SAPIENZA
UNIVERSITÀ DI ROMA

New methods for simulation-based optimization and applications to emergency department management

Dipartimento di Ingegneria Informatica, Automatica e Gestionale

Dottorato di Ricerca in Automatica, Bioingegneria e Ricerca Operativa – XXXIII Ciclo

Candidate

Tommaso Giovannelli

ID number 1552354

Thesis Advisors

Prof. Massimo Roma

Prof. Giovanni Fasano

January 2021

Thesis defended on May 19, 2021
in front of a Board of Examiners composed by:

Prof. Giuseppe Baselli, Prof. Stefano Panzieri, Prof. Fabio Tardella (chairman)

New methods for simulation-based optimization and applications to emergency department management

Ph.D. thesis. Sapienza – University of Rome

© 2021 Tommaso Giovannelli. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: tommaso.giovannelli@uniroma1.it

Abstract

The development of novel efficient algorithmic frameworks using simulation to provide solutions to real–world problems is prompted by the need to accurately represent the complex and uncertain processes of real systems, such as Emergency Departments (EDs). The resulting Simulation-Based Optimization (SBO) methodology has been receiving increasing attention in recent years, aiming to develop algorithms that do not require first-order information and support both continuous and integer variables. The trade-off between long-term goals and short-term decisions, as well as the computational cost of evaluating the black-box functions involved, determines whether to use exact Derivative-Free Optimization (DFO) algorithms, providing optimal solutions with long running time, or metaheuristic methods, returning fast solutions without optimality guarantees. Important SBO problems arise in dealing with ED management since a strong interest is shown in studying the impact of both the overcrowding phenomenon and sudden patient peak arrivals on everyday operations. To this end, further SBO approaches may be required to estimate the ED arrival rate and to recover the missing information from the real datasets in order to build Discrete Event Simulation (DES) models with a high level of reliability.

In this thesis, SBO is used with a twofold goal. On the one hand, to propose methodological contributions from an algorithmic point of view, namely a metaheuristic-based algorithm to solve a specific SBO problem and a globally convergent DFO method for mixed-integer nonsmooth constrained optimization problems, frequently arising in practice. On the other hand, to develop SBO approaches to improve the accuracy of a DES model representing an ED. In particular, an integer nonlinear black-box optimization problem is solved to determine the best piecewise constant approximation of the time-varying arrival rate function by finding the optimal partition of the 24 hours into a suitable number of nonequally spaced intervals. Black-box constraints are adopted to ensure the validity of the Nonhomogeneous Poisson process, which is commonly used in the literature to model the ED arrival process. Moreover, a model calibration procedure is proposed to estimate the incomplete information in the ED patient flow by minimizing the deviation between the real data and the simulation output. The resulting DES model is used for solving a simulation-based resource allocation problem to determine the optimal settings of the ED unit devoted to low-complexity patients. The objective is to reduce the overcrowding level without using an excessive amount of resources. Two real case studies are considered to demonstrate the effectiveness of the proposed methodology.

Acknowledgments

First, I would like to thank my supervisors, Professors Massimo Roma and Giovanni Fasano, who provided me with encouragement and patience throughout my PhD. Their rigor and expertise greatly inspired me to become the researcher I am today: very careful and aware that words and commas can make a difference in writing academic papers.

I am deeply indebted to Professor Stefano Lucidi, who was always willing to enthusiastically assist me, showing a profound belief in my abilities that boosted my self-confidence. His strong passion for doing what he loves made me aware that academia can be fun and exciting.

I would also like to extend my deepest gratitude to Professor Alberto De Santis and Dr. Mauro Messedaglia for their relentless support. Their contribution to my thesis has been invaluable and their humanity and generosity have greatly inspired me.

I am extremely grateful to Professor Luis Nunes Vicente, who welcomed me at Lehigh University like a friend and was always there during my last year of PhD, when the pandemic occurred. Not only my thesis benefited from what I learned from his rigor, innovation, and intensity in doing research, but he also offered me an invaluable opportunity to kick-start my career. Moreover, I will never forget how much I learned from the classes I took at ISE: I had never had such a productive and fruitful period in my student life.

My profound gratitude goes also to Professors Giampaolo Liuzzi and Francesco Rinaldi for allowing me to work with them on the main methodological topic of my thesis, which made me very proud and honored. I would also like to thank Professor Angel A. Juan, who kicked off my list of publications, and Ludovica Maccarrone, for her invaluable support when we traveled abroad. I enjoyed my days at Universitat Oberta de Catalunya.

I gratefully acknowledge the important collaboration with Policlinico Umberto I in Rome, which was facilitated by Professors Alberto Nastasi and Laura Palagi. To this end, I would like to extend my thanks to Prof. Ferdinando Romano, Dr. Laura De Vito, and Dr. Federico Petitti for the stimulating discussions and the precious information for developing the simulation model of the emergency department.

I would also like to thank Professors Anna Attias and Federica Ricca. My experience as a teaching assistant for their math classes made me realize that an academic career is what I want to pursue. Many thanks also to Professors Gianni Di Pillo and Massimo Maurici, Dr. Luca Paulon, and ACT Operations Research for involving me in their projects and for believing in me.

Special thanks go to my friends and colleagues at DIAG. I will never find again such a friendly environment. Having lunch together and watching movies at night were some of the activities I looked forward to.

I would like to dedicate this dissertation to my parents, who have always been there during my ups and downs, and to the burglars that stole the computer they bought for my Master's degree. This negative experience prompted me to work harder (after one month of depression) and to view my life from a different perspective. Many thanks also to my sister, one of the most important people in my life, even if I have never told her that. Her generosity and happiness bring into my life loads of joy. I hope my parents and my sister will understand one day why I spent so little time with them.

Many thanks to all my friends in Grosseto. My life would not be the same without them, even if they argue sometimes. Many thanks to all my friends in Rome and

Bethlehem too. I enjoyed the time spent together and I think that my PhD without them would have been much psychologically harder.

Finally, I would also like to thank SAPIENZA University of Rome for providing me with financial support during my life-changing experience in the United States, which would not have been possible otherwise.

Contents

Acronyms

DES . . . . . Discrete Event Simulation
DFO . . . . . Derivative-Free Optimization
DIT . . . . . Doctor-to-discharge Time
DOT . . . . . Door-to-doctor Time
DTCP . . . . . Diagnostic Therapeutic Care Path
ECDF . . . . . Empirical Cumulative Distribution Function
ED . . . . . Emergency Department
EMS . . . . . Emergency Medical Service
ECP . . . . . Emergency Care Pathway
ESI . . . . . Emergency Severity Index
GPS . . . . . Generalized Pattern Search
GRASP . . . . Greedy Randomized Adaptive Search Procedure
KPI . . . . . Key Performance Indicator
LOS . . . . . Length Of Stay
LWBS . . . . . Leave Without Being Seen
MADS . . . . . Mesh Adaptive Direct Search
MCS . . . . . Monte Carlo Simulation
MINLP . . . . Mixed-Integer Nonlinear Programming
MIU . . . . . Minor Injuries Unit
MU . . . . . Medical Unit
NHPP . . . . . Nonhomogeneous Poisson Process
OU . . . . . Orthopedic Unit
PFSP . . . . . Permutation Flow Shop Problem
RA . . . . . Resuscitation Area
R&S . . . . . Ranking and Selection
RSM . . . . . Response Surface Methodology
SA . . . . . Stochastic Approximation
SAA . . . . . Sample Average Approximation
SBO . . . . . Simulation-Based Optimization
SSU . . . . . Short Stay Unit
SU . . . . . Surgical Unit
TCTP . . . . . Time-Cost Trade-off Problem

Chapter 1

Introduction

Many real-world problems require the adoption of two powerful techniques from the field of Operations Research: *simulation* and *optimization*. The former methodology is used to represent the processes within the complex systems involved in decision problems related to real applications. The complexity of the operations performed in these systems and the uncertainty in the results of the processes prevent decision makers from using analytical models, thus requiring the adoption of simulation models. Such models allow for the assessment of the impact of changes in the current system, helping the managers to evaluate new policies and test the reliability of the processes. However, determining the optimal settings of the systems requires to embed the simulation models into an efficient algorithmic framework, thus providing a flexible and powerful tool for achieving optimal decision-making.

Important applications suited for the *Simulation-Based Optimization (SBO)* methodology emerge from several areas, ranging from the industrial field to the healthcare services. Long-term goals and short-term decisions determine the proper algorithms to use among a large number of alternatives. On the one hand, exact algorithms are adopted when problems require the best decisions regardless of the time for computation. On the other hand, metaheuristics are preferable when the requirement of optimality might be relaxed and fast decision-making is crucial.

Among the real-world problems of interest, the focus of this thesis is on the management of *Emergency Departments (EDs)*, which gives rise to important simulation-based optimization problems aimed at improving the accuracy of simulation models and providing practical solutions for enhancing the current status. The next sections illustrate the motivation underlying this work, the contents analyzed, and the contributions provided to the literature.

1.1 Motivation

The increasing availability of computational power observed in the recent years allows significant advances in the development of novel algorithmic frameworks that use simulation models to provide solutions to real-world problems. Although the number of simulation software packages provided with built-in optimization algorithms is increasing, there is still a wide margin for improvements. Indeed, most of the commonly used algorithms have been developed by adopting metaheuristic procedures, which provide fast solutions to the problems at the expense of the quality. Moreover, very often these approaches are used even though short-term decisions are not required. In these cases, using exact algorithms might be a better choice since solutions with optimality guarantees allow achieving improved results, even

if with longer running time. Moreover, when runs of a simulation are expensive (e.g., when designing new manufacturing systems), using algorithms requiring many objective function evaluations may be prohibitive, thus encouraging research on efficient optimization of black-box functions. Contrarily, when systems with dynamic features are involved (e.g., transportation requests of users of car-sharing services) and simulation is cheap to perform, metaheuristics are an appropriate choice, since they guarantee a satisfactory compromise between quality of the solutions and running time.

In many SBO problems arising from real-world applications, simulation may need to receive as input both continuous and integer values. Frequently, the integrality of the latter values is unrelaxable (i.e., number of beds and nurses in a hospital ward), giving rise to problems having integer variables that cannot be treated as continuous variables. Moreover, the objective and constraint functions, which are evaluated through simulation, may require long runs with many replications, thus resulting in computationally expensive black-box functions. These reasons prevent the use of derivative-based approaches based on continuous relaxation, which is the strategy adopted by many optimization methods for solving nonlinear problems with both continuous and integer variables when functions are not of the black-box type. The resulting optimization problems are called *Mixed-Integer Nonlinear Programming (MINLP)* problems and they represent one of the most active areas of research.

Several examples of real-world applications can benefit from the availability of effective algorithmic frameworks able to efficiently handle simulation. For instance, the management of the EDs is one of the areas where SBO appears to be fundamental, since complex processes and random events may prevent the use of approaches relying on analytical tractability. In particular, the great number of patients arriving at the ED leads very frequently to overcrowding, which is a worldwide phenomenon well perceived by the ED stakeholders: long patients waiting times before the medical examination, excessive number of patients in the ED, and high percentage of patients who leave without being seen are clear indications of such a problem. In order to analyze (and to possibly prevent) the overcrowding phenomenon, it is necessary to detect the time spent by the patient inside the ED during the different phases of the whole process. To this end, novel Italian guidelines recommend monitoring the time required by each clinical pathway in relation to the urgency of the patients. In light of these guidelines, a great interest is shown by the ED managers for tools that enable them to perform scenario analysis, like those provided by simulation models. The aim is to assess how the main Key Performance Indicators (KPIs) change after possible redesigning of the ED patient flows and changing of the model of care. A step forward is represent by SBO, which allows determining the optimal settings of the ED that correspond to the minimum level of overcrowding.

A second great and increasing interest in the ED management is to use the simulation modeling for studying the impact of patient arrival surges caused by some disaster. The related Italian guidelines state specific measures to be adopted in order to efficiently tackle such situations. In particular, the so called "Internal Emergency Plan for Massive Inflow of Injured" (in Italian: PEIMAF, Piano di Emergenza Interno per il Massiccio Afflusso di Feriti) has been issued in recent years. In this plan, different critical levels have been provided, and suited operative measures are indicated to reallocate ED human and physical resources, whenever it is activated due to critical events. Moreover, low complexity patients can be addressed to outpatient facilities, to enable the ED staff to timely deliver the most urgent treatments. The application of this emergency plan occurred in Genoa (Italy) on August 14, 2018, when Morandi's Polcevera viaduct collapsed, causing 43 deaths and many injuries and, more recently, in many Italian EDs for the maxi-emergency

due to COVID-19 pandemic.

Every SBO approach adopted for studying the two problems of interest mentioned in the previous paragraphs requires an accurate simulation model. In particular, two factors have the most significant impact on the accuracy. The external factor concerns the patient arrival process; the internal factor relates to the patient flow within the ED. Therefore, both aspects must be properly considered for a reliable study on ED systems. As regards the external factor, every modeling methodology is generally based on assumptions that, in some cases, may represent serious limitations when applied to complex real–world processes, such as ED operations. In particular, when dealing with stochastic modeling of the ED patient arrival, due to the nonstationarity of the process, a standard assumption is the use of a *Nonhomogeneous Poisson Process (NHPP)*. Proper approaches are required to check whether this assumption is in accordance with the representation of the arrival process used in a simulation model. As concerns the internal factor, the reliability of the simulation study is strongly affected by the problem of missing data, which consists in the unavailability of data related to some of the starting and ending times of the activities performed in the ED. This well-known issue is responsible for the lack of knowledge of the service time of some processes, which is required to estimate the corresponding probability distributions to use in the simulation model.

1.2 Overview and contributions

This thesis aims to propose both methodological and practical contributions to the fields of SBO and ED management. In particular, Chapter 2 provides the basic concepts related to *Discrete Event Simulation (DES)* and SBO to allow the reader to better understand all the topics discussed throughout the thesis. Moreover, the main works from the literature are reviewed, focusing on *Derivative-Free Optimization (DFO)* methods, which are an important class of SBO, and four ED applications: ED overcrowding, management of the ED under disaster conditions, ED arrival process, and calibration of ED simulation models. These applications are tackled throughout this thesis by using DES and formulating SBO problems. When dealing with SBO problems, DFO methods are often adopted, since first-order information is unavailable due to the lack of analytical models. Overcrowding and disaster conditions are two of the most studied ED applications since they have a strong impact on the everyday operations within EDs. The former has been widely considered in the specific literature, while the latter has received less attention. Instead, ED arrival process and model calibration are topics with a methodological purpose which can be useful to significantly improve the accuracy of simulation models adopted for practical goals.

Two algorithmic frameworks that can be used to solve SBO problems are discussed in Chapters 3–4. In particular, Chapter 3 proposes a SBO approach, called simheuristic, based on the hybridization of simulation with a metaheuristic. This methodology is applied to an integrated resource allocation and scheduling problem, which provides the opportunity to demonstrate how simulation can be effectively integrated with optimization strategies when simulation does not require long runs and function evaluations are computationally cheap. This allows determining approximate solutions in short time. Instead, Chapter 4 describes global convergent linesearch-based methods for mixed-integer nonsmooth constrained optimization problems, which are frequently involved in real-world applications. First-order information on the problem functions is assumed to be unavailable and the black-box functions involved are supposed to be computationally expensive. First, four algo-

rithms with different computational cost and convergence properties are proposed for mixed-integer bound constrained problems. Then, an exact penalty approach is used to tackle the presence of nonlinear (possibly nonsmooth) constraints. The global convergence properties toward stationary points are analyzed for all the proposed algorithms.

Chapter 5 focuses on two important SBO problems devoted to improving the accuracy of every DES model representing an ED. In particular, the first approach aims to estimate the unknown arrival rate of the NHPP arrival process by using a novel modeling methodology, based on a piecewise constant approximation of the arrival rate accomplished with nonequally spaced intervals. To this end, an integer SBO is used. Black-box constraints are adopted both to ensure the validity of the NHPP assumption, which is commonly adopted in the literature, and to prevent mixing overdispersed data for model estimation. The second approach aims to estimate the incomplete data to be used for building the DES model by adopting a model calibration procedure. In the proposed SBO problem, the objective function represents the deviation between simulation output and real data, while the constraints ensure that the response of the simulation is sufficiently accurate according to the required precision.

In Chapter 6, the effects of patient peak arrivals caused by the occurrence of critical events are studied for a medium–size ED located in a region of Central Italy recently hit by a severe earthquake. In particular, a DES model is proposed to analyze the patient flow through this ED, aiming to simulate unusual operational conditions due to a critical event, like a natural disaster, that causes a sudden spike in the number of patient arrivals. The availability of detailed data concerning the ED processes enables building an accurate DES model (without requiring a model calibration) and performing extensive scenario analyses.

The second case study, which is described in Chapter 7, concerns the ED of a large hospital in Rome, Italy. By using the data collected from the patient flow through the ED, this case study is adopted to test the effectiveness of the two approaches proposed in Chapter 5. Through the accurate simulation model resulting from the application of these two approaches, SBO is used for solving a resource allocation problem related to the specific case study considered. The goal is to determine the optimal settings of the ED unit devoted to the medical visit of low-complexity patients in order to reduce the overcrowding level. A multiobjective formulation of the problem is adopted to find a trade-off between the conflicting goals of reducing the management cost and guaranteeing patients timely treatments according to their urgency code.

Finally, Chapter 8 reports the conclusions and provides insight into future work.

1.3 Generalities on emergency departments

Every ED consists of two categories of stakeholders: *care providers*, such as physicians and nurses, and *patients*, who need a specific care. After arriving at the ED, each patient goes through different clinical paths. This flow comprises several steps, which generally consist in the triage, whose aim is to assign an urgency code to every patient, medical visits, examinations, reassessments and, finally, the leaving of the ED, with diverse kinds of discharge.

As a first step, a triage tag is assigned to every incoming patient, in order to determine the priority of treatment. Different systems of classification are usually adopted. The most commonly used scale in Italy is reported in Table 1.1. Moreover, in some regions a blue tag is also used as intermediate case between green and yellow

Table 1.1. Color coding scheme for triage of incoming patients.

Red tag	Very critical, danger of life. The patient must be visited immediately.
Yellow tag	Fairly critical, high risk. The patient should be visited as soon as possible.
Green tag	Minor injury, no risk of conditions worsening. The treatment can be delayed.
White tag	No injury, minimal pain with no risk features. The treatment can be deferred.

tags. In some countries, more fine-grained (sometimes numerical-valued) scales are adopted. In order to guarantee more appropriate clinical paths and following the main current international scientific evidence, the Italian Ministry of Health is going to adopt new guidelines based on a triage composed by five numerical urgency codes ([209, 183]), as detailed in Table 1.2. This classification closely resembles

Table 1.2. Numeric coding scheme for the triage of incoming patients.

Code 1	Very critical, immediate treatment.
Code 2	Fairly critical, high level of risk.
Code 3	Not very critical, no risk of worsening.
Code 4	Not critical, acute but not serious.
Code 5	Not critical, not serious, not acute.

the Emergency Severity Index (ESI) adopted in the US, which is based on an algorithm that rapidly yields grouping of patients into five classes, as described in [88]. Table 1.3 compares the old and new triage scales and reports the maximum waiting times recommended by the new guidelines for each triage code.

Table 1.3. Comparison between the old and new triage scales and maximum waiting times recommended by the new guidelines for each triage code.

FORMER TRIAGE SCALE	NEW TRIAGE SCALE	MAX WAITING TIME
Red tag	Code 1	0 min
Yellow tag	Code 2	15 min
Green tag	Code 3	1 h
	Code 4	2 h
White tag	Code 5	4 h

After the triage code is assigned, the more appropriate Diagnostic Therapeutic Care Path (DTCP) is activated. In particular, a patient can be usually sent: 1) to an ED room, 2) to outpatient facilities, 3) toward a *"Fast Track"*, 4) to the *"See and Treat"* service. The Fast Track and See and Treat are services that allow the ED to reduce the waiting times, the Length Of Stay (LOS) in the ED, as well as the percentage of patients who Leave Without Being Seen (LWBS). The patients directed to the ED rooms follow different clinical pathways, which include medical examination and diagnostic tests up to the definition of the outcome. The patient flow inside the ED rooms is very complex due to the many and different specific

needs (often even difficult to identify in short time) and the high variability of medical conditions of the incoming patients. Moreover, the flow is also strongly affected by the availability of the resources, such as staff on duty, number of rooms, machineries dedicated to different services, capacity of holding areas, and beds for hospitalization.

The ED process is usually characterized by the following outcomes: *discharged home* with reliance, if necessary, on territorial structures, which provide control at outpatient facilities; *hospitalization* at an hospital ward (if a bed is available) or *transfer* to another hospital; admission to the *Short Stay Unit (SSU)* (whenever such a unit exists). The SSU is an inpatient unit attached to the ED, managed under the clinical governance of the ED staff, designed for the short term treatment, observation, assessment, and re-evaluation of patients. When a patient is discharged at the end of the clinical pathway, a physician may assign an exit code corresponding to the outcome.

Chapter 2

Simulation-based optimization and emergency department applications: literature review

After recalling the basic concepts of discrete event simulation and simulation-based optimization, this chapter reviews the main works from the literature dedicated to derivative-free optimization and emergency department applications.

2.1 Discrete event simulation

Simulation is one of the most widely used tools in the fields of operations research and management science [161]. When real-world systems involve decision problems, which require to determine good solutions among several alternatives, powerful techniques, such as simulation, are often needed to reproduce the underlying complex processes and random occurrences. Indeed, such systems may be too complex to be studied through analytical representations, thus requiring to use a computer for building a simulation model able to represent all the system components necessary for a proper decision-making. Building such models requires also to evaluate the extent to which the simulation can be considered a reliable representation of the real system. Once the model is deemed as correct and accurate, it offers a cheap and easy tool to perform scenario analyses, which allow decision makers to assess new policies without incurring the high cost that a practical implementation causes if results are proved ineffective. However, there are several drawbacks to be faced when dealing with a simulation study: achieving the objectives of the analysis may be compromised by the quality of input data, which affects also the accomplishment of the right level of accuracy whenever it is not available in large amount; obtaining reliable results when complex systems are involved sometimes requires high computational burden and long running time; a simulation study allows the decision makers only to assess how the system responses may be affected when changes are introduced to the current status, while to determine the optimal settings it is necessary to integrate the simulation model with an optimization algorithm, as described in Section 2.2.

Before providing a formal definition of *Discrete Event Simulation (DES)*, some preliminary concepts need to be introduced. The state of a system can be defined as a collection of variables describing the system at a specific time. Based on the properties of the state, it is possible to distinguish among discrete and continuous systems. In a discrete system the state variables change only at discrete points (which are associated with events) over the time. By contrast, in a continuous system the

state variables change continuously across the time. The same classification applies also to distinguish among continuous and discrete simulation models. However, it is important to point out that a continuous system could be represented through a discrete simulation and vice versa, depending on the objectives of the study. Two further ways of classifying simulation models can be considered. In particular, a simulation is called static if it represents a system at a particular time, otherwise it is called dynamic. Moreover, simulation models can be deterministic, when random components are not included, or stochastic, when different simulation outputs are obtained, even if the inputs are the same, due to the presence of random variables. That said, a DES model can be defined as a simulation model which is discrete, dynamic, and stochastic (deterministic models are included as a special case of stochastic models).

After describing the main features of DES models, it is possible to formally define all the components that allow for the mathematical tractability required when such models are embedded in a simulation-based optimization scheme, which is the framework of this thesis. Every simulation model is associated with input and output data. The former affects both the structural components, which relate to the flow followed by the entities within the model, and the quantitative components, which are the parameters used in the simulation. Such parameters may have a direct interpretation in the simulation (e.g., number of workers, number of resources, and so forth) or may be parameters of the probability distributions used in the model to account for random events. In a simulation study, input data analysis is the stage that aims to explore the data collected from the real system in order to both identify the probability distributions underlying the stochastic processes and determine their parameters through goodness of fit tests. Instead, output data analysis refers to the study of the values returned by a simulation, which can be considered as measures of the system performance corresponding to a given input. The choice of the measures to be considered as KPIs depends on the objectives of the analysis and it is crucial to derive the proper conclusions on the impact on the system of changes in the current settings.

Denoted as $x \in \mathbb{R}^n$ the vector of the quantitative input parameters of a simulation model[1], let $Y(x, \xi)$ be the simulation output associated with x, where $\xi \in \mathbb{R}^p$ is a random vector[2] introduced to represent the uncertainty. In particular, the components of ξ are the random variables used within the simulation model. Since $Y : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^d$ is assumed to be a deterministic function (with $d \geq 1$), ξ is the only source of randomness and one may consider a realization ξ_i of this random vector as the i–th element of a set $\{\xi_1, \ldots, \xi_m\}$ of samples which represent independent and identically distributed (i.i.d.) random vectors in $\mathbb{R}^p$. Therefore, given x, Y can be considered as a random vector as well, and a realization $Y(x, \xi_i)$ is associated with each ξ_i in the set of samples described above. The set $\{Y(x, \xi_1), \ldots, Y(x, \xi_m)\}$ contains the simulation outputs obtained by m independent replications (runs) of the simulation. Since the mean of the simulation output $\mathbb{E}[Y(x, \xi)]$ is unknown due to the lack of knowledge about the true probability distribution of $Y(x, \xi)$, an estimator is required. A commonly used unbiased estimator is the sample mean, i.e., $\bar{Y}_m(x) = \sum_{i=1}^{m} Y(x, \xi_i)$, which is sometimes referred to as the simulation output itself.

[1] For the sake of simplicity, x is assumed to be a vector of continuous variables, but it is important to point out that x might actually have all discrete variables or a mix of both continuous and discrete variables. Technically, x is sometimes referred to as a scenario.

[2] More formally, $\xi : \Omega \to \mathbb{R}^p$ is a random vector with associated probability space $(\Omega, \mathcal{F}, P)$.

2.2 Simulation-based optimization

The expression *simulation-based optimization* refers to the branch of optimization that deals with the combination of an optimization algorithm with a simulation[3], whether stochastic or deterministic. When the simulation is stochastic (e.g., a DES model), another denomination for this branch is *simulation optimization* (see, e.g., [82] and [11]). From a thorough search of the relevant literature, the latter expression does not seem to be used with deterministic settings. However, there is not the same agreement on the use of simulation-based optimization, since this denomination appears when a deterministic (see, e.g., [66] and [157]) as well as a stochastic simulation (see, e.g., [94] and [63]) is adopted. Throughout this thesis, it is assumed that Simulation-Based Optimization (SBO) is a general expression to be used in both cases, whether a deterministic or a stochastic simulation is involved.

Assuming that a stochastic simulation is used, the general SBO problem is formulated as follows

$$\begin{aligned} \min\ & f(x) \\ s.t.\ & \mathbb{E}[g(x,\xi)] \leq 0, \\ & h(x) \leq 0, \\ & l \leq x \leq u, \\ & x_i \in \mathbb{R} \text{ for all } i \in I^c, \\ & x_i \in \mathbb{Z} \text{ for all } i \in I^z, \end{aligned}$$

where I^c is the index set of the continuous variables, I^z is the index set of the integer variables[4], f is the objective function, $\mathbb{E}$ is the expected value operator, g is the vector–valued function related to the stochastic constraints, h is the vector–valued function related to the deterministic constraints, and ξ is the random vector introduced in the formulation to represent the randomness. Denoted $Y(x,\xi)$ as the simulation output, there may be several alternatives for the objective and constraint functions, such as

$$f(x) = \begin{cases} \mathbb{E}[F(x,Y(x,\xi))] \\ q_\alpha(x) \\ P\{F(x,Y(x,\xi)) \geq 0\}, \end{cases} \tag{2.2.1}$$

$$g(x,\xi) = \begin{cases} G(x,Y(x,\xi)) \\ \mathbb{1}_{[0,+\infty)}[G(x,Y(x,\xi))] - \alpha, \end{cases} \tag{2.2.2}$$

where $F(x,Y(x,\xi))$ and $G(x,Y(x,\xi))$ are two arbitrary functions depending on the simulation output $Y(x,\xi)$, $\mathbb{1}_{[0,+\infty)}$ is the indicator function of the interval $[0,+\infty)$ (it is 1 if its argument is in $[0,+\infty)$, 0 otherwise), α is a scalar such that $0 \leq \alpha \leq 1$, and q_α is the *α-quantile*, defined as $q_\alpha(x) = \sup\{y : P(Y(x,\xi) \leq y) \leq \alpha\}$. It is worth noting that F, G, and Y may be either scalar–valued or vector–valued functions.

[3] Apart from DES, another type of simulation commonly used in practice is Monte Carlo Simulation (MCS), which is used to repeatedly sample from a probability distribution in order to estimate its unknown parameters. Differently from DES, which is often associated with computationally expensive simulations, MCS is usually cheap to perform.

[4] Although categorical variables should be included in this general formulation as well, they are here ignored for the sake of simplicity.

Moreover, the first case in (2.2.2) leads to an expected–value constraint, while the second case corresponds to a probabilistic (or chance) constraint, which follows from

$$\mathbb{E}\{\mathbb{1}_{[0,+\infty)}[G(x,Y(x,\xi))] - \alpha\} = P\{G(x,Y(x,\xi)) \geq 0\} - \alpha.$$

Note that although the setting of SBO is the same as stochastic optimization[5], in the former the methodology used for estimating the uncertain elements is based on simulation.

Strong relationships connect SBO to other branches of operations research. In particular, SBO shares with mathematical programming the same structure used to formulate a problem but, by contrast, the analytical expressions of the functions are not known and, accordingly, derivatives are unavailable. Moreover, all the techniques based on the availability of an algebraic model, such as the dual problem formulation, cannot be applied. Similarly to derivative-free optimization, SBO optimization aims to tackle problems where functions are represented by black-box models. However, although most of the algorithms from these two fields are similar, SBO algorithms may be designed to take into account also the stochastic nature of the problem, while derivative-free algorithms are traditionally applied to deterministic contexts (or when functions are subject to a moderate noise). Furthermore, in SBO the functions may not be computationally expensive, although most of the functions considered in the problems addressed in this thesis are to include in this category. Finally, other two fields related to SBO are machine learning and statistics. The former may be used to approximate the relation between input and output of a simulation model, thus allowing the algorithms to be faster. The latter provides an important contribution due to the possible roles that hypothesis testing, confidence intervals, and estimation of probability distribution may play in SBO algorithms.

2.2.1 Optimization methods

The different optimization methods significantly vary according to the type of variables considered in each problem, whether discrete or continuous. In discrete SBO problems, the feasible region is composed of points over a discrete domain. Hence, variables may be integer, binary or categorical (i.e., variables representing elements of a certain category, among which there is not any order relation). A possible formulation for the integer SBO problem is given below

$$\begin{aligned} &\min \ f(x) \\ &s.t. \ x \in \mathcal{F} \cap \mathbb{Z}^n, \end{aligned}$$

where $f(x) = \mathbb{E}[Y(x,\xi)]$ and $\mathcal{F} \subset \mathbb{R}^n$. If $\mathcal{F}$ is a compact set, the feasible region $\mathcal{F} \subset \mathbb{R}^n$ contains a finite number of points. Contrarily, in continuous SBO problems, the feasible region is composed of points over a continuous domain. A possible formulation for these types of problems is reported below

$$\begin{aligned} &\min \ f(x) \\ &s.t. \ x \in \mathcal{F}, \end{aligned}$$

where $f(x) = \mathbb{E}[Y(x,\xi)]$ and $\mathcal{F} \subset \mathbb{R}^n$. In both cases, some assumptions are introduced to make the problems well–defined and suitable to be solved by optimization algorithms. In particular, $Var[Y(x,\xi)]$ is supposed to be finite and an estimator

[5] Another expression used to refer to SBO is *computational stochastic optimization*.

$\hat{f}(x)$ of $f(x)$ such that $\hat{f}(x)$ converges with probability 1 to $f(x)$, i.e., $\hat{f}(x)$ is a consistent estimator, is supposed to be known.

One general methodology that can be applied to both discrete and continuous SBO problems is the *Sample Average Approximation (SAA)* approach, which allows solving a stochastic optimization problem through algorithms developed for deterministic problems. Alternative names used to refer to SAA are sample path optimization, stochastic counterpart, and Monte Carlo sampling approach. The application of this methodology requires ξ to be a random variable whose probability distribution does not depend on x and $\mathbb{E}[Y(x,\xi)]$ to take on a finite value at each x in the feasible region. Given a sample of i.i.d. random variables ξ_i, each one associated with a replication $i \in \{1,\ldots,m\}$ of the simulation, the sample mean $\bar{Y}_m(x) = \sum_{i=1}^{m} Y(x,\xi_i)$ is adopted as an estimator of $f(x)$. Under the hypotheses mentioned above and assuming that the random variables $Y(x,\xi_i)$ are i.i.d., the strong law of large numbers holds, implying that $\bar{Y}_m(x) \to \mathbb{E}[Y(x,\xi)]$ with probability 1 as $m \to \infty$. It is important to point out that the sample mean, which is the objective function of the approximate problem, is still a random variable. However, given a sample of realizations of the m random variables ξ_i, the sample mean is now a deterministic function of x and its value can be evaluated at any point of the feasible region. Therefore, after fixing the values of ξ_i, with $i \in \{1,\ldots,m\}$, and applying an optimization algorithm to solve the approximate problem, it can be obtained a candidate solution, depending on the values of the sample considered, which can be considered as an estimator of the optimal solution of the original problem. Solving the approximate problem repeatedly by varying the values of the sample leads to several candidate solutions, among which the optimal solution is considered as the candidate point that is associated with the lowest value of the sample mean. Due to the wide range of problems where SAA can be applied, in the literature many papers deal with this approach (see, e.g., [46, 107, 212, 227, 64, 197, 137]).

Other general approaches whose application is independent of the types of variables involved are random search and direct search methods. Random search algorithms (see [12] for a review) generate each iterate by sampling from a probability distribution defined over the feasible region and adjusted across the iterations. Such algorithms are divided into locally and globally convergent. Unlike the meaning used in nonlinear optimization, in the context of SBO the adjective *globally* means that the algorithm aims to find the global optimum point. Instead, in nonlinear optimization the adjective *globally* refers to algorithms that converge to a local optimizer regardless of the initial point. Therefore, in SBO, global random search methods need to visit the whole feasible region to guarantee the global convergence property. Also metaheuristics are included among the random search methods (see [154] for a review). Examples of metaheuristics are genetic algorithms [204, 233], simulated annealing [143, 35], tabu search [89], and scatter search [90]. In contrast with random search methods, direct search methods have been traditionally developed for deterministic settings and they have been derived from the field of derivative-free optimization. When applied to solve stochastic SBO problems, a sampling scheme is required to control the noise. Direct search methods used in derivative-free optimization are reviewed in Section 2.3.

After describing general methodologies, now we focus on specialized approaches for discrete and continuous SBO problems. As regards discrete problems, the nature of the optimization methods varies according to the size of the feasible region, which often is also called parameter space (see, e.g., [11]). In particular, two classes can be identified: methods for finite feasible regions, such as ranking and selection procedures, and methods for large or (countably) infinite feasible

regions, such as ordinal optimization, random search algorithms, and direct search methods. Although defining optimality conditions is a challenging task in both cases, an optimality guarantee is fundamental to ensure the correctness of the algorithm and to derive implementable stopping rules. Three reasons give rise to the difficulty in establishing optimality conditions: (i) the presence of random noise in the objective function $f(x)$, which needs to be properly estimated, for example through the sample mean $\bar{Y}_m(x)$; (ii) the unavailability of analytical expressions for $Y(x;\xi)$ and, as a consequence, for $f(x)$; (iii) the possible large number of feasible solutions in the feasible set. Accordingly, since feasible solutions cannot be ranked with 100% confidence due to (i), certifying the optimality of a solution requires an infinite amount of computational budget. Moreover, although finding the optimum requires to evaluate all the points in the feasible region due to (ii), (iii) prevents the algorithms from performing a complete enumeration of all the feasible solutions because of the high computational cost it would require.

When the feasible set is finite, the most popular methods are *Ranking and Selection (R&S)* procedures (see, e.g, [33, 138, 211]), which aim to find the best solution among a finite number of alternatives, which are the feasible solutions. Since this number is small, performing a simulation at each feasible solution is not an expensive task. One formulation of the R&S problem is based on the selection of a strategy to allocate the computational budget among the alternatives. One can claim the following general rules: adding replications to a feasible solution x decreases the variance of the estimator $\hat{f}(x)$ of the objective function; using the same number of replications m for each alternative is not efficient; most of the computing budget should be assigned to the critical solutions. This is shown in Figure (2.1), where an example related to a simple discrete optimization problem is illustrated. In this example, the feasible region is a set composed of five feasible solutions, hence $\{x^1, \ldots, x^5\}$. The figure shows the sample means of the simulation outputs $\bar{Y}(x)$ corresponding to each alternative x^i along with their confidence intervals. It can be observed that if the objective function is estimated through the sample mean, in the plot on the left–hand side the best solution is x^2, which is the point with the lowest sample mean. However, the lower boundary of its confidence interval is under $\bar{Y}(x_2)$, so it is not straightforward which feasible point is actually the best one. In fact, it might happen that adding more replications to the first solution leads to reducing $\bar{Y}(x_1)$, as shown in the plot on the right–hand side. This strategy of selecting the best point is used in the optimal computing budget allocation procedures (see, e.g., [47] and [45]), which belong to the class of Bayesian R&S. The other category of R&S is constituted by the frequentist procedures, which are based on the indifference–zone formulation. Here the idea is to select as the optimal solution $x^\star$ a feasible point whose true objective function is better than the alternatives by a parameter $\delta > 0$, guaranteeing a probability of correct selection (i.e., probability of selecting the true optimal solution) of $1 - \alpha$, where $\alpha \in [0, 1]$ is a positive scalar. Hence, every feasible point y such that $f(y) \leq f(x^\star) + \delta$ belongs to the indifference zone and it is considered as an optimal solution as well.

When the feasible set is large or (countably) infinite, in addition to random search and direct search methods, another effective approach is ordinal optimization (see, e.g., [115, 160, 116]). This technique is based on two principles: estimating the order among the objective function values evaluated at different points is easier than estimating the absolute value of the objective function at those points; searching for good enough solutions instead of the best ones leads to softening the optimization goal and, at the same time, to exponentially reducing the computational cost. The first idea allows the algorithm to direct the computational effort into achieving

enough accuracy so that the estimated objective function values of the considered points are in the same order as the true values. Thanks to the knowledge of this order, it is possible to avoid computing accurate estimates for those points having a large objective function value. The second idea enables the algorithm to efficiently achieve an accurate order as the number of solutions considered good enough increases, i.e., the optimization goal is softened since the optimal solution is not the only target of the procedure. An important role in this methodology is played by the alignment probability, which is the probability that a subset of feasible solutions contains satisfactory solutions. To apply this methodology, one has first to fix a given alignment probability, then select a subset from the feasible region and, finally, apply a R&S procedure to determine a good solution in this subset. Since the dimension of the subset is usually considerably smaller than the feasible region, this procedure turns out to be both effective and efficient.

For continuous SBO problems, in addition to the random search and direct search methods previously described, popular approaches are *Response Surface Methodology (RSM)* and *Stochastic Approximation (SA)*. RSM is also referred to as metamodel–based optimization, since these methods aim to approximate the unknown relation between simulation input and output through a surrogate model, which can be used to effectively determine the next points to evaluate (see, e.g., [30, 147]). Two different types of methods are considered based on where the emphasis is put, whether on exploitation or exploration. Classic RSM falls into the first case, which concerns methods that aim to find good points in a local area that is sequentially updated. In particular, at each iteration the objective function is evaluated by performing a simulation at points selected through design of experiments techniques. A first-order polynomial model is then built by fitting the objective function values at the sampled points, allowing the method to use the steepest descent direction of the model to determine the next local area for a new round of exploitation. When the first-order model fails to guarantee a good fitting error of the points, a second-order polynomial model is used to check if the point associated with its minimum provides a good estimate of the optimal solution of the original SBO problem. To this end, optimality conditions are checked through statistical hypothesis testing. Although RSM is considered as a sequential heuristic, in the last years trust region methods have been adopted in this framework, allowing researchers to deal with convergence properties. For instance, papers that have investigated trust region methods are [64] and [44]. When the emphasis of the algorithms is on exploration, surrogate models are built over the entire feasible region in order to identify areas where good points are likely to be. Kriging models (see, e.g., [224, 120, 146, 145]) are among the most popular surrogate models adopted by the methods falling in this class, which is often referred to as Bayesian global optimization.

SA methods are a natural adaptation of the steepest descent method in nonlinear optimization (see [217] for a survey). Since in SBO derivatives are unavailable, first-order information is based on an estimate $\hat{\nabla} f$ of the gradient of the objective function ∇f. Hence, the k–th iteration of these algorithms is defined by the update $x_{k+1} = x_k - \alpha \hat{\nabla} f(x)$, where α is a positive stepsize. SA dates back to the works by Robbins and Monro [206] and Kiefer and Wolfowitz [136]. While the former authors use an unbiased estimator of the true gradient, the estimator used by the latter authors is asymptotically unbiased. In particular, two types of gradient estimators are adopted: direct estimators (see, e.g., [241]), which compute an estimate by using knowledge of the underlying simulation model, such as input probability distribution; indirect estimators, which treat the simulation model as a black box and compute an estimate only by using function values. Algorithms that use indirect gradient

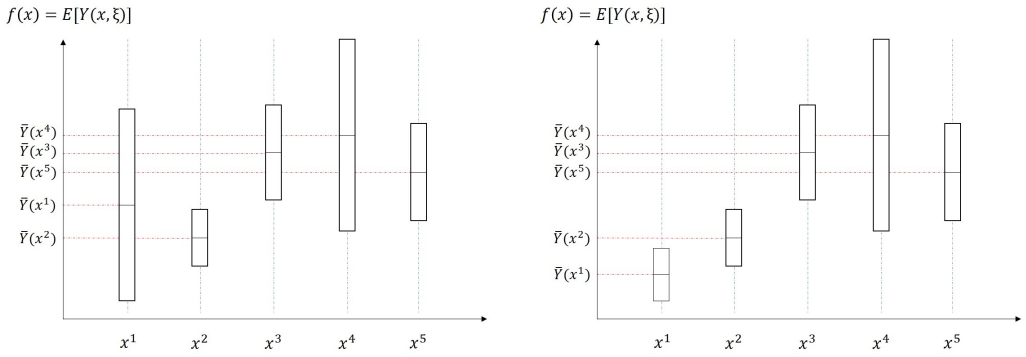


Figure 2.1. This figure shows an example of discrete SBO problem with 5 feasible solutions. In both plots, the sample mean of each simulation output is indicated with its confidence interval. The plots shows that adding more replications to the first feasible solution reduces its confidence interval, leading to a variation in the ranking of solutions.

estimators are also called gradient–free or stochastic zeroth–order methods. Two examples of indirect estimators are finite differences and simultaneous perturbation [217, Chapter 7]. While finite difference estimators require to sample at least $n + 1$ points, simultaneous perturbation requires only two points, regardless of the problem size, without worsening the convergence properties. It is important to point out that both SA and SAA have their analogues in the machine learning community, where they are referred to as stochastic and batch approaches, respectively (see [36]).

2.3 Derivative-free optimization

In *Derivative-Free Optimization (DFO)* the objective and constraint functions of the considered problems are assumed to be of black-box type[6]: the analytical expression is unknown, and the function value corresponding to a given point is the only available information. This implies that the first-order characterization provided by the derivatives is not available, leading to challenging optimization problems. In particular, several contexts give rise to the derivative unavailability that requires the adoption of a DFO algorithm: derivatives may be completely unavailable, as in case of legacy codes; derivatives may be available but too expensive to be computed; functions evaluation may be expensive, making impossible to approximate the derivatives; functions may be subject to noise, causing the first-order information to be unreliable. For further details, see, e.g., the textbooks [51, 23] and the recent reviews [157, 37]. Solving these problems is even harder when they involve both continuous and discrete variables, which require the adoption

[6] Actually, authors do not agree on the meaning of the *black-box* adjective. For example, in the foreword of book [23], John E. Dennis writes that it is misleading to include into black-box optimization those DFO methods that are not able to treat nonnumerical constraints. Within this book, DFO is interpreted as the mathematical study of optimization algorithms that do not employ derivatives, while black-box optimization is defined as the study of algorithms which assume that the objective and/or constraint functions are given by black boxes, the latters being referred to as processes that return an output when provided an input without analytical knowledge of the inner workings. An example of black box is a computer simulation or a laboratory experiment. The main difference is that DFO focuses on mathematical analysis, such as convergence analysis, while black-box optimization includes heuristics, which quickly provide approximate solutions without being supported by a rigorous analysis. At the end of Part 1, the authors admit that these definitions of DFO and black-box optimization represent their opinion and they are not universally acknowledged.

of different and specialized procedures. Such problems are called *Mixed-Integer Nonlinear Programming (MINLP)* problems. Moreover, additional complexity may be represented by nonsmooth functions over the continuous variables, which cause certain gradient–based methods to fail [37], making DFO methods particularly suited for this class of problems. Therefore, the methods described in this review are appropriate only for small–scale problems.

Among the DFO methods applied to solve MINLP problems, one can include direct search methods, model–based methods, and metaheuristics. While metaheuristics aim to find an approximate solution, the first two categories may be provided with global convergence properties. Specifically, model–based methods rely on the construction of mathematical models that approximate the original problem functions, such as quadratic approximation models, trust–region models (local search), and surrogate models (global search). Conversely, direct search methods generate iterates on the basis of function values computed at the search points. In particular, linesearch–type local searches employ the original functions in order to find a point that reduces the objective function value along given directions (see, e.g., [100, 86, 177, 67, 169, 101, 75, 170, 83, 84]). These latter methods belong to the class of directional direct search methods (as opposed to simplicial direct search methods, see [51]), which use objective function values to seek good points given a set of search directions.

All the methods described in the sequel refer to local search procedures. The choice of focusing only on local methods is due to both the scope of this thesis and the lack of significant advances in the area of global DFO, as highlighted in [37] with respect to the constrained case. However, also procedures developed for local searches gain more chance of convergence towards global solutions by applying a multistart approach [22] when the objective function is not too expensive, otherwise parallelization strategies are required [21]. Moreover, to the best of the author's knowledge and according to review [37], among the few papers that address the problem of finding a global solution of a MINLP problem, only model–based methods are adopted, such as [187].

To solve MINLP problems, all of the direct search methods perform an alternate minimization between continuous and discrete variables. While early methods explore the discrete variables by searching in a neighborhood of finite points and differ in the local search procedures applied to tackle the continuous variables, in recent years several works have proposed methods that use specialized procedures for discrete variables as well. This evolution can be interpreted in terms of stencil, i.e., the set of directions used for sampling the objective function over the discrete variables. Indeed, as opposed to the past, most of the recent methods adopt a skewed stencil using a dynamically changing stepsize for each direction corresponding to a discrete variable. Such a stencil provides the algorithm with more flexibility in neighborhoods exploration, leading to larger improvements in the objective function by using fewer iterations. In particular, as regards the early methods, [18] adapts the Generalized Pattern Search (GPS), proposed in [222], to solve a bound constrained MINLP problem with categorical variables. Three kinds of polling steps are used: continuous, discrete, and extended. While continuous and discrete polling aim to evaluate continuous and discrete points, respectively, in the extended polling a continuous search is carried out at every point found in the discrete polling. The GPS algorithm has been extended to address problems with general constraints on continuous variables [2] and stochastic objective function [218]. In [2], constraints are tackled by using the filter approach, which is introduced in [81] for sequential linear and quadratic programming and then it is extended to GPS in [19]. In [2], the authors consider a nonsmooth objective function with respect to the continuous

variables. Although the filter GPS algorithm works well in practice, in theory it is not guaranteed to converge to a Clarke stationary point. In [218], the polling phase of the GPS is combined with Ranking and Selection (R&S) procedures, thus allowing the algorithm to select the point leading to the largest improvement in the objective function, in spite of the noise due to its stochastic nature. Categorical variables and general constraints on continuous variables have been studied also in [176] and [3]. In particular, [176] proposes a general algorithmic framework whose global convergence holds for any continuous local search (e.g., a pattern search) satisfying suitable properties. In [3], the class of Mesh Adaptive Direct Search (MADS), originally introduced in [20, 17] to face nonsmooth nonlinearly constrained problems, is extended to solve MINLP problems. MADS algorithms are a generalization of GPS algorithms but with stronger convergence properties in the constrained case. Indeed, in both [20] and [3], the sequence of iterates converges to a Clarke stationary point under appropriate assumptions. This is made possible by the generation of a dense set of polling directions, while the filter GPS uses only a finite number of fixed polling directions, which do not necessarily ensure Clarke stationarity. Constraints are tackled through an extreme barrier approach. The original MADS algorithm has been recently extended in [25] to tackle MINLP problems where variables are also allowed to be granular, i.e. variables with a controlled number of decimals (integer variables can be seen as a special case), and the objective function is nonsmooth over the continuous variables. In this work, a new strategy for updating the mesh parameters is proposed.

In addition to the aforementioned references, one recent method that adopts the fixed stencil for the discrete variables is [201]. In this work, a mesh–based direct search algorithm is proposed for bound constrained mixed-integer problems whose objective function is allowed to be nonsmooth and noncontinuous. After a first polling phase involving both the continuous and the discrete variables, if a sufficient decrease is not found along the directions of continuous variables, a second phase is performed according to a tree–based exploration over the discrete variables that are in the neighborhood of the current point.

In [169] three algorithms are proposed for bound constrained MINLP problems with integer variables. Differently from the aforementioned works, the discrete neighborhood explored by the local search considered in this paper does not have a fixed structure but depends on a linesearch–type procedure. While the first and the third algorithms guarantee global convergence on a particular subsequence of iterates according to an opportunistic polling step, the second algorithm allows for stronger convergence property by adopting both complete polling step and ordering of variables. Moreover, the first algorithm was extended by [170] and [237]: the former deals with the constrained case by adopting a sequential penalty approach, while the latter replaces the maximal positive basis with a minimal positive basis based on a directions rotation technique. Bound constrained MINLP problems are considered also in [85], which extends the algorithm for continuous smooth and nonsmooth objective functions introduced in [86]. One of the authors of these works has recently proposed a unified convergence theory for nonmonotone direct search methods, which can be applied also to bound constrained MINLP problems with nonsmooth objective function over the continuous variables [83].

Other direct search methods are worth being mentioned for their influence on the development of new algorithms, even if they do not address MINLP problems. In [75], the authors propose a new linesearch-based method for nonsmooth nonlinearly constrained optimization problems, ensuring convergence towards Clarke-Jahn stationary points. The constraints are tackled through an exact penalty approach. While [75] uses two different pseudorandom sequences to generate the dense set of

search directions, [5] proposes a variant of the MADS algorithm in [20] that generates an orthogonal spanning set of polling directions in a deterministic way. In [55] and [56], the authors analyze the benefit in terms of efficiency deriving from different ways of incorporating the simplex gradient into direct search algorithms (e.g., GPS and MADS) for minimizing objective functions which do not necessarily require to be continuously differentiable. For example, one possibility is to use the simplex gradient for adjusting the order used to explore the polling directions. In [228], the authors analyze the convergence properties of direct search methods applied to the minimization of discontinuous functions, both smooth and nonsmooth. Moreover, this paper shows that imposing sufficient decrease along the polling directions can replace the use of integer lattices. As regards problems with only integer variables, both [156] and [171] aim to address the case with unrelaxable integer variables. In particular, the authors in [156] propose a method for minimizing convex black-box integer problems that uses secant functions interpolating previous evaluated points. In [171], a new method based on a nonmonotone linesearch and primitive directions is proposed to solve a more general problem where the objective function is allowed to be nonconvex. The primitive directions are used to help the algorithm not to get stuck in points that cannot be further improved. This provides the potential to find a global optimum, but it requires the exploration of large neighborhoods.

In the (possible) search phase of direct–search methods, every point producing an improvement of the objective function can be accepted without affecting the convergence properties of the method (see, i.e., [157]). For example, in NOMAD (see [4, 24, 162, 20]), which is the software package that implements the MADS algorithm, a surrogate–based model is used to generate promising points. Moreover, the combination of MADS with the variable neighborhood search metaheuristic (see, e.g., [184] and [111]) helps the algorithm to escape from local minima [16]. Due to this potential role in direct–search methods used for addressing MINLP problems, the recent advances for model–based methods and metaheuristics are reported below.

Most of the algorithms adopting model–based methods aim to solve problems involving computationally expensive functions. In [53], the authors propose the open–source library RBFOpt for solving MINLP problems with bound constraints. The algorithm employs radial basis functions to build a surrogate model of the black-box objective, which is less expensive to evaluate. Moreover, a noisy oracle is exploited to reduce the number of function evaluations required to achieve convergence. Bound constrained problems are addressed through quadratic models in [196], which extends to the mixed-integer case the trust–region derivative-free algorithm BOBYQUA introduced in [202] for continuous problems. Surrogate models employing radial basis functions are used also in [188], which proposes an algorithm, called SO–MI, able to converge to the global optimum almost surely. In particular, the surrogate model is used to select the next point to sample the objective function among the candidates generated by applying different perturbation methods to the current best point. A similar algorithm, called SO–I, is proposed by the same authors in [186] to address integer global optimization problems. Unlike SO–MI, SO–I does not require the user to provide an initial feasible solution. Moreover, at each iteration, only one point is used to sample the expensive objective function, instead of the 4 points used by SO–MI. In [185], the authors propose an algorithm for MINLP problems that modifies the sampling strategy used in SO–MI and uses also an additional local search. In [109], the Kriging model is employed to develop a new sequential algorithm for MINLP problems that can be considered as an extension of the well-known EGO algorithm, which is introduced in [124] for efficient global optimization. The Kriging model is used also in [113], which sequentially builds a surrogate model of the MINLP problem in hand and then applies a branch–and–bound approach for determining

its solution (since an analytical representation is available for the surrogate model). The effectiveness of this approach is shown by the computational results, where a genetic algorithm is used as a benchmark.

As regards metaheuristics (see, e.g., the books [221] and [87]), among the methods applied to solve MINLP problems there are evolutionary algorithms [54] such as genetic algorithms [117, 58, 61], particle swarm algorithms [239, 144, 238], and simulated annealing [208, 229]. The same classes include also the metaheuristics addressing integer problems (e.g., for genetic algorithms see [121], for particle swarm optimization see [213, 158], and for simulated annealing see [190]). Moreover, integer problems are also addressed through scatter search [155], ant colony optimization [242], and tabu search [80]. All these methods aim to solve the problem in hand without using the derivatives of the objective function, although they may be available. However, while in derivative-based optimization the metaheuristics enable obtaining approximate solutions in short time, in derivative-free optimization, which is frequently characterized by computationally expensive functions, using metaheuristics may be prevented by the many function evaluations that usually are required. To this aim, surrogate models can be employed to improve the efficiency of metaheuristics [198] and, in turn, metaheuristics can be used to improve the direct search methods [99].

2.4 Emergency department applications

Emergency Medical Service (EMS) represents one of the most important healthcare services, considering that it concerns people's lives. The recent survey paper [14] reports a comprehensive review on EMSs and introduces the novel concept of Emergency Care Pathway (ECP). Following the new patient–centered approach, which focuses on patients rather than caregivers, the ECP considers the whole healthcare chain, composed by several steps (namely the appropriate sequence of activities), aiming to increase patients' safety and gratification. Relying on ECP, the entire Emergency Care Delivery System is considered instead of single EMSs, thus allowing for optimal allocation of the resources requested along the whole pathway. Of course, for this purpose, an effective interaction among ECP stakeholders is needed to guarantee timeliness and fairness of the services delivered.

The Emergency Department (ED) represents the entry point of the ECP and its operational efficiency is fundamental for providing healthcare services to people who need urgent medical treatments. An ED is open 365 days a year and 24 hours a day, and people with different urgency arrive requiring treatment. Since the services delivered by an ED are time–critical, the main issue concerns the response time. Unfortunately, the well known and growing problem of overcrowding tends to enlarge the waiting times, endangering the life of critical patients. Today the overcrowding is an international phenomenon widely considered in the specific literature [118, 231, 232]. In particular, according to [118], possible causes of overcrowding are insufficient staff, shortcomings of the structures, flu season, request of nonurgent treatments, unavailability of hospital beds. Besides treatment delays, other possible consequences of overcrowding are reduced quality of the services and higher patient mortality, ambulance diversion, growing number of patients who leave without being visited, greater expenses for the service provider due to longer patient stay. Moreover, negative issues due to overcrowding are greatly amplified whenever mass casualty disasters occur. In this case, the rate of patient arrivals suddenly increases and some different tactical and strategic decisions must be adopted to ensure timely treatments.

The specific literature dedicated to ED overcrowding and ED management under disaster conditions is reviewed in Sections 2.4.1 and 2.4.2, respectively. Sections 2.4.3 and 2.4.4 focus on two important problems which emerge when dealing with ED simulation models. The first one regards the need for an efficient representation of the ED arrival process. The second one concerns the estimation of the information that is not available but it is required for building a reliable simulation model. The latter aspect depends also on the accuracy of the arrivals given as input, thus making the two problems interconnected.

2.4.1 Emergency department overcrowding

In the recent years, techniques from Operations Research have been frequently applied for studying the problem of ED overcrowding. Since 2005 the US *National Academy of Engineering* and the *Institute of Medicine* have highlighted the importance of using tools from Operations Research and Systems Engineering (statistical process controls, queuing theory, mathematical modeling and simulation) in healthcare delivery, in order to improve performance of care processes or units [205]. It is well known that complexity and high variability of the processes related to healthcare delivery in most cases make the application of standard techniques very difficult. *Simulation* is considered of fundamental importance for analyzing several healthcare settings (see, e.g. Reid et al. [205], and the several examples reported therein, and Almagooshi [10]). In particular, simulation models have been widely applied to emergency medical service operations (see Aboueljinane et al. [1] for a survey). Several papers in the simulation literature are devoted to study the patient flow through an ED by means of DES models [1, 34, 104, 126, 153, 203, 235, 243, 151] and Agent Based Simulation (ABS) models [13, 133, 167, 219, 230]. In particular, DES has been extensively used for studying causes and effects of ED overcrowding. One can refer to [200] (and to the many references reported therein) and to the more recent paper [194] for a review of simulation studies available in the literature devoted to the ED overcrowding phenomenon and for a discussion on the effectiveness of using DES models. Among the many papers that deal with ED overcrowding, it is important to mention [235], where an ED in Hong Kong is simulated in order to assess how modifying the path of the patient clinical process and the level of physician resources affects the performance; [126], where the simulation model is used to understand the process that allows shortening the waiting times and the length of stay by varying the workload among staff members and by giving nonurgent patients the possibility of returning afterwards; the many papers addressing the adoption of fast–track systems in the ED, such as [152] and [15], which propose to send less urgent patients to specific queues so that they receive the service early, thus being discharged in a shorter time. Furthermore, in [234] an in–depth study on patients interarrivals and lenght of stay in an ED located in Israel is reported, while in [57] a stochastic model is considered in order to reduce the average total patient waiting time in a university hospital. From the wide literature on the topic, it clearly emerges that a study based on DES provides important insights into ED overcrowding. Moreover, since the management of ED resources strongly affects overcrowding, an optimal resource allocation is considered a key tool for achieving a good level of service and possibly reducing overcrowding (see the recent survey paper [8] for a review on different analytical methods and modeling techniques used for improving patient flow through an ED by means of optimal resource allocation).

A simulation-based study can be also combined with optimization tools in order to determine which setting results the best, once one or more objectives (to be minimized or maximized) are defined. The SBO approach has been also applied in

healthcare contexts (see, e.g., [43, 95, 174, 175, 244] and the very recent systematic review [240]) and, in particular, in dealing with ED [7, 65, 105, 106]. For instance, in [105] a simulation model for the ED of a public hospital in Hong Kong is built and integrated with an optimization tool in order to find an optimal medical staff configuration to minimize the total labor cost given the service quality requirement; in [65], the patient flow through an Australian ED is studied and optimized on the basis of bed configurations.

It is important to mention that some measures have been proposed to give a formal assessment of the degree of overcrowding. They enable monitoring the state of the ED by describing the current situation, and they can also work as alarm bells to avoid reaching a critical level. The most commonly used are: the Real Time Emergency Analysis of Demand Indicators (READI), the Emergency Department Work Index (EDWIN), the Work Score, the National Emergency Department Overcrowding Scale (NEDOCS). These continuous–valued indicators are computed on the basis of some operational variables which enable quantifying the degree of overcrowding of an ED (see [119] and the references reported therein for the definition of these methods of measurement). However, the study reported in [119] shows that none of these measures actually provides a reliable predictive analysis at a low percentage of false warning.

2.4.2 Emergency department and disaster conditions

In the wide literature on ED management, few papers have been devoted to studying the effects of peak arrivals on an ED due to disaster or extreme events. In particular, [130] provides a characterization of different strain situations in an ED and introduces some "strain indicators"; moreover, a simulation-based decision support system is also developed to prevent and predict such situations. As regards papers more specifically devoted to ED peak arrivals caused by critical events, [104] reports where a literature review on ED simulation models for both normal and disaster conditions. In particular, in their systematic classification covering 106 reviewed papers, the authors indicate only 5 papers devoted to ED simulation model applications during disaster conditions: [236], where the patient workflow through an ED located in Western New York during extreme conditions is studied aiming to reconfigure the workflow for improving the overall management of the patient flow; [199], where the impact of a hypothetical bioterrorist attack on a medium–sized ED located in Texas is assessed; [9], where the performance of an ED is evaluated under critical conditions due to a disaster and several scenarios for disaster recovery plans are examined; [125], where different arrival patterns to an ED during a conventional terror disaster are considered along with an estimate of additional resources that would be required to accommodate all patients arriving at the ED; [42], where an "optimal scarce resource–rationing principle" for allocating scarce ED medical resources in natural disaster responses is sought. More recently, [102] and [103] have studied disaster scenarios corresponding to a patient flow surge for EDs located in an earthquake area in Istanbul, Turkey. In particular, [102] has developed a model that enables early preparedness of ED resources to overcome bottlenecks due to critical situations; [103] has proposed a hybrid framework which uses artificial neural networks to estimate the number of casualties and a DES model to analyze the effect of surge in patient arrival as consequence of a disaster in a network of five EDs located in a high earthquake risk region. All these papers dealing with the impact of disaster events on EDs use DES models to represent patient flow, highlighting the importance of such models to improve ED policies in these critical cases.

2.4.3 Arrival process to the emergency department

Statistical modeling for describing and predicting patient arrival to EDs represents a basic tool of each study concerning patient load and crowding. Indeed, all the approaches adopted for this purpose require an accurate model of the patient arrival process, which plays a key role in dealing with EDs. Every modeling methodology is generally based on assumptions that, in some cases, may represent serious limitations when applied to complex real–world cases, such as ED operations. In particular, when dealing with ED patient arrival stochastic modeling, due to the nonstationarity of the process, a standard assumption is the use of *Nonhomogeneous Poisson Process (NHPP)* [6, 7, 105, 139, 153, 243, 29]. It can be useful to recall that a counting process $X(t)$ is a NHPP if *1)* arrivals occur one at a time (no batch); *2)* the process has independent increments; *3)* increments have Poisson distribution, i.e. for each interval $[t_1, t_2]$,

$$P\left(X(t_1) - X(t_2) = n\right) = e^{-m(t_1,t_2)} \frac{[m(t_1,t_2)]^n}{n!},$$

where $m(t_1, t_2) = \int_{t_1}^{t_2} \lambda(s)ds$ and $\lambda(t)$ is the arrival rate. Unlike the Poisson process (where $\lambda(t) = \lambda$), NHPP has nonstationary increments and this makes the use of NHPP suitable for modeling ED arrival process, which is usually strongly time–varying. Of course, appropriate statistical tests must be applied to available data for checking whether NHPP fits. This is usually performed by assuming that NHPP has a rate which can be considered approximately piecewise constant. Hence, Kolmogorov–Smirnov (KS) statistical test can be applied in separate and equally spaced intervals and usually the classical Conditional–Uniform (CU) property of the Poisson process is exploited (see, e.g., [41, 139, 140]). Unlike standard KS test, in the CU KS test the data is transformed before applying the test. More precisely, by CU property, the piecewise constant NHPP is transformed into a sequence of i.i.d. random variables uniformly distributed on $[0, 1]$ so that it can be considered a (homogeneous) Poisson process in each interval. In this manner, the data from all the intervals can be merged into a single sequence of i.i.d. random variables uniformly distributed on $[0, 1]$. This procedure, proposed in [41], allows removing nuisance parameters and obtaining independence from the rate of the Poisson process on each interval. Hence, data from separate intervals (with different rates on each of them) and also from different days can be combined, avoiding the common drawback due to large within-day and day-to-day variation of the ED patient arrival rate. Actually, Brown et al. in [41] apply CU KS test after performing a further logarithmic data transformation. In [140, 141], this approach is extensively tested along with alternative data transformations proposed in early papers [72] and [163].

Kim and Whitt in [139] observe that the procedure described above needs special attention when applied to ED patient arrival data. This is due to the fact that the following three issues must be seriously considered: *1) data rounding*, *2) choice of the intervals*, *3) overdispersion*. Indeed, the first issue may produce batch arrivals (zero length interarrival times) that are not included in a NHPP, so that unrounded data (or an unrounding procedure) must be considered. The second is a major issue in dealing with ED patient arrivals, since arrival rate can rapidly change so that the piecewise constant approximation is reasonable only if the intervals are properly chosen. The third issue regards combining data from multiple days. Indeed, in studying the ED patient arrival process, it is common to combine data from the same time slot over different weekdays, being this necessary when data from a single day is not sufficient for statistical testing. Data collected from ED database usually shows large variability over successive weeks mainly due to seasonal phenomena,

such as flu season and holiday periods. However, the overdispersion phenomenon must be checked by using a dispersion test on the available data (see, e.g., [132]).

2.4.4 Calibration of emergency department simulation models

Many papers in the literature deal with quality of input data in ED simulations. This issue, which strongly affects the reliability of the results, is carefully analyzed. Indeed, the cost, time, and challenges required by collecting ED empirical data represents a serious limitation for every simulation model [200]. In order to replicate and predict the patient flows within the ED, [71] develops a process mining approach which handles the noise factors in the dataset after introducing assumptions on how to interpret the data. In particular, these factors include the following cases: starting and ending time of each activity performed in the ED may be unknown; information about urgent patients may be registered after the activity is completed and, in general, the timestamps of each activity may not be promptly recorded at the right time; for each patient, the same activity may be recorded multiple times for technical reasons, giving rise to misleading information. A framework to categorize all the ED data quality issues is proposed in [226], which also provides assessment techniques for each data quality problem category. Moreover, this paper highlights that most of the works in the literature focus on the problem of missing data, which is also the problem addressed in this section. It consists in the lack of information on some or all the key timestamps that define the activities performed in the ED, i.e., starting and ending time. This well-known issue prevents gaining knowledge about the duration of each activity, which is required to estimate the corresponding probability distributions to use in the simulation model. Several simulation-based optimization approaches are proposed in the literature to tackle this crucial problem. The idea behind each approach is to leverage the known information in order to estimate the parameters of the probability distributions underlying the missing data. This is accomplished by comparing the KPIs computed through the simulation model with the corresponding values derived by the data collected in the real system. The resulting procedure is known as *model calibration*.

The mathematical formulation of the optimization problem and the specific algorithm used for determining the optimal solution are the two features that distinguish the papers dealing with missing data. For instance, both [153] and [106] adopt similar approaches that leverage the time differences between the known timestamps. The former proposes an unconstrained optimization problem where the objective function is a consistency measure that compares the average, the standard deviation, and the proportions of the time differences for each triage tag. The latter uses a constrained optimization problem where the objective function is based on a modified chi–square goodness of fit and the constraints are introduced to guarantee each time difference the same level of accuracy. As regards the approaches used to solve the problem, both papers use metaheuristic procedures. In particular, [153] considers both a descent method and a simulated annealing algorithm, while [106] adopts an approach that combines a genetic algorithm with simulated annealing and optimal computing budget allocation. Moreover, in [106] the authors point out that the approach in [153] could be improved in several ways: including in the objective function all the time differences defined by the available timestamps, since delays in one activity may impact on downstream activities in the patient flow; modeling the possibility for each patient to have more than one medical visit, which may affect the time differences; using a more formal objective function and solving the resulting optimization problem through a more efficient algorithm.

Another paper in the literature on missing data that is worth being mentioned

is [168], which assumes an agent–based simulation model, as opposed to the DES model considered in [153] and [106]. Although the framework underlying this work is different, a simulation-based optimization problem is used for the same goal of minimizing the deviation between real data and simulation output, in order to estimate the missing parameters of the simulation model. The LOS, i.e., the difference between the discharge time and the arrival time to the ED, is the time difference considered in the objective function, which adopts the Jensen–Shannon divergence. A systematic method based on the pattern search method APPSPACK [98] is used to find the optimal configuration of parameters.

Chapter 3

A simheuristic algorithm for solving an integrated resource allocation and scheduling problem

One of the challenging tasks faced by numerous companies and organizations concerns the integrated allocation and scheduling of resources, such as machines, equipment, and personnel. Although these problems frequently arise in the management of healthcare services, like EDs, the SBO methodology proposed in this chapter draws inspiration from real–world applications related to the industrial field. These problems may be even more difficult when real–life uncertainty is considered, which requires the adoption of simulation. Therefore, the resulting integrated optimization problems provide the opportunity to demonstrate how simulation can be effectively integrated with optimization strategies in order to determine approximate solutions.

After introducing the integrated allocation and scheduling optimization problem with stochastic processing times, in this chapter a simheuristic algorithm is proposed to efficiently solve this SBO problem. The approach adopted is based on the hybridization of simulation with a metaheuristic, thus providing the proper tools to tackle the stochastic version of the integrated allocation–scheduling problem. The numerical experiments show the efficiency of the methodology proposed as well as the potential applications in real–life industrial settings.

3.1 The integrated resource allocation and scheduling problem

Several organizations face the problem of efficiently managing the resources necessary for carrying on their activities and processes. This is true, in particular, both for long-term objectives, which involve a proper estimate and planning of the needs, as well as for short-term decisions, such as the optimal sizing and assignment of resources to operations. Indeed, in most real–life environments, it is reasonable to assume that the higher the number of allocated resources, the lower the time

This chapter is based on Maccarrone, L., Giovannelli, T., Ferone, D., Panadero, J., Juan, A.A.: *A simheuristic algorithm for solving an integrated resource allocation and scheduling problem.* In: 2018 Winter Simulation Conference (WSC), pp. 3340–3351 (2018). DOI 10.1109/WSC.2018.8632296,
© 2018 IEEE.

needed to complete the planned tasks. Therefore, in order to efficiently address the management of resources, two conflicting goals must be balanced: *(i)* minimization of the cost of the resources; and *(ii)* minimization of the time needed to complete the assigned tasks.

In addition to this challenging trade-off between conflicting optimization objectives, real–life is also characterized by uncertainty, such as stochastic service times and unexpected events giving rise to disruption to the original plans. The approach proposed in this chapter aims to tackle the integrated allocation and scheduling of resources under stochastic service times. In particular, the resulting stochastic optimization problem can be described by the following bilevel decision making process:

- **Stage 1:** heterogeneous resources (machines, equipment, personnel, etc.) are allocated to different activities in order for them to be completed at a reasonably low cost.
- **Stage 2:** the activities, which are subject to stochastic service times, need to be properly scheduled in order to be completed in a reasonably low makespan (i.e., the time required to finish the processing of all the jobs on all the machines).

After being tentatively allocated to activities in the first stage, resources, along with their allocation mapping, are sent to the second stage. Here, the activities are scheduled considering the joint effect on the service times caused by both resource allocation and uncertainty. In the first stage, the cost associated with the allocated resources is computed. In the second stage, an evaluation of the solution, in terms of makespan, is performed. These two values are then weighted in an overall objective function, which must be able to take into account both financial and time goals (this is achieved by transforming time measures in monetary values). Since the scheduling plan in the second stage depends on the resource allocation defined in the first stage, different combinations of promising resource allocations need to be assessed in order to generate several solutions, each associated with the corresponding cost (cost of the allocated resources plus scheduling costs).

The main contribution of the approach considered in this chapter is to demonstrate how simulation can be combined with optimization to propose an algorithm that is able to deal with this integrated stochastic resource allocation and scheduling problem. The approach is based on the concept of simheuristics, which hybridize simulation with metaheuristics in order to solve complex combinatorial optimization problems with stochastic components (see [127] for a detailed review). Simheuristic algorithms can be seen to some extent as a specialized case of SBO where: *(i)* the optimization algorithm is a metaheuristic; and *(ii)* simulation is not only used to evaluate objective and constraint functions, but also to provide feedback that can be used by the metaheuristic procedure to improve the solution searching process ([97]). As regards the metaheuristic, the Greedy Randomized Adaptive Search Procedure (GRASP) is the algorithm selected [76]. GRASP can be easily integrated into a simheuristic framework providing a good trade-off between quality of the solutions and ease of implementation.

The structure of this chapter is summarized as follows: after reviewing close approaches from the literature in Section 3.1.1, Section 3.2 describes the main features of the integrated resource allocation and scheduling problem considered. The main ideas underlying the simheuristic algorithm are described in Section 3.3. Finally, an implementation of the method is described in Section 3.4, where test problems are used to assess its effectiveness.

3.1.1 Integrated resource allocation and scheduling problems in the literature

Resource allocation and scheduling of activities are some of the most well–studied problems in the literature. The integrated version of these problems involves finding the right assignment of resources to each activity, so that their duration is reduced and a balance between the cost of additional resources and the makespan is achieved. This topic is addressed in different research areas. For instance, in the healthcare literature, [178] deals with both assignment of patients to ED beds and scheduling of treatments, while [166] proposes a two–stage simulation-based heuristic for efficiently managing the resources and the appointment scheduling in outpatient clinics. In the job scheduling and project management literature, several works extend previous applications by considering the integrated problem. One of the first papers introducing the idea of simultaneously planning resource requirements and scheduling of activities is due to [134]. The problem they describe is known in the literature with the name of Time–Cost trade-off Problem (TCTP). In the area of project management, the same topic is also referred to as Activity Crashing or Project Compression problem [73]. Existing approaches differ in the way they represent the non–increasing pattern between allocated resources and activity duration. On the one hand, some continuous approaches assume a linear function [28, 142], while others make use of nonlinear functions [189, 60]. On the other hand, the discrete version of the TCTP assumes that a discrete set of possible activity durations is given [150, 62].

While the deterministic version of the TCTP has been widely studied during the last 60 years, the works considering the stochastic counterpart of the same problem are relatively rare [114]. The existing literature can be classified using different criteria. In particular, different groups can be obtained by considering:

- *the type of objective:* some works take into account conflicting objective functions in a multiobjective approach, while others deal with a single objective function. This function either includes both resources costs and total completion time or only one of them. In the latter case, the second objective is generally modeled as a constraint [91]. A multiobjective problem is proposed by [26] and [27], which use an interactive procedure with four objective functions related to the total cost and total completion time of the project. In [191] and [148], only one of the two measures is minimized and the other objective is forced under a given threshold by introducing an additional constraint. Finally, examples of approaches including time and cost goals in a single weighted objective function can be found in [74].

- *the type of relation between resource allocation and activity processing times:* as for the deterministic version of the TCTP, a basic distinction can be made considering the nature of patterns between costs and durations. The existence of a continuous relationship is assumed in [159] and [50], while [148] or [91] propose different approaches to solve the stochastic version of the TCTP in the discrete case.

- *the type of assignment (scheduling policy):* in general, the assignment procedure used to allocate resources can be either static or adaptive. The solution is static if the assignment of resources is fixed and does not change as the activities are processed [40, 191]. Otherwise, the policy is adaptive and decisions on the assignment can be adjusted over time [92, 131].

- *the method for considering stochasticity:* several approaches are applied to deal with uncertainty. According to [131], Monte Carlo Simulation (MCS) is used for the first time by [225]; then, it is adopted along with heuristic methods in many papers. Other common techniques include robust optimization [50], and stochastic programming [148].

The approach developed in this chapter considers the minimization of a single weighted objective function by assuming a continuous relationship between the resources allocated and the stochastic service times. MCS is integrated into a metaheuristic framework to simultaneously provide a static resource assignment policy and a feasible activity scheduling.

3.2 Statement of the simulation-based optimization problem

In the area of scheduling, numerous problems can be distinguished on the basis of the type of configuration considered. In the following, a description of the integrated resource–allocation and scheduling problem as an extension of the well-known Permutation Flow Shop Problem (PFSP) is provided. The classical version of PFSP can be described as follows: a set J of n jobs has to be processed by a set M of m machines. Every job $j \in J$ is composed by a set of m operations, each one indicated by O_{ij}. The operations must be sequentially performed by the m machines (one operation per machine). Moreover, the processing order of operations in machines is the same for all jobs, i.e., all jobs are processed by all machines in the same order. A positive processing time p_{ij} is associated with each operation O_{ij} and it is assumed to be known in advance. The goal is to find a sequence (permutation) of jobs so that a given criterion is optimized (see [128]). The most commonly and studied criterion is the minimization of the makespan. Adding uncertainty to the PFSP results in the PFSP with stochastic processing times (PFSPST), in which the processing time of each job j in each machine i is not a constant value but it is represented by a nonnegative random variable P_{ij}. For this purpose, common approaches use known probability distributions, such as lognormal and Weibull.

Integrating the resource allocation into the PFSPST involves the definition of additional variables to represent the amount of resources assigned to the machines. In the sequel, a simple case with homogeneous resources is considered. Let r be the vector whose components r_i indicate the assignment of resources to machine i. According to the nature of the trade-off problem, the processing times of the operations are assumed to change with the value of r_i. In general, the time required by a task decreases as more resources are assigned, and the relation between the time and the number of resources is usually sublinear [74]. Without loss of generality, in the numerical experiments the 'accelerated' processing times $\hat{P}_{ij}$ are considered. In particular, given the single–resource processing times P_{ij}, it follows that

$$\hat{P}_{ij} = \frac{P_{ij}}{\sqrt{r_i}} \text{ for all } i \in M \text{ and } j \in J. \tag{3.2.1}$$

However, note that the type of relation between processing times and number of resources is completely arbitrary in the framework proposed and this assumption could be easily modified to consider also other kinds of functions, eventually different for each particular machine. Moreover, it is important to point out that $\hat{P}_{ij}$ is a random variable as well.

An example with three machines and three jobs is reported in Figure 3.1. In the first Gantt chart, a single resource is assigned to each machine and the overall makespan is 26. In the second chart, adding a second resource to machine M2 results in lower processing times and a better makespan.

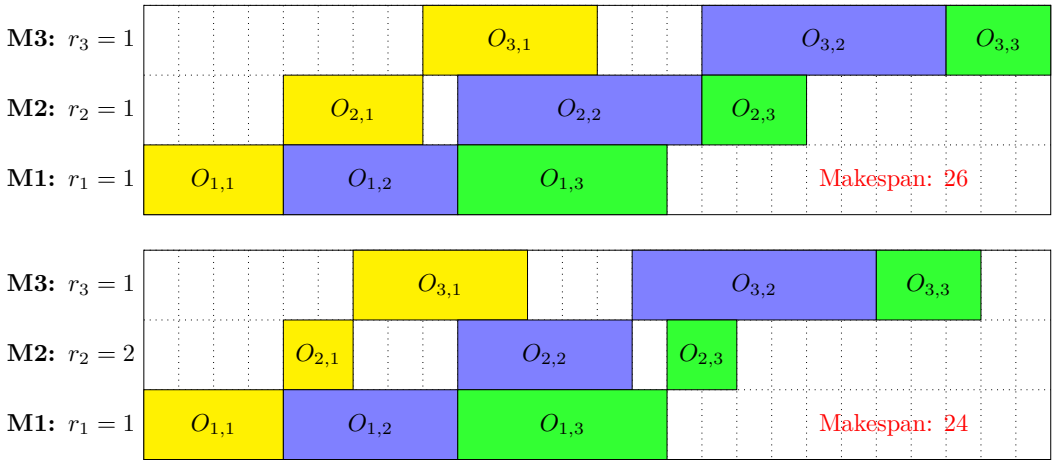


Figure 3.1. Difference in makespan with more resources assigned to machines (reproduced from [180], © 2018 IEEE).

On the one hand, adding more resources reduces the time needed to process all the operations. On the other hand, these resources increase the corresponding cost component in the weighted objective function. Therefore, apart from the cost directly linked to the production volume (e.g., row materials) and the fixed cost, the typical factors influencing the total cost of a manufacturing enterprise are linked to the acquisition of resources (e.g., costs for buying equipment and hiring personnel) and the time spent to carry on activities (e.g., renting cost, electricity cost, and salaries). To take into account these aspects in a single minimization goal, the following objective function is used

$$\lambda \cdot \sum_{i=1}^{m} r_i + \mu \cdot \mathbb{E}[C], \tag{3.2.2}$$

where $\mathbb{E}[C]$ is the expected makespan of the solution, while λ and μ are two positive weigths such that the condition $\lambda + \mu = 1$ holds. Here, homogeneous resources are assumed, i.e., each resource contributes in the same way to increase the value of function (3.2.2). Note that the makespan C is a function of the random variable $\hat{P}_{ij}$ along with the scheduling plan chosen. Therefore, C is a random variable as well, and an unbiased estimator for $\mathbb{E}[C]$ is represented by the sample mean, which can be computed through a MCS.

3.3 The simulation-based optimization approach

In order to deal with the integrated resource allocation and stochastic scheduling problem described in the previous section, a simheuristic algorithm is proposed. It is a multi–start algorithm which combines simulation techniques with a GRASP and it is based on a randomized construction process combined with a local search. During the construction phase, a feasible solution is iteratively constructed by adding a new element to the current solution. At each step, the next element to add to the current

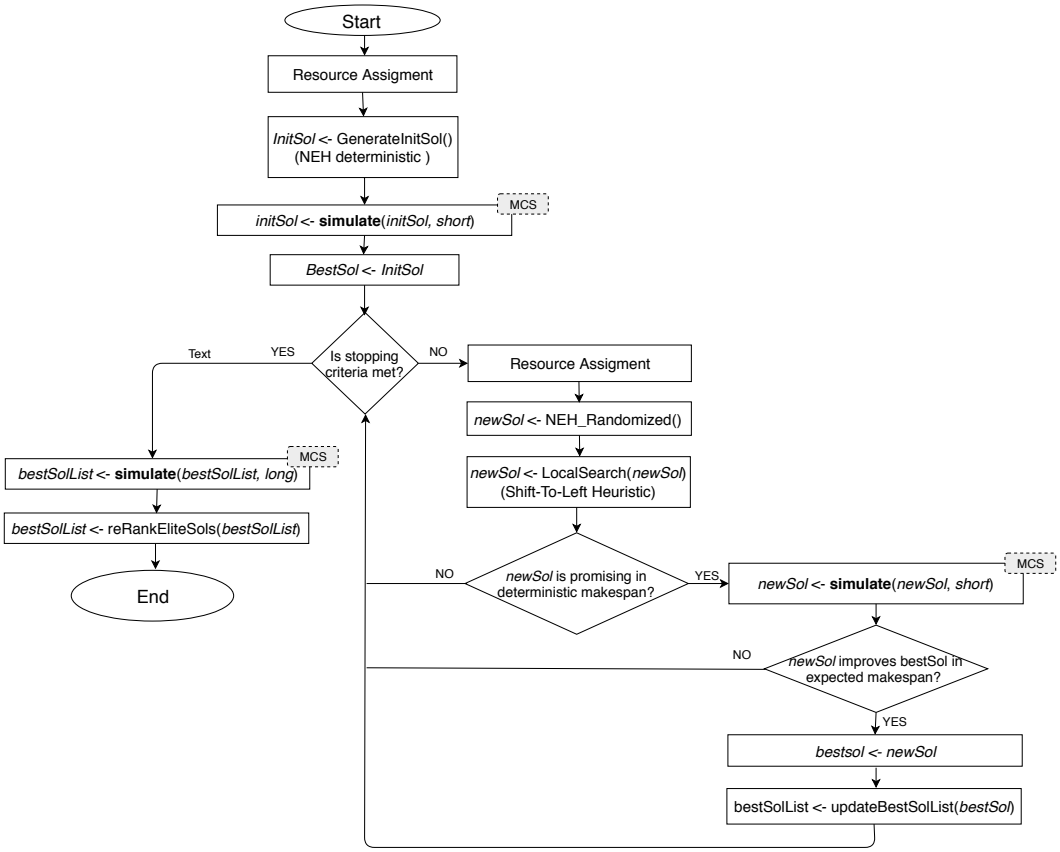


Figure 3.2. The proposed simheuristic algorithm (reproduced from [180], © 2018 IEEE).

solution is determined by ordering all candidate elements (i.e., all elements that can be feasibly added to the solution) according to a greedy function $g\colon C \to \mathbb{R}$. This function measures the benefit of including an element in the emerging solution. The element is randomly chosen among the most promising candidates, thus it is not necessarily the top one. Note that this technique tends to generate different solutions every time the multi–start procedure is run, which helps to avoid getting trapped into a local minimum point. Once a feasible solution has been built, it is locally improved with respect to a neighborhood, in order to find a local minimum point. This phase is needed, since the initial solution is unlikely to be optimal after the construction phase. The GRASP metaheuristic is chosen since it is relatively easy-to-implement, does not contain a large number of parameters requiring time-consuming setting processes, and has been successfully applied to a wide range of different optimization problems (see, e.g., [78, 79, 93]).

Figure 3.2 depicts the main characteristics of the simheuristic algorithm, which is composed of three stages. In the first stage, a feasible initial solution is constructed. Then, during the second stage a GRASP algorithm enhances the initial feasible solution by iteratively exploring the search space and conducting a short number of MCS runs. During this stage, a reduced set of promising solutions is obtained. In the third stage, a large number of simulation runs are performed to refine the promising solutions with the goal to obtain the best–possible solution in stochastic terms.

Note that the first stage is composed of two different phases. First of all, the

algorithm carries out a resource assignment. In particular, a number of resources between 1 (minimum) and q (maximum) is randomly assigned to each machine, thus determining its processing time, which is given by Equation (3.2.1). Once the assignment is performed, an initial solution is generated with the help of the well-known NEH constructive heuristic [195]. During the second step of the methodology, the initial solution (*initSol*) is improved using the GRASP algorithm. At the beginning of the algorithm, *initSol* is copied into *bestSol*. Next, an iterative procedure starts in order to create new solutions. Resources are randomly assigned. Then, in order to generate a new solution (*newSol*), the NEH heuristic is run using a biased–randomization strategy as proposed in [96]. This strategy relies on a skewed probability distribution to assign decreasing probabilities to each possible move of the constructive heuristic. In this case, a Geometric probability distribution with a single parameter β ($0 < \beta < 1$) is employed. A detailed explanation of this strategy, together with successful application examples, can be found in [70, 129] or [69].

Next, *newSol* undergoes a local search phase in order to find a local minimum. In the local search, the *shift-to-left* operator proposed in [128] is utilized. The main idea behind this operator is to iteratively examine all the jobs and try to shift them to the left. If the job movement improves the makespan, the movement is accepted, otherwise the solution is not modified. Once the local search returns a *newSol*, if this new solution is promising in terms of deterministic makespan, then it is processed by a 'fast' (200 runs) MCS in order to estimate the expected makespan. A reduced number of runs is applied to avoid the simulation jeopardizes the metaheuristic time. Whenever the stochastic solution outperforms the *bestSol*, the latter is updated to *newSol*, the new *bestSol* is saved in the pool of the best solutions, and the process continues until it reaches the maximum execution time. In the experiments, this maximum execution time has been defined as follows

$$maxTime = n \cdot m \cdot t, \tag{3.3.1}$$

where n is the number of jobs, m is the number of machines, and t is a time factor (initially set to 0.03 seconds). When the GRASP stage ends, the algorithm returns a selected list with the best solutions (in the experiments, the best five ones are selected). For each of these solutions, a more intensive simulation consisting of $1,000$ runs is performed. This longer simulation provides a more accurate estimate of the expected makespan $\mathbb{E}[C]$.

3.4 Numerical experiments

The algorithm is implemented in Java and tested on a personal computer with an i7 Quad core working at 2.4 GHz and with 6 GB of RAM. For the computational experiments, the classical PFSP benchmark set proposed in [220] is employed and extended to the integrated problem. The original set consists of 12 sets of 10 instances each one, ranging from 20 to 500 jobs to be completed on 5 to 20 machines. As these instances are deterministic, they are extended by considering lognormal distributions with $\mathbb{E}[P_{ij}] = p_{ij}$. In order to compare the results, 10 instances used in [77] are selected and their costs are adapted assuming that one resource is allocated to each machine. Each instance is executed 10 times using different seeds. The lognormal variance used in the experiments is given by $Var[P_{ij}] = k \cdot \mathbb{E}[P_{ij}]$, being k an experimental design parameter that influences the level of variability. Three different variance levels are used to test the methodology proposed: $k \in \{5, 10, 20\}$. It is important to point out that, in a real–life application, the specific value of this

parameter would be adjusted based on historical observations, but the methodology would remain the same.

Table 3.1 shows the best solutions found over 10 different runs of the algorithm, one for each different seed. The first column reports the instance name. The second column shows the cost summarized in [77] plus the resource cost assuming that each machine uses one resource. These solutions are used as a reference for comparison with the best solutions found by the metaheuristic approach. The next two columns (third and fourth), report the best deterministic value found by the algorithm and the percentage gap with respect to the previous results, respectively. Finally, the last three columns report the best stochastic cost for each variance level.

Table 3.1. Results for 10 Taillard instances with different variability levels

Instance	**[77]**	**Det. value**	**%-Gap(2-3)**	**Stochastic values**		
[1]	**[2]**	**[3]**	**[4]**	**[5](k=5)**	**[6](k=10)**	**[7](k=20)**
tai097	12882.00	12061.06	-6.37	12181.46	12108.56	12476.26
tai102	15425.00	15425.00	0.00	17321.55	17370.55	17465.48
tai103	15513.00	15513.00	0.00	17394.32	17468.20	17215.68
tai104	15470.00	15470.00	0.00	17252.67	17270.40	16603.26
tai105	15380.00	15380.00	0.00	17036.50	17215.58	17047.88
tai107	15517.00	15517.00	0.00	17360.07	17341.70	17290.16
tai108	15536.00	15536.00	0.00	16763.63	17029.10	16903.45
tai112	30810.00	29122.46	-5.48	29169.32	32738.78	32588.69
tai113	30613.00	28852.54	-5.75	32164.19	29117.31	32715.56
tai118	30767.00	29165.26	-5.21	29227.71	29250.82	29334.08
Average	19791.30	19204.23	-2.28	20587.14	20691.10	20964.05

First, the results produced by the algorithm for the deterministic case have been compared with the solutions provided in [77] (Column 2). Looking at Column 4, it can be observed that adding more resources to machines tends to reduce the total cost of the best solution. However, for other instances, the best solution is not improved, obtaining the same solutions as in Column 1 (where a single resource to each machine is assigned). This behavior seems to indicate that in the latter instances the makespan savings associated with increasing resources do not compensate their marginal cost.

In a similar way, the simheuristic approach is used in the stochastic scenario. On the average, the stochastic cost of the solutions increases with the variance level k. Note that, for most instances (all except tai102 and tai107), the makespan can decrease as the variance level increases. To better study this behavior, another set of experiments is conducted on instances tai104 and tai113, where the makespan decrease was noticeable. In this new experiment, the time factor t is increased (from 0.03 to 3 seconds), as well as the number of runs (they have been increased to 500 and 2000 for short and long simulations, respectively). The results are reported in Table 3.2. The makespan is still decreasing for instance tai104, but now the magnitude of this reduction is small. For instance tai113 this decreasing effect has disappeared. Therefore, results in Table 3.1 are probably due to the small number of simulation runs performed in that initial experiment.

Figure 3.3 reports the boxplot of the percentage gaps between the best stochastic solution and the best deterministic solution found at each run. It seems clear that the expected cost grows as the variance increases, inducing larger gaps. This effect is what one should expect and it adds credibility to the results obtained by the algorithm. It also illustrates that using the best deterministic solution in a stochastic

Table 3.2. Results for 2 Taillard instances with different variability levels and more computational times

Instance	Stochastic values			% gap	
[1]	[2](k=5)	[3](k=10)	[4](k=20)	[3-2]	[4-3]
tai104	16403.94	16342.74	16344.01	-0.37	0.01
tai113	27779.78	27832.76	27835.63	0.19	0.01

framework might be a bad decision, since it usually provides suboptimal values.

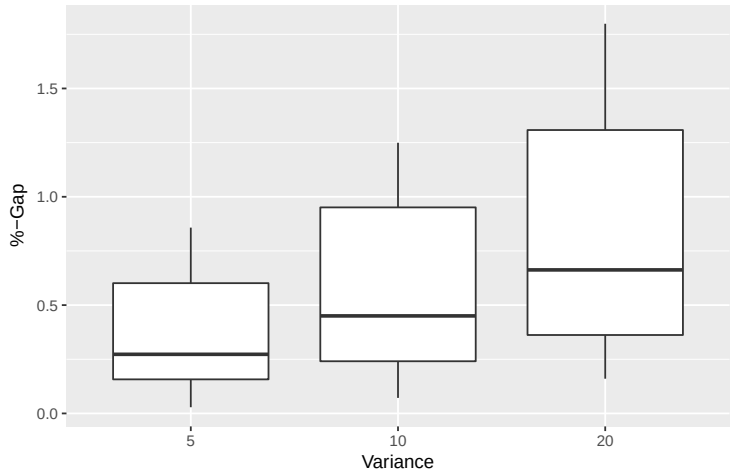


Figure 3.3. Comparison of gaps between stochastic and deterministic solutions.

Chapter 4

Derivative-free methods for mixed-integer nonsmooth constrained optimization problems

In this chapter, new linesearch-based methods for mixed-integer nonsmooth constrained optimization problems are proposed. First-order information on the problem functions is assumed to be unavailable. First, a general framework for mixed-integer bound constrained problems is described. Then, an exact penalty approach is used to tackle the presence of nonlinear (possibly nonsmooth) constraints. The global convergence properties toward stationary points are analyzed for all the algorithms proposed and results of a numerical experience on a set of test mixed-integer bound constrained problems are reported.

4.1 The mixed-integer nonsmooth constrained optimization problem

This chapter deals with the following mixed-integer nonlinearly constrained problem

$$\begin{array}{ll} \min & f(x) \\ s.t. & g(x) \leq 0, \\ & l \leq x \leq u, \\ & x_i \in \mathbb{R} \ \text{ for all } \ i \in I^c, \\ & x_i \in \mathbb{Z} \ \text{ for all } \ i \in I^z, \end{array} \tag{4.1.1}$$

where $x \in \mathbb{R}^n$, $l, u \in \mathbb{R}^n$, and $I^c \cup I^z = \{1, \ldots, n\}$, with $I^c \cap I^z = \emptyset$. We assume that $l_i < u_i$ for all $i \in I^c \cup I^z$ and $l_i, u_i \in \mathbb{Z}$ for all $i \in I^z$. Moreover, the functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$, which may be nondifferentiable, are supposed to be Lipschitz continuous with respect to x_i, with $i \in I^c$. We define the sets

$$X := \{x \in \mathbb{R}^n : l \leq x \leq u\}, \quad \mathcal{F} := \{x \in \mathbb{R}^n : g(x) \leq 0\},$$
$$\mathcal{Z} := \{x \in \mathbb{R}^n : x_i \in \mathbb{Z} \text{ with } i \in I^z\},$$

and we assume throughout the chapter that X is a compact set. Hence, l_i and u_i cannot be infinite. After defining the previous sets, Problem (4.1.1) can be reformulated as follows

$$\begin{aligned} &\min f(x) \\ &s.t. \quad x \in \mathcal{F} \cap \mathcal{Z} \cap X. \end{aligned} \tag{4.1.2}$$

The derivative-free linesearch-type algorithms proposed in this chapter extend the strategies successfully tested in [75] and [171] for continuous and integer problems, respectively, to the mixed-integer case. In particular, globally convergent algorithms are developed in [75] by using a projected linesearch procedure and a dense sequence of directions and in [171] by adopting a set of primitive directions. In this chapter, these strategies are here combined into several algorithms that are provided with global convergence properties towards stationary points of the mixed-integer problems considered. Specifically, different local searches are adopted to tackle the continuous and integer variables separately. On the one hand, a dense sequence of search directions is used to explore the continuous variables in order to effectively detect descent directions, whose cone can be arbitrarily narrow in the nonsmooth case. On the other hand, a set of primitive discrete directions is adopted to enable a thorough exploration of the integer lattice in order to escape points that could not be further improved by using a predetermined set of directions. The characteristics of the set of primitive directions, whether it contains all the feasible directions or not, and the type of polling, whether complete or opportunistic, give rise to algorithms that differ in terms of computational cost and convergence properties.

The presence of general nonlinear constraints is handled by using an exact penalty approach. Since only the violation of such constraints is included in the penalty function, the algorithms developed for bound constrained problems can be used to minimize the penalized problem, which is proved to be equivalent to the original one under reasonable conditions. It is important to point out that the strategy of penalizing only the general nonlinear constraints have been successfully adopted in many papers from the literature related to derivative-free optimization [164, 149, 172, 112, 173].

This chapter is organized as follows. Section 4.2 reports some definitions and preliminary results. In Section 4.3, the four algorithms proposed for mixed-integer problems with bound constraints are described and their convergence properties are analyzed. The same type of analysis is reported in Section 4.4 for the algorithm addressing mixed-integer problems with general nonlinear constraints. Section 4.5 describes the results of extensive numerical experiments performed for the bound constrained case.

4.2 Notation and preliminary results

Given a vector $v \in \mathbb{R}^n$, we introduce the subvectors $v_c \in \mathbb{R}^{|I^c|}$ and $v_z \in \mathbb{R}^{|I^z|}$, given by

$$v_c = [v_i]_{i \in I^c} \quad \text{and} \quad v_z = [v_i]_{i \in I^z},$$

where v_i denotes the i-th component of v. When a vector is an element of an infinite sequence of vectors $\{v_k\}$, the i-th component will be denoted as $(v_k)_i$, in order to avoid possible ambiguities. Moreover, throughout the chapter we denote by $\|\cdot\|$ the Euclidean norm.

The search directions considered in the algorithms proposed in the next sections have either a null continuous subvector or a null discrete subvector, meaning that we do not consider directions that update both continuous and discrete variables

simultaneously. First we report some definitions to define *primitive vectors*, which are used to characterize the subvectors of the search directions related to the discrete variables. Then we move on to the properties of the subvectors related to the continuous variables.

Definition 4.2.1 (Divisor) *Given two integers a and b, we say that a is divisor of b if an integer c exists such that $b = ca$.*

Definition 4.2.2 (Greatest common divisor) *Given $v \in \mathbb{Z}^n$, the greatest common divisor of its components $\{v_1, \dots, v_n\}$, denoted as $GCD(v_1, \dots, v_n)$, is a nonnegative integer d such that d is divisor of $\{v_1, \dots, v_n\}$ (i.e., it is a common divisor) and all other common divisors of $\{v_1, \dots, v_n\}$ are divisors of d.*

Definition 4.2.3 (Primitive vector) *A vector $v \in \mathbb{Z}^n$ is called primitive if $GCD(v_1, \dots, v_n) = 1$.*

Since the objective and constraint functions of the problem considered are assumed to be nonsmooth, proving convergence to a stationary point requires particular subsequences of the continuous subvectors of the search directions to be provided with the density property. In fact, since the feasible descent directions can form an arbitrarily narrow cone (see, e.g., [20] and [5]), a finite number of search directions is not sufficient. Denoted the unit sphere with center in the origin as $S(0,1) = \{s \in \mathbb{R}^n \ : \ \|s_c\| = 1 \text{ and } \|s_z\| = 0\}$, we extend to the mixed-integer case the definition of a dense subsequence of directions.

Definition 4.2.4 (Dense subsequence) *Let K be an infinite subset of indices (possibly $K = \{0, 1, \dots\}$). The subsequence of normalized directions $\{s_k\}_K$, with $(s_k)_i = 0$ for all $i \in I^z$, is dense in the unit sphere $S(0,1)$ if for any $\bar{s} \in S(0,1)$ and for any $\epsilon > 0$ an index $k \in K$ exists such that $\|s_k - \bar{s}\| \leq \epsilon$.*

To provide necessary optimality conditions for Problem (4.1.2), we extend to the mixed-integer case the definition of generalized directional derivative, which is also called Clarke directional derivative, given in [49]. We also recall the definition of generalized gradient.

Definition 4.2.5 (Generalized direc. derivative and generalized gradient) *Let $h : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz continuous function near $x \in \mathbb{R}^n$ with respect to its continuous variables x_c. The generalized directional derivative of h at x in the direction $s \in \mathbb{R}^n$, with $s_i = 0$ for $i \in I^z$, is*

$$h_c^{Cl}(x; s) = \limsup_{y_c \to x_c, y_z = x_z, t \downarrow 0} \frac{h(y + ts) - h(y)}{t}. \tag{4.2.1}$$

The generalized gradient of h at x is

$$\partial_c h(x) = \{v \in \mathbb{R}^{|I^c|} : h_c^{Cl}(x; s) \geq s^T v \ \text{ for all } s \in \mathbb{R}^n, \ \text{with } s_i = 0 \text{ for } i \in I^z\}.$$

Moreover, let us denote the orthogonal projection over the set X as $[x]_{[l,u]} = \max\{l, \min\{u, x\}\}$ and the interior of a set $\mathcal{C}$ as $\overset{\circ}{\mathcal{C}}$.

4.2.1 The bound constrained case

As a special case of Problem (4.1.1), we can consider the following bound constrained problem

$$\begin{array}{ll} \min & f(x) \\ s.t. & l \leq x \leq u, \\ & x_i \in \mathbb{R} \ \text{ for all } \ i \in I^c, \\ & x_i \in \mathbb{Z} \ \text{ for all } \ i \in I^z, \end{array}$$

which can be reformulated as follows

$$\begin{array}{ll} \min & f(x) \\ s.t. & x \in X \cap \mathcal{Z}. \end{array} \tag{4.2.2}$$

The next definitions are related to directions that are feasible with respect to $X \cap \mathcal{Z}$.

Definition 4.2.6 (Set of feasible primitive discrete directions) *Given a point $x \in X \cap \mathcal{Z}$,*

$$\begin{array}{rcl} D^z(x) & = & \{d \in \mathbb{Z}^n : d_z \text{ is a primitive vector, } d_i = 0 \text{ for all } i \in I^c, \text{ and} \\ & & x + d \in X \cap \mathcal{Z}\} \end{array}$$

is the set of feasible primitive discrete directions at x with respect to $X \cap \mathcal{Z}$.

Definition 4.2.7 (Union of the sets of feasible primitive discr. directions)

$$\bar{D} \ = \ \bigcup_{x \in X \cap \mathcal{Z}} D^z(x)$$

is the union of the sets of feasible primitive discrete directions with respect to $X \cap \mathcal{Z}$.

Proposition 4.2.8 *The union of the sets of feasible primitive discrete directions $\bar{D}$ has a finite number of elements.*

Proof Given $d \in \bar{D}$, it follows that $d \in D^z(x)$ for some $x \in X \cap \mathcal{Z}$. By the definition of $D^z(x)$, we have that $d_i = 0$ for all $i \in I_c$ and d_z is a primitive vector. Hence, by considering the boundedness of X, for all $x \in X \cap \mathcal{Z}$ the number of subvectors d_z, $d \in D^z(x)$, is finite. By the boundedness of X and by $d_i = 0 \in I_c$, it follows that the number of distinct x_z is finite and that the number of distinct $D^z(x)$, with $x \in X \cap \mathcal{Z}$, is finite as well. Therefore, the union of subvectors d_z from a finite number of distinct sets $D^z(x)$, with $x \in X \cap \mathcal{Z}$, has a finite number of elements. □

The cone of feasible continuous directions is defined according to the following definition.

Definition 4.2.9 (Cone of feasible continuous directions) *Given a point $x \in X \cap \mathcal{Z}$, the set*

$$\begin{array}{rcl} D^c(x) & = & \{s \in \mathbb{R}^n : s_i = 0 \quad \textit{for all} \quad i \in I^z, \\ & & \quad\ s_i \geq 0 \quad \textit{for all} \quad i \in I^c \quad \textit{and} \quad x_i = l_i, \\ & & \quad\ s_i \leq 0 \quad \textit{for all} \quad i \in I^c \quad \textit{and} \quad x_i = u_i, \\ & & \quad\ s_i \in \mathbb{R} \quad \textit{for all} \quad i \in I^c \quad \textit{and} \quad l_i < x_i < u_i\} \end{array}$$

is the cone of feasible continuous directions at x with respect to $X \cap \mathcal{Z}$.

Now we report a technical proposition whose proof can be derived by minor modifications of [165, Proposition 2.3].

Proposition 4.2.10 *Let $\{x_k\} \subset X \cap \mathcal{Z}$ for all k, and $\{x_k\} \to \bar{x} \in X \cap \mathcal{Z}$ for $k \to \infty$. Then, for k sufficiently large,*

$$D^c(\bar{x}) \subseteq D^c(x_k).$$

Moreover, we introduce two definitions of neighborhood with respect to discrete variables and we recall the definition of neighborhood with respect to continuous variables.

Definition 4.2.11 (Discrete neighborhood) *Given a point $\bar{x} \in X \cap \mathcal{Z}$, the discrete neighborhood of $\bar{x}$ is*

$$\mathcal{B}^z(\bar{x}) = \{x \in X \cap \mathcal{Z} : \quad x = \bar{x} + d, \quad \textit{with} \quad d \in D^z(\bar{x})\}.$$

Definition 4.2.12 (Weak discrete neighborhood) *Given a point $\bar{x} \in X \cap \mathcal{Z}$, the weak discrete neighborhood of $\bar{x}$ is*

$$\mathcal{B}^z_w(\bar{x}; \tilde{D}^z) = \{x \in X \cap \mathcal{Z} : \quad x = \bar{x} + d, \quad \textit{with} \quad d \in \tilde{D}^z\}$$

with $\tilde{D}^z \subset D^z(\bar{x})$.

Definition 4.2.13 (Continuous neighborhood) *Given a point $\bar{x} \in X \cap \mathcal{Z}$ and a scalar ρ, the continuous neighborhood of $\bar{x}$ is*

$$\mathcal{B}^c(\bar{x}; \rho) \quad = \quad \{x \in X : x_z = \bar{x}_z \textit{ and } \|x_c - \bar{x}_c\| \leq \rho\}.$$

Two definitions of local minimum points are given below.

Definition 4.2.14 (Local minimum point) *A point $x^* \in X \cap \mathcal{Z}$ is a local minimum point of Problem (4.2.2) if, for some $\epsilon > 0$,*

$$f(x^*) \leq f(x) \quad \textit{for all } x \in \mathcal{B}^c(x^*; \epsilon), \tag{4.2.3}$$

$$f(x^*) \leq f(x) \quad \textit{for all } x \in \mathcal{B}^z(x^*). \tag{4.2.4}$$

Definition 4.2.15 (Weak local minimum point) *A point $x^* \in X \cap \mathcal{Z}$ is a weak local minimum point of Problem (4.2.2) if, for some $\epsilon > 0$ and some $\tilde{D}^z \subset D^z(x^*)$,*

$$\begin{aligned} &f(x^*) \leq f(x) \quad \textit{for all } x \in \mathcal{B}^c(x^*; \epsilon), \\ &f(x^*) \leq f(x) \quad \textit{for all } x \in \mathcal{B}^z_w(x^*; \tilde{D}^z). \end{aligned} \tag{4.2.5}$$

Now we extend to the mixed-integer case the definition of Clarke-Jahn generalized directional derivative given in [123, Section 3.5]. As opposed to the Clarke directional derivative, in this definition the limit superior is considered only for points y and $y + ts$ in $X \cap \mathcal{Z}$, thus requiring stronger assumptions.

Definition 4.2.16 (Clarke-Jahn generalized directional derivative) *Given a point $x \in X \cap \mathcal{Z}$ with continuous subvector x_c, the Clarke-Jahn generalized directional derivative of function f along direction $s \in D^c(x)$ is given by*

$$f^\circ_c(x; s) = \limsup_{\substack{y_c \to x_c, y_z = x_z, y \in X \cap \mathcal{Z} \\ t \downarrow 0, y + ts \in X \cap \mathcal{Z}}} \frac{f(y + ts) - f(y)}{t}. \tag{4.2.6}$$

We now report a proposition that provides necessary optimality conditions. A similar result can be stated for weak local minimum points as well.

Proposition 4.2.17 *Let $x^* \in X \cap \mathcal{Z}$ be a local minimum point of Problem* (4.2.2)*. Then*

$$f_c^\circ(x^*; s) \geq 0 \qquad \text{for all } s \in D^c(x^*), \tag{4.2.7}$$
$$f(x^*) \leq f(x) \qquad \text{for all } x \in \mathcal{B}^z(x^*). \tag{4.2.8}$$

Proof Since relation (4.2.8) is true by the definition of local minimum given in Definition 4.2.14, it remains to prove relation (4.2.7). In particular, by the definition of Clarke-Jahn directional derivative, we want to prove that

$$f_c^\circ(x^*; s) \geq 0 \text{ for all } s \in D^c(x^*).$$

Proceeding by contradiction, we assume that a direction $\bar{s} \in D^c(x^*)$ exists such that

$$f_c^\circ(x^*; \bar{s}) < 0.$$

Then, for every $y \in X \cap \mathcal{Z}$ with y_c sufficiently close to x_c^* and $y_z = x_z^*$, and for every $t > 0$ sufficiently close to 0, we would have that $y + t\bar{s} \in X \cap \mathcal{Z}$ and $f(y + t\bar{s}) - f(y) < 0$. In particular, setting $y_c = x_c^*$ and $y_z = x_z^*$ yields $f(x^* + t\bar{s}) - f(x^*) < 0$, which contradicts the fact that x^* is a local minimizer of f. □

Proposition 4.2.18 *Let $x^* \in X \cap \mathcal{Z}$ be a weak local minimum point of Problem* (4.2.2)*. Then*

$$f_c^\circ(x^*; s) \geq 0 \qquad \text{for all } s \in D^c(x^*), \tag{4.2.9}$$
$$f(x^*) \leq f(x) \qquad \text{for all } x \in \mathcal{B}_w^z(x^*; \tilde{D}^z), \tag{4.2.10}$$

for some $\tilde{D}^z \subset D^z(x^*)$.

With reference to Problem (4.2.2), now we introduce the definitions of stationary point and weak stationary point based on the Clarke-Jahn generalized directional derivative. Then, we give the definition of stationary point based on the Clarke directional derivative and we prove a corollary stating that a stationary point according to the Clarke-Jahn definition is a stationary point according to the Clarke definition.

Definition 4.2.19 (Stationary point) *A point $x^* \in X \cap \mathcal{Z}$ is a stationary point of Problem (4.2.2) when it satisfies (4.2.7) and (4.2.8).*

Definition 4.2.20 (Weak stationary point) *A point $x^* \in X \cap \mathcal{Z}$ is a weak stationary point of Problem (4.2.2) when it satisfies (4.2.9) and (4.2.10).*

Definition 4.2.21 (Clarke stationary point) *A point $x^* \in X \cap \mathcal{Z}$ is a Clarke stationary point of Problem (4.2.2) when it satisfies*

$$f_c^{Cl}(x^*; s) \geq 0 \qquad \text{for all } s \in D^c(x^*), \tag{4.2.11}$$
$$f(x^*) \leq f(x) \qquad \text{for all } x \in \mathcal{B}^z(x^*). \tag{4.2.12}$$

Corollary 4.2.22 *Let $x^* \in X \cap \mathcal{Z}$ be a stationary point for Problem (4.2.2). Then x^* is a Clarke stationary point.*

Proof The proof easily follows by Proposition (4.2.17) and by the fact that the subsequences of feasible points for Problem (4.2.2) considered in Definition 4.2.16 satisfy also the more general conditions of the limit superior in Definition 4.2.5. □

4.2.2 The nonsmooth nonlinearly constrained case

Now we turn our attention to the general case defined through Problem (4.1.2). A local minimum point for this problem is defined as follows.

Definition 4.2.23 (Local minimum point) *A point $x^\star \in \mathcal{F} \cap \mathcal{Z} \cap X$ is a local minimum point of Problem (4.1.2) if, for some $\epsilon > 0$,*

$$\begin{aligned} &f(x^\star) \leq f(x) \quad \text{for all } x \in \mathcal{B}^c(x^\star;\epsilon) \cap \mathcal{F}, \\ &f(x^\star) \leq f(x) \quad \text{for all } x \in \mathcal{B}^z(x^\star) \cap \mathcal{F}. \end{aligned} \tag{4.2.13}$$

In order to prove some of the next results, we extend to the mixed-integer case the definition of Clarke stationary point for unconstrained problems.

Definition 4.2.24 (Clarke stationarity for unconstrained problems) *Given the unconstrained problem*

$$\begin{aligned} \min \quad & f(x) \\ s.t. \quad & x_i \in \mathbb{R} \ \text{ for all } \ i \in I^c, \\ & x_i \in \mathbb{Z} \ \text{ for all } \ i \in I^z, \end{aligned} \tag{4.2.14}$$

a point $\bar{x}$ is a Clarke stationary point if $0 \in \partial_c f(\bar{x})$ and $f(x^\star) \leq f(x)$ for all x such that $x_i \in \mathbb{R}$ for all $i \in I^c$ and $x_i \in \mathbb{Z}$ for all $i \in I^z$.

Proposition 4.2.25 (Fritz John Optimality Conditions) *Let $x^\star \in \mathcal{F} \cap \mathcal{Z} \cap X$ be a local minimum point of Problem (4.1.2). Then, multipliers $\lambda_0^\star, \lambda_1^\star, \ldots, \lambda_m^\star \in \mathbb{R}$ not all zero exist, with*

$$\lambda_0^\star \geq 0, \quad \lambda_i^\star \geq 0 \quad \text{and} \quad \lambda_i^\star g_i(x^\star) = 0 \ \text{ for all } \ i \in \{1, \ldots, m\},$$

such that for every $s \in D^c(x^\star)$

$$\max \left\{ \xi^\top s \, : \, \xi \in \partial_c f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial_c g_i(x^\star) \right\} \geq 0 \tag{4.2.15}$$

and

$$f(x^\star) \leq f(x) \text{ for all } x \in \mathcal{B}^z(x^\star) \cap \mathcal{F}. \tag{4.2.16}$$

Proof Given a scalar ρ, let us define the continuous neighborhood of $x^\star$ as

$$\hat{\mathcal{B}}^c(x^\star;\rho) \ = \ \left\{ x \in \mathbb{R}^n : x_z = x_z^\star, \ \|x_c - x_c^\star\|_2 \leq \rho \right\}.$$

Since relation (4.2.16) is true by the definition of local minimum point given in Definition 4.2.23, we have to prove relation (4.2.15). The local optimality of $x^\star$ with respect to the continuous variables implies that a constant $\epsilon > 0$ exists such that

$$f(x^\star) \leq f(x) \quad \text{for all } x \in \hat{\mathcal{B}}^c(x^\star;\epsilon) \cap \mathcal{F} \cap X. \tag{4.2.17}$$

Let us define the following functional

$$\begin{aligned} \Phi(x) \ = \ \max \Big\{ & f(x) - f(x^\star), g_1(x), \ldots, g_m(x), \\ & (l_1 - x_1), \ldots, (l_n - x_n), (x_1 - u_1), \ldots, (x_n - u_n) \Big\}. \end{aligned}$$

It follows that $\Phi(x) \geq 0$ for all $x \in \hat{\mathcal{B}}^c(x^\star;\rho)$. In fact, assume by contradiction that $\hat{x} \in \hat{\mathcal{B}}^c(x^\star;\rho)$ exists such that $\Phi(\hat{x}) < 0$. This implies that $g(\hat{x}) < 0$ and $l < \hat{x} < u$, i.e. $\hat{x} \in \mathcal{F} \cap \mathcal{Z} \cap \mathcal{X}$, and that $f(\hat{x}) < f(x^\star)$. The previous relation contradicts that $x^\star$ is a local minimum point.

Since $x^\star \in \mathcal{F} \cap \mathcal{Z} \cap \mathcal{X}$ and $\Phi(x^\star) = 0$, $x^\star$ is a local minimum point of $\Phi(x)$ with respect to the continuous variables. Hence, by Definition 4.2.24 we have that

$$0 \in \partial_c \Phi(x^\star).$$

Considering [49, Proposition 2.3.12], it follows that

$$0 \in \tilde{\lambda}_0 \partial_c f(x^\star) + \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i \partial_c g_i(x^\star) - \sum_{j \in I_l(x^\star)} \tilde{\mu}_j e_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h e_h, \tag{4.2.18}$$

where:

- $I_0(x^\star) = \{i : g_i(x^\star) = 0\}$, $I_l(x^\star) = \{j : x_j^\star = l_j\}$, and $I_u(x^\star) = \{h : x_h^\star = u_h\}$;
- $\tilde{\lambda}_0 \geq 0$ and $\tilde{\lambda}_i \geq 0$ for all $i \in I_0(x^\star)$, $\tilde{\mu}_j \geq 0$ for all $j \in I_l(x^\star)$, and $\tilde{\mu}_h \geq 0$ for all $h \in I_u(x^\star)$, with

$$\tilde{\lambda}_0 + \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i + \sum_{j \in I_l(x^\star)} \tilde{\mu}_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h = 1. \tag{4.2.19}$$

Now, the linear independence of the set $\{e_j : \ j \in I_l(x^\star)\} \cup \{e_h : \ h \in I_u(x^\star)\}$ implies that

$$\tilde{\lambda}_0 + \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i \neq 0,$$

otherwise from relation (4.2.18) we would have

$$0 = - \sum_{j \in I_l(x^\star)} \tilde{\mu}_j e_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h e_h,$$

which is true only if $\tilde{\mu}_j$ and $\tilde{\mu}_h$ are all zero, resulting in a contradiction with (4.2.19).

We can now divide (4.2.18) by $\Lambda := \tilde{\lambda}_0 + \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i \neq 0$. Calling

$$\begin{aligned}
\lambda_0 &:= \frac{\tilde{\lambda}_0}{\Lambda}, \\
\lambda_i &:= \frac{\tilde{\lambda}_i}{\Lambda}, \ \forall\, i \in I_0(x^\star), \\
\mu_j &:= \frac{\tilde{\mu}_j}{\Lambda}, \ \forall\, j \in I_l(x^\star), \\
\mu_h &:= \frac{\tilde{\mu}_h}{\Lambda}, \ \forall\, h \in I_u(x^\star),
\end{aligned}$$

we have

$$0 \in \lambda_0 \partial_c f(x^\star) + \sum_{i \in I_0(x^\star)} \lambda_i \partial_c g_i(x^\star) - \sum_{j \in I_l(x^\star)} \mu_j e_j + \sum_{h \in I_u(x^\star)} \mu_h e_h, \tag{4.2.20}$$

where $\lambda_0 \geq 0$ and $\lambda_i \geq 0$ for all $i \in I_0(x^\star)$, $\mu_j \geq 0$ for all $j \in I_l(x^\star)$, and $\mu_h \geq 0$ for all $h \in I_u(x^\star)$, with

$$\lambda_0 + \sum_{i \in I_0(x^\star)} \lambda_i = 1. \tag{4.2.21}$$

From (4.2.20), we have that ξ exists such that

$$\xi \in \lambda_0 \partial_c f(x^\star) + \sum_{i \in I_0(x^\star)} \lambda_i \partial_c g_i(x^\star)$$

and

$$0 = \xi \ - \sum_{j \in I_l(x^\star)} \mu_j e_j + \sum_{h \in I_u(x^\star)} \mu_h e_h.$$

Then, from Definition 4.2.9 of $D^c(x^\star)$, we have

$$\xi^\top s \geq 0 \tag{4.2.22}$$

for all $s \in D^c(x^\star)$. Thus, for all $s \in D^c(x^\star)$, we can write

$$\max \left\{ \xi^\top s \ : \ \xi \in \lambda_0 \partial_c f(x^\star) + \sum_{i \in I_0(x^\star)} \lambda_i \partial_c g_i(x^\star) \right\} \geq 0,$$

with $\lambda_0 \geq 0$ and $\lambda_i \geq 0$ for all $i \in I_0(x^\star)$ and, by (4.2.21), not all zero. Choosing $\lambda_i = 0$ for all $i \notin I_0(x^\star)$ allows us to prove the proposition. □

Taking into account the proof given above and, in particular, the inequality (4.2.22), the following result can be derived.

Lemma 4.2.26 *Let $x^\star \in \mathcal{F} \cap \mathcal{Z} \cap X$ be a local minimum point of Problem* (4.1.2)*. Then, multipliers $\lambda_0^\star, \lambda_1^\star, \ldots, \lambda_m^\star \in \mathbb{R}$ not all zero, with*

$$\lambda_0^\star \geq 0, \quad \lambda_i^\star \geq 0 \quad \textit{and} \quad \lambda_i^\star g_i(x^\star) = 0 \ \ \textit{for all } i \in \{1, \ldots, m\},$$

and a vector $\bar{\xi} \in \lambda_0^\star \partial_c f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial_c g_i(x^\star)$ exist such that

$$\bar{\xi}^\top s \geq 0$$

for every $s \in D^c(x^\star)$.

Proposition 4.2.27 (KKT Necessary Optimality Conditions) *Let $x^\star \in \mathcal{F} \cap \mathcal{Z} \cap X$ be a local minimum point of Problem (4.1.2) and assume that a direction $s \in D^c(x^\star)$ exists such that, for all $i \in \{1, \ldots, m : g_i(x^\star) = 0\}$,*

$$(\xi^{g_i})^\top s < 0, \quad \forall \ \xi^{g_i} \in \partial_c g_i(x^\star). \tag{4.2.23}$$

Then there exists a vector $\lambda^\star \in \mathbb{R}^m$ such that, for every $s \in D^c(x^\star)$

$$\max \left\{ \xi^\top s \ : \ \xi \in \partial_c f(x^\star) + \sum_{i=1}^{m} \lambda_i^\star \partial_c g_i(x^\star) \right\} \geq 0, \tag{4.2.24}$$

$$(\lambda^\star)^T g(x^\star) = 0 \ \ \textit{and} \ \ \lambda^\star \geq 0, \tag{4.2.25}$$

and

$$f(x^\star) \leq f(x) \ \ \textit{for all} \ \ x \in \mathcal{B}^z(x^\star) \cap \mathcal{F}. \tag{4.2.26}$$

Proof Since relation (4.2.26) is true by the definition of local minimum point given in Definition 4.2.23, we have to prove the remaining relations. In particular, by inequality (4.2.23), we know that $\bar{s} \in \mathbb{R}^n$ with $\bar{s}_i = 0$ for $i \in I^z$ exists such that

$$\begin{cases} (\xi^{g_i})^\top \bar{s} < 0 \text{ for all } \xi^{g_i} \in \partial_c g_i(x^\star) \text{ and for all } i \in I_0(x^\star), \\ -e_j^\top \bar{s} < 0 \text{ for all } j \in I_l(x^\star), \\ e_h^\top \bar{s} < 0 \text{ for all } h \in I_u(x^\star), \end{cases}$$

where $I_0(x^\star) = \{i : g_i(x^\star) = 0\}$, $I_l(x^\star) = \{j : x_j^\star = l_j\}$, and $I_u(x^\star) = \{h : x_h^\star = u_h\}$.

By the alternative theorem in [214, Theorem 2.3.4] and [122], there cannot exist multipliers $\tilde{\lambda}_i \geq 0$ for all $i \in I_0(x^\star)$, $\tilde{\mu}_j \geq 0$ for all $j \in I_l(x^\star)$, and $\tilde{\mu}_h \geq 0$ for all $h \in I_u(x^\star)$, with

$$\sum_{i \in I_0(x^\star)} \tilde{\lambda}_i + \sum_{j \in I_l(x^\star)} \tilde{\mu}_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h = 1, \tag{4.2.27}$$

such that

$$0 \in \sum_{i \in I_0(x^\star)} \tilde{\lambda}_i \partial g_i(x^\star) - \sum_{j \in I_l(x^\star)} \tilde{\mu}_j e_j + \sum_{h \in I_u(x^\star)} \tilde{\mu}_h e_h. \tag{4.2.28}$$

Moreover, by the Fritz John optimality conditions stated in Proposition 4.2.25, there exist multipliers $\lambda_0^\star \geq 0$ and $\lambda_i^\star \geq 0$, with $\lambda_i^\star = 0$ when $g_i(x^\star) < 0$, such that relation (4.2.24) holds.

Now, proceeding by contradiction, we assume that $\lambda_0^\star = 0$. This implies that the multipliers $\lambda_i^\star$, with $i \in \{1, \ldots, m\}$, cannot be all zero, otherwise the corresponding hypothesis stated in Proposition 4.2.25 would be contradicted. Therefore, we can define new multipliers

$$\bar{\lambda}_i = \lambda_i^\star / \beta, \qquad i \in I_0(x^\star),$$

where $\beta = \sum_{i \in I_0(x^\star)} \lambda_i^\star > 0$. Thus, it follows that

$$\bar{\lambda}_i \geq 0 \quad \text{and} \quad \sum_{i \in I_0(x^\star)} \bar{\lambda}_i = 1. \tag{4.2.29}$$

By Lemma 4.2.26, there exists a vector

$$\bar{\xi} \in \sum_{i=1}^{m} \bar{\lambda}_i \partial_c g_i(x^\star) \tag{4.2.30}$$

such that

$$\bar{\xi}^\top s \geq 0, \tag{4.2.31}$$

for every $s \in D^c(x^\star)$. Furthermore, let us consider the following system

$$\begin{cases} -\bar{\xi}^\top s > 0, \\ -e_j^\top s \leq 0 \text{ for all } j \in I_l(x^\star), \\ e_h^\top s \leq 0 \text{ for all } h \in I_u(x^\star), \end{cases}$$

where the last two sets of constraints imply that $s \in D^c(x^\star)$. By inequality (4.2.31), it follows that this system does not have solutions. Then, Farkas theorem (see, e.g., [181, Chapter 2]) implies that scalars not all zero $\alpha_j \geq 0$, with $j \in I_l(x^\star)$, and $\alpha_h \geq 0$, with $h \in I_u(x^\star)$, exist such that

$$-\bar{\xi} = -\sum_{j \in I_l(x^\star)} \alpha_j e_j + \sum_{h \in I_u(x^\star)} \alpha_h e_h. \tag{4.2.32}$$

Moreover, by the conditions in (4.2.29) it follows that

$$\sigma := \sum_{i \in I_0(x^\star)} \bar{\lambda}_i + \sum_{j \in I_l(x^\star)} \alpha_j + \sum_{h \in I_u(x^\star)} \alpha_h \geq 1.$$

Then, defining

$$\begin{aligned} \hat{\lambda}_i &:= \frac{\bar{\lambda}_i}{\sigma}, \qquad \forall\, i \in I_0(x^\star), \\ \hat{\alpha}_j &:= \frac{\alpha_j}{\sigma}, \qquad \forall\, j \in I_l(x^\star), \\ \hat{\alpha}_h &:= \frac{\alpha_h}{\sigma}, \qquad \forall\, h \in I_u(x^\star), \end{aligned}$$

we have that

$$\sum_{i \in I_0(x^\star)} \hat{\lambda}_i + \sum_{j \in I_l(x^\star)} \hat{\alpha}_j + \sum_{h \in I_u(x^\star)} \hat{\alpha}_h = 1. \tag{4.2.33}$$

Relation (4.2.30) and equations (4.2.32) and (4.2.33) contradict the fact that multipliers satisfying the conditions (4.2.27) and (4.2.28) cannot exist. □

As regards the stationarity conditions for Problem (4.1.2), we report the following definition by taking into account the above results.

Definition 4.2.28 (Stationary point) *A point $x^* \in \mathcal{F} \cap \mathcal{Z} \cap X$ is a stationary point of Problem (4.1.2) if there exists a vector $\lambda^\star \in \mathbb{R}^m$ such that, for every $s \in D^c(x^\star)$, the pair (x^*, λ^*) satisfies* (4.2.24), (4.2.25), (4.2.26).

4.3 Algorithms for bound constrained problems

In this section we propose different algorithms for solving the mixed-integer bound constrained problem defined through Problem (4.2.2) and we analyze their convergence properties. In particular, we propose four algorithms that differ in terms of convergence guarantees and computational burden. As expected, the stronger convergence results correspond to the most computationally expensive algorithms.

In all the algorithms described, the minimization over the continuous and discrete variables is performed by two local searches. The local search used to handle the minimization over the continuous variables is the same for each algorithm, while different types of local searches are adopted to tackle the discrete variables. The two line-search algorithms used within the local searches to explore the feasible search directions are similar to the procedures proposed in [177, 169, 170, 75]. In particular, the *Projected Continuous Search* described in Algorithm 1 and the *Discrete Search* described in Algorithm 2 are the methods adopted in this work to investigate the directions associated with the continuous and discrete variables, respectively. The idea behind the line searches is to return a positive stepsize α, namely to update the current iterate, whenever a point providing a sufficient reduction of the objective function is found. In Algorithm 1 the sufficient decrease is controlled by α, while in Algorithm 2 the same role is played by a parameter ξ. Once such a point is determined, an expansion step is performed in order to explore if the sufficient reduction may be achieved with a larger stepsize.

Algorithm 1 Projected Continuous Search $(\tilde{\alpha}, w, p; \alpha, \tilde{p})$

Data. $\gamma > 0$, $\delta \in (0, 1)$.

Step 0. Set $\alpha = \tilde{\alpha}$.
Step 1. **If** $f([w + \alpha p]_{[l,u]}) \leq f(w) - \gamma\alpha^2$ **then** set $\tilde{p} = p$ go to Step 4.
Step 2. **If** $f([w - \alpha p]_{[l,u]}) \leq f(w) - \gamma\alpha^2$ **then** set $\tilde{p} = -p$ and go to Step 4.
Step 3. Set $\alpha = 0$, **return** α and $\tilde{p} = p$.
Step 4. Let $\beta = \alpha/\delta$.
Step 5. **If** $f([w + \beta\tilde{p}]_{[l,u]}) > f(w) - \gamma\beta^2$ **return** α, $\tilde{p}$.
Step 6. Set $\alpha = \beta$ and go to Step 4.

Algorithm 2 Discrete Search $(\tilde{\alpha}, w, p, \xi; \alpha)$

Data. $\gamma > 0$.

Step 0. Compute the largest $\bar{\alpha}$ such that $w + \bar{\alpha}p \in X \cap \mathcal{Z}$.
Set $\alpha = \min\{\bar{\alpha}, \tilde{\alpha}\}$.

Step 1. **If** $\alpha > 0$ and $f(w + \alpha p) \leq f(w) - \xi$ **then** go to Step 2.
Else Set $\alpha = 0$, **return** α.

Step 2. Set $\beta = \min\{\ \bar{\alpha}, 2\alpha\}$.

Step 3. **If** $f(w + \beta p) > f(w) - \xi$, **return** α.

Step 4. Set $\alpha = \beta$ and go to Step 2.

4.3.1 An algorithm with simple decrease over the discrete variables

The first algorithm we report (see Algorithm 3 for the algorithmic scheme) shows strong convergence properties, since it is possible to prove that every limit point of the sequence of points yielded by the algorithm is a stationary point. However, achieving this result requires a high computational cost.

The algorithm is divided into two phases. Starting from a point $x_0 \in X \cap \mathcal{Z}$, in Phase 1 the minimization over the continuous variables is performed by using the Projected Continuous Search (**Step 7**). If the line search fails, i.e., $\alpha_k^c = 0$, the tentative step for continuous search directions is reduced (**Step 9**), otherwise the current iterate is updated (**Step 11**). Then, in Phase 2, every feasible primitive discrete direction in the set $D^z(\tilde{x}_k)$, where $\tilde{x}_k$ is the current iterate obtained at the end of Phase 1, is investigated through the Discrete Search (**Step 16**), which aims to find the point that produces the largest reduction in the objective function. We point out that in this case the Discrete Search requires only a simple reduction in the objective function since $\xi = 0$. If the stepsize returned by the line search performed along a given primitive direction is 0, the corresponding tentative step is halved (**Step 19**), otherwise the current iterate is updated (**Step 21**). The directions in D are explored until D becomes empty.

To achieve the strong convergence properties accomplished by this algorithm, it is fundamental that the set D of search directions explored at each iteration is equal

to $D^z(\tilde{x}_k)$ (see **Step 13**). Since the set $D^z(\tilde{x}_k)$ may have a large cardinality, this is a strong requirement which may not be implementable in practice. In light of this issue, Algorithms 5 and 6 are proposed to provide algorithms that may work more efficiently.

The condition at **Step 16** allows us to prove the main convergence result by ensuring that every point in the discrete neighborhood (see Definition 4.2.11) of the optimal point can be reached by exploring a proper direction in the set of feasible primitive discrete directions. Since the convergence result is based on a proof by contradiction, the latter statement ensures that the point leading to the contradiction can be eventually visited by the algorithm, as described in the proof of Proposition 4.3.6.

The following propositions guarantee that the algorithm is well–defined.

Proposition 4.3.1 *The* Projected Continuous Search *cannot infinitely cycle between Step 4 and Step 6.*

Proof We assume by contradiction that in the Projected Continuous Search an infinite monotonically increasing sequence of positive numbers $\{\beta_j\}$ exists such that $\beta_j \to \infty$ for $j \to \infty$ and

$$f([w + \beta_j p]_{[l,u]}) \leq f(w) - \gamma \beta_j^2.$$

Since by the instructions of the procedures we have that $[w + \beta_j p]_{[l,u]} \in X \cap \mathcal{Z}$, the previous relation is in contrast with the compactness of X, by definition of compact set, and with the continuity of function f. These arguments conclude the proof. □

Proposition 4.3.2 *The* Discrete Search *cannot infinitely cycle between Step 2 and Step 4.*

Proof We assume by contradiction that in the Discrete Search an infinite monotonically increasing sequence of positive numbers $\{\beta_j\}$ exists such that $\beta_j \to \infty$ for $j \to \infty$ and

$$f(w + \beta_j p) \leq f(w) - \xi.$$

Since by the instructions of the procedures we have that $w + \beta_j p \in X \cap \mathcal{Z}$, the previous relation is in contrast with the compactness of X, by definition of compact set. This argument concludes the proof. □

In the following proposition, we prove that Algorithm 3 returns stepsizes that eventually go to zero.

Proposition 4.3.3 *Let* $\{\alpha_k^c\}$ *and* $\{\tilde{\alpha}_k^c\}$ *be the sequences yielded by Algorithm 3 for each* s_k*. Then*

$$\lim_{k \to \infty} \max\{\alpha_k^c, \tilde{\alpha}_k^c\} = 0. \tag{4.3.1}$$

Proof The iteration sequence $\{k\}$ can be split into two sets K_1, K_2, with $K_1 \cup K_2 = \{k\}$ and $K_1 \cap K_2 = \emptyset$. For each s_k, we denote by

- K_1 the set of iterations such that $\tilde{\alpha}_{k+1}^c = \alpha_k^c$;
- K_2 the set of iterations such that $\tilde{\alpha}_{k+1}^c = \theta \tilde{\alpha}_k^c$ and $\alpha_k^c = 0$.

Algorithm 3 DFNDFL 1

DATA

Let $x_0 \in X \cap \mathcal{Z}$ and $\theta \in (0,1)$;
let $\{s_k\}$ be a sequence such that $s_k \in D^c(x_0)$ and $\|s_k\| = 1$ for all k;
let $\tilde{\alpha}_0^c = 1$ be the initial stepsize along s_k and let $\alpha_{threshold} > 0$;
let $D = D^z(x_0)$ be the set of the feasible primitive discrete directions at point x_0;
let $\tilde{\alpha}_0^{(d)} = 1$ be the initial stepsizes along $d \in D$.
For $k = 0, 1, \ldots$

PHASE 1 - Explore continuous variables

Compute α_k^c and $\tilde{s}_k$ by the *Projected Continuous Search*$(\tilde{\alpha}_k^c, x_k, s_k; \alpha_k^c, \tilde{s}_k)$.
If $(\alpha_k^c = 0)$ **then**
    $\tilde{\alpha}_{k+1}^c = \theta \tilde{\alpha}_k^c$ and $\tilde{x}_k = x_k$,
else
    $\tilde{\alpha}_{k+1}^c = \alpha_k^c$ and $\tilde{x}_k = [x_k + \alpha_k^c \tilde{s}_k]_{[l,u]}$.
End If

PHASE 2 - Explore discrete variables

Set $y^+ = \tilde{x}_k$, $y = y^+$, and $D = D^z(\tilde{x}_k)$.
While $D \neq \emptyset$ **do**
  Choose $d \in D$, set $D = D \setminus \{d\}$.
  If $\tilde{\alpha}_k^c < \alpha_{threshold}$ **then** Set $\tilde{\alpha}_k^{(d)} = 1$.
  Compute α by the *Discrete Search*$(\tilde{\alpha}_k^{(d)}, y, d, 0; \alpha)$.
  If $\alpha = 0$ **then**
      Set $y^* = y$ and $\tilde{\alpha}_{k+1}^{(d)} = \max\{1, \lfloor \tilde{\alpha}_k^{(d)}/2 \rfloor\}$,
  else
      Set $y^* = y + \alpha d$ and $\tilde{\alpha}_{k+1}^{(d)} = \alpha$.
  End If
  If $f(y^*) < f(y^+)$ **then** $y^+ = y^*$.
End While

PHASE 3 - Update iterates

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(y^+)$.
End For

Since K_1 and K_2 cannot be both finite, the following three cases are considered: (i) K_1 infinite and K_2 finite, (ii) K_1 finite and K_2 infinite, and (iii) both K_1 and K_2 infinite.

By the instructions of the algorithm and the Projected Continuous Search, for $k \in K_1$,

$$f(x_{k+1}) \leq f([x_k + \alpha_k^c \tilde{s}_k]_{[l,u]}) \leq f(x_k) - \gamma(\alpha_k^c)^2. \tag{4.3.2}$$

Hence,

$$f(x_{k+1}) \leq f(x_k) - \gamma(\alpha_k^c)^2. \tag{4.3.3}$$

Since X is compact and f is continuous, the previous relation implies that $\{f(x_k)\}$ tends to a limit f^*. Then, we get from (4.3.3) that

$$\lim_{k\to\infty, k\in K_1} \alpha_k^c = 0, \tag{4.3.4}$$

and, accordingly,

$$\lim_{k\to\infty, k\in K_1} \tilde{\alpha}_k^c = 0. \tag{4.3.5}$$

Relations (4.3.4) and (4.3.5) conclude the proof for case (i).

Regardless of K_1, when K_2 is infinite, we get by definition that $\alpha_k^c = 0$, $k \in K_2$, and, as a trivial implication,

$$\lim_{k\to\infty, k\in K_2} \alpha_k^c = 0. \tag{4.3.6}$$

Moreover, when $k \in K_2$ (whether infinite or finite), by the instructions of the algorithm, it follows

$$\tilde{\alpha}_{k+1}^c = \theta^{k-m_k} \tilde{\alpha}_{m_k}^c, \tag{4.3.7}$$

where $m_k < k$ is the largest integer such that $m_k \in K_1$ (if K_1 is empty, we assume $m_k = 0$).

In case (ii), K_1 finite and K_2 infinite imply $(k - m_k) \to \infty$. This limit along with (4.3.7) and $\theta \in (0, 1)$ gives

$$\lim_{k\to\infty, k\in K_2} \tilde{\alpha}_k^c = 0. \tag{4.3.8}$$

Relations (4.3.6) and (4.3.8) conclude the proof for case (ii).

In case (iii), K_1 infinite implies $m_k \to \infty$. This limit along with (4.3.7) and (4.3.5) leads to (4.3.8). Relations (4.3.4), (4.3.5), (4.3.6) and (4.3.8) conclude the proof for this last case.

□

The previous result is used to prove the next lemma, which in turn is essential to prove the global convergence result with respect to the continuous variables. This lemma states that the asymptotic convergence properties of the sequence $\{s_k\}$ still hold when the projection operator is adopted. Its proof closely resembles the proof in [75, Lemma 2.6].

Lemma 4.3.4 *Let $\{x_k\}$ and $\{s_k\}$ be the sequence of points and the sequence of directions yielded by Algorithm 3, respectively, and $\{\eta_k\}$ be a sequence such that $\eta_k > 0$, for all k. Further, let K be a subset of indices such that*

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}, \tag{4.3.9}$$

$$\lim_{k\to\infty, k\in K} s_k = \bar{s}, \tag{4.3.10}$$

$$\lim_{k\to\infty, k\in K} \eta_k = 0. \tag{4.3.11}$$

with $\bar{x} \in X \cap \mathcal{Z}$ and $\bar{s} \in D^c(\bar{x})$, $\bar{s} \neq 0$. Then,

(i) for all $k \in K$ sufficiently large,

$$[x_k + \eta_k s_k]_{[l,u]} \neq x_k;$$

(ii) the following limit holds

$$\lim_{k \to \infty, k \in K} v_k = \bar{s},$$

where

$$v_k = \frac{[x_k + \eta_k s_k]_{[l,u]} - x_k}{\eta_k}. \tag{4.3.12}$$

Proof Recalling that

$$[x_k + \eta_k s_k]_{[l,u]} = \max\{l, \min\{u, (x_k + \eta_k s_k)\}\},$$

we want to prove that for $k \in K$ sufficiently large we have

$$[x_k + \eta_k s_k]_{[l,u]} \neq x_k. \tag{4.3.13}$$

By contradiction, we assume that for $k \in K$ sufficiently large

$$[x_k + \eta_k s_k]_{[l,u]} = x_k. \tag{4.3.14}$$

Since $\bar{s} \neq 0$ by hypothesis, an index i with $\bar{s}_i \neq 0$ exists. Moreover, one of the following three cases holds:

1) $\bar{x}_i = l_i$ (hence $\bar{s}_i > 0$): we can write

$$([x_k + \eta_k s_k]_{[l,u]})_i = \max\{l_i, (x_k + \eta_k s_k)_i\};$$

from the fact that x_k is feasible and from (4.3.10), for k sufficiently large we have

$$\max\{l_i, (x_k + \eta_k s_k)_i\} > \max\left\{l_i, \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i\right\}.$$

Therefore, by (4.3.11) and $\eta_k > 0$, we get

$$\max\left\{l_i, \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i\right\} = \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i \neq (x_k)_i. \tag{4.3.15}$$

2) $\bar{x}_i = u_i$ (hence $\bar{s}_i < 0$): we can write

$$([x_k + \eta_k s_k]_{[l,u]})_i = \min\{u_i, (x_k + \eta_k s_k)_i\};$$

from the fact that y_k is feasible and from (4.3.10), for k sufficiently large we have

$$\min\{u_i, (x_k + \eta_k s_k)_i\} < \min\left\{u_i, \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i\right\}.$$

Therefore, by (4.3.11) and $\eta_k > 0$, we get

$$\min\left\{u_i, \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i\right\} = \left(x_k + \frac{\eta_k}{2}\bar{s}\right)_i \neq (x_k)_i. \tag{4.3.16}$$

3) $l_i < \bar{x}_i < u_i$ (hence $\bar{s}_i \neq 0$): we can write

$$([x_k + \eta_k s_k]_{[l,u]})_i = (x_k + \eta_k s_k)_i;$$

from the fact that x_k is feasible and from (4.3.10), for k sufficiently large we have

$$(x_k + \eta_k s_k)_i \neq (x_k)_i. \tag{4.3.17}$$

Therefore, (4.3.15), (4.3.16), and (4.3.17) are in contrast with (4.3.14). This argument proves *(i)*.

Recalling definition (4.3.12), notice that the vector v_k is eventually nonzero by (4.3.13). For the $i-$th component of the vector v_k, we can write by its definition that

$$(v_k)_i = \frac{\max\{l_i, \min\{u_i, (x_k + \eta_k s_k)_i\}\} - (x_k)_i}{\eta_k} \tag{4.3.18}$$

$$= \frac{\min\{u_i, \max\{l_i, (x_k + \eta_k s_k)_i\}\} - (x_k)_i}{\eta_k}. \tag{4.3.19}$$

Now we can consider three cases for k sufficiently large and $k \in K$:

1) $\bar{x}_i = l_i$: by (4.3.18) we have

$$(v_k)_i = \frac{\max\{l_i, (x_k + \eta_k s_k)_i\} - (x_k)_i}{\eta_k}.$$

When $\bar{x}_i = l_i$, we have that $\bar{s}_i \geq 0$ and

a) if $\bar{s}_i > 0$, then $(v_k)_i = \max\left\{\frac{l_i - (x_k)_i}{\eta_k}, (s_k)_i\right\} = (s_k)_i$;

b) if $\bar{s}_i = 0$, then

$$\lim_{k\to\infty, k\in K} (v_k)_i = \lim_{k\to\infty, k\in K} \max\left\{\frac{l_i - (x_k)_i}{\eta_k}, (s_k)_i\right\} = 0 = (\bar{s})_i.$$

2) $\bar{x}_i = u_i$: by (4.3.19) we have

$$(v_k)_i = \frac{\min\{u_i, (x_k + \eta_k s_k)_i\} - (x_k)_i}{\eta_k}.$$

When $\bar{x}_i = u_i$, we have that $\bar{s}_i \leq 0$ and

a) if $\bar{s}_i < 0$, then $(v_k)_i = \min\left\{\frac{u_i - (x_k)_i}{\eta_k}, (s_k)_i\right\} = (s_k)_i$;

b) if $\bar{s}_i = 0$, then

$$\lim_{k\to\infty, k\in K} (v_k)_i = \lim_{k\to\infty, k\in K} \min\left\{\frac{u_i - (x_k)_i}{\eta_k}, (s_k)_i\right\} = 0 = (\bar{s})_i.$$

3) $l_i < \bar{x}_i < u_i$: by (4.3.18) or (4.3.19) we have $(v_k)_i = (x_k + \eta_k s_k - x_k)_i/\eta_k = (s_k)_i$.

Since from this argument it follows that $\lim_{k\to\infty,k\in K} v_k = \bar{s}$, *(ii)* is proved. □

The main convergence result with respect to the continuous variables, which states that every limit point of the sequence yielded by Algorithm 3 is a stationary point with respect to the continuous variables, is proved in the next proposition.

Proposition 4.3.5 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 3. Let $\bar{x} \in X \cap \mathcal{Z}$ be any limit point of $\{x_k\}$ and K be the subset of indices such that*

$$\lim_{k\to\infty,k\in K} x_k = \bar{x}.$$

If the subsequence $\{s_k\}_K$, with $(s_k)_i = 0$ for $i \in I^z$, is dense in the unit sphere (see Definition 4.2.4), then $\bar{x}$ satisfies

$$f_c^\circ(\bar{x}; s) \geq 0 \qquad \text{for all } s \in D^c(\bar{x}).$$

Proof Proceeding by contradiction, we assume that a direction $\bar{s} \in D^c(\bar{x}) \cap S(0,1)$ exists such that

$$f_c^\circ(\bar{x}; \bar{s}) < 0. \tag{4.3.20}$$

By the instructions of the Projected Continuous Search, satisfying the condition at Step 1 implies $\alpha_k^c > 0$ and

$$f([x_k + (\alpha_k^c/\delta)s_k]_{[l,u]}) > f(x_k) - \gamma(\alpha_k^c/\delta)^2, \tag{4.3.21}$$

otherwise (i.e., the condition at Step 1 is not satisfied)

$$f([x_k + \tilde{\alpha}_k^c s_k]_{[l,u]}) > f(x_k) - \gamma(\tilde{\alpha}_k^c)^2. \tag{4.3.22}$$

Now, after setting

$$\eta_k = \begin{cases} \alpha_k^c/\delta & \text{if (4.3.21) holds} \\ \tilde{\alpha}_k^c & \text{if (4.3.22) holds,} \end{cases}$$

for every index $k \in K$, let us define v_k as in relation (4.3.12) of Lemma 4.3.4, that is

$$v_k = \frac{[x_k + \eta_k s_k]_{[l,u]} - x_k}{\eta_k}.$$

By the instructions of Algorithm 3 and by the definition of η_k, it follows that $\eta_k > 0$, for all $k \in K$. Moreover, by Proposition 4.3.3,

$$\lim_{k\to\infty} \eta_k = 0. \tag{4.3.23}$$

By Definition 4.2.4, it follows that a subset $\bar{K} \subseteq K$ exists such that

$$\lim_{k\to\infty,k\in\bar{K}} x_k = \bar{x}, \tag{4.3.24}$$

$$\lim_{k\to\infty,k\in\bar{K}} s_k = \bar{s}. \tag{4.3.25}$$

Therefore, (4.3.23), (4.3.24) and (4.3.25) satisfy the assumptions of Lemma 4.3.4. In particular, from point (i) it follows that $v_k \neq 0$ for $k \in \bar{K}$ sufficiently large, and from point (ii) we have

$$\lim_{k\to\infty,k\in K} v_k = \bar{s}.$$

Accordingly, relations (4.3.21) and (4.3.22) can be equivalently expressed as

$$f(x_k + \eta_k v_k) > f(x_k) - \gamma \eta_k^2.$$

Since $\eta_k > 0$, for $k \in \bar{K}$ and sufficiently large we have

$$\frac{f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} > -\gamma \eta_k. \tag{4.3.26}$$

Then we can write

$$\begin{aligned} f_c^\circ(\bar{x}; \bar{s}) = & \limsup_{\substack{(x_k)_c \to \bar{x}_c, (x_k)_z = \bar{x}_z, x_k \in X \cap \mathcal{Z} \\ t \downarrow 0, x_k + t\bar{s} \in X \cap \mathcal{Z}}} \frac{f(x_k + t\bar{s}) - f(x_k)}{t} \geq \\ & \limsup_{k \to \infty, k \in \bar{K}} \frac{f(x_k + \eta_k \bar{s}) - f(x_k)}{\eta_k} = \\ & \limsup_{k \to \infty, k \in \bar{K}} \frac{f(x_k + \eta_k \bar{s}) + f(x_k + \eta_k v_k) - f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} \geq \\ & \limsup_{k \to \infty, k \in \bar{K}} \frac{f(x_k + \eta_k v_k) - f(x_k)}{\eta_k} - L\|\bar{s} - v_k\|, \end{aligned}$$

where L is the Lipschitz constant of f. From the former relation, it follows by (4.3.26) and *(ii)* of Lemma 4.3.4 that

$$f_c^\circ(\bar{x}; \bar{s}) \geq 0,$$

which contradicts (4.3.20) and concludes the proof. □

The next proposition states that every limit point of the sequence yielded by Algorithm 3 is a stationary point also with respect to the discrete variables (see Definition 4.2.19).

Proposition 4.3.6 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 3 and $x^* \in X \cap \mathcal{Z}$ be an accumulation point. Then,*

$$f(x^*) \leq f(\bar{x}),$$

for all $\bar{x} \in \mathcal{B}^z(x^*)$.

Proof Let K be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*. \tag{4.3.27}$$

By the instructions of Algorithm 3, it follows that the sequence $\{f(x_k)\}$ is monotonically nonincreasing. Since X is compact and f is continuous, f is bounded from below and $\{f(x_k)\}$ is convergent to a limit f^*. Then, we have that

$$\lim_{k \to \infty, k \in K} f(x_k) = f(x^*) = f^*.$$

Let us assume by contradiction that a point $\bar{x} \in \mathcal{B}^z(x^*)$ exists such that

$$f(\bar{x}) < f(x^*). \tag{4.3.28}$$

By definition of discrete neighborhood $\mathcal{B}^z(x^*)$, a direction $\bar{d} \in D^z(x^*)$ exists such that

$$\bar{x} = x^* + \bar{d}.$$

We denote by $\delta > 0$ the constant that satisfies

$$f(x^* + \bar{d}) = f(x^*) - \delta < f(x^*), \tag{4.3.29}$$

and by $\epsilon > 0$ the radius of the neighborhood $\mathcal{B}^c(x^*; \epsilon)$ such that

$$\begin{aligned} &|f(x) - f(x^*)| \leq \delta/2, \\ &|f(x + \bar{d}) - f(x^* + \bar{d})| \leq \delta/2, \end{aligned}$$

for all $x \in \mathcal{B}^c(x^*; \epsilon)$. By considering (4.3.27) and Proposition 4.3.3, we have that

$$\lim_{k \to \infty, k \in K} \tilde{x}_k = x^*. \tag{4.3.30}$$

Hence, for $k \in K$ and sufficiently large, we have

$$\begin{aligned} &(\tilde{x}_k)_z = (x^*)_z, \\ &\tilde{x}_k \ \in \ \mathcal{B}^c(x^*; \epsilon), \\ &\tilde{x}_k + \bar{d} \ \in \ \mathcal{B}^c(x^* + \bar{d}; \epsilon). \end{aligned}$$

Hence, $\bar{d} \in D^z(\tilde{x}_k)$. Moreover, by Proposition 4.3.3, the instructions of the algorithm imply that for $k \in K$ sufficiently large $\tilde{\alpha}^{(\bar{d})} = 1$. Therefore

$$|f(\tilde{x}_k + \bar{d}) - f(x^* + \bar{d})| \leq \delta/2. \tag{4.3.31}$$

By relation (4.3.29), we have that

$$f(x^* + \bar{d}) + f(\tilde{x}_k + \bar{d}) - f(\tilde{x}_k + \bar{d}) = f(x^*) - \delta. \tag{4.3.32}$$

The above relation along with (4.3.31) implies that

$$f(\tilde{x}_k + \bar{d}) \leq f(x^*) - \delta/2.$$

Then, Algorithm 3 would generate the new iterate x_{k+1} such that

$$f(x_{k+1}) \leq f(\tilde{x}_k + \bar{d}) \leq f(x^*) - \delta/2, \tag{4.3.33}$$

thus contradicting the fact that $\{f(x_k)\}$ is convergent towards $f(x^*) = f^*$. □

Now we can prove the main convergence result of the algorithm.

Theorem 4.3.7 *Let $\{x_k\}$ be the sequence of points generated by Algorithm 3 and let $\{s_k\}$, with $(s_k)_i = 0$ for $i \in I^z$, be a dense sequence in the unit sphere (see Definition 4.2.4). Then,*

(i) a limit point of $\{x_k\}$ exists;

(ii) every limit point of $\{x_k\}$ is stationary for Problem (4.2.2).

Proof Point (i) can be proved by considering that $\{x_k\}$ belongs to the set $X \cap \mathcal{Z}$, which is compact by assumption. Proof of point (ii) is implied by Propositions 4.3.5 and 4.3.6. □

4.3.2 Two algorithms with sufficient decrease

We now turn our attention to two algorithms that improve on the efficiency of Algorithm 3. In spite of their similar structure, these algorithms have different global convergence properties. Their general algorithmic framework is first reported in Algorithm 4, then this scheme is adapted to both algorithms (see Algorithms 5 and 6) by focusing on Phase 2.B. Notice that two differences are observed compared to Algorithm 3. The first difference is related to the data since the set D is initialized as a subset of $D^z(x_0)$ in Algorithms 5 and 6, and the set $\bar{D}$ is included among the inputs of Algorithm 5. This modification reflects the strategy, adopted for both algorithms, of gradually adding directions to the set D, as opposed to Algorithm 3, in order to achieve the enhancement in terms of computational efficiency. The second difference is that the Discrete Search used in Phase 2.A requires a sufficient decrease in the objective function value by using the positive parameter ξ_k. This parameter has also another important role, which can be observed at **Step 2** of both Algorithms 5 and 6. In particular, ξ_k is decreased every time the current iterate is not updated in Phase 1 and the Discrete Search fails with unitary tentative stepsize $\tilde{\alpha}_k^{(d)}$ along each direction in D_k, which is the set of the search directions generated up to the current iteration. The importance of ξ_k, along with the sufficient decrease, is to ensure that all the feasible primitive discrete directions at the current iterate are eventually investigated through the Discrete Search and that a potential descent direction at the current iterate is eventually detected. Instead, in Algorithm 3, the sufficient decrease condition is not required because all the feasible primitive discrete directions at the current iterate are supposed to be available and, by the instructions of the algorithm, all of them are explored accordingly.

As regards the convergence results, Algorithm 5 shares the same strong convergence properties as Algorithm 3 but, thanks to the gradual addition of directions in D, it requires a lower computational cost. As already noticed, to achieve this result, a sufficient decrease in the objective function is required also in the Discrete Search.

At each iteration, Algorithm 5 enriches D_k with new feasible primitive discrete directions when both the current iterate is not updated in Phase 1 and the Discrete Search fails with unitary tentative step along each direction in D_k. If $\tilde{x}_k$ obtained at the end of Phase 1 is not modified in Phase 2.A or a direction in D_k along which the Discrete Search does not fail with $\tilde{\alpha}_k^{(d)} = 1$ exists, D_{k+1} is set equal to D_k and D is set equal to D_{k+1} (**Step 10**). Otherwise, ξ_k is reduced (**Step 2**) and, if all the feasible primitive discrete directions at $\tilde{x}_k$ have been generated, D_{k+1} and D are not changed compared to the previous iteration (**Step 4**). Instead, when $D_k \subset D^z(\tilde{x}_k)$, D_k is enriched with new feasible primitive discrete directions (**Steps 6–7**) and the initial tentative steps of the new directions are set equal to 1.

With respect to Algorithm 5, a further reduction in the computational cost is possible if D_k is enriched only with the primitive discrete directions that are feasible at the points where the Discrete Search fails with unitary stepsize, and not with all the feasible primitive discrete directions in $\bar{D}$. Introducing this modification leads to Algorithm 6 that, however, shows weaker convergence properties than both Algorithm 3 and Algorithm 5. Indeed, we can prove that only the limit points of a particular subsequence of iterates are stationary points. This is due to the fact that when there are directions in $\bar{D}$ that are not added to D, i.e., there are feasible primitive discrete directions that are not investigated, a thorough exploration of all the points in the discrete neighborhood of the current iterate is not guaranteed. As a consequence, there may exist subsequences of iterates such that potential descent directions are not detected by the algorithm. To achieve this convergence result, a sufficient decrease in the objective function is required in the Discrete Search, as in

Algorithm 4 Algorithmic scheme for DFNDFL 2 and DFNDFL 3

DATA

Let $x_0 \in X \cap \mathcal{Z}$, $\xi_0 > 0$, $\theta \in (0,1)$;
let $\{s_k\}$ be a sequence such that $s_k \in D^c(x_0)$ and $\|s_k\| = 1$ for all k;
let $\tilde{\alpha}_0^c = 1$ be the initial stepsize along s_k and let $\alpha_{threshold} > 0$;
let $D = D_0 \subset D^z(x_0)$ be a set of initial feasible primitive discrete directions at point x_0;
let $\bar{D}$ be the union of the sets of feasible primitive discrete directions (see Definition 4.2.7), used in DFNDFL 2;
let $\tilde{\alpha}_0^{(d)} = 1$ be the initial stepsizes along $d \in D$.
For $k = 0, 1, \ldots$

PHASE 1 - Explore continuous variables

Compute α_k^c and $\tilde{s}_k$ by the *Projected Continuous Search*$(\tilde{\alpha}_k^c, x_k, s_k; \alpha_k^c, \tilde{s}_k)$.
If $(\alpha_k^c = 0)$ **then**
    $\tilde{\alpha}_{k+1}^c = \theta \tilde{\alpha}_k^c$ and $\tilde{x}_k = x_k$,
else
    $\tilde{\alpha}_{k+1}^c = \alpha_k^c$ and $\tilde{x}_k = [x_k + \alpha_k^c \tilde{s}_k]_{[l,u]}$.
End If

PHASE 2.A - Explore discrete variables

Set $y^+ = \tilde{x}_k$ and $y = y^+$.
While $D \neq \emptyset$ **do**
  Choose $d \in D$, set $D = D \setminus \{d\}$.
  If $\tilde{\alpha}_k^c < \alpha_{threshold}$ **then** Set $\tilde{\alpha}_k^{(d)} = 1$.
  Compute α by the *Discrete Search*$(\tilde{\alpha}_k^{(d)}, y, d, \xi_k; \alpha)$.
  If $\alpha = 0$ **then**
      Set $y^* = y$ and $\tilde{\alpha}_{k+1}^{(d)} = \max\{1, \lfloor \tilde{\alpha}_k^{(d)}/2 \rfloor\}$,
  else
      Set $y^* = y + \alpha d$ and $\tilde{\alpha}_{k+1}^{(d)} = \alpha$.
  End If
  If $f(y^*) < f(y^+)$ **then** $y^+ = y^*$.
End While

PHASE 2.B - Update the set of discrete search directions

See Algorithm 5 for DFNDFL 2 and Algorithm 6 for DFNDFL 3.

PHASE 3 - Update iterates

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(y^+)$.
End For

Algorithm 5.

Let us begin with the convergence results for Algorithm 5, which are reported in the following propositions.

Algorithm 5 DFNDFL 2

See Algorithm 4 for [DATA] [PHASE 1] [PHASE 2.A]

[PHASE 2.B - Update the set of discrete search directions]

If $y^+ = \tilde{x}_k$ and *Discrete Search* fails with $\tilde{\alpha}_k^{(d)} = 1$ for all $d \in D_k$ **then**
  Set $\xi_{k+1} = \theta \xi_k$.
  If $D_k = \bar{D}$ **then**
    Set $D_{k+1} = D_k$ and $D = D_{k+1}$.
  else
    Generate D_{k+1} such that $D_{k+1} \subseteq \bar{D}$ and $D_{k+1} \supset D_k$, set $D = D_{k+1}$.
    Set $\tilde{\alpha}_{k+1}^{(d)} = 1$ for all $d \in D_{k+1} \setminus D_k$.
  End If
else
  Set $D_{k+1} = D_k$ and $D = D_{k+1}$.
End If

See Algorithm 4 for [PHASE 3]

Proposition 4.3.8 *Let $\{x_k\}, \{\tilde{x}_k\}, \{\xi_k\}, \{\alpha_k^c\}$ and $\{\tilde{\alpha}_k^c\}$ be the sequences yielded by Algorithm 5. Then*

(i) Algorithm 5 is well–defined;

(ii)
$$\lim_{k \to \infty} \max\{\alpha_k^c, \tilde{\alpha}_k^c\} = 0.$$

(iii)
$$\lim_{k \to \infty} \xi_k = 0.$$

Proof To prove point (i), we have to show that the Projected Continuous Search and Discrete Search cannot infinitely cycle between Step 4 and Step 6 and between Step 2 and Step 4, respectively. This is proved in Propositions 4.3.1 and 4.3.2. Proof of statement (ii) follows the same steps used to prove Proposition 4.3.3.

Now we prove assertion (iii). By the instruction of Algorithm 5, it follows that $0 < \xi_{k+1} \leq \xi_k$ for all k, meaning that the sequence $\{\xi_k\}$ is monotonically nonincreasing. Hence, $\{\xi_k\}$ converges to a limit $M \geq 0$. Suppose, by contradiction, that $M > 0$. This implies that an index $\bar{k} > 0$ exists such that $\xi_{k+1} = \xi_k = M$ for all $k \geq \bar{k}$. Moreover, for every index $k \geq \bar{k}$, a direction $d \in D^z(\tilde{x}_k)$ exists such that

$$f(x_{k+1}) \leq f(\tilde{x}_k + \alpha_k^{(d)} d) \leq f(\tilde{x}_k) - M \leq f(x_k) - M, \tag{4.3.34}$$

In fact, if such an index did not exist, the algorithm would set $\xi_{k+1} = \theta \xi_k$. Relation (4.3.34) implies $f(x_k) \to -\infty$, which is in contrast with the assumption that f is continuous on the compact set X. This concludes the proof. □

We point out that the hypotheses of Lemma 4.3.4 and Proposition 4.3.5 hold also for Algorithm 5. Therefore, it is possible to prove the following convergence result with respect to the continuous variables.

Proposition 4.3.9 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 5. Let $\bar{x} \in X \cap \mathcal{Z}$ be any limit point of $\{x_k\}$ and K be the subset of indices such that*

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}. \tag{4.3.35}$$

If the subsequence $\{s_k\}_K$, with $(s_k)_i = 0$ for $i \in I^z$, is dense in the unit sphere (see Definition 4.2.4), then $\bar{x}$ is a stationary point for Problem (4.2.2).

Proof The proof follows by reasoning as in Lemma 4.3.4 and Proposition 4.3.5. □

The convergence result with respect to the discrete variables is represented by the following proposition, which follows the reasoning used in Proposition 4.3.6.

Proposition 4.3.10 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 5 and $x^* \in X \cap \mathcal{Z}$ be an accumulation point. Then,*

$$f(x^*) \leq f(\bar{x}),$$

for all $\bar{x} \in \mathcal{B}^z(x^*)$.

Proof By point (iii) of Proposition 4.3.8 and by the instructions of the algorithm, it follows that, for k sufficiently large, in Phase 2.B of the algorithm we have either $D_k = \bar{D}$ or $D_k \subset \bar{D}$. If by contradiction the latter relation holds for any k, we have that at each iteration a set D_{k+1} is generated such that $D_{k+1} \supset D_k$. This is in contrast with Proposition 4.2.8, which implies that $\bar{D}$ has a finite number of directions. Hence, an index $\bar{k}$ exists such that $D_k = \bar{D}$ for all $k \geq \bar{k}$. By reasoning as in the proof of Proposition 4.3.6, the previous observation guarantees that, for k sufficiently large, $D_k \supseteq D^z(x^*)$ and that the algorithm eventually selects the direction $\bar{d} \in D^z(x^*)$ leading to point $\bar{x}$, used to obtain the contradiction. □

We can now state the main convergence result for Algorithm 5, which shows that every limit point of the sequence of iterates is stationary for Problem (4.2.2). This convergence result is the same as the one proved for Algorithm 3.

Theorem 4.3.11 *Let $\{x_k\}$ be the sequence of points generated by Algorithm 5. Then,*

(i) a limit point of $\{x_k\}$ exists;

(ii) every limit point of $\{x_k\}$ is stationary for Problem (4.2.2).

The convergence results for Algorithm 6 are reported in the following propositions.

Proposition 4.3.12 *Let $\{x_k\}, \{\tilde{x}_k\}, \{\xi_k\}, \{\alpha_k^c\}$ and $\{\tilde{\alpha}_k^c\}$ be the sequences yielded by Algorithm 6. Then*

(i) Algorithm 6 is well–defined;

(ii)

$$\lim_{k\to\infty} \max\{\alpha_k^c, \tilde{\alpha}_k^c\} = 0.$$

Algorithm 6 DFNDFL 3

See Algorithm 4 for DATA | PHASE 1 | PHASE 2.A

PHASE 2.B - Update the set of discrete search directions

If $y^+ = \tilde{x}_k$ and *Discrete Search* fails with $\tilde{\alpha}_k^{(d)} = 1$ for all $d \in D_k$ **then**
  Set $\xi_{k+1} = \theta \xi_k$.
  If $D_k \supseteq D^z(\tilde{x}_k)$ **then**
    Set $D_{k+1} = D_k$ and $D = D_{k+1}$,
  else
    Generate D_{k+1} such that $D_{k+1} \subseteq D^z(\tilde{x}_k)$ and $D_{k+1} \supset D_k$.
    Set $D = D_{k+1}$.
    Set $\tilde{\alpha}_{k+1}^{(d)} = 1$ for all $d \in D_{k+1} \setminus D_k$.
  End If
else
  Set $D_{k+1} = D_k$ and $D = D_{k+1}$.
End If

See Algorithm 4 for PHASE 3

(iii)

$$\lim_{k \to \infty} \xi_k = 0.$$

Proof The proof follows by reasoning as in Proposition 4.3.8. □

Remark 1 *By point (iii) of the preceding proposition and the updating rule of the parameter ξ_k used in Algorithm 6, it follows that the set*

$$H = \{k : \xi_{k+1} < \xi_k\}$$

is infinite.

Proposition 4.3.13 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 6. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 1 and let $x^* \in X \cap \mathcal{Z}$ be any accumulation point of $\{x_k\}_H$. If the subsequence $\{s_k\}_H$, with $(s_k)_i = 0$ for $i \in I^z$, is dense in the unit sphere (see Definition 4.2.4), then x^* is a stationary point for Problem* (4.2.2)*, i.e.*

$$f_c^\circ(x^*; s) \geq 0, \qquad \text{for all } s \in D^c(x^*). \tag{4.3.36}$$

Proof For any accumulation point x^* of $\{x_k\}_H$, let $K \subseteq H$ be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*. \tag{4.3.37}$$

Notice that, for all $k \in K$, $(\tilde{x}_k)_z = (x_k)_z$ and $\tilde{\alpha}_k^{(d)} = 1$, $d \in D_k$, by the instructions of Algorithm 6. Hence, for all $k \in K$, by recalling (4.3.37), the discrete variables are no longer updated. Then the proof follows by analogous reasoning as in Lemma 4.3.4 and Proposition 4.3.5. □

Proposition 4.3.14 *Let $\{x_k\}$, $\{\tilde{x}_k\}$, and $\{\xi_k\}$ be the sequences produced by Algorithm 6. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 1 and x^* be any accumulation point of $\{x_k\}_H$, then*

$$f(x^*) \leq f(\bar{x}), \qquad \text{for all } \bar{x} \in \mathcal{B}^z(x^*).$$

Proof Let $K \subseteq H$ be an index set such that

$$\lim_{k \to \infty, k \in K} x_k = x^*.$$

For every $k \in H$, we have

$$\begin{aligned} (\tilde{x}_k)_z &= (x_k)_z, \\ \tilde{\alpha}_k^{(d)} &= 1, \quad d \in D_k, \end{aligned}$$

meaning that the discrete variables are no longer updated by the Discrete Search.

Let us consider any point $\bar{x} \in \mathcal{B}^z(x^*)$. By the definition of discrete neighborhood $\mathcal{B}^z(x^*)$, a direction $\bar{d} \in D^z(x^*)$ exists such that

$$\bar{x} = x^* + \bar{d}. \tag{4.3.38}$$

Recalling the steps of Algorithm 6, we have, for all $k \in H$ and sufficiently large, that

$$(x^*)_z = (x_k)_z = (\tilde{x}_k)_z.$$

Further, by Proposition 4.3.12, we have

$$\lim_{k \to \infty, k \in K} \tilde{x}_k = x^*.$$

Then, for all $k \in K$ and sufficiently large, (4.3.38) implies

$$(x_k + \bar{d})_z = (\tilde{x}_k + \bar{d})_z = (x^* + \bar{d})_z = (\bar{x})_z.$$

Hence, for all $k \in K$ and sufficiently large, by the definition of discrete neighborhood we have $\bar{d} \in D^z(\tilde{x}_k)$ and

$$\tilde{x}_k + \bar{d} \in X \cap \mathcal{Z}.$$

Then, since $k \in H$, by the definition of H we have

$$f(\tilde{x}_k + \bar{d}) > f(\tilde{x}_k) - \xi_k. \tag{4.3.39}$$

Now, by (iii) of Proposition 4.3.12, and taking the limit for $k \to \infty$, with $k \in K$, in (4.3.39), the result follows. □

Theorem 4.3.15 *Let $\{x_k\}$ be the sequence of points generated by Algorithm 6. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 1 and let $\{s_k\}_H$, with $(s_k)_i = 0$ for $i \in I^z$, be a dense subsequence in the unit sphere (see Definition 4.2.4). Then,*

(i) a limit point of $\{x_k\}_H$ exists;

(ii) every limit point x^* of $\{x_k\}_H$ is stationary for Problem 4.2.2.

Proof As regards point (i), since $\{x_k\}_H$ belongs to the compact set $X \cap \mathcal{Z}$, it admits limit points. The proof of point (ii) follows by considering Propositions 4.3.13 and 4.3.14. □

4.3.3 A computationally efficient algorithm with sufficient decrease

The last algorithm we propose performs a distributed minimization along both the continuous and discrete variables (see Algorithm 7 for a detailed scheme). The convergence result is established only for a particular subsequence of iterates, as for Algorithm 6, but, in contrast with this, Algorithm 7 is computationally cheaper. Indeed, the current iterate is updated as soon as a point leading to a sufficient decrease in the objective function is found.

Proposition 4.3.16 *Let $\{x_k\}$, $\{\tilde{x}_k\}$, $\{\xi_k\}$, $\{\alpha_k^c\}$, $\{\tilde{\alpha}_k^c\}$ be the sequences produced by Algorithm 7. Then,*

(i) Algorithm 7 is well–defined;

(ii)
$$\lim_{k\to\infty} \max\{\alpha_k^c, \tilde{\alpha}_k^c\} = 0.$$

(iii)
$$\lim_{k\to\infty} \xi_k = 0.$$

Proof The proof follows the same steps used to prove Proposition 4.3.8. □

Remark 2 *By point (iii) of the preceding proposition and the updating rule of the parameter ξ_k used in Algorithm 7, it follows that the set*

$$H = \{k : \xi_{k+1} < \xi_k\}$$

is infinite.

Proposition 4.3.17 *Let $\{x_k\}$ be the sequence of points produced by Algorithm 7. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 1 and let $x^* \in X \cap \mathcal{Z}$ be any accumulation point of $\{x_k\}_H$. If the subsequence $\{s_k\}_H$, with $(s_k)_i = 0$ for $i \in I^z$, is dense in the unit sphere (see Definition 4.2.4), then x^* is a stationary point for Problem* (4.2.2)*, i.e.*

$$f_c^\circ(x^*; s) \geq 0, \qquad \text{for all } s \in D^c(x^*). \tag{4.3.40}$$

Proof For any accumulation point x^* of $\{x_k\}_H$, let $K \subseteq H$ be an index set such that

$$\lim_{k\to\infty, k\in K} x_k = x^*. \tag{4.3.41}$$

Notice that, for all $k \in K$, $(\tilde{x}_k)_z = (x_k)_z$ and $\tilde{\alpha}_k^{(d)} = 1$, $d \in D_k$, by the instructions of Algorithm 7. Hence, for all $k \in K$, by recalling (4.3.41), the discrete variables are no longer updated. Then the proof follows by analogous reasoning as in Lemma 4.3.4 and Proposition 4.3.5. □

Proposition 4.3.18 *Let $\{x_k\}$, $\{\tilde{x}_k\}$, and $\{\xi_k\}$ be the sequences produced by Algorithm 7. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 1 and $x^* \in X \cap \mathcal{Z}$ be any accumulation point of $\{x_k\}_H$, then*

$$f(x^*) \leq f(\bar{x}), \qquad \text{for all } \bar{x} \in \mathcal{B}^z(x^*).$$

Algorithm 7 DFNDFL 4

DATA

Let $x_0 \in X \cap \mathcal{Z}$, $\xi_0 > 0$, $\theta \in (0,1)$;
let $\{s_k\}$ be a sequence such that $s_k \in D^c(x_0)$ and $\|s_k\| = 1$ for all k;
let $\tilde{\alpha}_0^c = 1$ be the initial stepsize along s_k;
let $D = D_0 \subset D^z(x_0)$ be a set of initial feasible primitive discrete directions at point x_0;
let $\tilde{\alpha}_0^{(d)} = 1$ be the initial stepsizes along $d \in D$.
For $k = 0, 1, \ldots$

PHASE 1 - Explore continuous variables

Compute α_k^c and $\tilde{s}_k$ by the *Projected Continuous Search*$(\tilde{\alpha}_k^c, x_k, s_k; \alpha_k^c, \tilde{s}_k)$.
If $(\alpha_k^c = 0)$ **then**
    $\tilde{\alpha}_{k+1}^c = \theta \tilde{\alpha}_k^c$ and $\tilde{x}_k = x_k$,
else
    $\tilde{\alpha}_{k+1}^c = \alpha_k^c$ and $\tilde{x}_k = [x_k + \alpha_k^c \tilde{s}_k]_{[l,u]}$.
End If

PHASE 2.A - Explore discrete variables

Set $y^+ = \tilde{x}_k$.
While $D \neq \emptyset$ and $y^+ = \tilde{x}_k$ **do**
  Choose $d \in D$, set $D = D \setminus \{d\}$ and $y = y^+$.
  Compute α by the *Discrete Search*$(\tilde{\alpha}_k^{(d)}, y, d, \xi_k; \alpha)$.
  If $\alpha = 0$ **then**
      Set $y^+ = y$ and $\tilde{\alpha}_{k+1}^{(d)} = \max\{1, \lfloor \tilde{\alpha}_k^{(d)}/2 \rfloor\}$,
  else
      Set $y^+ = y + \alpha d$ and $\tilde{\alpha}_{k+1}^{(d)} = \alpha$.
  End If
End While

PHASE 2.B - Update the set of discrete search directions

If $y^+ = \tilde{x}_k$ and *Discrete Search* fails with $\tilde{\alpha}_k^{(d)} = 1$ for all $d \in D_k$ **then**
  Set $\xi_{k+1} = \theta \xi_k$.
  If $D_k \supseteq D^z(\tilde{x}_k)$ **then**
      Set $D_{k+1} = D_k$ and $D = D_{k+1}$,
  else
      Generate D_{k+1} such that $D_{k+1} \subseteq D^z(\tilde{x}_k)$ and $D_{k+1} \supset D_k$, set $D = D_{k+1}$.
      Set $\tilde{\alpha}_{k+1}^{(d)} = 1$ for all $d \in D_{k+1} \setminus D_k$.
  End If
else
  Set $D_{k+1} = D_k$ and $D = D_{k+1}$.
End If

PHASE 3 - Update iterates

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $f(x_{k+1}) \leq f(y^+)$.
End For

Proof Let $K \subseteq H$ be an index set such that

$$\lim_{k\to\infty, k\in K} x_k = x^*.$$

For every $k \in H$, we have

$$\begin{aligned} (\tilde{x}_k)_z &= (x_k)_z, \\ \tilde{\alpha}_k^{(d)} &= 1, \quad d \in D_k, \end{aligned}$$

meaning that the discrete variables are no longer updated by the Discrete Search.

Let us consider any point $\bar{x} \in \mathcal{B}^z(x^*)$. By the definition of discrete neighborhood $\mathcal{B}^z(x^*)$, a direction $\bar{d} \in D^z(x^*)$ exists such that

$$\bar{x} = x^* + \bar{d}. \tag{4.3.42}$$

Recalling the steps of Algorithm 7, we have, for all $k \in H$ and sufficiently large, that

$$(x^*)_z = (x_k)_z = (\tilde{x}_k)_z.$$

Further, by Proposition 4.3.12, we have

$$\lim_{k\to\infty, k\in K} \tilde{x}_k = x^*.$$

Then, for all $k \in K$ and sufficiently large, (4.3.42) implies

$$(x_k + \bar{d})_j = (\tilde{x}_k + \bar{d})_j = (x^* + \bar{d})_j = (\bar{x})_j, \text{ with } j \in I_z.$$

Hence, for all $k \in K$ and sufficiently large, by the definition of discrete neighborhood we have $\bar{d} \in D^z(\tilde{x}_k)$.

$$\tilde{x}_k + \bar{d} \in X \cap \mathcal{Z}.$$

Then, since $k \in H$, by the definition of H we have

$$f(\tilde{x}_k + \bar{d}) > f(\tilde{x}_k) - \xi_k. \tag{4.3.43}$$

Now, by (iii) of Proposition 4.3.12, and taking the limit for $k \to \infty$, with $k \in K$, in (4.3.43), the result follows. □

Theorem 4.3.19 *Let $\{x_k\}$ be the sequence of points generated by Algorithm 7. Let $H \subseteq \{1, 2, \dots\}$ be defined as in Remark 2 and let $\{s_k\}_H$, with $(s_k)_i = 0$ for $i \in I^z$, be a dense subsequence in the unit sphere (see Definition 4.2.4). Then,*

(i) a limit point of $\{x_k\}_H$ exists;

(ii) every limit point x^* of $\{x_k\}_H$ is stationary for Problem (4.2.2).

Proof As regards point (i), since $\{x_k\}_H$ belongs to the compact set $X \cap \mathcal{Z}$, it admits limit points. The prove of point (ii) follows by considering Propositions 4.3.17 and 4.3.18. □

4.4 An algorithm for nonsmooth nonlinearly constrained problems

In this section, we consider the nonsmooth nonlinearly constrained problem defined in Problem 4.1.2. The nonlinear constraints are handled through a simple penalty approach (see, e.g., [75]). In particular, given a positive parameter $\varepsilon > 0$, we introduce the following penalty function

$$P(x;\varepsilon) = f(x) + \frac{1}{\varepsilon}\sum_{i=1}^{m} \max\{0, g_i(x)\},$$

which allows us to define the following bound constrained problem

$$\begin{array}{ll} \min & P(x;\varepsilon) \\ s.t. & x \in X \cap \mathcal{Z}. \end{array} \tag{4.4.1}$$

Hence, only the nonlinear constraints are penalized and the minimization is performed over the set $X \cap \mathcal{Z}$. The algorithms described in Section 4.3 are thus suited for solving this problem, as highlighted in the following remark.

Remark 3 *Observe that, for any $\varepsilon > 0$, the structure and properties of Problem (4.4.1) are the same as Problem (4.2.2). The Lipschiz continuity (with respect to the continuous variables) of the penalty function $P(x;\varepsilon)$ follows by the Lipschitz continuity of f and g_i, with $i \in \{1,\dots,m\}$. In particular, called L_f and L_{g_i} the Lipschitz constants of f and g_i, respectively, we have that the Lipschitz constant of the penalty function $P(x;\varepsilon)$ is*

$$L \leq L_f + \frac{1}{\varepsilon}\sum_{i=1}^{m} L_{g_i}.$$

To prove the equivalence between Problem (4.1.2) and Problem (4.4.1), we report an extended version of the Mangasarian–Fromowitz Constraint Qualification (EMFCQ) condition for Problem (4.1.2) by taking into account its mixed-integer structure. Indeed, this condition states that at a point that is infeasible for Problem (4.1.2), a direction feasible with respect to $X \cap \mathcal{Z}$ (according to Definitions 4.2.6 and 4.2.9) that guarantees a reduction in the constraint violation exists. This direction considers either the continuous or the discrete variables.

Assumption 4.4.1 (EMFCQ for mixed-integer problems) *Given Problem (4.1.2), for any $x \in (X \cap \mathcal{Z}) \setminus \overset{\circ}{\mathcal{F}}$, one of the following conditions holds:*

(i) a direction $s \in D^c(x)$ exists such that

$$(\xi^{g_i})^\top s < 0,$$

for all $\xi^{g_i} \in \partial_c g_i(x)$ with $i \in \{h \in \{1,\dots,m\} : \ g_h(x) \geq 0\}$;

(ii) a direction $\bar{d} \in D^z(x)$ exists such that

$$\sum_{i=1}^{m} \max\{0, g_i(x+\bar{d})\} < \sum_{i=1}^{m} \max\{0, g_i(x)\}.$$

In particular, we first prove the equivalence between local minimum points and between global minimum points of the two problems. Then, we prove that any Clarke stationary point of Problem (4.4.1) (according to Definition 4.2.21) is a stationary point for Problem (4.1.2) (according to Definition 4.2.28).

Proposition 4.4.2 *Let Assumption 4.4.1 hold. A threshold value $\varepsilon^\star > 0$ exists such that the function $P(x;\varepsilon)$ has no Clarke stationary points in $(X \cap \mathcal{Z}) \setminus \mathcal{F}$ for any $\varepsilon \in (0, \varepsilon^\star]$.*

Proof We proceed by contradiction and assume that for any integer k, an $\varepsilon_k \leq 1/k$ and a stationary point for Problem (4.4.1) $x_k \in (X \cap \mathcal{Z}) \setminus \mathcal{F}$ exist. Then, let us consider a limit point $\bar{x} \in (X \cap \mathcal{Z}) \setminus \overset{\circ}{\mathcal{F}}$ of the sequence $\{x_k\}$ and, without loss of generality, let us call the corresponding subsequence as $\{x_k\}$ as well.

First we assume that point (i) of Assumption 4.4.1 holds at $\bar{x}$. Therefore, a direction $\bar{s} \in D^c(\bar{x})$ exists such that

$$(\xi^{g_i})^\top \bar{s} < 0 \ \text{ for all } \ \xi^{g_i} \in \partial_c g_i(\bar{x}) \text{ with } i \in \{h \in \{1,\dots,m\} : g_h(\bar{x}) \geq 0\}.$$

In particular, it follows that

$$\max_{\substack{\xi^{g_i} \in \partial_c g_i(\bar{x}) \\ i \in I(\bar{x})}} (\xi^{g_i})^\top \bar{s} = -\eta < 0, \tag{4.4.2}$$

where $I(\bar{x}) = \left\{ i \in \{1,\dots,m\} : g_i(\bar{x}) = \bar{\phi}(\bar{x}) \right\}$, $\bar{\phi}(x) = \max\{0, g_1(x), \dots, g_m(x)\}$ and η is a positive scalar. Note that $\bar{x} \notin \mathcal{F}$ implies $\bar{\phi}(\bar{x}) > 0$.

By [165, Proposition 2.3], it follows that $\bar{s} \in D^c(x_k)$. Moreover, since x_k satisfies the Definition 4.2.19 of stationary point, we have that

$$0 \leq P^{Cl}(x_k;\varepsilon,\bar{s}) = \max_{\xi \in \partial_c P(x_k;\varepsilon)} \xi^\top \bar{s}. \tag{4.4.3}$$

By [49], we know that

$$\partial_c P(x_k;\varepsilon) \subseteq \partial_c f(x_k) + \frac{1}{\varepsilon} \partial_c(\max\{0, g_1(x_k), \dots, g_m(x_k)\}) \tag{4.4.4}$$

and

$$\partial(\max\{0, g_1(x_k), \dots, g_m(x_k)\}) \subseteq Co\left(\{\partial_c g_i(x_k) : i \in I(x_k)\}\right), \tag{4.4.5}$$

where $\beta^i \geq 0$ for all $i \in I(x_k)$ and $Co(A)$ denotes the convex hull of a set A (see [207, Theorem 3.3]).

Therefore, by (4.4.3)–(4.4.5), $\xi_k^f \in \partial_c f(x_k)$, $\xi_k^{g_i} \in \partial_c g_i(x_k)$ and β_k^i with $i \in I(x_k)$ exist such that

$$\left(\xi_k^f + \frac{1}{\varepsilon_k} \sum_{i \in I(x_k)} \beta_k^i \xi_k^{g_i} \right)^\top \bar{s} \geq 0, \tag{4.4.6}$$

$$\sum_{i \in I(x_k)} \beta_k^i = 1 \ \text{ and } \ \beta_k^i \geq 0.$$

Since m is a finite number, there exists a subsequence of $\{x_k\}$ such that $I(x_k) = \bar{I}$.

The local boundedness of the generalized gradient of the locally Lipschitz continuous functions f and g_i and the fact that $X \cap \mathcal{Z}$ can be covered by a finite number of local sets implies that the sequences ξ_k^f and $\xi_k^{g_i}$, with $i \in \bar{I}$, are bounded. Hence, we get that

$$\xi_k^f \to \bar{\xi}^f, \tag{4.4.7a}$$

$$\xi_k^{g_i} \to \bar{\xi}^{g_i} \text{ for all } i \in \bar{I}, \tag{4.4.7b}$$

$$\beta_k^i \to \bar{\beta}^i \text{ for all } i \in \bar{I}. \tag{4.4.7c}$$

Now the upper semicontinuity of $\partial_c f$ and $\partial_c g_i$, with $i \in \bar{I}$, at $\bar{x}$ (see Proposition 2.1.5 in [49]) implies that $\bar{\xi}^f \in \partial_c f(\bar{x})$ and $\bar{\xi}^{g_i} \in \partial_c g_i(\bar{x})$ for all $i \in \bar{I}$.

The continuity of the problem functions guarantees that for k sufficiently large

$$\{i : g_i(\bar{x}) - \phi(\bar{x}) < 0\} \subseteq \{i : g_i(x_k) - \phi(x_k) < 0\},$$

and, in turn, this implies that for k sufficiently large

$$\{i : g_i(x_k) - \phi(x_k) = 0\} = I(x_k) \subseteq I(\bar{x}) = \{i : g_i(\bar{x}) - \phi(\bar{x}) = 0\}.$$

Since $I(x_k) \subseteq I(\bar{x})$, we have that

$$\bar{I} \subseteq I(\bar{x}). \tag{4.4.8}$$

Finally, for k sufficiently large, (4.4.2), (4.4.7), and (4.4.8) imply

$$\left(\xi_k^{g_i}\right)^\top \bar{s} \leq -\frac{\eta}{2} \text{ for all } i \in \bar{I}, \tag{4.4.9}$$

and (4.4.6), multiplied by ε_k, implies

$$\left(\varepsilon_k \xi_k^f + \sum_{i \in \bar{I}} \beta_k^i \xi_k^{g_i}\right)^\top \bar{s} \geq 0. \tag{4.4.10}$$

Equations (4.4.9) and (4.4.10) yield

$$0 \leq \left(\varepsilon_k \xi_k^f + \sum_{i \in \bar{I}} \beta_k^i \xi_k^{g_i}\right)^\top \bar{s} \leq \left(\varepsilon_k \xi_k^f\right)^\top \bar{s} - \frac{\eta}{2}.$$

which, by using (4.4.7), gives rise to a contradiction when $k \to \infty$ and, hence, $\varepsilon_k \to 0$.

Now we assume that point (ii) of Assumption 4.4.1 holds at $\bar{x}$. Let $\bar{d} \in D^z(\bar{x})$ be the direction such that

$$\sum_{i=1}^{m} \max\{0, g_i(\bar{x} + \bar{d})\} < \sum_{i=1}^{m} \max\{0, g_i(\bar{x})\}. \tag{4.4.11}$$

By minor modification of [165, Proposition 2.3], we have that for k sufficiently large $D^z(\bar{x}) \subseteq D^z(x_k)$. Hence, it follows that $\bar{d} \in D^z(x_k)$. By definition of stationary point and discrete neighborhood, we have

$$P(x_k; \varepsilon_k) \leq P(x_k + \bar{d}; \varepsilon_k), \qquad \text{where } x_k + \bar{d} \in \mathcal{B}^z(x_k).$$

Hence,

$$f(x_k) + \frac{1}{\varepsilon_k}\sum_{i=1}^{m}\max\{0, g_i(x_k)\} \leq f(x_k + \bar{d}) + \frac{1}{\varepsilon_k}\sum_{i=1}^{m}\max\{0, g_i(x_k + \bar{d})\}.$$

Multiplying by ε and considering that $\varepsilon_k \to 0$, if we take the limit for $k \to \infty$ we have that

$$\sum_{i=1}^{m}\max\{0, g_i(\bar{x})\} \leq \sum_{i=1}^{m}\max\{0, g_i(\bar{x} + \bar{d})\}.$$

The latter equation is in contradiction with (4.4.11). □

Now, we can prove that there exists a threshold value $\bar{\varepsilon}$ for the penalty parameter such that, for any $\varepsilon \in (0, \bar{\varepsilon})$, any local minimum of the penalized problem is also a local minimum of the original problem.

Proposition 4.4.3 *Let Assumptions 4.4.1 hold. Given Problem* (4.1.2) *and considering Problem* (4.4.1)*, a threshold value $\bar{\varepsilon} > 0$ exists such that for every $\varepsilon \in (0, \bar{\varepsilon})$, any local minimum point $\bar{x}$ of Problem* (4.4.1) *is also a local minimum of Problem* (4.1.2)*.*

Proof The Definition 4.2.14 of local minimum point implies that $\bar{x}$ satisfies the definition of stationary point given in Definition 4.2.19 (see Proposition 4.2.17). In particular, we have that relation (4.2.8) follows by (4.2.4) and that (4.2.7) follows by (4.2.3), which implies that $\bar{x}$ is an unconstrained local minimum point with respect to the continuous variables. By Corollary 4.2.22, we have that (4.2.11) and (4.2.12) hold as well. Therefore, any local minimum point of $P(x;\varepsilon)$ is also a stationary point according to Clarke definition for the continuous variables.

Now Proposition 4.4.2 implies that a threshold value $\varepsilon^\star > 0$ exists such that $\bar{x} \in \mathcal{F} \cap \mathcal{Z} \cap X$ for any $\varepsilon \in (0, \varepsilon^\star]$. Therefore, $P(\bar{x};\varepsilon) = f(\bar{x})$. This implies that any local minimum point of $P(x;\varepsilon)$ in $\mathcal{F} \cap X \cap \mathcal{Z}$ is also a local minimum point for Problem 4.1.2. □

Proposition 4.4.4 *Let Assumption 4.4.1 hold. Given Problem* (4.1.2) *and considering Problem* (4.4.1)*, a threshold value $\bar{\varepsilon} > 0$ exists such that for every $\epsilon \in (0, \bar{\varepsilon})$, any global minimum point $\bar{x}$ of Problem* (4.4.1) *is also a global minimum point of Problem* (4.1.2) *and vice versa.*

Proof We start by proving that any global minimum point of Problem (4.4.1) is also a global minimum point of Problem (4.1.2). Proceeding by contradiction, let us assume that for any integer k a positive scalar $\varepsilon_k < 1/k$ and a point x_k exist such that x_k is a global minimum point of $P(x_k;\varepsilon_k)$ but it is not a global minimum point of $f(x)$. If we denote as $\hat{x}$ a global minimum point of $f(x)$, we have that

$$P(x_k;\varepsilon_k) \leq P(\hat{x};\varepsilon_k) = f(\hat{x}). \tag{4.4.12}$$

Since x_k are global minimum points, by Proposition 4.2.17 and by Corollary 4.2.22 they are also stationary points of $P(x_k;\varepsilon_k)$ according to Clarke definition for the continuous variables. By Proposition 4.4.2 there exists a threshold value $\varepsilon^\star > 0$ such that $x_k \in \mathcal{F} \cap X \cap \mathcal{Z}$ for any $\varepsilon_k \in (0, \varepsilon^\star]$. Therefore, $P(x_k;\varepsilon_k) = f(x_k)$. By (4.4.12), it follows that $f(x_k) \leq f(\hat{x})$, contradicting the assumption that x_k is not a global minimum point of $f(x)$.

Now we prove that any global minimum point $\bar{x}$ of Problem (4.1.2) is also a global minimum point of Problem (4.4.1) for any $\varepsilon \in (0, \bar{\varepsilon})$. Since $\bar{x} \in \mathcal{F} \cap \mathcal{Z} \cap X$, we have that $P(\bar{x}; \varepsilon) = f(\bar{x})$. By the previous proof, a global minimizer x_ε of $P(x; \varepsilon)$ is feasible for Problem (4.1.2), hence $P(x_\varepsilon; \varepsilon) = f(x_\varepsilon)$. Furthermore, it is also a global minimum point of Problem (4.1.2), thus we have $f(x_\varepsilon) = f(\bar{x})$. Therefore, since $P(x_\varepsilon; \varepsilon) = f(\bar{x})$, $\bar{x}$ is also a global minimum point of $P(x; \varepsilon)$. □

In order to give stationarity results for Problem (4.4.1), we have the following proposition.

Proposition 4.4.5 *Let Assumption 4.4.1 hold. For any $\varepsilon > 0$, every stationary point $\bar{x}$ of problem* (4.4.1) *according to Clarke, such that $\bar{x} \in \mathcal{F} \cap \mathcal{Z} \cap X$, is also a stationary point of Problem* (4.1.2)*.*

Proof Since $\bar{x}$ is, by assumption, a stationary point of Problem (4.4.1) according to Clarke (see Corollary 4.2.22), then we have by definition of Clarke stationarity that for all $s \in D^c(\bar{x})$,

$$P^{Cl}(\bar{x}; \varepsilon) = \max \left\{ \xi^\top s \, : \, \xi \in \partial_c P(\bar{x}; \varepsilon) \right\} \geq 0, \tag{4.4.13}$$

and

$$P(\bar{x}; \varepsilon) \leq P(x; \varepsilon) \quad \text{for all } x \in \mathcal{B}^z(\bar{x}). \tag{4.4.14}$$

Hence, by 4.4.13, there exists $\xi_s \in \partial_c P(\bar{x}; \varepsilon)$ such that $(\xi_s)^\top s \geq 0$ for all $s \in D^c(\bar{x})$. Now, we recall that

$$\partial_c P(x; \varepsilon) \subseteq \partial_c f(x) + \frac{1}{\varepsilon} \sum_{i \in I(x)} \beta_i \partial_c g_i(x),$$

for some β_i, with $i \in I(x)$, such that $\sum_{i \in I(x)} \beta_i = 1$ and $\beta_i \geq 0$ for all $i \in I(x)$. Hence, we have that $\xi_s \in \partial_c f(\bar{x}) + \frac{1}{\varepsilon} \sum_{i \in I(\bar{x})} \beta_i \partial_c g_i(\bar{x})$. Then, denoting $\lambda_i = \beta_i / \varepsilon$ with $i \in I(\bar{x})$, and assuming $\lambda_i = 0$ for all $i \notin I(x)$, we can write, for all $s \in D^c(\bar{x})$,

$$\max \left\{ \xi^\top s \, : \, \xi \in \partial_c f(\bar{x}) + \sum_{i=1}^{m} \lambda_i \partial_c g_i(\bar{x}) \right\} \geq 0, \tag{4.4.15}$$

$$(\lambda)^T g(\bar{x}) = 0 \text{ and } \lambda \geq 0. \tag{4.4.16}$$

Since Proposition 4.4.2 implies that $\bar{x}$ is feasible for Problem (4.1.2), by (4.4.14) we have

$$f(\bar{x}) \leq f(x) \quad \text{for all } x \in \mathcal{B}^z(\bar{x}), \tag{4.4.17}$$

Considering that $\bar{x} \in \mathcal{F} \cap \mathcal{Z} \cap X$, 4.4.15, 4.4.16, and 4.4.17 prove that $\bar{x}$ is a KKT stationary point for Problem (4.1.2), thus concluding the proof. □

Proposition 4.4.6 *Let Assumption 4.4.1 hold. Then, a threshold value $\varepsilon^\star > 0$ exists such that, for every $\varepsilon \in (0, \varepsilon^\star]$, every stationary point $\bar{x}$ of Problem* (4.4.1) *is stationary (according to Definition 4.2.28) for Problem (4.1.2).*

Proof Since $\bar{x}$ is stationary for Problem (4.4.1), we have by Definition 4.2.19 that

$$P^{\circ}(\bar{x};\varepsilon,s) \geq 0 \quad \text{for all} \quad s \in D^{c}(\bar{x}), \tag{4.4.18}$$

and

$$P(\bar{x};\varepsilon) \leq P(x;\varepsilon) \quad \text{for all } x \in \mathcal{B}^{z}(\bar{x}). \tag{4.4.19}$$

Then, by Definitions 4.2.5 and 4.2.16, we have that for all $s \in D^{c}(\bar{x})$

$$\limsup_{y_c \to x_c,\, y_z = x_z,\, t \downarrow 0} \frac{P(y+ts;\varepsilon) - P(y;\varepsilon)}{t} = P^{Cl}(\bar{x};\varepsilon,s) \geq P^{\circ}(\bar{x};\varepsilon,s).$$

By (4.4.18), it follows that

$$P^{Cl}(\bar{x};\varepsilon,s) \geq 0 \quad \text{for all} \quad s \in D^{c}(\bar{x}).$$

The proof follows by considering Propositions 4.4.2, 4.4.5 and 4.4.19. □

As a result of the previous propositions, we can apply one of the four optimization algorithms proposed in Section 4.3 to solve Problem (4.4.1) provided that the penalty parameter is sufficiently small, as stated in the next proposition. Suppose to choose Algorithm 7. Algorithm 8 reports the scheme of the algorithm designed for solving Problem (4.1.2). It is obtained from Algorithm 7 by replacing $f(x)$ with $P(x;\varepsilon)$, where $\varepsilon > 0$ is a sufficiently small value. We point out that in Algorithm 8 both line-search procedures are performed by replacing $f(x)$ with $P(x;\varepsilon)$ as well.

Proposition 4.4.7 *Let Assumption 4.4.1 hold and let $\{x_k\}$ be the sequence produced by Algorithm 8. Let $\bar{x}$ be any limit point of $\{x_k\}$ and K be the subset of indices such that*

$$\lim_{k\to\infty, k\in K} x_k = \bar{x}.$$

If the subsequence $\{s_k\}_K$ is dense in the unit sphere (see Definition 4.2.4), then a threshold value ε^* exists such that $\bar{x}$ is stationary for Problem (4.1.2) for all $\varepsilon \in (0, \varepsilon^*]$.

Proof The proof follows from Proposition 4.4.6 and Propositions 4.3.7, 4.3.11, 4.3.15 or 4.3.19 based on the algorithm chosen. □

4.5 Numerical experiments

In this section we report the numerical experiments performed on a set of test bound constrained problems selected from the literature. The numerical experience on test problems with general constraints is left as future work. Among the four algorithms proposed in Section 4.3, only Algorithm 7 has been taken into account for these experiments since it is expected to show the best performance thanks to its low computational cost. Indeed, while the other algorithms are provided with stronger global convergence properties, on practical problems their performance in terms of efficiency may be affected by the higher computational cost required. In the sequel, Algorithm 7 is referred to as DFNDFL.

In this section, state-of-the-art solvers are used as benchmarks to test the efficiency and robustness of the algorithm proposed. In both cases, to improve the

Algorithm 8 DFNDFL–CON

DATA

Let $x_0 \in X \cap \mathcal{Z}$, $\varepsilon > 0$ sufficiently small, $\xi_0 > 0$, $\theta \in (0,1)$;
let $\{s_k\}$ be a sequence such that $s_k \in D^c(x_0)$ and $\|s_k\| = 1$ for all k;
let $\tilde{\alpha}_0^c = 1$ be the initial stepsize along s_k;
let $D = D_0 \subset D^z(x_0)$ be a set of initial feasible primitive discrete directions at point x_0;
let $\tilde{\alpha}_0^{(d)} = 1$ be the initial stepsizes along $d \in D$.
For $k = 0, 1, \ldots$

PHASE 1 - Explore continuous variables

Compute α_k^c and $\tilde{s}_k$ by the *Projected Continuous Search*$(\tilde{\alpha}_k^c, x_k, s_k; \alpha_k^c, \tilde{s}_k)$.
If $(\alpha_k^c = 0)$ **then**
  $\tilde{\alpha}_{k+1}^c = \theta \tilde{\alpha}_k^c$ and $\tilde{x}_k = x_k$,
else
  $\tilde{\alpha}_{k+1}^c = \alpha_k^c$ and $\tilde{x}_k = [x_k + \alpha_k^c \tilde{s}_k]_{[l,u]}$.
End If

PHASE 2.A - Explore discrete variables

Set $y^+ = \tilde{x}_k$.
While $D \neq \emptyset$ and $y^+ = \tilde{x}_k$ **do**
  Choose $d \in D$, set $D = D \setminus \{d\}$ and $y = y^+$.
  Compute α by the *Discrete Search*$(\tilde{\alpha}_k^{(d)}, y, d, \xi_k; \alpha)$.
  If $\alpha = 0$ **then**
    Set $y^+ = y$ and $\tilde{\alpha}_{k+1}^{(d)} = \max\{1, \lfloor \tilde{\alpha}_k^{(d)}/2 \rfloor\}$,
  else
    Set $y^+ = y + \alpha d$ and $\tilde{\alpha}_{k+1}^{(d)} = \alpha$.
  End If
End While

PHASE 2.B - Update set of discrete search directions

If $y^+ = \tilde{x}_k$ and *Discrete Search* fails with $\tilde{\alpha}_k^{(d)} = 1$ for all $d \in D_k$ **then**
  Set $\xi_{k+1} = \theta \xi_k$.
  If $D_k \supseteq D^z(\tilde{x}_k)$ **then**
    Set $D_{k+1} = D_k$ and $D = D_{k+1}$,
  else
    Generate D_{k+1} such that $D_{k+1} \subseteq D^z(\tilde{x}_k)$ and $D_{k+1} \supset D_k$, set $D = D_{k+1}$.
    Set $\tilde{\alpha}_{k+1}^{(d)} = 1$ for all $d \in D_{k+1} \setminus D_k$.
  End If
else
  Set $D_{k+1} = D_k$ and $D = D_{k+1}$.
End If

PHASE 3 - Update iterates

Find $x_{k+1} \in X \cap \mathcal{Z}$ such that $P(x_{k+1}; \varepsilon) \leq P(y^+; \varepsilon)$.
End For

performance of DFNDFL from a numerical point of view, a modification to Phase 1 is introduced by drawing inspiration from the algorithm CS-DFN proposed in [75] for continuous nonsmooth problems. In particular, recalling that $I^c \cup I^z = \{1, \ldots, n\}$, the change consists in investigating the set of coordinate directions $\{\pm e^1, \ldots, \pm e^{|I^c|}\}$ before exploring a direction from the sequence $\{s_k\}$. Since this set is constant over the iterations, the actual and tentative stepsizes $\alpha_k^{(i)}$ and $\tilde{\alpha}_k^{(i)}$ can be stored for each coordinate direction i, with $i \in I^c$. These stepsizes are reduced whenever the continuous line search (i.e., Algorithm 1 without the projection operator) does not determine any point that satisfies the sufficient decrease condition. When their values become sufficiently small, a direction from the dense sequence s_k is explored. This improvement allows the algorithm to benefit from the presence of the stepsizes $\alpha_k^{(i)}$ and $\tilde{\alpha}_k^{(i)}$, whose values depend on the knowledge across the iterations of the sensitivity of the objective function over the coordinate directions. Therefore, the efficiency of the modified DFNDFL is expected to be higher.

The comparison between Algorithm 7 and some state-of-the-art solvers is reported for 44 bound constrained problems. The first 33 problems, which are related to minimax and nonsmooth unconstrained optimization problems, have been selected from [179, Sections 2 and 3], while the remaining 13 problems have been chosen from [187, 186, 185], which deal with mixed-integer bound constrained optimization. Such problems, which are listed in Table 4.1 along with their dimension, have a number of variables that ranges from 4 to 60.

Problem name	Source	n	Problem name	Source	n
oet5	[179]	4	steiner 2	[179]	12
oet6	[179]	4	shell dual	[179]	15
gamma	[179]	4	watson	[179]	20
kowalik–osborne	[179]	4	wong3	[179]	20
rosen–suzuki	[179]	4	maxl	[179]	20
polak 6	[179]	4	maxq	[179]	20
davidon 2	[179]	4	tr48	[179]	48
shor	[179]	5	mxhilb	[179]	50
colville 1	[179]	5	l1hilb	[179]	50
exp	[179]	5	goffin	[179]	50
pbc1	[179]	5	SOMI prob.10	[187]	5
hs78	[179]	5	SO–I prob. 2	[186]	5
evd61	[179]	6	SO–I prob. 7	[186]	10
elattar	[179]	6	SO–I prob. 9	[186]	12
transformer	[179]	6	SO–I prob.10	[186]	30
wong1	[179]	7	SO–I prob.13	[186]	10
filter	[179]	9	SO–I prob.15	[186]	12
polak 2	[179]	10	SO–I prob.16	[186]	8
maxquad	[179]	10	MISO prob. 6	[185]	15
gill	[179]	10	MISO prob. 7	[185]	2
wong2	[179]	10	MISO prob. 8	[185]	15
polak 3	[179]	11	MISO prob. 9	[185]	3
osborne 2	[179]	11	MISO prob.10	[185]	60

Table 4.1. Unconstrained and bound constrained test problems collection

In order for the original unconstrained problems selected from [179] to suit the

class of problems addressed in this chapter, the following bound constraints are considered for each variable

$$\ell^i = (\tilde{x}_0)^i - 10 \leq \tilde{x}^i \leq (\tilde{x}_0)^i + 10 = u^i \quad \text{for all } i \in \{1, \ldots, n\},$$

where $\tilde{x}_0$ is the starting point. Since the original problems have only continuous variables, the rule applied to obtain mixed-integer problems is to consider a number of integer variables equal to $|I^z| = \lfloor n/2 \rfloor$ and a number of continuous variables equal to $|I^c| = \lceil n/2 \rceil$, where n denotes the dimension of each original problem and $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and ceil operators, respectively. In particular, let us consider both the continuous bound constrained optimization problems from [179] (the bound constraints are introduced as described above), whose formulation is

$$\begin{array}{ll} \min & \tilde{f}(\tilde{x}) \\ s.t. & \ell^i \leq \tilde{x}^i \leq u^i \qquad \text{for all } \; i \in \{1, \ldots, n\}, \\ & \tilde{x}_i \in \mathbb{R} \qquad\quad \text{for all } \; i \in \{1, \ldots, n\}, \end{array} \tag{4.5.1}$$

and the original mixed-integer bound constrained problems from [187, 186, 185], which can be stated as

$$\begin{array}{ll} \min & \tilde{f}(\tilde{x}) \\ s.t. & \ell^i \leq \tilde{x}^i \leq u^i \qquad \text{for all } \; i \in \{1, \ldots, n\}, \\ & x_i \in \mathbb{R} \qquad\quad \text{for all } \; i \in I^c, \\ & x_i \in \mathbb{Z} \qquad\quad \text{for all } \; i \in I^z. \end{array} \tag{4.5.2}$$

The resulting mixed-integer problem can be formulated as follows

$$\begin{array}{ll} \min & f(x) \\ s.t. & \ell^i \leq x^i \leq u^i \qquad \text{for all } \; i \in I^c, \\ & 0 \leq x^i \leq 100 \qquad \text{for all } \; i \in I^z, \\ & x_i \in \mathbb{R} \qquad\quad \text{for all } \; i \in I^c, \\ & x_i \in \mathbb{Z} \qquad\quad \text{for all } \; i \in I^z, \end{array} \tag{4.5.3}$$

where $f(x) = \tilde{f}(\tilde{x})$ with

$$\tilde{x}^i = \begin{cases} x^i, & \text{for all } i \in I^c, \\ \ell^i + x^i(u^i - \ell^i)/100, & \text{for all } i \in I^z. \end{cases}$$

Moreover, the starting point x_0 adopted for Problem (4.5.3) is

$$(x_0)^i = \begin{cases} (\ell^i + u^i)/2 & \text{for all } i \in I^c, \\ 50 & \text{for all } i \in I^z. \end{cases}$$

Algorithms for benchmarking

The algorithms selected as benchmarks for assessing the performance of DFNDFL are listed below.

- DFL *box* [169], a derivative-free linesearch algorithm for bound constrained problems.
- RBFOpt [53], an open–source library RBFOpt for solving black-box optimization problems with expensive function evaluations.

- NOMAD v.3.9.1 [4, 162], a software package which implements the mesh adaptive direct search algorithm.

All the algorithms reported above support mixed-integer problems, thus being suited for the comparison with DFNDFL. The maximum number of function evaluations allowed in each experiment is 5000.

As regards the parameters used in both DFNDFL and DFL *box*, the values used in the experiments are $\gamma = 10^{-6}$, $\delta = 0.5$, $\xi_0 = 1$, and $\theta = 0.5$. Moreover, the initial tentative steps along the coordinate directions $\pm e^i$ and s_k of the modified DFNDFL are

$$\begin{aligned} \tilde{\alpha}_0^i &= (u^i - \ell^i)/2 \qquad \text{for all } i \in I^c, \\ \tilde{\alpha}_0 &= \frac{1}{n}\sum_{i=1}^{n} \tilde{\alpha}_0^i, \end{aligned}$$

while for the discrete directions d the initial tentative steps $\tilde{\alpha}_0^{(d)}$ are fixed to 1. Another computational aspect that needs further discussion is the generation of the continuous and discrete directions. Indeed, in Phases 1 and 2 of DFNDFL (see Algorithm 7), new search directions may be generated to thoroughly explore neighborhoods of the current iterate. To this end, a dense sequence of directions $\{s_k\}$ is required in Phase 1 to explore the continuous variables. Similarly, new primitive discrete directions are generated when $D_{k+1} \supset D_k$. In Phase 1, the sequence s_k is obtained by the Sobol sequence [216, 39], which in [75] guarantees the algorithm with better performance than the Halton sequence [110] used, for instance, in NOMAD. Since Phase 1 is adapted from the algorithm proposed in [75], the Sobol sequence appears to be a reasonable choice. By contrast, the Halton sequence is used in Phase 2 to generate the primitive discrete directions as described in [171] for integer problems. This procedure has been adapted to the mixed-integer case.

As concerns the parameters used for running RBFOpt and NOMAD, while the former is executed by using the default values, for the latter two different algorithms are considered. The first one is based on the default settings, while the second one results from disabling the usage of models in the search phase, which precisely is performed by setting either `MODEL_SEARCH No` or `DISABLE MODELS`. This second version is denoted in the remainder of this document as NOMAD-NOMOD.

Data and performance profiles

The performance difference among the algorithms considered is assessed by using data and performance profiles, which are benchmarking tools widely used in derivative-free optimization (see [192]). In particular, given a set S of algorithms, a set P of problems, and a convergence test, data and performance profiles provide complementary information to assess the relative performance among the different algorithms in S when applied to solve problems in P. Specifically, data profiles allow gaining insight on the percentage of problems that are solved (according to the convergence test) by each algorithm within a given budget of function evaluations, while the performance profiles allow assessing how well an algorithm performs with respect to the other. For each $s \in S$ and $p \in P$, the number of function evaluations required by algorithm s to satisfy the convergence condition on problem p is denoted as $t_{p,s}$. Given a tolerance $0 < \tau < 1$ and denoted f_L as the lowest objective function value computed by any algorithm on problem p within a given number of function evaluations, the convergence test is

$$f(x_k) \leq f_L + \tau(f(x_0) - f_L),$$

which requires the best point to achieve a sufficient reduction from the value $f(x_0)$ of the objective function at the starting point. Note that the larger the value of the tolerance τ is, the higher accuracy is required at the best point. Performance and data profiles of solver s can be formally defined as follows

$$\begin{aligned} \rho_s(\alpha) &= \frac{1}{|P|}\left|\left\{p \in P : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in S\}} \leq \alpha\right\}\right|, \\ d_s(\kappa) &= \frac{1}{|P|}\left|\{p \in P : t_{p,s} \leq \kappa(n_p+1)\}\right|, \end{aligned}$$

where n_p is the dimension of problem p. While α indicates that the number of function evaluations required by algorithm s to achieve the best solution is α–times the number of function evaluations needed by the best algorithm, k denotes the number of simplex gradient estimates, with n_p+1 being the number associated with one simplex gradient. Important features for the comparison are $\rho_s(1)$, which is a measure of the efficiency of the algorithm, since it is the percentage of problems for which the algorithm s performs the best, and the height reached by each profile as the value of α or k increases, which measures the robustness of the algorithm.

Figure 4.1 reports the performance profiles corresponding to two levels of accuracy, i.e., $\tau = 10^{-1}$ and $\tau = 10^{-3}$. It can be observed that in both cases DFNDFL and DFL *box* perform the best in terms of efficiency. Indeed, they are the best algorithms for almost 55% of the problems in case of lower accuracy and 45% of the problems when higher accuracy is required. The performance difference between the two algorithms grows and then levels off as the value of α increases. The similar percentage of problems solved for $\alpha = 1$ is an important result for DFNDFL, since it shows that using more sophisticated directions than DFL *box* does not lead to a loss of efficiency. It is important to point out that the initial continuous and primitive search directions used by DFNDFL are the coordinate directions, which are employed in DFL *box*, thus leading to the same behavior of the algorithms in the first iterations. For each value of τ, despite the remarkable efficiency, DFL *box* does not show a strong robustness, which is significantly improved by DFNDFL. As for the other solvers, although NOMAD performs the best on a lower percentage of problems than DFNDFL and DFL *box*, it shows a robustness that is slightly better than DFNDFL when $\tau = 10^{-3}$. While NOMAD-NOMOD does not show a strong robustness when the level of accuracy is low, if $\tau = 10^{-3}$ the same robustness as DFNDFL is achieved as the value of α largely increases. Finally, although the efficiency of RBFOpt is higher than NOMAD-NOMOD when $\tau = 10^{-1}$, in case of higher accuracy, RBFOpt is outperformed by all the other algorithms.

Figure 4.2 reports the data profiles corresponding to the same levels of accuracy as before. Here, the behavior of the algorithms in terms of efficiency is similar. However, the percentage of problems solved by DFNDFL and DFL *box* grows rapidly as the number of simplex gradient estimates increases. If the user has a computational budget of simplex gradients greater than 25, only DFNDFL and the NOMAD algorithms are competitive. In particular, in case of low accuracy, DFNDFL shows the best robustness, which is reversed when $\tau = 10^{-3}$. However, in this latter case, DFNDFL is able to reach the same percentage of problems solved by NOMAD-NOMOD as the number of simplex gradients increases.

These numerical results show that DFNDFL demonstrates a remarkable efficiency and compares favorably to the state-of-the-art-solvers in terms of robustness, thus confirming and strengthening the properties of DFL *box* and providing a noticeable contribution to the derivative-free optimization solvers.

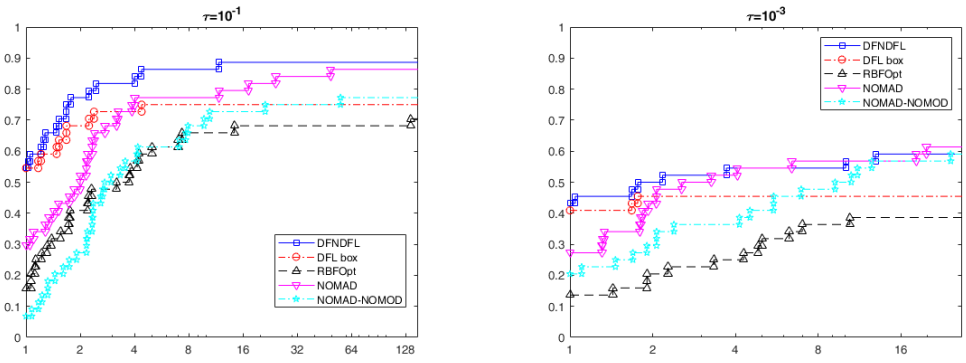


Figure 4.1. Performance profiles for the comparison among DFNDFL, DFL *box*, RBFOPT, NOMAD, and NOMAD-NOMOD on the 44 bound constrained problems.

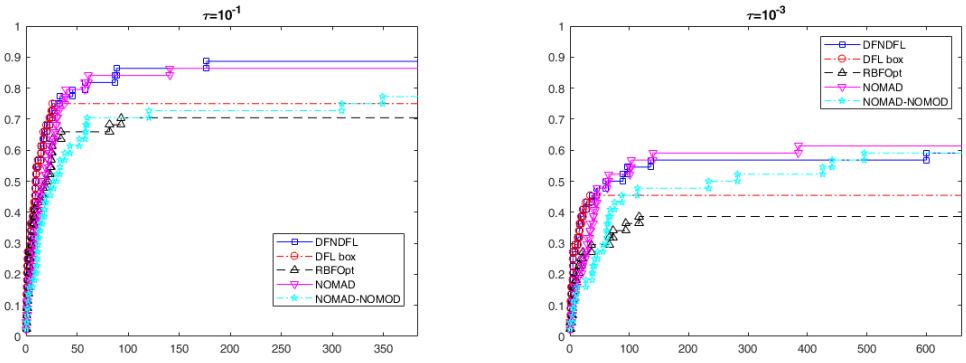


Figure 4.2. Data profiles for the comparison among DFNDFL, DFL *box*, RBFOPT, NOMAD, and NOMAD-NOMOD on the 44 bound constrained problems.

Chapter 5

Simulation-based optimization problems for discrete event simulation models of emergency departments

Modeling the arrival process to an ED is the first step of all studies dealing with the patient flow within the ED. Since DES models are often adopted with the aim to assess solutions for reducing the impact of the overcrowding problem, proper nonstationary processes are taken into account to reproduce time–dependent arrivals. Accordingly, an accurate estimation of the unknown arrival rate is required to guarantee reliability of results. Moreover, DES models concerning EDs are frequently affected by data quality problems, thus requiring a proper estimation of the missing parameters.

In this chapter, SBO is used with a twofold goal: on the one hand, to determine the best piecewise constant approximation of the time–varying arrival rate function by finding the optimal partition of the 24 hours into a suitable number of nonequally spaced intervals; on the other hand, to estimate the incomplete data to be used for building the DES model by adopting a model calibration procedure. As regards the first goal, the objective function of the resulting problem includes a fitting error term, for attaining an accurate solution, and a penalty term, to select an adequate degree of regularity of the arrival rate estimated. Moreover, black-box constraints are adopted both to ensure the validity of the nonhomogeneous Poisson assumption on the arrival process, which is commonly adopted in the literature, and to prevent mixing overdispersed data for model estimation. As concerns the second goal, the objective function represents the deviation between simulation output and real data, while the constraints ensure that the response of the simulation is sufficiently accurate according to the precision required. Sections 5.1–5.2 illustrate the ideas underlying these two approaches. Experimental results are reported in Section 7.3 for the real case study considered.

5.1 Arrival process to an emergency department

The focus of this section is on a new modeling approach for ED patient arrival

Section 5.1 and some parts of Chapter 7 are based on A. De Santis, T. Giovannelli, S. Lucidi, M. Messedaglia, M. Roma, *Determining the optimal piecewise constant approximation for the*

process based on a piecewise constant approximation of the arrival rate accomplished with nonequally spaced intervals. This choice is suggested by the typical situation that occurs in EDs where the arrival rate is low and varying during the night hours, and it is higher and more stable in the daytime. It is worth noting that using a piecewise constant function for approximating the arrival rate function is usually required by the most common DES software packages when implementing ED patient arrivals process as a NHPP. To obtain an accurate representation of the arrival rate $\lambda(t)$ by a piecewise constant function $\lambda_D(t)$, a finer discretization of the time–domain is required during the night hours, as opposed to the daytime. For this reason, the method proposed aims to determine the best partition of the 24 hours into intervals that are not necessarily equally spaced.

The use of an optimization method for identifying stochastic processes characterizing the patient flow through an ED is not new in the literature, since the approaches developed in [153] and [106] are proposed to determine the optimal service time distribution parameters by using metaheuristics. However, the ED arrival process is not involved in the study and a significant margin of improvement is expected by the application of global convergent algorithm based on optimality guarantees. Therefore, to the best of the author's knowledge, the approach proposed in this section represents the first attempt to adopt an optimization method for determining the best stochastic model for the ED arrival process. In the previous work [59], a preliminary study is performed by following a similar approach. Here, with respect to [59], a significantly enhanced statistical model is considered and the better results on the case study reported in Section 7.3 confirm the effectiveness of the new methodology.

In building a statistical model of the ED patient arrivals, a natural way of defining a selection criterion is to evaluate the fitting error between $\lambda(t)$ and its approximation $\lambda_D(t)$. However, the true arrival rate is unknown. As opposed to the work by [139], in the approach proposed no analytical model is assumed for $\lambda(t)$, which is replaced by an "empirical arrival rate model" $\lambda_F(t)$ obtained by a sample approximation corresponding to the very fine uniform partition of the 24 hours into intervals of 15 minutes. On each of these intervals, the average arrival rate values are supposed to be estimated from experimental data obtained by collecting ED patient arrival times for the same day of the week over different weeks. Hence, any other $\lambda_D(t)$ corresponding to a grosser partition of the day must be compared to $\lambda_F(t)$. In other words, an optimization problem is solved to select the partition of the 24 hours that leads to the piecewise constant approximation of the arrival rate with the best fitting to the empirical model by using nonequally spaced intervals. This is accomplished through the fitting error term, which is included in the objective function of the resulting minimization problem. Moreover, an additional penalty term is included aiming to obtain a sufficient degree of regularity of the approximating function, the latter being measured by the sum of the squares of the jumps between the values of $\lambda_D(t)$ on adjacent intervals. The rationale behind this term is to avoid optimal partitions with an excessively rough behavior, namely few long intervals with high jumps.

To obtain reliable results, proper constraints must be considered. First, the length of each interval of the partition cannot be smaller than a fixed value (e.g., half an hour or one hour) to allow for a number of arrivals that is sufficient for statistical purposes. Moreover, for each interval,

- the CU KS test must be satisfied to support the NHPP hypothesis;

Nonhomogenuous Poisson Process rate of Emergency Department patient arrivals (under review). Preprint at https://arxiv.org/abs/2101.11138.

- the dispersion test must be satisfied to ensure that data over different weeks is not overdispersed and can be considered as a realization of the same process, without being affected by seasonal effects.

The resulting problem is a black-box constrained optimization problem[2] and a method from the class of DFO is considered for determining its optimal solution. In particular, the algorithmic framework proposed in [171], which is suited for handling black-box problems with integer variables, is adopted.

In Section 5.1.1, the statistical model used in the approach is described. Section 5.1.2 states the optimization problem considered. Instead, the results of an extensive experimentation are reported in Section 7.3.1 for the specific case study considered.

5.1.1 Statistical model

The arrival process at EDs is usually characterized by a strong within-day variation both in the arrival rate and interarrival times: typically, experimental data shows rapid changes in the number of arrivals during the night hours, as opposed to a smoother profile in the daytime. For this reason, the ED arrival process is usually modeled as a NHPP.

Since no analytical model is available for the arrival rate $\lambda(t)$, a suitable representation of the unknown function is required. A realistic representation can be obtained by averaging the number of arrivals observed in experimental data on suitable intervals over the 24 hours of the day, not necessarily equally spaced. Let $\{T_i\}$ denote a partition P of the observation period $T = [0, 24]$ (hours) in N intervals, and let $\{\lambda_i\}$ be the corresponding sample average rates. Then a piecewise constant approximation of $\lambda(t)$ is written as follows

$$\lambda_D(t) = \sum_{i=1}^{N} \lambda_i \, \mathbf{1}_{T_i}(t), \quad t \in T \tag{5.1.1}$$

where $\mathbf{1}_{T_i}(t)$ is the indicator function of set T_i, i.e., it is 1 for $t \in T_i$ and 0 otherwise. Any partition P gives rise to a different approximation $\lambda_D(t)$, depending on the number of intervals and their lengths. Therefore, a criterion is needed to select the best partition $P^\star$ with some desirable features.

First of all, it is important to ensure that the arrival data is not overdispersed through the commonly used *dispersion test* proposed in [132] and reported in [139]. If it is satisfied, then it is possible to combine arrivals for the same day of the week over different weeks. To this end, for any partition P, let $\{k_i^r\}$ denote the number of arrivals in the i-th partition interval T_i in the r-th week, with $r \in \{1, \ldots, m\}$. Given the statistics

$$Ds_i = \frac{1}{\mu_i} \sum_{r=1}^{m} \left(k_i^r - \mu_i\right)^2, \text{ with } i = \{1, \ldots, N\},$$

[2]It is important to point out that the approach here proposed to study the ED arrival process does not properly belong to the class of SBO methods since a black-box is used in place of a simulation model. However, the stochastic SBO problems arising throughout this thesis are solved by applying the Sample Average Approximation (SAA) approach, which transform a stochastic problem into its deterministic counterpart (as described in Section 2.2), thus allowing for the use of methods from DFO. Therefore, the same methodology is applied to solve the black-box optimization problem described in this first section. Moreover, in addition to these motivations, the expression SBO is used because the black-box program adopted in this approach may be viewed as a deterministic simulation.

where $\mu_i = \frac{1}{m}\sum_{r=1}^{m} k_i^r$ is the average number of arrivals in the given interval for the same day of the week over the m weeks considered. Under the null hypothesis that the counts $\{k_i^r\}$ are a sample of m independent Poisson random variables with the same mean count μ_i (i.e., no overdispersion), then Ds_i is distributed as χ^2_{m-1}, the chi–squared distribution with $m-1$ degrees of freedom. Therefore, the null hypothesis is accepted with $1-\alpha$ confidence level if

$$Ds_i \leq \chi^2_{m-1,\alpha}, \text{ with } i = \{1, \ldots, N\}, \tag{5.1.2}$$

where $\chi^2_{m-1,\alpha}$ is the α level critical value of the χ^2_{m-1} distribution.

Furthermore, the partition is *feasible* if data is consistent with NHPP. Namely, if we denote by k_i the number of arrivals in each interval $T_i = [a_i, b_i)$ obtained by considering data of the same weekday, in the same interval, over m weeks, i.e., $k_i = \sum_{r=1}^{m} k_i^r$, with $i = \{1, \ldots, N\}$, the partition is feasible if each k_i has a Poisson distribution with rate λ_i obtained as $\mu_i/(b_i - a_i)$. To check the validity of the Poisson hypothesis, the CU KS test can be performed (see [41, 139]). It is preferable to use CU KS with respect to Lewis KS test since the latter is highly sensitive to rounding of the data and, moreover, the former has more power against alternative hypotheses involving exponential interarrival times (see [140] for a detailed comparison between the effectiveness of the two tests).

To perform CU KS test, for any interval $T_i = [a_i, b_i)$, let t_{ij}, with $j = \{1, \ldots, k_i\}$, be the arrival times within the i-th interval obtained as union over the m weeks of the arrival times in each T_i. Now consider the rescaled arrival times defined by $\tau_{ij} = \dfrac{t_{ij} - a_i}{b_i - a_i}$. The rescaled arrival times, conditionally to the value k_i, are a collection of i.i.d. random variables uniformly distributed over $[0, 1]$. Hence, in any interval the theoretical cumulative distribution function $F(t) = t$ is compared with the empirical cumulative distribution function

$$F_i(t) = \frac{1}{k_i}\sum_{j=1}^{k_i} \mathbf{1}_{\{\tau_{ij} \leq t\}}, \qquad 0 \leq t \leq 1.$$

The test statistic is defined as follows

$$D_i = \sup_{0 \leq t \leq 1} (|F_i(t) - t|). \tag{5.1.3}$$

The critical value for this test is denoted as $T(k_i, \alpha)$ and its values can be found on the KS test critical values table. Accordingly, the Poisson hypothesis is accepted if

$$D_i \leq T(k_i, \alpha), \text{ with } i \in \{1, \ldots, N\}. \tag{5.1.4}$$

This means that the CU KS test has to be satisfied on each interval T_i to qualify the partition P given by $\{T_i\}$ as feasible.

A further restriction is imposed on the feasible partitions. Given the experimental data, realistic partitions can not have a granularity too fine to avoid that some k_i being too small may unduly determine the rejection of the CU KS test. To this end, a suited lower threshold value for the interval length must be chosen, taking into account the specific case study considered.

Now let us evaluate the feasible partitions also in terms of the characteristics of function $\lambda_D(t)$. It would be amenable to define a fitting error with respect to $\lambda(t)$, which unfortunately is unknown. The problem can be resolved by considering a piecewise constant approximation $\lambda_F(t)$ over a very fine partition P_F of T. A

set of 96 equally space intervals of 15 minutes is considered and the corresponding average rates λ_i^F are supposed to be estimated from data. The function $\lambda_F(t)$ can be considered as an *empirical arrival rate model*. Note that partition P_F does not need to be feasible since it is only used to define the finest piecewise constant approximation of $\lambda(t)$. Therefore the following fitting error can be defined

$$E(P) = \sum_{j=1}^{N} \sum_{j=1}^{N_j} (\lambda_j - \lambda_{i_j}^F)^2 \tag{5.1.5}$$

where N_j is the number of intervals of 15 minutes contained in T_j, which are identified by the set of indexes $\{i_j\} \subset \{1, \ldots, 96\}$.

Finally, it is also advisable to characterize the "smoothness" of any approximation $\lambda_D(t)$ to avoid very gross partitions with high jumps between adjacent intervals. This is accomplished by means of the mean squared error

$$S(P) = \sum_{j=2}^{N} (\lambda_j - \lambda_{j-1})^2. \tag{5.1.6}$$

In Section 5.1.2, the model features illustrated above are organized in a proper optimization procedure that provides the selection of the best partition according to conflicting goals.

The approach proposed enables addressing the two issues, raised in [139], that arise when dealing with modeling ED patient arrivals, namely the *choice of the intervals* and the *overdispersion*. As concerns the third issue, the *data rounding*, it depends on the specific case study considered. In Section 7.3, the arrival times in the collected data are rounded to seconds (format `hh:mm:ss`), and actually occurrences of simultaneous arrivals which would cause zero interarrival times are not present. Therefore, the unrounding procedure is not needed. Moreover, as already pointed out above, the CU KS test is not very sensitive to the data rounding.

5.1.2 Statement of the black-box optimization problem

Any partition $P = \{T_i\}$ of $T = [0, 24]$ is characterized by the boundary points $\{x_i\}$ of its intervals and by their number N. Let us introduce a vector of variables $x \in \mathbb{Z}^{25}$ such that

$$T_i = [x_i, x_{i+1}),$$

with $i \in \{1, \ldots, 24\}$, $x_1 = 0$, and $x_{25} = 24$.

Functions in (5.1.5) and (5.1.6) are indeed functions of x and, therefore, will be denoted by $E(x)$ and $S(x)$, respectively. Therefore, the objective function that constitutes the selection criterion is given by

$$f(x) = E(x) + wS(x), \tag{5.1.7}$$

where $w > 0$ is a parameter that controls the weight of the smoothness penalty term with respect to the fitting error: the larger w, the smaller the difference between average arrival rates in adjacent intervals; this in turn implies that on a steep section of $\lambda_F(t)$, an increased number of shorter intervals is adopted to fill the gap with relatively small jumps.

The set $\mathcal{P}$ of feasible partitions is defined as follows:

$$\begin{aligned} \mathcal{P} = \Big\{ x \in \mathbb{Z}^{25} \mid\ & x_1 = 0, \quad x_{25} = 24, \quad x_{i+1} - x_i \geq \ell_i, \quad g_i(x) \leq 0, \\ & h_i(x) \leq 0, \text{ with } i \in \{1, \ldots, N\} \Big\} \end{aligned} \tag{5.1.8}$$

where

$$\ell_i = \begin{cases} 0 & \text{if} \quad x_i = x_{i+1}, \\ \ell & \text{otherwise}, \end{cases} \tag{5.1.9}$$

$$g_i(x) = \begin{cases} 0 & \text{if} \quad x_i = x_{i+1}, \\ D_i - T(k_i, \alpha) & \text{otherwise}, \end{cases} \tag{5.1.10}$$

$$h_i(x) = \begin{cases} 0 & \text{if} \quad x_i = x_{i+1}, \\ Ds_i - \chi^2_{m-1,\alpha} & \text{otherwise}, \end{cases} \tag{5.1.11}$$

with $i \in \{1, \ldots, N\}$. The value ℓ in (5.1.9) denotes the minimum interval length allowed and it is assumed that $\ell \geq 1/4$. Of course, constraints $g_i(x) \leq 0$ represent the satisfaction of the CU KS test in (5.1.4), while constraints $h_i(x) \leq 0$ concern the dispersion test in (5.1.2). Therefore, the best piecewise constant approximation $\lambda^\star_D(t)$ of the time–varying arrival rate $\lambda(t)$ is obtained by solving the following black-box optimization problem

$$\begin{aligned} \max \quad & f(x) \\ s.t. \quad & x \in \mathcal{P}. \end{aligned} \tag{5.1.12}$$

It is worth pointing out that the idea of using as a constraint of the optimization problem a test to validate the underlying statistical hypothesis on data, along with a dispersion test, is completely novel in the framework of modeling ED patient arrivals. The only proposal that uses a similar approach is in the previous paper [59].

It is important to note that in (5.1.7) the objective function does not have an analytical structure with respect to the independent variables and it can only be computed by a data–driven procedure once the x_i's values are given. The same is true for the constraints $g_i(x)$ and $h_i(x)$ in (5.1.8). Therefore, the problem in hand is an integer nonlinear constrained black-box optimization problem, and both objective and constraint functions are relatively expensive to compute, thus giving rise to a problem difficult to be efficiently solved. In fact, classical optimization methods neither can be applied (since based on the analytic knowledge of the functions involved) nor they are efficient, especially when evaluating the functions is computationally expensive. Therefore, to tackle problem (5.1.12), the attention is turned to the class of DFO and black-box methods, as discussed in Section 7.3.

5.2 Model calibration of emergency department simulation models

The approach proposed in this section aims to handle a common problem that is caused by the way used to collect the data related to the patient flow within the ED, namely the problem of missing timestamps, which affects many simulation models dealing with EDs, as evidenced by the papers in the specific literature (see Section 2.4.4). Among the sources of noise affecting the quality of the input data necessary to build a reliable simulation model, the problem of missing timestamps has a strong impact on the overall accuracy of the simulation, since it prevents

Section 5.2 and some parts of Chapter 7 are based on A. De Santis, T. Giovannelli, S. Lucidi, M. Messedaglia, M. Roma, *A simulation-based optimization approach for the calibration of a discrete event simulation model of an emergency department* (under review).

the knowledge of the values used to derive appropriate probability distributions. Therefore, while other issues may be resolved by either carefully cleaning the dataset or introducing assumptions on how to interpret the data, the unavailability of timestamps requires more sophisticated procedures.

Compared to the other papers dealing with missing data in ED datasets, the approach discussed in this section aims both to propose a new formulation of the resulting optimization problem and to improve on the optimization strategies typically used in the literature. Since building a simulation model is a process that requires a considerable amount of effort and thus is not expected to be completed in a short time, an exact algorithm providing optimal solutions may be preferable to metaheuristic procedures, whose final solutions are returned faster but without optimality guarantees. However, although global convergent algorithms appear to be a reasonable choice, metaheuristics are the methods mainly adopted in the literature dedicated to missing data (see, e.g., [153, 106]). To fill this gap, in this section a SBO approach is developed both to propose an alternative version of the optimization problem used in [106] and to use an optimization strategy based on global convergence, which allows the algorithm to find an optimal solution with optimality guarantees. Similarly to [153] and differently from [106], Weibull distributions are adopted to generate the values of the activity service times associated with missing data. Indeed, this probability distribution is considered suitable when data is unavailable (see, e.g., [161]). Moreover, the most critical patients, who are excluded from the two approaches developed by [153] and [106], are included to prevent the simulation model from returning inaccurate results for this important class of patients.

5.2.1 Data collection in emergency department

The timestamps defining the activities in the patient flow are represented in Figure 5.1 and described in Table 5.1. Although these timestamps are commonly recorded

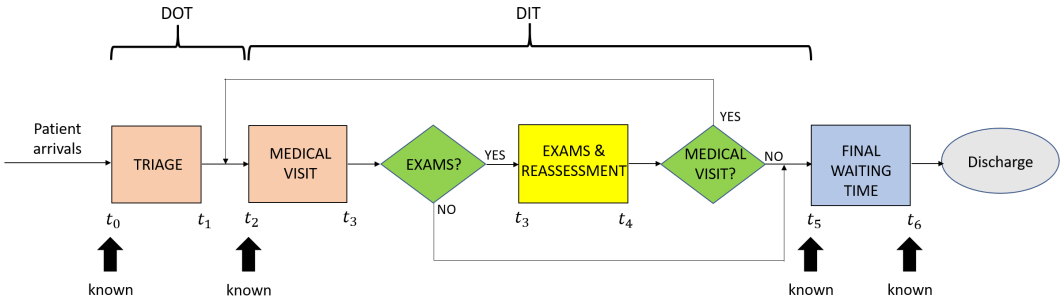


Figure 5.1. Timestamps collected throughout the patient flow.

Table 5.1. Description of the timestamps collected throughout the patient flow.

t_0	starting time of triage.
t_1	ending time of triage.
t_2	starting time of medical visit.
t_3	ending time of medical visit and starting time of examinations.
t_4	patient receives the latest medical report.
t_5	patient receives the last medical report.
t_6	patient is discharged and leaves the ED.

by the electronic systems used to collect data, the actual patient flow may include

extra times omitted from this scheme. For example, since the arrival time to the ED is usually not registered, before triage there may be an additional waiting time not included in any record. Moreover, extra waiting times and observation periods may be present both in the examination part and in the discharge phase. Since usually there is no available data for these times, in Figure 5.1 both waiting and service times for exams are included in a single box between t_3 and t_4, which are associated with the end of the medical visit and the availability of the latest medical report after a visit, respectively. After receiving the report, patients may be either subject to additional medical visits or discharged after a final waiting time, which is due to a further observation or to the time required by the ED staff to prepare the discharge. It is important to point out that the disposition, which refers to the decision to admit a patient to a hospital ward, is associated with t_6, like the discharge. This implies that the boarding time of the admitted patients (i.e., the waiting time before being transferred to the hospital ward) is assumed to start at t_6, thus being excluded from the ED scope. This choice is in accordance with the case study considered in Chapter 7. Contrarily, in some EDs the discharge of patients admitted may be considered as the time of admission to the hospital ward (see, e.g., [32]).

In most cases, the available timestamps are t_0, t_2, and t_6, which are marked in the scheme as *known*. Since this is a typical setting in practice, throughout this section they are supposed to be the only timestamps recorded by the ED along with t_5 (which may not be always available). As a consequence, for each patient it is possible to compute

- the *Door-to-Doctor Time* (**DOT**), which is the time difference between the start of the triage and the start of the medical visit, namely $t_2 - t_0$;
- the *Doctor-to-Discharge Time* (**DIT**), which is the time difference between the start of the medical visit and the discharge, namely $t_5 - t_2$ if t_5 is available, $t_6 - t_2$ otherwise.

It is important to remark that when patients require additional medical visits after the exams, DIT can be interpreted as the time difference between the timestamp of the last medical report (or the discharge if t_5 is not available) and the starting time of the first medical visit. Note that t_4 and t_5 are equal if a patient does not need additional medical visits.

Since in many cases only a subset of the timestamps related to the patient flow are known [226], the service time cannot be computed for all the activities. Apart from the final waiting time, which can be recovered for each patient through the difference $t_6 - t_5$, note that the durations of all the other activities are not completely defined. In fact, in case of triage and medical visit, the ending time is missing, while for exams both timestamps are unknown.

5.2.2 Statement of the simulation-based optimization problem

The goal of the approach proposed is to recover the information needed to build an accurate simulation model by leveraging the known information through the minimization of the deviation between real data and simulation output. In order to define the simulation-based optimization problem, let us introduce the sets below.

- Let C be the set of the triage tags.
- Let $U(c)$ be the set of units where patients with tag $c \in C$ can be visited and treated.

- Let $\mathcal{T} = \{DOT, DIT\}$ be the set of the time differences considered.

In the absence of data, suitable probability distributions are Weibull and lognormal [161]. Let us arbitrarily consider a Weibull distribution, whose probability density function is

$$f(w) = \begin{cases} \alpha\beta^{-\alpha} w^{\alpha-1} e^{-(w/\beta)^{\alpha}} \text{ if } w > 0, \\ 0 \text{ otherwise,} \end{cases}$$

where $\alpha > 0$ and $\beta > 0$ are the shape and the scale parameters, respectively. Therefore, a Weibull distribution is assumed for the triage, medical visit and exams service times and, for each of them, different pairs of shape and scale parameters are considered based on the triage tag c and on the unit u. This choice leads to a number of different pairs of parameters equal to $\sum_{c \in C} |U(c)|$, where $|U(c)|$ refers to the cardinality of the set $U(c)$. Let us denote as $x \in \mathbb{R}^{n_1}$, $y \in \mathbb{R}^{n_2}$, and $z \in \mathbb{R}^{n_3}$ the corresponding probability distribution parameters for the service times considered, where $n_v \leq 2 \sum_{c \in C} |U(c)|$ with $v \in \{1, 2, 3\}$. In particular, for all $c \in C$ and for all $u \in U(c)$, the shape and scale parameters of the Weibull distributions are

- x_1^{cu} and x_2^{cu} for the triage probability distribution,
- y_1^{cu} and y_2^{cu} for the medical visit probability distribution,
- z_1^{cu} and z_2^{cu} for the exams probability distribution.

Let F_{cui}^{sim} and F_{cui}^{real} be the empirical cumulative distribution functions of the values of the simulated and real time difference $i \in \mathcal{T}$ for patients with tag c visited in unit u. Moreover, let k_{cui}^{sim} and k_{cui}^{real} be the number of such patients from the simulation and from the real dataset. Hence, for all $j \in \{sim, real\}$, we have that

$$F_{cui}^{j}(t) = \frac{1}{k_{cui}^{j}} \sum_{h=1}^{k_{cui}^{j}} \mathbf{1}_{\{\tau_{cuih} \leq t\}} \quad \text{with } t \geq 0,$$

where τ_{cuih} is the value of the time difference i recorded for the h–patient, with $h \in \{1, \ldots, k_{cui}^{j}\}$, considered from $j \in \{sim, real\}$. It is important to point out that the values τ_{cuih} of the time differences computed from the simulation depend on the service times of triage, medical visit, and exams drawn from the Weibull distributions described above. Therefore, this dependence can be written explicitly as $F_{cui}^{sim}(t; x, y, z)$, where x, y, and z are the vectors containing all the associated shape and scale parameters.

The mathematical problem formulation is reported as follows

$$\begin{aligned} &\min_{x,y,z} \sum_{c \in C} \sum_{u \in U(c)} \sum_{i \in \mathcal{T}} \Big(\int_0^{\infty} (F_{cui}^{sim}(t; x, y, z) - F_{cui}^{real}(t))^2 \, dt \Big) \\ &s.t. \;\; x \in \mathcal{P}, \end{aligned} \tag{5.2.1}$$

where the feasible set

$$\mathcal{P} = \Big\{(x,y,z) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \mid \\ g_{cui}(x,y,z) \leq 0, \\ h_{cui}(x,y,z) \leq 0, \\ l_x \leq x \leq u_x \\ l_y \leq y \leq u_y \\ l_z \leq z \leq u_z \\ \text{for all } c \in C,\, u \in U(c),\, i \in \mathcal{T}\Big\}$$

is defined by the following functions

- $g_{cui}(x,y,z) = \left|\frac{\mu^{sim}_{cui}(x,y,z) - \mu^{real}_{cui}}{\mu^{real}_{cui}}\right| - tol^{cui}_{\mu}$, which compares the sample means $\mu^{sim}_{cui}(x,y,z)$ and μ^{real}_{cui} of the time difference i computed through the simulated (by averaging over the independent replications) and real data, respectively;
- $h_{cui}(x,y,z) = \left|\frac{\sigma^{sim}_{cui}(x,y,z) - \sigma^{real}_{cui}}{\sigma^{real}_{cui}}\right| - tol^{cui}_{\sigma}$, which compares the sample standard deviations $\sigma^{sim}_{cui}(x,y,z)$ and σ^{real}_{cui} of the time difference i computed through the simulated (by averaging over the independent replications) and real data, respectively;

and l_x, l_y, l_z, u_x, u_y, and u_z are vectors defining bound constraints. Positive thresholds, namely tol^{cui}_{μ} and tol^{cui}_{σ}, are used to state the degree of accuracy required for the simulation model. The objective function is the sum of the integrals of the squared difference between F^{sim}_{cui} and F^{real}_{cui} over the sets C, $U(c)$, and $\mathcal{T}$. The decision variables are the Weibull distribution parameters contained in the vectors x, y, and z. For the sake of simplicity, the previous formulation does not include variables that are not parameters of probability distributions, although they may be present. It is important to remark that in the general framework described, the dependence of each pair of parameters on both c and u implies that the triage service time is affected by the triage tag and the ED unit. This assumption, which turns out to be reasonable when applied to the duration of medical visit and exams, may lead to an excessive number of variables for the triage service time, if it does not hold. Indeed, while the color tag significantly affects the triage duration since urgent patients undergo a faster triage than less critical patients, the impact of the unit may be negligible. However, as concerns the case study described in Chapter 7, interviews with the ED staff have shown that slightly different procedures are adopted by the nurses in charge of triage based on the unit where a patient is assigned. This means that final conclusions should be drawn after analyzing the specific system considered. Since the probability distribution parameters affect the dimension of the optimization problem, avoiding unnecessary variables allows the algorithm to benefit from a lower computational cost.

Chapter 6

Case study: emergency department of Engles Profili hospital

A direct consequence of the ED overcrowding is a long wait for visit and treatment of people who require primary care, which may possibly endanger the lives of critical patients. Among the many approaches proposed in the healthcare management literature to address this problem, less attention is given to patient peak arrivals caused by the occurrence of critical events that put a strain on the operational efficiency of an ED. In this chapter, the effects of such an occurrence are studied for a medium–size ED located in a region of Central Italy recently hit by a severe earthquake. In particular, a DES model is proposed to analyze the patient flow through this ED, aiming to simulate unusual operational conditions due to a critical event, like a natural disaster, that causes a sudden spike in the number of patient arrivals. The availability of detailed data concerning the ED processes enables building an accurate DES model and performing extensive scenario analyses. The model provides a valid decision support system for the ED managers also in assessing specific emergency plans to be activated in case of mass casualty disasters.

6.1 Purpose of the analysis

In this chapter, a DES-based approach is used for analyzing the operations of an ED of a medium–size hospital located in an Italian region where the recent earthquakes have put a strain on the EDs of the area. For such an ED it is crucial to assess the impact of unusual/critical events that cause peak arrivals, in order to design suited contingency plans. Therefore, the effects of spikes in the number of patient arrivals on the ED operations are studied.

The availability of detailed data concerning the ED processes allows building an accurate DES model, well reproducing the actual ED operating modes. After a complete input analysis and an accurate construction of a conceptual model, the simulation model is implemented by using ARENA 16 Simulation Software [38, 135], which is one of the most commonly used general purpose DES packages. Based on flowchart modules, it enables building the simulation model and performing input analysis, simulation runs, and output analysis. The model is verified and validated

This chapter is based on G. Fava, T. Giovannelli, M. Messedaglia, M. Roma, *Effect of different patient peak arrivals on an emergency department via discrete event simulation* (under review).

in order to guarantee that it is an accurate representation of the system under consideration. Moreover, several scenario analyses are performed, aiming to evaluate the impact of possible changes in the ED operating conditions. Therefore, the main ED KPIs are assessed under different critical scenarios. In particular, since patient peak arrivals can be of different patterns, some artificial scenarios are created trying to reproduce situations really occurred that seriously affect the ED operations. The model proposed is helpful to the ED managers both for the daily management of the resources and for the assessment of suited emergency plans to be adopted in case of critical events.

This chapter is organized as follows. Section 6.2 describes the case study of the ED considered. In Section 6.3 the DES model is detailed, along with the input analysis and the model verification and validation performed. Finally, Section 6.4 reports results for the "as–is" status along with extensive scenario analyses, mainly focused on patient peak arrivals.

6.2 Description of the patient flow in the emergency department

This section describes the case study concerning the ED of the "E. Profili" Fabriano (Ancona) hospital. This hospital is located in the Italian region of Marche and the catchment area covers about 48000 inhabitants. Since every year about 27000 patients arrive to the ED requiring medical assistance, it can be considered of medium dimension. A detailed understanding of the ED operations has been gained through process mapping performed along with the ED staff. In the sequel, a brief description of the ED rooms and staff is reported; moreover, the patient flows through the ED is summarized. This ED is composed by

- a *triage area*, where a nurse assigns the colour tag at each incoming patient, one at a time;
- a *waiting room*, where patients either wait in line for the triage or wait for the medical examination (after the triage);
- three areas for medical treatment:

  Area A: the shock room for red tagged patients;

  Area B: the room for green and white tagged patients;

  Area C: the room for yellow tagged patients.
- an *holding area*;
- a *Short Stay Unit (SSU)*.

As regards the areas for the medical treatment, *Area A* (the shock room), which is the most equipped one, can host two critical patients simultaneously. Conversely, in *Area B* and in *Area C* one seat is available. During the night (9.00 p.m. – 8.00 a.m.), only *Area A* and *Area B* are in operation, so that also yellow tagged patients are visited in *Area B*. As regards the ED staff, physicians and nurses are on duty according to the shifts reported in Tables 6.1 and in Table 6.2. As regards the patient flow, arrivals are by ambulance or autonomously. All the incoming patients are registered at the check–in desk and then admitted to the triage area, where a nurse collects patient's health information and assigns the color tag. Critical patients arriving

Table 6.1. Number of physicians on duty in the weekdays (WD) and in the public holidays (PH).

	WD	PH
Morning (8.00 a.m. – 2.00 p.m.)	2	1
Afternoon (2.00 p.m. – 9.00 p.m.)	2	1
Night (9.00 p.m. – 8.00 a.m.)	1	1

Table 6.2. Number of nurses on duty each day.

Morning (7.00 a.m. – 2.00 p.m.)	3
Afternoon (2.00 p.m. – 10.00 p.m.)	3
Night (10.00 p.m. – 7.00 a.m.)	2

by ambulance are directly transferred to a medical area for immediate treatment, without going through the triage area. After triage, a patient waits for the call in the waiting room, where the estimated waiting time is displayed on a screen. Then the patient is transferred to an appropriate area inside the ED for medical visit according to the assigned color tag. In severely urgent cases (red tag), the patient is examined in the shock room (*Area A*) for possible immediate treatments. In less severe cases, physicians, after performing health assessment, decide the clinical pathway which must be followed by the patient, possibly changing the color tag assigned at triage. The pathways can be very differentiated on the basis of the acuity of patient's illness. In many cases, the physician requires additional examinations for the patient (e.g., clinical laboratory tests, X-ray, and electrocardiogram). In other cases, the patient is transferred to the SSU for a short observation. Moreover, patients having a less serious illness are assigned to the fast track service, namely a specific area of the ED provided by a multidisciplinary team, where timely treatment and discharge are ensured. In the ED, usually one physician and one nurse manage the examination and the treatment of a patient. However, in case of severe injuries, they are supported by the whole ED staff.

Whenever all the examinations are completed and the related reports have been issued, a reassessment of the patient is performed by the physician, who can require further examinations and/or an additional observation period. At the end of the pathway, the final diagnosis is delivered and the patient is discharged from the ED with an exit code corresponding to the outcome. Specifically, a detailed description of the outcome must be issued to identify the patient's clinical severity level. The outcome is encoded according to the following list:

O1: patient is discharged home;

O2: patient is discharged home with reliance to outpatient facilities or family physician;

O3: patient is hospitalized at a hospital ward;

O4: patient is transferred to another hospital due to bed unavailability at the appropriate ward of the hospital;

O5: patient refuses hospitalization and leaves the ED despite the medical request;

O6: patient leaves during examinations, i.e. the patient does not complete all the required tests and abandons the ED without informing the staff;

O7: patient leaves without being seen (LWBS), i.e. the patient abandons the ED waiting room before being examined by a physician;

O8: patient dies during the stay at the ED;

O9: patient arrives deceased at the ED.

In order to perform a complete process mapping of the ED, many interviews have been carried out to the staff (physicians, nurses, and managers) and direct observations have been used. Moreover, all the available data concerning patient flow during February 2018 have been anonymously collected. February has been chosen because, according to interviews to the ED staff, it is one of the most representative months regarding the functioning of the ED. Indeed, it is a month that includes requests related to the winter season, not affected by holidays (such as December and January), thus reflecting a standard workload of the ED. If data is available over several months, the simulation model can be easily extended to take into account different seasonal patterns. However, this work is focused on the effect of a sudden peak of the patient arrival rate, which can occur at any time throughout the year, overlapping the standard patient arrivals. Especially in the extremely loaded scenario described in Section 6.4, if the sudden increase in the arrival rate is considerably large, the results are expected to be slightly affected by changes in the standard patient arrival rate due to seasonality.

From 00:00 of February 1 to 23:59 of February 28, 2018, the overall number of patients arrived to the ED is 2046. The time–stamps recorded are reported in Figure 6.1. They have been extracted and organized in a suited database. Note that, since the *holding area* and the *SSU* are not subject to our study, these two ED units are not considered in the simulation model and, hence, not represented in Figure 6.1. In Table 6.3, the number and the percentage of color tags assigned at

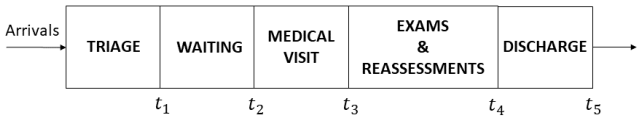


Figure 6.1. Collected timestamps for the ED process.

triage are reported. Moreover, Table 6.3 reports also the number and the percentage of color tags at discharge. As already noticed, in some cases, at the end of the clinical pathway the patient tag can be different with respect to the tag assigned at triage, since it can be changed by a physician during the visit. In the same table, the percentage of patients who leave without being seen (LWBS) is reported for each triage tag. Moreover, the number and the percentage of deceased patients are included in the table as well. A table with all the tag changes is not reported for the sake of brevity, but by observing Table 6.3, it is clear that some color tags may be changed to another tag (always an adjacent tag).

In Table 6.4, the distribution of patients is reported based on the tag at discharge and according to the list of outcomes.

6.3 The discrete event simulation model

This section describes the DES model of the ED under study. As regards the structure of the model, each area for medical treatment (*Area A*, *Area B*, and *Area C*) is represented by a submodel grouping the related modules; the *triage area* is

Table 6.3. Number and percentage of color tags assigned at triage to the incoming patients (columns 2-3). Number and percentage of color tags at discharge (columns 4-5). Percentage of patients LWBS (column 6).

	TRIAGE TAGS		TAGS on DISCH.		LWBS
WHITE	149	7.28%	171	8.36%	4.65%
GREEN	1448	70.77%	1612	78.78%	1.61%
YELLOW	434	21.21%	243	11.88%	0.41%
RED	15	0.74%	18	0.88%	-
Deceased			2	0.1%	
	2046		2046		

Table 6.4. Distribution of patients based on the tag at discharge and according to the list of outcomes.

	WHITE	GREEN	YELLOW	RED	tot.
O1	121	1025	23		1169
O2	39	496	21		556
O3		29	182	14	225
O4			4	4	8
O5		19	10		29
O6	3	17	2		22
O7	8	26	1		35
O8				2	2
O9					-
tot.	171	1612	243	20	2046

represented by a process with a single server and a single queue and the *waiting room* is handled as a queue with priority (with infinite capacity).

The model *entities* are patients: after being created on patient arrival, an entity flows through the different segments of the model according to specified logical rules, which enable reproducing the proper patient flow. The *resources* are the physicians, the nurses (whose availability is based on the schedules reported in Tables 6.1 and 6.2) and the seats at each ED area (fixed capacity resource).

A brief description of the flows through the model is the following: as soon as an entity is created, a check is performed to verify if the patient arrives deceased (outcome *O9*); in this case the entity is removed from the model. If an entity represents a critical (red tagged) patient arriving by ambulance, then it is directly sent to the shock room (*Area A*) for immediate care, and the corresponding resources (one physician, one nurse, and one seat in the area) are seized. Otherwise, the entity undergoes the triage process and a color tag is assigned as an entity attribute. Then the entity joins the queue of the waiting room and waits for the visit. During the wait, an entity may possibly leave (outcome *O7*). Entities waiting for the visit are selected from the queue based on the priority (the color tag) and FIFO criterion is used within each priority class, i.e., for entities with the same color tag. At the beginning of the visit, one physician, one nurse, and one seat of the corresponding area are seized. The service time of the visit is assigned by the probability distributions reported in Table 6.5, depending on the entity color tag. During the visit, the Diagnostic Therapeutic Care Path (DTCP) for the patient is decided, possibly changing the entity color tag assigned at triage. A change of the color tag implies that in all the downstream modules the entity is handled according to the new color tag. After the visit, an entity can be sent to many different segments of the model. In less severe

cases, an entity leaves the system (the patient is discharged) for outcomes *O1* and *O2*.

The phases following the visit are additional examinations and reassessments, which are represented by a single delay process, whose overall service time is given by means of the probability distributions reported in Table 6.5. Note that all possible further treatments are considered in this module, since the service time also includes duration of all treatments and reassessments after visit. At the end of the DTCP, the entity is discharged from the simulated system, according to the proper outcome. It is important to point out that the choice of using a single process module to represent additional examinations and reassessments is motivated by the availability of only timestamps t_3 and t_4 (see Figure 6.1). This reflects the well-known difficulty in modeling such phases due to both their high variability and lack of the specific timestamps of the activities involved. A simplified logic diagram of the simulation model is reported in Figure 6.2. The simulation model has been implemented by

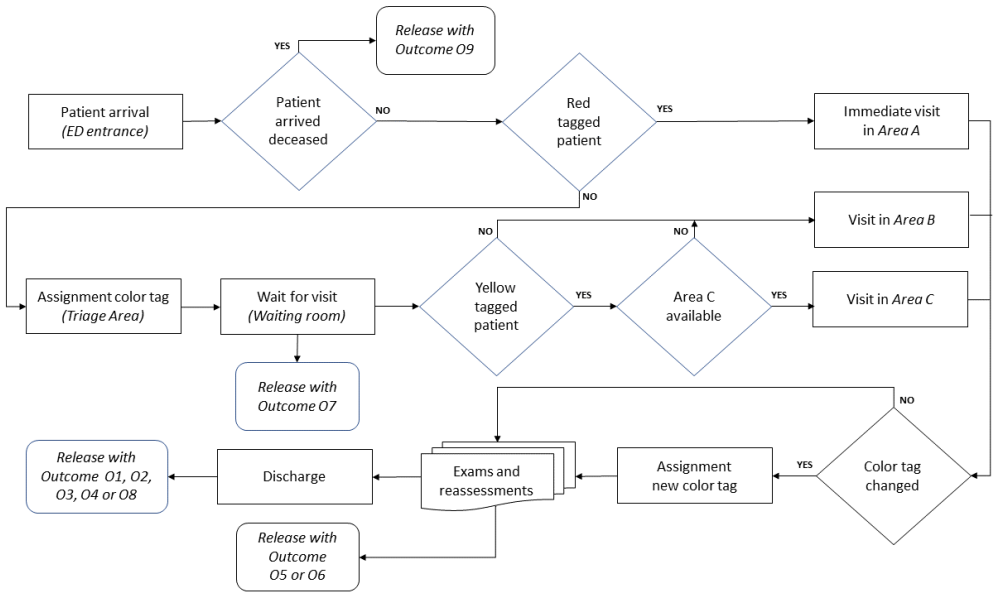


Figure 6.2. Conceptual representation of the patient flow. Release with outcome *O5* or *O6* in the "Exams and reassessments" block is possibly expected only for white, green, and yellow tag patients.

using ARENA 16 (64 bit) Simulation Software [38, 135].

As regards the KPIs of interest, to meet the specific demand of the ED managers, the analysis focuses on the patient flow starting from the end of the triage and, in particular, on monitoring, for each color tag,

- the *Waiting Time* (**WT**) between the end of the triage and the start of the visit, namely $t_2 - t_1$ in Figure 6.1;
- the *Total Time* (**TT**) after the triage, i.e. the time between the end of the triage and the discharge, namely $t_5 - t_1$ in Figure 6.1.

These indicators are very similar to "door-to-doctor time" and "door-to-disposition time" metrics used in literature (see, e.g., [32, 182]) with the only difference that triage time is not included in WT and TT. As already mentioned, this choice is motivated by the practitioners' request of focusing on all the processes following the

triage phase. Moreover, triage is typically carried out as soon as a patient arrives and triage service time is usually negligible with respect to waiting and treatment times. Therefore, similarly to door-to-doctor time, waiting time WT is a very important component of ED throughput, having strong implications for the percentage of patients who LWBS [48]. Furthermore, from a service quality point of view, WT significantly affects patient satisfaction. As concerns the total time TT, even if it does not coincide with LOS (since the initial part of the pathway until the end of the triage is not considered), it represents the most significant component of ED throughput, thus being mostly responsible for the ED crowding.

6.3.1 Input analysis

The data is used for a detailed input analysis of all the processes in the ED. As regards the arrival process, in accordance with the literature, the standard assumption that the arrival process to an ED is a NHPP is adopted [234, 153, 243, 7, 6, 105]. Indeed, the adoption of a nonhomogeneous process is necessary since patients interarrival times are strongly affected by the arrival hour. In order to obtain a good accuracy of the arrival rate, 24 time slots are considered for each day at an hourly basis, starting from 00:00. Therefore, by using a standard procedure [161], the arrival rate function is approximated by a piecewise constant function. A plot of the hourly arrival rate for each day of the week is reported in Figure 6.3. A within-day

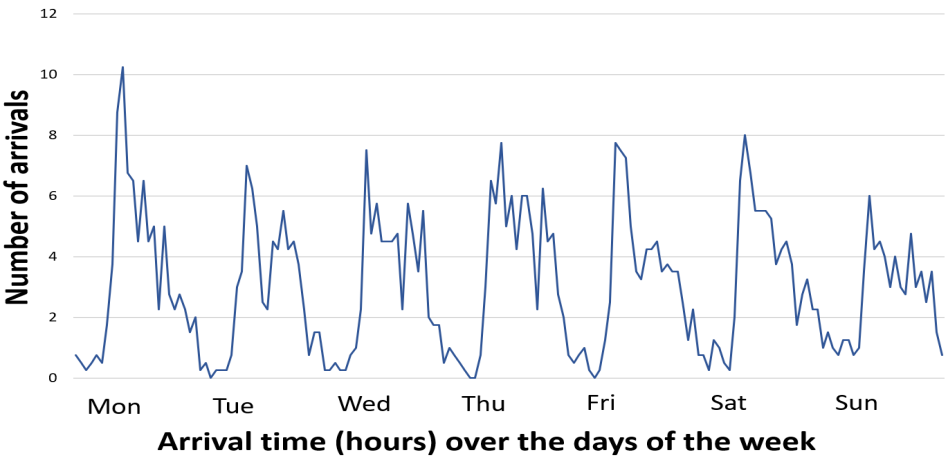


Figure 6.3. Plot of the hourly arrival rate over the days of the week.

and day-to-day variation in the number of arrivals is observed. In particular, the maximum value of the hourly arrival rate is attained on Monday, while Sunday is the day with the least number of arrivals. Moreover, a light increasing trend in the daily arrivals peak is observed from Tuesday to Saturday. On the basis of these considerations, the arrival rate is estimated by distinguishing among the different days of the week and in the simulation model an ARENA built-in tool [135] is used to generate the entity arrivals according to a nonstationary Poisson process with varying rate.

As regards the timing employed in the *visit* process and in the *additional examinations and reassessments* process, by using the collected data the probability distribution of the times (in minutes) is reported in Table 6.5 for each color tag.

Table 6.5. Probability distribution of *visit times* and *additional examinations and reassessments times.*

	VISIT	EXAMS & REASSES.
WHITE	Lognormal(7.87,9.77)	Weibull(23.5, 0.643)
GREEN	Lognormal(12.7,11.6)	Weibull(64.2, 0.549)
YELLOW	3 + Erlang(6.39, 3)	29 + Weibull(183, 0.635)
RED	11 + 39 Beta(0.673,1.30)	0.999 + Exp(69.3)

6.3.2 Model verification and validation

In order to guarantee that the DES model provides a sufficient accuracy of the output, the model has been widely verified and validated by using standard techniques. In particular, after a preliminary debugging, in order to determine whether all the logical paths have been correctly implemented, the simulation model has been run under several different settings of the input parameters, its functioning has been accurately checked, and the output of each run has been observed. Model trace has been also used for a deepened verification of the model.

As regards the model validation, the real system values have been compared with the corresponding simulation outputs, namely the average values (with their confidence interval) obtained from 50 independent simulation replications, each of them 35 days long, with a warm up period of 7 days. In this way, a fair comparison with data that refers to 28 days of February can be performed. In particular, some fundamental KPIs of the overall process are considered in terms of *times* and *entity counters*.

As concerns times, the focus is on the waiting time WT and on the total time TT previously defined. Figure 6.4 and Figure 6.5 report the current values, namely the values corresponding to the "as–is" status, and the simulation output of WT and TT (in minutes) with the relative confidence interval (with 95% confidence level). These plots evidence that the simulation output is a good approximation of the

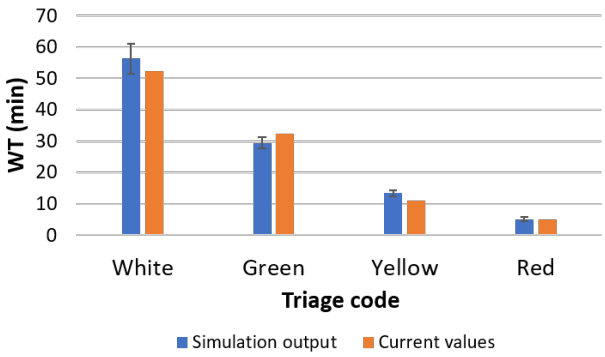


Figure 6.4. Plot of current values (in orange) and simulation output (in blue) of WT (in minutes) with the confidence interval.

real system values since for each color tag the current values are either within the corresponding confidence interval or close to it.

As regards the entity counters, the current values of the outcomes as reported in Table 6.4 are compared with the corresponding outputs of the simulation model reported in Table 6.6 (for the sake of brevity, the corresponding plots are omitted). A comparison between the two tables clearly evidences that the simulation model

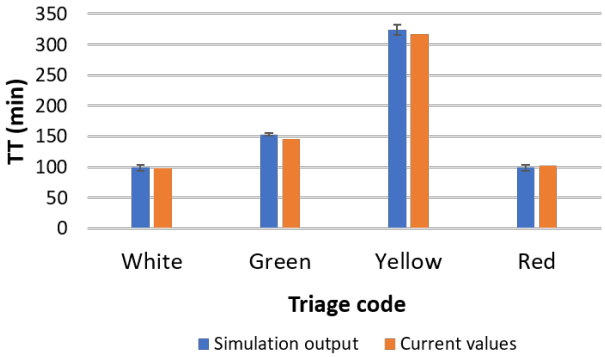


Figure 6.5. Plot of current values (in orange) and simulation output (in blue) of TT (in minutes) with the confidence interval.

Table 6.6. Output values (with confidence interval) for the outcomes returned by the simulation.

	White	Green	Yellow	Red
O1	122.08 ± 2.98	1023.04 ± 9.49	23.82 ± 1.36	
O2	39.06 ± 1.58	499.84 ± 6.61	21.56 ± 1.30	
O3		29.32 ± 1.51	184.00 ± 3.93	14.58 ± 1.12
O4			3.79 ± 0.42	3.66 ± 0.52
O5		18.72 ± 1.05	10.36 ± 0.89	
O6	3.06 ± 0.55	16.36 ± 1.16	1.88 ± 0.41	
O7	6.62 ± 0.82	23.44 ± 1.69	1.66 ± 0.35	
O8				1.92 ± 0.39

shows a good accuracy in representing such average values. Indeed, the model output values corresponding to each outcome and to each color tag are an accurate approximation of the current values, taking into account the confidence interval.

The previous paragraphs evidence that the simulation model provides an accurate representation of the actual system. As a last step needed to validate the model, the results obtained have been shown to ED personnel, acquiring important feedback and assessments.

6.4 Design of experiments and results

In this section, experimental results obtained by the DES model are reported. The aim is to determine performance measures of the ED, considering different scenarios, in order to evaluate the impact on the ED of patient peak arrivals due to a critical event. To this end, hypothetical scenarios where the patient arrival rate is artificially changed are considered. In particular, an increase of a prefixed percentage of the arrival rate due to the growth in demand is preliminarily considered. Then, a mildly loaded situation is analyzed, namely a gradual increase of patient arrivals over a period of few days of the week. Finally, the attention is turned to the main focus of this work, namely extremely loaded situations, possibly due to some critical conditions, for instance a natural disaster.

It is important to highlight that in the experimentation, the standard assumption that the service rate of ED personnel is unmodified during the patient peak arrivals is adopted. However, recent studies has shown that ED physicians and nurses could increase their service rate or adopt some particular strategies in case of increased workload. Some examples are multitasking techniques [68], "early task initiation" (where an upstream stage initiates tasks that are usually handled by a downstream stage, see, e.g., [31]), and other techniques consisting in adaptive response mechanisms to cope with critical situations emerging when treatments' requests exceed the normal capacity. Since the service rates are parameters of the simulation model, they can be easily modified. However, an analysis involving the service rate increase or the adoption of other adaptive techniques is not considered, since no information about these possible changes is available.

In the experimentation, the goal is to assess how the ED response changes in each scenario. In particular, the KPIs of interest are the metrics defined in Section 6.3, i.e., the waiting time WT and the total time TT. Moreover, the resources utilization is monitored focusing on the usage of *Area B* and *Area C*. In the literature on ED management under disaster conditions (see, e.g., Gul et al.[103]), utilization of ED resources and, in particular, utilization of treatment areas and medical staff are considered among the main indicators for evaluating the impact of a hypothetical critical event and for determining the appropriate resource and staff levels to cope with a disaster scenario. Note that, for an ED under normal conditions and for any general queuing system, the use of such an indicator could be questionable, since there are two conflicting viewpoints: on the one hand, the service manager aims to maximize the resources utilization (for economical reasons); on the other hand, service customers are penalized by such a behaviour that may cause deterioration of the service quality and possibly long queues. More specifically, in the experimentation, the resource usage is monitored as $(ScheduledTime - IdleTime)/ScheduledTime$, which is computed at each one-hour interval, i.e. the *resource usage on hourly basis*. It is preferable to consider this "continuous" resource utilization measure instead of the average usage, since high utilization can cause saturation and performance deterioration, even though usage is low when averaged over a long interval. Data

concerning the instantaneous utilization of the resources is collected by using VBA (Visual Basic for Applications) ARENA modules.

6.4.1 Increase of a prefixed percentage of the arrival rate

An increase of 10% in the hourly arrival rate with respect to the current value is considered. Overall, 50 independent replications are used. The length of each replication is 35 days with a warm up period of 7 days. In Figures 6.6-6.7, the comparison is reported in terms of WT and TT, respectively. Of course, the uniform

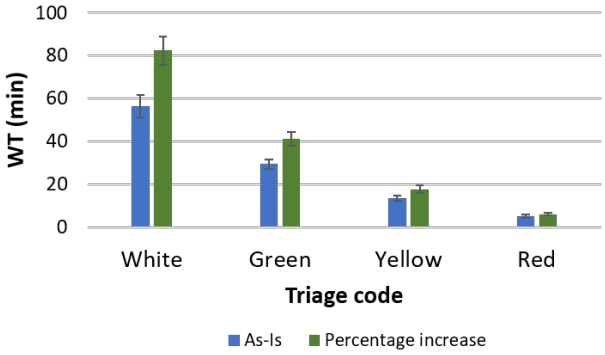


Figure 6.6. WT (in minutes): plot of the comparison between the current "as–is" status (in blue) and a prefixed percentage increase of the arrival rate (in green).

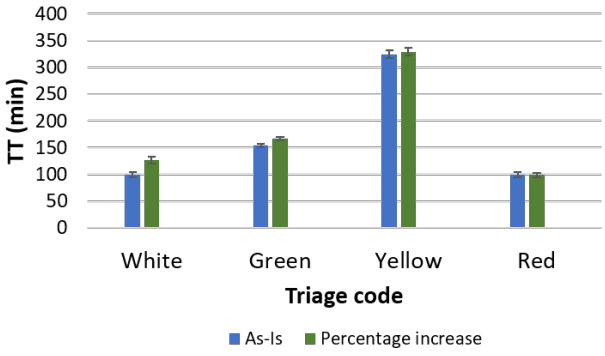


Figure 6.7. TT (in minutes): plot of the comparison between the current "as–is" status (in blue) and a prefixed percentage increase of the arrival rate (in green).

increase in the demand implies longer waiting/stay times. However, this growth in the arrival rate does not significantly affect waiting/stay times for yellow and red tagged patients. This is mainly due to the priority criterion and also to the small number of red and yellow tagged patients with respect to the green and white ones. In Figure 6.8-6.9, the comparison between the usage of *Area B* and *Area C* is reported over the week. For the sake of clarity of the plots, in the figures the usage is based on three-hours time slots. In this scenario analysis, the usage of *Area A* is not considered because of the reduced number of red tagged patients. From Figure 6.8 it can be observed that the usage of *Area B* in the rush hours is close to one even in the "as–is" status and that, as expected, the percentage increase

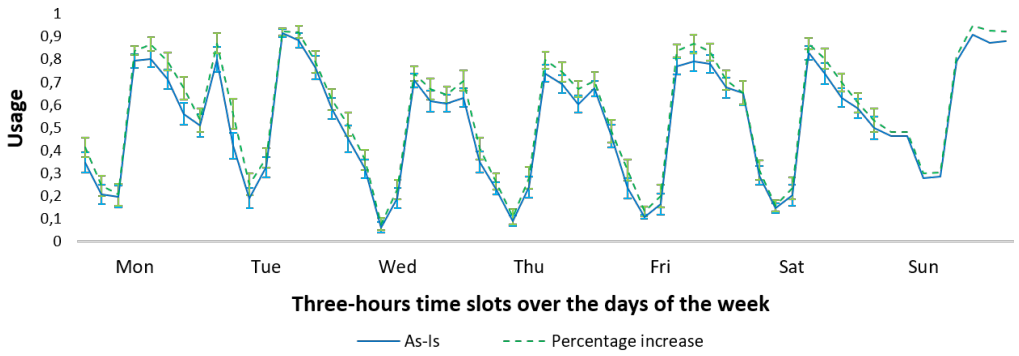


Figure 6.8. Plot of the *usage of Area B*.

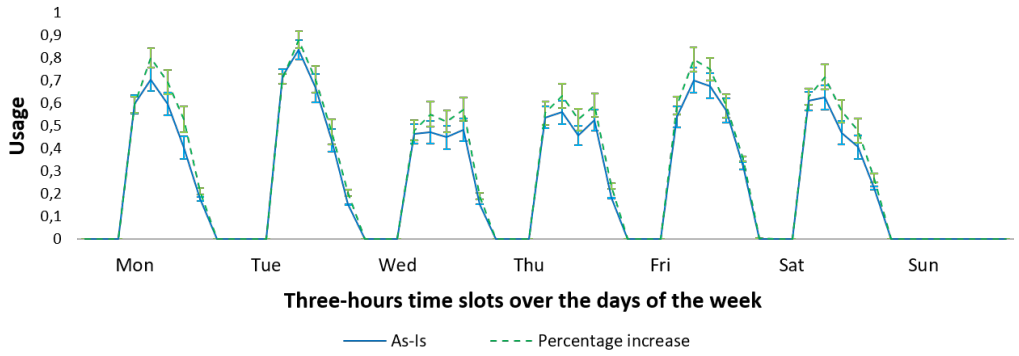


Figure 6.9. Plot of the *usage of Area C*.

causes a corresponding uniform growing of the utilization. Before discussing the usage of *Area C*, note that during the night only *Area B* is used, specifically for the treatment of both green and yellow tagged patients, since only one physician is on duty. Therefore, as shown in Figure 6.9, no patient is visited in *Area C* during the night. Conversely, this area is strongly used throughout the day, when two physicians are on duty, giving rise to a uniform increase in the usage, according to the growth in the average hourly arrival rate.

6.4.2 A mildly and an extremely loaded scenario

Now unexpected conditions due to a spike in the patient arrival rate related to a sudden and critical event (for instance, in the extreme case, an earthquake) are considered. The focus is on two possible artificial scenarios, namely both a mildly and an extremely loaded scenario corresponding to two different unpredictable occurrences. It is important to note that in this case a *terminating simulation* must be used since the interest is in monitoring the actual effect of arrival spikes on the ED as an unsteady system. The aim is to avoid that too many occurrences concerning standard days (without spikes) are included in the statistical analysis. This occurs if the KPIs are computed by averaging over a long run, e.g. 35 days. Therefore, 2 weeks are selected as replication length and a warm up period of 7 days to avoid bias due to initial conditions (empty system). In this manner the statistical analysis is focused on the week containing the peak arrivals and concentrates on the related transient state. For each experiment, 50 independent replications are used.

As regards the mildly loaded situation, we adopt a gradual increase/decrease in the arrival rate over the first three days of the week. More precisely, similarly to Ahalt et al. [6], the arrival rate is increased from 5% to 25%, depending on time slots, according to the scheme in Table 6.7. As concerns the extremely loaded

Table 6.7. Percentage increases in the arrival rate for *Monday*, *Tuesday* and *Wednesday* on different time slots

	Mon	Tue	Wed
00:00 – 08:00	+5%	+20%	+20%
08:00 – 14:00	+10%	+25%	+15%
14:00 – 20:00	+15%	+25%	+10%
20:00 – 24:00	+20%	+20%	+5%

scenario, a major emergency is reproduced. To this end, a 300% increase in the arrival rate centered over the 24 hours of Monday is considered. Figure 6.10 reports the increased hourly arrival rate for both scenarios along with the unmodified arrival rate. Figures 6.11–6.12 report the comparison between the current "as–is" status and the two scenarios in terms of WT. Similarly, in Figure 6.13, the same comparison is reported in terms of TT. As expected, in the extremely loaded scenario a huge increase is observed for both WT and TT of low-complexity patients (white and green tagged): the WT exceeds one day for white tagged patients and 10 hours for the green tagged ones, thus being not acceptable. As regards the mildly loaded scenario, a moderate increase is highlighted, showing that both WT and TT are actually still feasible. A different outcome is pointed out for higher complexity patients. As regards the red tagged ones, their current percentage with respect to the other color tagged patient is unchanged (scenarios with changes in this percentage are reported afterwards). In this case, even a huge increase in the overall number of arrivals does not lead to exceed one red tagged patient arrival per hour. Therefore, both WT and TT do not grow significantly, also due to the high priority assigned to these patients.

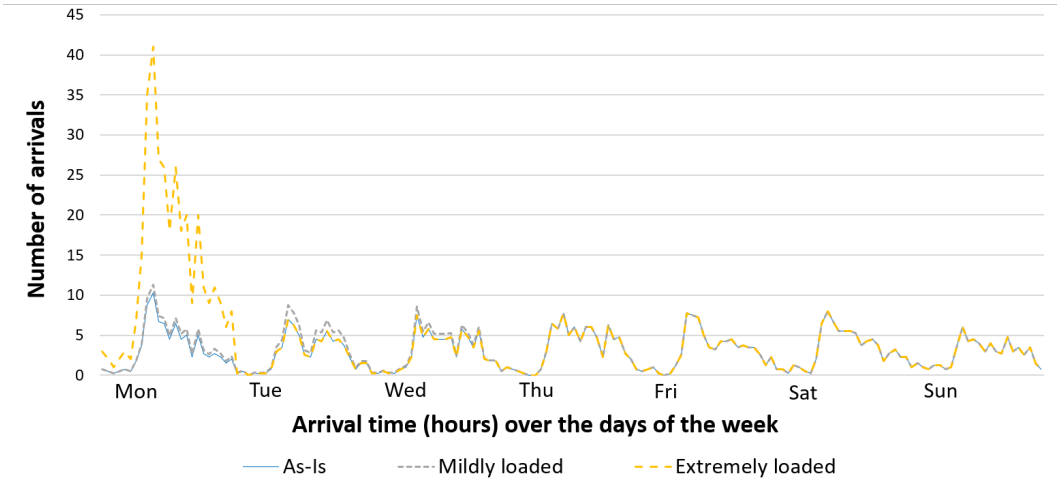


Figure 6.10. Plot of the increased patients arrival rate (mildly loaded in grey, extremely loaded in yellow) and the unmodified arrival rate (in blue).

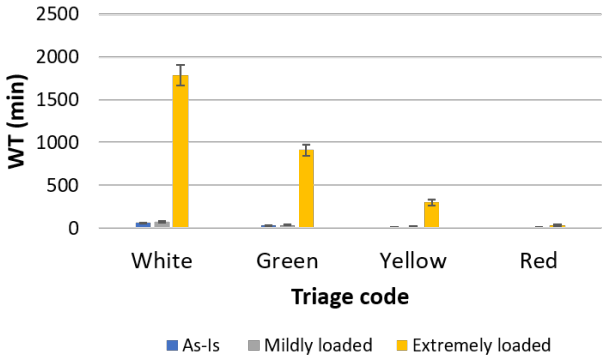


Figure 6.11. WT (in minutes): plot of the comparison between the current "as–is" status (in blue) and the mildly (in grey) and extremely (in yellow) loaded scenarios.

A similar result is observed for the yellow tagged patients. Moreover, note that the WT for red tagged patients is still approximately zero, in accordance with their high urgency level.

Figures 6.14-6.15 report the comparison between the current "as–is" status and the two artificial scenarios in terms of *Area B* and *Area C* usage. For the sake of clarity of the plots, in the figures the usage is based on three-hours time slots. From both these figures, it can be observed how, in the two scenarios, the peak in the patient arrivals causes a sudden increase in the usage of both resources (*Area B* and *Area C*). In case of extremely loaded scenario, the peak causes resource saturation even immediately before and after the peak center. Note how this phenomenon can be observed only by monitoring the instantaneous resource usage rather than the average utilization.

Now the extremely loaded scenario is analyzed more in detail. Indeed, due to the unpredictability of the phenomenon of peak arrivals caused by critical events, it is difficult to create artificial scenarios that actually reproduce what might happen in the real system. Therefore, analyze some variants of the 300% increase in the arrival rate are now analyzed. In particular, the comparison of the current "as–is"

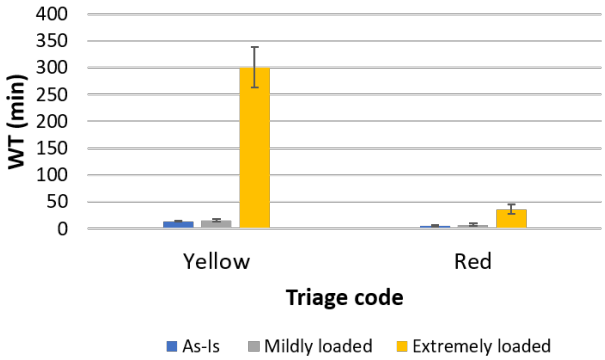


Figure 6.12. WT (in minutes): Detail for yellow and red tagged patients of the comparison reported in Figure6.11.

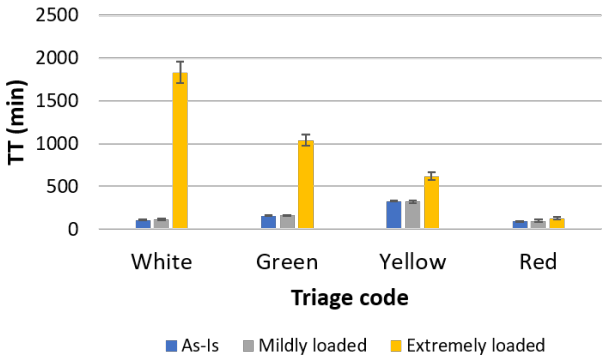


Figure 6.13. TT (in minutes): plot of the comparison between the current "as–is" status (in blue) and the mildly (in grey) and extremely (in yellow) loaded scenarios.

status is reported with respect to increases of 100%, 200%, and 400% in the arrival rate (always centered over the 24 hours of Monday and with the same percentage distribution of the color tags). In Figures 6.16–6.18, the corresponding WT and the TT are reported, along with those obtained for the 300% increase scenario. From Figure 6.17, it can be observed how WT for the red tagged patients still remains acceptable for all the four scenarios, and this is due to the low percentage of red tagged patients arrivals. As regards the yellow tagged ones, WT remains below 1 hour only for the 100% increase. As concerns the white and green tagged patients, WT becomes unacceptable even with the 100% increase. The TT reported in Figure 6.18 are direct consequence of the corresponding WT.

The same comparison between the current "as–is" status and the increases of 100%, 200%, and 400% in the arrival rate is reported in Figures 6.19–6.20 in terms of resource usage for *Area B* and *Area C*, respectively. Also in this case, the plots are based on three-hours time slots. These figures clearly highlight how, as expected, the percentage increase in the patient arrival rate strongly affects the utilization of both visit areas. Note that the usage of *Area B* also depends on the yellow tagged patients that are visited in this area during the night, when *Area C* is not operating. Both resources reach saturation points even in the most cautious scenario (100% increase).

All the scenarios up to now analyzed are based on increases in the patient arrival

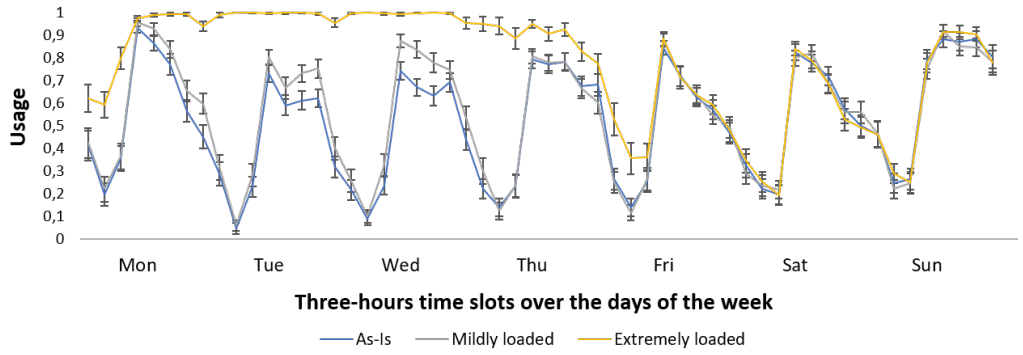


Figure 6.14. Plot of the *usage of Area B*: the current "as–is" status (in blue), the mildly loaded (in grey) and the extremely loaded (in yellow) scenarios.

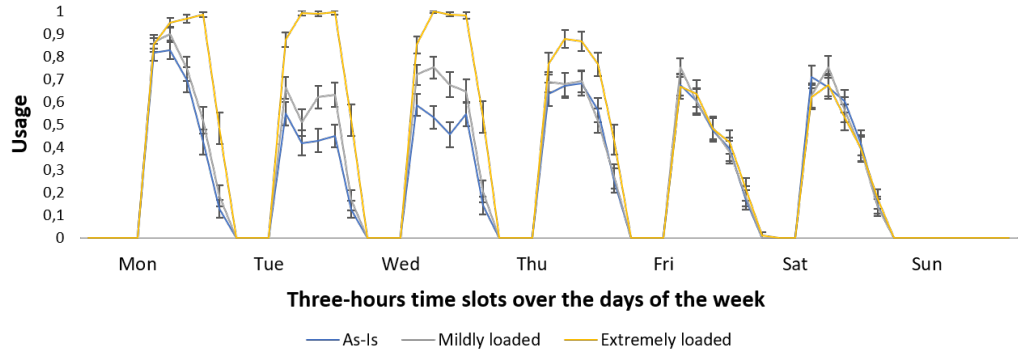


Figure 6.15. Plot of the *usage of Area C*: the current "as–is" status (in blue), the mildly loaded (in grey) and the extremely loaded (in yellow) scenarios.

rate, keeping unchanged the percentage distribution of different colors tagged patients. Actually, during a critical event, it is also likely to assume that the percentage of high priority patients grows during the arrival rate peak. During the latest earthquake (August 24, 2016) that hit Central Italian regions (where the ED considered in this work is located), in the hour corresponding to the peak in the arrivals, up to 14 patients with trauma due to crushing (i.e., red tagged patients) requested assistance. As already mentioned, in these cases, i.e., in case of the so called "maxi–emergency", Italian EDs adopt the "Internal Emergency Plan for Massive Inflow of Injured" (the Italian acronym PEIMAF is used), according to the current regulation. This implies the availability of additional resources and the adoption of different operating rules aimed at providing an adequate and timely assistance to all the patients who require it. Of course, although such a plan cannot be tested during the normal ED activity, an accurate assessment of its effectiveness must be performed in advance in view of its potential activation. A natural way for performing such a testing is to adopt a DES model. Therefore, the DES model is used also to provide the decision makers with useful insights concerning the design of the PEIMAF.

In the sequel, some analyses aiming to reproduce a critical situation corresponding to the extremely loaded scenario are reported. Additionally, an increase in the red tagged patient arrivals is considered. In particular, both an increase of 400% in the arrival rate and an increase of the percentage of red tagged patient arrivals is considered, in order to obtain about 14 of these patients arriving during the peak hour, as really occurred during the recent earthquake. Moreover, a possible maxi–emergency PEIMAF plan is assumed to be adopted and its effectiveness is assessed through the comparison of the KPIs obtained. More specifically, the following assumptions are adopted:

A1 *Patient arrivals:*

- the percentage of color tags assigned at triage is modified during the whole peak day, by assuming that 35% of the arrivals are red tagged patients.

A2 *Maxi–emergency plan (PEIMAF):*

- the number of physicians and nurses on duty is doubled during the peak day, starting from the peak hour (10:00 a.m.), then the shifts return to the normal scheme;

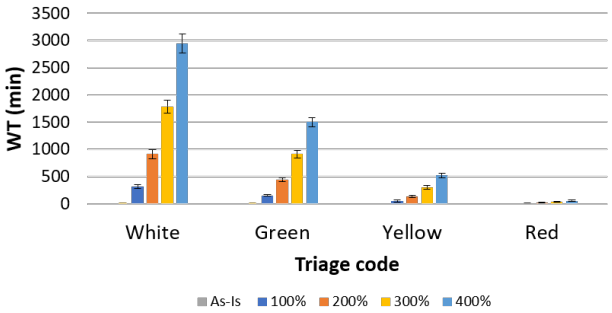


Figure 6.16. WT (in minutes): plot of the comparison between the current "as–is" status (in grey) and the extremely loaded scenario with increases in the arrival rate of 100% (in blue), 200% (in orange), 300% (in yellow), and 400% (in light blue).

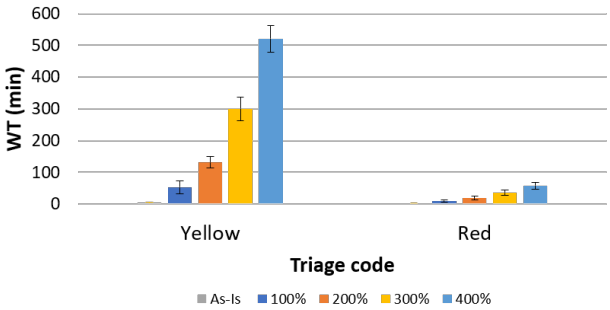


Figure 6.17. WT (in minutes): Detail for yellow and red tagged patients of the comparison reported in Figure6.16.

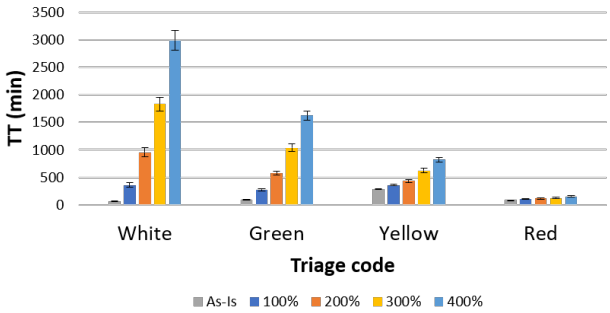


Figure 6.18. TT (in minutes): plot of the comparison between the current "as–is" status (in grey) and the extremely loaded scenario with increases in the arrival rate of 100% (in blue), 200% (in orange), 300% (in yellow), and 400% (in light blue)

- green and yellow tagged patients are not admitted to the ED, starting from the peak hour (10:00 a.m.), but they are sent to outpatients facilities;
- also Area B and Area C can be used for the treatment of red tagged patients.

In this manner, by **A1** about 14 red tagged patient arrivals are observed during the peak hour, thus reproducing a really occurred critical situation. By **A2** a possible simple configuration of a maxi–emergency plan is implemented, only for experimental purposes. Actually, the operational procedures provided by a PEIMAF plan are much more complex and articulate than the ones reported in this paper: here an illustrative example is considered only to show how the DES model can be fruitfully used to test a maxi–emergency plan.

Figures 6.21–6.22 report the WT and the TT for the "as–is" status and the extremely loaded scenario with 400% increase in the patient arrival rate and the percentage of red tags assigned at triage modified according to **A1**. The comparison concerns the values of these KPIs obtained without adopting a maxi–emergency plan and by using the plan described in **A2**. The figures clearly evidence the huge times resulting without adopting the maxi–emergency plan. Even by adopting the plan as specified above in **A2**, WT exceeds 3 hours for yellow-tagged patients and 1 hour for the red-tagged ones. This situation is confirmed by the plot of the usage of the visit areas, reported in Figures 6.23–6.25. First, note that when the emergency plan is not activated, since only two physicians are on duty during the peak day,

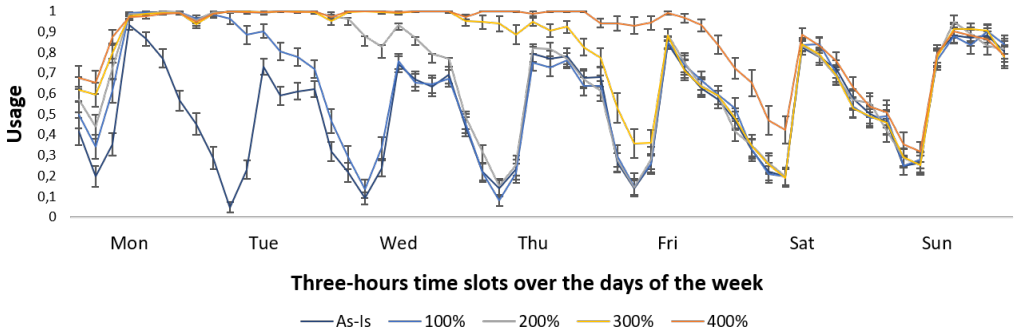


Figure 6.19. Plot of the *usage of Area B*: the current "as–is" status and the extremely loaded scenarios.

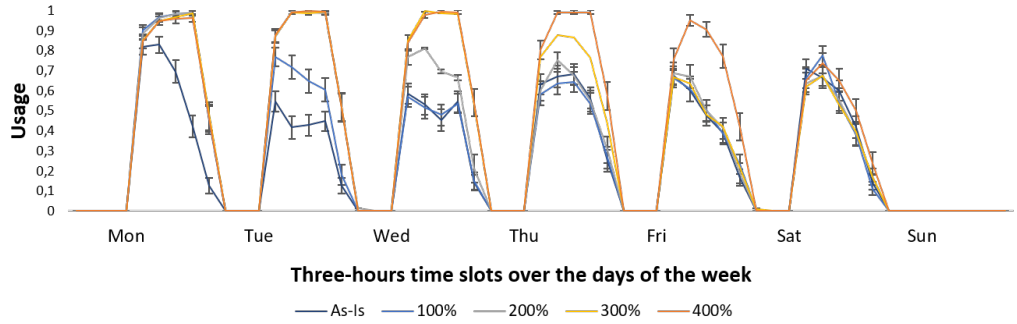


Figure 6.20. Plot of the *usage of Area C*: the current "as–is" status and the extremely loaded scenarios.

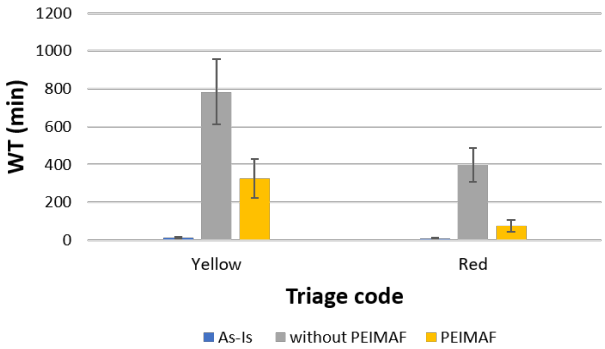


Figure 6.21. WT (in minutes): plot of the comparison between the current "as–is" status (in blue) and the extremely loaded scenario with the modified percentage of red tags assigned at triage as in **A1**, with (in yellow) and without (in grey) adopting a maxi–emegency plan as in **A2**.

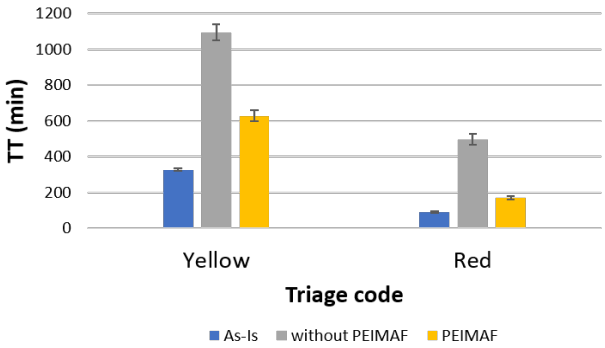


Figure 6.22. TT (in minutes): plot of the comparison between the current "as–is" status (in blue) and the extremely loaded scenario with the modified percentage of red tags assigned at triage as in **A1**, with (in yellow) and without (in grey) adopting a maxi–emergency plan as in **A2**.

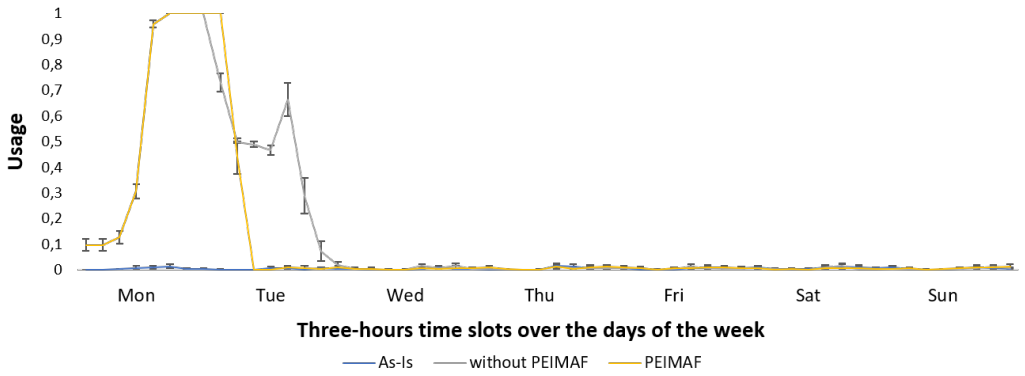


Figure 6.23. Plot of the *usage of Area A*: the current "as–is" status and the extremely loaded scenarios with and without the maxi–emergency plan.

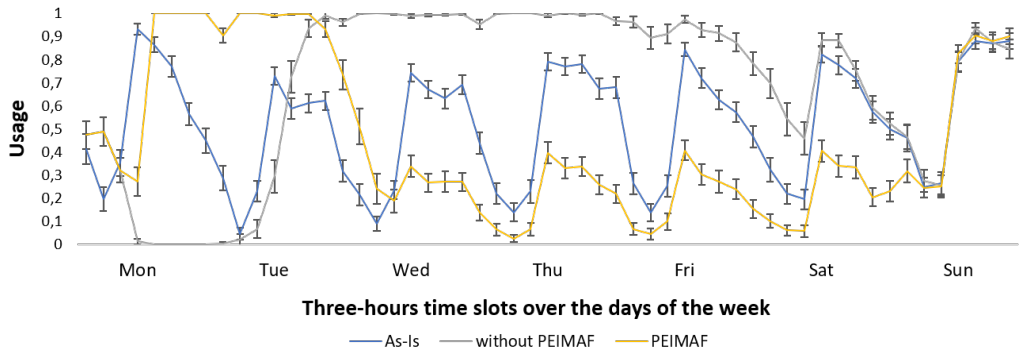


Figure 6.24. Plot of the *usage of Area B*: the current "as–is" status and the extremely loaded scenario with and without the maxi–emergency plan.

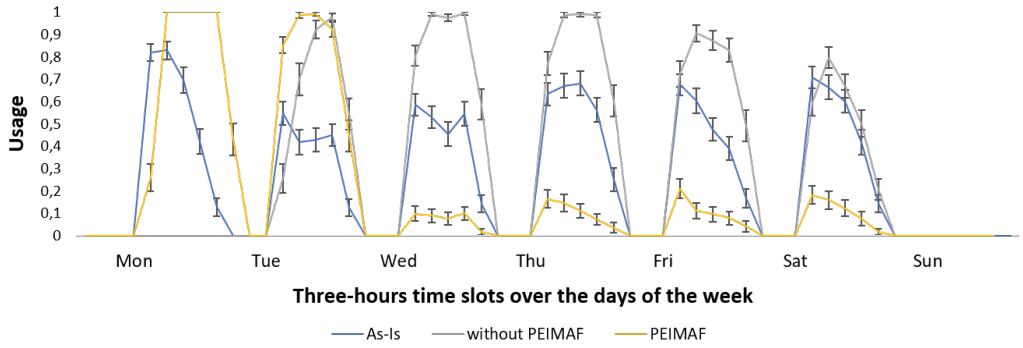


Figure 6.25. Plot of the *usage of Area C*: the current "as–is" status and the extremely loaded scenario with and without the maxi–emergency plan.

they both remain assigned to *Area A* throughout the day, due to the large number of red patients to be visited in this area. Since there are no physicians assigned to *Area B* and *Area C*, the usage level of these areas is zero until the distribution percentage of the red patients returns to the normal scheme. On the contrary, when the emergency plan is activated, since four physicians are on duty during the peak day, even *Area B* and *Area C* can be used for the treatment of higher complexity patients. As a consequence, all the visit areas are used. After the peak arrivals end, the utilization of *Area B* and *Area C* decreases since the extra human resources allow more patients to be visited at the same time, whereas the utilization of *Area A* drops to zero. In any case, even if the adoption of the maxi–emergency plan as assumed in **A2** leads to an improvement, its overall inadequacy is very evident. Indeed, the resource saturation is reached for long periods, implying that the resources cannot satisfy extra requests. To cope with this situation, an enlargement of the resources, whether human or physical, would be desirable even if not sufficient in most cases. Emergency plans usually provide for an increase of ED personnel, while an extension of physical rooms (even if temporary) is usually more difficult. Of course, patient diversion policy towards neighboring EDs should be adopted in case of an excessive ED congestion. Thanks to the high flexibility of the ARENA implementation of the DES model concerning the ED under study, an extensive scenario analysis can be performed aimed at assessing the ED operational capacity, namely all the KPIs of interest in many and different real critical situations.

Chapter 7

Case study: emergency department of Policlinico Umberto I

The second case study concerns the ED of *Policlinico Umberto I*, which is a very large hospital in Rome, Italy. It is the biggest ED in the region of Lazio in terms of number of patient arrivals per year (about 140,000 on average). By using the data collected from the patient flow through the ED for the whole year 2018, this case study is adopted to test the effectiveness of the two approaches proposed in Chapter 5. In particular, an extensive experimentation shows that the approach developed for the ED arrival process enables finding the number of intervals, along with their length, such that an accurate approximation of the empirical arrival rate is achieved, ensuring the consistency between the NHPP hypothesis and the arrival data. Moreover, results from a significant sensitivity analysis demonstrates that the regularity of the optimal piecewise constant approximation can be finely tuned by properly weighing the penalty term of the objective function with respect to the fitting error term. Similarly, the numerical experiment reported for the ED model calibration approach shows promising results and a significant improvement is expected to be achieved through further research. By using the accurate simulation model resulting from the application of these two approaches, SBO is used for solving a resource allocation problem related to the specific case study considered. The goal is to determine the optimal settings of the ED unit devoted to the medical visit of low-complexity patients in order to reduce the overcrowding level. A multiobjective formulation of the problem is adopted to find a trade-off between the conflicting goals of reducing the management cost and guaranteeing patients timely treatments according to their urgency code.

7.1 Description of the patient flow in the emergency department

The ED of Policlinico Umberto I is divided into several areas, each one associated with a medical specialty. The backbone of the ED is the *central area*, which is devoted to treating diseases and disorders related to internal medicine and general surgery, which affect the majority of patients. Separated from this main area, there are other parts of the ED that deal with the following medical specialties: ophthalmology, obstetrics, pediatrics, hematology, and dentistry. The focus of this chapter is on the

central area, whose number of arrivals in 2018 amounted to more than 50,000.

In order to gain a complete understanding of the ED processes, all the stages of the patient flow are reviewed. Since the first step is the arrival process, the first part of this section focuses on the arrivals data. In particular, some plots are reported to analyze the data collected from the 1st of January to the 31st of March. In Figure 7.1, the weekly average hourly arrival rate obtained by averaging the number of arrivals occurring in the same hourly time slot over the 13 weeks considered is reported. It is worth observing that, in accordance with the literature (see, i.e.,

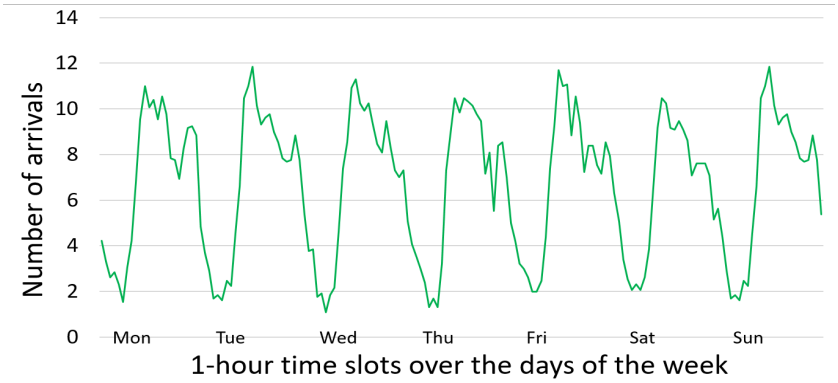


Figure 7.1. Plot of the weekly average hourly arrival rate for the first 13 weeks of the year.

[139]), the average arrival rates among the days of the week are significantly different. Therefore, since averaging over these days would lead to inaccurate results, the different days of the week must be considered separately. In Figure 7.2, the plot of the average hourly arrival rate for the Tuesdays over the 13 weeks is reported, while Figure 7.3 shows mean and variance of the interarrival times occurred on the first Tuesday of 2018. From this latter figure, we observe that these two statistics

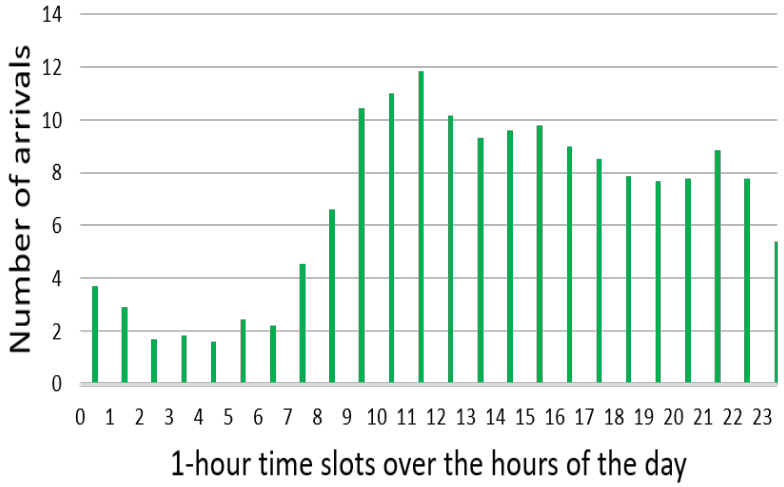


Figure 7.2. Plot of the average hourly arrival rate for the Tuesdays over the 13 weeks considered.

have similar values within each 3-hours time slot and this is in accordance with the property of the Poisson probability distribution for which mean and variance

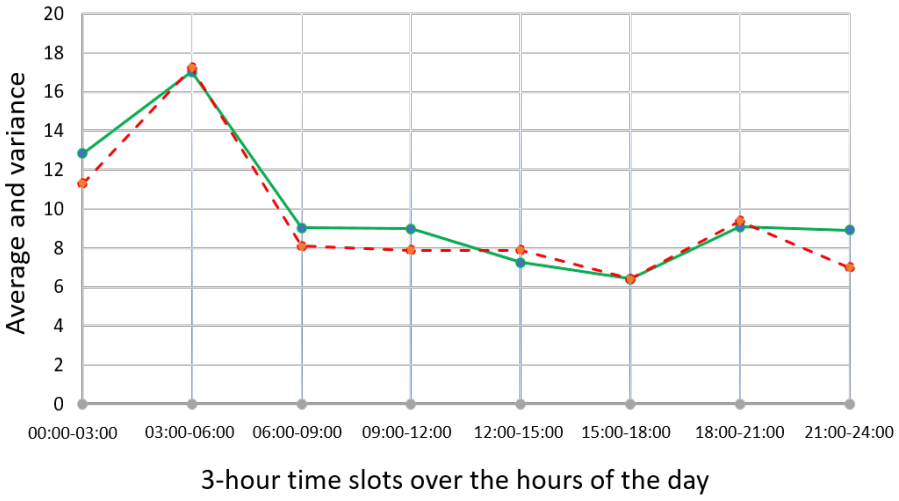


Figure 7.3. Plot of the average (in solid green) and variance (in dashed red) of the interarrival times for the first Tuesday of 2018. On the abscissa axis, 3-hour time slots are considered.

coincide. This evidence justifies the approach proposed in Section 5.1, which is based on a nonhomogeneous Poisson arrival process and will be applied to the case study in Section 7.3.

It is important to remark that seasonal phenomena might affect the number of weeks to be considered from the dataset for model estimation due to the potentially large variability over successive weeks. Indeed, overdispersion phenomenon may require a model calibration for each particular period of the year in order to take into account typical situations which occur, for instance, during the flu season. This important aspect clearly emerges also from the experimentation reported in Section 7.3.

After the analysis of the arrivals data, the description of the ED units and staff and the summary of the patient flow are reported in the sequel. Other than a *triage area*, where each incoming patient is assigned a color tag by a nurse in charge of this task, the ED is composed by

- a *Medical Unit (MU)*, devoted to patients who need specialized medications and treatments, with areas dedicated to the most critical patients;
- a *Surgical Unit (SU)*, devoted to patients who need either to receive a surgical operation or to recover from it, with areas dedicated to the most critical patients;
- a *Resuscitation Area (RA)*, for the most acutely ill and injured patients, who need timely treatments;
- a *Minor Injuries Unit (MIU)*, for the least urgent patients, whose treatment can be delayed or deferred;
- an *Orthopedic Unit (OU)*, for patients suffering from orthopedic disorders.

Moreover, all of these units have rooms where patients can either wait for exams or stay for observation. Red-tagged patients can be visited and treated in RA or in

dedicated areas within MU and SU, which are open 24 hours a day and are provided with equipment and staff specialized for dealing with life-threatening illnesses and injuries. In particular, 1 and 2 seats are available in the dedicated areas of MU and SU, respectively, and further 2 seats are available in RA. As concerns the medical treatment of the other patients, MU and SU can host up to three and two patients during the day (8.00 a.m.–8.00 p.m.), respectively. At night, MU can host two patients, while in SU one seat is available. Moreover, MIU has two seats, which are available from 8.00 a.m. to 8.00 p.m., Monday through Saturday. When patients experience excessive waiting times, two additional seats may be used to visit up to four patients simultaneously. All this information is summarized in Tables 7.1–7.2.

Table 7.1. Number of seats available for medical visit and treatment in MU, SU, and MIU, the latter being open from Monday to Saturday.

	MU	SU	MIU
Day (8.00 a.m.–8.00 p.m.)	3	2	2
Night (8.00 p.m.–8.00 a.m.)	2	1	0

Table 7.2. Number of seats available for medical visit and treatment of red-tagged patients in RA and in the dedicated areas of MU and SU.

	MU	SU	RA
Day (8.00 a.m.–8.00 p.m.)	1	2	2
Night (8.00 p.m.–8.00 a.m.)	1	2	2

Table 7.3. Feasible assignments of patients to the ED units according to the color tag. A cross at the entry (i, j) indicates that a patient with color tag i can be assigned to the unit j.

	MU	SU	RA	MIU	OU
White	-	-	-	X	X
Green	X	X	-	X	X
Yellow	X	X	-	-	X
Red	X	X	X	-	-

As regards the patient flow, after arriving autonomously or by emergency medical vehicles, all the incoming patients are admitted to the triage area, where a nurse assigns the color tag. After the triage, the patients are visited and treated in one of the units previously described, according to the color tag assigned and the severity of the illness/injury. Table 7.3 represents a scheme showing the units where patients may be assigned based on the color tags. In case of red-tagged patients, the medical visit is timely performed in RA or in the dedicated areas of MU and SU. As concerns the other color tags, the yellow and green tagged patients share the same resources in MU and SU. However, while the former can be assigned only to MU and SU, the latter may be sent to MIU during its opening hours if their health conditions are deemed as not likely to worsen. If MIU is closed, all the green-tagged patients are visited and treated in MU and SU. This diversion allows the ED to reduce the occurrence of work overload in SU and MU, which may give rise to overcrowded units. Moreover, it is important to point out that the white triage tag is assigned only if MIU is open, otherwise the green tag is used.

In many cases, after the medical visit, additional exams may be required. Other than performing reassessments of the patients and requiring additional exams,

physicians may also require further observation periods. Finally, at the end of the pathway, patients are discharged from the ED. This last stage includes a final waiting time whose duration depends on the type of outcome. Indeed, a longer wait is expected for patients that need to either be hospitalized at a hospital ward or transferred to another hospital, while patients discharged home can usually leave the ED in shorter time.

Although the collected data concerns the whole year 2018, the focus of the remaining part of this section is on the data related to January, which is used for building the DES model of the ED. The choice of focusing on January stems from the will of the ED management to reduce the overcrowding level observed in the winter season, which exhibits longer waiting times, as emerged by interviewing the ED staff. Among the winter months, January is observed to suffer from the heaviest workload, which puts a strain on the ED processes, thus requiring a careful analysis. However, the simulation model could be also easily adapted to include input parameters estimated from data related to different months.

From 00:00 of January 1 to 23:59 of January 31, 2018, the total number of patients arrived to the ED is 4192. The timestamps recorded are the ones marked as known in Figure 5.1. In Table 7.4, the number and the percentage of color tags assigned at triage are reported along with the number of patients who leave without being seen (LWBS). Although, in some cases, re-evaluations can lead to different patient tags at discharge with respect to those assigned at triage, in the dataset considered this information is unavailable. Since from Monday to Saturday MIU

Table 7.4. Number and percentage of color tags assigned at triage (columns 2-3) and number of patients LWBS (column 4).

	TRIAGE TAGS		LWBS
White	65	1.55%	1
Green	1804	43.17%	16
Yellow	2058	49.25%	-
Red	252	6.03%	-
	4179	100%	

is open from 8.00 a.m. to 8.00 p.m., it is worth reporting the comparison of the proportions of color tags between the daytime and the night, as shown in Table 7.5, so that the resulting change in the ED setting is considered. Finally, Tables 7.6–7.9

Table 7.5. Number and percentage of color tags assigned at triage in the daytime (columns 2-3) and at night (column 4-5).

	TRIAGE TAGS			
	Day (8.00 a.m.–8.00 p.m.)		Night (8.00 p.m.–8.00 a.m.)	
White	65	2.17%	-	-
Green	1306	44.32%	498	40.42 %
Yellow	1420	48.18%	638	51.79 %
Red	157	5.33%	95	7.71 %
	2948	100%	1231	100%

show the number and the proportion of the color tags among the units. Although OU is out of the scope of this analysis, these tables include also this unit since OU patients share the triage station with the other patients, thus affecting the counts reported in Tables 7.4– 7.5.

Table 7.6. Number of patients assigned to the ED units for each color tag.

	MU	SU	RA	MIU	OU
White	-	-	-	47	17
Green	248	628	-	260	668
Yellow	1316	693	-	-	49
Red	191	45	16	-	-
	1755	1366	16	307	734

Table 7.7. Proportion of patients assigned to the ED units for each color tag.

	MU	SU	RA	MIU	OU	
White	-	-	-	73.44 %	26.56 %	100%
Green	13.75 %	34.81%	-	14.41 %	37.03%	100%
Yellow	63.95 %	33.67%	-	-	2.38 %	100%
Red	75.79 %	17.86%	6.35 %	-	-	100%

Table 7.8. Number of green-tagged patients assigned to MU, SU, and MIU in the daytime (8.00 a.m. – 8.00 p.m.) and at night (8.00 p.m. – 8.00 a.m.).

	MU	SU	MIU
Daytime	132	403	260
Night	116	225	-
	248	628	260

Table 7.9. Proportion of green-tagged patients assigned to MU, SU, and MIU in the daytime (8.00 a.m. – 8.00 p.m.) and at night (8.00 p.m. – 8.00 a.m.).

	MU	SU	MIU	
Daytime	16.60 %	50.69%	32.70 %	100%
Night	34.02 %	65.98%	-	100%

7.2 The discrete event simulation model

This section describes the DES model of the ED of Policlinico Umberto I, which has been implemented by using Simmer [223], a process-oriented and trajectory-based DES package for R. In the DES model, patients are represented by the model *entities*, which are created according to the statistical model used for modeling the arrival process. After arriving at the ED, the simulated patient flow followed by each entity is based on trajectories determined by the logical rules used to build the model. Each trajectory is associated with the pathway followed by a patient according both to the color tag received at triage and to the ED unit assigned. Figures 7.4–7.8 show the patient flow from the arrival to the discharge according to the color tag assigned at triage. In particular, Figure 7.4 focuses on the logic underlying the assignment of color tag at triage. The other figures deal with the segments of the patient flow following the triage, providing in brackets the information about the *resources* seized in case the corresponding process is not a simple delay. Such resources represent the seats at each ED unit, whose capacity can be either fixed, as is the case with RA and the areas of MU and SU dedicated to red patients, or based on schedule, as for MU, SU, and MIU.

As regards the arrival process, the stochastic process defined by the arrivals to the ED is assumed to be well modeled by a NHPP, which is a standard assumption in the literature (see, e.g.,[234, 153, 243, 7, 6, 105]). The validity of this hypothesis for the case study considered is supported by Figure 7.3, where mean and variance of the interarrival times have similar values on each 3-hour time slot, which is in accordance with the corresponding property of the Poisson distribution. Moreover, a more formal justification of the NHPP hypothesis is also provided by the extensive analysis performed in Section 7.3, where statistical hypothesis testing is adopted. To achieve an accurate representation of the arrival rate, 24 time slots of unitary length are considered for each day of the week, thus obtaining a piecewise constant approximation. While this approach allows taking into account the within-day variation, the day-to-day variation is considered by estimating the hourly arrival rate function separately for each day of the week.

The patient flow through the model can be described as follows. After being created at the beginning of the simulation, the entities corresponding to deceased patients leave the model, while the remaining entities enter the *triage area* for the assignment of both color tag and ED unit, which are stored as entity attributes. The discrete uniform probability distributions used for assigning the color tag and the ED unit vary according to the time at which the entity starts the triage. If this event happens in the daytime, the corresponding probability distributions include also the white color tag and MIU among the alternatives to sample. The triage phase is represented by as many delay processes as the number of color tags and units. Each of these delays is associated with a probability distribution that returns the value of the triage duration for each patient. In particular, the 8 different processes considered correspond to the possible pairs of color tags and units reported in Table 7.3, with the exception of the pair given by the green color tag and MIU. Indeed, the triage duration for a green-tagged patient eligible for MIU depends on the specialty required, whether medical or surgical. Since this information is unknown, in the DES model half of the MIU patients are subject to the triage duration of MU patients, while the other half are subject to the triage duration of SU patients. It is important to remark that, since the processes are simple delay, no queue is generated before the triage. The reason underlying this choice is twofold: on the one hand, this allows for alignment between data e simulation model, since the timestamps denoted as t_0 in Figure 5.1 are used to derive the arrival probability

Table 7.10. Probability distribution of the number of visits for each patient and of the final waiting time before discharge.

		NUMBER OF VISITS	FINAL WAITING TIME
WHITE	MIU	Geom(1)	Weib(1.12, 0.41)
GREEN	MIU	Geom(0.99)	Weib(0.83, 0.57)
	SU	Geom(0.85)	Weib(0.61, 1.07)
	MU	Geom(0.85)	Weib(0.57, 4.91)
YELLOW	SU	Geom(0.95)	Weib(0.63, 2.33)
	MU	Geom(0.90)	Weib(0.62, 7.22)
RED	RA	Geom(0.95)	Weib(0.67, 9.26)
	SU	Geom(0.95)	Weib(0.71, 4.47)
	MU	Geom(0.85)	Weib(0.69, 6.65)

distribution, which consequently returns the starting time of triage (and not the starting time of the waiting time for triage); on the other hand, the waiting time before triage can be considered negligible, according to the ED staff.

After the triage process, some entities leave the model, thus representing the patients who leave without being seen, while the other ones start waiting for the medical visit. The entity selected for the visit depends both on the priority class, i.e., the color tag, and on the FIFO criterion. Then, when the visit begins, one seat in the corresponding unit is seized and the duration of the visit is returned by a suitable probability distribution based on the entity color tag.

After the medical visit, the following phase includes *exams and reassessments*, whose duration is generated by means of a suitable probability distribution. Like the triage, a single delay process is used for this phase as well, due to the lack of knowledge of the resources required. It is important to point out that all possible further treatments are considered included in the service time of this process. After the exams, the geometric probability distributions reported in Table 7.10 return the number of times a patient repeats the visit and the exams. If these processes are no longer required, the entities corresponding to patients *refusing hospitalization*, *leaving during exams*, and *being transferred to another hospital* are removed from the model, while the other entities proceed to the next stage, i.e., the *final waiting time* before discharge. The probability distributions used to generate values for this final service time are reported in Table 7.10 as well.

Since in the dataset some timestamps defining the starting and ending time of triage, medical visit, and exams are not available, the service times corresponding to these activities cannot be recovered directly. Moreover, additional delay processes are considered to reproduce all the service times that cannot be directly computed through the data, such as the setup times that are sometimes needed for sanitizing the ED areas and the idle times caused by unexpected requests for personnel from other ED units, which give rise to sudden activity disruption. In order to effectively model these times, which impact on all the patients without predictable patterns, uniform probability distributions are considered according to the color tag. Due to the lack of specific data, a model calibration procedure is required to determine good estimates for the parameters of the probability distributions underlying these processes, as described in Section 7.4.

As regards the KPIs of interest, the focus is on the time differences *DOT* and *DIT* described in Figure 5.1, which can be computed through the data of the real system and then compared with the corresponding values returned by the simulation.

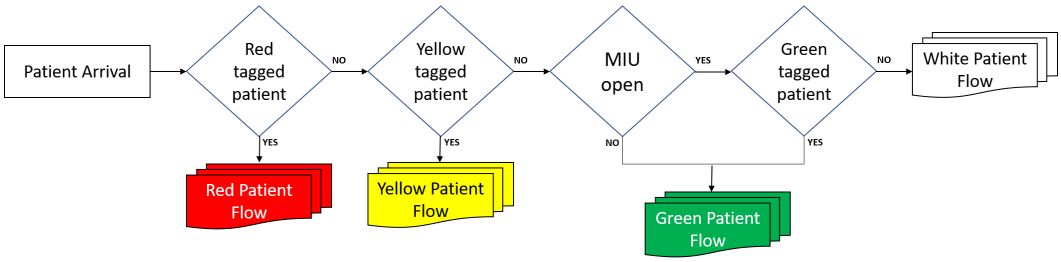


Figure 7.4. Plot of the patient flow in the ED from the arrival to the assignment of the color tag.

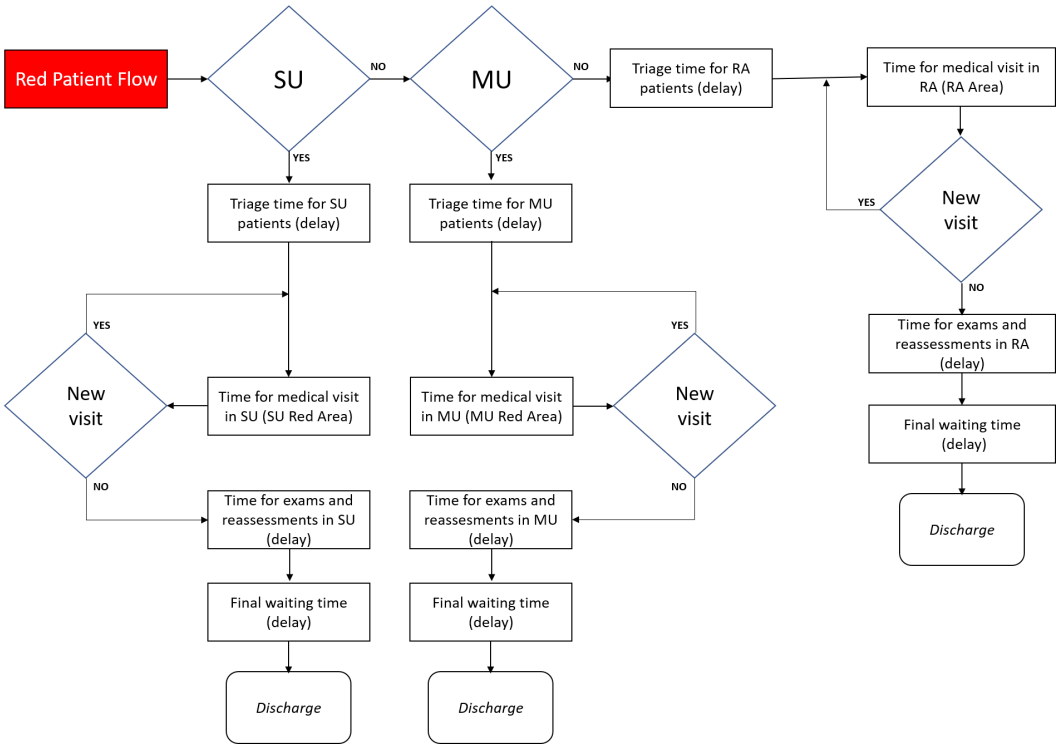


Figure 7.5. Plot of the red-tagged patient flow.

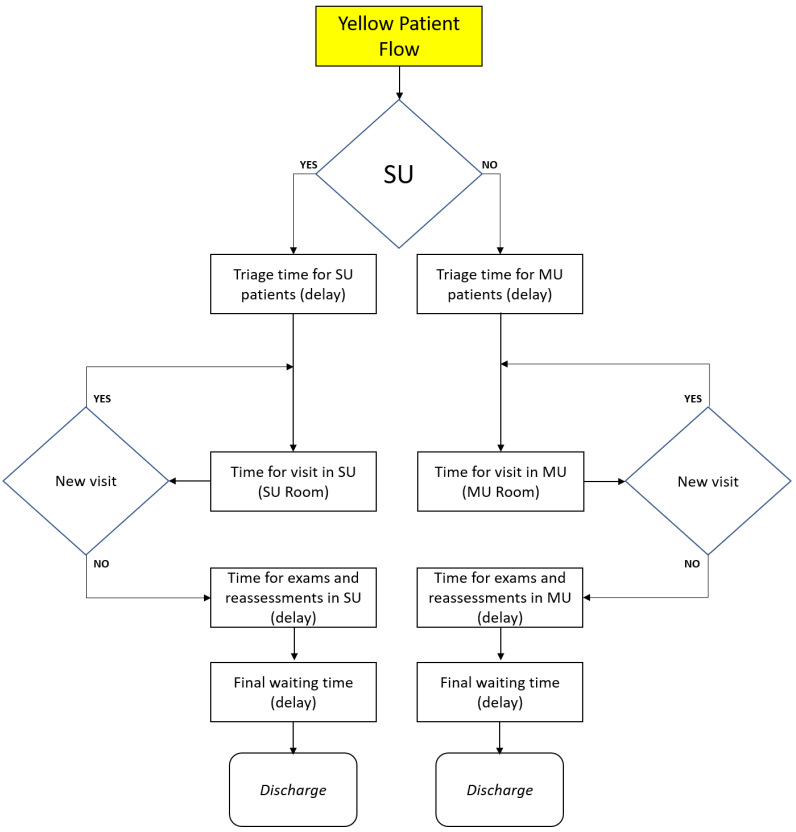


Figure 7.6. Plot of the yellow-tagged patient flow.

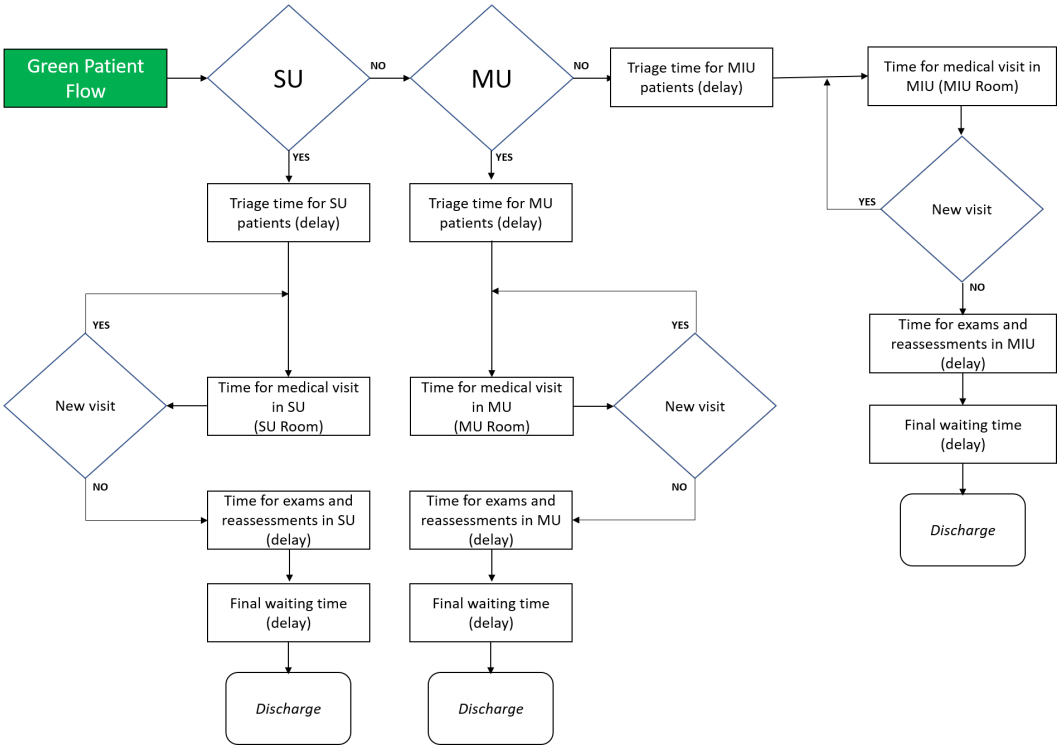


Figure 7.7. Plot of the green-tagged patient flow.

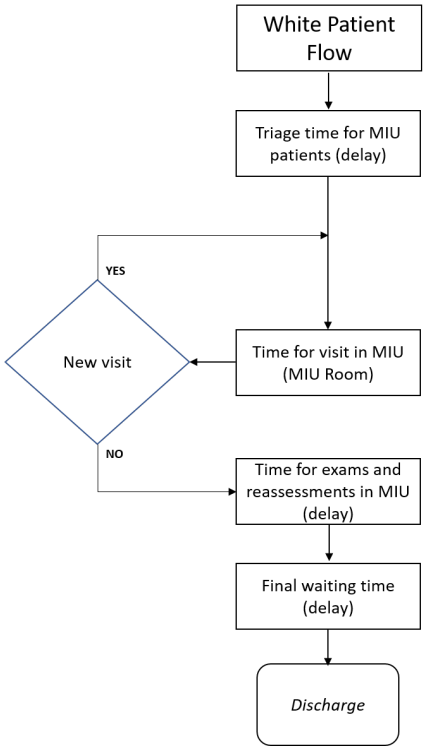


Figure 7.8. Plot of the white-tagged patient flow.

Therefore, the KPIs of interest are

- *DOT*, which is the time difference between the starting time of the triage and the starting time of the visit, namely $t_2 - t_0$ in Figure 5.1;
- *DIT*, which is the time difference between the starting time of the visit and the time of the discharge, namely $t_6 - t_2$ in Figure 5.1.

7.3 Nonhomogeneous arrival process

To show the effectiveness of the approach described in Section 5.1, let us focus on the patient arrivals data collected in the first m weeks of the year. Both walk-in patients and patients transported by emergency medical service vehicles are considered. Figure 7.9 reports the plot of the average rates λ_i^F estimated from the data of the first $m = 13$ weeks of 2018 over 96 equally spaced intervals of 15 minutes. From this figure it can be easily observed that, as expected, the arrival rate

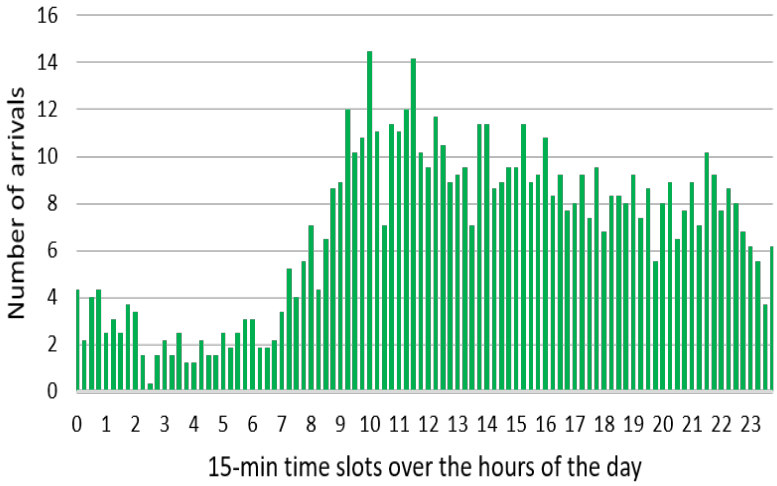


Figure 7.9. Plot of the empirical arrival rate model $\lambda_F(t)$ obtained by averaging the data over the first $m = 13$ weeks of 2018.

drastically changes from night hours to day hours, with a significant grow during the morning hours. In Figure 7.10 the average hourly arrival rate is reported for each day of the week. By observing this figure, being the shape of each rate similar, the approach proposed is expected to enable obtaining similar partitions of the 24 hours on different days of the week. This allows focusing only on one arbitrary day of the week. Specifically, any day of the week can be selected to apply the methodology under study and the same way would apply to the other days, thus obtaining a different partition for each day. As an example, Tuesday is the day chosen for the experimental results.

7.3.1 Experimental results

In this section, the results of an extensive experimentation is reported on the data concerning the case in hand, namely the ED patient arrivals collected in the

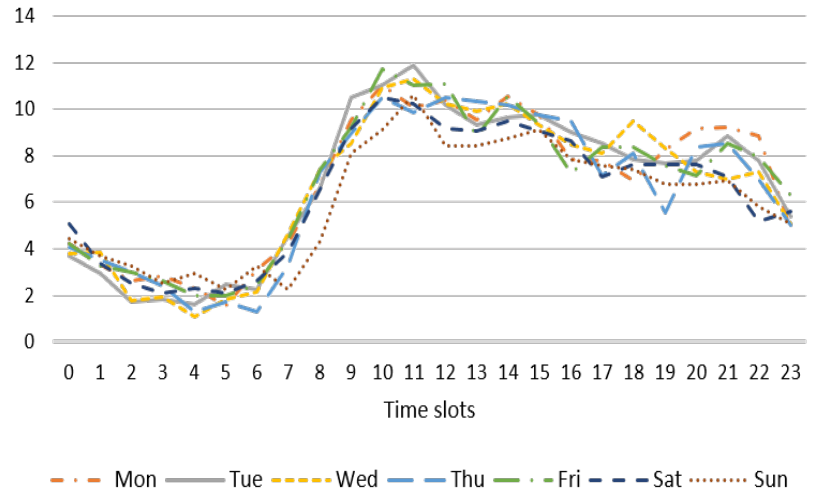


Figure 7.10. Plot of the comparison among the average hourly arrival rates for each day of the week obtained by averaging the data over the first $m = 13$ weeks of the year.

first m weeks of year 2018. Different values of the number of weeks m are considered. Standard significance level $\alpha = 0.05$ is used in the CU KS and dispersion tests.

As regards the integer nonlinear constrained black-box optimization problem described in Section 5.1.2, the value of ℓ in (5.1.9) is set to 1 hour. Moreover, it is important to note that different values of the weight w in the objective function (5.1.7) lead to various piecewise constant approximations with different fitting accuracy and degree of regularity. Therefore, a careful tuning of this parameter is performed, aiming to determine a value which represents a good trade-off between a small fit error and the smoothness of the approximation.

The algorithmic framework proposed in [171] is adopted to solve the problem. As already mentioned in Section 2.3, this algorithm represents a novel strategy for solving black-box problems with integer variables and it is based on the use of suited search directions and a nonmonotone linesearch procedure. Moreover, it can handle generally-constrained problems by using a penalty approach. It is worth highlighting that the results reported in [171] clearly show that this algorithm framework is particularly efficient in tackling black-box problems like the one in (5.1.12). As concerns the parameters of the optimization algorithm used in the experimentation, the default values are adopted (see [171]). The stopping criterion is based on the maximum number of function evaluations, which is set to 5000. The following point

$$x_i^0 = i - 1, \qquad i = 1, \ldots, 25, \tag{7.3.1}$$

which corresponds to the case of 24 intervals of unitary length, is adopted as a starting point x^0 of the optimization algorithm. This is the partition commonly used in most of the approaches proposed in literature (see, e.g., [6, 139]). Table A.1 in the Appendix reports the results of the CU KS and dispersion tests applied to the partition corresponding to the starting point x^0, considering $m = 13$ weeks. In particular, in Table A.1 for each one-hour slot the sample size k_i is reported along with the p-value and the rejection/not rejection of the null hypothesis of the corresponding test. It can be observed that the arrivals are not overdispersed in any interval of the partition corresponding to x^0, i.e., all the constraints $h_i(x) \leq 0$ are satisfied and this allows combining data for the same day of the week over successive

weeks. However, this partition is even infeasible, i.e., $g_i(x) > 0$, for some i; this corresponds to reject the statistical hypothesis on some T_i. Notwithstanding, even if the starting point is infeasible, the optimization algorithm is able to find an optimal solution.

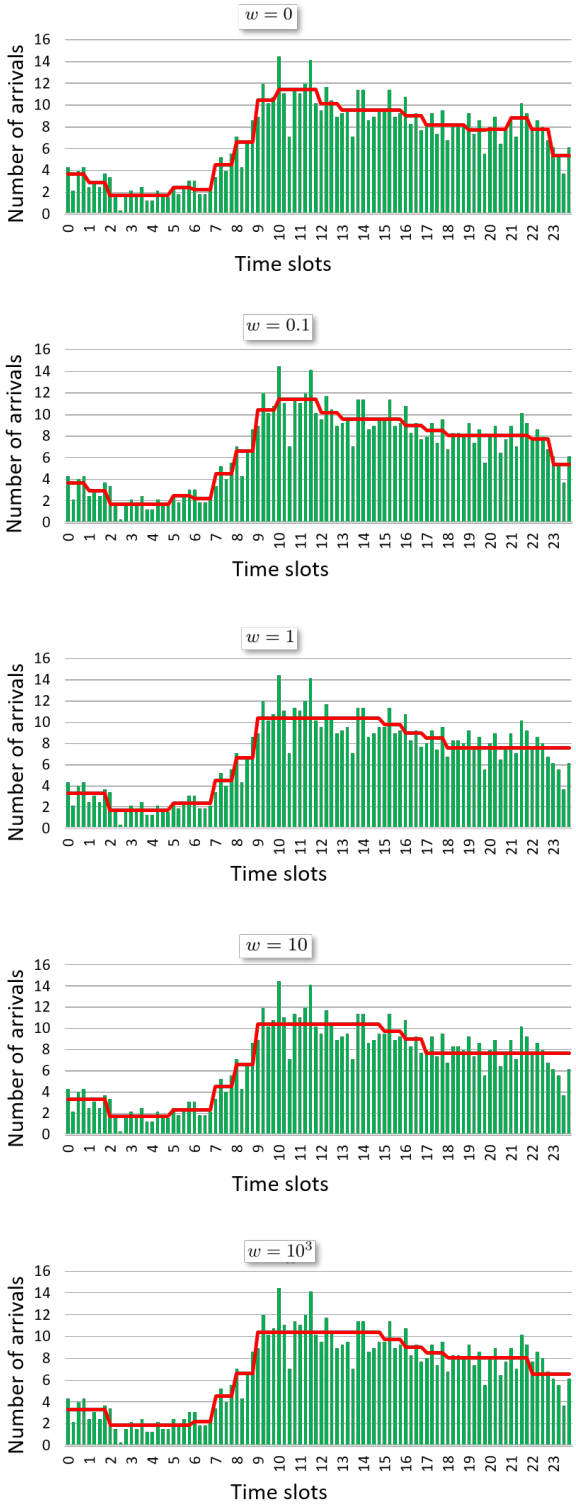
As already mentioned, the choice of a proper value for the weight w in the objective function (5.1.7) is not straightforward. Moreover, the number m of the weeks considered also affects both the accuracy of the approximation, through the average rates estimated on each interval, and the consistency of the results, which is ensured by constraints (5.1.10) and (5.1.11). However, while w is related to the statement of the optimization problem (5.1.12) and it can be arbitrarily selected, the choice of m is strictly connected to the available data. In [139, Section 4], the authors assert that, having 10 arrivals in the one-hour slot 9–10 a.m., it is necessary to combine data over 20 weeks in order to have a sufficient sample size (200 patient arrivals). However, being their approach based on equally-spaced intervals, one-hour slots are also adopted during off-peak hours, for instance during the night. This implies that the sample size resulting from the combination of the data over 20 weeks might not be sufficient to guarantee good results on these slots. This is clearly pointed out in Table A.1, where the sample size k_i corresponding to some of the one-hour night slots is very low considering $m = 13$ weeks and it remains insufficient even if 26 weeks are considered (see subsequent Table A.3). The approach proposed overcomes this drawback since, for each choice of m, the length of the intervals is determined as solution of the optimization problem (5.1.12). Of course, there might be values of m such that Problem (5.1.12) does not have feasible solutions, i.e., a partition such that the NHPP hypothesis holds and the results are consistent does not exists for such m.

In order to deeper examine how the parameters w and m affect the optimal partition, a sensitivity analysis is performed focusing first on the case with fixed m and w varying. In particular, the value chosen for m is 13 weeks, which enables achieving an optimal solution by running the optimization algorithm without high computational burden. Anyhow, no substantial changes in the conclusions are expected with different values of m and this is confirmed by further experimentation whose results are not reported here for the sake of brevity.

This analysis allows obtaining several partitions that may be considered for a proper fine-tuning of w. In particular, different values of w are considered within the set $\{0,\ 0.1,\ 1,\ 10,\ 10^3\}$. Table A.2 in the Appendix reports the optimal partitions obtained by solving Problem (5.1.12) for these values of w. In particular, Table A.2 includes the intervals of the partition, the value of the sample size k_i corresponding to each interval over 13 weeks and the results of the CU KS and dispersion tests, namely the p-value and the rejection/not rejection of the null hypothesis of the corresponding test.

For the sake of graphical comparison, Figure 7.11 reports the plots of the empirical arrival rate model $\lambda_F(t)$ and its piecewise constant approximation $\lambda_D(t)$ corresponding to the optimal partition obtained. Two effects can be clearly observed as w increases: on the one hand, on steep sections of $\lambda_F(t)$, shorter intervals are adopted to reduce large gaps between adjacent intervals; on the other hand, when $\lambda_F(t)$ is approximately flat, a lower number of intervals may be sufficient to guarantee small gaps. This is confirmed by the two top plots in Figure 7.11, which correspond to $w = 0$ and $w = 0.1$. In fact, in the first plot ($w = 0$), where only the fit error is included in the objective function, and in the second one ($w = 0.1$), where anyhow the fit error is the dominant term of the objective function, the optimal partition is composed by a relatively large number of intervals. In particular, in the partition corresponding to $w = 0.1$, fewer intervals are adopted during the daytime. As

Figure 7.11. Graphical comparison between the empirical arrival rate model $\lambda_F(t)$ (in green) and the piecewise constant approximation $\lambda_D(t)$ (in red) corresponding to the optimal partition obtained by solving Problem (5.1.12) (with $m = 13$) for different values of the parameter w. From top to bottom: $w = 0, 0.1, 1, 10, 10^3$.



expected, a smaller number of intervals is attained when $w = 1$, $w = 10$ and $w = 10^3$. Note that, since on the steep section corresponding to the time slot 7:00–10:00 a.m. the maximum number of allowed intervals (due to the lower threshold value of one hour given by the choice $\ell = 1$ in (5.1.9)) is already used, the only way to decrease the smoothness term of the objective function is to enlarge the intervals during both the day and the night. It is worth noting that for $w = 10^3$, the number of intervals increases if compared with the case $w = 10$. This occurs to offset the increase in the fit error term due to the use of a smaller number of intervals on the flatter sections. As a consequence, the partition has an unexpected interval at the end of the day.

It is important to point out that for each value of w, the optimization algorithm finds an optimal partition (of course feasible with respect to all the constraints), despite some constraints related to the CU KS test are violated at the initial partition, i.e., the one corresponding to x^0 in (7.3.1), namely the standard assumption of one-hour slots usually adopted. This means that the data is in accordance with the NHPP hypothesis and it is sufficient to appropriately define the piecewise constant approximation of the ED arrival rate.

Conversely, when the optimization algorithm does not find a feasible partition, the CU KS test or the dispersion test related to some T_i are never satisfied. This implies that the process is not conforming to the NHPP hypothesis or that the data is overdispersed. This is clearly highlighted by our subsequent experimentation, where w is set to 1, letting m vary within the set $\{5, 9, 17, 22, 26\}$.

First, in Table A.3 of the Appendix, the results of CU KS and dispersion tests are reported when applied to the partition corresponding to the starting point x^0 in (7.3.1), for these different values of m. Once more, this table evidences that using equally spaced intervals of one-hour length during the whole day can be inappropriate. As an example, see the results of the tests on the time slot 02:00–03:00. Moreover, note that the initial partition corresponding to the starting point x_0 is infeasible for all these values of m, except when $m = 5$. Indeed, the constraints corresponding to CU KS and dispersion tests are violated for some T_i, meaning that the validity of the standard assumption of one-hour time slots strongly depends on the time period considered for using the collected data. To this end, a strength of the approach proposed is its ability to assist in the selection of a reasonable value for m. If there is no value of m such that the optimization algorithm determines an optimal solution (due to infeasibility), then it may be inappropriate to consider the ED arrival process in hand as NHPP.

The subsequent Table A.4 includes the optimal partitions obtained by solving Problem (5.1.12) for the values of $m \in \{5, 9, 17, 22, 26\}$. Like the previous tables, Table A.4 includes the intervals of the partition, the value of the sample size k_i corresponding to each interval, and the results of the CU KS and dispersion tests. For all the values of m considered, the optimization algorithm determines an optimal solution with the only exception of $m = 26$. In this latter case, the maximum number of function evaluations allowed is not enough to compute an optimal solution: in fact, an infeasible solution is obtained since the CU KS test related to the last interval of the day is not satisfied. This might be partially unexpected, since more accurate results should be obtained when considering a greater sample size. However, by adding the last four weeks (passing from $m = 22$ to $m = 26$), which correspond to the month of June, the data becomes affected by a seasonal trend and the NHPP assumption is no longer valid.

Figure 7.12 reports a graphical comparison between the empirical arrival rate model $\lambda_F(t)$ and the piecewise constant approximation $\lambda_D(t)$ corresponding to the optimal partitions obtained for the values of m considered. It can be observed that the variability of $\lambda_F(t)$ reduces as the value of m increases since averaging on more

data leads to flattening the fluctuation. Despite these rapid oscillations and unlike the other values of m considered, for $m = 5$ the empirical model $\lambda_F(t)$ shows a constant trend during both the night and day hours. This results in a piecewise constant approximation $\lambda_D(t)$ that is flat in all the time slots of the 24 hours of the day except the ones related to the morning hours, for which many intervals are used. In fact, to guarantee a good fitting error between $\lambda_D(t)$ and $\lambda_F(t)$, it would be necessary to use shorter intervals, but this is not allowed by the choice $\ell = 1$ in the constraints (5.1.9). For the other values of m considered, the number of intervals increases, leading to partitions that improve the fitting error if compared with the case $m = 5$. In particular, the piecewise constant approximation $\lambda_D(t)$ obtained for $m = 22$ is observed to benefit from the lower fluctuations resulting from averaging more data. Therefore, as expected, using the maximum number of available data leads to the most accurate piecewise constant approximation. However, when considering too much data, seasonal phenomena might give rise to the rejection of the null hypothesis of the tests considered, as observed for the case $m = 26$. Moreover, as highlighted at the end of Section 5 in [139], a tendency to reject the NHPP hypothesis (i.e., the null hypothesis of the CU KS test) may be encountered when the sample size is large. In fact, a larger sample size requires a stronger evidence of the null hypothesis in order for the test to be passed. Notwithstanding, the approach proposed is able to overcome these drawbacks, providing an optimal strategy to identify the best way of using the collected data.

7.4 Model calibration

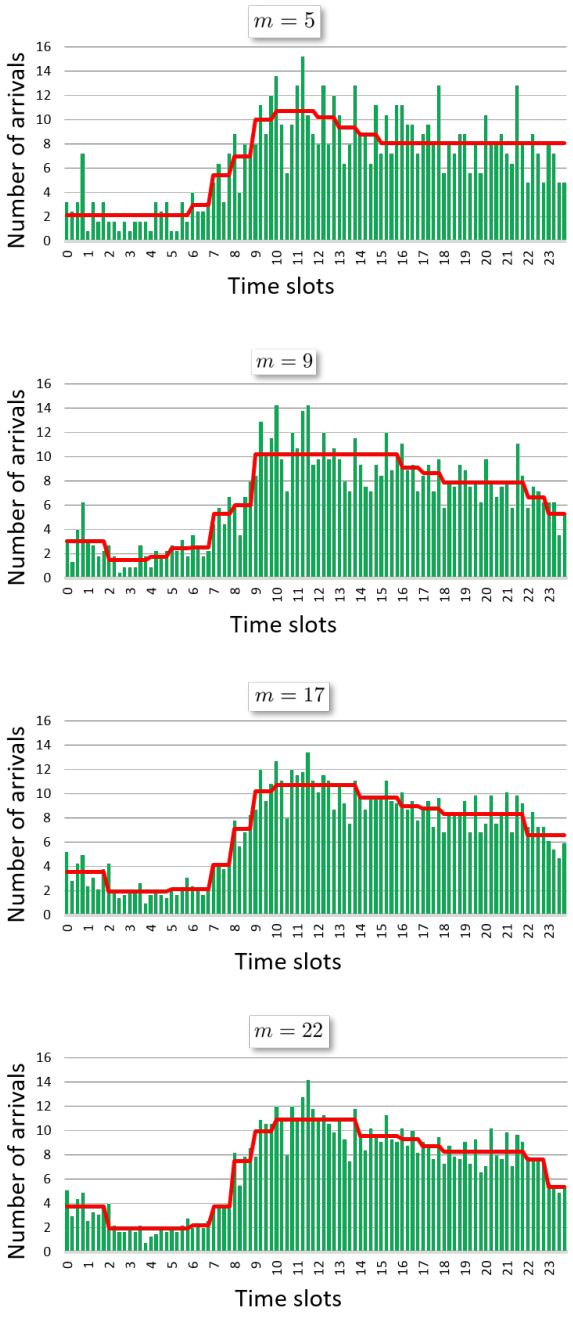
The aim of the DES model of the ED of Policlinico Umberto I is to provide a reliable tool for assessing the impact of changes to the current settings on the overcrowding level. To this end, verification and validation are important steps to be carefully considered. While standard techniques have been adopted to verify the model, such as debugging and model trace, validation has required a more sophisticated analysis. Indeed, a model calibration procedure has been used to achieve an accurate simulation output that well reproduces the real system data.

The sets defined in Section 5.2.2 can be adapted to this case study as follows.

- Let $C = \{\text{W,G,Y,R}\}$ be the set of the triage tags.
- Let $U(c) \subseteq \{\text{MU}, \text{SU}, \text{RA}, \text{MIU}\}$ be the set of the ED units where patients with tag $c \in C$ can be visited and treated. In particular,
  - $U(W) = \{\text{MIU}\}$,
  - $U(G) = \{\text{MU}, \text{SU}, \text{MIU}\}$,
  - $U(Y) = \{\text{MU}, \text{SU}\}$,
  - $U(R) = \{\text{MU}, \text{SU}, \text{RA}\}$.

For the specific instance represented by this case study, 9 different pairs of parameters are considered for the Weibull distributions representing the service times of medical visit and exams. Each pair is associated with an element of the sets $U(c)$. Instead, 8 pairs are used for the probability distributions of triage for the reasons stated in Section 7.2, thus resulting in 26 overall pairs of parameters, i.e., 52 unknown parameters to be determined through the calibration procedure. Further parameters treated as decision variables are the upper boundary parameters of the uniform probability distributions introduced to match the setup and idle times that affect the real system data, thus increasing the number of variables to 60 (a total of 8

Figure 7.12. Graphical comparison between the empirical arrival rate model $\lambda_F(t)$ (in green) and the piecewise constant approximation $\lambda_D(t)$ (in red) corresponding to the optimal partition obtained by solving Problem (5.1.12) (with $w = 1$) for different values of the parameter m. From top to bottom: $m = 5, 9, 17, 22, 26$.



uniform distributions are considered among the patient flows associated with the four color tags).

A crucial and preliminary step is the choice of the starting point of the optimization, given by x^0, y^0, and z^0 (where x, y, and z are the vectors containing all the corresponding shape and scale parameters, as defined in Section 5.2.2), which appears not to be straightforward since the missing data prevents the computation of good initial values for the parameters. Instead of starting from randomly generated values, a better strategy is to leverage the known information in a reasonable manner. While the parameters x_{cu} of the triage probability distributions are arbitrarily fixed so that generated the service times are in accordance with the ED staff suggestions, for the parameters of the medical visit and exams distributions a different approach is followed. Indeed, in addition to the timestamps shown in Figure 5.1, which are the most commonly known timestamps in every ED, for this specific case study further available information is represented by the time at which the physician requires exams. In general, this happens during the medical visit, although sometimes exams are required throughout the patient flow when periodic check-ups are performed. To provide the parameters y_{cu} of the medical visit probability distributions with initial values, a good starting point can be obtained by computing for each patient the duration between the start of the visit and the latest time at which an exam is required. The latter time is considered within a reasonable period from the start of the visit that should reflect the medical visit's maximum service time. Once this timestamp is identified and the durations are consequently available, their values can be used to initialize the parameters by fitting the corresponding Weibull probability distributions. Moreover, the time between the presumptive end of the visit and either the start of the next visit or the discharge can be used in the same manner to obtain good initial values for the parameters z_{cu} of the exams distributions.

Since the objective and constraint functions compare the real system data and the simulation output, achieving adequate accuracy is imperative for a fair comparison. To this end, the simulation output is estimated through 30 independent simulation replications, each of them 38 days long, with a warm up period of 7 days, thus matching the 31 days of January.

Important KPIs to describe the ED status are the two time differences contained in $\mathcal{T}$ and the number of entities generated by the simulation and associated with each color tag. While the former KPIs are part of the objective function and are calculated during the calibration process performed by the optimization algorithm, the latter do not require a continuous monitoring, since their values are not affected by the calibration. Indeed, the only variations are due to the different number of entities discharged before the simulation ends, which leads to different results in terms of KPI computation. Table 7.11 reports the values of the patient counts obtained from the simulation of the starting point of the calibration procedure along with their confidence intervals. By comparing these values with the counts in Table 7.6, it can be observed that the simulation model provides an accurate output in terms of number of patients.

7.4.1 Experimental results

Although Problem (5.2.1) is a continuous optimization problem, its variables are considered as discrete to efficiently solve the problem. In particular, at the time of solving the problem, x_p^{cu}, y_p^{cu}, and z_p^{cu}, with $p \in \{1, 2\}$, are treated as granular variables, i.e., variables with a controlled number of decimals (see, e.g., [25]). This choice is motivated by the fact that the output of the simulation model is insensitive to small changes in the values of the decision variables, thus making discrete variables

Table 7.11. Output values (with confidence interval) for the number of patients returned by the simulation of the starting point of the calibration procedure.

	White	Green	Yellow	Red
SU		609 ± 8.10	704 ± 7.72	54 ± 3.44
MU		264 ± 5.83	1318 ± 21.51	220 ± 5.54
RA				20 ± 1.45
MIU	42 ± 2.47	241 ± 6.20		

preferable. To this end, a specific granularity is considered based on the role of the parameter associated with the variable in the corresponding probability distribution. All the variables are pairs of shape and scale parameters, which have a different impact on the simulation output. Indeed, while the former parameters determine the shape of the probability distribution, the latter affect the scale of the values generated. With respect to the Weibull distribution adopted, in practice the behavior observed is that the larger the value of the shape parameter, the larger the dispersion of the values of the random variables associated. Moreover, such random variables take on values with a larger scale as the value of the scale parameter increases. Since some preliminary analyses have shown that the impact on the simulation output, measured in terms of the two KPIs DOT and DIT, is more noticeable for variations in the scale parameter, for each pair of variables the granularity δ_p^{min} of the values is fixed to 1 if $p = 1$ (i.e., shape parameter) and to 0.1 if $p = 2$ (i.e., scale parameter). Note that treating the variables as granular requires the new variables to be equal to $x_p^{c\,u}/\delta_p^{min} \in \mathbb{Z}$, thus meaning that $x_p^{c\,u}$ have to take on real values that are multiple of δ_p^{min}. The same type of constraint applies to the variables of the visit and exams probability distributions.

To solve the SBO problem for calibrating the simulation model of the ED in hand, the approach of the SAA is adopted. As a consequence, the resulting optimization problem is deterministic and it can be solved by applying an algorithm from the class of DFO. In particular, the optimization algorithm proposed in [171] is used for solving the problem and its default values are adopted for the parameters. The maximum number of function evaluations, which represents the stopping condition, is set to 2500. Note that, by using the SAA approach, the empirical cumulative distribution functions used in the optimization problem are estimated through the corresponding sample means over the 30 independent simulation replications.

As concerns the optimization problem to solve, the values of the lower and upper bounds introduced in Problem (5.2.1) are reported in Table 7.12. Moreover, in the numerical experimentation described in the sequel, the constraints on the sample variances are omitted and the tolerance $\tau_\mu^{c\,u\,i}$ is fixed to 0.3 for yellow and green-tagged patients visited and treated at MU and SU, 0.2 otherwise.

Table 7.12. Values of the lower and upper bounds for each pair of variables $(x_1^{c\,u}, x_2^{c\,u})$, $(y_1^{c\,u}, y_2^{c\,u})$, and $(z_1^{c\,u}, z_2^{c\,u})$, for all $c \in C$ and $u \in U(c)$. .

$l_{x_1^{c\,u}}$	$l_{x_2^{c\,u}}$	$l_{y_1^{c\,u}}$	$l_{y_2^{c\,u}}$	$l_{z_1^{c\,u}}$	$l_{z_2^{c\,u}}$
0.01	0.01	0.01	0.01	0.01	0.01
$u_{x_1^{c\,u}}$	$u_{x_2^{c\,u}}$	$u_{y_1^{c\,u}}$	$u_{y_2^{c\,u}}$	$u_{z_1^{c\,u}}$	$u_{z_2^{c\,u}}$
1000	0.5	1000	4	1000	40

Given these settings, the experimental results, which are obtained by using a PC with Intel Core i7-4790K quad-core 4.00 GHz Processor and 32 GB RAM, are

shown in Appendices B and C. The former reports the comparison between the histograms of the values of the time differences *DOT* and *DIT* computed for each patient through the real data and the simulation output resulting from the model calibration procedure, respectively. The latter reports the comparison in terms of the Empirical Cumulative Distribution Functions (ECDFs). In both appendices, the comparisons are performed for all the color tags $c \in C$ and for all the ED units $u \in U(c)$ where a patient with tag c can be assigned. Moreover, both histograms and ECDFs are obtained as an average over the independent simulation replications. The choice of using two types of plots to assess the effectiveness of the model calibration procedure is due to the different reliability of the information provided. In particular, compared to ECDFs, the description of the data provided by histograms is strongly affected by the choice of the width of each bin, which is not straightforward and may lead to distributions with different shapes based on how data is grouped. Both types of plots show that the shape of the histograms and ECDFs that correspond to the system data is well reproduced by the simulation output. However, strong dissimilarities may be still present in spite of the calibration, especially for green and yellow-tagged patients. This is due to the difficulty in reproducing the patient flow when sharing of the resources is involved between patients with different triage tag, as is the case for the medical visit of green and yellow-tagged patients at both *MU* and *SU*. Moreover, the experimental results show that, in general, the comparison between real data and simulation output is more accurate for the time differences DIT. Indeed, larger errors are observed in both types of plots when time differences DOT are considered due to the difficulty in reproducing the actual waiting times of patients, which depend both on the service time of the medical visit and on the number of patients that are in queue. Hence, it is the result of a *seize-delay-release* process used within the simulation, while DIT is given by the mere sum of service times (except for patients that need more than one medical visit).

By observing the ECDF plots in Appendix C, it is also possible to gain insight that histograms fail to provide. For instance, the ECDF related to yellow-tagged patients at SU shows that the time differences DOT collected from the simulation model are systematically larger than the corresponding real values, meaning that either the estimated parameters of the probability distribution of the medical visit service time or the modeling of the interaction with green-tagged patients in SU may not be appropriate. To avoid ambiguities, it is important to remark that since the simulated ECDFs are obtained as an average over independent simulation replications, the jumps (or steps) in these simulated functions correspond to patients observed across all the replications. This is a consequence of averaging different step functions, such as the ECDFs, which have jumps at different points in their domain.

The same conclusions on the accuracy of the simulation model can be drawn also by observing Figures 7.13–7.14, which report the comparison between the average current and simulated values of DOT and DIT for all the triage tags and all the ED units. As already mentioned, the simulation output related to the time differences DIT is a good approximation of the real system values since for each color tag the current values are either within the corresponding confidence interval or close to it. Instead, even though for the time differences DOT the accuracy of the simulation responses as estimates of the current values is lower, the relative error is within the tolerances $\tau_{\mu}^{c\,u\,i}$ chosen. Indeed, all the constraints of Problem (5.2.1) are (of course) satisfied at the optimal solution obtained by applying the model calibration approach. Note that reducing the values of the tolerances potentially allows achieving more accurate solutions, however, the optimization algorithm is not guaranteed to determine a feasible solution.

The experimental results discussed above show that the model calibration proce-

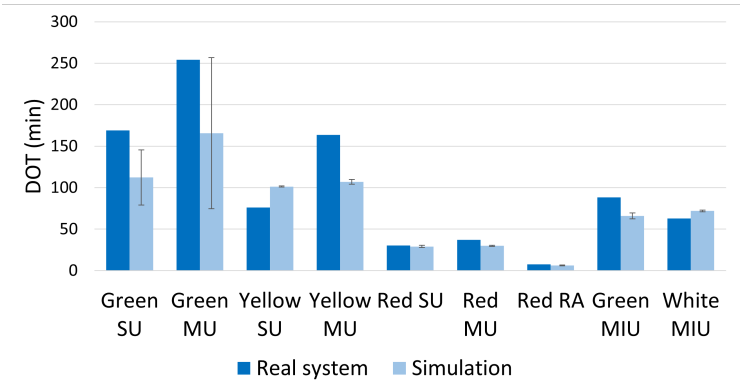


Figure 7.13. Plot of current values and simulation output of DOT with the confidence interval.

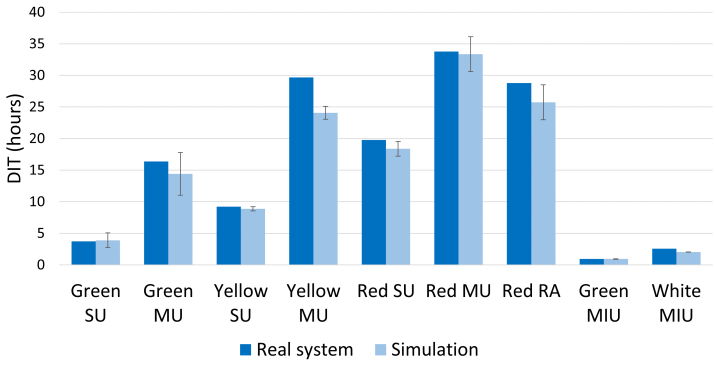


Figure 7.14. Plot of current values and simulation output of DIT with the confidence interval.

dure enables estimating the missing parameters in order for the simulation model to satisfactorily reproduce the ED operations despite the unavailable timestamps, which represent the main hurdle for achieving a high level of accuracy. The constraints on the sample means used in Problem (5.2.1) ensure that the responses of the simulation model are on average close to the corresponding real values. However, when looking more closely into the inner details of the simulation, some dissimilarities between real data and simulation output may be observed as a result of the impact of the problem of missing data, which necessarily undermines the overall accuracy. Notwithstanding, the simulation model may be deemed sufficiently reliable with respect to the specific objectives of the analysis.

Since there is still a wide margin for improvement, several ideas are considered for further research. For instance, introducing the constraints on the comparison between the real and simulated variances of the KPIs may help the optimization algorithm to avoid points associated with inaccurate results, such as those observed for yellow-tagged patients. Moreover, since the worst results are observed for the time differences DOT, a significant improvement may be obtained by using different weights for the terms of the objective function of Problem (5.2.1) and assigning larger values to the terms associated with DOT. In general, different objective functions and starting points may be taken into account also to assess the robustness of the approach. For example, a nonmonotone algorithm could be used. Finally, increasing the number of function evaluations used as stopping condition may lead to better solutions allowing a more thorough exploration of the feasible region.

7.5 A resource allocation problem for reducing the overcrowding

The simulation model of the ED of Policlinico Umberto I is used to achieve the final goal of finding a setting that allows reducing the level of overcrowding, which is measured through the waiting times at each ED unit. By interviewing the ED managers, a great interest emerged about exploring the existence of a margin for visiting and treating a significantly larger number of green-tagged patients in MIU. This request is motivated by the fact that nurses in charge of assigning the color tag at triage tend to assume a precautionary behavior, thus sending fewer patients to MIU than its actual capacity. Indeed, they would be deemed responsible in case of worsening patient health conditions caused by a wrong choice of the ED unit at triage. However, reducing the overcrowding by leveling out the workload within the ED units requires to carefully monitor the overall number of patients to avoid that the benefit achieved in one unit gives rise to long waiting times in other segments of the ED. Sending more patients to MIU is only one of the options available to achieve the final goal of reducing the overcrowding. A further tool considered by the ED managers is the choice of the working hours of MIU, which is currently open from 8.00 a.m. to 8.00 p.m., Monday through Saturday. Due to the high flexibility of the ED staff, every combination of opening and closing time for MIU on each day of the week is considered feasible by the managers if it guarantees an improvement in the current status (provided that the working hours are not out of the current window 8.00 a.m.–8.00 p.m.). It is important to point out that the largest reduction in the waiting times is expected for low-complexity patients, namely patients with green or white triage tag, since these are the categories assigned to MIU. The most critical patients, namely patients with yellow or red tag, may experience a lower reduction in the waiting times since their priority guarantees a shorter time in the queue even in normal conditions. Anyhow, some benefits are expected, since in the

ED units they will share resources with a lower number of low-complexity patients.

The resource allocation in the heading of this section is intended in a broad sense since the term resource is here referred to the working hours of MIU for each day of the week. However, it is also possible to claim that each working hour requires to use certain amounts of resources, thus linking the class of resource allocation problems to the specific problem in hand in a more straightforward way. Moreover, other decision variables considered are the number of rooms used in MIU for the medical visit and the percentage of green-tagged patients that are assigned to MIU by the nurse in charge of triage. Although the nurse makes this decision based on the seriousness of patient's health, the choice of considering this percentage as a decision variable aims to demonstrate the potentially significant improvement that may be obtained if the optimal number of green-tagged patients is assigned to MIU. The hope is that the potential benefit shown from the experimental results can encourage the nurse to send a larger number of patients to MIU, thus reducing the workload at SU and MU. However, it is important to remark that the solution may be only ideal and not of practical implementation in the real system since this choice is strictly related to the gravity of the patient.

Diverting patients with low-acuity illnesses and injuries is a strategy analyzed in many studies which aim to assess its impact on reducing the waiting times and, accordingly, the ED overcrowding [126, 210, 133, 52, 200, 193]. Generally speaking, the aim of adopting such a diversion policy, like the fast-track system, is to reduce the ED overcrowding by discharging the patients earlier. In particular, [200] points out that the expected benefits of using fast-track systems are observed in every ED, regardless of the single case studies considered. In [210], the authors show that reducing the number of low-complexity patients does not significantly affect the waiting times of the high-complexity patients. This result is supposed to hold when the resources in charge of the visit and treatment of low-acuity patients are dedicated. Contrarily, when such resources are shared with critical patients, a worsening in the total average waiting times of the most urgent patients may be experienced [152]. Moreover, the authors in [152] show that most benefits of the fast-track system are observed in EDs with a considerable number of urgent patients. Indeed, the improvement in terms of waiting time is higher when the low-complexity patients can bypass a larger number of patients. Although some papers use simulation modeling to assess the effectiveness of the diversion policy [133, 126, 152], to the best of our knowledge, papers using SBO are still missing. Indeed, while many papers examine the impact of the adoption of these strategies through scenario-based analyses, only a few of them aim to find the settings providing an effective diversion policy. For instance, in [215], a multi-criteria method is proposed to determine the best configuration for the fast-track system in terms of five performance indicators. Instead of optimizing KPIs to determine the settings of the diversion policy applied, [108] uses an optimization model to find optimal values of the ED parameters; the resulting simulation model allows the assessment of fast-track strategies through a scenario analysis.

7.5.1 Statement of the simulation-based optimization problem

Now let us formally define the simulation-based optimization problem. Let W be the set of the days of the week, namely $W = \{1, \dots, 7\}$. Since the level of overcrowding in the ED units is measured in terms of patient waiting time, the simulation output considered in this problem is the waiting time Y_{cu} of a patient with tag $c \in C$ visited and treated in unit $u \in U(c)$, where C and $U(c)$ are the sets defined in Section 7.4. The decision variables are introduced as follows.

- Let $x_w \in \mathbb{Z}$ be the opening time of MIU on day $w \in W$.
- Let $v_w \in \mathbb{Z}$ be the closing time of MIU on day $w \in W$.
- Let $z \in \mathbb{Z}$ be the number of rooms used in MIU.
- Let $p \in \mathbb{R}$ be the percentage of green patients assigned to MIU during its working hours.

Given positive weights α_c, β_u, γ_w, and ρ, the optimization problem is formally stated as follows

$$
\begin{aligned}
\min_{x,v,z,p} \; & f_1(x,v,z,p) \equiv \sum_{c\in C} \alpha_c \sum_{u\in U(c)} \beta_u \, \mathbb{E}[Y_{c\,u}(x,v,z,p,\xi)] \\
\min_{x,v} \; & f_2(x,v) \equiv \sum_{w\in W} \gamma_w \, (x_w - v_w) \\
\min_{z} \; & f_3(z) \equiv \rho \, z \\
s.t. \; & x_w - v_w \le 0 \qquad \text{for all } \; w \in W, \\
& l_{x_w} \le x_w \le u_{x_w} \qquad \text{for all } \; w \in W, \\
& l_{v_w} \le v_w \le u_{v_w} \qquad \text{for all } \; w \in W, \\
& l_z \le z \le u_z, \\
& l_p \le p \le u_p, \\
& x \in \mathbb{Z}^{|W|}, v \in \mathbb{Z}^{|W|}, \; z \in \mathbb{Z}, \; p \in \mathbb{R}.
\end{aligned}
\tag{7.5.1}
$$

Since three real–valued objective functions are considered, which are defined as $f_1 : \mathbb{Z}^{|W|} \times \mathbb{Z}^{|W|} \times \mathbb{Z} \times \mathbb{R} \to \mathbb{R}$, $f_2 : \mathbb{Z}^{|W|} \times \mathbb{Z}^{|W|} \to \mathbb{R}$, and $f_3 : \mathbb{Z} \to \mathbb{R}$, the problem is multiobjective. The first objective function is the weighted sum over the sets C and $U(c)$ of the patient waiting times at the ED units. Note that when the terms of f exceed the corresponding maximum average waiting time recommended by the guidelines (see Table 1.3), f_1 can be considered as a direct measure of overcrowding. The weights α and β control the importance of the color tag and the ED unit for minimizing the waiting times. Larger values should be assigned to the patient categories for which bigger improvements are required. The second objective is the minimization of the working hours of MIU over each day of the week, while the third objective is the minimization of the number of rooms. In general, the functions f_2 and f_3 have the same impact on the minimization of f_1: the longer MIU is open and the more rooms are used, the lower the overall waiting times in the ED are. Therefore, if γ_w and ρ are assigned small values, at the final solution found by the optimization algorithm, MIU is expected to be open as much as possible and to use all the available rooms. However, note that at the final solution the percentage p is increased compared to the "as–is" status, i.e., more green-tagged patients are assigned to MIU, only if the grow in waiting times observed at MIU, which becomes more crowded, is balanced with a larger reduction in the other terms of f_1. Finally, in addition to the bound constraints used to specify the range of variation of each variable, one linear constraint is adopted to require the closing time of MIU on each day of the week to be larger than the corresponding opening time.

7.5.2 Experimental results

The simulation model resulting from the calibration procedure described in the previous section is used to evaluate the simulation output $Y_{c\,u}$ used in the

objective function f_1. Due to the problem of missing data, the waiting time Y_{cu} is approximated through the time difference DOT, which includes the waiting time before the medical visit as well as the service time of triage. Since triage is relatively short compared to the waiting time, using DOT appears to be a reasonable choice. In the remainder of this section, DOT will be referred to as a waiting time. Retaining the design of experiments chosen for the model calibration, an accurate estimate of the average waiting time for each color tag is obtained by adopting 30 independent simulation replications, each of them 38 days long, with a warm up period of 7 days.

Although the variable p is supposed to be a real number in the general formulation described above, its discretization is required to perform the experimental results, since the simulation output is almost insensitive to small variations of this decision variable. Therefore, following the approach used in Section 7.4.1 for the model calibration procedure, a granularity parameter δ^{min} is introduced to treat p as a granular variable. In particular, δ^{min} is fixed to 5 to ensure that the steps performed on p by the optimization algorithm guarantee sufficiently large values, in order for the variation of the response of the simulation output not to be negligible.

The approach of the SAA is adopted to deal with the expected value of the simulation output, which is estimated through the sample mean over the independent simulation replications. Therefore, once the realizations of the random variables involved in the DES model are fixed, the resulting integer simulation-based optimization problem is deterministic and it can be solved multiple times by using different sets of realizations, in order to obtain a good estimate of the true optimal solution. The DFO algorithm proposed in [171] is used with the default settings of its parameters. The stopping criterion, based on the maximum number of function evaluations, is set to 1000.

The values of the lower and upper bounds introduced in Problem (7.5.1) are reported in Table 7.13. Moreover, in the numerical experimentation described in the

Table 7.13. Values of the lower and upper bounds for the variables x_w, v_w, z, and p, with $w \in W$.

l_{x_w}	l_{v_w}	l_z	l_p	u_{x_w}	u_{v_w}	u_z	u_p
8	8	1	0	20	20	4	100

sequel, the third objective is omitted by fixing ρ to 0, while α_c, β_u, and γ_w are all fixed to 1. The weighted-sum method is used to deal with the resulting biobjective problem, following the approach described in [175]. Therefore, the two objective functions f_1 and f_2 are aggregated into a single weighted function.

For the experimental results, a PC with Intel Core i7-4790K quad-core 4.00 GHz Processor and 32 GB RAM is used to run the optimization algorithm. Figure 7.15 reports the approximate Pareto front obtained by different minimizations of the single–objective problem, each one associated with a different combination of the weights used for scalarization, namely

$$\eta_1 f_1(x, v, z, p) + \eta_2 f_2(x, v),$$

where η_1 and η_2 are nonnegative parameters such that $\eta_1 + \eta_2 = 1$. The point in blue is defined by the values of f_1 and f_2 obtained at the starting point of every minimization, which corresponds to the current operating conditions ("as–is" status) and it is is reported in Table 7.14. It is worth remarking the trade-off between the two objectives considered, which is clearly shown in the previous figure, where high values of the total waiting time correspond to a lower number of MIU weekly working hours, and vice versa.

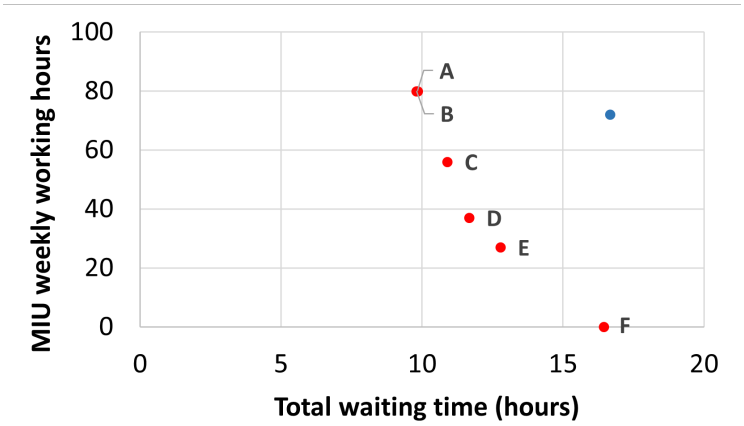


Figure 7.15. Approximate Pareto front obtained by different minimizations of the single–objective problem. The point in blue corresponds to the starting point.

Table 7.14. Starting point of every single–objective minimization. It corresponds to the "as–is" status.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	v_1	v_2	v_3	v_4	v_5	v_6	v_7	z	p
8	8	8	8	8	8	8	20	20	20	20	20	20	8	2	30

A comparison among the waiting times associated with four different settings is shown for each color tag in Figure 7.16. In particular, the maximum average waiting time recommended by the guidelines is compared with the average waiting times resulting from the current operating conditions and with the average waiting times obtained by simulating two points of the Pareto front. These points are associated with the combinations of weights given by $\eta_1 = 1$ and $\eta_2 = 0$ (*point A*) and by $\eta_1 = 0$ and $\eta_2 = 1$ (*point F*), which represent extreme conditions. Note that to obtain point F it is not necessary to run the optimization algorithm, since the minimum value of the objective function is trivially attained when $x_w = v_w$ for all $w \in W$, namely, MIU is never open. However, simulation is needed to estimate the total waiting time resulting from the settings associated with point F. The figure shows that, in the "as–is" status, the largest deviations from the maximum average waiting times recommended by the guidelines are observed for yellow and green-tagged patients. Moreover, although the waiting time of red-tagged patients exceeds only by a small amount the maximum limits, their wait cannot be considered negligible due to their life-threatening conditions. Note that at point F the white tag is not assigned to any patient since MIU is closed. Therefore, only the green tag is assigned to low-complexity patients, thus resulting in longer waiting times due to the overcrowding of the ED units where green-tagged patients are visited and treated. This is expected, since point F corresponds to the settings minimizing f_2, namely, the ED management cost.

The different amount of improvement from the current average waiting time obtained for each color tag at point A evidences that the proposed approach is effective mainly for low-complexity patients. As already remarked, this is expected, since only white and green-tagged patients are directly affected by the settings of MIU. It is also important to point out that the difference between the waiting times of green and white-tagged patients in the "as–is" status and from the settings corresponding to point A is mainly due to the value of p, as shown by comparing the

values of p in the Tables 7.14–7.15. Since the value of p at point A is less than the actual value, it is possible to conclude that sending more green-tagged patients to MIU than the current number is not the best choice when the settings corresponding to point A are considered. It is important to remark that this conclusion is affected by the stopping criterion of the optimization algorithm adopted. Indeed, allowing more function evaluations might lead to a different and improved final solution.

Table 7.15. Settings corresponding to the points A and F of the approximate Pareto front.

Point	x_1	x_2	x_3	x_4	x_5	x_6	x_7	v_1	v_2	v_3	v_4	v_5	v_6	v_7	z	p
A	8	8	8	8	8	8	8	20	20	20	20	20	19	17	2	20
F	8	8	8	8	8	8	8	8	8	8	8	8	8	8	2	30

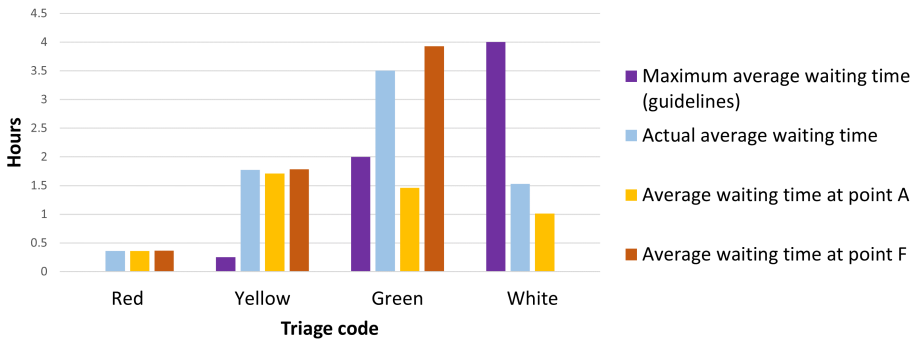


Figure 7.16. Comparison among the waiting times for each color tag associated with four different settings: maximum average waiting times recommended by the guidelines, average waiting times resulting from the current operating conditions, and average waiting times obtained from the settings associated with points A and F of the approximate Pareto front.

Chapter 8

Conclusion and further research

The development of novel efficient algorithmic frameworks using simulation to provide solutions to real–world problems is prompted by the increasing need to accurately represent the complex and uncertain processes of the systems. The resulting SBO methodology has been receiving increasing attention in the recent years, aiming to develop algorithms that both do not require first-order information, due to the lack of analytical expressions for the black-box functions, and support the handling of continuous and integer variables. Both the trade-off between long-term goals and short-term decisions and the computational cost of function evaluations determine the proper algorithm to use, whether exact DFO algorithms providing optimal solutions with long running time or metaheuristic methods returning fast solutions without optimality guarantees. Important problems related to real–world applications that suit this context are provided by the field of ED management, which shows a strong interest in adopting techniques from Operations Research to study the impact of both overcrowding and sudden patient peak arrivals on everyday operations. To this end, further SBO approaches are required to estimate the ED arrival rate and the missing data in the real datasets in order to build a DES model with a high level of accuracy.

In this thesis, SBO methods have been proposed to solve problems where simulation plays a key role to model stochastic systems. In particular, a SBO approach based on a metaheuristic is used to solve an integrated allocation and scheduling problem with stochastic processing times. The algorithm has been tested over a set of stochastic instances based on the corresponding classical instances for deterministic and simplified flow–shop problems. The results show that the algorithm is able to provide good solutions in short computing times. Moreover, the computational experiments show that the expected makespan associated with each instance grows as the variability of the random processing times increases. It is important to point out that the algorithm provides a simple example of how allocation of resources and scheduling could be integrated in a unique decision process to find a better solution in terms of system performance. Of course, the extension to real–life applications could involve different strategies. The main goal of this approach is to demonstrate how simulation can be effectively integrated with optimization in problems where function evaluations are computationally cheap. In contrast with the previous approach, which is based on a metaheuristic, globally convergent algorithms are proposed for solving hard problems emerging from real–world applications, where simulation models typically require long runs. In particular, new linesearch-based methods for mixed-integer nonsmooth constrained optimization problems have been developed. Among the four algorithms proposed, numerical experiments have been performed on a set of bound constrained problems only for the most efficient algorithm in

practice. The results show that the algorithm demonstrates a remarkable efficiency and compares favorably to the state-of-the-art-solvers in terms of robustness, thus providing a noticeable contribution to the derivative-free optimization solvers. The numerical experiments on a set of test problems with general nonlinear constraints are left as future work.

To test the effectiveness of DES and SBO on real instances, two EDs have been considered as case studies. In particular, a DES model is proposed for studying the ED of a medium dimension hospital in a region of Center Italy recently hit by a severe earthquake. The aim is to assess how fundamental KPIs change in response to different increase patterns of the patient arrival rate. In particular, the focus is on extremely loaded situations for the ED, due to critical events like a natural disaster. To evaluate the performance of the EDs, time–related measurements have been considered as well as resources usage. Several scenarios have been taken into account, including someones artificially created, trying to reproduce real mass casualty occurrences. The experimental results show that, when the increase in the arrival rate is low or moderate, the ED performance does not significantly deteriorate. Instead, in case of extreme events with high patient peak arrivals, the adoption of an exceptional emergency plan is necessary to ensure effective and timely assistance. The DES model adopted refers to a specific ED, but thanks to the flexibility of its implementation, it can be easily adapted to reproduce the patient flow of other EDs. The model has a twofold merit: on the one hand, it represents an effective decision support system that allows for the assessment of the performance of the ED in hand, highlights possible operational improvements, and enables decision makers to better allocate ED resources. On the other hand, the model can be used to perform scenario analyses to help managers to define in advance maxi–emergency plans which, of course, cannot be tested during the normal activity of the ED. As future work, the model can be extended to include further components, like fast track or see and treat pathways, as well as changes of physician and nurse service rates.

A second DES model was implemented for reproducing the processes within the ED of a big hospital in Italy. This second case study allows testing the effectiveness of the two SBO approaches proposed to improve the reliability of every simulation model dealing with EDs. In particular, the arrival process to EDs was examined by providing a novel methodology that is able to improve the accuracy of the stochastic modeling of the arrivals. In accordance with the literature, the standard assumption of representing the ED arrival process as a NHPP, which is suitable for modeling strongly time–varying processes, was adopted. The final goal of the proposed approach is to accurately estimate the unknown arrival rate, i.e. the time–dependent parameter of the NHPP, by using a reasonable piecewise constant approximation. To this end, an integer nonlinear black-box optimization problem is solved to determine the optimal partition of the 24 hours into a suitable number of nonequally spaced intervals. To guarantee the reliability of this estimation procedure, two types of statistical tests are considered as constraints for each interval of any candidate partition: the CU KS test must be satisfied to ensure the consistency between the NHPP hypothesis and the ED arrivals; the dispersion test must be satisfied to avoid the overdispersion of data. The extensive experimentation performed on data collected from an ED of a big hospital in Italy shows that the approach is able to find a piecewise constant approximation which represents a good trade-off between a small fitting error with the empirical arrival rate model and the smoothness of the approximation. This result is accomplished by the optimization algorithm despite some constraints are violated at the starting point, which corresponds to the commonly adopted partition composed by one-hour time slots. Moreover, some

significant sensitivity analyses are performed to investigate the fine-tuning of the two parameters affecting the quality of the piecewise constant approximation: the weight of the smoothness of the approximation in the objective function (with respect to the fitting error) and the number of weeks considered from the arrivals data. While the former can be arbitrarily chosen according to the desired level of smoothness, the latter affects the accuracy of the arrival rate estimation. In general, the more weeks are considered, the more accurate is the arrival rate approximation, as long as the NHPP assumption still holds and the data does not become overdispersed. As regards future lines of research, in order to deeper analyze the robustness of the proposed approach, alternative statistical tests, such as the Lewis and the Log tests described in [139], could be used in place of the CU KS test.

The second SBO approach is related to the data quality problems that frequently affect the DES models concerning EDs. In particular, the focus is on the problem of missing data, which consists in the unavailability of data related to some of the starting and ending times of the activities performed in the ED. This well-known issue is responsible for the lack of knowledge of the service time of some processes, which is required to estimate the corresponding probability distributions to use in the simulation model. For this purpose, after assuming suitable probability distributions for the processes associated with the unknown service times, a model calibration procedure is proposed to estimate the missing parameters of such distributions. The experimental results show that the model calibration procedure enables estimating the missing parameters in order for the simulation model to satisfactorily reproduce the ED operations despite the unavailable timestamps, which represent the main hurdle for achieving a high level of accuracy. Although proper constraints ensure that the responses of the simulation model are on average close to the corresponding real values, when looking more closely into the inner details of the simulation, some dissimilarities between real data and simulation output may be observed as a result of the impact of the problem of missing data, which necessarily undermines the overall accuracy. Notwithstanding, the simulation model may be deemed sufficiently reliable with respect to the specific objectives of the analysis. Since there is still a wide margin for improvement, several ideas are considered for further research. For instance, introducing the constraints on the comparison between the real and simulated variances of the KPIs may help the optimization algorithm to avoid points associated with inaccurate results, such as those observed for yellow-tagged patients. Moreover, since the worst results are observed for the time differences DOT, a significant improvement may be obtained by using different weights for the terms of the objective function considered and assigning larger values to the terms associated with DOT. In general, different objective functions and starting points may be taken into account also to assess the robustness of the approach. For example, a nonmonotone algorithm could be used. Finally, increasing the number of function evaluations used as stopping condition may provide better solutions allowing a more thorough exploration of the feasible region.

By using the accurate simulation model resulting from the application of these two approaches, SBO is used for solving a resource allocation problem related to the specific case study considered. The goal is to determine the optimal settings of the ED unit devoted to the medical visit of low-complexity patients in order to reduce the overcrowding level. A multiobjective formulation of the problem is adopted to find a trade-off between the conflicting goals of reducing the management cost and guaranteeing patients timely treatments according to their urgency code. In the computational experiments, the objective functions are weighted in a single function and the resulting optimization problem is repeatedly solved by using a DFO algorithm in order to approximate the Pareto front, according to the weighted-sum

method. The experimental results show that the proposed approach is able to provide the ED managers with an effective tool to determine and assess different settings for reducing the patient waiting times by choosing the desired level of effort. Further research is required to develop or adapt a SBO algorithm devoted to multiobjective problems, without relying on the scalarization of the different objective functions.

Appendices

Appendix A

Arrival process - CU KS and dispersion tests

This Appendix reports the detailed results of the CU KS and dispersion tests related to the partitions considered in Section 7.3.

Table A.1. Results of the CU KS and dispersion tests (with a significance level of 0.05) applied to each interval of the partition corresponding to the starting point x^0. The considered number of weeks is $m = 13$. For each interval of each partition, the sample size of the dispersion test is m. H_0 denotes the null hypothesis of the corresponding test.

		CU KS test		Dispersion test	
Interval	k_i	p-value	H_0	p-value	H_0
00:00 – 01:00	48	0.836	not rejected	0.801	not rejected
01:00 – 02:00	38	0.950	not rejected	0.450	not rejected
02:00 – 03:00	22	0.027	rejected	0.521	not rejected
03:00 – 04:00	24	0.752	not rejected	0.652	not rejected
04:00 – 05:00	21	0.668	not rejected	0.366	not rejected
05:00 – 06:00	32	0.312	not rejected	0.524	not rejected
06:00 – 07:00	29	0.634	not rejected	0.538	not rejected
07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
09:00 – 10:00	136	0.635	not rejected	0.803	not rejected
10:00 – 11:00	143	0.039	rejected	0.966	not rejected
11:00 – 12:00	154	0.325	not rejected	0.999	not rejected
12:00 – 13:00	132	0.858	not rejected	0.948	not rejected
13:00 – 14:00	121	0.738	not rejected	0.984	not rejected
14:00 – 15:00	125	0.885	not rejected	0.500	not rejected
15:00 – 16:00	127	0.928	not rejected	0.610	not rejected
16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
17:00 – 18:00	111	0.769	not rejected	0.516	not rejected
18:00 – 19:00	102	0.458	not rejected	0.912	not rejected
19:00 – 20:00	100	0.095	not rejected	0.527	not rejected
20:00 – 21:00	101	0.656	not rejected	0.586	not rejected
21:00 – 22:00	115	0.763	not rejected	0.604	not rejected
22:00 – 23:00	101	0.916	not rejected	0.305	not rejected
23:00 – 24:00	70	0.864	not rejected	0.104	not rejected

Table A.2. Results of the CU KS and dispersion tests (with a significance level of 0.05) applied to each interval of the optimal partition obtained by solving problem (5.1.12) for different values of the parameter w, with m fixed to 13 weeks. From top to bottom: $w = 0, 0.1, 1, 10, 10^3$. For each interval of each partition, the sample size of the dispersion test is equal to m. H_0 denotes the null hypothesis of the corresponding test.

			CU KS test		Dispersion test	
w	Interval	k_i	p-value	H_0	p-value	H_0
0	00:00 – 01:00	48	0.836	not rejected	0.801	not rejected
	01:00 – 02:00	38	0.950	not rejected	0.450	not rejected
	02:00 – 05:00	67	0.504	not rejected	0.100	not rejected
	05:00 – 06:00	32	0.312	not rejected	0.524	not rejected
	06:00 – 07:00	29	0.634	not rejected	0.538	not rejected
	07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
	08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
	09:00 – 10:00	136	0.635	not rejected	0.803	not rejected
	10:00 – 12:00	297	0.433	not rejected	0.994	not rejected
	12:00 – 13:00	132	0.858	not rejected	0.948	not rejected
	13:00 – 16:00	373	0.958	not rejected	0.502	not rejected
	16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
	17:00 – 19:00	213	0.999	not rejected	0.937	not rejected
	19:00 – 20:00	100	0.095	not rejected	0.527	not rejected
	20:00 – 21:00	101	0.656	not rejected	0.586	not rejected
	21:00 – 22:00	115	0.763	not rejected	0.604	not rejected
	22:00 – 23:00	101	0.916	not rejected	0.305	not rejected
	23:00 – 24:00	70	0.864	not rejected	0.104	not rejected
0.1	00:00 – 01:00	48	0.836	not rejected	0.801	not rejected
	01:00 – 02:00	38	0.950	not rejected	0.450	not rejected
	02:00 – 05:00	67	0.504	not rejected	0.100	not rejected
	05:00 – 06:00	32	0.312	not rejected	0.524	not rejected
	06:00 – 07:00	29	0.634	not rejected	0.538	not rejected
	07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
	08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
	09:00 – 10:00	136	0.635	not rejected	0.803	not rejected
	10:00 – 12:00	297	0.433	not rejected	0.994	not rejected
	12:00 – 13:00	132	0.858	not rejected	0.948	not rejected
	13:00 – 16:00	373	0.958	not rejected	0.502	not rejected
	16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
	17:00 – 18:00	111	0.769	not rejected	0.516	not rejected
	18:00 – 22:00	418	0.660	not rejected	0.987	not rejected
	22:00 – 23:00	101	0.916	not rejected	0.305	not rejected
	23:00 – 24:00	70	0.864	not rejected	0.104	not rejected
1	00:00 – 02:00	86	0.825	not rejected	0.709	not rejected
	02:00 – 05:00	67	0.504	not rejected	0.100	not rejected
	05:00 – 07:00	61	0.739	not rejected	0.313	not rejected
	07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
	08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
	09:00 – 15:00	811	0.073	not rejected	0.955	not rejected
	15:00 – 16:00	127	0.928	not rejected	0.610	not rejected
	16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
	17:00 – 18:00	111	0.769	not rejected	0.516	not rejected
	18:00 – 24:00	589	0.059	not rejected	0.922	not rejected

10	00:00 – 02:00	86	0.825	not rejected	0.709	not rejected
	02:00 – 05:00	67	0.504	not rejected	0.100	not rejected
	05:00 – 07:00	61	0.739	not rejected	0.313	not rejected
	07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
	08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
	09:00 – 15:00	811	0.073	not rejected	0.955	not rejected
	15:00 – 16:00	127	0.928	not rejected	0.610	not rejected
	16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
	17:00 – 24:00	700	0.063	not rejected	0.720	not rejected
10^3	00:00 – 02:00	86	0.825	not rejected	0.709	not rejected
	02:00 – 06:00	99	0.451	not rejected	0.162	not rejected
	06:00 – 07:00	29	0.634	not rejected	0.538	not rejected
	07:00 – 08:00	59	0.424	not rejected	0.252	not rejected
	08:00 – 09:00	86	0.393	not rejected	0.734	not rejected
	09:00 – 15:00	811	0.073	not rejected	0.955	not rejected
	15:00 – 16:00	127	0.928	not rejected	0.610	not rejected
	16:00 – 17:00	117	0.479	not rejected	0.987	not rejected
	17:00 – 18:00	111	0.769	not rejected	0.516	not rejected
	18:00 – 22:00	418	0.660	not rejected	0.987	not rejected
	22:00 – 24:00	171	0.053	not rejected	0.681	not rejected

Table A.3. Results of the CU KS and dispersion tests (with a significance level of 0.05) applied to each interval of the partition corresponding to the starting point x^0. From top to bottom: $m = 5, 9, 17, 22, 26$. For each interval of each partition, the sample size of the dispersion test is m. H_0 denotes the null hypothesis of the corresponding test.

			CU KS test		Dispersion test	
m	Interval	k_i	p-value	H_0	p-value	H_0
5	00:00 – 01:00	20	0.167	not rejected	0.240	not rejected
	01:00 – 02:00	11	0.616	not rejected	0.151	not rejected
	02:00 – 03:00	7	0.887	not rejected	0.160	not rejected
	03:00 – 04:00	7	0.892	not rejected	0.683	not rejected
	04:00 – 05:00	12	0.217	not rejected	0.856	not rejected
	05:00 – 06:00	8	0.426	not rejected	0.219	not rejected
	06:00 – 07:00	15	0.884	not rejected	0.504	not rejected
	07:00 – 08:00	27	0.820	not rejected	0.164	not rejected
	08:00 – 09:00	35	0.875	not rejected	0.534	not rejected
	09:00 – 10:00	50	0.378	not rejected	0.844	not rejected
	10:00 – 11:00	48	0.083	not rejected	0.884	not rejected
	11:00 – 12:00	59	0.484	not rejected	0.966	not rejected
	12:00 – 13:00	51	0.594	not rejected	0.765	not rejected
	13:00 – 14:00	47	0.651	not rejected	0.689	not rejected
	14:00 – 15:00	44	0.817	not rejected	0.412	not rejected
	15:00 – 16:00	45	0.811	not rejected	0.168	not rejected
	16:00 – 17:00	47	0.679	not rejected	0.987	not rejected
	17:00 – 18:00	49	0.486	not rejected	0.534	not rejected
	18:00 – 19:00	37	0.731	not rejected	0.344	not rejected
	19:00 – 20:00	35	0.436	not rejected	0.839	not rejected
	20:00 – 21:00	44	0.904	not rejected	0.794	not rejected
	21:00 – 22:00	43	0.459	not rejected	0.693	not rejected
	22:00 – 23:00	32	0.967	not rejected	0.667	not rejected
	23:00 – 24:00	31	0.306	not rejected	0.552	not rejected
9	00:00 – 01:00	33	0.106	not rejected	0.527	not rejected
	01:00 – 02:00	22	0.658	not rejected	0.488	not rejected
	02:00 – 03:00	13	0.031	rejected	0.390	not rejected
	03:00 – 04:00	14	0.258	not rejected	0.857	not rejected
	04:00 – 05:00	16	0.441	not rejected	0.471	not rejected
	05:00 – 06:00	22	0.707	not rejected	0.335	not rejected
	06:00 – 07:00	23	0.580	not rejected	0.608	not rejected
	07:00 – 08:00	48	0.500	not rejected	0.484	not rejected
	08:00 – 09:00	54	0.338	not rejected	0.573	not rejected
	09:00 – 10:00	97	0.391	not rejected	0.886	not rejected
	10:00 – 11:00	97	0.149	not rejected	0.836	not rejected
	11:00 – 12:00	108	0.384	not rejected	0.999	not rejected
	12:00 – 13:00	95	0.911	not rejected	0.821	not rejected
	13:00 – 14:00	82	0.733	not rejected	0.923	not rejected
	14:00 – 15:00	75	0.979	not rejected	0.753	not rejected
	15:00 – 16:00	89	0.909	not rejected	0.456	not rejected
	16:00 – 17:00	82	0.429	not rejected	0.923	not rejected
	17:00 – 18:00	78	0.804	not rejected	0.596	not rejected
	18:00 – 19:00	69	0.277	not rejected	0.734	not rejected
	19:00 – 20:00	69	0.218	not rejected	0.477	not rejected
	20:00 – 21:00	72	0.731	not rejected	0.731	not rejected

	21:00 – 22:00	75	0.449	not rejected	0.541	not rejected
	22:00 – 23:00	60	0.989	not rejected	0.681	not rejected
	23:00 – 24:00	48	0.521	not rejected	0.689	not rejected
	00:00 – 01:00	73	0.729	not rejected	0.472	not rejected
	01:00 – 02:00	48	0.708	not rejected	0.291	not rejected
	02:00 – 03:00	39	0.009	rejected	0.010	rejected
	03:00 – 04:00	32	0.203	not rejected	0.622	not rejected
	04:00 – 05:00	28	0.706	not rejected	0.652	not rejected
	05:00 – 06:00	38	0.125	not rejected	0.607	not rejected
	06:00 – 07:00	35	0.908	not rejected	0.327	not rejected
	07:00 – 08:00	70	0.788	not rejected	0.075	not rejected
	08:00 – 09:00	121	0.786	not rejected	0.577	not rejected
	09:00 – 10:00	174	0.421	not rejected	0.729	not rejected
	10:00 – 11:00	186	0.332	not rejected	0.939	not rejected
	11:00 – 12:00	203	0.474	not rejected	0.999	not rejected
17	12:00 – 13:00	176	0.698	not rejected	0.986	not rejected
	13:00 – 14:00	164	0.589	not rejected	0.992	not rejected
	14:00 – 15:00	161	0.983	not rejected	0.570	not rejected
	15:00 – 16:00	168	0.506	not rejected	0.815	not rejected
	16:00 – 17:00	153	0.361	not rejected	0.996	not rejected
	17:00 – 18:00	149	0.596	not rejected	0.528	not rejected
	18:00 – 19:00	134	0.761	not rejected	0.909	not rejected
	19:00 – 20:00	140	0.101	not rejected	0.637	not rejected
	20:00 – 21:00	141	0.709	not rejected	0.760	not rejected
	21:00 – 22:00	153	0.938	not rejected	0.855	not rejected
	22:00 – 23:00	129	0.887	not rejected	0.393	not rejected
	23:00 – 24:00	94	0.950	not rejected	0.296	not rejected
	00:00 – 01:00	95	0.509	not rejected	0.720	not rejected
	01:00 – 02:00	70	0.938	not rejected	0.529	not rejected
	02:00 – 03:00	52	0.008	rejected	0.022	rejected
	03:00 – 04:00	36	0.094	not rejected	0.507	not rejected
	04:00 – 05:00	34	0.536	not rejected	0.420	not rejected
	05:00 – 06:00	46	0.045	rejected	0.703	not rejected
	06:00 – 07:00	48	0.833	not rejected	0.590	not rejected
	07:00 – 08:00	83	0.805	not rejected	0.062	not rejected
	08:00 – 09:00	165	0.576	not rejected	0.108	not rejected
	09:00 – 10:00	219	0.105	not rejected	0.737	not rejected
	10:00 – 11:00	235	0.282	not rejected	0.960	not rejected
	11:00 – 12:00	274	0.585	not rejected	0.962	not rejected
22	12:00 – 13:00	233	0.956	not rejected	0.984	not rejected
	13:00 – 14:00	216	0.515	not rejected	0.999	not rejected
	14:00 – 15:00	207	0.872	not rejected	0.789	not rejected
	15:00 – 16:00	213	0.841	not rejected	0.905	not rejected
	16:00 – 17:00	204	0.491	not rejected	0.999	not rejected
	17:00 – 18:00	192	0.534	not rejected	0.683	not rejected
	18:00 – 19:00	173	0.818	not rejected	0.968	not rejected
	19:00 – 20:00	177	0.072	not rejected	0.768	not rejected
	20:00 – 21:00	181	0.655	not rejected	0.681	not rejected
	21:00 – 22:00	196	0.977	not rejected	0.810	not rejected
	22:00 – 23:00	167	0.688	not rejected	0.412	not rejected
	23:00 – 24:00	118	0.963	not rejected	0.209	not rejected
	00:00 – 01:00	112	0.171	not rejected	0.679	not rejected

	01:00 – 02:00	75	0.933	not rejected	0.377	not rejected
	02:00 – 03:00	67	0.012	rejected	0.053	not rejected
	03:00 – 04:00	46	0.458	not rejected	0.450	not rejected
	04:00 – 05:00	38	0.987	not rejected	0.465	not rejected
	05:00 – 06:00	57	0.308	not rejected	0.535	not rejected
	06:00 – 07:00	56	0.935	not rejected	0.739	not rejected
	07:00 – 08:00	100	0.882	not rejected	0.128	not rejected
	08:00 – 09:00	198	0.566	not rejected	0.142	not rejected
	09:00 – 10:00	259	0.341	not rejected	0.844	not rejected
	10:00 – 11:00	289	0.091	not rejected	0.942	not rejected
	11:00 – 12:00	320	0.725	not rejected	0.984	not rejected
26	12:00 – 13:00	274	0.915	not rejected	0.996	not rejected
	13:00 – 14:00	257	0.228	not rejected	0.999	not rejected
	14:00 – 15:00	243	0.872	not rejected	0.835	not rejected
	15:00 – 16:00	242	0.574	not rejected	0.892	not rejected
	16:00 – 17:00	236	0.630	not rejected	0.942	not rejected
	17:00 – 18:00	231	0.808	not rejected	0.753	not rejected
	18:00 – 19:00	204	0.682	not rejected	0.980	not rejected
	19:00 – 20:00	209	0.170	not rejected	0.830	not rejected
	20:00 – 21:00	219	0.610	not rejected	0.735	not rejected
	21:00 – 22:00	237	0.803	not rejected	0.905	not rejected
	22:00 – 23:00	198	0.614	not rejected	0.366	not rejected
	23:00 – 24:00	147	0.972	not rejected	0.032	not rejected

Table A.4. Results of the CU KS and dispersion tests (with a significance level of 0.05) applied to each interval of the final (infeasible) partition obtained by solving problem 5.1.12 for different values of the parameter m, with w fixed to 1. From top to bottom: $m = 5, 9, 17, 22, 26$. For each interval of each partition, the sample size of the dispersion test is m. H_0 denotes the null hypothesis of the corresponding test.

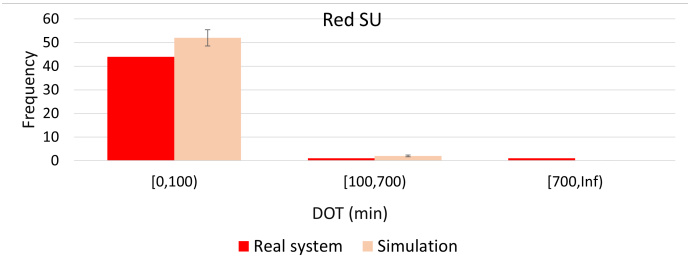
			CU KS test		Dispersion test	
m	Interval	k_i	p-value	H_0	p-value	H_0
5	00:00 – 06:00	65	0.068	not rejected	0.472	not rejected
	06:00 – 07:00	15	0.884	not rejected	0.504	not rejected
	07:00 – 08:00	27	0.820	not rejected	0.164	not rejected
	08:00 – 09:00	35	0.875	not rejected	0.534	not rejected
	09:00 – 10:00	50	0.378	not rejected	0.844	not rejected
	10:00 – 12:00	107	0.734	not rejected	0.938	not rejected
	12:00 – 13:00	51	0.594	not rejected	0.765	not rejected
	13:00 – 14:00	47	0.651	not rejected	0.689	not rejected
	14:00 – 15:00	44	0.817	not rejected	0.412	not rejected
	15:00 – 24:00	363	0.214	not rejected	0.568	not rejected
9	00:00 – 02:00	55	0.249	not rejected	0.607	not rejected
	02:00 – 04:00	27	0.309	not rejected	0.501	not rejected
	04:00 – 05:00	16	0.441	not rejected	0.471	not rejected
	05:00 – 06:00	22	0.707	not rejected	0.335	not rejected
	06:00 – 07:00	23	0.580	not rejected	0.608	not rejected
	07:00 – 08:00	48	0.500	not rejected	0.484	not rejected
	08:00 – 09:00	54	0.338	not rejected	0.573	not rejected
	09:00 – 16:00	643	0.060	not rejected	0.717	not rejected
	16:00 – 17:00	82	0.429	not rejected	0.923	not rejected
	17:00 – 18:00	78	0.804	not rejected	0.596	not rejected
	18:00 – 22:00	285	0.919	not rejected	0.989	not rejected
	22:00 – 23:00	60	0.989	not rejected	0.681	not rejected
	23:00 – 24:00	48	0.522	not rejected	0.689	not rejected
17	00:00 – 02:00	121	0.094	not rejected	0.535	not rejected
	02:00 – 05:00	99	0.098	not rejected	0.067	not rejected
	05:00 – 07:00	73	0.650	not rejected	0.203	not rejected
	07:00 – 08:00	70	0.788	not rejected	0.075	not rejected
	08:00 – 09:00	121	0.786	not rejected	0.577	not rejected
	09:00 – 10:00	174	0.421	not rejected	0.729	not rejected
	10:00 – 14:00	729	0.089	not rejected	0.995	not rejected
	14:00 – 16:00	329	0.982	not rejected	0.410	not rejected
	16:00 – 17:00	153	0.361	not rejected	0.996	not rejected
	17:00 – 18:00	149	0.596	not rejected	0.528	not rejected
	18:00 – 22:00	568	0.586	not rejected	0.926	not rejected
	22:00 – 24:00	223	0.071	not rejected	0.793	not rejected
22	00:00 – 02:00	165	0.198	not rejected	0.743	not rejected
	02:00 – 06:00	168	0.117	not rejected	0.122	not rejected
	06:00 – 07:00	48	0.833	not rejected	0.590	not rejected
	07:00 – 08:00	83	0.805	not rejected	0.062	not rejected
	08:00 – 09:00	165	0.576	not rejected	0.108	not rejected
	09:00 – 10:00	219	0.105	not rejected	0.737	not rejected
	10:00 – 14:00	958	0.097	not rejected	0.994	not rejected
	14:00 – 16:00	420	0.952	not rejected	0.561	not rejected
	16:00 – 17:00	204	0.491	not rejected	0.999	not rejected

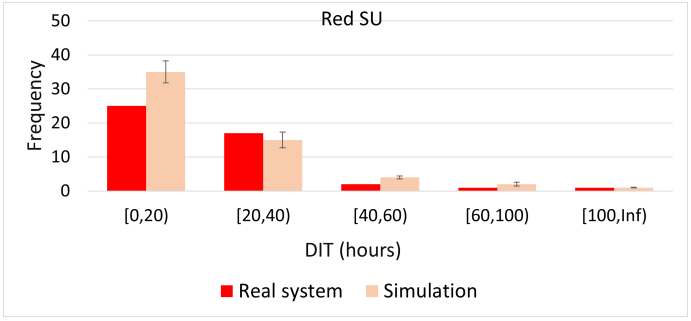
	17:00 – 18:00	192	0.534	not rejected	0.683	not rejected
	18:00 – 22:00	772	0.436	not rejected	0.968	not rejected
	22:00 – 23:00	167	0.688	not rejected	0.412	not rejected
	23:00 – 24:00	118	0.963	not rejected	0.209	not rejected
26	00:00 – 01:00	112	0.171	not rejected	0.679	not rejected
	01:00 – 02:00	75	0.933	not rejected	0.378	not rejected
	02:00 – 06:00	208	0.072	not rejected	0.080	not rejected
	06:00 – 07:00	56	0.935	not rejected	0.739	not rejected
	07:00 – 08:00	100	0.882	not rejected	0.128	not rejected
	08:00 – 09:00	198	0.566	not rejected	0.142	not rejected
	09:00 – 10:00	259	0.341	not rejected	0.844	not rejected
	10:00 – 11:00	289	0.091	not rejected	0.942	not rejected
	11:00 – 12:00	320	0.725	not rejected	0.984	not rejected
	12:00 – 13:00	274	0.915	not rejected	0.996	not rejected
	13:00 – 15:00	500	0.439	not rejected	0.971	not rejected
	15:00 – 16:00	242	0.574	not rejected	0.892	not rejected
	16:00 – 18:00	467	0.895	not rejected	0.939	not rejected
	18:00 – 21:00	632	0.643	not rejected	0.950	not rejected
	21:00 – 22:00	237	0.803	not rejected	0.905	not rejected
	22:00 – 24:00	345	0.034	rejected	0.440	not rejected

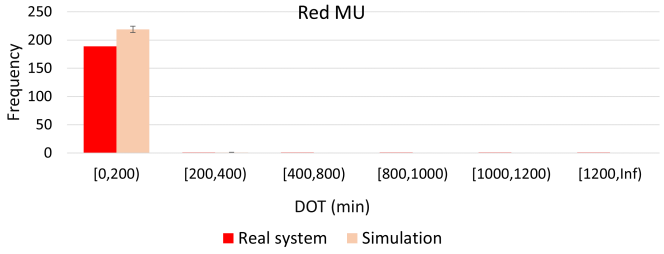
Appendix B

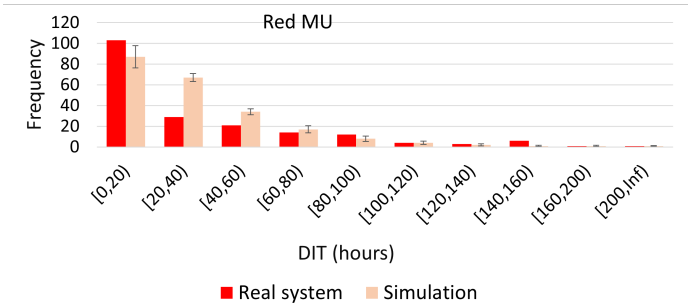
Model calibration - Plots I

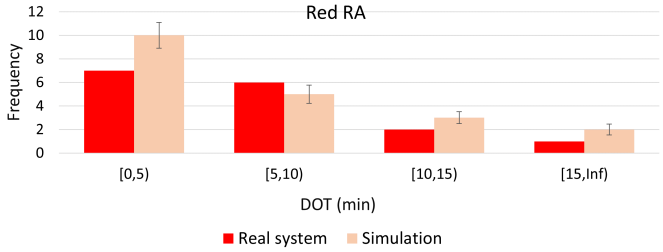
This appendix reports the comparison between the histograms of the time differences *DOT* and *DIT* computed through the real data and the simulation output, the latter reporting the 95% confidence interval. The comparisons are performed for all patients with color tag $c \in C$ assigned to unit $u \in U(c)$.

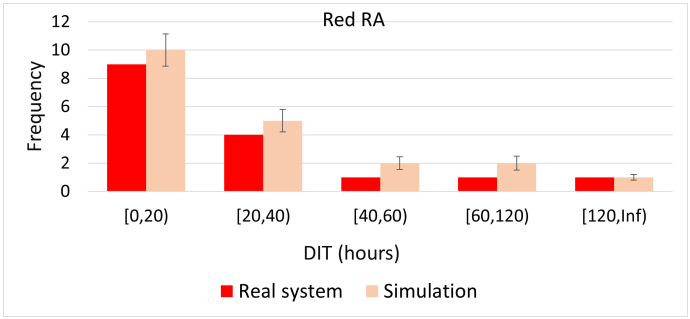


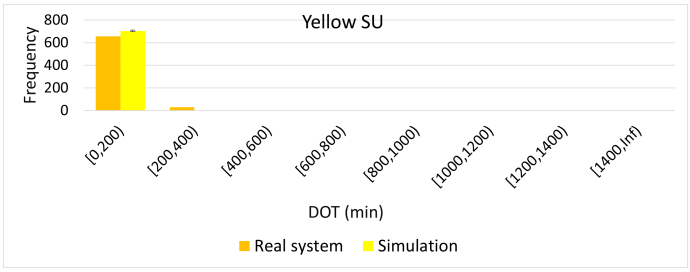


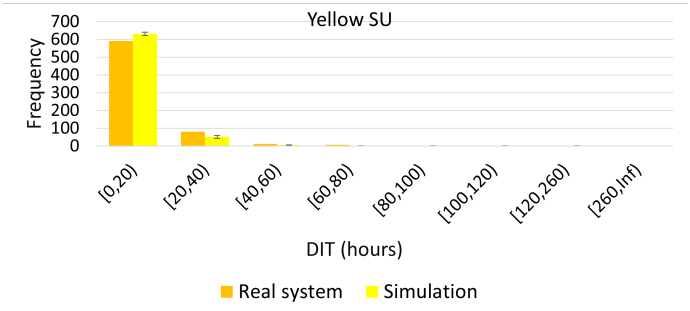


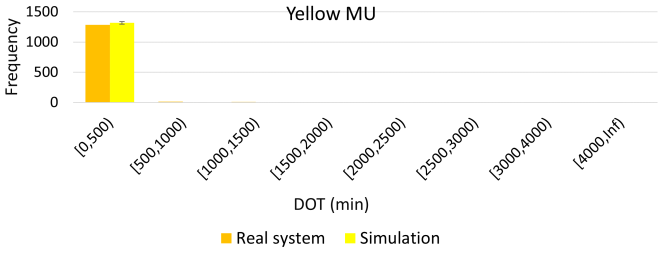

Red MU
Frequency
120
100
80
60
40
20
0
[0,20)
[20,40)
[40,60)
[60,80)
[80,100)
[100,120)
[120,140)
[140,160)
[160,200)
[200,Inf)
DIT (hours)
Real system
Simulation

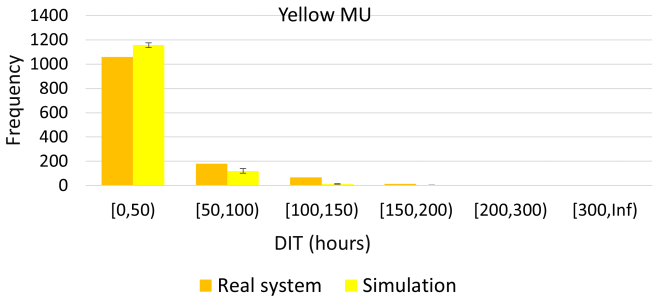

Red RA
Frequency
12
10
8
6
4
2
0
[0,5)
[5,10)
[10,15)
[15,Inf)
DOT (min)
Real system
Simulation

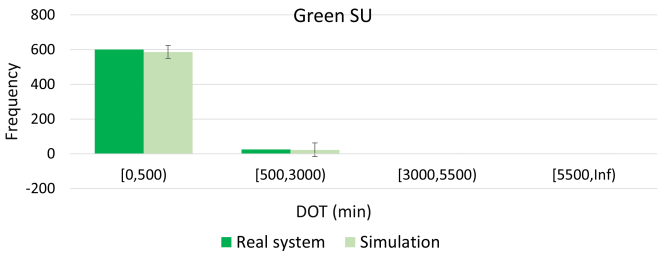

Red RA
Frequency
12
10
8
6
4
2
0
[0,20)
[20,40)
[40,60)
[60,120)
[120,Inf)
DIT (hours)
Real system
Simulation

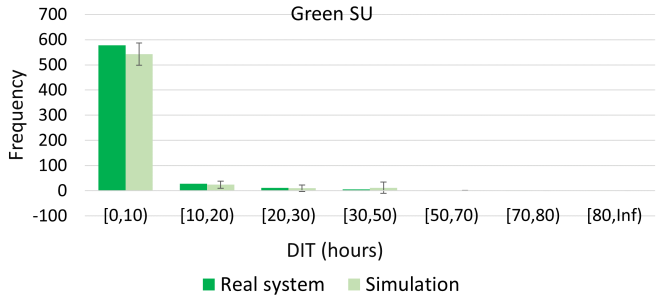

Yellow SU
Frequency
800
600
400
200
0
[0,200)
[200,400)
[400,600)
[600,800)
[800,1000)
[1000,1200)
[1200,1400)
[1400,Inf)
DOT (min)
Real system
Simulation

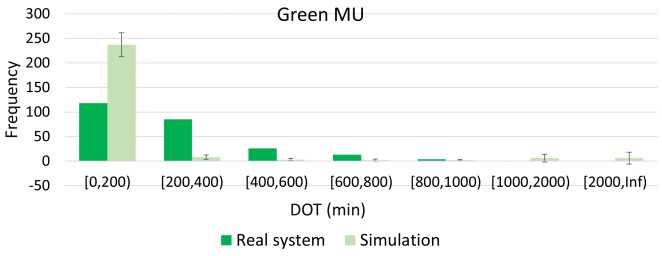

Yellow SU
Frequency
700
600
500
400
300
200
100
0
[0,20)
[20,40)
[40,60)
[60,80)
[80,100)
[100,120)
[120,260)
[260,Inf)
DIT (hours)
Real system
Simulation

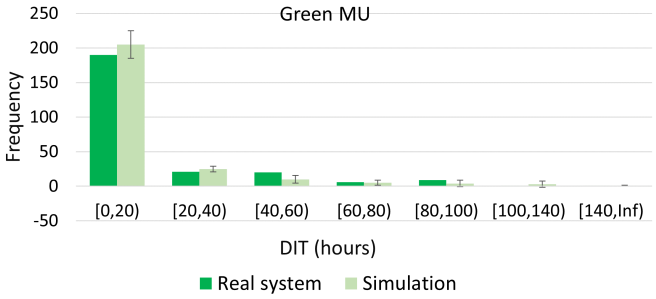

Yellow MU
Frequency
1500
1000
500
0
[0,500)
[500,1000)
[1000,1500)
[1500,2000)
[2000,2500)
[2500,3000)
[3000,4000)
[4000,Inf)
DOT (min)
Real system
Simulation

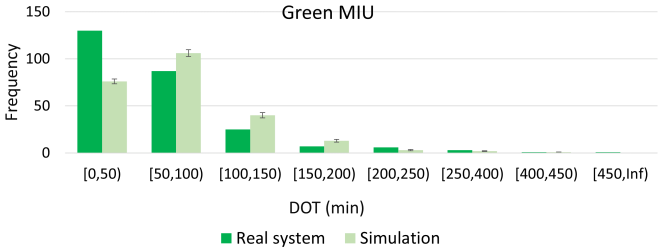

Yellow MU
Frequency
1400
1200
1000
800
600
400
200
0
[0,50)
[50,100)
[100,150)
[150,200)
[200,300)
[300,Inf)
DIT (hours)
Real system
Simulation

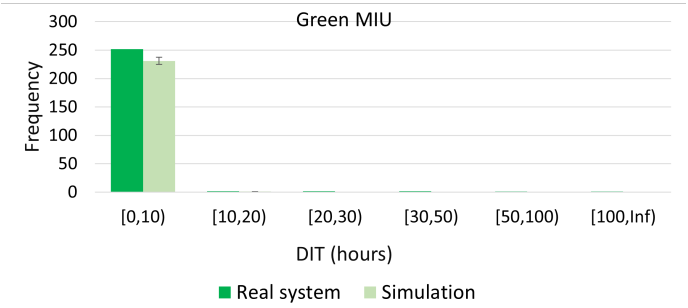

Green SU
Frequency
800
600
400
200
0
-200
[0,500)
[500,3000)
[3000,5500)
[5500,Inf)
DOT (min)
Real system
Simulation

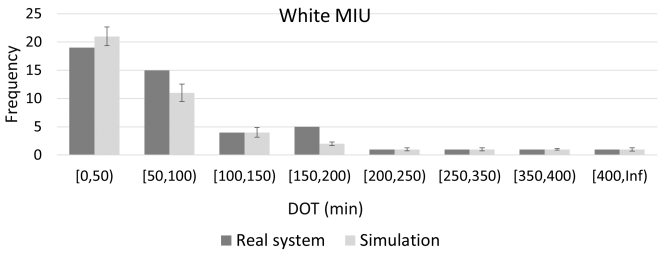

Green SU
Frequency
700
600
500
400
300
200
100
0
-100
[0,10)
[10,20)
[20,30)
[30,50)
[50,70)
[70,80)
[80,Inf)
DIT (hours)
Real system
Simulation

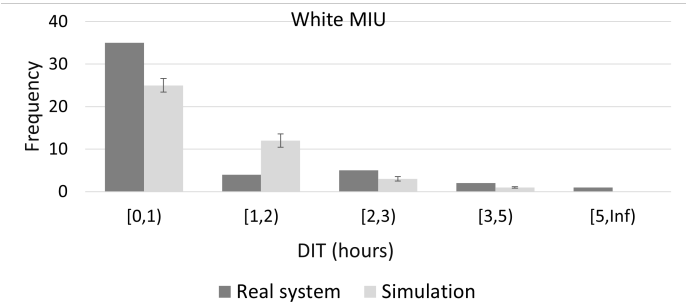

Green MU
Frequency
300
250
200
150
100
50
0
-50
[0,200)
[200,400)
[400,600)
[600,800)
[800,1000)
[1000,2000)
[2000,Inf)
DOT (min)
Real system
Simulation


Green MU
250
200
150
100
50
0
-50
Frequency
[0,20)
[20,40)
[40,60)
[60,80)
[80,100)
[100,140)
[140,Inf)
DIT (hours)
Real system
Simulation


Green MIU
150
100
50
0
Frequency
[0,50)
[50,100)
[100,150)
[150,200)
[200,250)
[250,400)
[400,450)
[450,Inf)
DOT (min)
Real system
Simulation


Green MIU
300
250
200
150
100
50
0
Frequency
[0,10)
[10,20)
[20,30)
[30,50)
[50,100)
[100,Inf)
DIT (hours)
Real system
Simulation

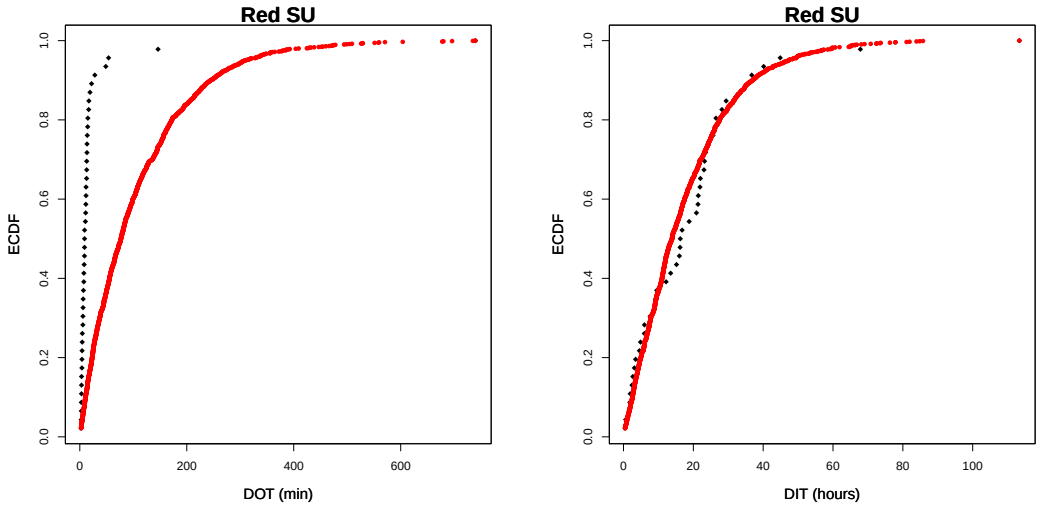

White MIU
25
20
15
10
5
0
Frequency
[0,50)
[50,100)
[100,150)
[150,200)
[200,250)
[250,350)
[350,400)
[400,Inf)
DOT (min)
Real system
Simulation

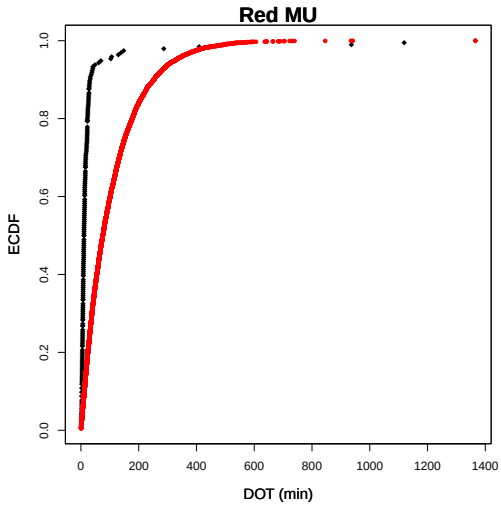

White MIU
40
30
20
10
0
Frequency
[0,1)
[1,2)
[2,3)
[3,5)
[5,Inf)
DIT (hours)
Real system
Simulation

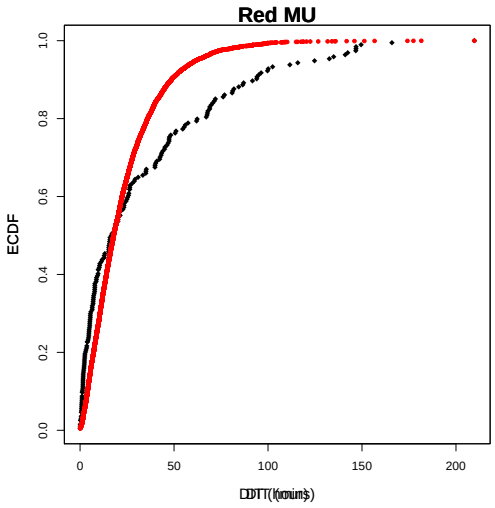
Appendix C

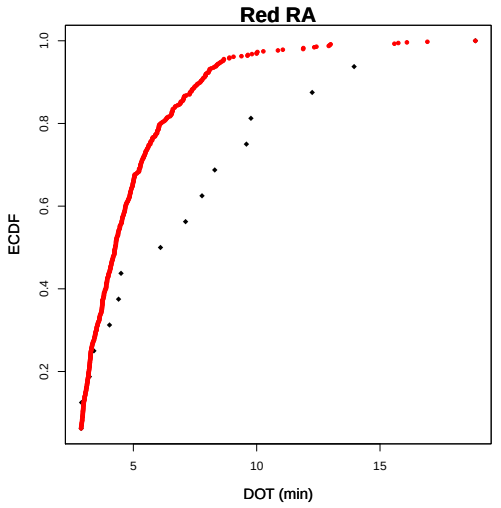
Model calibration - Plots II

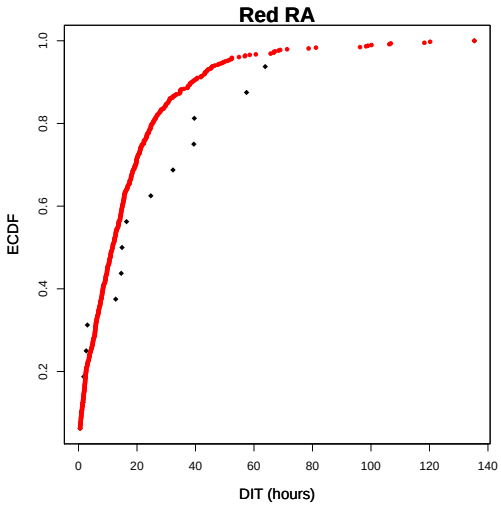
This appendix reports the comparison between the Empirical Cumulative Distribution Functions (ECDFs) of the time differences DOT and DIT computed through the real data and the simulation output resulting from the calibration procedure. The comparisons are performed for all patients with color tag $c \in C$ assigned to unit $u \in U(c)$. The colored curves correspond to the simulation output.

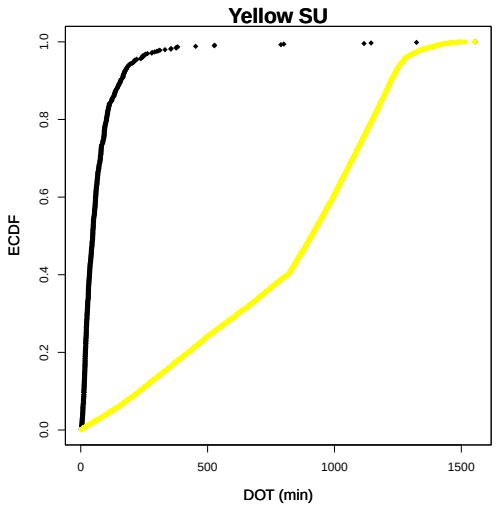


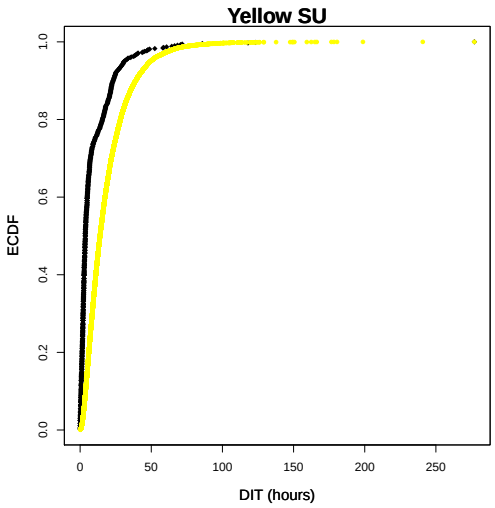

Red MU
ECDF
DOT (min)

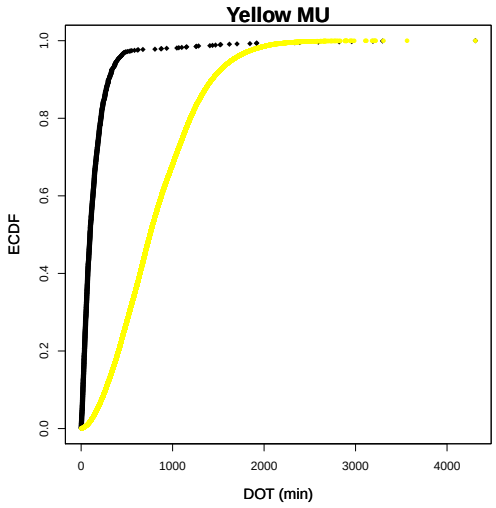

Red MU
ECDF

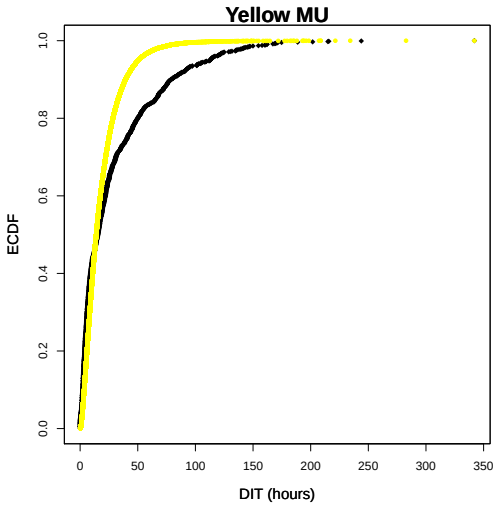

Red RA
ECDF
DOT (min)

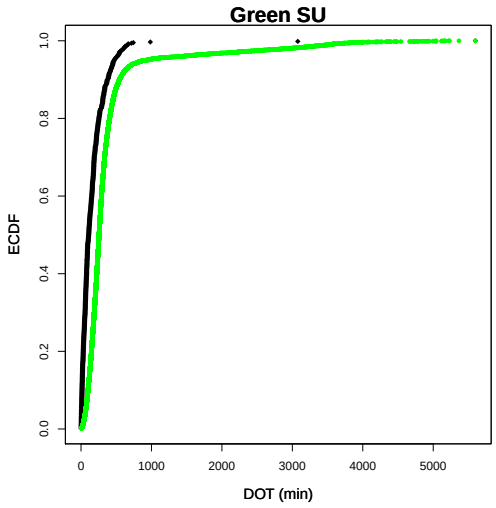

Red RA
ECDF
DIT (hours)

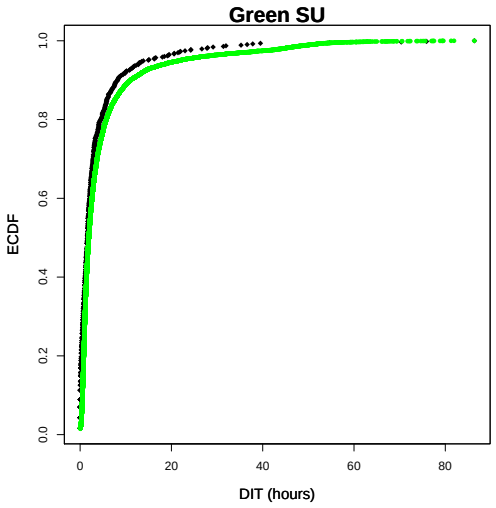

Yellow SU
ECDF
DOT (min)

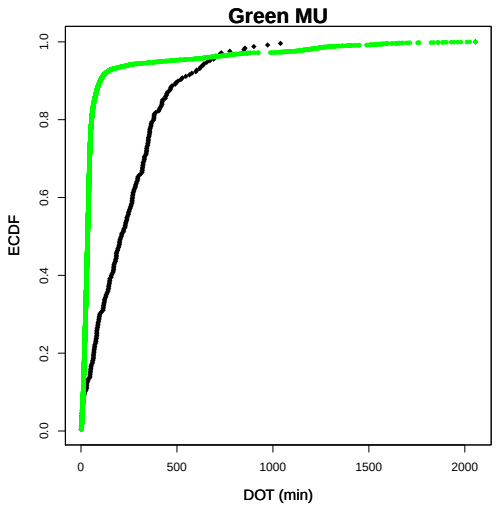

Yellow SU
ECDF
DIT (hours)

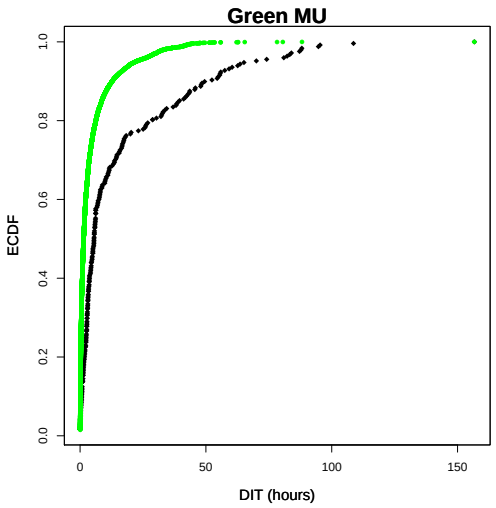

Yellow MU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
DOT (min)

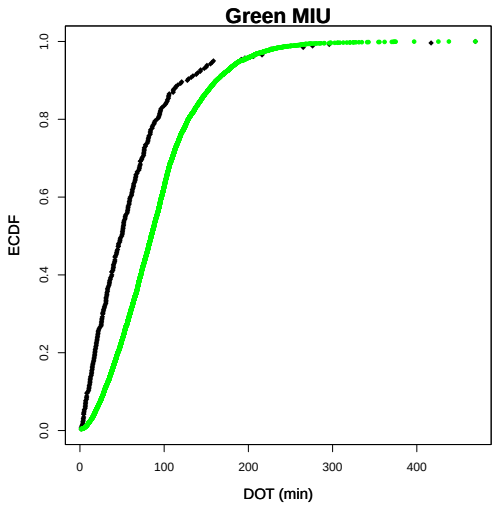

Yellow MU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
250
300
350
DIT (hours)

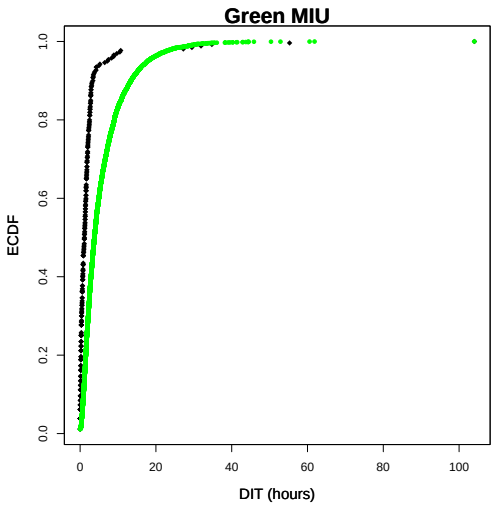

Green SU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
1000
2000
3000
4000
5000
DOT (min)

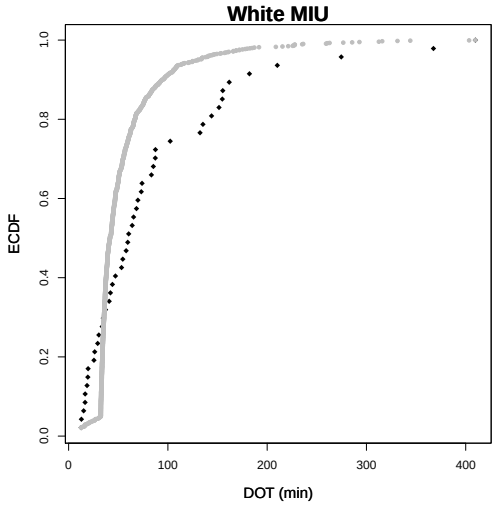

Green SU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
DIT (hours)

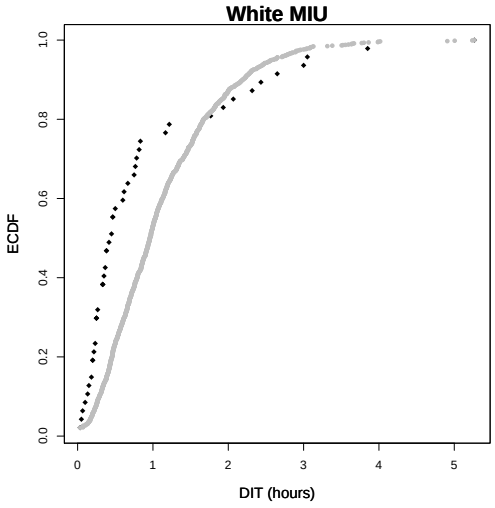

Green MU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
DOT (min)


Green MU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
DIT (hours)


Green MIU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
DOT (min)


Green MIU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
100
DIT (hours)


White MIU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
DOT (min)


White MIU
ECDF
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
DIT (hours)

Bibliography

[1] Aboueljinane, L., Sahin, E., Jemai, Z.: A review on simulation models applied to emergency medical service operations. Computers & Industrial Engineering **66**(4), 734 – 750 (2013)

[2] Abramson, M., Audet, C.: Filter pattern search algorithms for mixed variable constrained optimization problems. Pacific Journal of Optimization **3** (2004)

[3] Abramson, M., Audet, C., Chrissis, J., Walston, J.: Mesh adaptive direct search algorithms for mixed variable optimization. Optimization Letters **3**, 35–47 (2009). DOI 10.1007/s11590-008-0089-2

[4] Abramson, M., Audet, C., Couture, G., Dennis, Jr., J., Le Digabel, S., Tribes, C.: The NOMAD project. Software available at https://www.gerad.ca/nomad/. URL https://www.gerad.ca/nomad/

[5] Abramson, M., Audet, C., Dennis, J., Le Digabel, S.: Orthomads: a deterministic mads instance with orthogonal directions. SIAM Journal on Optimization **20**, 948–966 (2009). DOI 10.1137/080716980

[6] Ahalt, V., Argon, N., Strickler, J., Mehrotra, A.: Comparison of emergency department crowding scores: a discrete-event simulation approach. Health Care Management Science **21**, 144–155 (2018)

[7] Ahmed, M.A., Alkhamis, T.M.: Simulation optimization for an emergency department healthcare unit in Kuwait. European Journal of Operational Research **198**(3), 936 – 942 (2009)

[8] Ahsan, K., Alam, M., Morel, D., Karim, M.: Emergency department resource optimisation for improved performance: a review. Journal of Industrial Engineering International **15 (Supp 1)**, S253–S266 (2019)

[9] Al-Kattan, I., Abboud, B.: Disaster recovery plan development for the emergency department – case study. Public Administration & Management **13**(3), 75–99 (2009)

[10] Almagooshi, S.: Simulation modelling in healthcare: Challenges and trends. Procedia Manufacturing **3**, 301–307 (2015)

[11] Amaran, S., Sahinidis, N., Sharda, B., Bury, S.: Simulation optimization: a review of algorithms and applications. 4OR **12**, 301–333 (2014)

[12] Andradóttir, S.: Chapter 20 an overview of simulation optimization via random search. In: S.G. Henderson, B.L. Nelson (eds.) Simulation, *Handbooks in Operations Research and Management Science*, vol. 13, pp. 617 – 631. Elsevier (2006). DOI https://doi.org/10.1016/S0927-0507(06)13020-0

[13] Aringhieri, R., Bonetta, G., Duma, D.: Reducing overcrowding at the emergency department through a different physician and nurse shift organisation: a case study. In: P. Daniele, L. Scrimali (eds.) New Trends in Emergency Complex Real Life Problems, *AIRO Springer Series*, vol. 1, pp. 43–53. Springer Nature Switzerland (2018)

[14] Aringhieri, R., Bruni, M., Khodaparasti, S., van Essen, J.: Emergency medical services and beyond: Addressing new challenges through a wide literature review. Computers & Operations Research **78**, 349 – 368 (2017)

[15] Aroua, A., Abdulnour, G.: Optimization of the emergency department in hospitals using simulation and experimental design: Case study. 2017 Winter Simulation Conference (WSC) pp. 4511–4513 (2017)

[16] Audet, C., Béchard, V., Le Digabel, S.: Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. J. Global Optimization **41**, 299–318 (2008). DOI 10.1007/s10898-007-9234-1

[17] Audet, C., Custódio, A., Dennis, J.: Erratum: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization **18**, 1501–1503 (2008)

[18] Audet, C., Dennis, J.: Pattern search algorithms for mixed variable programming. SIAM Journal on Optimization **11** (2001). DOI 10.1137/S1052623499352024

[19] Audet, C., Dennis, J.: A pattern search filter method for nonlinear programming without derivatives. SIAM Journal on Optimization **14** (2004). DOI 10.1137/S105262340138983X

[20] Audet, C., Dennis, J.: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization **17**, 188–217 (2006)

[21] Audet, C., Dennis, J., Le Digabel, S.: Parallel space decomposition of the mesh adaptive direct search algorithm. SIAM Journal on Optimization **19**, 1150–1170 (2008). DOI 10.1137/070707518

[22] Audet, C., Dennis, J., Le Digabel, S.: Globalization strategies for mesh adaptive direct search. Computational Optimization and Applications **46**, 193–215 (2010). DOI 10.1007/s10589-009-9266-1

[23] Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer International Publishing (2017). DOI 10.1007/978-3-319-68913-5

[24] Audet, C., Le Digabel, S., Tribes, C.: NOMAD user guide. Tech. Rep. G-2009-37, Les cahiers du GERAD (2009). URL https://www.gerad.ca/nomad/Downloads/user_guide.pdf

[25] Audet, C., Le Digabel, S., Tribes, C.: The mesh adaptive direct search algorithm for granular and discrete variables. SIAM Journal on Optimization **29**, 1164–1189 (2019). DOI 10.1137/18M1175872

[26] Azaron, A., Katagiri, H., Sakawa, M., Kato, K., Memariani, A.: A multi-objective resource allocation problem in pert networks. European Journal of Operational Research **172**(3), 838 – 854 (2006)

[27] Azaron, A., Tavakkoli-Moghaddam, R.: Multi-objective time–cost trade-off in dynamic pert networks using an interactive approach. European Journal of Operational Research **180**(3), 1186 – 1200 (2007)

[28] Baker, B.M.: Cost/time trade-off analysis for the critical path method: a derivation of the network flow approach. Journal of the Operational Research Society **48**(12), 1241–1244 (1997)

[29] Ballarini, P., Duma, D., Horváth, A., Aringhieri, R.: Petri nets validation of markovian models of emergency department arrivals. In: R. Janicki, N. Sidorova, T. Chatain (eds.) Application and Theory of Petri Nets and Concurrency, pp. 219–238. Springer International Publishing, Cham (2020)

[30] Barton, R., Meckesheimer, M.: Chapter 18 metamodel-based simulation optimization. Handbooks in Operations Research and Management Science **13**(C), 535–574 (2006). DOI 10.1016/S0927-0507(06)13018-2

[31] Batt, R., Terwiesch, C.: Early task initiation and other load-adaptive mechanisms in the emergency department. Management Science **63**(11), 3531–3551 (2017)

[32] Baumlin, K., Shapiro, J., Weiner, C., Gottlieb, B., Chawla, N., Richardson, L.: Clinical information system and process redesign improves emergency department efficiency. The Joint Commission Journal on Quality and Patient Safety **36**(4), 179–185 (2010)

[33] Bechhofer, R.E., Santner, T.J., Goldsman, D.M.: Design and analysis of experiments for statistical selection, screening, and multiple comparisons. John Wiley and Sons, New York, NY, USA (1995)

[34] Bedoya-Valencia, L., Kirac, E.: Evaluating alternative resource allocation in an emergency department using discrete event simulation. Simulation **92**, 1041–1051 (2016)

[35] Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statistical Science **8**(1), 10–15 (1993)

[36] Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning (2018)

[37] Boukouvala, F., Misener, R., Floudas, C.: Global optimization advances in mixed-integer nonlinear programming, minlp, and constrained derivative-free optimization, cdfo. European Journal of Operational Research **252** (2015). DOI 10.1016/j.ejor.2015.12.018

[38] Allen Bradley, R.S.: Arena User's guide. Rockwell Automation (2010)

[39] Bratley, P., Fox, B.L.: Algorithm 659: Implementing sobol's quasirandom sequence generator. ACM Trans. Math. Softw. **14**(1), 88–100 (1988). DOI 10.1145/42288.214372. URL https://doi.org/10.1145/42288.214372

[40] Bregman, R.L.: A heuristic procedure for solving the dynamic probabilistic project expediting problem. European Journal of Operational Research **192**(1), 125 – 137 (2009)

[41] Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Shen, H., Zeltyn, S., Zhao, L.: Statistical analysis of a telephone call center: A queueing-science perspective. Journal of the American Statistical Association **100**, 36–50 (2005). DOI 10.2307/27590517

[42] Cao, H., Huang, S.: Principles of scarce medical resource allocation in natural disaster relief: A simulation approach. Medical Decision Making **32**(3), 470–476 (2012)

[43] Chanchaichujit, J., Tan, A., Meng, F., Eaimkhong, S.: Optimization, Simulation and Predictive Analytics in Healthcare, pp. 95–121. Palgrave Pivot (2019)

[44] Chang, K.H., Hong, L.J., Wan, H.: Stochastic trust-region response-surface method strong—a new response-surface framework for simulation optimization. INFORMS J. on Computing **25**(2), 230–243 (2013). DOI 10.1287/ijoc.1120.0498. URL https://doi.org/10.1287/ijoc.1120.0498

[45] Chen, C.H., Yücesan, E., Dai, L., Chen, H.C.: Optimal budget allocation for discrete-event simulation experiments. IIE Transactions **42**(1), 60–70 (2009). DOI 10.1080/07408170903116360

[46] Chen, H., Schmeiser, B.W.: Retrospective approximation algorithms for stochastic root finding. In: Proceedings of the 26th Conference on Winter Simulation, WSC '94, p. 255–261. Society for Computer Simulation International, San Diego, CA, USA (1994)

[47] Chun-Hung Chen: An effective approach to smartly allocate computing budget for discrete event simulation. In: Proceedings of 1995 34th IEEE Conference on Decision and Control, vol. 3, pp. 2598–2603 vol.3 (1995). DOI 10.1109/CDC.1995.478499

[48] Clarey, A., Cooke, M.: Patients who leave emergency departments without being seen: literature review and English data analysis. Emergency Medicine Journal **29**(8), 617–621 (2011)

[49] Clarke, F.: Optimization and Nonsmooth Analysis. Wiley New York (1983)

[50] Cohen, I., Golany, B., Shtub, A.: The stochastic time–cost tradeoff problem: A robust optimization approach. Networks **49**(2), 175–188 (2007)

[51] Conn, A., Scheinber, K., Vicente, L.: Introduction to Derivative-Free Optimization. MPS-SIAM Book Series on Optimization, SIAM, Philadelphia (2009)

[52] Copeland, J., Gray, A.: A daytime fast track improves throughput in a single physician coverage emergency department. CJEM **17 6**, 648–55 (2015)

[53] Costa, A., Nannicini, G.: RBFOpt: an open-source library for black-box optimization with costly function evaluations. Mathematical Programming Computation **10** (2018). DOI 10.1007/s12532-018-0144-7

[54] Costa, L., Oliveira, P.: Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. Computers & Chemical Engineering **25**(2), 257 – 266 (2001). DOI https://doi.org/10.1016/S0098-1354(00)00653-0

[55] Custódio, A., Vicente, L.: Using sampling and simplex derivatives in pattern search methods. SIAM Journal on Optimization **18**, 537–555 (2007). DOI 10.1137/050646706

[56] Custódio, A.L., Dennis J. E., J., Vicente, L.N.: Using simplex gradients of nonsmooth functions in direct search methods. IMA Journal of Numerical Analysis **28**(4), 770–784 (2008). DOI 10.1093/imanum/drn045

[57] Daldoul, D., Nouaouri, I., Bouchriha, H., Allaoui, H.: A stochastic model to minimize patient waiting time in an emergency department. Operations Research for Health Care **18**, 16 – 25 (2018). EURO 2016–New Advances in Health Care Applications

[58] Davis, L. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold (1991)

[59] De Santis, A., Giovannelli, T., Lucidi, S., Messedaglia, M., Roma, M.: An optimal non–uniform piecewise constant approximation for the patient arrival rate for a more efficient representation of the emergency departments arrival process. Technical Report 1–2020, Dipartimento di Ingegneria Informatica Automatica e Gestionale "A. Ruberti", SAPIENZA Università di Roma (2020)

[60] Deckro, R., Hebert, J., Verdini, W.: Project scheduling with work packages. Omega **20**(2), 169 – 182 (1992)

[61] Deep, K., Singh, K.P., Kansal, M., Mohan, C.: A real coded genetic algorithm for solving integer and mixed integer optimization problems. Applied Mathematics and Computation **212**(2), 505 – 518 (2009). DOI https://doi.org/10.1016/j.amc.2009.02.044

[62] Demeulemeester, E.L., Herroelen, W.: Project Scheduling : a Research Handbook. Kluwer Academic Publishers, Boston, USA (2002)

[63] Deng, G.: Simulation-based optimization. University of Wisconsin–Madison (2007)

[64] Deng, G., Ferris, M.C.: Adaptation of the uobyqa algorithm for noisy functions. In: Proceedings of the 2006 Winter Simulation Conference, pp. 312–319 (2006). DOI 10.1109/WSC.2006.323088

[65] Diefenbach, M., Kozan, E.: Effects of bed configurations at a hospital emergency department. Journal of Simulation **5**(1), 44–57 (2011)

[66] Digabel, S.L., Wild, S.M.: A taxonomy of constraints in simulation-based optimization (2015)

[67] Diniz-Ehrhardt, M., Martínez, J., Raydan, M.: A derivative-free nonmonotone line-search technique for unconstrained optimization. Journal of Computational and Applied Mathematics **219**(2), 383 – 397 (2008). DOI https://doi.org/10.1016/j.cam.2007.07.017. Special Issue dedicated to Prof. Claude Brezinski, on the occasion of his retirement from the University of Sciences and Technologies of Lille in June 2006

[68] Diwas Singh, K.: Does multitasking improve performance? Evidence from the emergency department. Manufacturing & Service Operations Manamgement **16**(2), 168–183 (2014)

[69] Dominguez, O., Juan, A.A., Barrios, B., Faulin, J., Agustin, A.: Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. Annals of Operations Research **236**(2), 383–404 (2016)

[70] Dominguez, O., Juan, A.A., Faulin, J.: A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. International Transactions in Operational Research **21**(3), 375–398 (2014)

[71] Duma, D., Aringhieri, R.: An ad hoc process mining approach to discover patient paths of an emergency department. Flexible Services and Manufacturing Journal **32** (2018). DOI 10.1007/s10696-018-9330-1

[72] Durbin, J.: Some methods for constructing exact tests. Biometrika **48**, 41–55 (1961)

[73] Elmaghraby, S.: Activity Networks: Project Planning and Control by Network Models. A Wiley-Interscience publication. Wiley (1977)

[74] Elmaghraby, S., Morgan, C.: Resource Allocation in Activity Networks under Stochastic Conditions. A Geometric Programming-Sample Path Optimization Approach. Review of Business and Economic Literature **0**(3), 367–390 (2007)

[75] Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for nonsmooth constrained optimization. SIAM Journal on Optimization **24**, 959–992 (2014). DOI 10.1137/130940037

[76] Feo, T.A., Resende, M.G.: Greedy randomized adaptive search procedures. Journal of global optimization **6**(2), 109–133 (1995)

[77] Ferone, D., Festa, P., Gruler, A., Juan, A.A.: Combining simulation with a grasp metaheuristic for solving the permutation flow-shop problem with stochastic processing times. In: T.M.K. Roeder et al. (ed.) Proceedings of the 2016 Winter Simulation Conference, pp. 2205–2215. IEEE Press, Piscataway, New Jersey (2016)

[78] Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP - Part I: Algorithms. International Transactions in Operational Research **16**(1), 1–24 (2009)

[79] Festa, P., Resende, M.G.C.: An annotated bibliography of GRASP - Part II: Applications. International Transactions in Operational Research **16**(2), 131–172 (2009)

[80] Fiechter, C.N.: A parallel tabu search algorithm for large traveling salesman problems. Discrete Applied Mathematics **51**(3), 243 – 267 (1994). DOI https://doi.org/10.1016/0166-218X(92)00033-I

[81] Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. math. program. ser. a 91, 239-269. Mathematical Programming **91** (2002). DOI 10.1007/s101070100244

[82] Fu, M.C. (ed.): Handbook of Simulation Optimization. Springer, New York (2015)

[83] Garcìa-Palomares, U.M.: A unified convergence theory for non monotone direct search methods (dsms) with extensions to dfo with mixed and categorical variables (2019). DOI 10.13140/RG.2.2.29624.49929

[84] Garcìa-Palomares, U.M.: Non-monotone derivative-free algorithm for solving optimization models with linear constraints: extensions for solving nonlinearly constrained models via exact penalty methods. TOP (2020). DOI 10.1007/s11750-020-00549-y

[85] Garcìa-Palomares, U.M., Costa-Montenegro, E., Asorey Cacheda, R., González-Castaño, F.: Adapting derivative free optimization methods to engineering models with discrete variables. Optimization and Engineering **13** (2012). DOI 10.1007/s11081-011-9168-9

[86] Garcìa-Palomares, U.M., Rodriguez, J.F.: New sequential and parallel derivative-free algorithms for unconstrained minimization. SIAM Journal on Optimization **13**, 79–96 (2002). DOI 10.1137/S1052623400370606

[87] Gendreau, M., Potvin, J.Y.: Handbook of Metaheuristics, 2nd edn. Springer Publishing Company, Incorporated (2010)

[88] Gilboy, N., Tanabe, T., Travers, D., Rosenau, A.: Emergency Severity Index (ESI): A Triage Tool for Emergency Department Care, Version 4, Implementation Handbook 2012 edn. Agency for Healthcare Research and Quality, Rockville, MD (2011). AHRQ Publication No. 12-0014

[89] Glover, F.: Tabu search: A tutorial. Interfaces **20**, 74–94 (1990). DOI 10.1287/inte.20.4.74

[90] Glover, F., Laguna, M., Marti, R.: Fundamentals of scatter search and path relinking. Control and Cybernetics **29** (2000)

[91] Godinho, P., Branco, F.G.: Adaptive policies for multi-mode project scheduling under uncertainty. European Journal of Operational Research **216**(3), 553 – 562 (2012)

[92] Goh, J., Hall, N.G.: Total cost control in project management via satisficing. Management Science **59**(6), 1354–1372 (2013)

[93] Gonzalez-Neira, E.M., Ferone, D., Hatami, S., Juan, A.A.: A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. Simulation Modelling Practice and Theory **79**, 23–36 (2017)

[94] Gosavi, A.: Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning, 2nd edn. Springer Publishing Company, Incorporated (2014)

[95] Granja, C., Almada-Lobo, B., Janela, F., Seabra, J., Mendes, A.: An optimization based on simulation approach to the patient admission scheduling problem using a linear programing algorithm. Journal of Biomedical Informatics **52**, 427–437 (2014)

[96] Grasas, A., Juan, A.A., Faulin, J., de Armas, J., Ramalhinho, H.: Biased randomization of heuristics using skewed probability distributions: a survey and some applications. Computers & Industrial Engineering **110**, 216—228 (2017)

[97] Grasas, A., Juan, A.A., Lourenço, H.R.: SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. Journal of Simulation **10**(1), 69–77 (2016)

[98] Gray, G.A., Kolda, T.G.: Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. ACM Trans. Math. Softw. **32**(3), 485–507 (2006). DOI 10.1145/1163641.1163647. URL https://doi.org/10.1145/1163641.1163647

[99] Griffin, J., Fowler, K., Gray, G., Hemker, T., Parno, M.: Derivative-free optimization via evolutionary algorithms guiding local search. Pacific Journal of Optimization **7** (2011)

[100] Grippo, L., Lampariello, F., Lucidi, S.: Global convergence and stabilization of unconstrained minimization methods without derivatives. Journal of Optimization Theory and Applications **56** (1988). DOI 10.1007/BF00939550

[101] Grippo, L., Rinaldi, F.: A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations. Computational Optimization and Applications **60**, 1–33 (2014). DOI 10.1007/s10589-014-9665-9

[102] Gul, M., Guneri, A.: Simulation modelling of a patient surge in an emergency department under disaster conditions. Croatian Operational Research Review **6**, 429–443 (2015)

[103] Gul, M., Guneri, A., Gunal, M.: Emergency department network under disaster conditions: The case of possible major Istanbul earthquake. Journal of the Operational Research Society **71**(5), 733–747 (2020)

[104] Gul, M., Guneri, A.F.: A comprehensive review of emergency department simulation applications for normal and disaster conditions. Comput. Ind. Eng. **83**(C), 327–344 (2015)

[105] Guo, H., Gao, S., Tsui, K., Niu, T.: Simulation optimization for medical staff configuration at emergency department in Hong Kong. IEEE Transactions on Automation Science and Engineering **14**(4), 1655–1665 (2017)

[106] Guo, H., Goldsman, D., Tsui, K.L., Zhou, Y., Wong, S.Y.: Using simulation and optimisation to characterise durations of emergency department service times with incomplete data. International Journal of Production Research **54**(21), 6494–6511 (2016)

[107] Gürkan, G., Özge, A., Robinson, S.M.: Sample-path optimization in simulation. In: S.M. J.D. Tew, D. Sadowski, A. Seila (eds.) Proceedings of the 1994 Winter Simulation Conference, pp. 247–254. IEEE, USA (1994)

[108] Hajjarsaraei, H., Shirazi, B., Rezaeian, J.: Scenario-based analysis of fast track strategy optimization on emergency department using integrated safety simulation. Safety Science **107**, 9 – 21 (2018). DOI https://doi.org/10.1016/j.ssci.2018.03.025. URL http://www.sciencedirect.com/science/article/pii/S0925753517316181

[109] Halstrup, M.: Black-box optimization of mixed discrete-continuous optimization problems (2016). Retrieved from Eldorado - Repository of the TU Dortmund

[110] Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. **2**(1), 84–90 (1960). DOI 10.1007/BF01386213. URL https://doi.org/10.1007/BF01386213

[111] Hansen, P., Mladenović, N.: Variable neighborhood search: Principles and applications. European Journal of Operational Research **130**(3), 449 – 467 (2001). DOI https://doi.org/10.1016/S0377-2217(00)00100-4

[112] Hare, W.: Using derivative free optimization for constrained parameter selection in a home and community care forecasting model. In: Conference: International Perspectives on Operations Research and Health Care, Proceedings of the 34th Meeting of the EURO Working Group on Operational Research Applied to Health Sciences, pp. 61–73 (2010)

[113] Hemker, T., Fowler, K., Farthing, M., Von Stryk, O.: A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. Optimization and Engineering **9**, 341–360 (2008). DOI 10.1007/s11081-008-9048-0

[114] Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research **165**(2), 289 – 306 (2005)

[115] Ho, Y.C., Sreenivas, R., Vakili, P.: Ordinal optimization of deds. Discrete Event Dynamic Systems **2**, 61–88 (1992). DOI 10.1007/BF01797280

[116] Ho, Y.C., Zhao, Q., Jia, Q.S.: Ordinal Optimization: Soft Optimization for Hard Problems. Springer US (2007). DOI 10.1007/978-0-387-68692-9

[117] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975). Second edition, 1992

[118] Hoot, N., Aronsky, D.: Systematic review of emergency department crowding: causes, effects, and solutions. Annals of Emergency Medicine **52**(2), 126–136 (2008)

[119] Hoot, N.R., Zhou, C.H., Jones, I., Aronsky, D.: Measuring and forecasting emergency department crowding in real time. Annals of emergency medicine **49 6**, 747–55 (2007)

[120] Huang, D., Allen, T., Notz, W., Zeng, N.: Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models. Journal of Global Optimization **34**(3), 441–466 (2006). DOI 10.1007/s10898-005-2454-3

[121] Iba, K.: Reactive power optimization by genetic algorithm. IEEE Transactions on Power Systems **9**(2), 685–692 (1994)

[122] Ishizuka, Y., Shimizu, K.: Necessary and sufficient conditions for the efficient solutions of nondifferentiable multiobjective problems. IEEE Transactions on Systems, Man, and Cybernetics **SMC-14**(4), 625–629 (1984)

[123] Jahn, J.: Introduction to the Theory of Nonlinear Optimization, 3rd edn. Springer Publishing Company, Incorporated (2014)

[124] Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**, 455–492 (1998). DOI 10.1023/A:1008306431147

[125] Joshi, A., Rys, M.: Study on the effect of different arrival patterns on an emergency department's capacity using discrete event simulation. International Journal of Industrial Engineering **18**(1), 40–50 (2011)

[126] Joshi, V., Lim, C., Teng, S.G.: Simulation study: Improvement for non-urgent patient processes in the emergency department. Engineering Management Journal **28:3**, 145–157 (2016)

[127] Juan, A.A., Faulin, J., Grasman, S.E., Rabe, M., Figueira, G.: A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. Operations Research Perspectives **2**, 62–72 (2015)

[128] Juan, A.A., Lourenço, H.R., Mateo, M., Luo, R., Castella, Q.: Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. International Transactions in Operational Research **21**(1), 103–126 (2014)

[129] Juan, A.A., Pascual, I., Guimarans, D., Barrios, B.: Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem. International Transactions in Operational Research **22**(4), 647–667 (2015)

[130] Kadri, F., Chaabane, S., Tahonauthor, C.: A simulation-based decision support system to prevent and predict strain situations in emergency department systems. Simulation Modelling Practice and Theory **32**, 42–52 (2014)

[131] Kang, C., Choi, B.C.: An adaptive crashing policy for stochastic time-cost tradeoff problems. Computers & Operations Research **63**, 1–6 (2015)

[132] Kathirgamataby, N.: Note on the Poisson index of dispersion. Biometrika **40**, 225–228 (1953)

[133] Kaushal, A., Zhao, Y., Peng, Q., Strome, T., Weldon, E., Zhang, M., Chochinov, A.: Evaluation of fast track strategies using agent-based simulation modeling to reduce waiting time in a hospital emergency department. Socio-Economic Planning Sciences **50**, 18–31 (2015)

[134] Kelley, J.E.: Critical-path planning and scheduling: Mathematical basis. Operations research **9**(3), 296–320 (1961)

[135] Kelton, W., Sadowski, R., Zupick, N.: Simulation with Arena, 6th edn. McGraw–Hill Professional (2014)

[136] Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics **23**(3), 462–466 (1952)

[137] Kim, S., Pasupathy, R., Henderson, S.: A Guide to Sample Average Approximation, vol. 216, pp. 207–243. Springer, New York, NY (2015). DOI 10.1007/978-1-4939-1384-8_8

[138] Kim, S.H., Nelson, B.L.: Chapter 17 selecting the best system. In: S.G. Henderson, B.L. Nelson (eds.) Simulation, *Handbooks in Operations Research and Management Science*, vol. 13, pp. 501 – 534. Elsevier (2006). DOI https://doi.org/10.1016/S0927-0507(06)13017-0

[139] Kim, S.H., Whitt, W.: Are call center and hospital arrivals well modeled by nonhomogeneous Poisson process ? Manufactory & Service Operations Management **16**, 464–480 (2014)

[140] Kim, S.H., Whitt, W.: Choosing arrival process models for service systems: Tests of a nonhomogeneous Poisson process. Naval Research Logistics **61**, 66–90 (2014)

[141] Kim, S.H., Whitt, W.: The power of alternative Kolmogorov–Smirnov tests based on transformations of the data. ACM Transactions on Modeling and Computer Simulation **25**(4), 1–22 (2015)

[142] Kimms, A.: Mathematical Programming and Financial Objectives for Scheduling Projects. Kluwer Academic Publishers, Boston, USA (2001)

[143] Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science (New York, N.Y.) **220**, 671–80 (1983). DOI 10.1126/science.220.4598.671

[144] Kitayama, S., Arakawa, M., Yamazaki, K.: Penalty function approach for the mixed discrete nonlinear problems by particle swarm optimization. Structural and Multidisciplinary Optimization **32**, 191–202 (2006). DOI 10.1007/s00158-006-0021-2

[145] Kleijnen, J., Beers, W., Nieuwenhuyse, I.: Expected improvement in efficient global optimization through bootstrapped kriging. Journal of Global Optimization **54**, 1–15 (2011). DOI 10.1007/s10898-011-9741-y

[146] Kleijnen, J.P.: Kriging metamodeling in simulation: A review. European Journal of Operational Research **192**(3), 707 – 716 (2009). DOI https://doi.org/10.1016/j.ejor.2007.10.013

[147] Kleijnen, J.P.C.: Design and Analysis of Simulation Experiments, 1st edn. Springer Publishing Company, Incorporated (2007)

[148] Klerides, E., Hadjiconstantinou, E.: A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations. Computers & Operations Research **37**(12), 2131–2140 (2010)

[149] Kolda, T.G., Lewis, R.M., Torczon, V.: A generating set direct search augmented Lagrangian algorithm for optimization with a combination of general and linear constraints. Tech. Rep. SAND2006-5315, Sandia National Laboratories (2006). DOI 10.2172/893121. URL http://www.osti.gov/scitech/biblio/893121

[150] Kolisch, R., Padman, R.: An integrated survey of deterministic project scheduling. Omega **29**(3), 249–272 (2001)

[151] Kramer, A., Dosi, C., Iori, M., Vignoli, M.: Successful implementation of discrete event simulation: the case of an italian emergency department (2020)

[152] Kuo, Y.H., Leung, J.M., Graham, C.A., Tsoi, K.K., Meng, H.M.: Using simulation to assess the impacts of the adoption of a fast-track system for hospital emergency services. Journal of Advanced Mechanical Design, Systems, and Manufacturing **12**(3), 1–11 (2018)

[153] Kuo, Y.H., Rado, O., Lupia, B., Leung, J.M.Y., Graham, C.A.: Improving the efficiency of a hospital emergency department: a simulation study with indirectly imputed service-time distributions. Flexible Services and Manufacturing Journal **28**(1), 120–147 (2016)

[154] Ólafsson, S.: Chapter 21 metaheuristics. In: S.G. Henderson, B.L. Nelson (eds.) Simulation, *Handbooks in Operations Research and Management Science*, vol. 13, pp. 633 – 654. Elsevier (2006). DOI https://doi.org/10.1016/S0927-0507(06)13021-2

[155] Laguna, M., Gortázar, F., Gallego, M., Duarte, A., Marti, R.: A black-box scatter search for optimization problems with integer variables. Journal of Global Optimization **58** (2013). DOI 10.1007/s10898-013-0061-2

[156] Larson, J., Leyffer, S., Palkar, P., Wild, S.M.: A method for convex black-box integer global optimization (2019)

[157] Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numerica **28**, 287–404 (2019). DOI 10.1017/s0962492919000060. URL http://dx.doi.org/10.1017/S0962492919000060

[158] Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for integer programming. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600), vol. 2, pp. 1582–1587 vol.2 (2002)

[159] Laslo, Z.: Activity time–cost tradeoffs under time and cost chance constraints. Computers & Industrial Engineering **44**(3), 365–384 (2003)

[160] Lau, T.W.E., Ho, Y.C.: Universal alignment probabilities and subset selection for ordinal optimization. J. Optim. Theory Appl. **93**(3), 455–489 (1997). DOI 10.1023/A:1022614327007. URL https://doi.org/10.1023/A:1022614327007

[161] Law, A.: Simulation Modeling and Analysis, fifth edn. McGraw-Hill (2015)

[162] Le Digabel, S.: Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. ACM Transactions on Mathematical Software **37**(4), 1–15 (2011)

[163] Lewis, P.: Some results on tests for poisson processes. Biometrika **52**, 67–77 (1965)

[164] Lewis, R., Torczon, V.: A globally convergent augmented lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM J. Optim. **12**, 1075–1089 (2002)

[165] Lin, C.J., Lucidi, S., Palagi, L., Risi, A.: Decomposition algorithm model for singly linearly-constrained problems subject to lower and upper bounds. Journal of Optimization Theory and Applications **141**, 107–126 (2009). DOI 10.1007/s10957-008-9489-9

[166] Lin, C.K.Y., Ling, T.W.C., Yeung, W.K.: Resource allocation and outpatient appointment scheduling using simulation optimization. Journal of Healthcare Engineering **2017** (2017)

[167] Liu, Z., Cabrera, E., Taboada, M., Epelde, F., Rexachs, D., Luque, E.: Quantitative evaluation of decision effects in the management of emergency department problems. Procedia Computer Science **51**, 433–442 (2015)

[168] Liu, Z., Rexachs, D., Epelde, F., Luque, E.: A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity. Comput. Ind. Eng. **103**(C), 300–309 (2017). DOI 10.1016/j.cie.2016.11.036. URL https://doi.org/10.1016/j.cie.2016.11.036

[169] Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for bound constrained mixed-integer optimization. Computational Optimization and Applications - COMPUT OPTIM APPL **53** (2012). DOI 10.1007/s10589-011-9405-3

[170] Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for mixed-integer constrained optimization problems. Journal of Optimization Theory and Applications **164** (2014). DOI 10.1007/s10957-014-0617-4

[171] Liuzzi, G., Lucidi, S., Rinaldi, F.: An algorithmic framework based on primitive directions and nonmonotone line searches for black-box optimization problems with integer variables. Mathematical Programming Computation **12**(4), 673–702 (2020). DOI 10.1007/s12532-020-00182-7. URL https://doi.org/10.1007/s12532-020-00182-7

[172] Liuzzi, G., Lucidi, S., Sciandrone, M.: A derivative-free algorithm for linearly constrained finite minimax problems. SIAM Journal on Optimization pp. 1054–1075 (2006)

[173] Liuzzi, G., Lucidi, S., Sciandrone, M.: Sequential penalty derivative-free methods for nonlinear constrained optimization. SIAM J. Optim. **20**, 2614–2635 (2010)

[174] Lucidi, S., Maurici, M., Paulon, L., Rinaldi, F., Roma, M.: A derivative-free approach for a simulation-based optimization problem in healthcare. Optimization Letters **10**, 219–235 (2016)

[175] Lucidi, S., Maurici, M., Paulon, L., Rinaldi, F., Roma, M.: A simulation-based multiobjective optimization approach for health care service management. IEEE Transactions on Automation Science and Engineering, **13**(4), 1480–1491 (2016)

[176] Lucidi, S., Piccialli, V.: An algorithm model for mixed variable programming. SIAM Journal on Optimization **15** (2005). DOI 10.1137/S1052623403429573

[177] Lucidi, S., Sciandrone, M.: On the global convergence of derivative-free methods for unconstrained optimization. SIAM Journal on Optimization **13**, 97–116 (2002). DOI 10.1137/S1052623497330392

[178] Luscombe, R., Kozan, E.: Dynamic resource allocation to improve emergency department efficiency in real time. European Journal of Operational Research **255**(2), 593 – 603 (2016). DOI https://doi.org/10.1016/j.ejor.2016.05.039

[179] Lukšan, V., Vlček, J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report VT798-00, Institute of Computer Science, Academy of Sciences of the Czech Republic (2000)

[180] Maccarrone, L., Giovannelli, T., Ferone, D., Panadero, J., Juan, A.A.: A simheuristic algorithm for solving an integrated resource allocation and scheduling problem. In: 2018 Winter Simulation Conference (WSC), pp. 3340–3351 (2018). DOI 10.1109/WSC.2018.8632296

[181] Mangasarian, O.: Nonlinear programming. Classics in Applied Mathematics (1994)

[182] Mazen, J., Ghada, R., Saliba, M., Jabbour, R., Hitti, E.: Improving emergency department door to doctor time and process reliability. a successful implementation of lean methodology. Medicine **94**(42), 1–6 (2015)

[183] Presidenza del Consiglio dei Ministri: Accordo Stato Regioni, Riorganizzazione del sistema di emergenza urgenza in rapporto alla continuità assistenziale. Rep. Atti n.36/CSR (2013)

[184] Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & Operations Research **24**(11), 1097 – 1100 (1997). DOI https://doi.org/10.1016/S0305-0548(97)00031-2

[185] Müller, J.: MISO: mixed-integer surrogate optimization framework. Optimization and Engineering **17**, 1–27 (2015). DOI 10.1007/s11081-015-9281-2

[186] Müller, J., Shoemaker, C., Piché, R.: SO-I: A surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications. Journal of Global Optimization (2013). DOI 10.1007/s10898-013-0101-y

[187] Müller, J., Shoemaker, C.A., Piché, R.: SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Computers & Operations Research **40**(5), 1383 – 1400 (2013). DOI https://doi.org/10.1016/j.cor.2012.08.022

[188] Müller, J., Shoemaker, C.A., Piché, R.: SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Computers & Operations Research **40**(5), 1383 – 1400 (2013). DOI https://doi.org/10.1016/j.cor.2012.08.022

[189] Moder, J., Phillips, C., Davis, E.: Project Management with CPM, PERT, and Precedence Diagramming. Blitz Publishing Company (1983)

[190] Mohan, C., Nguyen, H.: A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems. Computational Optimization and Applications **14**, 103–132 (1999). DOI 10.1023/A:1008761113491

[191] Mokhtari, H., Aghaie, A., Rahimi, J., Mozdgir, A.: Project time–cost trade-off scheduling: a hybrid optimization approach. The International Journal of Advanced Manufacturing Technology **50**(5), 811–822 (2010)

[192] Moré, J., Wild, S.: Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization **20**, 172–191 (2009). DOI 10.1137/080724083

[193] Morley, C., Unwin, M., Peterson Gregory M andStankovich, J., Kinsman, L.: Emergency department crowding: A systematic review of causes, consequences and solutions. PLoS One **13** (2018)

[194] Nahhas, A., Alwadi, A., Reggelin, T.: Simulation and the emergency department overcrowding problem. Procedia Engineering **178**, 368–376 (2017)

[195] Nawaz, M., Enscore Jr, E.E., Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega **11**(1), 91–95 (1983)

[196] Newby, E., Ali, M.: A trust-region-based derivative free algorithm for mixed integer programming. Computational Optimization and Applications **60**, 199–229 (2014). DOI 10.1007/s10589-014-9660-1

[197] Pagnoncelli, B.K., Ahmed, S., Shapiro, A.: Sample Average Approximation Method for Chance Constrained Programming: Theory and Applications. Journal of Optimization Theory and Applications **142**(2), 399–416 (2009). DOI 10.1007/s10957-009-9523-6. URL https://ideas.repec.org/a/spr/joptap/v142y2009i2d10.1007_s10957-009-9523-6.html

[198] Parno, M.D., Hemker, T., Fowler, K.R.: Applicability of surrogates to improve efficiency of particle swarm optimization for simulation-based problems. Engineering Optimization **44**(5), 521–535 (2012). DOI 10.1080/0305215X.2011.598521. URL https://doi.org/10.1080/0305215X.2011.598521

[199] Patvivatsiri, L.: A simulation model for bioterrorism preparedness in an emergency room. In: L. Perrone, F.P. Wieland, J. Liu, B. Lawson, D. Nicol, R. Fujimoto (eds.) Proceedings of the 2006 Winter Simulation Conference, pp. 501–508 (2006)

[200] Paul, S.A., Reddy, M.C., De Flitch, C.J.: A systematic review of simulation studies investigating emergency department overcrowding. Simulation **86**(8-9), 559–571 (2010)

[201] Porcelli, M., Toint, P.: BFO, a trainable derivative-free brute force optimizer for nonlinear bound-constrained optimization and equilibrium computations with continuous and discrete variables. ACM Transactions on Mathematical Software **44**, 1–25 (2017). DOI 10.1145/3085592

[202] Powell, M.: The bobyqa algorithm for bound constrained optimization without derivatives. Technical Report, Department of Applied Mathematics and Theoretical Physics (2009)

[203] Rado, O., Lupia, B., Leung, J.M.Y., Kuo, Y.H., Graham, C.A.: Using simulation to analyze patient flows in a hospital emergency department in hong kong. In: A. Matta, J. Li, E. Sahin, E. Lanzarone, J. Fowler (eds.) Proceedings of the International Conference on Health Care Systems Engineering, pp. 289–301. Springer International Publishing, Cham (2014)

[204] Reeves, C.: Genetic algorithms for the operations researcher. INFORMS Journal on Computing **9**, 231–250 (1997)

[205] Reid, P., Compton, W., Grossman, J., Fanjiang, G.: Building a Better Delivery System: A New Engineering/Health Care Partnership. The National Academies of Press, Washington, D.C. (2005)

[206] Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics **22**, 400–407 (1951)

[207] Rockafellar, R.T.: Convex Analysis. Princeton University Press (1970)

[208] Romeijn, H.E., Zabinsky, Z.B., Graesser, D.L., Neogi, S.: New reflection generator for simulated annealing in mixed-integer/continuous global optimization. J. Optim. Theory Appl. **101**(2), 403–427 (1999). DOI 10.1023/A:1021745728358. URL https://doi.org/10.1023/A:1021745728358

[209] Ministero della Salute: Documento di proposta di aggiornamento delle linee guida sul triage intraospedaliero. G.U. Serie Generale, n. 285, 07 dicembre 2001

[210] Schull, M., Kiss, A., Szalai, J.P.: The effect of low-complexity patients on emergency department waiting times. Annals of emergency medicine **49 3**, 257–64, 264.e1 (2007)

[211] Seong-Hee Kim, Nelson, B.L.: Recent advances in ranking and selection. In: 2007 Winter Simulation Conference, pp. 162–172 (2007). DOI 10.1109/WSC.2007.4419598

[212] Shapiro, A.: Simulation based optimization. In: Proceedings of the 28th Conference on Winter Simulation, WSC '96, p. 332–336. IEEE Computer Society, USA (1996). DOI 10.1145/256562.256644. URL https://doi.org/10.1145/256562.256644

[213] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), pp. 69–73 (1998)

[214] Shimizu, K., Ishizuka, Y., Bard, J.: Nondifferentiable and Two-Level Mathematical Programming. Springer US (1997). DOI 10.1007/978-1-4615-6305-1

[215] Shirazi, B.: Fast track system optimization of emergency departments: Insights from a computer simulation study. International Journal of Modeling, Simulation, and Scientific Computing **07** (2016). DOI 10.1142/S179396231650015X

[216] Sobol, I.: Uniformly distributed sequences with an additional uniform property. USSR Computational Mathematics and Mathematical Physics **16**(5), 236 – 242 (1976). DOI https://doi.org/10.1016/0041-5553(76)90154-3

[217] Spall, J.C.: Introduction to Stochastic Search and Optimization: estimation, simulation, and control., 1 edn. John Wiley & Sons, Inc., USA (2003)

[218] Sriver, T.A., Chrissis, J.W., Abramson, M.A.: Pattern search ranking and selection algorithms for mixed variable simulation-based optimization. European Journal of Operational Research **198**(3), 878 – 890 (2009). DOI https://doi.org/10.1016/j.ejor.2008.10.020

[219] Taboada, M., Cabrera, E., Iglesias, M.L., Epelde, F., Luque, E.: An agent-based decision support system for hospitals emergency departments. Procedia Computer Science **4**, 1870–1879 (2011). Proceedings of the International Conference on Computational Science, ICCS 2011

[220] Taillard, E.: Benchmarks for basic scheduling problems. European Journal of Operational Research **64**, 278–285 (1993)

[221] Talbi, E.G.: Metaheuristics: From Design to Implementation. Wiley Publishing (2009)

[222] Torczon, V.: On the convergence of pattern search algorithms. SIAM Journal on Optimization **7**(1), 1–25 (1997)

[223] Ucar, I., Smeets, B., Azcorra, A.: simmer: Discrete-event simulation for R. Journal of Statistical Software **90**(2), 1–30 (2019). DOI 10.18637/jss.v090.i02

[224] van Beers, W.C.M., Kleijnen, J.P.C.: Kriging interpolation in simulation: a survey. In: Proceedings of the 2004 Winter Simulation Conference, 2004., vol. 1, p. 121 (2004). DOI 10.1109/WSC.2004.1371308

[225] Van Slyke, R.M.: Monte carlo methods and the pert problem. Operations Research **11**(5), 839–860 (1963)

[226] Vanbrabant, L., Martin, N., Ramaekers, K., Braekers, K.: Quality of input data in emergency department simulations: Framework and assessment techniques. Simulation Modelling Practice and Theory **91**, 83 – 101 (2019)

[227] Verweij, B., Ahmed, S., Kleywegt, A.J., Nemhauser, G., Shapiro, A.: The sample average approximation method applied to stochastic routing problems: A computational study. Comput. Optim. Appl. **24**(2–3), 289–333 (2003). DOI 10.1023/A:1021814225969. URL https://doi.org/10.1023/A:1021814225969

[228] Vicente, L., Custódio, A.: Analysis of direct searches for discontinuous functions. Mathematical Programming **133**, 1–27 (2009). DOI 10.1007/s10107-010-0429-8

[229] Wah, B., Chen, Y., Wang, T.: Simulated annealing with asymptotic convergence for nonlinear constrained global optimization. Journal of Global Optimization **39**, 153–162 (2002)

[230] Wang, L.: An agent-based simulation for workflow in emergency department. In: 2009 Systems and Information Engineering Design Symposium, pp. 19–23 (2009)

[231] Weiss, S., Derlet, R., Arndahl, J., Ernst, A., Richards, J., Fernández-Frankelton, M., Schwab, R., Stair, T., Vicellio, P., Levy, D., Brautigan, M., Johnson, A., Nick, T.: Estimating the degree of emergency department overcrowding in academic medical centers: Results of the national ed overcrowding study (NEDOCS). Academic Emergency Medicine **11**(1), 38–50 (2004)

[232] Weiss, S., Ernst, A.A., Nick, T.G.: Comparison of the national emergency department overcrowding scale and the emergency department work index for quantifying emergency department crowding. Academic emergency medicine: official journal of the Society for Academic Emergency Medicine **13 5**, 513–8 (2006)

[233] Whitley, D.: A genetic algorithm tutorial. Statistics and Computing **4** (1998). DOI 10.1007/BF00175354

[234] Whitt, W., Zhang, X.: A data-driven model of an emergency department. Operations Research for Health Care **12**, 1 – 15 (2017)

[235] Wong, Z.S.Y., Lit, A.C.H., Leung, S.Y., Tsui, K.L., Chin, K.S.: A discrete-event simulation study for emergency room capacity management in a hong kong hospital. 2016 Winter Simulation Conference (WSC) pp. 1970–1981 (2016)

[236] Xia, N., Sharman, R., Rao, H., Dutta, S.: A simulation-based study for managing hospital emergency departmet's capacity in extreme events. International Journal of Business Excellence **5**, 140–154 (2012)

[237] Yang, S., Liu, H., Pan, C.: An efficient derivative-free algorithm for bound constrained mixed-integer optimization. Evolutionary Intelligence (2019). DOI 10.1007/s12065-019-00326-2

[238] Yiqing, L., Xigang, Y., Yongjian, L.: An improved pso algorithm for solving non-convex nlp/minlp problems with equality constraints. Computers & Chemical Engineering **31**(3), 153 – 162 (2007). DOI https://doi.org/10.1016/j.compchemeng.2006.05.016

[239] Yoshida, H., Kawata, K., Fukuyama, Y., Takayama, S., Nakanishi, Y.: A particle swarm optimization for reactive power and voltage control considering voltage security assessment. IEEE Transactions on Power Systems **15**(4), 1232–1239 (2000)

[240] Yousefi, M., Yousefi, M., Fogliatto, F.: Simulation-based optimization methods applied in hospital emergency departments: A systematic review. Simulation (2020). DOI 10.1177/0037549720944483. First published online: August 5, 2020

[241] Yu-Chi Ho: Performance evaluation and perturbation analysis of discrete event dynamic systems. IEEE Transactions on Automatic Control **32**(7), 563–572 (1987). DOI 10.1109/TAC.1987.1104665

[242] Yun-Chia Liang, Smith, A.E.: An ant colony optimization algorithm for the redundancy allocation problem (rap). IEEE Transactions on Reliability **53**(3), 417–423 (2004)

[243] Zeinali, F., Mahootchi, M., Sepehri, M.: Resource planning in the emergency departments: a simulation-base metamodeling approach. Simulation Modelling Practice and Theory **53**, 123–138 (2015)

[244] Zhang, H., Best, T., Chivu, A., Meltze, D.: Simulation-based optimization to improve hospital patient assignment to physicians and clinical units. Health Care Management Science (2019)