A Decision Support System for User-Based Vehicle Relocation in Car Sharing Systems

Publisher version DOI: 10.1109/TSMC.2017.2665881

A Decision Support System for User-Based Vehicle Relocation in Car Sharing Systems

Monica Clemente, Maria Pia Fanti, *Senior Member, IEEE,* Giorgio Iacobellis, Massimiliano Nolich, *Member, IEEE*, and Walter Ukovich, *Member, IEEE*

***Abstract*—Car Sharing (CS) services are promising solutions complementary to the classic public transport forms. In order to make CS effectively competitive, suitable planning and management strategies are required. This paper presents a Decision Support System for handling the user-based vehicle relocation problem by applying economic incentives ruled by a threshold policy. Unlike the existing approaches, a methodology is proposed for determining the optimal threshold, which considers explicitly the stochastic reactions of the customers to the incentives. To this aim, the CS system is described in detail by Unified Modelling Language diagrams and is modelled in a Discrete Event System framework. Moreover, a closed-loop control strategy is introduced to implement the vehicle relocation policy on the basis of the system state and the best threshold values, evaluated by discrete event simulation and Particle Swarm Optimization. A case study simulation analysis shows that the proposed DSS management strategy can significantly improve the system performance.**

***Index Terms*—Car Sharing, Decision Support System, Optimization, Discrete Event Simulation.**

I. INTRODUCTION

In recent years, the pressing need of improving the quality of the air in urban areas encourages the research of new mobility solutions able to reduce the pollutant emissions and the traffic congestion. 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 [1].

Nevertheless, in such 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 very complex. In this context, the application of the modern Information and Communication Technologies (ICT) is essential [2].

Rental rules play a central role in determining the attractiveness of a 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 [1]. Consequently, vehicle relocation activities are necessary to satisfy users' requests at any time: for this reason, the so-called *vehicle relocation problem* [3] is a fundamental CS management problem.

Different approaches and solutions for the vehicle relocation problem have been proposed: in particular, a pursued solution is to influence the customers travel behavior in order to make them ensure the vehicles number balance among the parking areas (*user-based approach*). In this case, a particularly critical issue is determining how to effectively influence the customers: a solution proposed by different authors is to introduce a set of economic incentives based on a threshold policy [4], [5]. However, generally these papers do not consider a methodology for determining the values of the thresholds, but they take as reference the mean number of vehicles available in the system. Hence, they do not consider the difference of the demands in each parking area and the real time state of the system.

This paper presents a model based Decision Support System (DSS) devoted to face the vehicle relocation problem by a user-based approach. In particular, the proposed vehicle relocation strategy is a closed-loop control scheme with the objective of incentivizing the users to drop off the vehicles in suitable parking areas. The control uses the state of the system (i.e., the number of vehicles and the customers waiting for an available vehicle) and an optimization algorithm to determine the minimum number of vehicles necessary in each parking areas (threshold). Then, the control applies the incentive policy in order to influence the customers' behaviour.

The DSS is specified by following three steps: first, a meta-modeling technique based on the Unified Modeling Language (UML) is presented [6]; second, the system is described as a Discrete Event System (DES) model and is simulated in a discrete-event simulation framework; third, an optimization procedure based on the simulation and the Particle Swarm Optimization (PSO) [7] is formulated to select the best value of the thresholds. In particular, the thresholds are determined by utilizing the performance indicators obtained by the discrete event simulation and taking into account two important aspects: i) the probability that the users drop off the car in a different parking area from their original destinations; ii) the degree of the affinity between two parking areas, determined considering the urban and population characteristics.

Manuscript received....

M. Clemente, M. Nolich and W. Ukovich are with the Department of Engineering and Architecture, University of Trieste, 34127 Trieste, Italy (e-mail: monica.clemente@phd.units.it; {mnolich; ukovich}@units.it).

M. P. Fanti and G. Iacobellis are with the Department of Electrical and Information Engineering, Polytechnic of Bari, 70125 Bari, Italy (e-mail: {mariapia.fanti; giorgio.iacobellis}@poliba.it).

Hence, the contribution of the paper is threefold: i) the formalization of a CS system model by a UML description and a DES framework; ii) the development of a discrete-event simulator to mime the CS system dynamics, taking into account the users' behaviour; iii) the presentation of a user-based relocation strategy for the vehicle relocation problem and of a simulation-optimization procedure for the determination of the optimal threshold values on the basis of the system state knowledge.

The remainder of the paper is structured as follows. In Section II a taxonomy of the vehicle relocation problem is introduced. Section III proposes the DSS solution and approach. Section IV describes the CS system by UML diagrams and Section V presents the DES and simulation models. Moreover, Section VI outlines the considered optimization procedure and Section VII analyses the application of the DSS for solving the relocation problem of a CS system designed for Trieste (Italy). Finally, in Section VIII conclusions and future works are summarized.

II. Background about the Vehicle Relocation Problem

In this Section, a taxonomy of the vehicle relocation problem is presented in order to clearly point out the motivation and the original contribution of this work.

Several approaches to the vehicle relocation problem are studied in 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 and *operator-based* and *user-based* strategies can be distinguished. In an operator-based strategy, system staff relocates the vehicles when needed, but in this case additional trips without customers are necessary [8], [9], [10], [11], [12], [13], [14]. 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 [15], [5], [4]. From both an economic and an environmental point of view, the second solution is preferable. In recent years, additional solutions to perform vehicle relocations are examined: for example, an easy relocation technique exploiting automatic parking and platooning is studied in [16].

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). In this case, additional strategies can be considered in order to reduce the number of relocation operations, such as particular schemes for allocating vehicles to trips [17]. On the other hand, when a *real-time* monitoring of the system is implemented, relocations are performed on the basis of the current system state.

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 [8].

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 [14], [9]. 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 [4], [5].

Table I 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*). If it is not possible to apply one of the classification factors for a specific paper, "n.a." is reported in the corresponding cell of Table I.
As Table I 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 methodologies to evaluate and implement the optimal values of the vehicle thresholds for the user-based relocation is an open problem.

Due to the complexity of the vehicle relocation problem, some authors propose a DSS approach to handle it, and a combination of optimization and simulation is applied: however, most of such works focus on the operator-based relocation strategy. In particular, [9] 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 [14] 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 [22] 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 [9], [14] and [22] 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.
This paper gives a contribution in this context: a DSS is proposed to solve the user-based relocation problem and a system of economic incentives is introduced in order to invite the customers to return the cars where they are mostly needed. With the aim of determining when such incentives have to be applied, a threshold policy similar to the one considered in

TABLE I
LITERATURE REVIEW CLASSIFICATION

Model / Reference	Relocation Modality		Relocation Time		Relocation Control		Set-point	
	Operator	User	Offline	Real-time	Non-predictive	Predictive	A-priori	Optimized
Alfian et al., 2014 [13]	*		*			*	*	
Barth and Todd, 1999 [8]	*			*	*	*	*	
Bianchessi et al., 2013 [5]		*		*	*		*	
Boyaci et al., 2015 [11]	*			*	*			*
Bruglieri et al., 2014 [12]	*			*	*		*	
Correia et al., 2012 [17]	*		*		*	*	*	
Di Febbraro et al., 2012 [4]		*		*	*		*	
Jorge et al., 2014 [10]	*			*		*		*
Jorge et al., 2015 [18]		*		*		*		*
Kek et al., 2006 [19]	*			*		*	*	
Kek et al., 2009 [9]	*			*	*			*
Marouf et al., 2014 [16]	*			*	*		n.a.	n.a.
Nourinejad and Roorda, 2014 [14]	*			*	*			*
Santos and Correia, 2015 [20]	*			*		*		*
Uesugi et al., 2007 [15]		*		*	*		*	
Zakaria et al., 2014 [21]	*			*		*		*

the classical *stochastic inventory problem* [23] is defined: if the number of vehicles in the parking area is less than a given threshold, then the incentive is applied for such parking area. Moreover, an optimization algorithm based on the PSO and the system simulation is presented to solve the critical issue of the optimal threshold determination.
More precisely, in designing the DSS the following assumptions are considered:

- the relocation activities are performed directly by the users (*user-based*);
- the incentive for a parking area is triggered only when a minimum number of available vehicles is reached (*non-predictive*);
- the system status is monitored at regular intervals throughout the day (*real-time*);
- the minimun number of vehicles that should be available in each parking area is determined with the aim of optimizing the overall system performance (*optimized set-point*);
- the suggestion of the incentivized parking areas is proposed to the customer at the beginning of the rental period (before the effective usage of the car).

III. THE DECISION SUPPORT SYSTEM STRUCTURE

In this Section the architecture of the DSS proposed to solve the user-based vehicle relocation problem is described.

A DSS is an "interactive computer-based system, which helps decision makers utilize data and models to solve unstructured problems" [24]. According to [25], the DSS proposed in this work includes three main components: the *data component*, which handles all the data and the information that the DSS needs to operate; the *interface component*, which interacts with the real system by means of a set of geographically distributed communication modules and maintains the consistency between the models contained in the DSS and the real system that they represent; the *model component*, which includes all the knowledge and the tools useful to provide support to the decision makers.

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 system is the *model component*. It includes three main modules: the *simulation module*, the *optimization module* and the *decision module*.

Fig. 1 shows the DSS components and modules and the two main actors with whom it interacts: the *decision maker*, i.e., the park manager, and the *CS system*, which includes all the service parking areas and vehicles.
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.

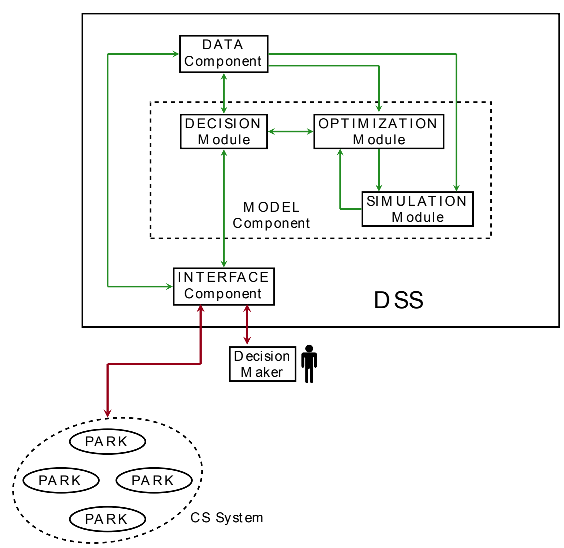


Fig. 1. The Decision Support System architecture and the connections with the CS System and the Decision Maker.

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. The *decision module* receives the current system state from the *data component* and, on the basis of it, determines if it is necessary to trigger a new simulation-optimization procedure in order to optimize in real time the overall system performances.

The *simulation module* is used to evaluate how the CS system operates, i.e., to mime accurately the behaviour of the considered system on the basis of the DSS historical data and the configuration inputs. Whenever it is necessary, the *decision module* triggers a new decision process, the *optimization module* starts a new simulation campaign and the *data component* provides to the *simulation module* the necessary input parameters.

The role of the *optimization module* is to identify the optimal thresholds vector for the incentive mechanism. In order to determine such vector, it is necessary to evaluate its impact on the overall system performance. However, this strictly depends on the customers reaction to the received incentives. Therefore, a closed-loop simulation-optimization approach is considered: indeed, the simulation is suitable to take into account the stochasticity of customers' behaviour while identifying the optimal threshold vector.

The proposed DSS is designed by a three-phase approach.

- First, a detailed analysis of the problem is carried out: with the aim of identifying the structure and the behaviour 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 [6]. The choice of using such a graphical and textual formalism is due to its straightforward translation into simulation models [26]. Moreover, the structural and environmental aspects of the considered system are modelled by *class* and *activity* diagrams, respectively.
- Second, on the basis of the UML description, a DES model for a generic CS system is specified: the UML activity diagram and the DES model allow specifying the simulation module of the DSS model component.
- Third, the optimization procedure is defined: in order to solve the user-based vehicle relocation problem, a simulation-optimization approach is proposed. In particular, the optimization module of the DSS implements a PSO algorithm whose fitness function is evaluated through the simulation module.

IV. Car Sharing System Description

This Section describes the first phase of the development of the proposed DSS by considering two aspects: i) the structural description; ii) the behavioral description.

A. Structural description

The structural view of the CS service is described by the UML class diagrams [6], useful to represent the different types of objects that compose a generic system and the relationships between them. In Fig. 2 the class diagram representing the structure of a generic CS service is depicted: 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.

In particular, the following structural components are identified: the *Customer*; the *Operator*, i.e., the employee of the CS organization, with the child class *Park Manager*; the *Vehicle*, with the two child classes *Traditional Vehicle* and *Electric Vehicle*; the *Parking Area*, with the two child classes *Traditional* and *Charging Station*, i.e., a parking equipped with an EV charging infrastructure.

Moreover, the following association classes are highlighted: *Rental* (between the classes *Customer*, *Parking Area* and *Vehicle*); *Relocation* (between the classes *Parking Area*, *Vehicle* and *Operator*); *Maintenance* (between the classes *Vehicle* and *Operator*); *Emergency* (between the classes *Vehicle* and *Operator*); *Purchase/Substitution* (between the classes *Vehicle* and *Operator*); *Share a Vehicle* (between different instances of class *Customer*).

The structure highlighted by the class diagram determines the requirements of the *data component* of the DSS and, therefore, guides its development.

B. Behavioural Description

Every process characterizing the dynamics of the CS system can be described by an activity diagram. Indeed, the purpose of the activity diagrams is to clarify the sequence and the dependences of the actions representing a process occurring in the system. Every process involves different actors, which are objects derived from the class diagram of Fig. 2: in the activity diagrams it is clearly defined which actors are responsible for which actions and the sequences of such actions.

Fig. 3 describes the vehicle rental process after the introduction of the proposed user-based relocation strategy. 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, 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

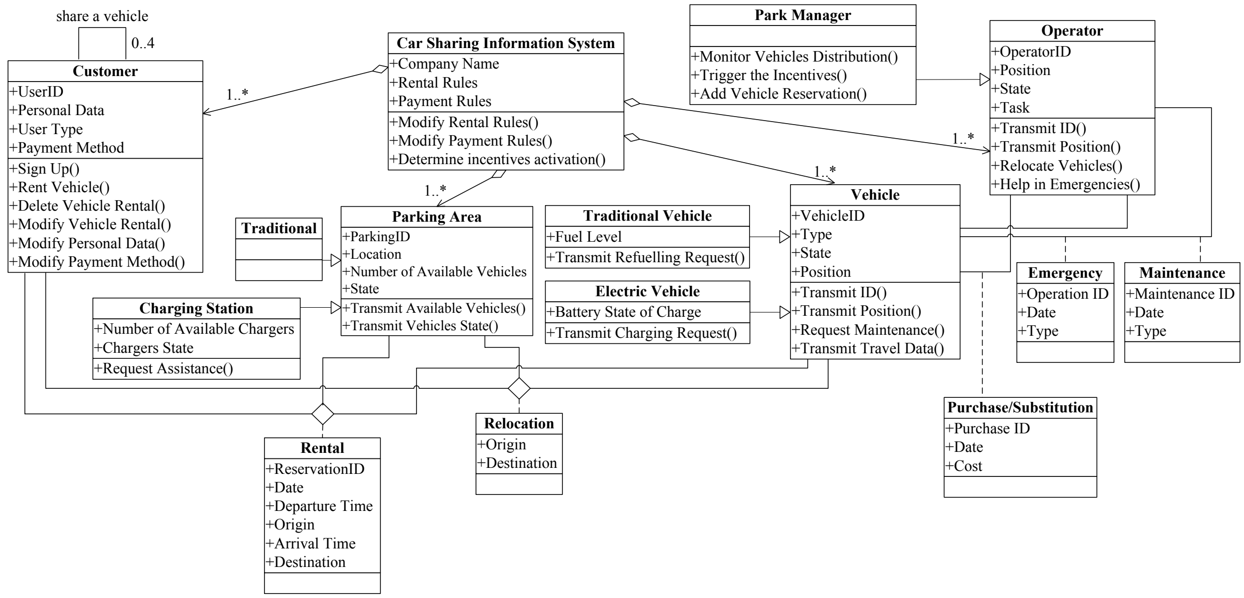


Fig. 2. The CS service class diagram pointing out the main components of a generic CS system and their relationships: *Customer*, *Parking Area*, *Vehicle*, *Operator*, *CS Information System*.

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

V. Discrete Event System and Simulation Models

This Section describes the second phase of the development of the proposed DSS. In particular, first a DES model for a generic CS system is described. Second, the proposed user-based relocation strategy is introduced and the DSS simulation model is designed.

A. Discrete Event System Model

In this subsection 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 [27].

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, 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:

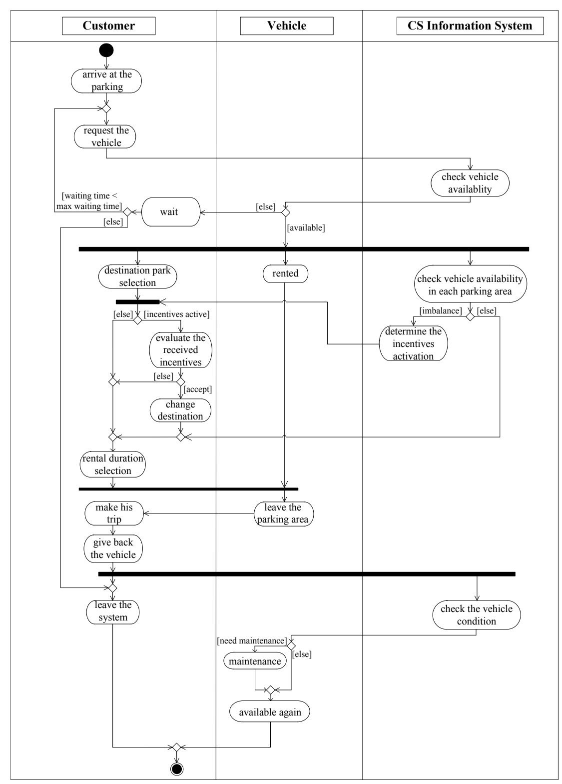


Fig. 3. The vehicle rental process activity diagram.

$$\mathcal{E}=\{a_i,r_i,p_i,d_i,m_i : i\in\mathcal{P}\} \tag{1}$$

.

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

$$\mathbf{x_i} = \begin{bmatrix} q_i \\ v_i \end{bmatrix}, \tag{2}$$

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.

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

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} \mathbf{x_2} \dots \mathbf{x_N} \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix} \tag{3}$$

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

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 cannot 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:

$$\mathcal{X} = \left\{ \mathbf{X} \mid v_i = 0, 1, \dots, V \quad q_i = 0, 1, ..., Q \quad i = 1, 2, ..., N \right\}. \tag{4}$$

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

$$\mathbf{X}^k = \mathbf{f}(\mathbf{X}^{k-1}, e^k), \tag{5}$$

where $\mathbf{X}^k = [\mathbf{x_1^k} \mathbf{x_2^k} \dots \mathbf{x_n^k}]$, 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:

$$\mathbf{x_i^k} = f_i(\mathbf{x_i^{k-1}}, e^k) = \begin{cases} [q_i^k \quad v_i^k + 1]^T & if \quad e^k = d_i \\ [q_i^k - 1 \quad v_i^k - 1]^T & if \quad e^k = p_i \\ [q_i^k - 1 \quad v_i^k]^T & if \quad e^k = r_i \\ [q_i^k + 1 \quad v_i^k]^T & if \quad e^k = a_i \\ [q_i^k \quad v_i^k - 1]^T & if \quad e^k = m_i. \end{cases} \tag{6}$$

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.

B. User-Based Relocation Strategy

The relocation strategy is specified by introducing two matrices that allow describing the availability of a customer 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 relationship between two different parking areas, i.e., their distance and reciprocal positions. The two matrices are the following.

- 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 value is determined considering the time necessary to reach the parking area j from the parking area i on foot or using the public transport, the time of the day, the day of the week and the type of customers.
- The *affinity matrix* $\mathbf{A} \in \{0, 1, \dots, N-1\}^{N \times N}$. 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 the time of the day, the day of the week, weather conditions, public transport alternatives and type of customers. Formally, $a_{ij} = 0$ ($= N-1$) means that parking area i has no affinity (maximum affinity) with respect to parking area 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. 4.
In particular, 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 threshold 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.

C. Discrete Event Simulation Module

The DES model and the UML activity diagram of Fig. 3 can be translated in a discrete event simulation model, whose dynamics depends on the interaction of the described events. Indeed, even if the CS system can be modelled by the automaton $\mathcal{A}$, the complexity of the complete CS system dynamics needs a more detailed description by simulation models. In particular, the UML activity diagrams can be easily translated in the Arena environment, a discrete-event simulation software particularly suitable for dealing with large-scale and modular systems [28], [29]. More in detail, the Arena simulation model can be implemented by applying the following three steps [26].

- 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.
- 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.
- Simulations are run and the performance indices are determined and evaluated by means of suitable statistics functions.

In order to realistically evaluate the availability of a customer 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 is selected among the incentivized parking areas instead of the original destination j. Such a probability can be determined as follows:
$$p_{select}(i|j,\mathbf{u}) = \frac{a_{ij}u_i}{\sum_{h=1}^{N} a_{hj}u_h} \qquad (7)$$
Note that $p_{select}(i|j,\mathbf{u}) = 0$ if the parking area i is not incentivized ($u_i = 0$) or if there is no 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 driver accepts the selected parking area i instead of the original destination j. Such a probability can be determined as follows, using the affinity matrix:
$$p_{available}(ij) = \frac{a_{ij}}{\max_h a_{ih}} \cdot \vartheta, \qquad (8)$$
where ϑ is the maximum value of the probability that a 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
$$p_{accept}(i|j,\mathbf{u}) = p_{select}(i|j,\mathbf{u}) \cdot p_{available}(ij) \qquad (9)$$
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 received incentive.

The performance of the system is evaluated by studying the Level Of Service (LOS), a typical index suitable for evaluating the behaviour of this kind of services [22], [14], [13], [5]. In paricular, in this case the LOS is defined as follows:

$$LOS = \frac{\text{number of served users per day}}{\text{total number of vehicle requests per day}} \qquad (10)$$

and, according to the DES model, as

$$LOS = \frac{\sum_{i=1}^{N} \nu(p_i)}{\sum_{i=1}^{N} \nu(a_i)} \qquad (11)$$

VI. Optimization Module Specification

This Section specifies the *decision* and *optimization modules* of the DSS model component. Fig. 5 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, a simulation-optimization approach is considered: in particular, the simulation module of the DSS is exploited to evaluate the LOS and an optimization algorithm is used to choose the threshold vector that maximizes it.

Different techniques are applied in the related literature for optimizing objective functions obtained by simulations [30], [31], [32], [33]. Such strategies consist of searching iteratively in the domain of decision variables variation, and this approach necessitates a connection between the optimization algorithm and the simulation model. Analysing recent trends on evolutionary optimization, several evolutionary strategies can be considered to solve this kind of problems, including PSO, genetic algorithms, differential evolution and hybrid systems [34], [35], [36], [37]. In the considered application context, we implement a PSO technique for the following reasons:

- the search space of the threshold vector has a straightforward representation using PSO particles, thus avoiding complex encoding and decoding operations;
- the combined global and local search mechanism allows fast convergence, which is necessary for the proposed DSS real-time operations;
- the evaluation of the objective function requires a limited number of simulations that are typically time consuming.

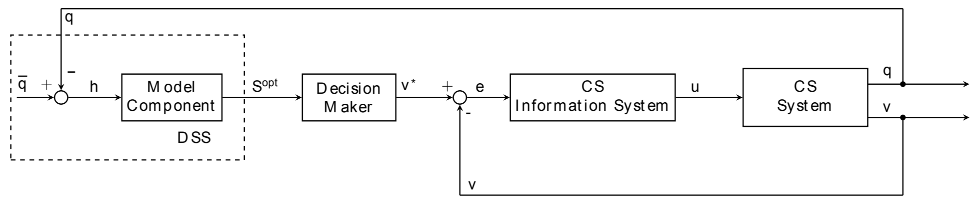


Fig. 4. Complete control scheme resulting from the introduction of the proposed DSS.

In the proposed DSS, 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 $\mathbf{S}^{opt}_{new}$ has been reached, the optimization module provides it to the decision module that suggests to the decision maker the new candidate threshold vector $\mathbf{S}^{opt}$ through the interface component.

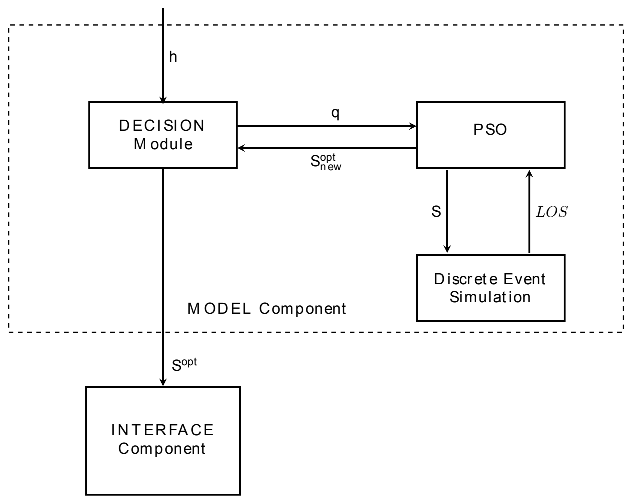


Fig. 5. Interactions among the decision, optimization and simulation modules of the DSS model component.

A. 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:

$$\mathbf{p}_j = (\mathbf{ppos}_j, \mathbf{pbest}_j, \mathbf{pvel}_j) \tag{12}$$

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.
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 $\mathbf{gbest}$.
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 $\varphi_1 = c_1 \cdot \mathbf{R}_1$ and $\varphi_2 = c_2 \cdot \mathbf{R}_2$: 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

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

Phase 4 - PSO: Stop criteria

if $LOS_{\mathbf{gbest}} \geq LOS^*$ or $numiter >= MAXITER$ **then** ▷ stop criteria
    $\mathbf{S}^{opt}_{new} = \mathbf{S}_{\mathbf{gbest}}$ ▷ optimal value of threshold $\mathbf{S}$ computed by PSO
    Return $\mathbf{S}^{opt}_{new}$
    End of the optimization-simulation process ▷ EXIT
end if

Phase 5 - PSO: Particle swarm update

for $j = 1 : K$ **do**
    $\mathbf{pvel}'_j = (\varphi_1 \cdot (\mathbf{pbest}_j - \mathbf{S}_j) + \varphi_2 \cdot (\mathbf{gbest} - \mathbf{S}_j))$ ▷ new velocity
    $\mathbf{S}'_j = \mathbf{S}_j + \mathbf{pvel}'_j$ ▷ new position
    $\mathbf{p}_j$ = ($\mathbf{S}'_j$, $\mathbf{pbest}_j$, $\mathbf{pvel}'_j$) ▷ update position and velocity of particle j
end for
Go to Phase 2

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 [7].

VII. A 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 the following the CS system and the parking areas are specified for the simulation model by determining the necessary parameters on the basis of stored city data or suitable interview analysis. Obviously, during the practical operations the used parameters and data will be obtained from the service historical data.

- **Number of parking areas**: $N = 5$.
- **Daily customer demand**. Three levels of demand characterized by different inter-arrival times are considered, in correspondence to different levels of demand during a typical day: *high* (λ_h minutes), *medium* (λ_m minutes) and *low* (λ_l minutes).
- **Routing**. The following matrix is determined by considering the proposed locations for the five parking areas:

$$\mathbf{R} = \begin{bmatrix} 0.08 & 0.20 & 0.32 & 0.10 & 0.30 \\ 0.15 & 0.05 & 0.35 & 0.25 & 0.20 \\ 0.23 & 0.23 & 0.08 & 0.23 & 0.23 \\ 0.18 & 0.25 & 0.20 & 0.02 & 0.35 \\ 0.15 & 0.30 & 0.20 & 0.30 & 0.05 \end{bmatrix} \tag{13}$$

  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. The values of elements r_{ij} are determined by considering a particular period of the year (the winter), during the rush hours, and by evaluating the walking distance and the possible public means that connect parking area i and parking area j. However, the probabilities on the principal diagonal of $\mathbf{R}$ ($i = j$) are chosen with low values in order to stress the relocation problem. Of course, during the practical operations the values of $\mathbf{R}$ will be derived from historical data.
- **Maximum waiting time**. Considering the dimensions of the town and the high frequency of service that the local public transport offers, it is assumed that, if a user cannot 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 need 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 centred 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:

$$\mathbf{A} = \begin{bmatrix} 0 & 3 & 4 & 2 & 1 \\ 2 & 0 & 1 & 3 & 4 \\ 3 & 2 & 0 & 3 & 2 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 1 & 2 & 0 \end{bmatrix} \tag{14}$$

  In particular, the elements a_{ij} of $\mathbf{A}$ are determined

considering two aspects: the distance between parking areas i and j and the possibility of using quick and reliable public transport means between the two parking areas. In particular, $a_{ij} = 4$ if the walking distance between parking area i and parking area j is smaller than 15 minutes and high frequency public transport services from j to i are available; $a_{ij} = 3$ if the walking distance between parking areas i and j is greater than 15 minutes and high frequency public transport services from j to i are available; $a_{ij} = 2$ if the walking distance between parking areas i and j is smaller than 15 minutes and low frequency public transport services from j to i are available; $a_{ij} = 1$ if the walking distance between parking areas i and j is greater than 15 minutes and low frequency public transport services from j to i are available.
- **System monitoring**. The status of the system is monitored every 10 minutes in order to determine the incentive activation status.

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 performance 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 are 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 [7].

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 users' inter-arrival times.

A. *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 λ_h, λ_m, and λ_l are considered. Each scenario is studied in two cases: deterministic and stochastic inter-arrival times. Table II 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) OC1: the incentives are not applied (LOS_{ni});

TABLE II
SCENARIOS CONSIDERED TO ASSESS THE IMPACT OF THE PROPOSED INCENTIVE MECHANISM

Scenario	Fleet size	Demand		
		λ_h	λ_m	λ_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

2) OC2: the incentives are applied with the thresholds $\mathbf{S}^{av}$ equal to the average number of vehicles available in the system (LOS_{av}),;
3) OC3: the incentives are applied with optimized thresholds $\mathbf{S}^{PSO}$ obtained by Algorithm 1 (LOS_{PSO}).

Table III reports the 5-elements vectors $\mathbf{S}^{av}$ and $\mathbf{S}^{PSO}$ for OC2 and OC3. Moreover, Table III shows the values of the LOS obtained in the three simulated 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 4% 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 OC3 the thresholds are not determined a-priori but on the basis of the customers preferences and the relative locations of the parking areas. As a consequence, the incentive for each parking area is triggered less frequently in OC3 than in OC2.

In order to enlighten this result, the following additional performance indexes are determined and compared in Table IV:

$$t^{av} = \frac{\begin{pmatrix}\text{average time during which} \\ \text{the incentives are active in OC2}\end{pmatrix}}{\text{working day duration}}, \tag{15}$$

$$t^{PSO} = \frac{\begin{pmatrix}\text{average time during which} \\ \text{the incentives are active in OC3}\end{pmatrix}}{\text{working day duration}}, \tag{16}$$

$$\delta = (1 - \frac{t^{PSO}}{t^{av}})100. \tag{17}$$

In particular, t^{av} is the average fraction of time during which the incentive mechanism is active in OC2 case; t^{PSO} is the average fraction of time during which the incentive mechanism is active in the OC3 case; finally, δ is the reduction, expressed in percentage, of the incentive activation time guaranteed by the introduction of the optimized threshold.

It is apparent that the period of activation of the incentives is significantly reduced in OC3, with clear benefits for the CS company, which obtains a better LOS while incentivizing fewer customers.

TABLE III
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		$[1\,2\,2\,2\,1]^T$

TABLE IV
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%

TABLE V
SCENARIOS WITH A FLEET SIZE OF 20 VEHICLES.

Scenario	Demand		
	λ_h	λ_m	λ_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

As Fig. 6 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 [22], [14], [13], [5], [12].

B. 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. 7 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.

C. 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 behaviour, a fleet of 20 vehicles, as in the Scenario A, is considered and the demand is gradually increased as described in Tab. V. Fig. 8 points out that the incentive mechanism is very effective if the fleet size is coherent with the demand, and, obviously, the benefit decreases if the demand increases too much.

Moreover, comparing the proposed DSS with the systems presented in the related literature [9], [14], [22], 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 performance; iii) the proposed relocation strategy is applied in closed-loop on the basis of the system state knowledge.

VIII. CONCLUSION

This paper proposes a Decision Support System (DSS) devoted to an effective Car Sharing (CS) system management: in particular, the DSS is designed for the solution of the user-based vehicle relocation problem. To this aim, the system is described in detail by Unified Modeling Language tools and a Discrete Event System model is formulated. A closed-loop control strategy is proposed in order to invite the users to drop off the vehicles in suitable parking areas through the application of an incentivation policy. The choice of the parking areas is performed by a PSO optimization procedure that optimizes the Level Of Service (LOS) obtained by a discrete event simulation.

The DSS is assessed by a case study analysis of a CS system designed for Trieste, a city in the north of Italy: the results show 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. Indeed, for typical demand levels, the LOS improvement is about 16% for a wide range of fleet sizes. The proposed incentives operate before car rental, so no special equipments are required on board of the vehicles.

Future research will focus on the evaluation of other 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

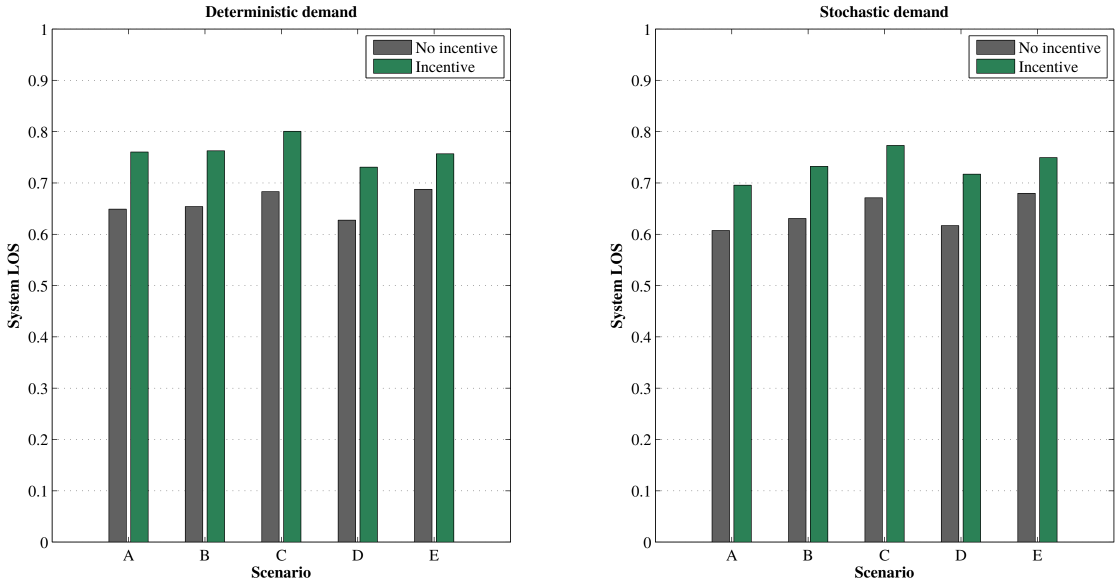


Fig. 6. System LOS before and after the application of the incentive with optimized thresholds.

into account, leading to a more complex customers decision process. Second, the determination of the optimal economic incentives on the basis of the specific considered population will be studied. Finally, the results obtained by applying genetic algorithms could be compared with the use of the PSO algorithm.

REFERENCES

[1] M. Barth and S. A. Shaheen, "Shared-use vehicle systems: Framework for classifying carsharing, station cars, and combined approaches," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1791, no. 1, pp. 105–112, 2002.

[2] M. Barth, M. Todd, and S. Shaheen, "Intelligent transportation technology elements and operational methodologies for shared-use vehicle systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1841, no. 1, pp. 99–108, 2003.

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

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

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

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

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4. IEEE, 1995, pp. 1942–1948.

[8] M. Barth and M. Todd, "Simulation model performance analysis of a multiple station shared vehicle system," *Transportation Research Part C: Emerging Technologies*, vol. 7, no. 4, pp. 237–259, 1999.

[9] A. G. Kek, R. L. Cheu, Q. Meng, and C. H. Fung, "A decision support system for vehicle relocation operations in carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, no. 1, pp. 149–158, 2009.

[10] D. Jorge, G. H. Correia, C. Barnhart *et al.*, "Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 1667–1675, 2014.

[11] B. Boyacı, K. G. Zografos, and N. Geroliminis, "An optimization framework for the development of efficient one-way car-sharing systems," *European Journal of Operational Research*, vol. 240, no. 3, pp. 718–733, 2015.

[12] M. Bruglieri, A. Colorni, and A. Luè, "The relocation problem for the one-way electric vehicle sharing," *Networks*, vol. 64, no. 4, pp. 292–305, 2014.

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

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

[15] K. Uesugi, N. Mukai, and T. Watanabe, "Optimization of vehicle assignment for car sharing system," in *Knowledge-based intelligent information and engineering systems*. Springer, 2007, pp. 1105–1111.

[16] M. Marouf, E. Pollard, and F. Nashashibi, "Automatic parallel parking and platooning to redistribute electric vehicles in a car-sharing application," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 486–491.

[17] G. H. de Almeida Correia and A. P. Antunes, "Optimization approach to depot location and trip selection in one-way carsharing systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 233–247, 2012.

[18] D. Jorge, G. Molnar, and G. H. de Almeida Correia, "Trip pricing of one-way station-based carsharing networks with zone and time of day price variations," *Transportation Research Part B: Methodological*, vol. 81, pp. 461–482, 2015.

[19] A. Kek, R. Cheu, and M. Chor, "Relocation simulation model for multiple-station shared-use vehicle systems," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1986, pp. 81–88, 2006.

[20] G. Santos and G. Correia, "A mip model to optimize real time maintenance and relocation operations in one-way carsharing systems," *Transportation Research Procedia*, vol. 10, pp. 384–392, 2015.

[21] R. Zakaria, M. Dib, L. Moalic, and A. Caminada, "Car relocation for carsharing service: comparison of cplex and greedy search," in *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on*. IEEE, 2014, pp. 51–58.

[22] M. Nourinejad and M. J. Roorda, "Carsharing operations policies: a comparison between one-way and two-way systems," *Transportation*, vol. 42, no. 3, pp. 497–518, 2015.

[23] E. L. Porteus, *Foundations of stochastic inventory theory*. Stanford University Press, 2002.

[24] G. A. Gorry and M. S. S. Morton, *A framework for management*

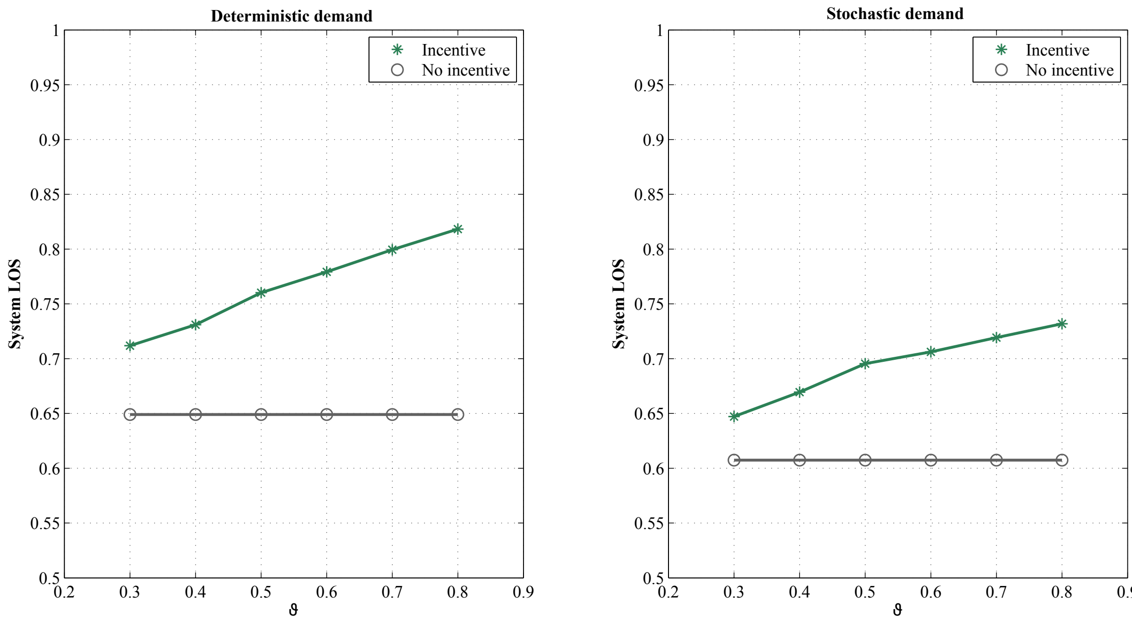


Fig. 7. Sensitivity analysis about the acceptance variation.

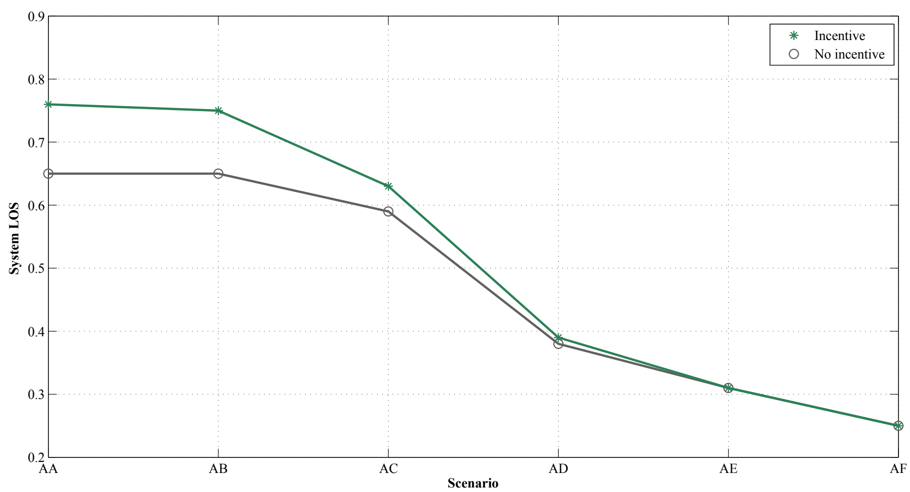


Fig. 8. Dependency of the incentive mechanism effectiveness on the coherent fleet sizing.

information systems. Massachusetts Institute of Technology, 1971, vol. 13.

[25] E. Turban, R. Sharda, D. Delen, and T. Efraim, *Decision support and business intelligence systems*. Pearson Education India, 2007.

[26] V. Boschian, M. Dotoli, M. P. Fanti, G. Iacobellis, and W. Ukovich, "A metamodeling approach to the management of intermodal transportation networks," *Automation Science and Engineering, IEEE Transactions on*, vol. 8, no. 3, pp. 457–469, 2011.

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

[28] W. D. Kelton, R. P. Sadowski, and D. A. Sadowski, *Simulation with ARENA*. McGraw-Hill New York, 2002, vol. 3.

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

[30] R. Kuo and C. Yang, "Simulation optimization using particle swarm optimization algorithm with application to assembly line design," *Applied Soft Computing*, vol. 11, no. 1, pp. 605–613, 2011.

[31] H. Pierreval and J.-L. Paris, "Distributed evolutionary algorithms for simulation optimization," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 30, no. 1, pp. 15–24, 2000.

[32] D. Rani and M. M. Moreira, "Simulation–optimization modeling: a survey and potential application in reservoir systems operation," *Water resources management*, vol. 24, no. 6, pp. 1107–1138, 2010.

[33] L. Wang, "A hybrid genetic algorithm–neural network strategy for simulation optimization," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1329–1343, 2005.

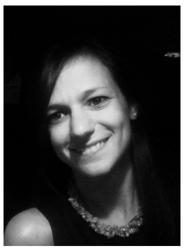
[34] V. Kachitvichyanukul, "Comparison of three evolutionary algorithms: Ga, pso, and de," *Industrial Engineering and Management Systems*, vol. 11, no. 3, pp. 215–223, 2012.

[35] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path

planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[36] A. Deb, J. S. Roy, and B. Gupta, "Performance comparison of differential evolution, particle swarm optimization and genetic algorithm in the design of circularly polarized microstrip antennas," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 8, pp. 3920–3928, 2014.

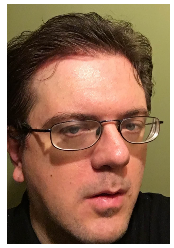
[37] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 309–313, 2015.



Monica Clemente received the Laurea degree cum laude in Management and Logistics Engineering from the University of Trieste, Italy, in 2012, and the Ph.D. degree in Information Engineering from the University of Trieste, Italy, in 2016. Her research interest focuses mainly on the application of decision support systems for the management of complex systems in transportation and logistics.

Maria Pia Fanti (M'92-SM'02-F'17) received the Laurea degree in electronic engineering from the University of Pisa, Pisa, Italy, in 1983. She was a visiting researcher at the Rensselaer Polytechnic Institute of Troy, New York, in 1999. Since 1983, she has been with the Department of Electrical and Information Engineering of the Polytechnic of Bari, Italy, where she is currently a Full Professor of system and control engineering and Chair of the Laboratory of Automation and Control. Her research interests include discrete-event systems; Petri net; consensus protocols; management and modeling of complex systems, such as production systems, logistic and healthcare systems. She has published more than 250 papers and two textbooks on these topics. Prof. Fanti was General Chair of the 2011 IEEE Conference on Automation Science and Engineering. She is Editor of the IEEE Trans. on Automation Science and Engineering and Associate Editor of the IEEE Trans. on Systems, Man, and Cybernetics: Systems. She is member at large of the Board of Governors of the IEEE SMC Society, Co-Chair of the Technical committee on Discrete Event Systems of the IEEE SMC Society, and of the Technical Committee on Automation in Logistics of the IEEE Robotics and Automation Society.

Giorgio Iacobellis received the Laurea degree in electronic engineering and the Ph.D. degree in computer science engineering from the Polytechnic of Bari, Italy, in 2004 and 2009, respectively. From February 2005 to February 2006, he was a Research Fellow at Politecnico di Bari and from September 2009 to September 2012 at the University of Trieste. Actually, Dr. Iacobellis is research fellow at the Polytechnic of Bari and his research interests include modeling, simulation, and control of discrete-event systems, Petri nets, supply chains and urban traffic networks, distribution and internal logistics, management of hazardous materials, management of drug distribution systems, and healthcare systems.



Massimilano Nolich received the Laurea degree (Honors) in Electronic Engineering, University of Trieste, Italy in 1999 and the Ph.D. degree in Information Engineering, University of Trieste, in 2003. He is research fellow at the University of Trieste, Italy. His research interests include artificial intelligence, intelligent autonomous systems, decision support systems, simulation and optimization of complex systems, operating systems, signal and data processing, embedded systems. He is an IEEE member and has published almost 80 papers in international journals and conferences. He has been involved in several European research projects: KLEAN, X-WALD, WINFC, SCOUT, CO-GISTICS.

Walter Ukovich (M'08) received the Laurea degree in electronic engineering at the University of Trieste, Italy, in 1970. From 2006 to 2010, he was Director of the Department of Electrical Engineering, Electronics and Computer Science of the University of Trieste, where he was responsible for the curriculum in management engineering. He is currently a Full Professor in operations research at the University of Trieste. His current research interests include optimization, logistics, transportation, production planning and control, and health management systems. He has authored over 210 scientific papers on these topics. Prof. Ukovich has been a member of the Evaluation Boards of the National Research Council of Italy, the University of Trieste, the University of Macerata, the Polytechnic of Turin, and the University Institute of Architecture of Venice, Venice, Italy. He has been local chair of the 2010 IEEE Workshop on Health Care Management, and local and financial chair of the 2011 IEEE Conference on Automation Science and Engineering. Currently, he is member of the scientific committee of the CIRO (Italian Operations Research Council) and Co-Chair of the Technical committee on Automation in Logistics of the IEEE Robotics and Automation Society.