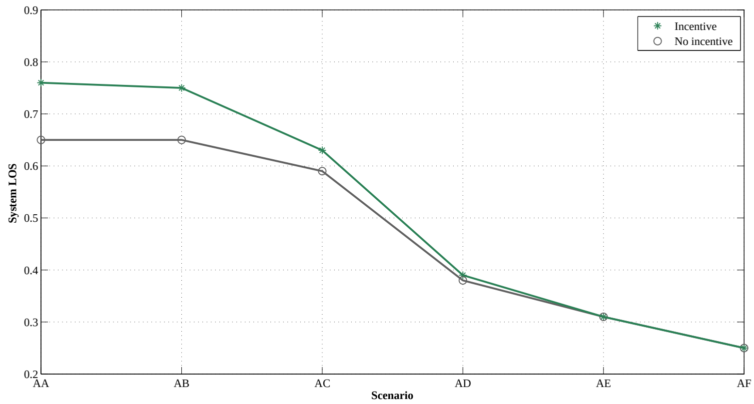

Figure 3.12: Dependency of the incentive mechanism effectiveness on the coherent fleet sizing.

Moreover, comparing the proposed DSS with the systems presented in the related literature [Kek et al., 2009], [Nourinejad and Roorda, 2014], [Nourinejad and Roorda, 2015], two main differences are pointed out: i) the presented DSS considers a user-based vehicle relocation strategy based on the optimization of the selected performance index; ii) the simulation is used in a closed-loop strategy to optimize the performance index and it is not only a mean to evaluate the performances; iii) the proposed relocation strategy is applied in closed-loop on the basis of the system state knowledge.

3.6 Concluding Remarks

In this Chapter, the user-based vehicle relocation problem has been addressed through a DSS approach. In particular, two distinct problems have been considered: the a-priori set-point problem and the optimized set-point problem.

The contributions of this Chapter are the following:

1. a taxonomy for the vehicle relocation problem;

2. the formalization of a CS system model by a UML description;

3. the formalization of a CS system model in a TPN framework for the a-priori set-point problem;

4. the formalization of a CS system model in a DES framework for the optimized set-point problem;

5. the development of a discrete-event simulator to mime the CS system dynamics, taking into account the users behaviour;

6. a methodology for determining the values of the thresholds in the optimized set-point problem based on discrete event simulation and PSO algorithm.

In particular, for the a-priori set-point problem, the obtained results have underlined that a system of economic incentives which does not consider the instantaneous balance conditions of the service and which suggests always to customers to return the rented vehicle as soon as possible is not a solution for the imbalance problem, leading to economic losses for the CS organization. On the other hand, a simple ICT application and the real time monitoring of the system can increase the number of served users and, therefore, improve the overall service performance. However, the effectiveness of this solution decreases as the congestion level of the system grows, and this fact underlines that such an action is not able to overcome problems linked to a service undersized in terms of number of vehicles initially made available in each station.

On the other side, for the optimized set-point problem, the results have shown that the economic incentives allow an effective relocation and can be used to improve the system LOS even in the case of nearly saturated offer. Moreover, the application of optimized thresholds leads to LOS increase of about 5% compared to the case without optimized thresholds and, above all, the period of activation of the incentives is significantly reduced, since the values of the thresholds determined by the PSO are significantly lower than the mean number of available vehicles: this leads to economic benefits for the CS company, which obtains a better LOS while incentivizing a lower number of customers.

The results of this Chapter are based on publications [Clemente et al., 2013c], [Clemente et al., 2013a], [Clemente et al., 2013b], [Clemente et al., 2015], [Clemente et al., 2016b].

Chapter 4

Electric Vehicles Smart Charging Management

In this Chapter, the smart management of Electric Vehicles charging operations is considered and the requirements of a DSS useful to handle this problem are analysed.

4.1 Motivation

In recent years, EVs are receiving a lot of attention due to their potential to alleviate both the growing environmental problems in urban areas and the reliance of mobility on fossil fuels. The foreseen widespread diffusion of such an alternative technology will introduce some challenges that have to be considered as of now in order to guarantee an effective and reliable deployment. In particular, the integration of the EVs with the power system, especially at the distribution level, is a central issue. Indeed, the demand of electrical power for the EVs charging operations could lead to severe grid disruptions, with extra power losses and voltage deviations, if a proper coordination of such operations is not achieved (the so-called *dumb charging*).
The identification of the optimal *smart charging strategy* is a problem that involves two classes of actors with different requirements and, often, conflicting objectives.

Indeed, for the EVs drivers it is essential to meet the same level of flexibility they are used to with the traditional vehicles, easily refillable wherever and whenever they want. On the other side, from the electric grid operators point of view, the minimization of the impact of EVs charging on the power system is fundamental: to this aim, it would be suitable to defer such operations to *off-peak* hours and to shift them to areas characterized by low electricity demand in order to avoid transformer overloads. The cooperation between these two actors is, therefore, crucial and the modern Smart Grids (SGs) will enable it: a SG is, indeed, a *Cyber-Physical System* (CPS) in which the tight combination of cyber and physical components guarantees communication, control and computation directly inside the system [Jin et al., 2013]. The continuous bidirectional real time information flow between users and suppliers allows to optimize the operation of the grids while taking into account users requirements.

4.1.1 EVs Smart Charging Management Background

Literature contributions to the *EVs charging management problem* can be categorized on the basis of the considered *optimization strategy*, the *timing of the control* and the *paradigm of the control*.

Different *optimization strategies* are considered to determine the optimal charging profile for the EVs: *grid requirements*, *drivers utility* or both. When grid requirements are taken into account, the most common objective function is represented by the minimization of power losses and voltage deviations and the flattening of the overall load profile during the day is sought ([Andreotti et al., 2012], [Clement-Nyns et al., 2010], [Li et al., 2011]). At the same time, the ability of the EVs to provide a number of ancillary services and, so, contribute to the integration of the Distributed Generation (DG) into the grid is widely analyzed ([Clement-Nyns et al., 2011], [Vandael et al., 2011]).

On the other hand, the maximization of the users utility turns into a spatial assignment and a temporal scheduling of the charging operations able to minimize a given cost function, e.g., the total waiting time or the total charging cost ([Gharbaoui et al., 2012], [Qin and Zhang, 2011], [Xu and Pan, 2012]).

The *timing of the control* could be based on forecasts of energy and travel demand and in this case a *day-ahead planning* is possible,([Gan et al., 2012]); on the other side, when *real-time control* is considered, a continuous monitoring of the system conditions is required.([Peng et al., 2012], [Li et al., 2011]).

Finally, the *paradigm of the control* can be centralized or distributed. In particular, in the centralized approach a central controller determines the optimal charging profile for a population of EVs on the basis of the grid conditions. However, this solution is computationally efficient only for a limited number of EVs, since a large amount of information and a remarkable communication effort are required ([Clement-Nyns et al., 2010], [Xu and Pan, 2012]).
Alternatively, in a distributed control scheme EVs themselves calculate their charging schedules, for example responding to a price signal broadcast by the grid operators in order to influence users behavior ([Gan et al., 2013], [Karfopoulos and Hatziargyriou, 2013], [Jin et al., 2013]), or each charging station determines which EVs recharge and when operate the charging activities by negotiating with a set of neighbor stations ([Qin and Zhang, 2011]).

4.2 Objectives

The aim of this Chapter is to identify the requirements of a DSS enabling a smart management strategy for the coordination of the EVs charging operations. In particular, charging operations involving the public charging infrastructure are considered.
The approach traditionally followed in literature in order to take into account simultaneously drivers and grid requirements is to determine the minimum cost charging

profile for each EV on the basis of a control signal broadcast by the grid operators. However, the assignment of the vehicles to the best available charging station is a problem handled separately from grid concerns.

Therefore, the objective of this Chapter is to identify a tool that allows to solve such a resources allocation problem considering not only the traditional assignment and capacity constraints, but also grid requirements. In our idea, this turns into a time-varying configuration of the considered public charging infrastructure. In particular, maximum charging power and energy price at the different available stations throughout the day are settled in order to influence drivers behaviour and so minimize power losses and voltage deviations on the grid. Then, on the basis of such time varying parameters, EVs are assigned to the charging stations while maximizing drivers utility. For this purpose, a hierarchical bi-level decision structure is introduced: the upper-level optimization problem deals with the optimal charging infrastructure configuration, while the lower-level problem handles the allocation of the charging stations to the EVs.

The remainder of the Chapter is structured as follows. In Section 4.3 the structure and the assumptions of a DSS for the EVs Smart Charging Problem are presented. In Section 4.4 the optimization module for the *Vehicle-to-Charging Station Assignment Problem* (VCSA) is formalized. Finally, Section 4.5 summarizes the remarks and the contributions of the present Chapter.

4.3 DSS for the EVs Smart Charging Problem

4.3.1 Problem Statement

In this Chapter the problem of coordinating the daily charging operations of a fleet of EVs in an urban area by optimally assigning each vehicle to a charging station and identifying the optimal charging period for each driver is considered.

The following assumptions are made:

- *electric grid*: a smart grid is considered and bidirectional communication capabilities between the single EV and a system operator are supposed;
- *type of charging*: only charging requests involving public charging stations are taken into account, while home-recharges are not considered in this context;
- *charging stations*: each charging station is equipped with one or more charging outlets and its charging power and energy cost are time-varying;
- *EVs*: when it needs to be recharged, each vehicle is able to communicate its position, its battery residual state of charge, when it desires to start the recharge (i.e., *EV release time*) and the time within which it wants to leave the charging infrastructure (i.e., *EV deadline*);
- *charging operations*: incomplete recharges are admitted, but charging operations of a vehicle cannot be interrupted and restarted later (i.e., no *preemption* is allowed). Moreover, the charging cannot start before the stated release time and it must be interrupted within the specified deadline;
- *charging fees*: it is considered that the total charging monetary cost paid by each driver depends on the unit energy price characterizing the assigned charging station at the time interval during which the charging starts.

4.3.2 DSS Formalization

Fig. 4.1 shows the structure of the DSS and the interactions among its components. In particular:

- *Interface Component.* This component has to communicate with different actors of the systems in order to obtain the data necessary for the decision procedure. In particular, it has to: receive the charging requests from the EVs;

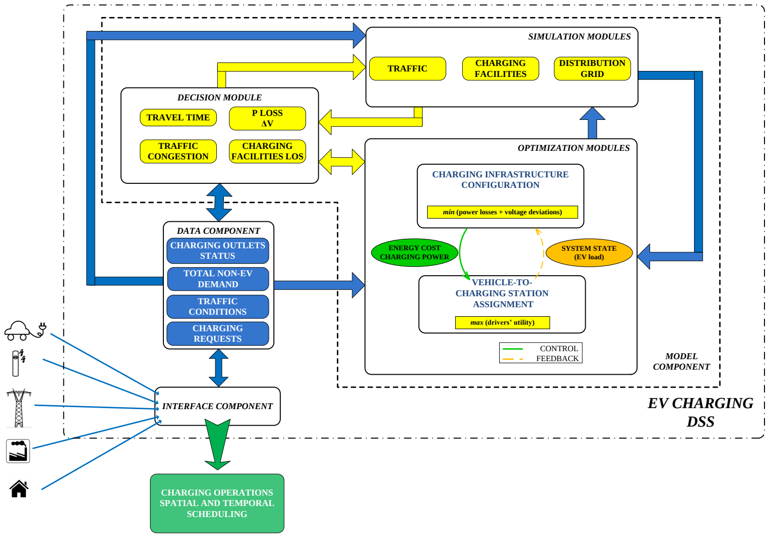

Figure 4.1: DSS for the EVs Smart Charging Problem.

monitor the distribution grid status in terms of non-EV loads (domestic loads, industrial loads, and so on); check the status of the charging infrastructure in order to know if there are anomalies or malfunctions; analyse the urban traffic conditions.

- *Data Component.* This component stores all the data collected by the interface and, whenever it is necessary, it provides them to the Simulation and the Optimization Modules. The basic data required are: the charging outlets status; the traffic conditions; the total non-EV load; the charging requests.

- *Model Component.*

  1. *Decision Module.* This module has to decide when a new optimization procedure has to be triggered and to assess the impacts of the management rules proposed by the optimization modules by the evaluation of some KPIs. For the problem of the EV charging operations management such indexes could be defined as follows: the charging infrastructure LOS,

expressed in terms of the number of charging requests that it is able to satisfy; the traffic congestion, i.e., a measure of the impact that the assignment of the EVs to the charging stations has on the overall urban area traffic; the average travel time that the EV drivers have to face in order to reach a charging station; the distribution grid power losses (P LOSS) and the voltage deviations (ΔV).

2. *Simulation Module.* Three main simulation modules have to be defined in this case: one for the electric distribution grid, one for the charging facilities, and, finally, a traffic simulator. The outputs of these simulation modules are the KPI useful to the decision module in order to determine if to apply the solution proposed by the optimization module.

3. *Optimization Module.* In order to consider both the drivers and the grid operators point of view, the EV charging operation problem is outlined as the interplay of two different decision-makers who act sequentially and whose choices are mutually dependant: therefore, two optimization structures coupled through a *leader-follower approach* are proposed. In particular, the **upper-level optimization** (*Charging Infrastructure Configuration Problem*) is referred to the grid operators and determines the optimal charging power and energy price at the different charging stations in order to spatially and temporally reshape the energy demand and minimize power losses and voltage deviations. In order to determine the optimal solution, this module should take into account not only the load derived from the EV charging operations, but also the traditional non-EV loads. The action of the upper-level optimizator turns out in a time-varying configuration of the charging infrastructure. On the other side, the **lower-level optimization** (*Vehicle-to-Charging Station Assignment Problem*) assigns optimally the EVs to the available charging stations in order to maximize a given users utility function. The value

of such a function should depend on the parameters determined by the upper-level problem as well as on the current traffic conditions, which influence the time required for a driver to reach the assigned station.

Once the Vehicle-to-Charging Station Assignment Problem has been solved, the resulting system state should be communicated to Charging Infrastructure Configuration Problem, which will update accordingly its strategy: therefore, the optimization problems are alternately iterated. Hence, from the interaction between the two levels, a dynamic configuration of the charging infrastructure and a spatial and temporal scheduling of the charging operations results.

This problem can be handled both in a centralized and in a decentralized approach: in the first case a Mixed Integer Linear Programming (MILP) formulation can be introduced (see Section 4.4), while in the second one a consensus framework in which each charging station determines what EV to recharge can be developed [Fanti et al., 2014]. The first approach is able to handle a more detailed and realistic model, however for great instances the distributed approach is preferable since it requires a lower computational effort.

4.4 Vehicle to Charging Station Assignment Problem

The objective pursued in solving the Vehicle to Charging Station Assignment Problem (VCSA) is the maximization of the EVs drivers utility: to this aim, an *users cost function* made up of 4 different entries is considered:

1. the *waiting time* for the charging;

2. the *charging monetary cost*, meant as the unit energy price that each driver

has to pay for the charge;

3. the *distance* that the driver has to go through to reach the assigned charging station;

4. the *penalty for incomplete charging*, i.e., a quantification of the users' annoyance resulting from leaving the charging station with a not fully charged vehicle.

Moreover, in the proposed model a linear combination of such functions is considered (single-objective formulation).

4.4.1 Mathematical Formulation

A Time Indexed Formulation (TIF) is considered: the planning horizon is discretized into T time intervals, each lasting Δ time units. Each time interval starts at time $t-1$ and ends at time t, i.e., we consider the time periods $1, 2, \ldots, T$: hereafter, the time interval is denoted with its ending time t.
In order to describe the Mixed Integer Linear Programming (MILP) formulation, the following notation is introduced.
Numerical Sets

- $\mathbb{R}^+$: set of all positive real numbers
- $\mathbb{R}^+_0$: set of all real numbers including 0
- $\mathbb{N}^+$: set of all positive natural numbers

Sets

- $\mathcal{V} = \{1, 2, \ldots, N\}$: set of *charging stations*
- $\mathcal{U}_k = \{1, 2, \ldots, M\}$: set of *EVs* that make a charging request during the iteration k of the optimization problem

- $\mathcal{T} = \{1, 2, \ldots, T\}$: set of *time periods.*

Parameters

1. *Charging Stations Parameters*

   Each charging station $n \in \mathcal{V}$ is characterized by:

   - $cost_n^t \in \mathbb{R}^+$: unit charging cost at charging station n during time interval t
   - $p_n^t \in \mathbb{R}_0^+$: charging power at charging station n during time interval t
   - $r_n \in \mathbb{R}_0^+$: maximum number of charging outlets available at charging station n.

2. *EVs Parameters*

   Each EV $m \in \mathcal{U}_k$ is characterized by:

   - $t_m^{min} \in \{1, \ldots, T\}$: release time of vehicle m
   - $t_m^{max} \in \{1, \ldots, T\}$: deadline of vehicle m
   - $cap_m \in \mathbb{R}^+$: battery capacity of vehicle m
   - $res_m^0 \in \mathbb{R}^+$: residual battery state of charge (SoC) of vehicle m when it makes its charging request
   - $f_m \in \mathbb{R}^+$: energy consumption per unit distance of vehicle m
   - $v_m \in \mathbb{R}^+$: average speed of vehicle m
   - $\eta_m \in [0, 1]$: charging efficiency of vehicle m
   - $d_{n,m} \in \mathbb{R}_0^+$: distance between charging station $n \in \mathcal{V}$ and vehicle m, when the vehicle makes its charging request.

3. *Model Parameters*

- $B \in \mathbb{N}^+$: a sufficiently large integer.

Decision Variables

For each charging station $n \in \mathcal{V}$ and EV $m \in \mathcal{U}_k$, the following decision variables are defined:

$$y_{n,m} = \begin{cases} 1 & \text{if } m \text{ is assigned to } n \\ 0 & \text{otherwise} \end{cases}$$

$$h^t_{n,m} = \begin{cases} 1 & \text{if the charging of } m \text{ at } n \\ & \text{starts during time interval } t \\ 0 & \text{otherwise} \end{cases}$$

$$w^t_{n,m} = \begin{cases} 1 & \text{if } m \text{ is being charged at } n \\ & \text{during time interval } t \\ 0 & \text{otherwise} \end{cases}$$

$$s_{n,m} \in \mathbb{N}^+ = \text{time interval during which the charging of } m \text{ at } n \text{ starts.}$$

Moreover, the following time indexed variables describing the state of each EV $m \in \mathcal{U}_k$ at time interval t are defined:

$$e^t_m \in \mathbb{R}^+_0 = \text{amount of energy received by vehicle } m \text{ during time interval } t;$$

$$res^t_m \in \mathbb{R}^+_0 = \text{residual battery SoC of vehicle } m \text{ at the beginning of time interval } t;$$

$$o^t_m \in \mathbb{R}^+_0 = \text{maximum amount of energy that vehicle } m \text{ could receive during time interval } t;$$

$q_m^t \in \mathbb{R}_0^+ =$ amount of energy requested by vehicle m at the beginning of time interval t.

Finally, the following auxiliary decision variable for each $m \in \mathcal{U}_k, t \in \mathcal{T}$ is introduced:

$$x_m^t = \begin{cases} 1 & \text{if } q_m^t \leq o_m^t; \\ 0 & \text{otherwise.} \end{cases}$$

The problem can be formulated as follows:

$$\text{minimize } z = \sum_{i=1}^{4} (\alpha_i \cdot z_i) \tag{4.1}$$

where

$$\alpha_i \in [0, 1] \tag{4.2}$$

$$z_1 = \sum_{m=1}^{M} \left(\sum_{n=1}^{N} \big(s_{n,m} - t_m^{min} \cdot y_{n,m} \big) \right) \tag{4.3}$$

$$z_2 = \sum_{m=1}^{M} \left(\sum_{n=1}^{N} \sum_{t=1}^{T} \big(cost_n^t \cdot h_{n,m}^t \big) \right) \tag{4.4}$$

$$z_3 = \sum_{n=1}^{N} \sum_{m=1}^{M} \big(d_{n,m} \cdot y_{n,m} \big) \tag{4.5}$$

$$z_4 = \sum_{m=1}^{M} \left(cap_m - \big(res_m^T + e_m^T \big) \right) \tag{4.6}$$

s.t.

$$\sum_{n=1}^{N} y_{n,m} = 1 \qquad \forall m \in \mathcal{U}_k \tag{4.7}$$

$$d_{n,m} \cdot f_m \cdot y_{n,m} \leq res_m^0 \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.8}$$

$$s_{n,m} = \sum_{t=1}^{T} t \cdot h_{n,m}^t \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.9}$$

$$\sum_{n=1}^{N} \sum_{t=1}^{T} h_{n,m}^t = 1 \qquad \forall m \in \mathcal{U}_k \tag{4.10}$$

$$s_{n,m} \geq t_m^{min} \cdot y_{n,m} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.11}$$

$$s_{n,m} \leq T \cdot y_{n,m} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.12}$$

$$s_{n,m} \geq \left\lceil \frac{d_{n,m}}{v_m \cdot \Delta} \right\rceil \cdot y_{n,m} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.13}$$

$$\sum_{m=1}^{M} w_{n,m}^t \leq r_n \qquad \forall n \in \mathcal{V}, t \in \mathcal{T} \tag{4.14}$$

$$t \cdot w_{n,m}^t \leq t_m^{max} \cdot y_{n,m} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.15}$$

$$\sum_{t=1}^{T} w_{n,m}^t \geq y_{n,m} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.16}$$

$$w_{n,m}^t \leq q_m^t \cdot B \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.17}$$

$$cap_m \geq res_m^t \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.18}$$

$$res_m^t = \begin{cases} res_m^0 - \sum_{n=1}^{N} \big(d_{n,m} \cdot f_m \cdot y_{n,m} \big) & \forall m \in \mathcal{U}_k, t = 1 \\ res_m^{t-1} + e_m^{t-1} & \forall m \in \mathcal{U}_k, t \neq 1 \end{cases} \tag{4.19}$$

$$o_m^t = \sum_{n=1}^{N} \big(p_n^t \cdot \Delta \cdot \eta_m \cdot w_{n,m}^t \big) \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.20}$$

$$q_m^t = cap_m - res_m^t \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.21}$$

$$T + s_{n,m} \geq t \cdot w_{n,m}^t + T \cdot w_{n,m}^t \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t = 1 \tag{4.22}$$

$$s_{n,m} - T \leq t \cdot w_{n,m}^t - T \cdot w_{n,m}^t \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t = 1 \tag{4.23}$$

$$T + s_{n,m} \geq t \cdot w_{n,m}^{t} - T \cdot w_{n,m}^{t-1} + T \cdot w_{n,m}^{t} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t \neq 1 \tag{4.24}$$

$$s_{n,m} - T \leq t \cdot w_{n,m}^{t} + T \cdot w_{n,m}^{t-1} - T \cdot w_{n,m}^{t} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t \neq 1 \tag{4.25}$$

$$e_m^t \leq q_m^t \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.26}$$

$$e_m^t \leq o_m^t \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.27}$$

$$cap_m + e_m^t \geq q_m^t + cap_m \cdot x_m^t \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.28}$$

$$cap_m + e_m^t \geq o_m^t + cap_m \cdot (1 - x_m^t) \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.29}$$

$$y_{n,m}, h_{n,m}^t, w_{n,m}^t, x_m^t \in \{0,1\} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k, t \in \mathcal{T} \tag{4.30}$$

$$s_{n,m} \in \mathbb{N} \qquad \forall n \in \mathcal{V}, m \in \mathcal{U}_k \tag{4.31}$$

$$e_m^t, res_m^t, o_m^t, q_m^t \geq 0 \qquad \forall m \in \mathcal{U}_k, t \in \mathcal{T}. \tag{4.32}$$

The objective function (4.1) represents the total assignment cost and, as mentioned in the previous section, is a linear combination of 4 different functions: (4.3) is the *total waiting time*, expressed as the difference between the *release time* specified by each driver and the effective starting time of the charging operations; (4.4) is the *monetary cost* associated to the recharges, formulated as the sum of the unit energy prices that each driver has to pay for the charging; (4.5) expresses the *distance* between the EVs and the assigned charging station; finally, (4.6) is the *penalty for incomplete charging*. The weights (4.2) are used to combine such heterogeneous quantities in a generalized cost function and vary between 0 and 1.

Constraints (4.7) ensure that each EV is assigned to one and only one charging station, while constraints (4.8) impose that a vehicle can be assigned to a certain facility only if its initial residual battery SoC is sufficient to reach it. Constraints (4.9) describe the relationship between the decision variable expressing the time interval during which the charging operation starts and the binary variable $h^t_{n,m}$, while constraints (4.10) ensure that for each vehicle there is only 1 charging start interval. Constraints (4.11) $\div$ (4.13) specify the feasible values for the charging starting time: in particular, (4.11) impose that the charging operations of a specific EV cannot start before the stated *release time*, (4.12) ensure that such a value can be different from zero only if the considered vehicle has been assigned to that specific charging station and, finally, (4.13) take into account the time required by the EV to reach the assigned facility. (4.14) are the charging stations capacity constraints, while constraints (4.15) guarantee that charging operations of each vehicle end within the specified deadline. (4.16) ensure that an assigned vehicle is effectively recharged, while constraints (4.17) impose that an EV seizes a charging station only if it still needs to be charged. (4.18) impose that the battery capacity of each vehicle is not exceeded during the charging operations and only the required amount of energy is supplied; constraints (4.19) describe the update rules of the vehicle residual SoC, (4.20) express the maximum possible amount of energy that a certain vehicle can receive during a time interval and (4.21) describe the amount of energy requested by each EV at each time interval. Constraints (4.22) and (4.23) and constraints (4.24) and (4.25) express the relationship between decision variables $s_{n,m}$ and $w^t_{n,m}$ for $t = 1$ and for $t \neq 1$, respectively. Constraints (4.26) $\div$ (4.29) ensure that the amount of energy received by each vehicle during a specific time interval is equal to the minimum value between the amount of energy requested by such a vehicle and the maximum possible amount of energy that the charging station it has been assigned to can supply to it.

4.4.2 Tests

In order to prove the effectiveness of the proposed MILP model in providing the optimal solution considering different users utility functions, some tests are performed.

To solve them, the parameters that have to be determined by the upper-level problem (i.e., p_n^t, $cost_n^t$) are assumed as given.
A planning horizon of 12 hours discretized into 48 time intervals, each lasting 15 minutes, is considered. The objective is to optimally assign a population of 50 EVs to a set of 5 charging stations and to determine the optimal charging operations scheduling.

The main parameters of the model are based on typical values from the related literature and settled as listed in Tab. 4.1. Furthermore, in order to take into account the effects of the traffic on the time required to reach the assigned charging station, vehicles are characterized by values of speed typical of the urban areas.

Three different cases, characterized by different assignment policies, are taken into account by varying the values of the objective function weights α_i (4.2). In particular, in *Case 1* only the penalty for incomplete charging is considered and, so, it is assumed $\alpha_4 = 1$ and $\alpha_i = 0$ for $i = 1, 2, 3$. *Case 2* is characterized by $\alpha_1 = \alpha_3 = \alpha_4 = 1$, while the charging costs are not optimized. Finally, in *Case 3* all the objective function entries are taken into account and, therefore, $\alpha_i = 1$ for $i = 1, \ldots, 4$.

The problem is solved using IBM ILOG CPLEX 12.5 on a PC with a 1.40 GHz processor and 6 GB RAM: in the worst case, the computation time required to find the optimal solution is 62 seconds.

Table 4.2 summarizes the solution performance indexes in the three cases previously described. In addition, for each case, a diagram representing the optimal solution is reported (Fig. 4.2, Fig. 4.3 and Fig. 4.4). Time slots are on the x-axis, while on the y-axis the different charging stations, with their different outlets, are listed. Finally, bars of different colors identify the vehicles. As can be seen, in all

Table 4.1: Example parameters.

Parameter type	Name	Value	Condition
charging station	r_n	1	$n = 1$
		2	$n \geq 2$
	$cost_n^t$ [€]	$[0.10, 0.20]$	$\forall n, t$
	p_n^t [kW]	$[3, 24]$	$\forall n, t$
vehicle	cap_m [kWh]	$[10, 25]$	$\forall m$
	$d_{n,m}$ [km]	$[0, 5]$	$\forall n, m$

Table 4.2: Results

Performance Index (average)	Case 1	Case 2	Case 3
waiting time [time slots]	2.82	0.00	0.50
unit charging cost [€]	0.16	0.15	0.12
distance [km]	3.02	1.50	1.90
incomplete charging [kWh]	0	0	5

the cases all the vehicles are successfully distributed.

4.5 Concluding Remarks

In this Chapter, the EVs Smart Charging Management problem has been addressed, identifying the features of a DSS devoted to handle it. The contribution of this Chapter is twofold: first the general architecture of a *leader-follower* management approach for the *EVs charging management problem* is introduced; second, a MILP formulation for the lower-level problem, i.e., the *Vehicle-to-Charging Station Assignment Problem* is proposed: an example of application proves its effectiveness in

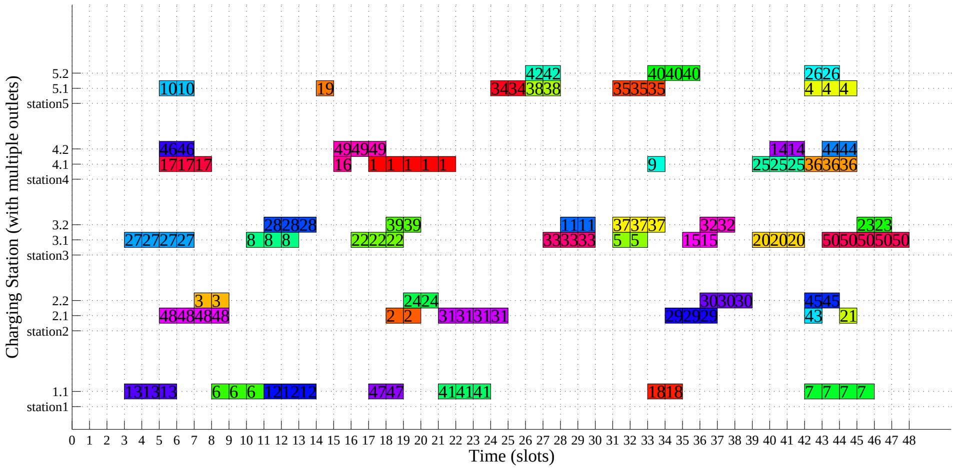

Figure 4.2: Optimal assignment and charging operations scheduling (**CASE A**).

providing the optimal solution considering different users utility functions. However, due to the complexity of the proposed formulation, a distributed approach appears to be more suitable to handle real case future scenarios.

The results of this Chapter are based on publications [Clemente et al., 2014] and [Fanti et al., 2015].

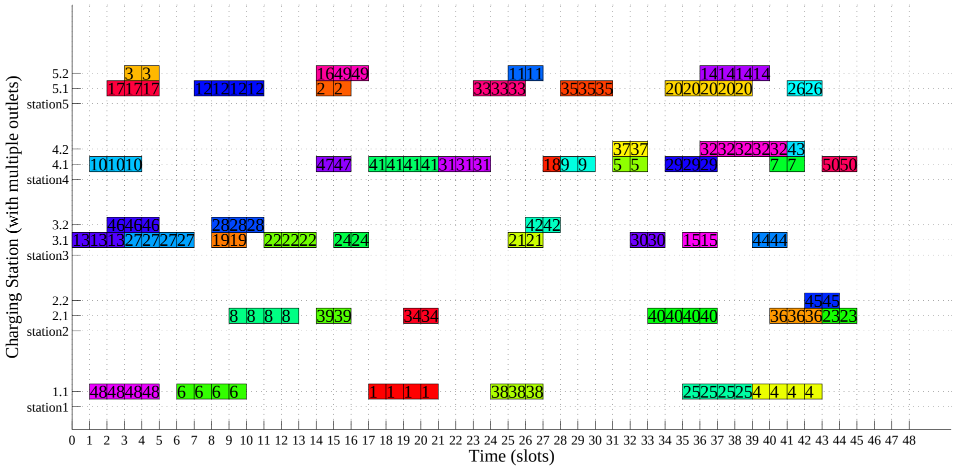

Figure 4.3: Optimal assignment and charging operations scheduling (**CASE B**).

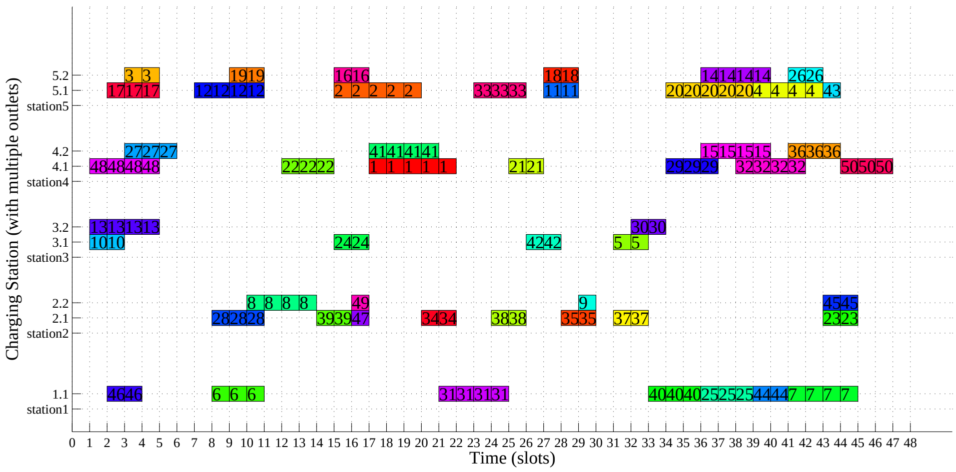

Figure 4.4: Optimal assignment and charging operations scheduling (**CASE C**).

Chapter 5

Container Drayage Problem

In this Chapter the multi-day container drayage problem is considered.

5.1 Motivation

In the related literature, the container transportation by trucks between a terminal and customers is usually referred to as *container drayage operation.* Drayage operations take on great importance in the context of container transportation, since neither ships nor trains can provide door-to-door services. However, such operations are responsible for a significant portion of the total transportation cost ([Cheung et al., 2008]) and, therefore, improving their efficiency is a necessity.

The container drayage problem is characterized by the presence of the following distinctive elements:

- a fleet of *trucks*;
- a set of *customers* (shippers or receivers);
- a (set of) *container terminal*(s);
- a (set of) *trucking company depot*(s);

- a set of *orders*, i.e., requests of moving a container from a given origin to a given destination.

  Three kinds of orders are possible. When *import* orders are considered, filled containers are located at the terminal and need to be moved to the depots or the receivers. On the contrary, *export* orders refer to the containers located at depots or customers places that need to be delivered to the terminals in order to be shipped. Finally, a particular type of container transportation order is the so-called *empty order*, i.e., the request to move an empty container from a given container terminal to another one.

Given such elements, the objective in the container drayage problem is to determine which truck performs which task (i.e., executes which order) while minimizing a given generalized cost function.

5.1.1 Container Drayage Problem Background

Traditionally, the container drayage problem leads back to one of the following models.

- **Vehicle Routing Problem** (VRP). The VRP is the "problem of minimizing the total travel distance of a number of vehicles, under various constraints, where every customer must be visited exactly once by a vehicle" [Hashimoto et al., 2006]. In order to apply such model to the container drayage problem, time constraints have to be added to the classical formulation and, therefore, the VRP with Time Windows (VRPTW) must be considered. Given that the VRP is NP-hard, several heuristics have been developed to solve it and its variations.

- **Multiple Traveling Salesman Problem** (mTSP). A mTSP is a generalization of the classic traveling salesman problem (TSP), where more than one

salesman is allowed to be used in the solution [Bektas, 2006]. In the classic version, all the salesmen start from and turn back to a home city (called *depot*). In the multiple depots variation, conversely, salesmen can either return to their original depot after completing their tour or return to any other depot, with the restriction that the initial number of salesmen at each depot must remain the same after all travels. Furthermore, if certain nodes need to be visited in specific time periods (which is the case of the container drayage problem), the Time Windows variation (mTSPTW) has to be taken into account.
The mTSP can be considered a relaxation of the VRP where the vehicle capacity restrictions are removed: therefore, all the approaches for the VRP can be applied to the mTSP by assigning sufficiently large capacities to the salesmen.

- **Pickup and Delivery Problem** (PDP). The PDP is a generalization of the VRP in which goods, commodities or people have to be transported between an origin and a destination [Dumas et al., 1991].
As above, different variations of the PDP exist. In the Full Truckload PDP (FTPDP) each vehicle carries a single load, while in the PDP with Time Windows (PDPTW) time constraints at customer locations are considered. FT-PDPTW formulations are, therefore, suitable to formulate container drayage problems since the containers are usually required to be picked up from somewhere and delivered to somewhere else at certain specific time intervals.
FTPDPTW can be reduced to a mTSPTW by collapsing each trasport request into a single node (i.e., by merging the pickup and delivery nodes of an order).

- **Assignment Problem** (AP). The problem is to find a one-to-one matching between n tasks and n agents while minimizing the total cost of the assignment [Pentico, 2007]. Many variations of the classic AP have been proposed in order to consider different further assumptions, e.g., the fact that not every agent is qualified to do every task or only a given subset of the taks has to be assigned. Particular interesting for the container drayage application is the Generalized

Assignment Problem (GAP): as in the classic AP, each task has to be assigned to an agent, but in this case multiple tasks can be assigned to the same agent.

5.1.1.1 Literature Review

Many authors have addressed the container drayage problem and different approaches for the optimization of drayage operations can be found in the related literature.

An overview of the papers dealing with such problem is reported in Tab. 5.1. In particular, the following features are highlighted for each reference (for each aspect, the possible values to be inserted in the table are reported in curly brackets):

1. **Objective** (**Obj.**: {single, multi}). We distinguish between *single*- and *multi*-objective approaches. Typical objectives for the container drayage problem are the minimization of the *total cost*, the minimization of the *total operating time*, the minimization of the *total distance travelled without any load* and the minimization of the *number of vehicles used*.

2. **Assumptions**. We characterize each reference with respect to the assumptions regarding the following aspects.

    - **Approach** (**S1**: {stat, dyn}). In most existing papers, drayage problems are addressed in a *static* environment, i.e., all the orders are assumed to be known in advance or, at least, probabilistic information about the future is required. However, in real-life scenarios such information is not available or not accurate: therefore, a *dynamic* approach in which the problem is re-solved any time more information becomes available would be preferable. Nevertheless, few articles focus on dynamic drayage problems.

      In Tab. 5.1, "stat" refers to static approach, "dyn" to dynamic approach.

- **Time Constraints** (**S2**: {HTW, STW, N, trips}). Usually, Time Windows (TW) at customers or containers terminals are considered. Moreover, when *Hard Time Windows* ("HTW") are taken into account, late services are not allowed and the time constraints must be satisfied exactly. Conversely, *Soft Time Windows* ("STW") can be violated, but the violation is usually penalised by adding a penalty cost to the objective function.
  On the other side, in some papers time windows in relation to trips are neglected ("N"), and trips can start and finish at any moment.
  Finally, some authors consider a maximum number of working hours for each truck ("trucks"), i.e., take into account service hours regulations.

- **Resources Suitability** (**S3**:{Y, N}). In most papers, homogeneous container type and size and homogeneous fleet are considered: therefore, the suitability of drivers and equipments for a specific load is not taken into account ("N"). If non-homogeneous trucks and orders are considered, "Y" is reported in Tab. 5.1.

- **Resources Availability** (**S4**: {Y, N}). In most papers, the number of available trucks is assumed to be adequate for meeting the demand: therefore, a feasible solution for the problem always exists.
  In Tab. 5.1, "Y" indicates that Resources Availability constraints are considered, otherwise "N" is reported.

- **Resource Typology** (**S5**: {sep, not sep}). A common assumption is that tractors and trailers (or trucks and containers) cannot be uncoupled ("not sep") during unpacking operations and, therefore, a truck have to wait at customer till such operations are completed.
  In order to increase the utilization of the fleet, some authors assume that trucks are allowed to leave the customer while the container is being unpacked: in this case, "sep" is reported in 3.1.

- **Order pairings** (**S6**: {Y, N}). In many papers, rules for combining trips are identified ("Y"). However, since usually homogeneous orders are considered, in merging two tasks only the origin and the destination of the trips, together with the associated time windows, are taken into account, while incompatibility of the loads is neglected.

- **Balancing Issues** (**S7**: {Y, N, Y (loads)}. Since in most papers only one depot is considered and each truck returns to the depot at the end of the time horizon, the problem of the fleet balancing with respect to the demand is usually not taken into account ("N").
  In some papers, load balancing issues are considered, i.e., distance already travelled by a truck during the day is taking into account in the order assignment process ("Y (loads)").

3. **Model**. As described in Section 2, the classical models applied for the container drayage problem are the "VRP", the "mTSP", the "PDP" and the "assignment problem". In Table 5.1, "TW" indicates that time windows are considered, "a" that an asymmetric cost matrix is taken into account, "1D" that only one depot is considered, while "MD" indicates multiple depots.

4. **Solution Technique**. Given its nature, different heuristics for the container drayage problem are proposed in literature. Basically, they are variations of classical heuristic approaches such as Tabu Search ("TS"), Reactive TS ("RTS"), Window Partition Based method ("WPB"), cluster method, ant colony optimization algorithms ("ACO"), Genetic Algorithms (GA), Insertion Heuristics ("IH"). Usually, a multi-stage approach is considered and a continuous refinement of the solution is pursued.
   (Note that in Tab. 5.1, row 10, "D-2PIH" stands for "Dynamic-2 Phase IH"; row 13, "2PDA" stands for "2-Phase Deterministic Annealing"; row 14, "2PHDA" stands for "2-Phase Hybrid Deterministic Annealing").

5. **CPU time [s]**. The time (in seconds) required to solve an instance of the problem is reported together with the instance size (in terms of number of orders to be served, when available). If no indications about the computation time are given, "n/s" (not specified) is reported.

   Note that in literature different benchmark instances for the different categories of problems considered are available (e.g., see [Solomon, 1987]). Usually, each of these instances provide information about: the number of depots; the number of terminals; the number of customers; the number of requests; time windows at origin and destination.

Table 5.1: Literature review.

Paper	Obj.	Assumptions							Model	Solution Technique	CPU time [s]
		S1	S2	S3	S4	S5	S6	S7			
[Xue et al., 2014]	single min(cost)	stat	STW	N	N	sep	Y	N	VRP - 1D	TS	9812.44 (200 orders)
[Zhang et al., 2014]	single min(time)	dyn	STW+HTW	N	N	not sep	N	N	am-TSPTW - MD	WPB method	$\leq$ 120 (66 orders)
[Caballini et al., 2013]	single min(cost)	stat	trucks	N	N	not sep	Y	N	VRP - 1D	3-stage heuristic	n/s
[Nossack and Pesch, 2013]	single min(cost)	stat	HTW	N	N	not sep	Y	N	FTPDPTW - MD	2-stage heuristic	$\leq$ 6 (75 orders)
[Vidović et al., 2012]	single min(cost)	stat	HTW	Y	N	not sep	Y	N	multiple-assignment - 1D	exact method	$\leq$ 125 (60 orders)
									VRPTW - 1D	exact method	1800 (19 orders)
[Caris and Janssens, 2009]	single min(cost)	stat	HTW	N	N	not sep	Y	N	FTPDPTW - 1D	3-stage heuristic	$\leq$ 1800 (200 orders)
[Zhang et al., 2009]	single min(time)	stat	HTW	N	N	not sep	N	N	m-TSPTW - MD	cluster method	67 (200 orders)
										RTS	5127,69 (200 orders)
[Hamdouni et al., 2013]	single min(distance)	stat	N	N	N	not sep	N	Y (loads)	assignment - 1D	ACO	711,58 (400 orders)

Paper	Obj.	Assumptions							Model	Solution Technique	CPU time [s]
		S1	S2	S3	S4	S5	S6	S7			
[Lai et al., 2013]	single min(cost)	stat	N	N	N	not sep	Y	N	VRP - 1D	2-stage heuristic	≤ 60 (≤ 250 orders)
[Escudero et al., 2013]	single min(cost)	dyn	STW	N	N	not sep	N	N	am-TSPTW - 1D	D-2PIH+GA	≤ 1 (100 orders)
[Ileri et al., 2006]	single min(cost)	stat	STW	Y	N	not sep	N	N	PDPTW - 1D	set partitioning + column generation	$180 \div 600$ ($5 \div 28$ orders)
[Yang et al., 2004]	single min(cost)	dyn	STW	N	Y	not sep	N	N	PDPTW - 1D	rolling horizon strategies (with simple heuristics)	20 (for each decision epoch)
[Sun et al., 2014]	single min(cost)	dyn	STW	Y	Y	not sep	N	N	PDPTW - MD	set partitioning + column generation	119 (407 orders)
[Pazour and Neubert, 2013]	multi max(loads) min(time)	dyn	STW	N	Y	not sep	N	Y	PDPTW - MD	route-generation heuristic	≤ 7.35 (real data)
[Braekers et al., 2013]	multi min(vehicles) min(distance)	stat	HTW	N	N	sep	N	N	am-TSPTW -1D	2PDA	≤ 7 (200 orders)
[Braekers et al., 2014]	multi min(♯ vehicles) min(distance)	stat	HTW	N	N	not sep	N	N	am-TSPTW -1D	2PHDA - TS	≤ 15 (200 orders)

Paper	Obj.	Assumptions							Model	Solution Technique	CPU time [s]
		S1	S2	S3	S4	S5	S6	S7			
[Zhang et al., 2013]	mono min(time)	stat	HTW+STW	N	N	not sep	N	N	m-TSPTW - 1D	improved RTS	≤ 400 (500 orders)
[Janssens and Braekers, 2011]	mono min(cost)	stat	HTW	N	N	not sep	N	N	FTPDPTW - 1D	set partitioning	n/s
[Zhang et al., 2010]	mono min(time)	stat	HTW	N	N	not sep	N	N	m-TSPTW - MD	WPB method	≤ 170 (75 orders)
[Coslovich et al., 2006]	mono min(cost)	stat	HTW	N	N	not sep	Y	Y	set covering	Lagrangian relaxation	1507.34 (100000 variables)

5.1.1.2 DSS application for the container drayage problem

The container drayage problem represents a suitable context for the application of a DSS.

In the related literature, different examples of applications at several relevant container transportation companies can be found. For example, in [Pazour and Neubert, 2013] the experience at the J. B. Hunt Transport Services, Inc., is described. J. B. Hunt is one of the largest transportation logistics companies in North America, with 15223 employees, including 10172 company drivers, and $ 3.8 billion consolidated revenue in 2010. A substantial portion of J. B. Hunt intermodal transportations includes drayage operations. In order to improve the efficiency, a systematic routing and scheduling methodology instead of a manual one was required. With this aim, in [Pazour and Neubert, 2013] a heuristic solution approach to determine driver load assignments and routing and to schedule these drivers such that the maximum number of loads are covered with minimum empty moves is described.

The authors report that, after two years from the introduction of the cross-town application, J. B.Hunt has been able to measure the benefits of the project and, in particular: a more automated and enhanced planning workflow; increased productivity of the truck planners; improved synchronization between demand and company capacity, with consequent reduction of the number of loads outsourced to third party drayage companies; improved timeliness and accuracy of planning information; capacity of generating schedules within seconds to immediately reflect operational changes; improvement of the operational efficiency with related positive financial impact (J. B. Hunt has documented annual cost savings of $ 581000).

[Sun et al., 2014] describes the development of a computer based solution for the daily drayage optimization problem (called “Short Haul Optimizer”) at Schneider National, Inc.

Schneider National, Inc. operates a large intermodal freight transportation network, which encompasses the continental United States, with significant coverage in Canada and Mexico. This network has 24 rail hubs, served by a fleet of more than 1300 trucks and 14000 containers, and moves more than 4000 dray shipments per day, including pickup, delivery and cross-town transfers between railroads, and repositioning moves. In order to address the recurring daily problem of assigning drivers to both maximize driver productivity and minimize the total operations cost, an approach based on set-partitioning formulation and column generation heuristic is considered. In particular, an operational DSS is implemented in order to provide real-time recommendations for the driver-assignment process considering the constantly changing data. Schneider National, Inc. reports that, thanks to the implementation of the "Short Haul Optimizer" many benefits can be highlighted and, in particular: a 5% decrease in the reliance on foreign carriers, resulting in a roughly 3% reduction in overall drayage cost; 10% improvement of the fleet utilization; increased number of shipments converted from foreign carrier outsourcing to coverage by company. The corresponding annualized savings are in the range of $8 to $10 million.

5.1.2 Discussion

It is apparent that container drayage problem has attracted, and continues to attract, a lot of attention in the scientific literature. However, in most of the cited papers a lot of simplifications compared to real case studies are introduced. In particular:

- in most of the papers the trucks availability is considered coherent with the number of orders that have to be performed, i.e., the considered optimization problems admit always a feasible solution, without the necessity of delaying any transportation request;
- usually a single depot is considered and all the drivers return to it at the end of *each* working day. Therefore, the considered planning horizon is the single

day and the effects of the orders schedule are limited to the current working day;

- in most of the papers an homogeneous fleet of trucks is assumed and, therefore, the problem of the suitability of the loads is not taken into account;
- service hours regulations usually are not taken into account while planning the orders schedule;
- usually the allocation of empty containers is a problem optimized directly by the company on the basis of its needs and not a particular type of order received from the customers;
- the reported computation times are extremely variable, depending most on the considered assumptions.

From the evidence of two real case studies, it is apparent that the application of a DSS is a promising solution to improve the orders scheduling process, but it is also clear that an *ad-hoc* implementation based on the specific case study has to be developed in order to take care of all the particular requirements.

5.2 Objectives

This Chapter deals with the development of a DSS to support a company truck manager, i.e., the company staff member in charge of assigning container transportation orders to the available fleet of trucks, in his operations. In particular, the DSS should be able to operate on-line, addressing real-life instances without introducing unrealistic hypotheses.

To this aim, the Optimization and the Decision Module of the general DSS architecture described in Chapter 2 are specified. In particular, a MILP model for the multi-day container drayage problem is developed. Even if such formulation is able to capture the essential features of the considered problem, it is well-suited only for small-size instances due to computational time issues. For this reason, a fast heuristic based on the rolling horizon approach is introduced.

The remainder of this Chapter is structured as follows. Section 5.3 describes the considered problem and specifies the main features of the DSS. Section 5.4 specifies the DSS Optimization and Decision Module, describing the MILP model and the developed heuristic, as well as the KPIs considered to evaluate the proposed solutions. Moreover, a test case useful to demonstrate the effectiveness of the proposed heuristic is presented. Finally, Section 5.4 summarizes the remarks and the contributions of the present Chapter.

5.3 DSS for the Container Drayage Problem

In this Section, the multi-day container drayage problem considered in this dissertation is described, and the features of a DSS devoted to its management are highlighted. In particular, the DSS has to operate on-line, guaranteeing responsive suggestions to the decision maker.

5.3.1 Problem Statement

Given a heterogeneous fleet of trucks and a set of container transportation orders, the objective is to optimally assign the orders to trucks in order to maximize the total number of assigned orders and minimize a generalized cost function, which takes into account the total distance travelled without any container and the number of delayed orders. In particular, the following assumptions are introduced:

A1 Resources. The resources of the drayage problem are the *trucks*.

A1.1 Each truck is the association among a driver, a tractor and a trailer.

A1.2 Each truck can carry only certain types of container.

A1.3 Each truck has its own depot, where it has to return without load by a defined ending time.

A1.4 Truck operations can be performed from defined starting time and position.

A1.5 Each truck should respect a minimum rest period each night. Moreover, possible statutory vacations periods during the planning horizon are considered: in case of vacation, the truck has to return without load to its own depot.

A1.6 The truck fleet size and composition are given.

Summing up, each truck is characterized by the following parameters: types of container that it can carry; starting time; starting position; ending time; depot.

A2 Tasks. The tasks of the drayage problem are the container transportation orders.

A2.1 Each container transportation order is characterized by three locations to be visited: the starting point A, where the truck carries the container; the intermediate point B, where the container is loaded/unloaded; the destination point C, where the container is delivered.

A2.2 Each order is characterised by hard/soft time windows to be fulfilled at the three locations A, B, C.

A2.3 An order can be performed by a truck only if such truck can carry the required typology of container.

Summing up, each order is characterized by the following parameters: type of container; requested locations A, B, C to be visited; requested times at

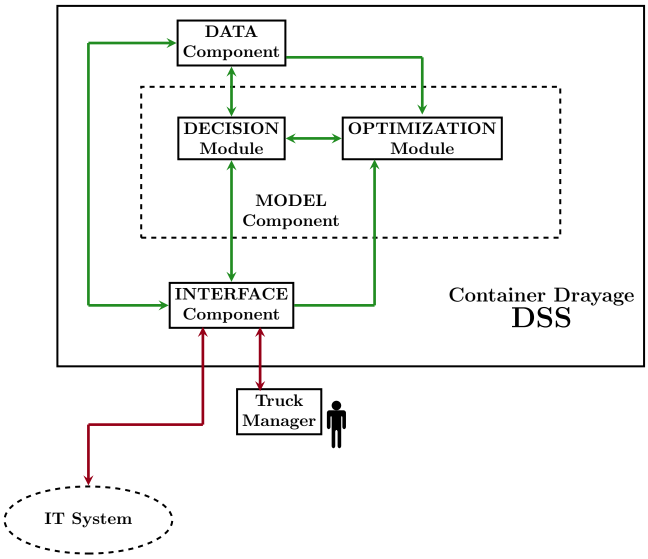

Figure 5.1: DSS for the container drayage problem.

the locations A, B, C, which define the earliest and the latest times for the hard/soft time windows.

A3 Approach. The drayage problem is addressed in a *deterministic* framework, i.e., all the tasks and resources parameters are assumed to be known at the beginning of the planning horizon.

5.3.2 DSS Formalization

Fig. 5.1 outlines the DSS structure considered in this Chapter.

In particular, considered that in this application the DSS is devoted for operational level decisions, the support is given by exploiting the Decision Module and the Optimization Module. The Simulation Module, not relevant in this context, can be used for other types of decisions: for example, at the strategic level, it could be useful to address the fleet sizing problem.

The optimization process can be triggered directly by the decision maker, i.e., the truck manager. The input data required by the DSS to operate, i.e., trucks and orders parameters, are supplied by the trucking company IT system. Finally, the outputs of the decision process suggested to the decision maker are the assignment of the orders to the trucks and a set of KPIs particular relevant in this context: in particular, they consider the total number of container orders assigned, the total distance travelled without any load and the possible delays (see "Decision Module Specification").

5.3.2.1 Assignment of a container transportation order to a truck

As already stated, the first phase in the development of the DSS is the analysis of the problem to be addressed. In particular, the DSS considered in this Chapter has the aim of supporting the activities performed by the company truck managers while assigning container transportation orders to the available fleet of trucks. Therefore, understanding such process is fundamental: the UML activity diagram of Fig. 5.2 shows the main phases that typically characterise it.

As can be seen, six actors are involved:

- the *customer*, that makes a request for a container transportation;
- the *customer care*, i.e., the company staff member responsible for the communication with the customers;
- the company *IT system*, where all the information about container transportation orders and trucks position and availability are stored;
- the *truck manager*, i.e., the company staff member in charge of assigning container transportation orders to the available fleet of trucks. The truck manager is, therefore, the decision maker whose operations have to be supported by the DSS;

- the *dispatcher*, i.e., the company staff member that constantly interacts with the drivers, checking their availability to effectively perform a given container transportation order;
- the *driver*, that represents the driver/truck combination.

The assignment of a container transportation order to a truck can be outlined as follows. The truck manager checks if there are available trucks to be assigned to new orders. If there are not available trucks, but there are still orders to be assigned, then the truck manager communicates it to the customer care, who asks to the customer if it is possible to postpone the pending orders. On the contrary, if there are available trucks to be assigned, the truck manager checks if there are compatible orders (i.e., orders for container typologies that can be transported by the considered trucks): if so, the truck manager assigns the truck to an order and store the information in the IT system. If the truck is actually available to perform the assigned task, then the order is removed from the list of the pending orders, otherwise it has to be assigned again.

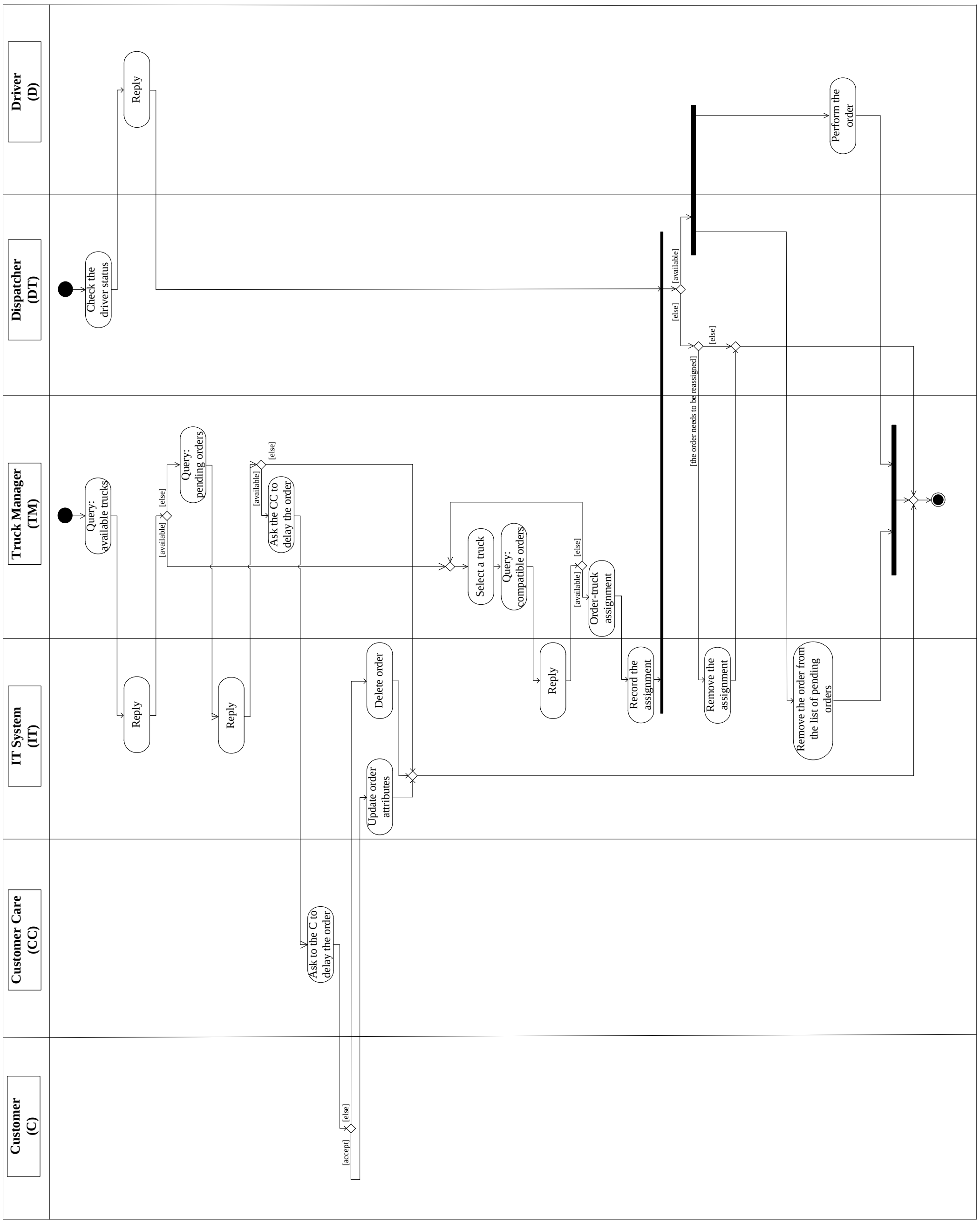

Figure 5.2: The process of assigning a container transportation order to a given truck.

5.4 DSS Module Specification

In this Section, the Optimization and the Decision Modules of the DSS are specified.

5.4.1 Optimization Module Specification

5.4.1.1 MILP Model: Mathematical Formulation

In order to describe the MILP formulation, the following notation is introduced.

Numerical Sets

- $\mathbb{N}$: set of natural numbers.
- $\mathbb{N}^+$: set of all positive natural numbers.
- $\mathbb{R}_0^+$: set of all positive real numbers including 0.

A Time Indexed Formulation (TIF) is considered: the planning horizon includes Q days, with $Q \in \mathbb{N}^+$, and each day is discretized into K time periods, $K \in \mathbb{N}^+$, each lasting 1 time unit (t.u.). Therefore, the planning horizon starts at time 1 and ends at time $T = K \cdot Q$. Moreover, each time period starts at time t and ends at time $t+1$, i.e., we consider the time periods $1, 2, \ldots, T$: hereafter, the time period is denoted with its starting time t.

Sets

- $\mathcal{T} = \{1, 2, \ldots, T\}$: set of *time periods.*
- $\mathcal{G}_i = \{\text{sleep}_{min} + K \cdot (i-1), \text{sleep}_{min} + K \cdot (i-1) + 1, \ldots, \text{sleep}_{max} + K \cdot (i-1)\}$, $i \in \{1, 2, \ldots, Q-1\}$: set of *time periods* during which it is allowed to begin the night rest, where sleep_{min} (respectively, sleep_{max}) is the earliest time (latest time), expressed in t.u., at which the night rest can start every day.

- $\mathcal{R} = \{1, 2, \dots, R\}$: set of available resources, i.e., fleet of trucks [assumptions **A1**], where $R \in \mathbb{N}^+$ is the fleet size [assumptions **A1.6**].

- $\mathcal{S} = \{1, 2, \dots, S\}$: set of tasks to be performed during the planning horizon, i.e., container transportation orders [assumption **A2**].

- $\mathcal{S}_{sleep} = \{1, 2, \dots, S_{sleep}\}$: set of dummy orders modelling the night rest, with $S_{sleep} = R \cdot (Q-1) \cdot (\text{sleep}_{max} - \text{sleep}_{min} + 1)$ [assumption **A1.5**]. Indeed, for each truck of the fleet, for each day of the planning horizon and for each time period included between sleep_{min} and sleep_{max}, a dummy order is considered.

- $\mathcal{S}^* = \mathcal{S} \cup \mathcal{S}_{sleep}$: total set of tasks to be assigned to the fleet of trucks, including both real orders and dummy orders.

- $\mathcal{E} = \{1, 2, \dots, E\}$: number associated to each typology of containers [assumptions **A1.2** and **A2.3**].

Parameters

1. *Resources Parameters*

   Each truck $c \in \mathcal{R}$ is characterized by [assumption **A1**]:

   - $r_{c,e} \in \{0, 1\}$: flag equal to 1 if truck c can carry a container of type $e \in \mathcal{E}$ [assumption **A1.2**].
   - $t_c^{avail} \in \mathcal{T}$: starting time of truck c [assumption **A1.4**].
   - $t_c^{finish} \in \mathcal{T}$: ending time of truck c [assumption **A1.4**].
   - $d'_{c,v} \in \mathbb{N}$: time distance, expressed in t.u., between the starting position of truck c and location A of order $v \in \mathcal{S}$ [assumption **A1.4**].
   - $d''_{c,v} \in \mathbb{N}$: time distance, expressed in t.u., between location C of order $v \in \mathcal{S}$ and the depot of truck c [assumptions **A1.3** and **A1.4**].

2. *Tasks Parameters*

   Each order $v \in \mathcal{S}^*$ (both real orders and dummy orders) is characterized by [assumption **A2**]:

   - $s_{v,e} \in \{0,1\}$: flag equal to 1 if order v is for a container of type $e \in \mathcal{E}$. For $v \in \mathcal{S}_{sleep}$, $s_{v,e}$ is always equal to 0 [assumption **A2.3**].
   - $t_v^{A,open} \in \mathcal{T}$: earliest time at location A for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{A,open}$ is not defined.
   - $t_v^{A,close} \in \mathcal{T}$: latest time at location A for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{A,close}$ is not defined.
   - $t_v^{C,open} \in \mathcal{T}$: earliest time at location C for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{C,open}$ is not defined.
   - $t_v^{C,close} \in \mathcal{T}$: latest time at location C for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{C,close}$ is not defined.
   - $t_v^{B,earliest} \in \mathcal{T}$: earliest time at location B for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{B,earliest}$ is not defined.
   - $t_v^{B,latest} \in \mathcal{T}$: latest time at location B for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, $t_v^{B,latest}$ is not defined.
   - $\delta_v^B \in \mathbb{N}$: maximum delay admitted at location B for order v [assumption **A2.2**]. For $v \in \mathcal{S}_{sleep}$, δ_v^B is not defined.
   - $\tau_v \in \mathbb{N}$: time distance, expressed in t.u., between location A and location B of order v [assumption **A2.1**]. For $v \in \mathcal{S}_{sleep}$, τ_v is always equal to 0.
   - $\theta_v \in \mathbb{N}$: time distance, expressed in t.u., between location A and location C of order v, i.e., overall length of order v [assumption **A2.1**]. For $v \in \mathcal{S}_{sleep}$, θ_v is always equal to the length of the night rest.
   - $d_{v,w} \in \mathbb{N}$: time distance, expressed in t.u., between location C of order v and location A of order w, $w \in \mathcal{S}, w \neq v$.

3. *Model Parameters*

- $M \in \mathbb{N}^{+}$: a sufficiently large number.

Decision variables.

For each truck $c \in \mathcal{R}$, order $v \in \mathcal{S}^*$ and $t \in \mathcal{T}$, the following decision variable is defined:

$$x_{c,v}(t) = \begin{cases} 1 & \text{if } v \text{ is assigned to } c \text{ and } c \text{ moves from its current} \\ & \text{position to perform } v \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

Moreover, for each truck $c \in \mathcal{R}$, orders $v, w \in \mathcal{S}$, $v \neq w$, and each $t \in \mathcal{T}$, the following decision variable is defined:

$$z_{c,v,w}(t) = \begin{cases} 1 & \text{if } w \text{ is assigned to } c \text{ and } c \text{ moves from its current position} \\ & \text{to perform } w \text{ at time } t\text{, immediately after order } v \\ 0 & \text{otherwise} \end{cases}$$

For each truck $c \in \mathcal{R}$, for each $v \in \mathcal{S}$, each $w \in \mathcal{S}_{sleep}$ and each $t \in \mathcal{T}$, the following decision variable is defined:

$$p_{c,v,w}(t) = \begin{cases} 1 & \text{if } c \text{ performs a night rest } w \text{ at time } t\text{, after having started} \\ & v \\ 0 & \text{otherwise} \end{cases}$$

For each truck $c \in \mathcal{R}$ and $t \in \mathcal{T}$, the following decision variable is defined:

$$y_c(t) = \begin{cases} 1 & \text{if } c \text{ begins its night rest at time } t \\ 0 & \text{otherwise} \end{cases}$$

Finally, for each truck $c \in \mathcal{R}$ and each order $v \in \mathcal{S}$, the following decision variables are defined:

$$l_{c,v} = \begin{cases} 1 & \text{if } c \text{ performs } v \text{ arriving at location B after } t_v^{B,latest} \text{ and by} \\ & t_v^{B,latest} + \delta_v^B \\ 0 & \text{otherwise} \end{cases}$$

$$z'_{c,v} = \begin{cases} 1 & \text{if } c \text{ performs order } v \text{ from its starting position} \\ 0 & \text{otherwise} \end{cases}$$

$$z''_{c,v} = \begin{cases} 1 & \text{if } c \text{ performs order } v \text{ and then goes to its ending position} \\ 0 & \text{otherwise} \end{cases}$$

Note that

$$z'_{c,v} = \sum_{t \in \mathcal{T}} x_{c,v}(t) - \sum_{\substack{u \in \mathcal{S} \\ u \neq v \\ t \in \mathcal{T}}} z_{c,u,v}(t) \qquad \forall c \in \mathcal{R}, v \in \mathcal{S}$$

and

$$z''_{c,v} = \sum_{t \in \mathcal{T}} x_{c,v}(t) - \sum_{\substack{u \in \mathcal{S} \\ u \neq v \\ t \in \mathcal{T}}} z_{c,v,u}(t) \qquad \forall c \in \mathcal{R}, v \in \mathcal{S}$$

The problem can be formulated as follows:

$$\max\,[f_1, -f_2, -f_3,] \tag{5.1}$$

where

$$f_1 = \sum_{\substack{c\in\mathcal{R}\\ v\in\mathcal{S}\\ t\in\mathcal{T}}} x_{c,v}(t) \tag{5.2}$$

$$f_2 = \left(\sum_{\substack{c\in\mathcal{R}\\ u,v\in\mathcal{S}\\ t\in\mathcal{T}}} z_{c,u,v}(t)\cdot d_{u,v} \right) + \left(\sum_{\substack{c\in\mathcal{R}\\ v\in\mathcal{S}}} z'_{c,v}\cdot d'_{c,v} \right) + \left(\sum_{\substack{c\in\mathcal{R}\\ v\in\mathcal{S}}} z''_{c,v}\cdot d''_{c,v} \right) \tag{5.3}$$

$$f_3 = \sum_{\substack{c\in\mathcal{R}\\ v\in\mathcal{S}}} l_{c,v} \tag{5.4}$$

s. t.:

$$\sum_{\substack{c\in\mathcal{R}\\ t\in\mathcal{T}}} x_{c,v}(t) \le 1 \qquad \forall v\in\mathcal{S} \tag{5.5}$$

$$\begin{aligned}\sum_{t\in\mathcal{T}} t\cdot x_{c,w}(t) - \Bigg(d'_{c,v}\cdot z'_{c,v} + \sum_{t\in\mathcal{T}} ((t+\theta_v)\cdot x_{c,v}(t)) + \\ + \sum_{\substack{u\in\mathcal{S}\\ u\neq v\\ t\in\mathcal{T}}} d_{u,v}\cdot z_{c,u,v}(t) + \sum_{\substack{u\in\mathcal{S}_{sleep}\\ t\in\mathcal{T}}} \theta_u\cdot p_{c,v,u}(t) \Bigg) \ge \\ \ge \left(\sum_{t\in\mathcal{T}} z_{c,v,w}(t) - 1 \right)\cdot M \qquad \forall c\in\mathcal{R}, v\in\mathcal{S}, w\in\mathcal{S}\end{aligned} \tag{5.6}$$

$$\sum_{t\in\mathcal{T}} t\cdot x_{c,w}(t) - \sum_{t\in\mathcal{T}} t\cdot x_{c,v}(t) \ge \left(\sum_{t\in\mathcal{T}} p_{c,v,w}(t) - 1 \right)\cdot M \quad \forall c\in\mathcal{R}, v\in\mathcal{S}, w\in\mathcal{S}_{sleep} \tag{5.7}$$

$$\sum_{t\in\mathcal{T}} t\cdot x_{c,w}(t) \geq t_c^{avail}\cdot z'_{c,w} + \sum_{\substack{v\in\mathcal{S}\\ t\in\mathcal{T}}} t\cdot z_{c,v,w}(t) \qquad \forall c\in\mathcal{R}, w\in\mathcal{S} \tag{5.8}$$

$$\sum_{t\in\mathcal{T}} t\cdot x_{c,w}(t) \leq t_c^{finish}\cdot z'_{c,w} + \sum_{\substack{v\in\mathcal{S}\\ t\in\mathcal{T}}} t\cdot z_{c,v,w}(t) \qquad \forall c\in\mathcal{R}, w\in\mathcal{S} \tag{5.9}$$

$$\sum_{\substack{v\in\mathcal{S}\\ t\in\mathcal{T}}} t\cdot p_{c,v,w}(t) = \sum_{t\in\mathcal{T}} t\cdot x_{c,w}(t) \qquad \forall c\in\mathcal{R}, w\in\mathcal{S}_{sleep} \tag{5.10}$$

$$\sum_{\substack{c\in\mathcal{R}\\ w\in\mathcal{S}\\ w\neq v\\ t\in\mathcal{T}}} z_{c,v,w}(t) \leq 1 \qquad \forall v\in\mathcal{S} \tag{5.11}$$

$$\sum_{\substack{c\in\mathcal{R}\\ v\in\mathcal{S}\\ v\neq w\\ t\in\mathcal{T}}} z_{c,v,w}(t) \leq 1 \qquad \forall w\in\mathcal{S} \tag{5.12}$$

$$\sum_{\substack{c\in\mathcal{R}\\ w\in\mathcal{S}_{sleep}\\ t\in\mathcal{T}}} p_{c,v,w}(t) \leq 1 \qquad \forall v\in\mathcal{S} \tag{5.13}$$

$$\sum_{\substack{t\in\mathcal{T}\\ t\leq t_c^{avail}-1}} x_{c,v}(t) = 0 \qquad \forall c\in\mathcal{R}, v\in\mathcal{S}^* \tag{5.14}$$

$$\begin{aligned}\sum_{\substack{v\in\mathcal{S}\\ t\in\mathcal{T}}} &((t+\theta_w+d_{v,w})\cdot z_{c,v,w}(t)) + (d''_{c,w}\cdot z''_{c,w}) + \\ &+ \sum_{\substack{u\in\mathcal{S}_{sleep}\\ t\in\mathcal{T}\\ t\leq t_w^{C,close}}} \theta_u\cdot p_{c,w,u}(t) \leq t_c^{finish} \qquad \forall c\in\mathcal{R}, w\in\mathcal{S}\end{aligned} \tag{5.15}$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} (t \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \le t_w^{A,close}}} (\theta_u \cdot p_{c,w,u}(t)) \geq$$
$$\geq t_w^{A,open} \cdot \left(\sum_{t \in \mathcal{T}} x_{c,w}(t) \right) \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.16)$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} (t \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \le t_w^{A,close}}} (\theta_u \cdot p_{c,w,u}(t)) \leq$$
$$\leq t_w^{A,close} \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.17)$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} ((t + \theta_w) \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \le t_w^{C,close}}} (\theta_u \cdot p_{c,w,u}(t)) \geq$$
$$\geq t_w^{C,open} \cdot \left(\sum_{t \in \mathcal{T}} x_{c,w}(t) \right) \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.18)$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} ((t + \theta_w) \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \le t_w^{C,close}}} (\theta_u \cdot p_{c,w,u}(t)) \leq$$
$$\leq t_w^{C,close} \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.19)$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} ((t + \tau_w) \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \le t_w^{B,latest}}} (\theta_u \cdot p_{c,w,u}(t)) \geq$$
$$\geq t_w^{B,earliest} \cdot \left(\sum_{t \in \mathcal{T}} x_{c,w}(t) \right) \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.20)$$

$$d'_{c,w} \cdot z'_{c,w} + \sum_{t \in \mathcal{T}} ((t + \tau_w) \cdot x_{c,w}(t)) + \sum_{\substack{v \in \mathcal{S} \\ t \in \mathcal{T}}} (d_{v,w} \cdot z_{c,v,w}(t)) + \sum_{\substack{u \in \mathcal{S}_{sleep} \\ t \in \mathcal{T} \\ t \leq t_w^{B,latest}}} (\theta_u \cdot p_{c,w,u}(t)) \leq$$
$$\leq t_w^{B,latest} + \delta_w^B \cdot l_{c,w} \qquad \forall c \in \mathcal{R}, w \in \mathcal{S} \quad (5.21)$$

$$s_{v,e} \cdot \sum_{t \in \mathcal{T}} x_{c,v}(t) \leq r_{c,e} \qquad \forall e \in \mathcal{E}, c \in \mathcal{R}, v \in \mathcal{S} \quad (5.22)$$

$$\sum_{\substack{t \in \mathcal{G}_i \\ t_c^{avail} \leq t \leq t_c^{finish}}} y_c(t) = 1 \qquad \forall c \in \mathcal{R}, i \in \{1, 2, \ldots, Q\} \quad (5.23)$$

$$\sum_{\substack{t \in \mathcal{T} \setminus \{\cup_i \mathcal{G}_i\} \\ i \in \{1,2,\ldots,Q\}}} y_c(t) = 0 \qquad \forall c \in \mathcal{R} \quad (5.24)$$

$$\sum_{\substack{t \in \mathcal{G}_{i+1} \\ t_c^{avail} \leq t \leq t_c^{finish}}} t \cdot y_c(t) - \sum_{\substack{t \in \mathcal{G}_i \\ t_c^{avail} \leq t \leq t_c^{finish}}} t \cdot y_c(t) \leq K \quad \forall c \in \mathcal{R}, i \in \{i = 1, \ldots, Q-2\} \quad (5.25)$$

$$\sum_{\substack{v \in \mathcal{S}_{sleep} \\ t \in \mathcal{G}_i}} t \cdot x_{c,v}(t) = \sum_{t \in \mathcal{G}_i} t \cdot y_c(t) \qquad \forall c \in \mathcal{R}, i \in \{1, 2, \ldots, Q\} \quad (5.26)$$

$$x_{c,v}(t) \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S}^*, t \in \mathcal{T} \quad (5.27)$$

$$z_{c,v,w}(t) \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S}, w \in \mathcal{S}, t \in \mathcal{T} \quad (5.28)$$

$$p_{c,v,w}(t) \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S}, w \in \mathcal{S}_{sleep}, t \in \mathcal{T} \quad (5.29)$$

$$y_c(t) \in \{0, 1\} \qquad \forall c \in \mathcal{R}, t \in \mathcal{T} \quad (5.30)$$

$$l_{c,v} \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S} \quad (5.31)$$

$$z'_{c,v} \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S} \quad (5.32)$$

$$z''_{c,v} \in \{0, 1\} \qquad \forall c \in \mathcal{R}, v \in \mathcal{S} \quad (5.33)$$

A multi-objective optimization framework is considered. In particular, (5.2) is the total number of *covered orders*, i.e., the total number of orders that are performed by all trucks during the planning horizon; (5.3) is the total distance travelled by all trucks without any load, and is given by the sum of three terms: the distance travelled between location C of an order and location A of the following order, the distance travelled between the depot and the location A of the first order and, finally, the distance between the location C of the last order to be performed in the planning horizon and the depot. (5.4) is the total number of delayed orders.

Constraints (5.5) are the *assignment uniqueness constraints* and ensure that each order is performed at most by one truck.

Constraints (5.6)÷(5.13) are the *orders sequencing constraints*. In particular, constraints (5.6) ensure that, if order $w \in \mathcal{S}$ is performed immediately after order $v \in \mathcal{S}$ (note that both w and v are real order), then order w can not start before order v has been completed. On the other hand, as expressed in (5.7), if w is a dummy order, it can interrupt the previous real order v, and therefore it is sufficient to guarantee that the starting time of w is subsequent to the starting time of v. Constraints (5.10) ensure that a real order that follows a night rest can start only if the real order preceding the night rest has been completed. Moreover, constraints (5.8) and (5.9) ensure that order $w \in \mathcal{S}$ can be executed after order $v \in \mathcal{S}$ by truck $c \in \mathcal{R}$ if and only if w is effectively assigned to c; at the same time, constraints (5.10) ensure that a truck c can have a night rest $w \in \mathcal{S}_{sleep}$ after having started order $v \in \mathcal{S}$ if and only if w is effectively assigned to c. Constraints (5.11) and (5.12) guarantee that each order can have at most one successor and one predecessor, respectively. Finally, constraints (5.13) ensure that each order can have at most one night rest as successor.

Constraints (5.14) and (5.15) are the *resource availability constraints*.

Constraints (5.16) ÷ (5.21) are the *time constraints* at locations A, B and C, respectively, defined only for the real orders.

Constraints (5.22) are the *container type constraints* and guarantee that a truck performs a given order only if it can carry the required type of container.

Finally, constraints (5.23)÷(5.26) are the *service hours constraints*, i.e., guarantee that each driver has a night rest of θ_v t.u., $v \in \mathcal{S}_{sleep}$, every day. More in detail, constraints (5.23) and (5.24) ensure that only one night rest per day is performed, and only in the allowed interval [$\text{sleep}_{min}, \text{sleep}_{max}$] defined above, while constraints (5.25) impose that the interval between two night rests is at maximum equal to 1 day. At the same time, constraints (5.26) define the night rest as a dummy order that must be assigned to a truck as the real orders.

Constraints (5.27) ÷ (5.33) are the binary variables definitions.

5.4.1.2 Fast Rolling Horizon Heuristic

The MILP formulation presented in the previous Section is well-suited for small dimensions instances, but it has computational complexity issues when applied to real-life scenarios. For example, consider an instance characterized by 1000 resources, 1000 tasks and a planning horizon of 10 days, using minutes as t.u.: the occurrences of the decision variables $z_{c,v,w}(t)$ are of the order of 10^{13}.

For this reason, a heuristic algorithm based on the rolling horizon approach is introduced [Wolsey, 1998], [Dotoli et al., 2006].

Rolling Horizon Heuristic

Usually, rolling horizon heuristics are used to address problems where input data are gradually revealed during the planning horizon and decisions have to be taken dynamically as new information arrives. For this reason, they result to be particularly suitable for production scheduling and supply chain problems, and most of the literature focuses on them [Sahin et al., 2013].
At the same time, a rolling horizon approach is suitable when the problem data

are perfectly known, but the computational complexity and the need of addressing large-scale instances make impossible to apply exact methods for the solution of the overall problem. In this case, the problem is decomposed into a number of smaller sub-problems, which are consecutively solved [Beraldi et al., 2008].

The heuristic proposed in this work decomposes the problem into Q sub-problems, each having a single-day planning horizon, where Q, coherently with the notation already introduced for the MILP model, is the total number of days considered in the overall multi-day planning horizon of length T. The choice of such an approach is due to the fact that usually the number of orders to be assigned is greater than the number of available trucks and the typical overall length of the orders does not exceed one day: therefore, each truck can perform more than one order per day.

Each sub-problem q, $q = 1, 2, \ldots, Q$, considers the constraints derived from the orders schedule of day $(q-1)$ and takes into account some information about the container transportation demand of day $(q+1)$.
As the planning horizon is limited, dummy days $q = 0$ and $q = (Q+1)$ are introduced. Therefore, in each sub-problem, a time window of length 3 days is considered, as shown in Fig. 5.3: the planning day q is highlighted with the solid green slot, and the whole rolling horizon time window is shown by the green lined slots before and after the considered day ($(q-1)$ and $(q+1)$). Note that K is the number of time periods per day, as already stated for the MILP model.

Tasks to be assigned
Each sub-problem q, $q = 1, 2, \ldots, Q$, considers as tasks to be assigned the real container transportations orders of day q, i.e., it holds: $\mathcal{S}^q = \{v \in \mathcal{S} | \, ((q-1) \cdot K) \leq t_v^{B,latest} < (q \cdot K)\}$, $\bigcup\limits_{q=1,2,\ldots,Q} \mathcal{S}^q = \mathcal{S}$, $\bigcap\limits_{q=1,2,\ldots,Q} \mathcal{S}^q = \emptyset$, where $\mathcal{S}$ is the set of real container transportation orders, as introduced in the MILP formulation. Correspondingly, $\mathcal{R}^q$ is the set of trucks that are available to perform the orders of sub-problem q.

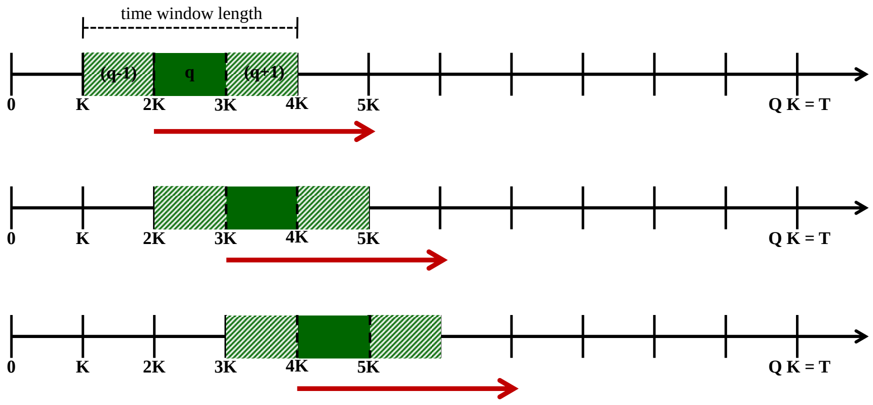

Figure 5.3: 3-day rolling horizon approach.

Indeed, considering a single-day planning horizon allows to simplify the management of the service hours regulation: thus, with respect to the MILP formulation, the dummy orders representing the night rests are replaced with an additional term to be considered in the time constraints, as will be described in the Section "Solution of the sub-problem q".

Information about the previous and the following days

In each sub-problem q, *the information about day* $(q-1)$ is exploited to update the starting position and time availability for each truck of the fleet.
More formally, for each $c \in \mathcal{R}$, parameters t_c^{avail} (which represents the starting time of truck c at the beginning of the planning horizon) and $d'_{c,v}$ (which represents time distance between the starting position of truck c and location A of order $v \in \mathcal{S}$) are replaced, in each sub-problem q, by $t_c^{avail,q}$ and $d'^q_{c,v}$, respectively.
In particular, $t_c^{avail,q}$ is the time availability of truck c at the beginning of day q, resulting from the schedule of day $(q-1)$, while $d'^q_{c,v}$ is the time distance between the starting position of truck c at day q and location A of order $v \in \mathcal{S}^q$ to be assigned.

Moreover, in each sub-problem q, *the information about day* $(q+1)$ is exploited

in order to minimize the overall distance travelled without any load.
To this aim, the locations to be visited while performing the container transportation orders are divided into *operational regions*, and the number and the type of container orders for each operational region at day $(q+1)$ are evaluated. Formally, denoting with N the total number of operational regions, the matrix $\mathbf{M}^{(q+1)} \in \mathbb{N}^{N \times E}$, with $n = 1, 2, \ldots, N$, $e \in \mathcal{E}$, is introduced (remind that $E = |\mathcal{E}|$ is the total number of possible container typologies). The generic element $m_{n,e}^{(q+1)}$ of such matrix is the number of orders with container of type e at operational region n for day $(q+1)$.

Solution of the sub-problem q

Each sub-problem q is in turn decomposed in h interdependent assignment problems, with $h \in \mathbb{N}^+$.

In particular, define $I_h \subset \mathcal{T}$ the set of time periods included in the planning horizon of the assignment problem h, $\mathcal{S}^{q,h}$ the set of container orders to be assigned by the h-th assignment problem, and $\mathcal{R}^{q,h}$ the set of available trucks considered in the h-th assignment problem.
Define $\mathbf{O} \in \mathbb{R}^{\mathcal{R}^{q,h} \times \mathcal{S}^{q,h}}$ the *generalized cost matrix* and $\mathbf{A} \in \{0,1\}^{\mathcal{R}^{q,h} \times \mathcal{S}^{q,h}}$ the *adjacency matrix*, i.e., the matrix whose generic element $a_{c,v}$ is equal to 1 if the truck c can perform the order v and 0 otherwise. Finally, consider the binary decision variables $\overline{x}_{c,v}$, defined for each $c \in \mathcal{R}^{q,h}, v \in \mathcal{S}^{q,h}$.
Each assignment problem h of sub-problem q, $h \in \mathbb{N}^+$, is formulated as follows [Munkres, 1957]:

$$min \sum_{\substack{o \in \mathcal{R}^{q,h} \\ v \in \mathcal{S}^{q,h}}} o_{c,v} \cdot \overline{x}_{c,v} \tag{5.34}$$

s.t.

$$\sum_{v \in \mathcal{S}^{q,h}} \overline{x}_{c,v} = 1 \qquad \forall c \in \mathcal{R}^{q,h} \tag{5.35}$$

$$\sum_{c \in \mathcal{R}^{q,h}} \overline{x}_{c,v} \leq 1 \qquad \forall v \in \mathcal{S}^{q,h} \tag{5.36}$$

$$\overline{x}_{c,v} \in \{0,1\} \qquad \forall (c,v) \in \{(c,v)|a_{c,v} = 1\} \tag{5.37}$$

$$\overline{x}_{c,v} = 0 \qquad \forall (c,v) \in \{(c,v)|a_{c,v} = 0\} \tag{5.38}$$

where $o_{c,v}$ is the generic element of matrix $\mathbf{O}$ and $a_{c,v}$ is the generic element of the adjacency matrix $\mathbf{A}$. Note that, without loss of generality, it is supposed that in each assignment problem there are more orders than available resources.

The following relationship between variable $\overline{x}_{c,v}$ and the variable $x_{c,v}(t)$ of the MILP formulation holds:

$$\overline{x}_{c,v} = \sum_{t \in I_h} x_{c,v}(t) \tag{5.39}$$

Two preliminary steps have to be performed in order to address the generic assignment problem h: the definition of the adjacency matrix $\mathbf{A}$ and the definition of the generalized cost matrix $\mathbf{O}$.

To this aim, the following binary variables are defined for each truck-order pair (c,v), with $c \in \mathcal{R}^{q,h}$ and $v \in \mathcal{S}^{q,h}$:

$$\overline{\gamma}^{B}_{c,v} = \begin{cases} 1 & \text{if truck } c \text{ has a night rest before reaching location B of} \\ & \text{order } v \\ 0 & \text{otherwise} \end{cases}$$

$$\overline{\gamma}^{C}_{c,v} = \begin{cases} 1 & \text{if truck } c \text{ has a night rest between location B and location} \\ & \text{C of order } v \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{l}_{c,v} = \begin{cases} 1 & \text{if order } v \text{ is performed with delay by truck } c \\ 0 & \text{otherwise} \end{cases}$$

Trucks and orders are characterized by the same parameters considered for the MILP formulation. Moreover, the following additional parameters are introduced:

- $t_c^{avail,q,h} \in \mathcal{T}$: starting time of truck c for the h-th assignment problem of sub-problem q. For $h = 1$, $t_c^{avail,q,h} = t_c^{avail,q}$.
- $d'^{q,h}_{c,v} \in \mathbb{N}$: time distance, expressed in t.u., between the starting position of truck c and location A of order $v \in \mathcal{S}^{q,h}$. For $h = 1$, $d'^{q,h}_{c,v} = d'^{q}_{c,v}$.
- $d''^{q,h}_{c,v} \in \mathbb{N}$: time distance, expressed in t.u., between location C of order $v \in \mathcal{S}^{q,h}$ and the depot of truck c.
- $duty_c^{q,h} \in \mathbb{N}$: remaining duty hours of truck c during time intervals I_h of day q, i.e., the residual working hours of truck c according to the service hours regulations.
- $rest \in \mathbb{N}$: length of the night rest, expressed in t.u..

The adjacency matrix $\mathbf{A}$ and the generalized cost matrix $\mathbf{O}$ can be calculated as follows.

1. **Adjacency Matrix**. The adjacency matrix $\mathbf{A}$ has to be calculated by considering the constraints (5.15) $\div$ (5.22) already introduced in the MILP formulation.
   In particular, for each pair (c, v), $c \in \mathcal{R}^{q,h}, v \in \mathcal{S}^{q,h}$, the following procedure is defined to compute the generic element $a_{c,v}$.

STEP 1 : Initialize the generic element of the adjacency matrix and the binary variables.

Set $a_{c,v} = 0$.

Set $\overline{\gamma}^B_{c,v} = 0$.

Set $\overline{\gamma}^C_{c,v} = 0$.

Set $\bar{l}_{c,v} = 0$.

STEP 2 : Determine if truck c has to take a night rest.

If $duty^{q,h}_c \leq \left(\tau_v + d'^{q,h}_{c,v}\right)$ **then** $\overline{\gamma}^B_{c,v} = 1$

elseif $duty^{q,h}_c \leq \left(\theta_v + d'^{q,h}_{c,v}\right)$ **then** $\overline{\gamma}^C_{c,v} = 1$.

STEP 3 : Check the time and container type constraints.

Compute $cond_1$:

$$t^{A,open}_v \leq t^{avail,q,h}_c + \left(\tau_v + d'^{q,h}_{c,v}\right) \cdot \overline{x}_{c,v} + rest \cdot \overline{\gamma}^B_{c,v} \leq t^{A,close}_v$$

Compute $cond_2$:

$$t^{B,earliest}_v \leq t^{avail,q,h}_c + \left(\tau_v + d'^{q,h}_{c,v}\right) \cdot \overline{x}_{c,v} + rest \cdot \overline{\gamma}^B_{c,v} \leq t^{B,latest}_v + \delta^B_v \cdot \bar{l}_{c,v}$$

Compute $cond_3$:

$$t^{C,open}_v \leq t^{avail,q,h}_c + \left(\theta_v + d'^{q,h}_{c,v}\right) \cdot \overline{x}_{c,v} + rest \cdot \left(\overline{\gamma}^B_{c,v} + \overline{\gamma}^C_{c,v}\right) \leq t^{C,close}_v$$

Compute $cond_4$:

$$t^{avail,q,h}_c + \left(\theta_v + d'^{q,h}_{c,v}\right) \cdot \overline{x}_{c,v} + rest \cdot \left(\overline{\gamma}^B_{c,v} + \overline{\gamma}^C_{c,v}\right) \leq t^{finish}_c$$

Compute $cond_5$:

$$s_{v,e} \leq r_{v,e}$$

If ($cond_1$ **and** $cond_2$ **and** $cond_3$ **and** $cond_4$ **and** $cond_5$) **then** $a_{c,v} = 1$, **Exit**

elseif ($cond_1$ **and** $cond_3$ **and** $cond_4$**and** $cond_5$) **then Go to STEP 4**

else Exit.

STEP 4 : Check the time constraint at location B allowing order delay.

Set $\bar{l}_{c,v} = 1$.

Compute $cond_2$ of **STEP3**.

If $cond_2$ **then** $a_{c,v} = 1$, **Exit**

else $a_{c,v} = 0$, **Exit**.

The first condition of **STEP 2** imposes that, if the sum of the time distance between locations A and B of order v and the time distance between the actual position of truck c and location A of order v exceeds the remaining duty hours of truck c, then the truck has to have a night rest before reaching location B of order v. On the contrary, the second condition of **STEP 2** imposes that the truck c has a night rest between locations B and C of order v if the total length of order v added to the time distance between the actual position of truck c and location A of order v exceeds the remaining duty hours of truck c.
In **STEP 3** the time constraints and the container type constraint are checked. In particular, if all the constraints are respected, then the considered truck-order pairing is marked as feasible. Otherwise, if the constraint on the time required at location B of order v is violated, but all the others are met, then a further check has to be performed. Note that, in this step, the values of $\overline{\gamma}^{B}_{c,v}$ and $\overline{\gamma}^{C}_{c,v}$ determined at **STEP 2** are used.
Finally, **STEP 4** checks if the time constraint at location B of order v is met by delaying the order: if so, the considered truck-order pair is feasible, otherwise the pair is marked as infeasible.

2. **Cost Matrix**. The cost matrix **O** considers, for each truck $c \in \mathcal{R}^{q,h}$ and each order $v \in \mathcal{S}^{q,h}$, the weighted sum of four heterogeneous factors conveniently normalized. In particular, the following costs are considered:

   a) $o^1_{c,v}$ is the distance travelled without any load, and, for the h-th assignment problem, it holds: $o^1_{c,v} = d'^{q,h}_{c,v}$.

b) $o^2_{c,v}$ is a flag equal to 1 if the order is delayed and, for the h-th assignment problem, it holds: $o^2_{c,v} = \bar{l}_{c,v}$.

c) $o^3_{c,v}$ is the time distance between location C of order v and the depot of truck c. For the h-th assignment problem, it holds: $o^3_{c,v} = d''^{q,h}_{c,v}$. This term is considered due to the necessity that each truck returns to its own depot at the end of the overall planning horizon (time T) minimizing the distance travelled without any container.

d) $o^4_{c,v}$ is the term that takes into account the information about the future containers orders. In particular, considered the matrix $\mathbf{M}^{(q+1)}$, $o^4_{c,v}$ penalises the assignment of the order v with a container of type e to the truck c if the location C of order v has low demand for container of type e in the following.

The weights p_i, with $i = 1, 2, 3, 4$, used to combine such cost functions are varied depending on I_h and q. In particular, p_3 is increased as q increases; weight p_4 is increased over h.

Once defined how each assignment problem h is formalized, it is possible to describe how the overall sub-problem q is addressed. In particular, the steps outlined in Fig. 5.4 and described in the following are applied.

1. **Analysis of the truck availability**. For each truck $c \in \mathcal{R}$, parameters $t^{avail,q}_c$ and $d'^{q}_{c,v}$ have to be determined starting from the orders schedule of problem $(q-1)$, as described in the Section "Information about the previous and the following days".

2. **Analysis of the container orders**. The matrix $\mathbf{M}^{(q+1)}$ described in the Section "Information about the previous and the following days" has to be calculated, storing the number and the typologies of container transportation orders at each defined operational region for day $(q+1)$.

3. **Determination of the first subset of orders to be assigned**. The subset $\mathcal{S}^{q,1}$ on which to solve the assignment problem $h = 1$ have to be determined. In particular, the type of container to be transported $e \in \mathcal{E}$ and the time required at location B, $t_v^{B,latest}$, are considered to define such set. Moreover, the subset $\mathcal{R}^{q,1}$ is the set of trucks for which $t_c^{avail,q}$ does not exceed the maximum time in I_1.

4. **Check if the assignment process is completed**. The assignment process is considered completed if all the orders for day q have been assigned or if all the order subsets $\mathcal{S}^{q,h}$ have been analysed.

5. **Truck-order assignment**. The assignment problem on the selected subset of orders is solved through the application of the Munkres algorithm [Munkres, 1957]. In particular, parameters $t_c^{avail,q,h}$, $d_{c,v}^{\prime q,h}$ and $duty_c^{q,h}$ are initialized on the basis of the solution of problem $(q-1)$.
Moreover, the cost matrix weights p_i, $i = 1, 2, 3, 4$, have to be set on the basis of the actual values of h and q.

6. **Update of the trucks availability**. Once solved the assignment problem h, trucks parameters have to be updated as follows:

$$t_c^{avail,q,h+1} = t_c^{avail,q,h} + \sum_{v \in \mathcal{S}^{q,h}} \left(\theta_v + d_{c,v}^{\prime q,h}\right) \cdot \overline{x}_{c,v} + \sum_{v \in \mathcal{S}^{q,h}} rest \cdot \left(\overline{\gamma}_{c,v}^{B} + \overline{\gamma}_{c,v}^{C}\right) \cdot \overline{x}_{c,v} \qquad \forall c \in \mathcal{R}^{q,h}$$

$$duty_c^{q,h+1} = duty_c^{q,h} - \sum_{v \in \mathcal{S}^{q,h}} \left(\theta_v + d_{c,v}^{\prime q,h}\right) \cdot \overline{x}_{c,v} \qquad \forall c \in \mathcal{R}^{q,h}, v \in \mathcal{S}^{q,h}$$

7. **Determination of the new set of orders to be assigned**. The new set of orders to which apply the assignment procedure $\mathcal{S}^{q,h+1}$ is determined on the type of container required and the time requested at location B. Moreover, $\mathcal{R}^{q,h+1}$ is defined accordingly.

8. **KPIs evaluation**. The KPIs described in Section "Decision Module Specification" are calculated for the sub-problem q.

9. **Show results**. The assignment of container transportation orders to trucks and the KPIs characterizing such solutions are proposed to the truck manager in order to support his operations.

5.4.1.3 Discussion about the proposed heuristic

The proposed rolling horizon heuristic allows to address on-line the problem described by the MILP formulation.
The following features should be pointed out:

1. the use of the rolling horizon technique allows to consider some information about the future container transportation demand: to improve the fleet daily truck placement, the future demand per container typology and operational region is analysed;

2. the use of the rolling horizon technique allows the application of the proposed approach also to dynamic frameworks, i.e., contexts in which not all the problem parameters are known at the beginning of the planning horizon;

3. the dummy orders representing the night rests have been replaced by additional terms to be considered in the time constraints. This allows to extend straightforwardly the model to address additional issues of real-life problems, such as the need to perform the night rest at specific parking areas for reefer containers or high value freight transportations.

5.4.2 Decision Module Specification

In this Chapter, three KPIs are considered in order to evaluate the assignment of container transportation orders to trucks proposed by the DSS. In particular,

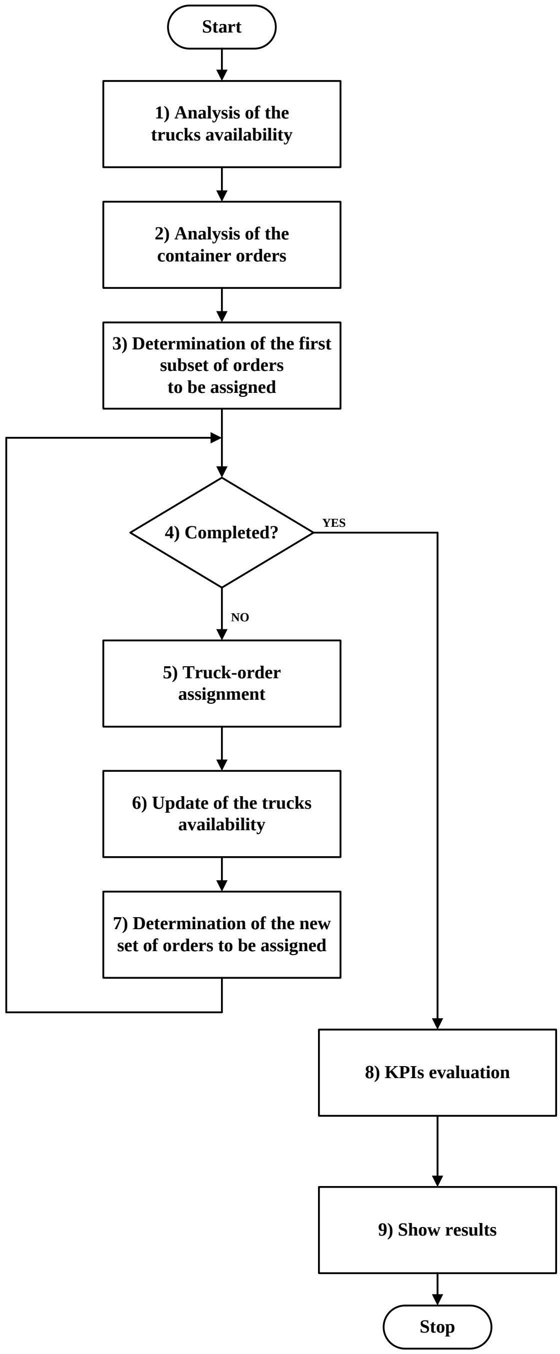

Figure 5.4: Steps performed to solve the single-day problem.

1. **LOS**. In this context, the LOS is defined as the fraction of container transportation orders effectively assigned, i.e., the ratio between the number of orders assigned to a truck and the total number of container transportation requests made by customers. According to the notation introduced for the MILP model and recalling that S is the cardinality of the set of orders to be assigned $\mathcal{S}$, it states:

$$LOS = \frac{f_1}{S} \tag{5.40}$$

At the same time, accordingly to the heuristic notation, for each sub-problem q, it states:

$$LOS^q = \frac{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} \overline{x}_{c,v}}{S^q} \tag{5.41}$$

where S^q is the cardinality of $\mathcal{S}^q$.

2. **Timeliness** (TL). Such KPI is defined as the ratio between the number of container transportation orders performed on time and the total number of container transportation orders effectively assigned, i.e., according to the notation of the MILP:

$$TL = 1 - \frac{f_3}{f_1} \tag{5.42}$$

It states:

$$TL^q = 1 - \frac{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} \overline{l}_{c,v}}{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} \overline{x}_{c,v}} \tag{5.43}$$

3. **Trucks placement** (TP). This KPI considers the total distance travelled by the trucks of the fleet without any load and is defined as the ratio between

the distance required by the orders conformation and the distance effectively driven. Formally,

$$TP = \frac{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S} \\ t \in \mathcal{T}}} (\theta_v \cdot x_{c,v}(t))}{f_2 + \left(\sum_{\substack{c \in \mathcal{R} \\ v \in \mathcal{S} \\ t \in \mathcal{T}}} (\theta_v \cdot x_{c,v}(t)) \right)} \tag{5.44}$$

and

$$TP^q = \frac{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} (\theta_v \cdot \overline{x}_{c,v})}{\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} d'^q_{c,v} + \left(\sum_{\substack{c \in \mathcal{R}^q \\ v \in \mathcal{S}^q}} (\theta_v \cdot \overline{x}_{c,v}) \right)} \tag{5.45}$$

5.4.3 Computational Results

To evaluate the effectiveness of the proposed heuristic, a set of tests on a planning horizon of $Q = 5$ days (Monday to Friday) and $K = 1440$ t.u., where the minute is considered as t.u., has been conducted.
In particular, the considered real-life sized instance is characterized as follows.

Sets

- Total number of *time periods*: $|\mathcal{T}| = T = K \cdot Q = 7200$.
- Total number of *trucks*: $|\mathcal{R}| = R = 500$.
- Total number of *orders* to be assigned during the overall planning horizon of length T: $|\mathcal{S}| = S = 4000$.
- Average number of *orders* to be assigned during each day q of the overall planning horizon of length T: $|\mathcal{S}^q| = S^q = 800$.
- Number of different *container typologies*: $|\mathcal{E}| = E = 6$.

Parameters

- *Geographical area*: locations A, B, C are cities of the Northern and Central Italy.
- *Opening* and *closing times* at locations A and C: $t_v^{A,open}$, $t_v^{A,close}$, $t_v^{C,open}$ and $t_v^{C,close}$ are between 6 a.m. and 8 p.m for all the orders $v \in \mathcal{S}$.
- *Earliest* time and *latest* time at locations B: $t_v^{B,earliest}$ and $t_v^{B,latest}$ are between 8 a.m. and 5 p.m. for all the orders $v \in \mathcal{S}$.

Heuristic implementation

- h: for each sub-problem q, $h \leq 4$, i.e., at most 4 assigned problems are solved per day.
- *Operational regions*: the number of operational regions N is 12.
- *Cost matrix weights*: in order to determine effective weights p_i, $i = 1, 2, 3, 4$, to combine the cost factors, different tests have been performed. Note that, in each assignment problem h, $\sum_i p_i = 1$.

The proposed fast heuristic has been implemented in the MATLAB®software environment and solved on a PC equipped with a Intel i7 3.6 GHz processor and 16 GB RAM. The truck-to-order assignments of a single day are computed in less than 8 minutes in all the cases.

Table 5.2 shows that the performance of the system on the overall planning horizon: in particular, LOS and TL are mainly related on *how* the container orders are performed, while TP expresses the *economic benefit* derived from performing the assigned orders.

Table 5.2: KPIs on the overall planning horizon.

KPI	**Value**
LOS	0.9284
TL	0.8564
TP	1.0170

Table 5.3: KPIs in sub-problem q.

Day	**KPI**		
q	LOS^q	TL^q	TP^q
1	0.9172	0.7847	0.9514
2	0.9463	0.8550	1.0360
3	0.9504	0.8977	1.0686
4	0.8989	0.9171	1.0785
5	0.9293	0.8273	0.9504

Moreover, Table 5.3 shows that most of container orders are effectively assigned, as highlighted by the high values of LOS^q in all the working days of the week. The values of KPI TL^q point out that the possibility of delaying an order is effectively exploited and this allows to improve the values of LOS^q. Finally, TP^q is greater than 1 during the week, i.e., the truck placement is optimal. At the same time, values of TP^q smaller than 1 at day 1 and day 5 can be ascribed to the need of starting and returning to the depot without any load.

5.5 Concluding Remarks

In this Chapter, the container drayage problem has been addressed through the application of the DSS approach. In particular, the pursued objective was to obtain a DSS able to operate on-line, guaranteeing responsive suggestions to the decision maker.

The contributions of this Chapter are the following:

1. a taxonomy for the container drayage problem;
2. the formalization in a UML framework of the typical process followed by the company truck managers in order to assign the requested transportation orders to the available fleet of trucks;
3. the development of a MILP model for the multi-day container drayage problem;
4. the development of a fast heuristic for the multi-day container drayage problem based on the rolling horizon approach, which allows the DSS to operate on-line.

In particular, the developed heuristic allows to handle on-line real-life sized instances, making the proposed DSS a valuable tool to support the trucking companies truck managers in their operations.
The rolling horizon approach guarantees good performance on the overall planning horizon, considering the total number of orders effectively assigned, the total distance travelled without any load and the number of delayed orders.

The results of this Chapter will be included in [Clemente et al., 2016a].

Chapter 6

Conclusions

In this dissertation, the application of a general approach based on the Decision Support System concept to the management of complex systems in transportation and logistics is discussed.
In particular, three problems of great interest nowadays have been addressed: 1) the user-based relocation problem in Car Sharing systems, 2) the smart management of Electric Vehicles charging operations, and 3) the container drayage operations optimization.

A Decision Support System made up of three main components has been applied in all the three cases.

In particular,

1. **User-based relocation problem in Car Sharing systems**. The application of the DSS has underlined that a system of economic incentives based on a simple ICT application and the real time monitoring of the system can increase the number of served users and, therefore, improve the overall service performance. Moreover, the exploitation of the DSS to solve an open problem of the related scientific literature (the optimization of the thresholds of the incentive mechanism) has led to further improvement of the system performance, with benefits for both the users and the Car Sharing company.

2. **Smart management of Electric Vehicles charging operations**. The introduction of the DSS has enabled the formulation of a leader follower approach for the smart management of the charging operations of electric vehicles, which takes into account simultaneously grid and drivers requirement. Moreover, a MILP formulation for the vehicle-to-charging station problem has been proven to be effective in providing the optimal solution considering different drivers utility function.

3. **Container drayage problem**. The development of a fast heuristic based on the rolling horizon approach has allowed to deal with real-life sized instances with a computational time compatible with the typical truck managers operating times without introducing unrealistic simplifications. In particular, the application of the heuristic to a five-day test case has led to good performance on the overall planning horizon, considering the total number of orders effectively assigned, the total distance travelled without any load and the number of delayed orders.

In conclusion, this Thesis contributed to demonstrate the benefits derived from the application of a DSS approach to completely different types of problems in transportation and logistics. Future research will address:

- for the *user-based vehicle relocation problem*, the evaluation of two solutions that could improve the effects of the proposed optimal user-based relocation policy. First, the incentive proposal could be performed during the trips, and not only at the beginning of the rental period. In this case, the time at which the users are asked to change their destinations has to be taken into account, leading to a more complex customers decision process. Second, the determination of the optimal economic incentives on the basis of the specific population considered will be studied;

- for the *smart management of electric vehicles charging operations*, the formulation of the upper-level optimization problem, i.e., the charging infrastructure optimal configuration, and the identification of the best strategy to deal with the whole EVs Charging Smart Management Procedure;

- for the *container drayage problem*, a deeper analysis of the policies to be applied for the determination of the subsets of orders and trucks considered in each sub-problem of the fast heuristic, as well as the identification of a rule to determine the coefficients used to weight the cost matrix of each assignment problem.

Moreover, the application of the DSS to completely different types of problems will be considered: in particular, the design of a DSS based on Quasi-Artificial Intelligence techniques for the cyber-physical security of space control ground stations will be addressed.

Chapter 7

List of Acronyms

The following table describes the meaning of the acronyms and abbreviations used throughout the thesis.

Abbreviation	**Meaning**
AI	Artificial Intelligence
CPS	Cyber-Physical System
CS	Car Sharing
DBMS	DataBase Management System
DES	Discrete Event System
DG	Distributed Generation
DSS	Decision Support System
EV	Electric Vehicle
EXPO	EXPOnential Distribution
HTW	Hard Time Windows
ICE	Internal Combustion Engine
ICT	Information and Communication Technology
IT	Information Technology
KPI	Key Performance Indicator
LOS	Level of Service
MILP	Mixed Integer Linear Programming

PN	Petri Net
PSO	Particle Swarm Optimization
RS	Random Switch
SoC	State of Charge
SoS	System of System
STW	Soft Time Windows
TPN	Timed Petri Net
TRIA	TRIAngular Distribution
t.u.	time units
UML	Unified Modeling Language
VCSA	Vehicle-to-Charging Station Assignment Problem

Bibliography

[Alfian et al., 2014] Alfian, G., Rhee, J., and Yoon, B. (2014). A relocation simulation model for one-way carsharing service. In *Industrial Technology (ICIT), 2014 IEEE International Conference on*, pages 718–723. IEEE.

[Alla and David, 1998] Alla, H. and David, R. (1998). Continuous and hybrid petri nets. *Journal of Circuits, Systems, and Computers*, 8(01):159–188.

[Alter, 1980] Alter, S. (1980). *Decision support systems: current practice and continuing challenges*, volume 157. Addison-Wesley Reading, MA.

[Andreotti et al., 2012] Andreotti, A., Carpinelli, G., Mottola, F., and Proto, D. (2012). A review of single-objective optimization models for plug-in vehicles operation in smart grids part ii: Numerical applications to vehicles fleets. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE.

[Barnhart and Laporte, 2006] Barnhart, C. and Laporte, G. (2006). *Handbooks in Operations Research & Management Science: Transportation: Transportation*, volume 14. Elsevier.

[Barth and Shaheen, 2002] Barth, M. and Shaheen, S. A. (2002). Shared-use vehicle systems: Framework for classifying carsharing, station cars, and combined approaches. *Transportation Research Record: Journal of the Transportation Research Board*, 1791(1):105–112.

[Barth and Todd, 1999] Barth, M. and Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. *Transportation Research Part C: Emerging Technologies*, 7(4):237–259.

[Barth et al., 2003] Barth, M., Todd, M., and Shaheen, S. (2003). Intelligent transportation technology elements and operational methodologies for shared-use vehicle systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1841(1):99–108.

[Bektas, 2006] Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219.

[Beraldi et al., 2008] Beraldi, P., Ghiani, G., Grieco, A., and Guerriero, E. (2008). Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. *Computers & Operations Research*, 35(11):3644–3656.

[Bianchessi et al., 2013] Bianchessi, A., Formentin, S., and Savaresi, S. (2013). Active fleet balancing in vehicle sharing systems via feedback dynamic pricing. In *Proc. IEEE 16th Conf. Intelligent Transport Systems*.

[Bidgoli, 1989] Bidgoli, H. (1989). *Decision support systems: Principles and practice*. West Group.

[Boccara, 2010] Boccara, N. (2010). *Modeling complex systems*. Springer Science & Business Media.

[Booch et al., 1999] Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The unified modeling language user guide*. Pearson Education India.

[Boschian et al., 2011] Boschian, V., Dotoli, M., Fanti, M. P., Iacobellis, G., and Ukovich, W. (2011). A metamodeling approach to the management of intermodal transportation networks. *Automation Science and Engineering, IEEE Transactions on*, 8(3):457–469.

[Boyacı et al., 2015] Boyacı, B., Zografos, K. G., and Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. *European Journal of Operational Research*, 240(3):718–733.

[Braekers et al., 2013] Braekers, K., Caris, A., and Janssens, G. K. (2013). Integrated planning of loaded and empty container movements. *OR spectrum*, 35(2):457–478.

[Braekers et al., 2014] Braekers, K., Caris, A., and Janssens, G. K. (2014). Bi-objective optimization of drayage operations in the service area of intermodal terminals. *Transportation Research Part E: Logistics and Transportation Review*, 65:50–69.

[Bruglieri et al., 2014] Bruglieri, M., Colorni, A., and Luè, A. (2014). The relocation problem for the one-way electric vehicle sharing. *Networks*, 64(4):292–305.

[Caballini et al., 2013] Caballini, C., Sacone, S., and Saeednia, M. (2013). A decomposition approach for optimizing truck trips for a single carrier. pages 1625–1630.

[Caris and Janssens, 2009] Caris, A. and Janssens, G. K. (2009). A local search heuristic for the pre-and end-haulage of intermodal container terminals. *Computers & Operations Research*, 36(10):2763–2772.

[Cascetta et al., 2015] Cascetta, E., Carteni, A., Pagliara, F., and Montanino, M. (2015). A new look at planning and designing transportation systems: A decision-making model based on cognitive rationality, stakeholder engagement and quantitative methods. *Transport policy*, 38:27–39.

[Cassandras and Lafortune, 2008] Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer US.

[Cats-Baril and Huber, 1987] Cats-Baril, W. L. and Huber, G. P. (1987). Decision support systems for ill-structured problems: an empirical study. *Decision Sciences*, 18(3):350–372.

[Cepolina and Farina, 2012] Cepolina, E. and Farina, A. (2012). Urban car sharing: An overview of relocation strategies. *WIT Transactions on the Built Environment*, 128:419–431.

[Charbonnier et al., 2005] Charbonnier, S., Garcia-Beltan, C., Cadet, C., and Gentil, S. (2005). Trends extraction and analysis for complex system monitoring and decision support. *Engineering Applications of Artificial Intelligence*, 18(1):21–36.

[Cheung et al., 2008] Cheung, R. K., Shi, N., Powell, W. B., and Simao, H. P. (2008). An attribute–decision model for cross-border drayage problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(2):217–234.

[Clement-Nyns et al., 2010] Clement-Nyns, K., Haesen, E., and Driesen, J. (2010). The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *Power Systems, IEEE Transactions on*, 25(1):371–380.

[Clement-Nyns et al., 2011] Clement-Nyns, K., Haesen, E., and Driesen, J. (2011). The impact of vehicle-to-grid on the distribution grid. *Electric Power Systems Research*, 81(1):185–192.

[Clemente et al., 2013a] Clemente, M., Fanti, M. P., Iacobellis, G., and Ukovich, W. (2013a). A discrete-event simulation approach for the management of a car sharing service. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 403–408. IEEE.

[Clemente et al., 2013b] Clemente, M., Fanti, M. P., Iacobellis, G., and Ukovich, W. (2013b). Modeling and simulation of an electric-car sharing system. In *25th European Modeling and Simulation Symposium*, pages 587–592. IEEE.

[Clemente et al., 2013c] Clemente, M., Fanti, M. P., Mangini, A. M., and Ukovich, W. (2013c). The vehicle relocation problem in car sharing systems: Modeling and simulation in a petri net framework. In *Petri Nets*, pages 250–269. Springer.

[Clemente et al., 2016a] Clemente, M., Fanti, M. P., Nolich, M., Stecco, G., and Ukovich, W. (2016a). Modeling and solving the multi-day container drayage problem. In *Conference on Decision and Control (CDC), 2016 IEEE Conference on Decision and Control,[to be submitted]*.

[Clemente et al., 2015] Clemente, M., Fanti, M. P., Nolich, M., and Ukovich, W. (2015). A decision support system for the management of an electric-car sharing system. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 533–538. IEEE.

[Clemente et al., 2016b] Clemente, M., Fanti, M. P., Nolich, M., and Ukovich, W. (2016b). A decision support system for user-based vehicle relocation in car sharing systems. *Systems, Man and Cybernetics, IEEE Transactions on,[under review]*.

[Clemente et al., 2014] Clemente, M., Fanti, M. P., and Ukovich, W. (2014). Smart management of electric vehicles charging operations: the vehicle-to-charging station assignment problem. In *19th World Congress of the International Federation of Automatic Control, Cape Town, South Africa*, pages 24–29.

[Coslovich et al., 2006] Coslovich, L., Pesenti, R., and Ukovich, W. (2006). Minimizing fleet operating costs for a container transportation company. *European Journal of Operational Research*, 171(3):776–786.

[Di Febbraro et al., 2012] Di Febbraro, A., Sacco, N., and Saeednia, M. (2012). One-way carsharing: Solving the relocation problem. *Transportation Research Record: Journal of the Transportation Research Board*, pages 113–120.

[Dotoli et al., 2006] Dotoli, M., Fanti, M. P., and Meloni, C. (2006). A signal timing plan formulation for urban traffic control. *Control Engineering Practice*, 14(11):1297–1311.

[Dumas et al., 1991] Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22.

[Elam et al., 1980] Elam, J. J., Henderson, J. C., and Miller, L. W. (1980). Model management systems: An approach to decision support in complex organizations. Technical report, DTIC Document.

[Escudero et al., 2013] Escudero, A., Muñuzuri, J., Guadix, J., and Arango, C. (2013). Dynamic approach to solve the daily drayage problem with transit time uncertainty. *Computers in Industry*, 64(2):165–175.

[Fanti et al., 2015] Fanti, M. P., Clemente, M., and Ukovich, W. (2015). A decision support system approach for systems of systems: an application to the electric vehicle management. In *Toward Emerging Technology for Harbour sYstems and Services, TETHYS, 2015*.

[Fanti et al., 2013] Fanti, M. P., Mangini, A. M., Dotoli, M., and Ukovich, W. (2013). A three-level strategy for the design and performance evaluation of hospital departments. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(4):742–756.

[Fanti et al., 2014] Fanti, M. P., Mangini, A. M., Pedroncelli, G., and Ukovich, W. (2014). Assignment of electrical vehicles to charging stations by a distributed approach. In *Control Conference (ECC), 2014 European*, pages 1888–1893. IEEE.

[Filip, 2008] Filip, F. G. (2008). Decision support and control for large-scale complex systems. *Annual Reviews in Control*, 32(1):61–70.

[Gan et al., 2012] Gan, L., Topcu, U., and Low, S. H. (2012). Stochastic distributed protocol for electric vehicle charging with discrete charging rate. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–8. IEEE.

[Gan et al., 2013] Gan, L., Topcu, U., and Low, S. H. (2013). Optimal decentralized protocol for electric vehicle charging. *Power Systems, IEEE Transactions on*, 28(2):940–951.

[George and Xia, 2011] George, D. K. and Xia, C. H. (2011). Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, 211(1):198–207.

[Gharbaoui et al., 2012] Gharbaoui, M., Valcarenghi, L., Bruno, R., Martini, B., Conti, M., and Castoldi, P. (2012). An advanced smart management system for electric vehicle recharge. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*, pages 1–8. IEEE.

[González et al., 2012] González, E. J., Sánchez, L. A., and Castro, A. F. H. (2012). *Artificial Intelligence Resources in Control and Automation Engineering*. Bentham Science Publishers.

[Gorry and Morton, 1971] Gorry, G. A. and Morton, M. S. S. (1971). *A framework for management information systems*, volume 13. Massachusetts Institute of Technology.

[Hamdouni et al., 2013] Hamdouni, G., Elomri, A., and Moussaid, K. (2013). Assignement of container trucks of a road transport company with consideration of the load balancing problem. *International Journal of Computer Science and Applications*, 10(2):1–14.

[Hashimoto et al., 2006] Hashimoto, H., Ibaraki, T., Imahori, S., and Yagiura, M. (2006). The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics*, 154(16):2271–2290.

[Hassan et al., 2005] Hassan, R., Cohanim, B., De Weck, O., and Venter, G. (2005). A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, pages 1–13.

[Holsapple, 2008] Holsapple, C. W. (2008). Dss architecture and types. In *Handbook on Decision Support Systems 1*, pages 163–189. Springer.

[Ileri et al., 2006] Ileri, Y., Bazaraa, M., Gifford, T., Nemhauser, G., Sokol, J., and Wikum, E. (2006). An optimization approach for planning daily drayage operations. *Central European Journal of Operations Research*, 14(2):141–156.

[Jamshidi, 2011] Jamshidi, M. E. (2011). *Systems of systems engineering: innovation for the twenty-first century.* New York: John Wiley & Sons.

[Janssens and Braekers, 2011] Janssens, G. and Braekers, K. (2011). An exact algorithm for the full-truckload pick-up and delivery problem with time windows: concept and implementation details.

[Jin et al., 2013] Jin, R., Wang, B., Zhang, P., and Luh, P. B. (2013). Decentralised online charging scheduling for large populations of electric vehicles: a cyber-physical system approach. *International Journal of Parallel, Emergent and Distributed Systems*, 28(1):29–45.

[Jorge et al., 2014] Jorge, D., Correia, G. H., Barnhart, C., et al. (2014). Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. *Intelligent Transportation Systems, IEEE Transactions on*, 15(4):1667–1675.

[Karfopoulos and Hatziargyriou, 2013] Karfopoulos, E. L. and Hatziargyriou, N. D. (2013). A multi-agent system for controlled charging of a large population of electric vehicles. *Power Systems, IEEE Transactions on*, 28(2):1196–1204.

[Keen and Morton, 1978] Keen, P. G. and Morton, M. S. S. (1978). *Decision support systems: an organizational perspective*, volume 35. Addison-Wesley Reading, MA.

[Kek et al., 2009] Kek, A. G., Cheu, R. L., Meng, Q., and Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):149–158.

[Kelton et al., 2002] Kelton, W. D., Sadowski, R. P., and Sadowski, D. A. (2002). *Simulation with ARENA*, volume 3. McGraw-Hill New York.

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE.

[Lai et al., 2013] Lai, M., Crainic, T. G., Di Francesco, M., and Zuddas, P. (2013). An heuristic search for the routing of heterogeneous trucks with single and double container loads. *Transportation Research Part E: Logistics and Transportation Review*, 56:108–118.

[Li et al., 2011] Li, Q., Cui, T., Negi, R., Franchetti, F., and Ilic, M. D. (2011). Online decentralized charging of plug-in electric vehicles in power systems. *arXiv preprint arXiv:1106.5063*.

[Maier, 1996] Maier, M. W. (1996). Architecting principles for systems-of-systems. In *INCOSE International Symposium*, volume 6, pages 565–573. Wiey Online Library.

[Mallach, 1994] Mallach, E. (1994). Understanding decision support and expert systems.

[Mallach, 2000] Mallach, E. G. (2000). *Decision support and data warehouse systems*. McGraw-Hill Higher Education.

[Mann and Watson, 1984] Mann, R. I. and Watson, H. J. (1984). A contingency model for user involvement in dss development. *MIS Quarterly*, pages 27–38.

[Manzini and Bindi, 2009] Manzini, R. and Bindi, F. (2009). Strategic design and operational management optimization of a multi stage physical distribution system. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):915–936.

[Marsan et al., 1994] Marsan, M. A., Balbo, G., Conte, G., Donatelli, S., and Franceschinis, G. (1994). *Modelling with generalized stochastic Petri nets*. John Wiley & Sons, Inc.

[Munkres, 1957] Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

[Nakayama et al., 2002] Nakayama, S., Yamamoto, T., and Kitamura, R. (2002). Simulation analysis for the management of an electric vehicle-sharing system: Case of the kyoto public-car system. *Transportation Research Record: Journal of the Transportation Research Board*, (1791):99–104.

[Nossack and Pesch, 2013] Nossack, J. and Pesch, E. (2013). A truck scheduling problem arising in intermodal container transportation. *European Journal of Operational Research*, 230(3):666–680.

[Nourinejad and Roorda, 2014] Nourinejad, M. and Roorda, M. J. (2014). A dynamic carsharing decision support system. *Transportation research part E: logistics and transportation review*, 66:36–50.

[Nourinejad and Roorda, 2015] Nourinejad, M. and Roorda, M. J. (2015). Carsharing operations policies: a comparison between one-way and two-way systems. *Transportation*, 42(3):497–518.

[Pazour and Neubert, 2013] Pazour, J. A. and Neubert, L. C. (2013). Routing and scheduling of cross-town drayage operations at jb hunt transport. *Interfaces*, 43(2):117–129.

[Peng et al., 2012] Peng, H., Jinkuan, W., and Yinghua, H. (2012). Dynamic-priority-based real-time charging management for plug-in electric vehicles in smart grid. In *Control Conference (CCC), 2012 31st Chinese*, pages 2432–2436. IEEE.

[Pentico, 2007] Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176(2):774–793.

[Perez and Behdinan, 2007] Perez, R. and Behdinan, K. (2007). Particle swarm approach for structural design optimization. *Computers & Structures*, 85(19):1579–1588.

[Peterson, 1981] Peterson, J. L. (1981). Petri net theory and the modeling of systems.

[Porteus, 2002] Porteus, E. L. (2002). *Foundations of stochastic inventory theory.* Stanford University Press.

[Power, 2002] Power, D. (2002). Decision support systems: Concepts and resources for managers. *Studies in Informatics and Control*, 11(4):349–350.

[Qin and Zhang, 2011] Qin, H. and Zhang, W. (2011). Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. In *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*, pages 51–60. ACM.

[Sage, 1991] Sage, A. P. (1991). *Decision support systems.* Wiley Online Library.

[Sahin et al., 2013] Sahin, F., Narayanan, A., and Robinson, E. P. (2013). Rolling horizon planning in supply chains: review, implications and directions for future research. *International Journal of Production Research*, 51(18):5413–5436.

[Solomon, 1987] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265.

[Sprague Jr and Watson, 1996] Sprague Jr, R. H. and Watson, H. J. (1996). *Decision support for management*. Prentice-Hall, Inc.

[Sun et al., 2014] Sun, X., Garg, M., Balaporia, Z., Bailey, K., and Gifford, T. (2014). Optimizing transportation by inventory routing and workload balancing: Optimizing daily dray operations across an intermodal freight network. *Interfaces*.

[Turban et al., 2007] Turban, E., Sharda, R., Delen, D., and Efraim, T. (2007). *Decision support and business intelligence systems*. Pearson Education India.

[Uesugi et al., 2007] Uesugi, K., Mukai, N., and Watanabe, T. (2007). Optimization of vehicle assignment for car sharing system. In *Knowledge-based intelligent information and engineering systems*, pages 1105–1111. Springer.

[Vandael et al., 2011] Vandael, S., Boucké, N., Holvoet, T., De Craemer, K., and Deconinck, G. (2011). Decentralized coordination of plug-in hybrid vehicles for imbalance reduction in a smart grid. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 803–810. International Foundation for Autonomous Agents and Multiagent Systems.

[Vidović et al., 2012] Vidović, M., Nikolić, M., and Popović, D. (2012). Two mathematical formulations for the containers drayage problem with time windows. In *2nd International Conference on Supply Chains ICSC*.

[Wallach, 1993] Wallach, E. G. (1993). *Understanding decision support systems and expert systems*. McGraw-Hill Professional.

[Wolsey, 1998] Wolsey, L. A. (1998). *Integer programming*, volume 42. Wiley New York.

[Xu and Pan, 2012] Xu, Y. and Pan, F. (2012). Scheduling for charging plug-in hybrid electric vehicles. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 2495–2501. IEEE.

[Xue et al., 2014] Xue, Z., Zhang, C., Lin, W.-H., Miao, L., and Yang, P. (2014). A tabu search heuristic for the local container drayage problem under a new operation mode. *Transportation Research Part E: Logistics and Transportation Review*, 62:136–150.

[Yang et al., 2004] Yang, J., Jaillet, P., and Mahmassani, H. (2004). Real-time multivehicle truckload pickup and delivery problems. *Transportation Science*, 38(2):135–148.

[Zhang et al., 2014] Zhang, R., Lu, J.-C., and Wang, D. (2014). Container drayage problem with flexible orders and its near real-time solution strategies. *Transportation Research Part E: Logistics and Transportation Review*, 61:235–251.

[Zhang et al., 2010] Zhang, R., Yun, W. Y., and Kopfer, H. (2010). Heuristic-based truck scheduling for inland container transportation. *OR spectrum*, 32(3):787–808.

[Zhang et al., 2013] Zhang, R., Yun, W. Y., and Kopfer, H. (2013). Multi-size container transportation by truck: modeling and optimization. *Flexible Services and Manufacturing Journal*, pages 1–28.

[Zhang et al., 2009] Zhang, R., Yun, W. Y., and Moon, I. (2009). A reactive tabu search algorithm for the multi-depot container truck transportation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):904–914.