Preprint published in the *Computers & Industrial Engineering*. DOI: 10.1016/j.cie.2016.11.036

A Simulation and Optimization Based Method for Calibrating Agent-based Emergency Department Models Under Data Scarcity

Zhengchun Liu[a], Dolores Rexachs[a], Francisco Epelde[b], Emilio Luque[a]

[a] *Computer Architecture & Operating Systems, University Autonoma of Barcelona, Barcelona, Spain*
[b] *Medicine Department, Hospital Universitari Parc Taulí, Barcelona, Spain*

Abstract

To tackle the problem of efficiently managing increasingly complex systems, simulation models have been widely used. This is because simulation is safer, less expensive, and faster than field implementation and experimenting. To achieve high fidelity and credibility in conducting prediction and exploration of the actual system with simulation models, a rigorous calibration and validation procedure should firstly be applied. However, one of the key issues in calibration is the acquisition of valid source information from the target system. The aim of this study is to develop a systematic method to automatically calibrate a general emergency department model with incomplete data. The simulation-based optimization was used to search for the best value of model parameters. Then we present a case study to particularly demonstrate the way to calibrate an agent-based model of an emergency department with real data scarcity. The case study indicates that the proposed method appears to be capable of properly calibrating and validating the simulation model with incomplete data.

Keywords: Simulation-based Optimization, Model Calibration, Agent-Based Model, Emergency Department,

1. Introduction

With the rapid growth of computational techniques, computational thinking brings researchers and practitioners into a new dimension of traditional modeling and simulation tasks. That is, the computational science transforms observed complex phenomena into conceptual models. Then the models are formulated into algorithms that can be executed to yield predictions and estimate hidden parameters. This generates an additional understanding of the phenomenon and leads to more specific models of the phenomenon [1]. From a theoretical computation perspective, the simulation of a system can be defined as an *"imitation (on a computer) of a system as it progresses through time"* [2]. Although a simulator is mostly designed for prediction, the simulator should firstly be able to imitate the real system. Generally, a simulator of a specific system is comprised of the following: input (X), the model or transformation function ($f(X)$), and output (Y). For an accurate simulator, when we put the same input as in a real system, the output of simulator should be close enough to the output of the actual system. Since $f(X)$ is based on abstractions, idealization, and many disputable assumptions, the model must be fine-tuned according to some historical input-output samples from the target system in order to get reliable simulation results.

The emergency department (ED) is a typical complex system, which serves essential needs in society, delivering emergency health care and simultaneously acting as a safety net provider [3]. In recent years, simulation has emerged as an increasingly effective tool to study ED related problems and support making

☆Corresponding author at: University Autonoma of Barcelona, Bellaterra 08193, Spain. Tel.: +34 935 812 888; fax: +34 935 812 478.
Email addresses: lzhengchun@caos.uab.es (Zhengchun Liu), emilio.luque@uab.cat (Emilio Luque)

decisions to efficiently manage the complex ED system. While these simulation models can be advantageous to engineers, the models must be calibrated and validated, i.e., the model should first be able to accurately imitate the real system. Advances in computational technology, along with the increased complexity of system design and management, have created an environment in which microscopic simulation models have become useful tools for managing complex system. Among which, the Agent-based Model (ABM) is one of the most important tools for exploring emergent behavior (a phenomenon that describes the behavior of a system, which cannot be explained alone by the sum of its parts [4]), mostly because it can provide a way to see the forest through the trees and insight is often more important than sheer numbers [5–7].

The agent-based simulation models encompass numerous independent parameters to describe the individual behavior of the system components. Reliable and complete real data from the target system is obviously the precondition for setting up an accurate simulator. Unfortunately, many of the parameters are either unavailable in historical data or difficult to measure in a real situation, yet they can have a substantial impact on the model's accuracy. Thus, when real data was incomplete to allow direct estimation of the model parameters, a calibration process (also known as tuning) has to be conducted to indirectly estimate good values for those unknown parameters. However, the calibration of model parameters for an ABM is a big challenge for standard calibration techniques, due to the large parameter search-space, long simulation runtime, uncertainties in the structural model design and different observation levels upon which the model needs to be calibrated [8]. Given this, the model parameter calibration problem can be formulated as a stochastic programming problem whose objective function is an associated measurement of an experimental simulation. Nevertheless, the objective function is typically (a) subject to various levels of randomness, (b) not necessarily differentiable, and (c) computationally expensive to evaluate due to the complexity of the model.

Accordingly, conventional calibration, which is carried out manually by using the trial-and-error method, is time consuming and tedious. A systematic method to automatically search for the optimal value of model parameters is promising. The simulation-based optimization is an emerging field which integrates optimization techniques into simulation analysis. The primary goal of simulation-based optimization is to optimize the performance of a system through simulation. More specifically, it is a way to find the optimal set of parameters for a given criterion. Then the optimal parameter set will enable the model to achieve a specific function optimally or the results of the simulation are close enough to actual data. Therefore, if we set the model input the same as reality and we consider the unknown model parameters as variables, and the similarity between simulation output and actual system output as objective, the optimization is a model calibration process. When some of the model parameters are missing and impossible to get from the real system, this optimization process will be able to find the optimal values for setting up the model. Thus, the precondition for the calibration process is a set of reliable input-output pairs from the target system.

In this article, we will address a critical step in simulating a complex system - the systematic model calibration in the face of data scarcity. To the best of our knowledge, limited research has been conducted on this thorny and critical problem of estimation in the face of data scarcity. The simulation-based optimization was conducted by using an existing tool [9–11] developed by Sandia National Laboratory. According to the practical requirements of evaluating a simulation-based objective function, an initial distance-based lookup mechanism was proposed to further speed up the optimization. The rest of the paper is structured as follows: section 2 gives a literature review on related work. The method to calibrate agent-based ED model is given in section 3. With the presented method, section 4 gives a case study which calibrates a general agent-based model of an ED to simulate the ED of the Hospital of Sabadell (a university tertiary level hospital in Barcelona, Spain) with incomplete data (missing duration of key services). This case study will thoroughly demonstrate the way to calibrate an agent-based ED model by using the presented method. Finally, section 5 draws the conclusions.

2. Related work

Model calibration is the task of adjusting an already existing model to a reference system. T.G. Trucano et al. thoroughly discussed the relation of calibration and validation in Ref. [12]. They identified some technical challenges that must be resolved for successful validation and calibration of a predictive modeling

capability. Their findings proved the possibility of validation and highlighted great practical difficulties associated with model parameter calibration and validation. M. Hofmann [13] introduced a formal approach to model calibration, within the frame of the presented formalism it is shown that the computational complexity of model calibration is NP-complete. The author addressed the issue that for huge model federations the complexity of parameter calibration could draw a serious line with respect to the validation of the federation and its cost-benefit ratio. This is mostly because in a huge model of a complex system, no single person has an overview of the whole simulation, and the interpretation of unexpected results is extremely difficult. Therefore, a manual trial-and-error method does not work for this kind of model (e.g., an agent-based model).

As described in section 1, the model parameter calibration process can be easily formed as a simulation-based optimization process. Due to the complex behavior of the objective function, Evolutionary Algorithms (EAs) are often used to efficiently explore large parameter spaces. However, EA still takes a considerable amount of time because it requires a large number of simulation runs, and each run takes considerable length of time in simulation. To this end, M. Wagner et al. [4] proposed the use of complexification to improve the performance of EAs as it emulates the natural way of evolution. This method has been used for parameter estimation of multi-agent based models. J. Zhong et al.[14] proposed an evolutionary framework to automate the crowd model calibration process. In the proposed framework, a density-based matching scheme is introduced. By using the dynamic density of the crowd over time, and a weight landscape to emphasize important spatial regions, the proposed matching scheme provides a generally applicable way to evaluate the simulated crowd behaviors. Besides, the authors also proposed a hybrid search mechanism based on differential evolution to efficiently tune parameters of crowd models. And in Ref [15], J. Zhong et al. proposed another novel evolutionary algorithm named differential evolution with sensitivity analysis and Powell's method (DESAP) for model calibration. The proposed DESAP first applies an entropy-based sensitivity analysis operation to dynamically identify important parameters of the model. Then, Powell's method is performed periodically to fine-tune the important parameters of the best individual in the population. Finally, in each generation, the evolutionary operators are performed on a small number of better individuals in the population. Their new search mechanisms are integrated into the differential evolution framework to improve search efficiency. In summary, different from conventional mathematical models, the calibration of agent-based models has its own challenge and are attracting researchers' attention. However, all of these developed algorithms are mostly focused on solving agent-based crowd behavior model calibration problems with complete data, in which the system metrics and objective function are different from the requirement of tuning an agent-based ED model.

In contrast to traditional black box search methods, which only consider the input and output of simulation model, M. Fehler et al. [8, 16] proposed a promising white box calibration approach, which uses the knowledge of the agent-based model to improve the tuning process. In this, the idea is to reduce the parameter space by breaking down the model into smaller sub-models. Each of the sub-models is then calibrated before merging them back to form the model. However, in this method, the division and fusion operations are difficult steps and they require additional knowledge about the model, and this knowledge may not be available for simulation users (non-developer). Moreover, the fusion operation has to merge calibrated sub-models into a calibrated higher model, which is not automatic.

In summary, although parameter calibration is critical and one of the key steps in modeling & simulation work, and it can be easily formalized as a simulation-based optimization problem, to the best of our knowledge, such model parameter calibration problems under data scarcity have not been explicitly addressed in the literature. No literature was found providing an automatic calibration tool for simulation users to calibrate the general model for a new system without the involvement of model developers. Having shown that, the overall goal of this work is to provide a practical calibration method to automatically calibrate an agent-based ED model for simulating a new system.

3. A way to calibrate with data scarcity

Calibration is traditionally conceptualized as a step in model validation. It involves systematic adjustment of model parameters so that model outputs can accurately reflect the actual system behavior [12].

To calibrate a model, three important issues need to be addressed. The first issue is to select significant metrics to represent the emergent behavior of the target system and to specify a general and effective fitness function to measure the distance between a simulated scenario and the real situation. The second issue is to reduce the computation time because exhaustive search in parameter space is expensive (exponential growth with the number of parameters). The third issue is to obtain robust solutions for avoiding the over-fitting problem. That is, the calibrated model is not only able to fit historical dataset (dataset for calibration), but is also able to predict reliable result with new input data. Due to the fact that all the services in an ED are interdependent, it is unreasonable to characterize parameters one-by-one or evaluate fitness process-by-process. To address this issue, one way is to consider all unknown parameters as a set, then simulate with the set and evaluate the similarity of system metrics as a whole. That is to say, a full simulation has to be carried out to evaluate one set of parameters, and changes to any of the parameters will result in one new simulation scenario. The following subsections 3.1 - 3.3 will detail all the issues and processes on calibrating the model parameters under data scarcity. A case study will be given in section 4.

3.1. Problem formulation

The goal of simulation is to imitate the behavior of a real system so as to accurately predict system behavior under unknown scenarios. Due to data scarcity, some of the model parameters are difficult to obtain directly from actual data, we thus have to tune these parameters indirectly with the goal of producing similar macroscopic behaviors as in real situation. Thus, the calibration process of agent-based model of ED is defined as: *Given an agent-based model, a setting of parameter X to be calibrated, the task is to find the global optimal X^* that minimizes the fitness function.* From an optimization point of view, the calibration can simply be expressed as:

$$\begin{aligned} &Minimize \quad f_{fitness}\left(p_1, p_2, \cdots, p_n\right) = K\left(actual, simulation\right) \\ &Subject \quad to: \\ &p_1, p_2, \cdots, p_n \quad make\ sense\ in\ real\ situation \end{aligned}$$

Where, $K\left(actual, simulation\right)$ is a function to evaluate the similarity between simulation results and actual data. The $\{p_1, p_2, \cdots, p_n\}$ is the set of parameter values (also called scenario in this study). However, there are two main challenges in solving this global optimization problem. One is that the condition – *make sense* is difficult to describe in the optimization model because these parameters represent the behavior of a physical system (rather than sheer numbers). The other challenge is that the fitness function is non-convex, it has a very complex response surface, and it is computationally expensive to evaluate. However, if we decompose the condition, i.e., only consider the lower/upper bounds of the parameters (main part of the condition), though the optimization process is not an absolute global optimization problem because the best fitness parameters may not make sense in reality, it becomes solvable. Therefore, a solution that gives the best of both worlds is: searching for the local minimum points under boundary condition, then manually checking if the solution makes sense in reality. Considering the over-fitting problem and model validation, a systematic method to calibrate and validate a general model is illustrated in Figure 1.

It is worth noting that the proposed method focused on solving the practical problem. The method is based on the assumption that the calibration result is acceptable with a certain margin of error. That is, the proposed method cannot guarantee finding the theoretical global optimum point, but it can find an acceptable point in a practical application. As shown in Figure 1, the reference data (actual input-output pairs) was divided into three parts for training, test and validating respectively. The key difference between test and validation processing is the feedback, i.e., simulator performance on validation tests will not affect the calibration process, while performance on test sets will affect the adjustment of the parameter set and the Monte Carlo scheme (e.g., boundary constraints). The Monte Carlo method (under the boundary constraint , e.g., as shown in Table 1) is used to generate initial value for the optimization solver. To make the calibration process more automatic, the proposed method will try to find a certain number of local minimum points, then gradually eliminate on test and validation datasets, and only provide several candidates to carry out manual checking. Thus, the simulation users only need to be involved in the calibration at the end. More

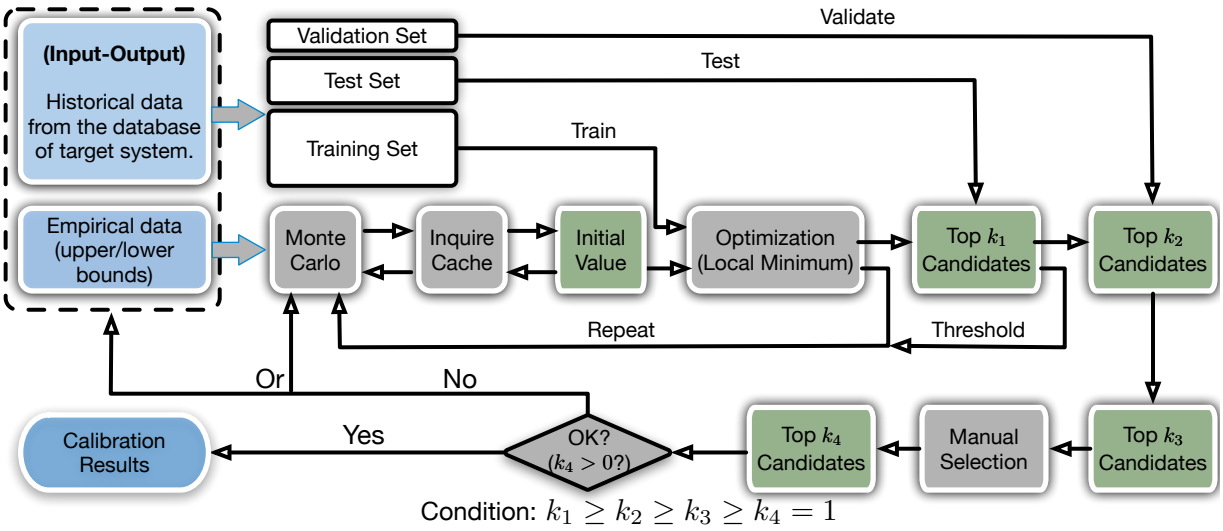


Figure 1: The systematic model calibration and validation process. The k_2 and k_3 is what is left after applying threshold selection on test and validation datasets separately. The cache checking modular is designed to avoid duplicate optimization from close starting points. The manual selection is designed for experienced ED staff, to eliminate some solutions that could result in good fitness but make less sense in reality.

specifically, the k_1 local optimum points will be evaluated on testing and validation datasets in sequence, a fitness threshold should be fulfilled and top $k_i (i = 2, 3)$ candidates will be selected, if there is more than one candidate left after evaluating validation datasets, the top k_3 candidates will be manually checked by experienced ED staff and one which makes the best sense in reality will be chosen as the final solution. If any of $\{k_2, k_3, k_4\}$ is zero (none can pass the threshold nor make sense in reality), the calibration process will either return to Monte Carlo to search more local minimum or re-divide the historical datasets for training and testing, and start over again. This depends on k_1 and overlap ratio of optimum points (assessed with inter-distance detailed in subsection 3.3, as shown in Figure 4a).

3.2. Evaluation metrics

As an agent-based model, the individual's behaviors, e.g., behaviors of a single patient, are highly dynamic and stochastic, matching these behaviors individually is usually unfeasible and unnecessary. Namely, the similarity between simulator and actual ED should be evaluated in a systematic manner rather than getting entangled in each of the agents. Thus, to compare behavior of two complex system, the selection of system key performance indicators (KPIs) is crucial, and two issues must be addressed. On the one hand, the selected KPIs should be able to significantly reflect the macroscopic behavior of the target system. On the other hand, it should be possible to retrieve from historical data, and the historical data should be convincing for the KPIs. Ref. [17] and [18] listed various metrics by which ED operations can be measured. Among which, the LoS (length of stay), LWBS (percentage of patients who leave without being seen), door-to-diagnostic evaluation by a qualified medical professional (arrival time to provider contact time, also known as "door-to-doctor" time) and ambulance diversion (amount of time ambulances are diverted away from the ED) are commonly used. All of those metrics are possible to extract from the agent-based simulator. Given this, and considering that patient-centered records are the real data we have, the records include the time stamp of patients' arrival and discharge, thus the patients' length of stay in the ED could be retrieved. Moreover, the LoS is comprised of all the time on service and waiting/pending. It is one of the composite indicators which is able to indicate patients' flow as well as the system's efficiency. Thus LoS was used as the setting of metrics for system performance in this work.

Furthermore, patients' LoS is one of the aggregate behaviors of the ED system, when comparing simulated LoS with actual LoS, the absolute difference of their average cannot fully represent their differences because

the same average may come from quite different distributions (e.g., uniform versus exponential distribution). In view of this, we analyze the actual LoS distribution by using a histogram. For each of the simulation outputs, we perform the same analysis. Thus, we will get two distributions and the goal is to measure the similarity between them, and the similarity will be used to evaluate the similarity between actual system and simulation results.

3.3. Optimization method

As described in subsection 3.1, the calibration process can be formulated as a series of local minimum searching problems. There are many ready-made methods for searching local minimum value of a given fitness function. However, as explained in section 1, different from a pure mathematic problem, the simulation is just such a problem for which it is hard to formulate the relationship between inputs and outputs. Thus the objective function has some special character, e.g, non-convex, non-differentiable, computationally expensive. There are also some optimization methods for finding the minimum of a function of several variables without calculating derivatives. For example, Powell's method [19], which is an algorithm proposed by Michael J. D. Powell for finding a local minimum of a function. The function need not be differentiable, and no derivatives are taken. However, due to the nature of Powell's method, it is almost impossible to parallelize (parallel asynchronous versions [20] have strict condition to objective function). Since each of the fitness function evaluations needs considerable computation time, Powell's method results in very long computation time. According to our tests, it takes around 50 hours to find the closest local minimum point with a given initial value. It is fairly unacceptable for our calibration because it needs to find a considerable number of local minimum points.

Given this, a parallel optimization method is crucial for our requirement. The APPSPACK [9–11], developed by Sandia National Laboratories, implements an asynchronous parallel pattern search method that has been specifically designed for problems characterized by expensive function evaluations. The framework enables parallel operations using Message Passing Interface (MPI), and allows multiple solvers to run simultaneously and interact to find solution points. While considering our practical requirements and initial experiments, further optimization could still be conducted to speed up the calibration process. Given that the parameters to be calibrated represent the behavior of a practical agent, it is reasonable to assume that slight changes to parameters would not lead to a big difference in outputs. Considering that searching for a local minimum is computationally expensive (hours for one process), we cached the initial values by Monte Carlo, as well as the local minimum found by APPSPACK, as a pair (initial-optimum pair) to collection $C_p = \{(init, opt)_i\}$, thus when Monte Carlo generates a new set of initial values for finding other local minimum points, we firstly check the distance (d) between the new initial value and each of the initial values in collection C_p (as shown in Figure 1, the *Inquire Cache* step). The process is explained as follows:

$$if \quad \exists P^\circ \in C_p : \quad d = \sqrt{\sum_{i=1}^{n} |P_i^* - P_i^\circ|^2 / n} < \varepsilon \quad then : \quad f(P^*) := f(P^\circ) \tag{1}$$

Where, P° is the initial value set of one pair (initial-optimum) in collection C_p. P^* is the new initial value generated by the Monte Carlo method, n is the number of parameters in p_i, and ε is the tolerance. Therefore, as shown in procedure (1), if the new initial value set is close to any of the solved pair (overlapped), it will be discarded and call Monte Carlo to generate a new initial set. If there are considerable number of overlapped initial value sets found (searching space is well covered), k_1 in Figure 1 should be considered as reduced. This mechanism could avoid some duplicated optimization, especially in small search-space. A similar cache mechanism is also applied for fitness function evaluation (each one takes around 15 minutes), all the scenarios (a set of parameter values) to fitness pair (scenario - fitness pair) among all the optimization processes (which start with different initial value) were cached to a collection $C_s = \{(scenario, fitness)_k\}$. Thus, for a new scenario created by APPSPACK, procedure (1) (in C_s instead of C_p) is performed before invoking simulation. If the new scenario is close to any of the scenarios that have previously been evaluated,, then the function returns the fitness directly, thus no simulation need to be invoked. Since there are several repetitions for one evaluation process, and generally there are hundreds of evaluations per each optimization,

and many independent optimization processes needed for the calibration, this global cache mechanism could save considerable time. The experiments showed that around 10 % of fitness function evaluations were from cached value.

4. A case study

Typical EDs have common interacting elements such as doctors (physicians), nurses, technicians, receptionists, beds, medical devices that are interconnected via flows of patients, information and processes (registration, triage, diagnostic, discharge). This section gives the brief introduction of the system as well as the general model. Firstly, subsection 4.1 gives a brief introduction of the system and model, then subsection 4.2 describes the parameters which are impossible to obtain from real data and need to be calibrated. Then, subsection 4.3 gives the fitness function and subsection 4.4 describes the design of the experiment for simulation based optimization. At the end, the calibration results and discussion are given in subsection 4.5.

4.1. General process and model overview

As shown in Figure 2, typically, a patient enters the ED through one of two ways: by themselves or by ambulance. Upon arrival, walk-in patients need to walk to the registration window, briefly give their personal information to the registration staff. After that, they have to stay in a waiting room until triage. Once the information system assigns a triage nurse to the patient, they will go to the corresponding triage box and interact with the nurse. Triage consists of a brief assessment of the patient's body condition and an acuity level will be assigned to the patient according to their severity. Then, patients will wait in the second waiting room before entering the diagnosis & treatment area. For those patients who arrive by ambulance, they are registered and triaged in the ambulance, and thus go to the second waiting room directly. The Spanish scale of triage is very similar to the worldwide Canadian Emergency Department Triage and Acuity Scale [21, 22]. The scale consists of 5 levels, with 1 being the most critical (resuscitation), and 5 being the least critical (non-urgent). The triage process also determines the order and priority with which the patient will be attended and the treatment area where they will be treated. The registration and triage service are first-come first-served (FCFS) for all the patients, whereas entering the diagnosis & treatment area is acuity-level-dependent FCFS (patients with acuity level 1 have the highest priority).

With regard to the treatment area, as shown in Figure 2, in most Spanish EDs there are two treatment areas (labeled as A and B in this study) which operate independently to provide a diagnosis & treatment service. Area A is for those patients with acuity levels 1, 2 and 3, while area B is a dedicated stream of resources to process lower acuity patients with levels 4 and 5 more quickly. Area A is made up of careboxes, which is a small room that contains essential medical equipment and supplies that could be used for patients' treatment. Patients attended in area A will stay in their own carebox throughout the diagnosis & treatment phase, and any transporting should be done by auxiliary staff. In area B, there are several attention boxes in which doctors and nurses interact with patients, and a large waiting room in which all patients will remain while not having interaction with the ED staff. Note that the doctors and nurses are specified for different areas, their behavior is different, but medical image test-room and laboratory testing services are shared by area A and B.

In the diagnosis & treatment phase, once the patient has got a free space in the treatment area, the doctor will have an interaction with the patient, then the doctor makes one of the following decisions: (1) a patient needs to receive an imaging test (e.g., X-ray, B ultrasound); (2) assign laboratory tests (e.g., blood test, urinalysis); (3) discharge the patient and; (4) make out a prescription.. If testing was assigned, and when the results become available, the patient needs to have an interaction with the same doctor who conducted the consultation in order to receive a reassessment with their test results. Notably, as shown in Figure 2, some patients need to repeat the consultation-test-reassessment/treatment more than once. In summary, as marked by circled number in Figure 2, there are 8 different types of service (provided by different providers). In previous studies [23–25], the ED was modeled as a pure spatial agent-based model. It is formed entirely from the rules governing the behavior of the individual agents which populate the system, no higher-level behavior is modeled. Thus, the system behavior emerges as a result of micro-level actions and interactions.

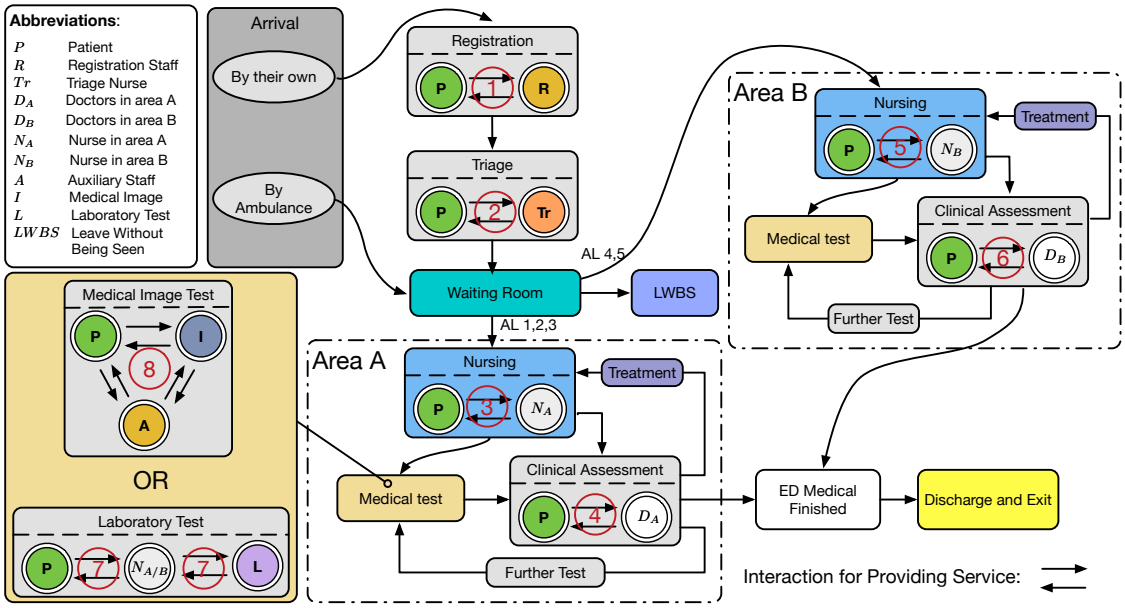


Figure 2: Diagram of patient flow through the emergency departments. Eight service processes, marked by the circled number, drive all aspects of patient flow. Most of the services are interdependent, the duration of service is different for each service. Note: Area A and area B are designed for urgent and non-urgent patients separately, they have different groups of staff and work independently.

The full model has been implemented in the NetLogo [26] simulation environment, which is an agent-based programming language and an integrated modeling environment. The work proposed in this article, on calibrating the general model to imitate a real ED, was challenged by the fact that the parameters for characterizing the service-time distributions (eight in total marked by circled number in Figure 2) are not directly obtainable from historical data or a real system.

4.2. Model parameters

In this study, Hospital of Sabadell in Catalonia, is the target system to imitate. It is a university tertiary level hospital in Spain that provides care service to a catchment area of 500,000 people, and attends more than 160,000 patients per year in the ED. In order to calibrate our general model to imitate the Hospital of Sabadell, we requested 12 months (Jan. 1st, 2014 - Dec. 31st, 2014) historical operation data from the information system's database. The missing values and invalid records have been carefully handled. Taking into consideration that August is holiday period, lots of people go on vacation, accordingly the configuration of ED is different (e.g., fewer staff or fewer senior staff), so August's operation records are discarded for this calibration study.

The ABM requires numerous parameters to characterize the behavior and features of each agent. Some of them can be retrieved directly from actual operation data of the target ED system, such as patients' features, the number of medical testing, the number of treatment processes and the number of doctor interactions with one patient. However, the service time information was not recorded by the information system (outside the scope of an information system). Thus, the parameter for the entire service-time models could not be determined directly with the real data. As illustrated in Figure 2, there are 8 service processes (marked by a circled number), all the service is carried out by interacting between agents. According to the research findings in queue theory [27], exponential distribution is typically used to make mathematically simplifying assumptions. Given this and the empirical data from ED staff, an exponential distribution was used to fit the duration of each type of service, but the parameters for these distributions should be calibrated in accordance with the target system.

More specifically, the service time is defined as the interval actually spent on receiving service (i.e., the time differences when the services started and ended). Note that in the agent-based ED model, the duration of the service mentioned in this article only represents the time spent on actually interacting, waiting time is excluded because it is an emergent property of the system. In principle, the time for a medical imaging test is composed of two parts, the interaction between patient and the test-room technicians, and time for processing test results. Given that the second part is determinable, the key is to calibrate the duration for the interaction. Similar to a laboratory test, which is composed of two parts, samples taken by a nurse and analyzing samples by machine. The second part is easy to obtain from the machine's specification, so only the duration of interaction for taking sample needs calibration.

Accordingly, we have the model input (patient arrival and their features), output (systemic performance indicator such as length of stay), and part of the model parameters retrieved directly from real data. With respect to the unknown parameters, empirical information such as boundary constraints and typical value can be obtained from experienced staff. Although the empirical information is not accurate, it can dramatically reduce search space-size. Table 1 lists all the parameters to be calibrated, as well as their boundary constraints. Therefore, the task is to search for an optimum set of parameters which can lead to good (acceptable) fitness between the simulation results and actual data.

Table 1: The parameters to be calibrated for the general agent-based model of emergency departments, in order to imitate the Hospital of Sabadell's emergency department. Note: **LB** and **UB** denotes Lower and Upper Boundary respectively, **TV** represents the Typical Value; all the units of time are in minutes. The **Identity** column corresponds to the circled numbers in Figure 2 denote the type of service.

Identity	**Notation**	**Description**	**LB**	**UB**	**TV**
1	$T_{service}^{register}$	the parameter for registration service-time distribution model.	2	15	5
2	$T_{service}^{triage}$	the parameter for triage service-time distribution model.	5	20	10
3	$T_{service}^{nurseA}$	the average duration of service of nurses in area A.	8	30	16
4	$T_{service}^{doctorA}$	the average duration of service of doctors in area A.	8	30	18
5	$T_{service}^{nurseB}$	the average duration of service of nurses in area B.	5	20	12
6	$T_{service}^{doctorB}$	the average duration of service of doctors in area B.	5	20	15
7	$T_{service}^{imaging}$	the average duration for taking medical imaging.	20	40	25
8	$T_{service}^{lab}$	the average duration for taking laboratory test sample.	10	30	15

In summary, due to data scarcity, although the distribution of specific service duration cannot be fitted by such standard techniques as maximum likelihood estimation, we had some other time stamps which enable us to derive an indirect approach to estimate the service-time distribution parameters.

4.3. Fitness function

In view of the above–mentioned facts, a proper method has to be applied to measure the similarity between actual LoS distribution and the simulated one. It is about comparing statistical characteristics of empirical data against emergent behavior of simulation models. In probability theory and statistics, the Jensen–Shannon Divergence (JSD) is a popular method of measuring the similarity between two probability distributions [28–30]. Considering that patients in ED are classified in five categories (acuity level, also known as emergency severity index) according to their severity. Patients with different acuity levels have different routes and priority in receiving service. Their LoS are quite different on average. Accordingly, it is more reasonable to evaluate patients' LoS separately due to their acuity level. Moreover, the number of patients with different acuity level is quite different, it is about 1 %, 8 %, 32 %, 44 % and 15 % respectively from acuity level 1 to 5. According to the law of large numbers, when sample size is not big enough, the statistical information would be less accurate. Given this, as defined in Equation 2, we used a weighted

average to calculate the overall fitness with JSD of the 5 categories of patients. Proper weights could be determined by sample size and the standard deviation of actual LoS.

$$f_{fitness} = \sum_{j=1}^{5} W_j D_{JS}^j \tag{2}$$

$$D_{JS}^j = \frac{1}{2} D_{KL}\left(P||Q\right) + \frac{1}{2} D_{KL}\left(Q||P\right) \tag{3}$$

Where, D_{JS}^j represents the Jensen–Shannon Divergence (JSD) similarity on LoS of patients with acuity level j, W_j is the weights according to patient category (acuity level) and $\sum_{j=1}^{5} W_j = 5$ (there are five patient categories), and D_{KL} denotes the Kullback–Leibler divergence (D_{KL}), which is defined as [28–30]:

$$D_{KL}\left(P||Q\right) = \sum_{i=1}^{n} P\left(i\right) \log_2 \frac{P\left(i\right)}{Q\left(i\right)}, \quad D_{KL}\left(Q||P\right) = \sum_{i=1}^{n} Q\left(i\right) \log_2 \frac{Q\left(i\right)}{P\left(i\right)} \tag{4}$$

Where, $Q(i)$ is the frequency/probability of LoS located in ith interval extract from simulation results, and $P(i)$ denotes the same information extracted from real data. Having shown that, the range of $f_{fitness}$ function value will be 0.0 to 5.0, The lower it is, the closer the difference between simulation and actual will be.

As described in subsection 4.2, parameter constraint is defined by boundaries, although each of the parameters is guaranteed to fulfill the boundaries constraint, the combination of parameters may become unreasonable for the model. This case may occur either in the initial value set generated by the Monte Carlo method, or an evaluation scenario requested by the optimization solver. According to our primary experiments, some parameter sets created by optimization algorithm may cause ED saturation, i.e., patients waiting in any of the waiting rooms increases day-by-day. For example, the number of patients waiting to enter the treatment area is greater than daily arrival. These scenarios cannot result in good fitness because it is not a valid case. Since the complexity of an agent-based model is proportional to the number of agents in the simulation environment, system saturation will result in much longer simulation time. Give this, when the system is saturated, it is better to terminate the simulation evaluation and return the worst fitness evaluation as a penalty.

Furthermore, the patient leaving-without-being-seen (LWBS) is a common phenomenon and a crucial metric to EDs, so it has been carefully considered as a possible decision patients may take in the model [31–33] . As the real data does not include the LWBS records, the final tuned simulator should not have patients who LWBS (equivalent to those patients not going to ED). However, our primary results showed that some of the parameters set (either generated by Monte Carlo or created by optimization solver) resulted in LWBS. Instead of discarding the evaluations that have LWBS, which may result in a lot of failure in optimization and waste lots of computing time, we added LWBS to the objective function as a part of the penalty (i.e., the optimization solver should be allowed to make mistakes). Our final experiments indicated the effectiveness of considering LWBS in fitness function. As shown in Figure 3, most of the initial values that lead to LWBS could converge in less than ten iterations. In summary, the final fitness function could be defined as:

$$F_{fitness}\left(P\right) = \begin{cases} f_{fitness}(P) + \lambda R_{lwbs} & (simulation\ succeed) \\ F_{max} & (system\ saturated) \end{cases} \tag{5}$$

Where, $P = \{p_1, p_2, \cdots, p_8\}$ denotes a parameter set from the Monte Carlo method or the optimization solver, R_{lwbs} is the ratio of patients LWBS (range from 0 to 1.0), λ is an adjustable parameter which represents the weight of LWBS. F_{max} is the maximum penalty to the solver, which is the maximum of $F_{fitness}$ in the first case (simulation succeed). Given this, if we set λ as 5.0, that is to say, the D_{JS} similarity and LWBS have the same weight on the fitness evaluation, the value of $F_{fitness}$ will be between 0 to 10. The lower it is, the closer it will be to actual data.

4.4. Design of experiment

As illustrated in Figure 1, the real dataset was divided into three subsets for training, testing and validating separately. To this end, the 11-month historical data from the ED information system database (Jan. - Dec. 2014, excluding August) has been randomly divided into three parts. More specifically, six months for training (training set), three months for testing (test set), and two months for validation.

Considering that the patients' LoS are statistics on patients who attended the ED. Due to the statistical nature of this model, the sample size should be guaranteed in order to provide reliable LoS. The minimum number of patients for retrieving LoS depends on deviation of LoS, confidence interval as well as margin of error, and could be determined by Chebyshev's inequality [34]. Therefore, multiple runs must be conducted for each scenario in order to reduce stochastic variability and average performance metrics will be used for evaluating the fitness by Equation 5. More specifically, the number of simulation replications are determined by deviation of LoS from the real dataset and the simulation time. Namely, shorter time simulation will require more replications in order to meet the sample size requirements. In this study, according to the statistic characteristics of LoS in real dataset, 4 random seeded runs were performed for each scenario in training dataset, 8 replications were performed for each scenario on testing dataset, and 12 replications for validation dataset.

The calibration was carried out on an 8-node cluster with total number of 512 AMD Opteron^{TM} Processor 6262 HE cores, and 2TB RAM. All the nodes works in master/worker way, i.e., each one of the node (worker) runs the parallel version of APPSPACK to find the local minimum start from an initial value given by the master. The APPSPACK evaluators, which takes input (the parameter set) and returns fitness, were implemented with Python programming language. In the evaluator, the NetLogo controlling API (which comes with NetLogo.jar from the released version) was used to invoke and control NetLogo by another Java program running on the Java Virtual Machine. That is, for one fitness evaluation, the Python program will first read the value of variables and invoke several processes (the same as the number of repetitions) in order to evaluate fitness with the same parameter but different random seeds, then each of the processes will call a Java program via system call with parameters as arguments. In the last step, the Java program will initiate the model in NetLogo via NetLogo controlling API and start the simulation. When all the simulations with the same parameter have finished, the program will return to Python, and a post-processing function will be called to analyze the system metrics in order to calculate the fitness value (via Equation 5).

4.5. Results and discussion

Use Equation 5 as fitness function, the iterations of optimization on training dataset with different initial value is shown in Figure 3.

It is clear to see from Figure 3 that different initial values resulted in a different number of iterations. Most optimums (the converged fitness values) are in the same level (i.e., no significant global minimum). Some initial values have caused high LWBS, i.e., their initial fitness is greater than 5 (maximum of $f_{fitness}$ part in Equation 5 is 5.0), and drop to normal after several iterations. Most optimization processes are completed in less than 20 iterations. To analyze the location of local optimum points we found, Figure 4a shows the distribution of Euclidean distance between optimums points (there are $k_1 (k_1 - 1)/2$ distance, where k_1 is the total number of local optimum points found, the same k_1 as it in Figure 1).

From Figure 4a, it is clear that most optimum parameter sets (note that in order to make all the parameter values for APPSPACK on a similar scale, here the parameter values represent the ratio to the typical values in Table 1) are far from each other (due to the initial value control by Equation 1), while there are some optimums that are carried out by different initial values, converged to the same point (distance could not be zero because of the random nature of the simulator and the tolerance setting in APPSPACK). According to the search scheme of APPSPACK [9], each iteration requires many fitness function evaluations in several directions, the number of fitness evaluations has direct influence on optimization time. Figure 4b shows the distribution of the number of fitness evaluations. It is worth noting that, in each function evaluation, there are several replications on simulation with different random seeds, i.e., 4, 8, 12 for training, testing and validation separately. In this study, when $k_1 = 30, k_2 = 10, k_3 = 5, k_4 = 1$, the total time taken on the calibration is about 60 hours with the above-mentioned cluster.

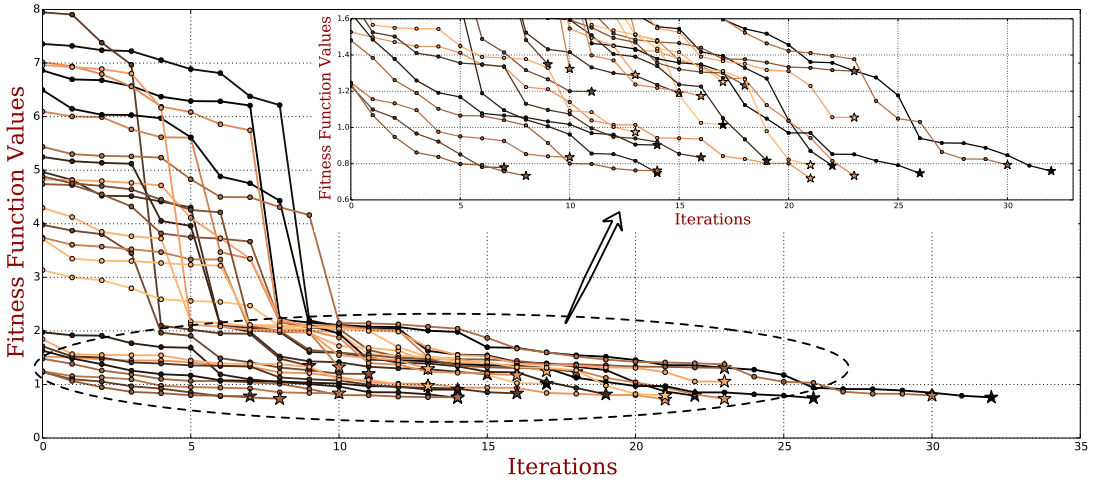
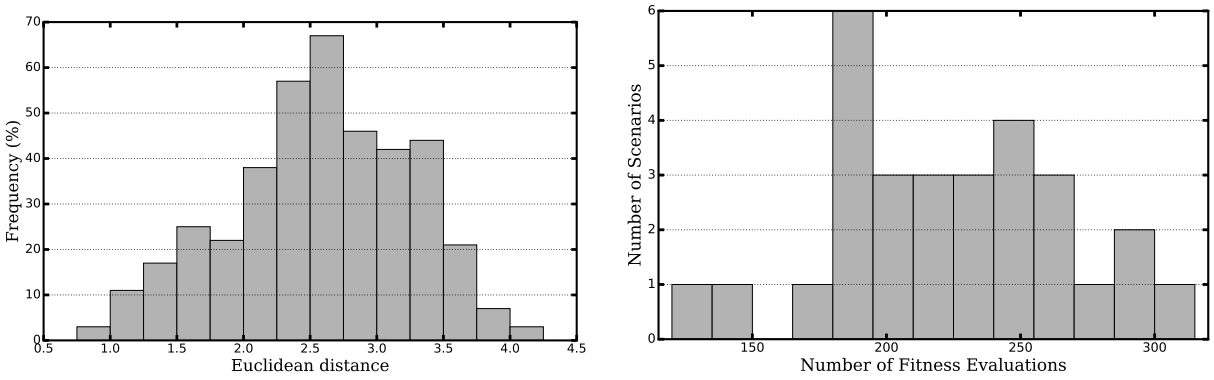


Figure 3: Fitness optimization on training dataset with different initial value, fitness values versus iterations. One broken line represents one optimization process with a given starting point from boundary constrained Monte Carlo.

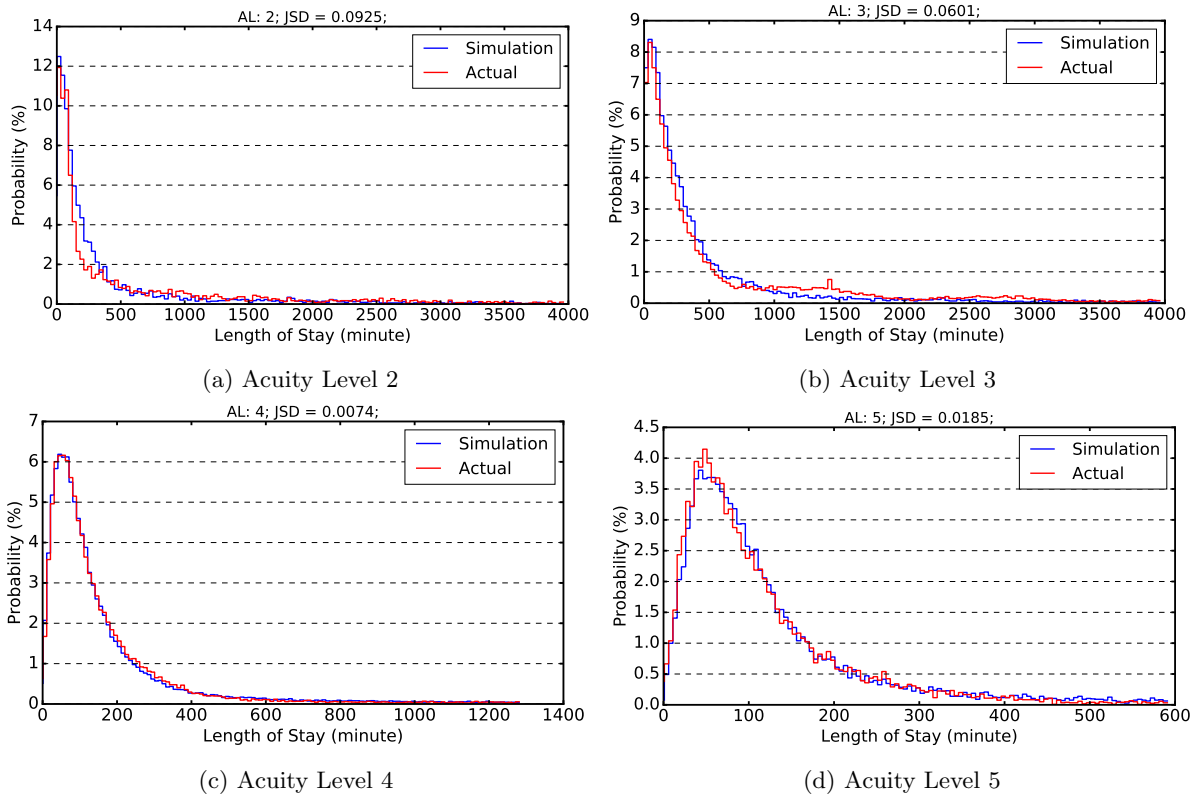


(a) Euclidean distance between optimum points.

(b) Number of fitness evaluation distribution.

Figure 4: Training process analysis. The distribution of the number of fitness evaluations needed in finding local minimum points starting from different initial values, and the distribution analysis of distance between optimal points.

By following the process illustrated in Figure 1, one set of parameter values was selected manually from the k_3 candidates. With the selected parameter set and input (patient arrival) from the validation dataset, the comparison (actual data versus simulation) of patients' LoS distribution, classified by patient's acuity level, is illustrated in Figure 5. Considering that the validation dataset is composed of two-month's real data and, there are very few patients (less than 1 %, about 160 patients in two months) triaged with acuity level 1, the sample size is not enough for statistical comparison, thus the LoS distribution of patients with acuity level 1 was not shown in Figure 5.



(a) Acuity Level 2

(b) Acuity Level 3

(c) Acuity Level 4

(d) Acuity Level 5

Figure 5: The comparison of model prediction results (patient length of stay distribution) on the validation dataset. Results about patients with acuity level 1 are not illustrated here because very few patients (less than 1 %) attend to ED with acuity level 1, the sample size (in two months) is not enough for statistical comparison. The JSD denotes Jensen-Shannon Divergence. Note that the statistical interval widths are: 30 minutes for acuity level 2 and 3; 10 and 5 minutes for acuity level 4 and 5 respectively.

Simulation results in Figure 5 demonstrate that the proposed framework is effective to calibrate the model parameters. As a result of the small number of patients attending with acuity level 2, the fitness (the D^j_{JS} in Equation 3) of patients with acuity level 2 (Figure 5a, $D^2_{JS} = 0.0925$) is not as good as the others. Since the calibration process happens only once in the simulation, 60 hours is acceptable and further speedup can be reached via executing on clusters with more computing nodes.

5. Conclusion and Future Work

An Emergency Department (ED) is a complex, stochastic environment, which has time-dependent behavior. Advances in computational technology give us the ability to simulate complex models and analyze massive datasets. Given this, simulation has become an effective method to improve policies on operational, tactical and strategic decisions for EDs. However, the difficulty in collecting reliable and complete data can subsequently lead to invalid simulation results. To this end, this paper proposed a systemic method

to calibrate and validate a general model to imitate an actual ED under data scarcity (missing duration of service). Our final results indicated that the proposed approach can find the model parameters accurately within an acceptable time frame. With the parameter value we found, the general agent-based model of EDs can carry out accurate predictions. Although our work was focused on calibrating an ED model, we are confident that the proposed method could also make some contribution to calibrating other computationally expensive simulation models.

There are a number of limitations to our study, including the use of exponential distribution for fitting all the duration of service. Although it was commonly used in the conventional queue theory method, further research should be carried out to consider the features of service type in more detail. Another limitation is the selection of system KPIs for calculating fitness. In our method, we only considered two indicators, i.e., patients' length of stay and leave-without-being-seen. Although both are commonly used in emergency healthcare literature, further indicators such as door-to-doctor time and patients' length of waiting time should be investigated in future improvements. Furthermore, as the proposed method has only been tested in one institution though no institution-specific assumption has been made, one of our future studies will apply the method on another ED.

In summary, the proposed systematic method has been proved to be able to find the parameters for fitting the duration of service, with which the simulated results and the actual data were consistent. The duration of healthcare staff's service time is among the most common missing pieces of information because it is out of the scope of the information system. Moreover, an automatic calibration tool released with a general ED model is promising for promoting the application of simulation in ED studies. This tool will enable the simulation users, e.g., ED managers, to calibrate parameters for their own ED system without the involvement of model developers.

References

[1] P. M. Sloot, R. Quax, Information processing as a paradigm to model and simulate complex systems, Journal of Computational Science 3 (5) (2012) 247 – 249, advanced Computing Solutions for Health Care and Medicine. doi:10.1016/j.jocs.2012.07.001.

[2] S. Robinson, Simulation: The Practice of Model Development and Use, Vol. 67, Jhon Wiley & Sons, 2004. doi:10.1057/palgrave.jos.4250031.

[3] N. Hoot, S. Epstein, T. Allen, S. e. a. Jones, Forecasting Emergency Department Crowding: An External, Multicenter Evaluation, Annals of Emergency Medicine 54 (4) (2009) 514–522.e19. doi:10.1016/j.annemergmed.2009.06.006.

[4] M. Wagner, W. Cai, M. H. Lees, H. Aydt, Evolving agent-based models using self-adaptive complexification, Journal of Computational Science 10 (2015) 351–359. doi:10.1016/j.jocs.2015.03.005.

[5] D. Heard, G. Dent, T. Schifeling, D. Banks, Agent-based models and microsimulation, Annual Review of Statistics and Its Application 2 (2015) 259–272. doi:10.1146/annurev-statistics-010814-020218.

[6] C. Rudin, Algorithms for interpretable machine learning, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, p. 1519.

[7] M. Gul, A. F. Guneri, A comprehensive review of emergency department simulation applications for normal and disaster conditions, Computers & Industrial Engineering 83 (2015) 327–344. doi:10.1016/j.cie.2015.02.018.

[8] M. Fehler, F. Klügl, F. Puppe, Approaches for Resolving the Dilemma between Model Structure Refinement and Parameter Calibration in Agent-Based Simulations, in: the proceeding of AAMAS 2006, 2006, pp. 120–122. doi:10.1145/1160633.1160651.

[9] G. A.Gray, T. G.Kolda, Algorithm 856: Appspack 4.0: Asynchronous parallel pattern search for derivative-free optimization, ACM Transactions on Mathematical Software 32(3) (2006) 485–507.

[10] T. G. Kolda, Revisiting asynchronous parallel pattern search for nonlinear optimization, SIAM Journal on Optimization 16(2) (2006) 563–586. doi:10.1137/040603589.

[11] J. D. Griffin, T. G. Kolda, Asynchronous parallel generating set search for linearly-constrained optimization, Technical Report, Sandia National Laboratories, Livermore, CA July 2006.

[12] T. Trucano, L. Swiler, T. Igusa, W. Oberkampf, M. Pilch, Calibration, validation, and sensitivity analysis: What's what, Reliability Engineering & System Safety 91 (1011) (2006) 1331 – 1357, the Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004)SAMO 2004The Fourth International Conference on Sensitivity Analysis of Model Output (SAMO 2004). doi:10.1016/j.ress.2005.11.031.

[13] M. Hofmann, On the Complexity of Parameter Calibration in Simulation Models, The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology 2 (4) (2005) 217–226. doi:10.1177/154851290500200405.

[14] J. Zhong, N. Hu, W. Cai, M. Lees, L. Luo, Density-based evolutionary framework for crowd model calibration, Journal of Computational Science 6 (2015) 11–22. doi:10.1016/j.jocs.2014.09.002.

[15] J. Zhong, W. Cai, Differential evolution with sensitivity analysis and the Powell's method for crowd model calibration, Journal of Computational Science 9 (2015) 26–32. doi:10.1016/j.jocs.2015.04.013.

[16] M. Fehler, F. Klügl, F. Puppe, Techniques for analysis and calibration of multi-agent simulations, in: M.-P. Gleizes, A. Omicini, F. Zambonelli (Eds.), Engineering Societies in the Agents World V, Vol. 3451 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 305–321. doi:10.1007/11423355_22.
[17] S. Welch, J. Augustine, C. A. Camargo, C. Reese, Emergency Department Performance Measures and Benchmarking Summit, Academic Emergency Medicine 13 (10) (2006) 1074–1080. doi:10.1197/j.aem.2006.05.026.
[18] S. J. Welch, B. R. Asplin, S. Stone-Griffith, S. J. Davidson, J. Augustine, J. Schuur, Emergency department operational metrics, measures and definitions: Results of the second performance measures and benchmarking summit, Annals of Emergency Medicine 58 (1) (2011) 33–40. doi:10.1016/j.annemergmed.2010.08.040.
[19] M. J. D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, The Computer Journal 7 (2) (1964) 155–162. doi:10.1093/comjnl/7.2.155.
[20] C. Sutti, Local and global optimization by parallel algorithms for mimd systems, Annals of Operations Research 1 (2) (1984) 151–164. doi:10.1007/BF01876145.
[21] R. S. Bermejo, C. C. Fadrique, B. R. Fraile, E. F. Centeno, S. P. Cueva, E. María, Triage in Spanish hospitals, Emergencias 25 (1) (2013) 66–70.
[22] M. J. Bullard, B. Unger, J. Spence, E. Grafstein, Revisions to the Canadian emergency department triage and acuity scale (CTAS) adult guidelines, Cjem 10 (2) (2008) 136–151.
[23] Z. Liu, E. Cabrera, D. Rexachs, E. Luque, A Generalized Agent-Based Model to Simulate Emergency Departments, in: The Sixth International Conference on Advances in System Simulation, IARIA, 2014, Nice, France, 2014, pp. 65–70.
[24] Z. Liu, E. Cabrera, M. Taboada, F. Epelde, D. Rexachs, E. Luque, Quantitative evaluation of decision effects in the management of emergency department problems, in: International Conference on Computational Science, 2015, pp. 433–442. doi:10.1016/j.procs.2015.05.265.
[25] Z. Liu, E. Cabrera, D. Rexachs, F. Epelde, E. Luque, Simulating the Micro-level Behavior of Emergency Departments for Macro-level Features Prediction, in: Proceedings of the 2015 Winter Simulation Conference, 2015, pp. 171–182. doi:10.1109/WSC.2015.7408162.
[26] U. Wilensky, NetLogo. http://ccl.northwestern.edu/netlogo/., Center for Connected Learning and ComputerBased Modeling Northwestern University Evanston IL 2009 (26.02.2009) (1999) Evanston, IL.
[27] J. L. Devore, Probability and Statistics for Engineering and the Sciences, Duxbury Press, 2011.
[28] F. Osterreicher, I. Vajda, A new class of metric divergences on probability spaces and its applicability in statistics, Annals of the Institute of Statistical Mathematics 55 (3) (2003) 639–653. doi:10.1007/BF02517812.
[29] D. Endres, J. Schindelin, A new metric for probability distributions, IEEE Transactions on Information Theory 49 (7) (2003) 1858–1860. doi:10.1109/TIT.2003.813506.
[30] J. Lin, Divergence measures based on the shannon entropy, Information Theory, IEEE Transactions on 37 (1) (1991) 145–151. doi:10.1109/18.61115.
[31] R. Ding, M. L. McCarthy, G. Li, T. D. Kirsch, J. J. Jung, G. D. Kelen, Patients Who Leave Without Being Seen: Their Characteristics and History of Emergency Department Use, Annals of Emergency Medicine 48 (6) (2006) 686–693. doi:10.1016/j.annemergmed.2006.05.022.
[32] M. Johnson, S. Myers, J. Wineholt, M. Pollack, A. L. Kusmiesz, Patients who leave the emergency department without being seen, Journal of Emergency Nursing 35 (2) (2009) 105 – 108. doi:10.1016/j.jen.2008.05.006.
[33] M. Kennedy, C. E. MacBean, C. Brand, V. Sundararajan, D. McD Taylor, Review article: Leaving the emergency department without being seen, Emergency medicine Australasia 20 (4) (2008) 306–313. doi:10.1111/j.1742-6723.2008.01103.x.
[34] W. J. Stewart, Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling, Princeton University Press, 2009.