Universitat Autònoma de Barcelona

Departament d'Arquitectura de Computadors i Sistemes Operatius

Màster en Computació d'altes prestacions

Agent Based Simulation to Optimise Emergency Departments

Memoria del trabajo de ”Iniciación a la Investigación. Trabajo de Fin de Máster” del “Máster en Computació d'altes prestacions”, realizada por Eduardo César Cabrera Flores, bajo la dirección de Emilio Luque Fadón. Presentada en la Escuela de Ingeniería (Departamento de Arquitectura de Computadores y Sistemas Operativos)

Julio, 2010

Iniciación a la investigación. Trabajo de fin de máster

Máster en Computación de Altas Prestaciones.

Título Agent Based Simulation to Optimise Emergency Departments

Realizada por Eduardo César Cabrera Flores
en la Escuela de Ingeniería, en el Departamento Arquitectura de Computadores y Sistemas Operativos

Dirigida por: Dr. Emilio Luque Fadón

Firmado

Director Estudiante

Dedication.

This work is dedicated to my mother and father.

Acknowledgments.

To my siblings.

To my family.

To Emilio Luque and Dolores Rexachs for their support and advices.

To all my partners of CAOS.

Abstract

Nowadays, many of the health care systems are large and complex environments and quite dynamic, specifically Emergency Departments, **EDs**. It is opened and working 24 hours per day throughout the year with limited resources, whereas it is overcrowded. Thus, is mandatory to simulate **EDs** to improve qualitatively and quantitatively their performance. This improvement can be achieved modelling and simulating **EDs** using Agent-Based Model, **ABM** and optimising many different staff scenarios.

This work optimises the staff configuration of an **ED**. In order to do optimisation, objective functions to minimise or maximise have to be set. One of those objective functions is to find the best or optimum staff configuration that minimise patient waiting time. The staff configuration comprises: doctors, triage nurses, and admissions, the amount and sort of them. Staff configuration is a combinatorial problem, that can take a lot of time to be solved.

HPC is used to run the experiments, and encouraging results were obtained. However, even with the basic **ED** used in this work the search space is very large, thus, when the problem size increases, it is going to need more resources of processing in order to obtain results in an acceptable time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction.

1.1 Context.

It is stated that modelling and simulation is the third way of doing science [1], complementary to observation and direct experimentation, first and second way, respectively. Usually, classical sciences as physics, chemistry, and astronomy use the three of them, but specially computer simulation when collected data and experimentation become a challenge to calculate. For the last 20 years, simulation is also applied, succesfully, in other disciplines as economics, ecology, psychology, anthropology, education, health care and biology, amongst other social sciences. At the present time, modelling and simulation of different kind of complex social systems, like the fields previously cited, is quite common and promising research. By complex systems, it is mean systems that exhibit nontrivial emergent and self-organising behaviour [2].

Health care systems, as well of many social sciences, lack of standard models that characterised them. Hospitals, as core member of health care systems, are made of many independent distributed complex departments [3], one of these complex units is Emergency Department (**ED**), that could be the most dynamic department of hospitals, and so can be modelled and simulated isolated.

Due to the absence of any formal description for **EDs**, alternative methods must be used to characterise them. Therefore, Agent-Based Model (**ABM**) or Individual-oriented Model (**IoM**) is used, since this framework describes the dynamic of systems in which agent behaviour is complex, non-linear, the combined interaction of all the agents can create rich emergent behaviour and agents show memory [4].

1.1.1 Computational science.

Computational science is neither computer science, mathematics, engineering, social science, nor a discipline of humanities. It is a blend, the intersection between applied mathematics, computer science and application sciences as it is shown in Figure 1.1. It is a field that concentrate on the effective use of computer software, hardware and mathematics to solve real problems. It is a useful concept when it is desirable to distinguish the more pragmatic aspects of computing from computer science, theory concepts, from computing or engineering, design and construction of computers. Mathematics set up the algorithm or the formal model, while the physical model or application is provided by the applied field, which is the science objective of study, that could be any as physics or chemistry, and, finally, computer science adds the software i.e., the implementation of the mathematical model, and hardware where the simulation will take place.

Another definition according to *The Society of Industrial and Applied Mathematics (SIAM)*:

> Computational science and engineering is a rapidly growing multidisciplinary area with connections to the sciences, engineering, mathematics and computer science. It focuses on the development of problem-solving methodologies and robust tools for the solution of scientific and engineering problems.

Thus, in the broad sense, it is *Science, done computationally.*

This work is an interdisciplinary one, since there is a relationship between health science and health care systems, computer science, engineering and sociology. It belongs to Computational Science applications in *Individual Oriented Behaviour*, specifically. Thus, computers are used to simulate the model, which is not a mathematical or standard one of an **ED**, as it is said above, in order to optimise its staff configuration to enhance its performance.

1.2 Motivation.

Nowadays, many of the health care systems are large and complex environments and quite dynamic, specifically **EDs**. The **ED** is a *sui generis* unit of hospitals. On the one hand, it is opened and working 24 hours per day through-

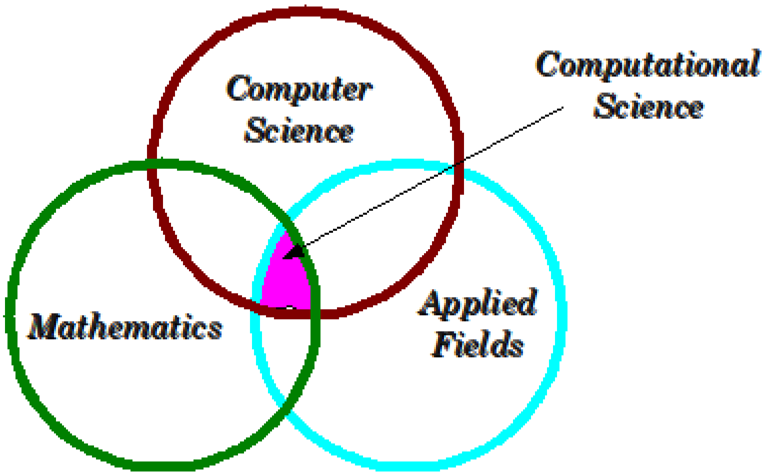


Figure 1.1: Computational Science as a multidisciplinary field.

out the year with limited resources, specially with the present financial crisis, when there are several budget reductions that could compromise health care systems, but on the other hand, is under a huge and growing demand of services, i.e., it is overcrowded. Such critical service must be satisfied with the best quality and effort. **Emergency Department** is supposed to be the unit where severe illness and injury, emergent cases, is handle, but due to the high demand of services, it is no longer anymore. As a matter of fact, **EDs** has become a unit where converge urgent, nonurgent, and severe cases, which decrease the amount of time, quality and resources given to the patients. Therefore, is mandatory to simulate **EDs** to improve qualitatively and quantitatively the performance of such crucial department, because health is one of the most appraised gift to human life. Such improvement can be achieved simulating **EDs** using **ABM** and optimising many different staff scenarios, that includes different amount and sort of doctors, nurses, and admissions, amongst others, with huge demand of heterogeneous service, in order to find the best or optimum staff configuration.

Moreover, High Performance Computing **HPC** has been associated and utilised mainly in classical sciences as physics, astronomy and chemistry or hard difficult engineering problems, but now social sciences are becoming to use it lately. The systems modelled by these sciences are quite complex and demand huge amount of data space, and to preserve the data new file systems must be develop; furthermore, the simulation of such systems has long run-time on convential computers. In addition, the models and the phenomena being modeled are inherently probabilistic. Hence, social sciences and **EDs** are demanding **HPC**, in order to simulate, analyse, understand and generate knowledge.

1.3 The Problem.

An almost steady stream of patients arrives into **EDs**, specifically nonurgent or urgent cases, but also serious ones too. The latter cases are, or at least it is supposed to be, the main target of **EDs**, even though all cases have to be received and addressed. Patients can arrive either by their own or by ambulance. Moreover, there are days, periods or extreme events which modify such almost steady stream of patients and increase the demand of services that compromise the whole department and the *ad hoc* or ideal patient care. Nevertheless, patient input cannot be modified, i.e., it is a fact, even if it is steady or not.

EDs are units constituted by the place, physical resources as beds, test equipments, and, finally, but the most important, patients and their companions, and staff members, which includes nurses, doctors, and admission staff, amongst others.

Usually, patients have the following flow in the **EDs**: arrive by walk in, if no require immediate care they proceed to the admission place, whereas both of those who need immediate attention, and those that arrive by ambulance are sent directly to a treatment area, if admission staff is busy patients wait; thus, patients go to a triage area, if triage nurses are busy patients wait, again, but in another area -in this step patients are evaluated for the seriousness or acuity of their condition, and a priority level is assigned based on it; finally, patients wait for a diagnosis and treatment room and a doctor. At last, patients could be admitted into the service or discharge. All these phases depend, not only on the distribution stream of patients, but also on the configuration of the staff members. That is, the human resources of **EDs**. These human resources imply cost, because of their income, as well as the costs related to the tests that have to be taken to the patient, since the budget is a major constraint.

Optimise this staff configuration: doctors, triage nurses, and admission, which includes the sort of the members —junior or senior, less and more expert, respectively— and the amount of them —from 1 to 3 or 4— that are shown in the table 1.1 represents a combinatorial problem, that can take a lot of time to be solved. This implies that specific scenarios or configurations have to be simulated several times, changing parameters to show different probabilities, for the purpose of generate set of results from which particular effects can be conclude.

Even with this simple setting of an Emergency Department the search space is big. The search space has 4536 combinations to find out which is the best or optimum that minimise or maximise a desire index, under some restrictions. The initial indexes, or objective functions in which this work is concerned are: to find the best or the optimum that minimise patient waiting time, under certain costs, and minimise staff cost configuration, restricted to some waiting patient time.

Table 1.1: Staff configuration parameters of a basis *Emergency Department*

	Junior / Senior
Doctor	1-4
Triage nurse	1-3
Admission	1-3

1.4 Objectives.

- The first and general objective is to create a Decision Support System (*DSS*) to help Emergency Departments in order to set up strategies and management guidelines to enhance the performance of the **EDs**.

- The second and main objective of this work is to optimise the staff members of *Emergency Departments*, which includes admission staff, triage nurses, and doctors, in order to get the best or optimum solution to minimise or maximise objective functions.

1.5 Organisation of this dissertation.

This dissertation is organised as follows:

- In the next Chapter, concepts of modelling are discussed. Also the framework used Individual Oriented Behaviour or **ABM** is presented, as well as briefly the model of the *Emergency Department*. Topics about simulation and its importance are outlined, also the simulator used is presented. Finally, the related works are also presented in this Chapter.

- In Chapter 3 topics of optimisation are explained. They include definitions, numerical methods, taxonomy of optimisation approaches for single and multiple objectives. Finally, concepts and examples of metaheuristics are sketched.

- Experimental Evaluation and results are shown and discussed in Chapter 4, according to the objectives posed.

- Finally, in Chapter 5 the final conclusions and future works are stated.

Chapter 2

Individual Oriented Behavioural.

2.1 Introduction.

This framework is defined by the behaviour of the individuals, their interactions between each other, as well as the interactions between the individual and the environment. This modelling is used on complex systems that are difficult to tackle by classical or formal methods, which are unable to define such problems.

Concepts about what modelling is, its purposes, and characteristics are outlined in this chapter. The definition and vindication on using an alternative method, the Agent Based Model *ABM*, to model complex systems is presented later. The present Agent Based Model of Emergency Department *ED* is briefly shown, as well as concepts of simulation and related works are discussed.

2.2 Modelling.

It is said that models are only an abstract representation of a real system. The model can be defined as a set of assumptions or approximations about how the system works, i.e., it describes the system. It is of paramount importance remembering that the model is not the reality, but merely a human construction to help to better understand real world systems.

What are the purposes of modelling? Amongst them are the following:

- allow to study, and analyse the model instead of the real system. It is easier, faster, cheaper, and safer.
- to training or for educational aim merely.
- can generate new insights.
- testable predictions can be made.
- can help to proof the hypothesis.
- rule out a particular explanation for an experimental observation.
- can try wide-range of ideas or experimental; rather than to make the mistakes for real make them on the computer model.

Models can be either mental, that are subjective, incomplete, lack a of formal statement, e.g., ideas and concepts, or formal, which are based on rules, and are easy transmitable, for example diagrams, and planes. Models can be characterised as: *deterministic*, *stochastic*, *static*, *dynamic*, *continous*, and *discrete* [5]. This taxonomy is shown in Figure 2.1

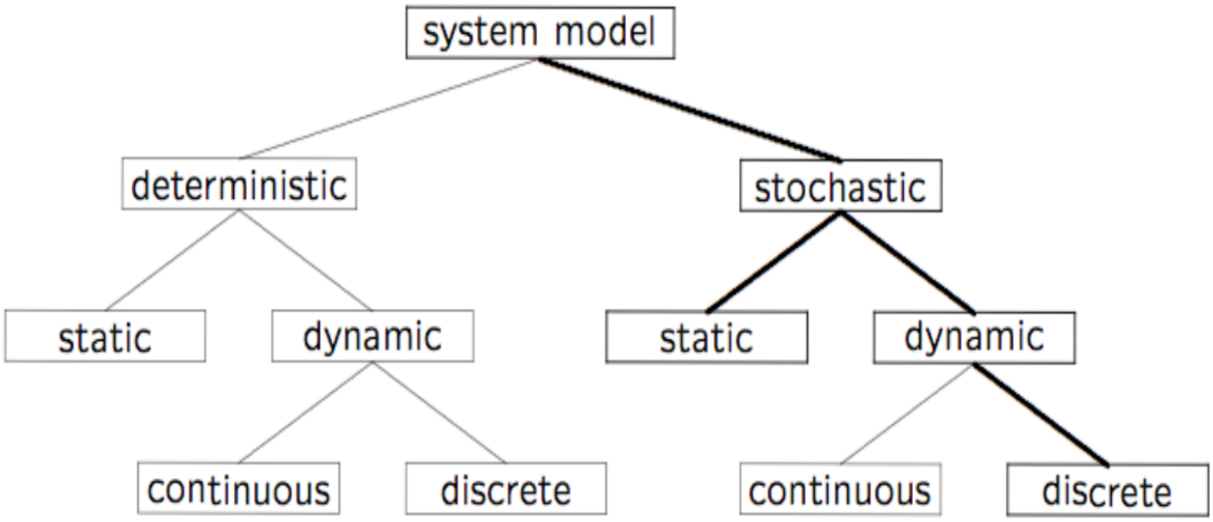


Figure 2.1: Classification of models.

The overall modelling process is usually iterative, and comprise the following methodology:

1. Determine the goals and objectives.
2. Build a *conceptual* model.
3. Convert into a *specification* model.
4. Transform into a *computational* model.
5. Verify.
6. Validate.

Before entering the modelling process, there are some questions that have to be made: are the expected savings from using the model greater than the cost of developing and implementing it? Is there enough time to develop and implement it before the recommendation is needed? Is it more easy doing an experiment on the real system than to build the model? In general, if the purposes as well as the objectives are satisfied.
However, answering previous questions is not only one the most difficult task when doing the *conceptual* model, but also it is to identify the simplifications that ought to be made without sacrificing the needed or useful accuracy of the model. If unimportant details are kept out of the model, it would be more easier to change and to use. Another, key part of this *conceptual* and abstract process is how comprehensive should the model be?

The *specification* phase comprise equations and pseducode of the *conceptual* model of the prior step; whereas the *computational* model is the numerical implementation, i.e., a computer program.

Verification implies that *computational* model must be consistent with the *specification* model. Finally, the model is validated if it is consistent with the system being analized. Moreover, if an expert cannot be able to distinguish simulation output from system output, thus, it is the right the model. Nevertheless, if the output model differs from what happened in the real world, it is necesary to recall that the model is not the reality. Hence, the model still has some flaws. As an iterative process if any phase is not satisfied it has to back to the previous steps.

Usually, models are stochastic, dynamic and discrete. They are complex [2], difficult to analyse and their results could be only valid over a range. The next section deals with these kind of models, that specially lack of a formal model.

2.3 Agent Based Model.

Although, there is no a general accepted definition, it can be said that an Agent Based Model ABM is a computational model of a heterogeneous population of agents and their interactions. The result of the micro-level interactions can produce macro-level behaviour like cooperation, segregation, and culture, amongst others. This framework describes the dynamic of the systems in which agent behaviour is complex, stochastic, non-linear, the combined interaction of all the agents can create rich emergent behaviour and shows memory [4]. ABM are fundamentally decentralised. The behaviour is defined at individual level, and the global behaviour emerges as a result of many individuals, each following its own behaviour rules. Hence, ABM is also called bottom-up modelling. An agent is a discrete entity with its own goals and behaviours. It is also autonomous, with a capability to adapt and modify its behaviours.

ABM can be a useful tool of analysis complementary to mathematics when a model is either not totally solved mathematically or apparently insoluble. This is the case for social sciences where usually there is a lack of mathematical model which defines the problem.

2.3.1 Agents Based Models in Social Sciences.

ABM is widely used in social science . The three fields in which agent-based models are most utilised are economics, social sciences, and biology [6]. *ABMs* are used in social sciences in situations where human behaviour cannot be predicted using classical methods such as qualitative or statistical analysis [7]. Human behaviour is also modelled with *ABMs* in the fields of psychology[8], epidemiology[9], and tourism planning[10], amongst a long list of others.

Before enter the discussion about the Emergency Department Model, in the next section 2.4, some suggestions about the relevance of using *ABM* [11] are listed following:

- When there is a natural representation as agents.
- When there are decisions and behaviours that can be defined discretely (with boundaries).
- When it is important that agents adapt and change their behaviours.
- When it is important that agents learn and engage in dynamic strategic behaviours.
- When it is important that agents have a dynamic relationships with other agents, and agent relationships form and dissolve.
- When it is important that agents have a spatial component to their behaviours and interactions.

2.4 The Emergency Department Model.

The Emergency Department model defined by this work is a pure Agent-Based Model, and so is formed entirely of the rules governing the behaviour of the individual agents which populate the system.

Through the information obtained during the interviews carried out with ED staff of the Hospital of Mataró and the Hospital of Sabadell, two kind of agents have been identified, these are active and passive agents. The active agents represent people and other entities that act upon their own initiative (*patients*, *companions of patients*, *admission staff*, *sanitarian technicians*, *triage and emergency nurses*, *staff emergency doctors*, and others *specialists*, and *social workers*). These are described by state machines, specifically Moore machines. A Moore machine has a single output for each state; transitions between states are specified by the input. On the other hand, passive agents represent systems that are solely reactive, such as *loudspeaker system*, *patient information system*, *pneumatic pipes*, and *central diagnostic services* (*radiology service* and *laboratories*).

Even though the model is not the aim of this work, it is sketched in the rest of this section.

2.4.1 Active agents.

The current state of an active agent is represented by a collection of state variables, known as the state vector (**T**). Each unique combination of values for these variables defines a distinct state.

Every time step the state machine moves to the next state as defined by the current state and the input vector as described below.

Table 2.1: Some active state variables, and their values.

Variable	Values	Observability
Name / Identifier	Unique per agent	I
Personal details	Gender; Medical history; Allergies; Origin	I
Location	Department entrance, Admissions, Waiting room, Triage, Consultancy room, Treatment box	E
Action	Idle, Requesting information from `<id>`, Giving information to `<id>`, Searching, Moving to `<location>`	E
Physical condition	Healthy, Hemodynamic-Constant; Barthel Index	E / I / N
Symptoms	None; Level 1 - Resuscitation; Level 2 - Emergent; Level 3 - Urgent ; Level 4 - Less Urgent; Level 5 Non Urgent	E/ I
Level of communication	Low, Medium, High	E
Level of experience (doctor)	None, Resident, Junior, Senior and Consultant	I
Level of experience (triage and emergency nurses)	None, Low, Medium, High	I
Level of experience (admissions)	None, Low, Medium, High	I

2.4.1.1 State variables.

In order for the state machine to function, all state variables must be enumerable in some manner. This may be discrete variables or variables representing continuous quantities which have had their possible values divided into ranges.

Variables also have another property, observability. A variable that is externally observable (E) indicates that any agent can discern the value of that variable merely by being within a certain proximity of the agent in question. An internally observable (I) variable is one where the agent is aware of the value of the variable, but other agents are not. An unobservable variable (N) is one which no agent, and thus nothing within the system, knows the value of.

It is possible that a variable may have some values which are observable, and others which are not or a group of values which will all appear the same to an observer, this is a partly observable variable. In the case of an agent representing a person and a variable representing their physical condition certain values may be externally observable (for instance a broken arm), others may be only internally observable (a stomach ache).

This observability is represented as implicit 1-to-location communication, each agent in the location receives, for instance, a message that another agent has a broken arm. Most agents will not respond to this input, but it is available to all as in the corresponding real life situation all people in a room would be able to see that a patient has a broken arm without the need to specifically ask this person about it.

Through the round of interviews an initial set of state variables has been defined, based on the minimum amount of information required to model each patient and member of staff. The variables, their values and their kind of observability are shown in Table 2.1. Some of the state variables will have a potentially very large set of possible values, e.g. the symptoms or physical condition.

2.4.1.2 Inputs, outputs, and state transitions.

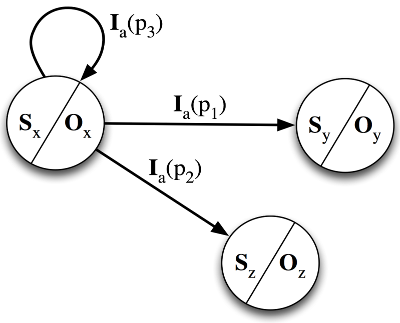
Upon each time step the state machine moves to the next state. This may be another state or the same one it was in before the transition. The next state the machine takes is dependent on the input during that state. The input may be more accurately described as an input vector ($\mathbf{I}$) that contains a number of input variables, each one of which may take a number of different values. As this is a Moore machine, the output depends only on the state, so each state has its own output, although various states may have outputs that are identical. Again, the output is more accurately described as an output vector ($\mathbf{O}$), a collection of output variables, each with a number of defined possible values. Transitions between states are dependent on the current state at time t ($\mathbf{S}_t$) and the input at time t ($\mathbf{I}_t$). Following the transition the state machine will be in a new state ($\mathbf{S}_{t+1}$). The state machine can be represented as a state transition table, as shown in Table 2.2, where each row represents a unique state-input combination, showing the output (defined solely by the current state) and the state in the next time step (defined by the current state and the input).

Table 2.2: State transition table.

Current state / output	Input	Next state / output
$\mathbf{S}_0$ / $\mathbf{O}_0$	$\mathbf{I}_0$	$\mathbf{S}_i$ / $\mathbf{O}_i$
$\mathbf{S}_0$ / $\mathbf{O}_0$	$\mathbf{I}_1$	$\mathbf{S}_j$ / $\mathbf{O}_j$
$\mathbf{S}_0$ / $\mathbf{O}_0$	$\mathbf{I}_2$	$\mathbf{S}_k$ / $\mathbf{O}_k$
$\vdots$	$\vdots$	$\vdots$
$\mathbf{S}_x$ / $\mathbf{O}_x$	$\mathbf{I}_0$	$\mathbf{S}_y$ / $\mathbf{O}_y$
$\mathbf{S}_x$ / $\mathbf{O}_x$	$\mathbf{I}_1$	$\mathbf{S}_z$ / $\mathbf{O}_z$
$\vdots$	$\vdots$	$\vdots$

2.4.1.3 Probabilistic state transitions.

In some specific cases the state machine involves probabilistic transitions, where a given combination of current state and input has more than one possible next state. Which transition is made is chosen at random at the time of the transition, weights on each transition provide a means for specifying transitions that are more or less likely for a given individual. Each one of the input variable of the input vector ($\mathbf{I}$)) may take a number of different values with a certain probability. In these cases our state transition table is defined with probabilities on the input as shown in Figure 2.2b. An agent in state $\mathbf{S}_x$ receiving input $\mathbf{I}_a$ may move to either one of states $\mathbf{S}_y$, $\mathbf{S}_z$ or remain in the same state, with a probability of p_1, p_2, and p_3 respectively. One of these transitions will always occur, which is to say $p_1 + p_2 + p_3 = 1$. The state diagram would then have three different transitions for that state-input combination as shown in Figure 2.2a.



(a) Probabilistic state transition graph

Current state / output	Input	Next state / output
⋮	⋮	⋮
$\mathbf{S}_x$ / $\mathbf{O}_x$	$\mathbf{I}_a(p_1)$	$\mathbf{S}_y$ / $\mathbf{O}_y$
$\mathbf{S}_x$ / $\mathbf{O}_x$	$\mathbf{I}_a(p_2)$	$\mathbf{S}_z$ / $\mathbf{O}_z$
$\mathbf{S}_x$ / $\mathbf{O}_x$	$\mathbf{I}_a(p_3)$	$\mathbf{S}_x$ / $\mathbf{O}_x$
⋮	⋮	⋮

(b) Probabilistic state transition table

Figure 2.2: Probabilistic state transition graph and its corresponding table.

Probabilities may be different for each agent, in this way heterogeneity is provided to agents as people, since agent behaviour can be probabilistically defined external to their state.

2.4.2 Passive agents.

Passive agents represent services within the hospital system such as the IT infrastructure that allows patient details to be stored, radiology services and other laboratory tests as well as specialist systems such as pneumatic tube networks that some larger hospitals use to quickly transfer samples from one part of the department to another.

2.4.3 Communication model.

The communication model represents three basic types of communication. First type is 1-to-1 communication, such between two individuals, for instance admission staff and patient, where a message has a single source and a single destination. Second is 1-to-n communication, where a message has a single source and a specific set of recipients, for example when a doctor communicates with both patient and his companion. The final type is 1-to-location communication, where a message has a single source, but is received by every agent within a certain area or location. This occurs when triage nurse call send a message to the patients of the waiting room, through the loudspeaker system.

Implicit, or passive, communication also exists, where an agent may be producing communication just be remaining in a certain area. This is the manner in which agent vision, what each agent sees, can be represented using the same model. An agent is continuously emitting messages with regard to its visible physical status and location, other agents receive these 1-to-location messages and may act upon them in certain circumstances. For instance, an agent waiting for another agent in a certain area will receive communication that the agent has entered and act upon it, representing, for instance, a nurse seeing a patient, enter a triage room and attending to them.

Each message is comprised of a number of components. The source and destination of the message, where the source is the individual and the destination is either defined as an individual, a group of individuals, or location (where all individuals within that location will receive the message). The actual content of the message is the final part, creating a message tuple of the form `(<src>, <dst>, <content>)`.

The `<dst>` component of the message is the implicit destination of the message, in the real world case of 1-to-1 or 1-to-n communication is communicated via body language such as an individual facing another and making eye contact while talking. In the case of a 1-to-location message the implicit destination is the location. In some cases, a 1-to-location message is actually only meant for a certain agent, in which case the `<content>` component of the message will need to contain an explicit destination. A real world example of this is a loud speaker, all individuals within hearing distance of the loud speaker will hear it, but if it is only directed at a certain individual their name or some other identifier will need to be used, so the specific individual knows it is for them and the remaining individuals know it is not for them.

2.4.4 Environment.

All actions and interactions modelled take place within certain locations, collectively known as the environment. The environment itself can be defined to different levels depending on the positional precision required of the model.

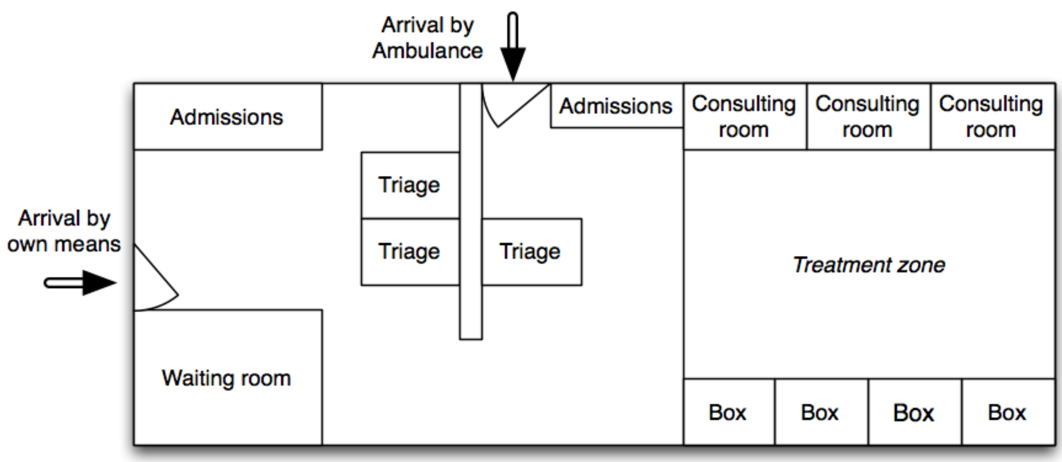


Figure 2.3: Simplified emergency department layout.

The environment in which the agents move and interact is passive and discrete. There is little distinction made between agents in the same location, a patient in the waiting room does not have any more specific sense of position than that they are in the waiting room. Certain locations may be physically distinct, but functionally identical, for instance there are usually a number of triage rooms, an agent in any one of these will act as if they are in any triage room, however they are distinct in order to represent that each available room may only be used by one nurse-patient group at a time. The environment also contains representations of the relative distances between different discrete locations.

The Figure 2.3 illustrates a representation of topographical distribution of the emergency department.

2.5 Simulation.

It is quite difficult, even almost impossible, to separate modelling from simulation, since why taking too much time in modelling if that model is not going to be tested? The word simulation comes from the Latin verb *simulare* and means to imitate, to simulate the operations of different kinds of some real thing or processes. Simulation is quite important since it allows us to understand the behaviour of a system, and to evaluate different strategies within a given structure. Computer simulation utilises the computers to carry out experimentation on a model of the system of interest.

Simulation is *ad hoc* or mandatory when: It is almost impossible to do experimentation in reality, because either the system does not exist, or would be dangerous or quite expensive. Also, when the system cannot be interrupted, and time scale have to be changed.

Amongst the advantages of doing simulation are:

- Cost, experiments on real systems might be quite expensive.
- Time, it is possible to simulate weeks, months, or even years in seconds.
- Safety, effects of extreme conditions can be studied.
- Replication, simulations are exactly replicable.

There are different sort of simulations as:

- Static or Dynamic; does the time play an important role in the model?
- Continuous or Discrete; the *state* of the system changes all over the time or only at specific or discrete times?
- Deterministic or Stochastic; is everything for sure or is there uncertainty?

The Figure 2.4 shows different ways to study a system [5].

Most of the time the systems as well as the simulation are *dynamic, discrete*, and *stochastic*, which is the case of the *ED*.

2.5.1 Agent Based Emergency Department Simulator.

The simulator for this work is used as a black box, but the more realistic the simulator is, the better results and optimisations are. It is implemented in NetLogo.

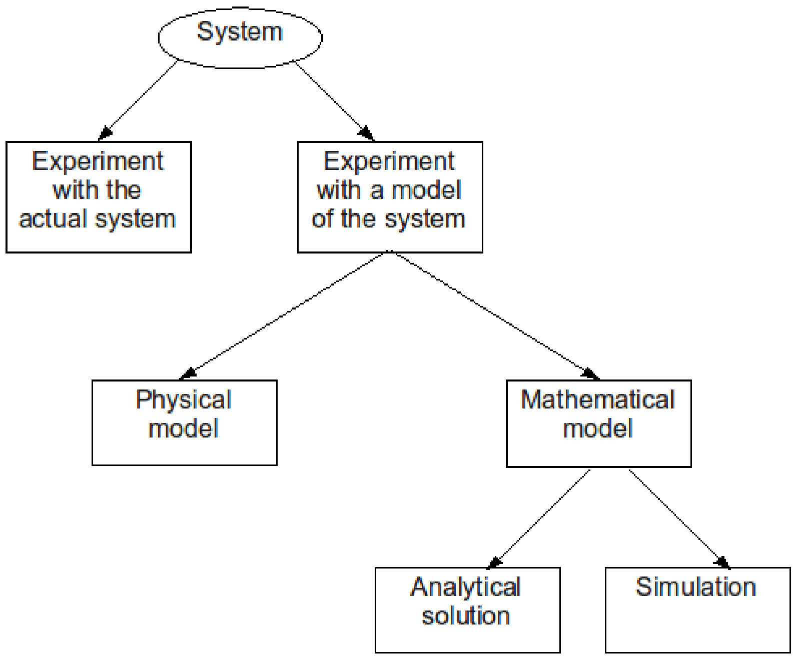


Figure 2.4: Ways to study a system.

The present version of the simulator is illustrated in Figure 2.5. It has up to 4 diagnostic rooms, 3 triaje rooms, 2 waiting rooms, an area of admissions, the entrance, and the exit. Some parameters can be set from the *GUI*, they are amount and sort of staff members: doctors, triaje nurses, and admissions, and senior or junior, respectively. Also, the input arrival patient, in percentage, as well as the maximum number of iterations can be set. Finally, also it can be selected if information about times, costs, and debugging is needed.

One of the most useful tools that NetLogo has, is the BehaviorSpace, which allows doing experiments without using the *GUI*. BehaviorSpace runs a model many times, systematically varying the settings of the model and recording the results of each model run. This process is sometimes called *parameter sweeping*. It lets you explore the *space* of the model of possible behaviours and determine

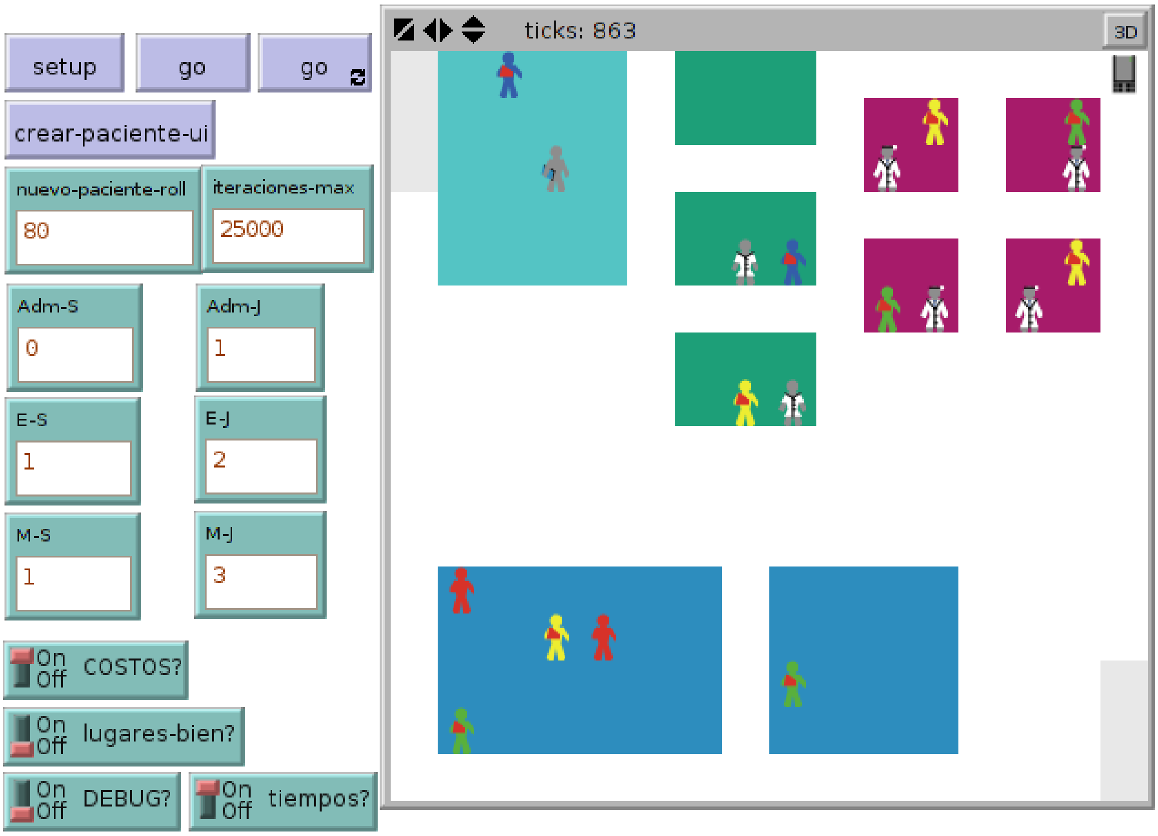


Figure 2.5: Present version of the simulator.

which combinations of settings cause the behaviours of interest.

2.6 Related works.

The interest on simulating healthcare systems is not new, in 1979 computer simulation was applied to hospital systems to improve the scheduling of staff members [12], and in [13] the aim was to quantify the impact that the amount of staff members, and beds had on patient throughput time. Moreover, a survey of discrete-event simulation in health care clinics was presented in [14].

Although, discrete-event simulation is widely used in simulating healthcare systems, agent technology is a good option to be used in healthcare applications, since it characterise better the operation of complex systems as *ED*s. Previous works modelling healthcare systems have focused on patient scheduling under variable pathways and stochastic process durations, the selection of an optimal mix for patient admission in order to optimise resource usage and patient throughput [15]. Work has been performed evaluating patient waiting times under the effects of different ED physician staffing schedules, and the only one found until now that utilises real data [16] or patient diversion strategies [17], both using differing degrees of agent-based modelling.

There is a relevant article which uses *ABM* for simulaton the workflow in *ED* [18], it focus on triaje and radiology process, but not real data is used, the acuity of patients are not consider, and healthcare providers do not always serve patients in a first-come-first-serve basis.

This proposal addresses many of the issues surrounding the modelling and simulation of a hospital emergency department using agent-based technologies. The basic rules governing the actions of the individual agents are defined, in an attempt to understand micro level behaviour. The macro level behaviour, that of the system as a whole, emerges as a result of the actions of these basic building blocks, from which an understanding of the reasons for system level behaviour can be derived [19].

Simulation optimisation is used to improve the operation of *ED* in [20], using a commercial simmulation package, and in [21] combines simulation with optimisation, which involves a complex stochastic objective function under to deterministic and stochastic set of restrictions.

Finally, an evolutionary multiobjective optimisation approach is using for dynamic allocation of resources in hospital practice [22], while in [23] found that combining agent-based approaches and classical optimisation techniques complement each other, and multiobjective evolutionary optimisation of agent-based models is found in [24], in order to obtain approximated Pareto fronts.

Chapter 3

Optimisation.

3.1 Introduction.

Optimisation is a common sense process, but dificult to specify neatly and rigorously, due to misconceptions or superficial knowing of the field. Optimisation, in general, is defined as finding the best solutions for a given problem under some conditions. Optimisation is applied in quite different fields, so, it makes hard to express an exact definition. For example, in engineering the aim is to maximise the performance of a system with minimal resources and run-time, while the objective of mathematics is to find the minimum or maximum of a real function from within an allowable set of variables, and in industries the goal is to enhance the quality and efficiency of the production process. Even into dayly life, there are many cases where the maximum profit with minimal effort is search. Optimisation is a widely used process and difficult to define and find truly optimal solution, which could be for a particular application or period, but not for all cases.

The purpose of this chapter is to present the terminology used, basic mathematical concepts, as well as the classification, and utilisation of some algorithms about this useful, and challenging field. Finally, metaheuristics are concisely presented.

3.2 Optimisation.

Mathematically an optimisation problem can be stated as:

$$\begin{aligned} \max / \min \quad & f(x) \\ \text{subject to} \quad & x \in C \end{aligned}$$

where x is the variable, f is a function ($f : C \rightarrow \mathbb{R}$), C is the constraint set, and $\exists x_0 \in C$ such that $f(x_0) \leq f(x) \quad \forall x \in C$ for minimisation, and $f(x_0) \geq f(x) \quad \forall x \in C$ for its counterpart, maximisation.

The function $f : C \rightarrow \mathbb{R}$ is known as the objective function or performance index, and it is not necesarily pure mathematical formulation, but could be a complex algorithm. The objectives are general statements of what to optimise under some restrictions, where the optimisation process is going to apply.

Usually, the domain $C \subseteq \mathbb{R}^n$ represents the *problem or search space* that can be any sort of elements, e.g. arrays, numbers or equipments, amongst others. The domain is quite common specified by a set of conditions or constraints that its elements have to satisfy. These elements are known as *candidate solutions*, which define a *feasible region*. The Figure 3.1 illustrates these definitons.

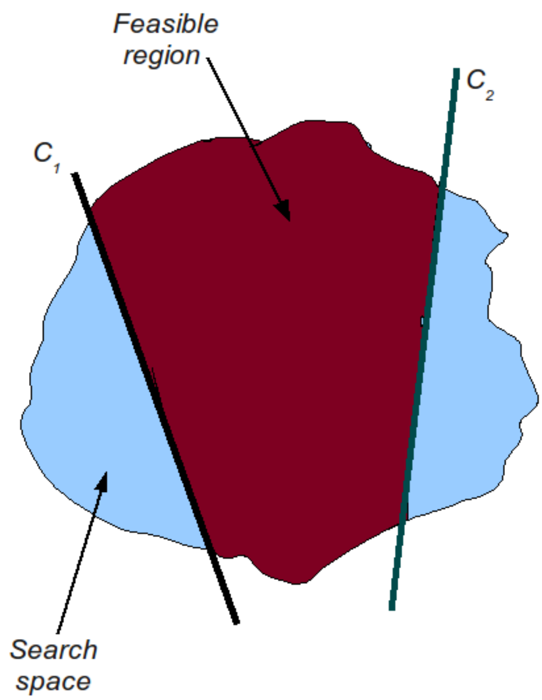


Figure 3.1: Optimisation problem under constrains C_1 and C_2 .

In order to define the process of optimisation, at least, the next three elements ought to be determined [25]:

- the system description or the model,
- objective function or the performance index, and
- the optimisation method.

And last, but not least important, one more concept requires to be define, the level of optimisation, which express the degree of precision in a formal or mathematical formulation, and imposes or specifies the desired implementation. This concept shows the accuracy or practical reason rather than analytical. Because sometimes a quick answer is desired even with loss of generality or rigorousness, but on the other hand, when there is no space to impreciseness, an exact solution, which implies high degree of accuracy, is desired, even though it takes lots of time to get it.

The model of the problem is described in section 2.4, whereas the objective function or performance index is detailed following.

3.2.1 Objective function.

It is stated that the objective is a general statement about what to optimise subject to certain restrictions or constraints, which implies the degree of searched solution as well as how this solution should be computed. The performance index or objective function is a rigorous mathematical expression, which allows quantitative comparisons. These comparisions depend on the level or degree of the optimisation. The more level of optimisation, the higher the quality of the solution is. The performance index or objective function is also known, amongst others, as cost function or criterium, in economy and control theory, respectively.
Setting the objective function is a hard task, but control theory helps to assign it, without loss of generality, to one of two categories [25]:

1. A time (to be minimised or maximised).

2. An amplitude (range or a benefit to be maximised or an error to be minimised).

3.2.2 Constraints.

Constraint is a limitation, restriction, and, in general, a condition which any solution to an optimisation problem have to satisfy. It can be either an:

- equality constraint, or
- inequality constraint

And, as it stated above, *feasible region* (illustrated on Figure 3.1), is composed by those *candidate solutions* that satisfy all restrictions.

3.3 Numerical methods.

To solve an equation analitically is quite difficult in most cases, unless such equation is extremely simple. These difficulties could be the following:

- exists an analytical solution, but the order of equations is of high order,
- geometry is very complex,
- there is no solution or analytical procedure,
- algorithm exists, but do not has polynomial time solution.

Engineers and scientifics have chosen experimental approches to many of the hard real systems. Although, there are limitations of those approximations, such as inherent method, or experimental errors, and coarse accuracy of the results. In the present time, it is almost imposible to separate the utilisation of computers throughout the design, analysis, and simulation processes. Numerical methods belongs to numerical analysis, which is part of mathematics and computer science that creates, analyses, and implements algorithms for solving numerically the problems of continuous mathematics using computers.

3.3.1 Optimisation methods.

One of the fields which concern to numerical analysis is optimisation theory, and before entering into the methods some topics should be discuss. In order to choose an optimisation method some properties ought to be demanded to such method or technique that solve the model using a system of algebraic or differential equations, or any other mathematical model if it is available. However, there are some techniques available when there is no possibility of mathematical treatment. These characteristics can be divided whether the solution is analytical or numerical [25].

Mathematical qualities are:

- Existence of the solution, is there any solution?
- Uniqueness of the solution, is there only one?
- Necessary conditions, that have to be satisfied in all cases.
- Sufficiency conditions, if are satisfied guarantee an extremum.
- Absolute or local extremum, is the solution valid over a small are or over the whole search space?
- Weak and strong extremum.

While computational characteristics are (more practical rather than mathematical):

- Existence of a numerical computing method.
- Kind of computer used.
- Convergence, if the method uses an iterative procedure.
- Computing time.

Convergence is quite important property, it is a condition *sine qua non* for any numerical method. It is said that it is convergent if the numerical solution approaches the exact solution as the step size goes to zero.

Since optimisation inherently implies control, controllability and observability are defined. It is said that a system is controlable at the instant t_0 if it is possible to take from any initial state, $x(t_0)$ to any other state in finite time. And, a system is observable at time t if, with the system at state $x(t)$, it is possible to determine such state in finite time using only its outputs. Controllability and observability are dual aspects of the same problem.

Once the problem is characterised, the objective function and the constraints are set. Thus, the next step is to pick up a method to solve the defined problem. Such method will depend on whether [25]:

- These settings are *static or dynamic.*
- The objective function is restricted or not.

- These settings are linear or nonlinear.
- These settings are one-dimensional or multidimensional.

Whithout loss of generality, it can be stated that the static version is the simplified form of the dynamic method, as well as the linear problem is the simplified form of the nonlinear problem.

3.4 Optimisation taxonomy.

The optimisation methods can be broadly classified as analytical or not. This taxonomy, shown in Figure 3.2, is based on the possibility to solve the problem using a mathematical model. The analytical method imposes the existence of the derivates of the objective function; unfortunately, not always the function has such property. Therefore, alternative methods have to be used. The classical optimisation methods can be seen as search methods. If the size of the search space is little, or its computational search time is polynomical and not *NP-hard* or *NP-complete*, then an exhaustive search can be used. This approach allows to find the best solution, even though, it takes too much time to do it, and can be the first approximation or classical method utilised to tackle search problems, which lack of mathematical model.

Generally, optimisation algorithms can be divided in another two classes: deterministic and probabilistic algorithms [26]. The former is used if the search space can be efficiently explored. To solve a problem deterministically could be quite difficult or hard if the dimension of the search space is very large, as stated above. It is when probabilistic methods appear. The Monte Carlo-based

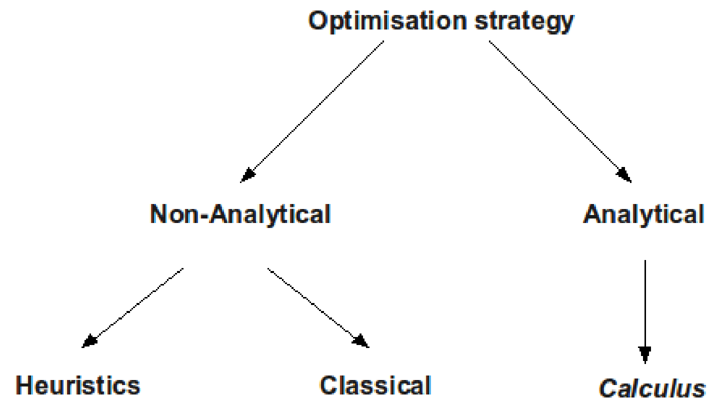


Figure 3.2: Optimisation classification.

approach is one of the most relevant of them. Its purpose is to offer a solution, which could be not the global optima, in a short time. The Figure 3.3 illustrates this classification.

Before continuing the discussion over optimisation algorithms, it is needed to define what an optimum is.

3.5 Optimum.

Global optimisation, that is about finding the best possible solutions for given problems, can be done over a single or multiple functions.

3.5.1 Single objective functions.

When optimise a single function f, an optimum can be either a maximum, if it is a maximisation problem, otherwise a minimum, when a minimisation problem is. It can be local or global optimum. The latter is an optimum of the whole domain, whereas the former is an optimum of only a subset of such

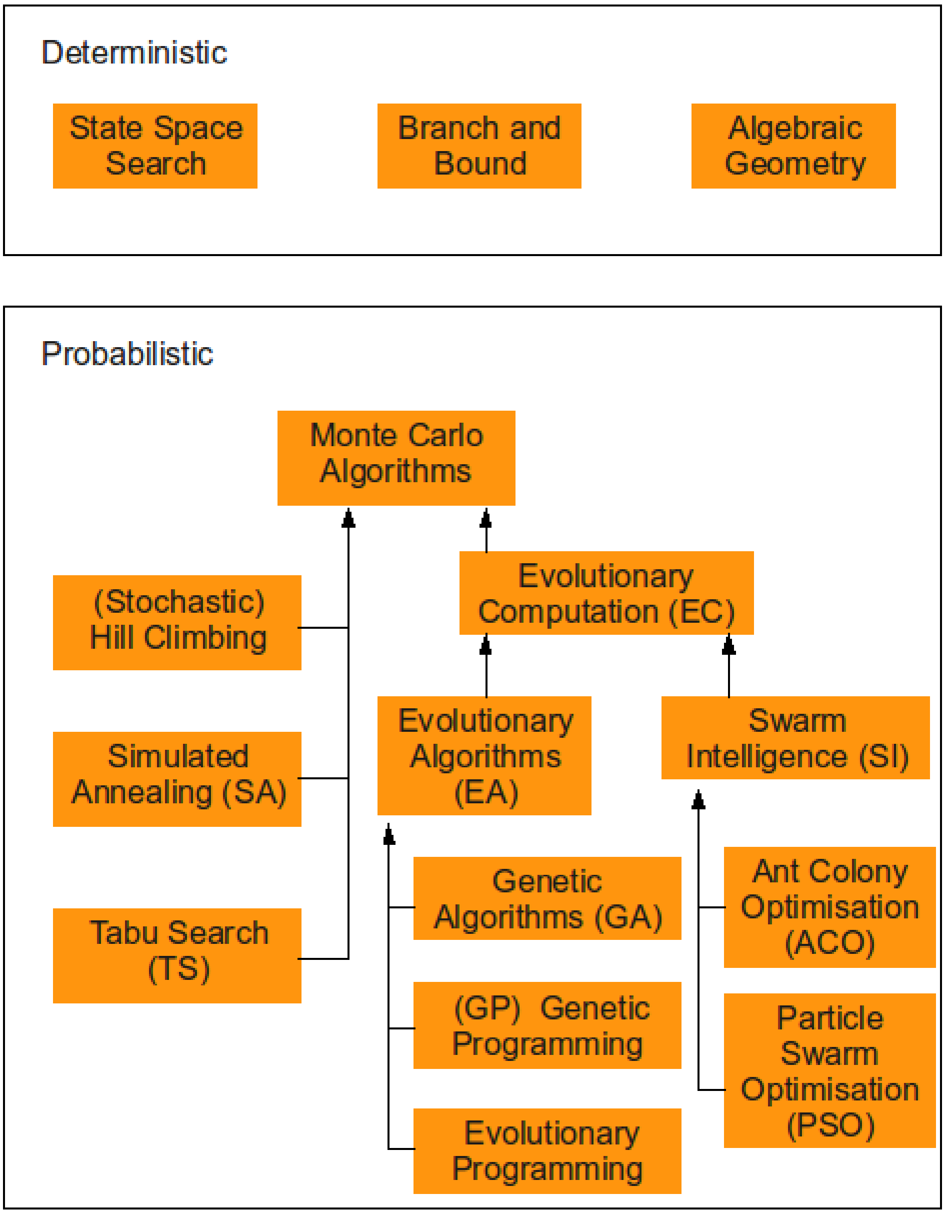


Figure 3.3: Taxonomy of some global optimisation algorithms.

domain. Usually, the maximum and minimum of a set are the *greatest* and *least* values in such set. The Figure 3.4 shows both local and global maximum.

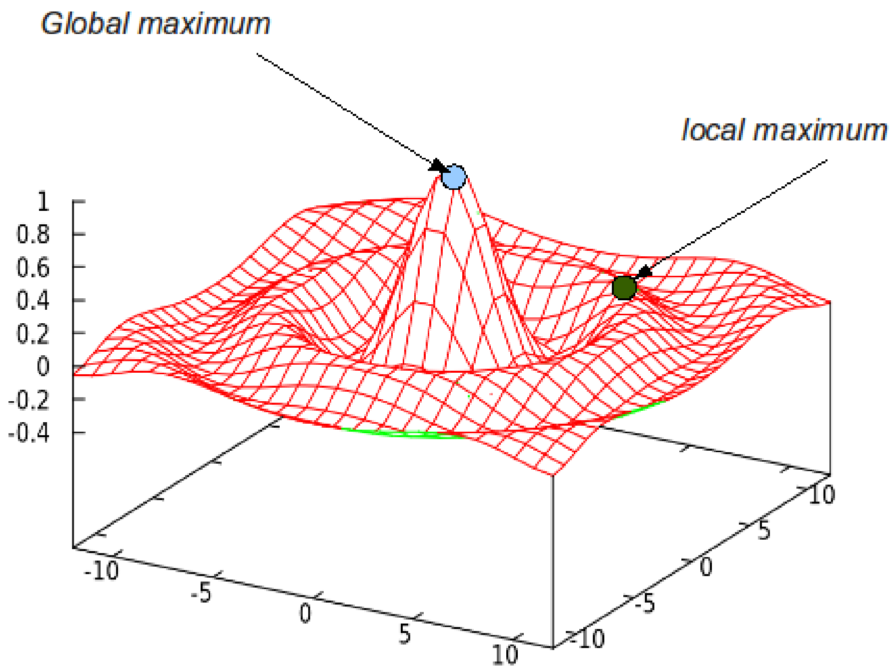


Figure 3.4: Global and local maximum.

Hence, following are the definitions for different sort of optima in single objective functions.

Definition 1. *A local maximum* $\vec{x_l} \in \mathbb{X}$ *of one objective function* $f : \mathbb{X} \rightarrow \mathbb{R}$ *is an input element with* $f(\vec{x_l}) \geq f(x) \forall x$ *neighboring* $\vec{x_l}$.

If $\mathbb{X} \subseteq \mathbb{R}$, it can be write as

$$\forall \vec{x_l} \exists \varepsilon > 0 : f(\vec{x_l}) \geq f(x) \forall x \in \mathbb{X}, \mid x - \vec{x_l} \mid < \varepsilon$$

Definition 2. *A local minimum* $\vec{x_l} \in \mathbb{X}$ *of one objective function* $f : \mathbb{X} \rightarrow \mathbb{R}$ *is an input element with* $f(\vec{x_l}) \leq f(x) \forall x$ *neighboring* $\vec{x_l}$.

If $\mathbb{X} \subseteq \mathbb{R}$, it can be write as

$$\forall \vec{x_l} \exists \varepsilon > 0 : f(\vec{x_l}) \leq f(x) \forall x \in \mathbb{X}, \mid x - \vec{x_l} \mid < \varepsilon$$

Definition 3. *A local optimum $x_l^* \in \mathbb{X}$ of one objective function $f : \mathbb{X} \to \mathbb{R}$ is either a local maximum or a local minimum.*

Definition 4. *A global maximum $\hat{x} \in \mathbb{X}$ of one objective function $f : \mathbb{X} \to \mathbb{R}$ is an input element with $f(\hat{x}) \geq f(x) \forall x \in \mathbb{X}$.*

Definition 5. *A global minimum $\check{x} \in \mathbb{X}$ of one objective function $f : \mathbb{X} \to \mathbb{R}$ is an input element with $f(\check{x}) \leq f(x) \forall x \in \mathbb{X}$.*

Definition 6. *A global optimum $x^* \in \mathbb{X}$ of one objective function $f : \mathbb{X} \to \mathbb{R}$ is either a global maximum or a global minimum.*

Even a one-dimensional function $f : \mathbb{X} = \mathbb{R} \to \mathbb{R}$ may have more than one global maximum, multiple global minima, or even both in its whole domain $\mathbb{X}$.

3.5.2 Multiple objective functions.

Even though single objective optimisation methods models lots of real problems, there are many applications where these models are unsuitable, since it is almost impossible to get a single solution that at the same time optimises all the objectives. To overcome this case multiobjective optimisation comes into play.

Problems with two or more objectives functions, known as *multiobjective functions*, are quite common in many fields. The solution of those problems is very hard since their objectives tend to be in conflict with each other. Nevertheless, to simplify, many of these problems are modelled as single objective using only one of the original functions, and handle the others as constraints.

The multiple optimisation problem can be stated as following

$$\text{optimise} \quad [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]$$

subject to m inequality constraints:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{1}$$

and the p equality constraints:

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \ldots, p \tag{2}$$

;where k is the number of objective functions

$$f_i : \mathbb{R}^n \to \mathbb{R} \qquad \text{and}$$
$$\vec{x} = [x_1, x_2, \ldots, x_n]^T$$

is the vector of decision variables, and it is desired to determine from amongst the set $\mathcal{PF}$ of all vectors which satisfy (1) and (2) the particular set of values $x_1^*, x_2^*, \ldots, x_n^*$ which yield the optimum values of all the objective functions.

The *optimality* concept is quite different, since it is very rare that exists a single point, which at the same time, optimises all such objective functions. Hence, when dealing with multiobjective optimisation problems, instead of seeking single solutions it is usual to look for *trade-offs*.

3.5.2.1 Pareto optimality.

The more accepted notion of *optimum* into the multiobjective optimisation problems was originally propposed by Francis Ysidro Edgeworth in 1881 [27], and then generalised by Vilfredo Pareto in 1896 [28]. It is well known as *Pareto optimality* [29].

It is say that a vector of decision variables $\vec{x}^* \in \mathcal{PF}$ is a *Pareto optimal* if $\nexists$ another $\vec{x} \in \mathcal{PF}$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*) \quad \forall i = 1, \ldots, k \quad \text{y} \quad f_j(\vec{x}) < f_j(\vec{x}^*)$ for at least one j.

This definition says that x^* is *Pareto optimal* if there exists no feasible vector of decision variables $x \in \mathcal{PF}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions known the *Pareto optimal set.* The vectors x^* corresponding to the solutions included in the *Pareto optimal* set are called *nondominated.* The plot of the objective functions whose nondominated vectors are in the Pareto optimal set is called the *Pareto front.*

The Figures 3.5 and 3.6 graphically describes the Pareto-dominance concept for a minimisation problem with two objectives (k_1, k_2). The Figure 3.5 illustrates the location of several solutions. The filled circles represent *non-dominated* solutions, while the non-filled ones symbolize dominated solutions. In Figure 3.6 is shown the relative distribution of the solutions in reference to x. There exist solutions that are worse (in both objectives) than x, better (in both objectives) than x, and indifferent (better in one objective and worse in the other).

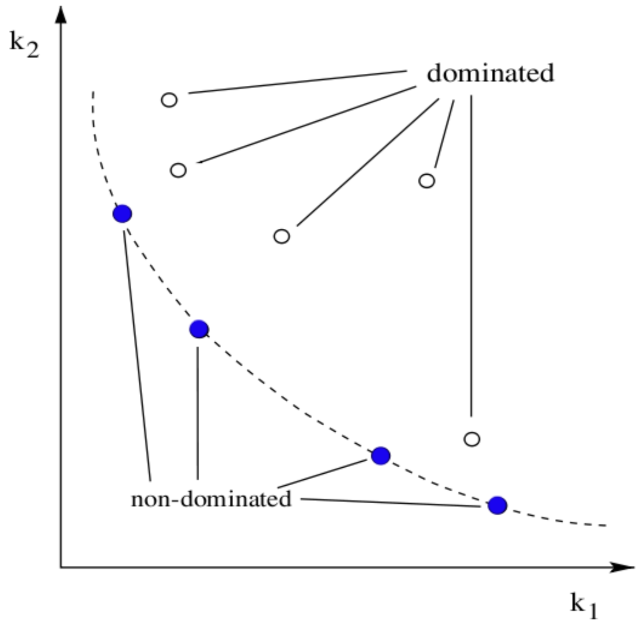


Figure 3.5: Pareto front with *non-dominated*, and dominated solutions.

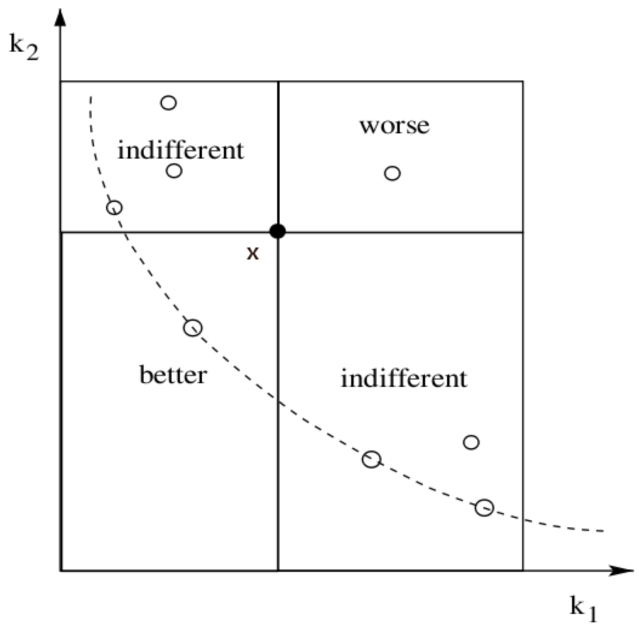


Figure 3.6: Pareto front, different sort of solution in reference to x.

Classical techniques for multiobjective optimisation tend to generate elements of the Pareto optimal set one at a time, this implies that many runs, using different starting points, are required in order to generate lots of those elements. Moreover, most of these techniques are quite sensitive to the shape of the Pareto front, and might not work when the Pareto front is non convex. Therefore, there is needed for techniques that overcome these difficulties.

3.6 Metaheuristics.

Metaheuristic, which are the most general of stochastic optimisation or algorithms, combines objective functions or heuristics commonly by treating them as black-box procedures. A heuristic, as a part of an optimisation algorithm, helps to decide which solution candidate should be tested next or how the next individual can be generated. Many current heuristics are population-based, which means that can generate many elements of the Pareto optimal set in a single run.

There are two main types of multi-objective evolutionary algorithms:

1. Algorithms that do not incorporate the concept of Pareto dominance in their selection mechanism.

2. Algorithms that rank the population based on Pareto dominance.

Amongst all of the plenty of multiobjective algorithms evolutionary, only one of the so-called second generation, from the beginning of 2000s, is presented. This second generation introduced the concept of elitism either by using $(\mu+\lambda)$

selection, and using a secondary (usually external) population. Also, this second generation emphasises on computational efficiency both at an algorithmic level, and at the data structures level.

3.6.1 Nondominated Sorting Genetic Algorithm II (NSGA-II).

[30] It is an improved version of the NSGA. In the NSGA-II, for each solution one has to determine how many solutions dominate it and the set of solutions to which it dominates.The NSGA-II estimates the density of solutions surrounding a particular solution in the population by computing the average distance of two points on either side of this point along each of the objectives of the problem. This value is the so-called crowding distance. During selection, the NSGA-II uses a crowded-comparison operator which takes into consideration both the nondomination rank of an individual in the population and its crowding distance (i.e., nondominated solutions are preferred over dominated solutions, but between two solutions with the same nondomination rank, the one that resides in the less crowded region is preferred). The NSGA-II does not use an external memory as the other MOEAs previously discussed. Instead, the elitist mechanism of the NSGA-II consists of combining the best parents with the best offspring obtained (i.e., a $(\mu + \lambda)$ selection). Due to its clever mechanisms, the NSGA-II is much more efficient (computationally speaking) than its predecessor, and its performance is so good.

Finally, four more alternatives of stochastic optimisation algorithms are sketched [31].

3.6.2 Simulated Annealing.

Based on an algorithm originally to simulate the evolution of a solid in a heat bath until it reaches its thermal equilibrium. It generates local movements in the neighborhood of the current state, and accepts a new state based on a function depending on the current *temperature* t. The two main parameters of the algorithm are $NITER$ (the number of iterations to apply the algorithm) and CS (the cooling schedule), since they have the most serious impact on the performance of the algorithm. The key in extending simulated annealing to handle multiple objectives lies in determining how to compute the probability of accepting an individual $\vec{y}'$ where $f(\vec{y}')$ is dominated with respect to $f(\vec{y})$.

3.6.3 Ant System.

It is inspired by colonies of real ants, which deposit a chemical substance on the ground called pheromone. This substance influences the behavior of the ants: they tend to take those paths where there is a larger amount of pheromone. Pheromone trails can thus be seen as an indirect communication mechanism among ants. From a computer science perspective, it is a multi-agent system where low level interactions between single agents (i.e., artificial ants) result in a complex behaviour of the entire ant colony.

It was originally proposed for the traveling salesman problem (TSP), and most of the current applications of the algorithm require the problem to be reformulated as one in which the goal is to find the optimal path of a graph. A way to measure the distances between nodes is also required in order to apply the algorithm.

3.6.4 Particle Swarm Optimisation.

It is inspired by the choreography of a bird flock. The idea of this approach is to simulate the movements of a group (or population) of birds which aim to find food. The approach can be seen as a distributed behavioural algorithm that performs multidimensional search. In the simulation, the behaviour of each individual is affected by either the best local (i.e., within a certain neighborhood) or the best global individual.

It is worth mentioning that it is an unconstrained search technique. Thus, it is also necessary to develop an additional mechanism to deal with constrained multiobjective optimisation problems.

To extend it for multiobjective optimisation, it is necessary to modify the guidance mechanism of the algorithm such that nondominated solutions are considered as leaders. Note however, that it is important to have a diversity maintenance mechanism. Also, an additional exploration mechanism (e.g., a mutation operator) may be necessary to generate all portions of the Pareto front (mainly in disconnected fronts).

3.6.5 Tabu Search.

It is composed by the three following elements:

- A short-term memory to avoid cycling.

- An intermediate-term memory to intensify the search.
- A long-term memory to diversify the search.

The basic idea of tabu search is to create a subset T of $\mathcal{N}$, whose elements are called *tabu moves* (historical information of the search process is used to create T). Membership in T is conferred either by a historical list of moves previously detected as improductive, or by a set of tabu conditions (e.g., constraints that need to be satisfied). Therefore, the subset T constrains the search and keeps tabu search from becoming a simple hillclimber. At each step of the algorithm, a *best* movement (defined in terms of the evaluation function $opt()$) is chosen. It is worthy mentioning that this approach is more aggressive than the gradual descent of simulated annealing.

Tabu search tends to generate moves that are in the area surrounding a candidate solution. Therefore, the main problem when extending this technique to deal with multiple objectives is how to maintain diversity so that the entire Pareto front can be generated. The proper use of the historial information stored is another issue that deserves attention.

Chapter 4

Experimental Evaluation.

4.1 Introduction.

The purpose of the experiments is to reach the objectives stated for this work, that is to optimise the amount of staff configuration of an *ED*, in order to enhance the performance of such department. It means doctors, triaje nurses and admissions. It also includes some characteristics of staff, that belongs to the model itself, like sort of staff as —junior or senior, less and more expert, respectively, time and cost of each sort. This is shown in Table 4.1, and represents a combinatorial problem. This can be appreciated in Tables 4.2, 4.3, and 4.4 given a total of 1134 scenarios $14 * 9 * 9$ (14D, 9N, and 9A) for staff allocation, but when four different patient arrival probabilities, defined in Table 4.5, are taken into account the total amount of scenarios is 4536.

The period simulated was 24 hrs., one day, which are 25000 ticks for all experiments, as well as the same random seed.

All simulations were done using the simulator previouly explained, utilising the *BehaviorSpace* tool, serially and using the cluster *IBM* of the department,

Table 4.1: Staff members with their associated costs, and time according to their sort.

	Cost		**Time (ticks)**		#
	Senior	Junior	Senior	Junior	
Doctor	1000	500	260	350	***1 − 4***
Nurse	500	350	90	130	***1 − 3***
Admin	200	150	20	35	***1 − 3***

which has 32 Compute nodes with 2 x Dual-Core Intel(R) Xeon(R) CPU 5160 running at 3.00GHz, with 12 GB of RAM, and 4MB of L2 share cache (2x2).

Three different indexes were set in order to evaluate the utility of the *Agent Based Emergency Department* simulator for optimising resources.

4.2 First Experiment.

4.2.1 Index 1.

The first objective set was to minimise patient service stay time in the *ED*, with cost configuration less than 3500 euros. Thus, the first index expressed mathematically is in equation (1):

$$\begin{array}{ll} \text{Min. waiting patient time} & f(D, N, A) \\ \text{subject to} & D_{cost} + N_{cost} + A_{cost} \in Cost < 3500 \quad euros. \end{array} \tag{1}$$

The results are shown in Figures 4.1, 4.2, 4.3, and 4.4; where the blue points are the staff configuration that satisfy the restriction, while the green and red points are the minimum for each case. Each staff configuration that got the

Table 4.2: 14 Doctor cases (D). DJ means Doctor Junior. DS is Doctor Senior, and DRi Diagnostic Room.

DR	DR2	DR3	DR4
DJ	-	-	-
DS	-	-	-
DJ	DJ	-	-
DS	DS	-	-
DJ	DJ	DJ	-
DS	DS	DS	-
DJ	DJ	DJ	DJ
DS	DS	DS	DS
DJ	DS	-	-
DJ	DJ	DS	-
DJ	DJ	DS	DS
DJ	DJ	DJ	DS
DJ	DS	DS	DS
DJ	DS	DS	-

Table 4.3: 9 Triaje nurse cases (N). NJ means Triaje Nurse Junior. NS is Triaje Nurse Senior, and TRi Triaje Room.

TR1	TR2	TR3
NJ	-	-
NS	-	-
NJ	NJ	-
NS	NS	-
NJ	NJ	NJ
NS	NS	NS
NJ	NS	–
NJ	NJ	NS
NJ	NS	NS

Table 4.4: 9 Admission cases (A). AJ means Admission Junior. AS is Admission Senior, and ARi Admission Space.

A1	A2	A3
AJ	-	-
AS	-	-
AJ	AJ	-
AS	AS	-
AJ	AJ	AJ
AS	AS	AS
AJ	AS	–
AJ	AJ	AS
AJ	AS	AS

Table 4.5: Probability of incoming patients.

Patient Arrival (P)
20%
40%
60%
80%

minimum is presented in Tables 4.6, 4.7, 4.8, and 4.9.

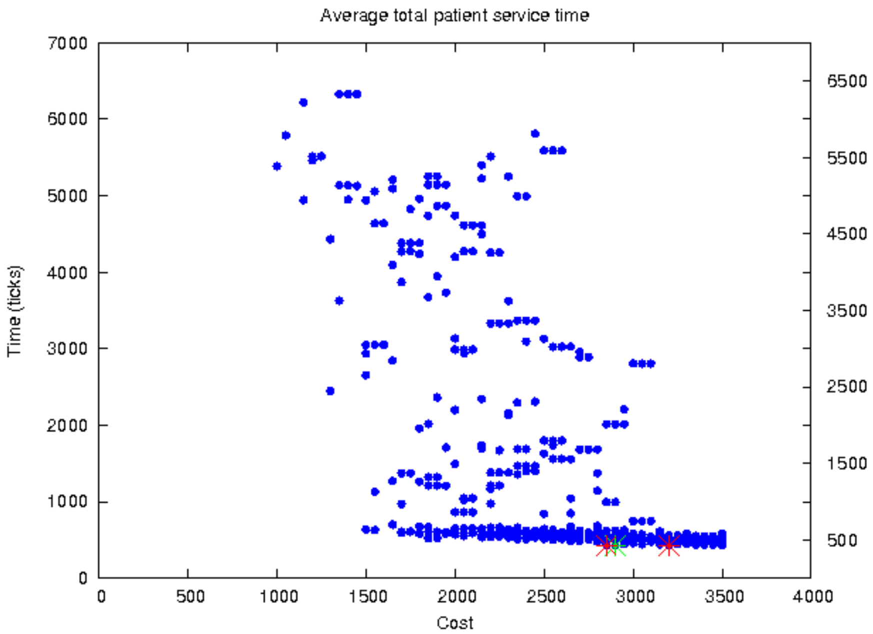


Figure 4.1: Average waiting patient time with P = 20%. Red and green points are the minimum.

From Figure 4.1 and Table 4.6, there are three different staff configurations that got the minimum time, but with different cost. Also, in Figure 4.1 can appreciate that there are many other staff configurations that are quite close to the minimum time, but with much less cost.

In the other cases, where the patient arrival increases there are only few staff configurations around the minimum, or clearly only one. But not only the patient arrival increases, but also the minimum average patient stay time, as expected, but also the standard deviation, patients at waiting rooms, both WR0 and WR1 at times t1, t2, t3, and t4, and finally the amount of unattended patients increases too. In Table 4.10 all these results are shown. In Figure 4.5

Table 4.6: Staff configurations, where S is Senior and J is Junior, that got the average minimum time with P = 20%. They are shown in green and red in Figure 4.1.

Min	Euros	Time (ticks)	# staff	D	N	A
1	3200	428	5	2 S	2 S	1 S
2	2900	428	5	2 S	1 S	2 S
3	2850	428	5	2 S	1 S	1 S, 1 J

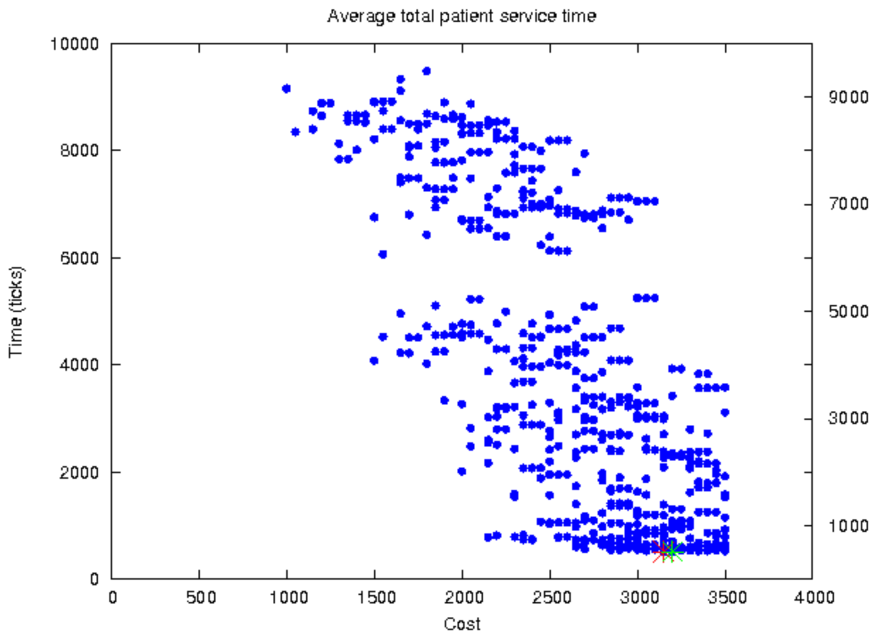


Figure 4.2: Average waiting patient time with P = 40%. Red and green points are the minimum.

Table 4.7: Staff configurations, where S stands for Senior and J for Junior, that got the minimum time with P = 40%. They are shown in green and red in Figure 4.2.

Min	Euros	Time (ticks)	# staff	D	N	A
1	3150	514	5	2 S, 1 J	1 S	1 J
2	3200	514	7	4 J	2 S	1 S

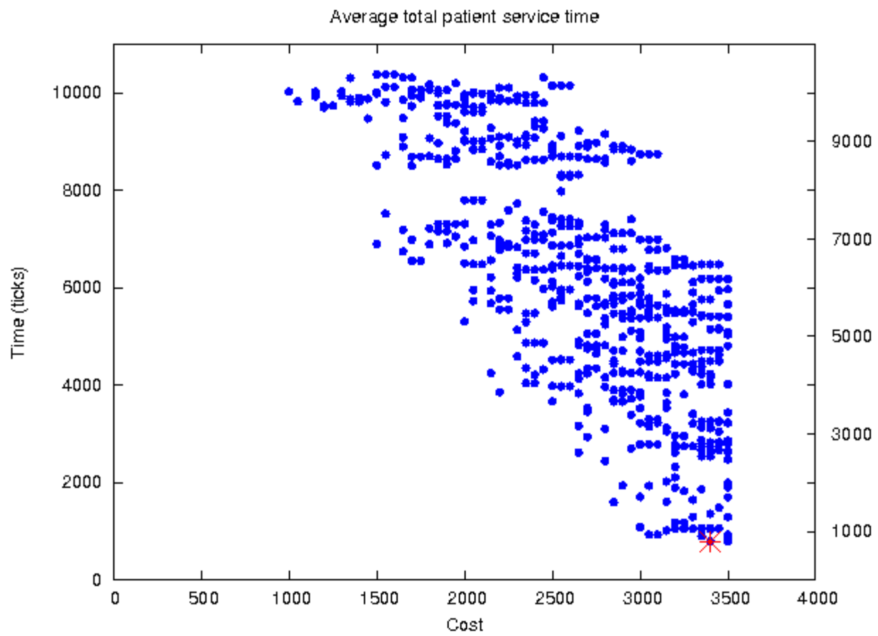


Figure 4.3: Average waiting patient time with P = 60%. The red point shows the minimum.

Table 4.8: Staff configuration, where S is Senior and J is Junior, that got the minimum time with P = 60%. It is shown in red in Figure 4.3.

Min	Euros	Time (ticks)	# staff	D	N	A
1	3400	790	7	1 S, 3 J	2 J	1 S

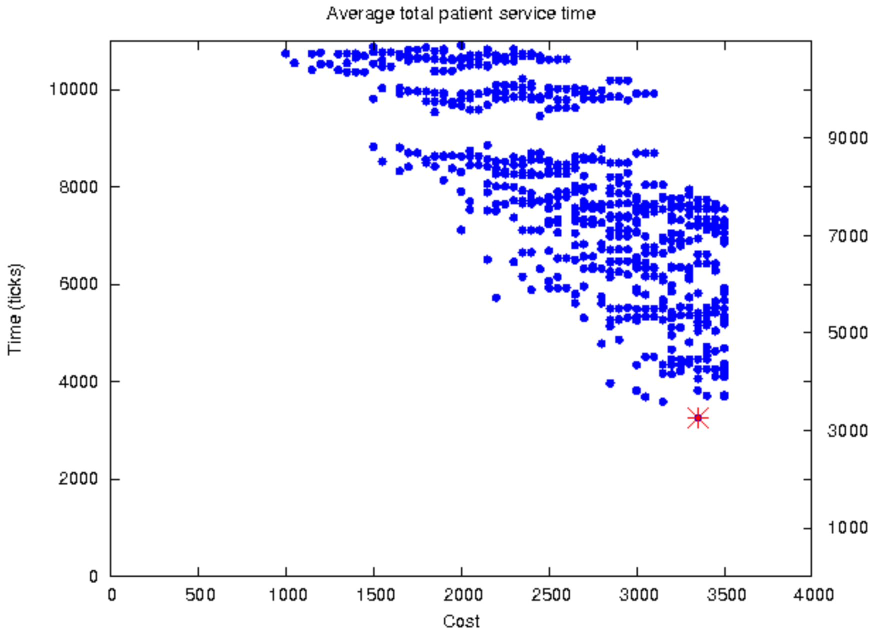


Figure 4.4: Average waiting patient time with P = 80%. The red point indicates the minimum.

Table 4.9: Staff configuration, where S stands for Senior and J for Junior, that got the minimum time with P = 80%. It is represented in red in Figure 4.4.

Min	Euros	Time (ticks)	# staff	D	N	A
1	3350	3266	7	1 S, 3 J	2 J	1 J

the amount of patients in WR are shown, WR0 plus WR1, at four different moments, t1=6500, t2=12500, t3=18750, t4=25000, during the simulation for all the seven cases reported. It is noticed when the patient arrival is high, 80%, patients at waiting rooms increased.

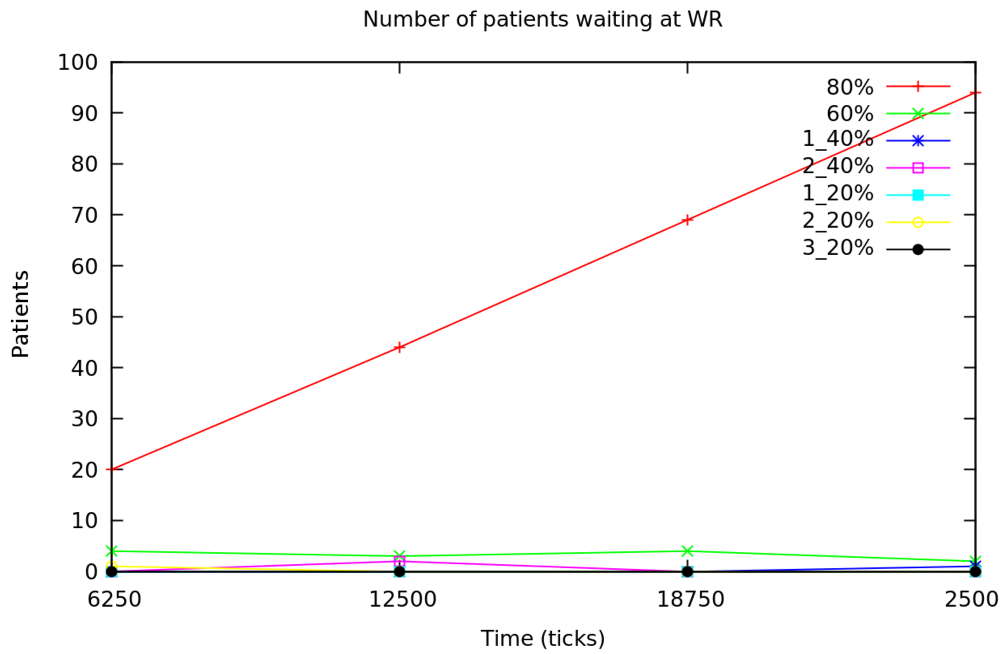


Figure 4.5: Amount of patients at WR, WR0 plus WR1, at four different moments, during the simulation for all the seven cases reported.

4.3 Second Experiment.

4.3.1 Index 2.

The second objective set was to find out, when patient arrival is high, 80%, which staff configuration with the minimum cost, guarantee less or the same waiting patient time than that was gotten when the patient arrival is low, 20% 4.6, time = 428. Thus, this second index expressed mathematically is in (2):

Table 4.10: Results for the best average minimum for each of the four presented scenarios.

P	T1	σ	Euros	# attended patients	# unattended patients	WR0	WR1
						t1, t2, t3, t4	t1, t2, t3, t4
20	428	48 (11%)	2850	83	1	0,0,0,0	0,0,0,0
40	514	81.5 (15.9%)	3150	182	4	0,0,0,1	0,0,0,0
60	790	174.5 (22.1%)	3400	290	8	1,1,0,1	3,2,4,1
80	3266	1670.4 (51.2%)	3350	294	100	8,19,32,43	12,25,37,51

$$\begin{aligned} &\text{Minimise staff cost configuration} \quad && f(D, N, A) \\ &\text{subject to} && Time < 428. \end{aligned} \tag{2}$$

The results are shown in the Figure 4.6, where the blue points are the staff configuration that satisfy the restriction, while the red point is the minimum. Its configuration is presented in Table 4.11. It order to get the same time or less, the superior limit for staff members was increased to: $doctors < 9$; $nurses < 7$, and $admissions < 5$. This index can be seen as looking for a staff configuration that guarantee, at least, the same quality for high patient arrival, 80%, as the same that was gotten for low patient arrival, 20%.

Table 4.11: Minimum staff configuration cost that guarantee the same, at least, quality of service.

Min	Euros	Time (ticks)	# staff	D	N	A
1	6200	421	8	5 S	2 S	1 S

Once the staff configuration has gotten, it was test with the others patient

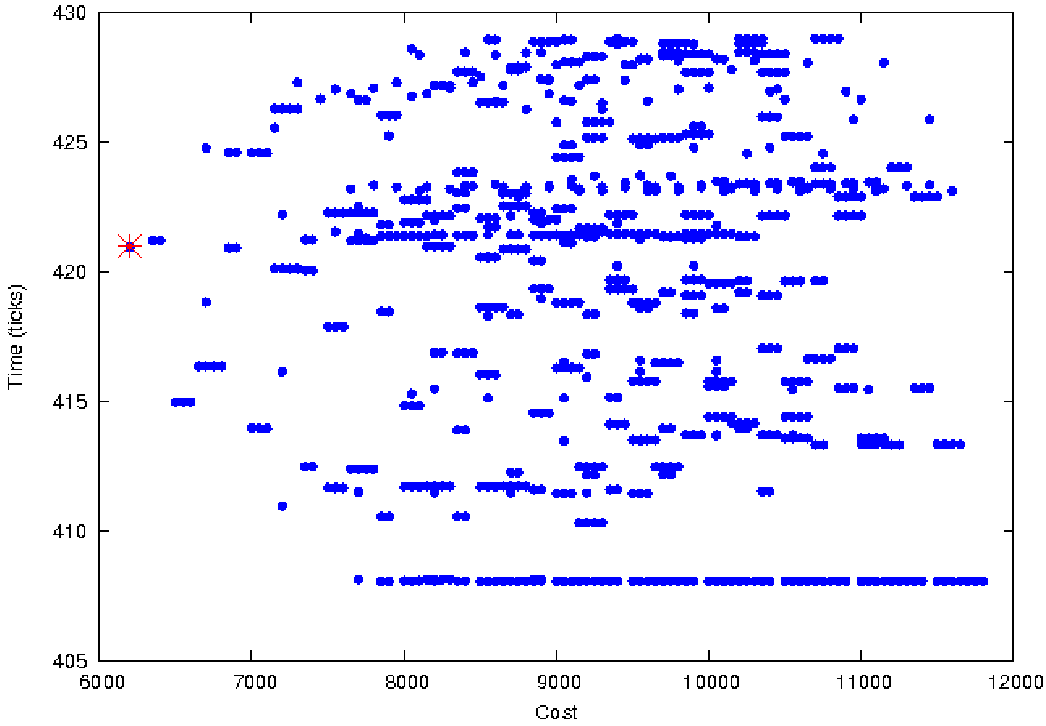


Figure 4.6: Costs of staff configurations, which guarantee same quality. The red point is the minimum staff cost.

arrival , 20%, 40%, and 60%. Each staff configuration with minimum cost is presented in Table 4.12.

Table 4.12: Results for each staff configuration with minimum cost are presented for the four scenarios.

P	T2	σ	Euros	# attended patients	# unattended patients	WR0	WR1
						t1, t2, t3, t4	t1, t2, t3, t4
20	408	2.7 (0.6%)	6200	102	1	0,0,0,0	0,0,0,0
40	408	2.8 (0.7%)	6200	203	3	0,0,0,0	0,0,0,0
60	411	11.7 (2.8%)	6200	300	5	0,0,0,0	0,1,0,0
80	421	18 (4.3%)	6200	394	7	0,0,0,0	1,0,1,0

From the results in Table 4.12 it is worthy to notice that when the cost is increased less than two times the minimum time is reduced almost eight times, and the standard deviation is almost only 4%, as well as patients at waiting rooms is nearly zero.

4.4 Third Experiment.

4.4.1 Index 3.

The third objective set was to minimise a compound index: $cost \times time$, CT, without any restriction. This index is expressed mathematically in equation (3) as:

$$\text{Minimise } cost \times time(CT) \quad f(D, N, A) \tag{3}$$

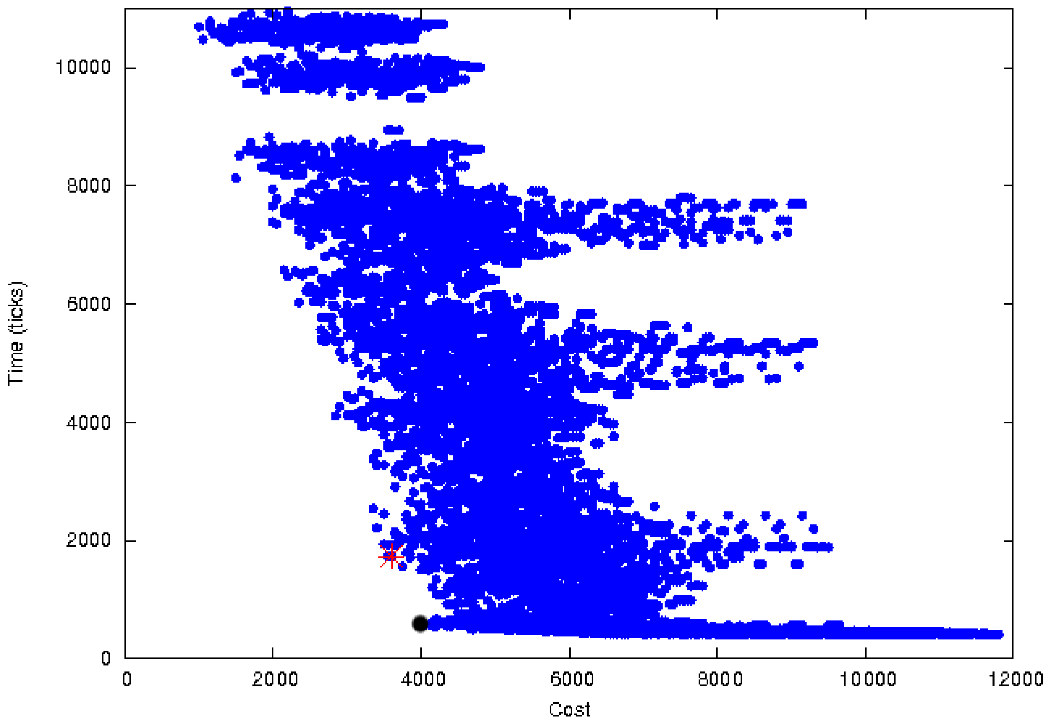


Figure 4.7: Results $y = cost \times time$. Red and black points are minimum, and a worthy staff configuration.

Table 4.13: Minimum staff configuration cost that guarantee the same, at least, quality of service.

Best	Euros	Time (ticks)	$Cost \times Time$	# attended patients	σ	# staff	D	N	A
1	3550	1725	6123750	340	602.4 (34.9%)	9	5 J	2 J	1 S, 1 J
0	4000	585	2342835	378	58.6 (10%)	9	6 J	2 J	1 J

The Figure 4.7 shows all the search space, 16632 staff configurations, but there are two of them that are the most important, and also they are reported and in Table 4.13. Although, both staff configuration are almost the same, they have quite differents average minimum time, this is the reason that explains why the staff configuration label as *Best 1*, despite its lower cost has a worst index evaluation. It is important to notice that a staff configuration a bit more expensive has less than, almost, 15% of standard desviation.

4.5 Performance.

So far, only all the experiments and their results have been presented, but not the execution time of each experiment. All the execution times are presented in Table 4.14, except the very first approach, which was done on my personal computer, using the GUI of the simulator. It had an execution time of 480 secs., per one case. So, approximately 604 hrs. must be needed, in order to run the 4536 staff configuration cases.

Table 4.14: Execution time.

	20%	**40%**	**60%**	**80%**	**Problem size**
divided by patient arrival (4 cases)	1.03 hrs.	2.19 hrs.	4.55 hrs.	**8.06 hrs**	4536
parametric aproach, split configurations (64 cases)	0.19 hrs	0.63 hrs.	1.5 hrs.	**2.7 hrs**	4536
parametric aproach, split configurations (64 cases)				**13.01 hrs**	16632

In the first case, where the search space was divided by the patient arrival, 4 independent runs were gotten. The best only last almost an hour, but the worst last almost eight hours. For the second approach, the search space was divided by both the patient arrival, and the staff configurations, 64 independent runs, and for the same problem size it last 4 times less than the previous one. However, there are clearly imbalance cases, in both experiments, since the runnings for 20%, 40%, and 60% have quite short execution times. The last experiment was run only for one case of patient arrival, 80%, the worst case, it had an execution time of thirteen hours using 64 independent runs.

These quite small configurations of an *ED* demand a lot of time to simulate and optimise only one day, without doing statistical sensitivity analysis. The bigger and the more detail an *ED* is, the longer the execution time is.

4.6 Experiment conclusions.

Three different indexes were set to evaluate the operation of the *Agent Based Emergency Department* simulator. The results were encouraging, since not only they showed what it was expected, the more amount and experienced staff, the less average waiting patient time is. It is, simulation allows to understand, and analyse better the problem.

However, even with this pretty small problem size the amount of combinations are large, as well as the execution time. Moreover, the resources that this problem will demand in order to do statistical sensitivity analysis for longer periods, first to reproduce, and then to foretell are huge. Therefore, a better scheme must be done in order to use more processors, but efficiently, since, at the present time, the imbalance is great and so the wasted of resources.

Chapter 5

Conclusions and future work.

5.1 Conclusions.

- *Agent Based Model* naturally suits modelling and simulating *Emergency Departments*, due to the absence of any formal description for **EDs**, and their fundamentally decentralised, complex, and non-linear characteristics.

- Modelling and simulation is a powerful tool to imitate real systems, that cannot be stopped, or experiment *in situ*, and specially for those systems that lack of a formal or mathematical model.

- To optimise is not an easy task, specially if the problem is multiobjective, or the parameter space is very large.

- Even with a very little configuration of the problem, as is the present status of the *Emergency Department* utilised, and simple objective functions the parameter space is quite large. This is because of the nature of the problem, that is a combinatorial one.

- As a first approach, exhaustively search throughout the problem space was done, and too much time is consumed, even with the minimum configuration that has the present *Emergency Department*.

- Three experiments were done sucesfully, in each of them the index set was different: minimise minimum time of service restricted to some cost; minimise a compound index (cost times time) without constraints, and minimise cost staff configuration that guarantee the same quality service for high ratio of incoming patient as it did when such ratio ws low.

- Dividing the parameter space reduced the execution time of the simulations, but it is not enough, since imbalance showed up, and the processors used were wasted or subutilised.

- The results of the experiments were encouraging, since they showed what common sense said, the more amount and experienced staff, the less average waiting patient time is.

- Simulation allows to understand, and analyse better the problem.

5.2 Future work.

- Do sensitivity statistically analisys of the variables of the simulator, to find out which are the most important parameters?
- Utilise a better approach to search the optima in the problem space, that inevitable includes metaheuristics.
- Set more indexes together with the people from the *Emergency Department* of the Hospital of Sabadell (Parc Tauli Health Corporation).

References.

[1] Kevin Kelly. Essays on science and society: The third culture. *Science*, 279(5353):992–993, 1998.

[2] Melanie Mitchell. *Complexity: A Guided Tour*. Oxford University Press, Inc., New York, NY, USA, 2009.

[3] K. Decker and J. Li. Coordinated hospital patient scheduling. In *ICMAS '98: Proceedings of the 3rd International Conference on Multi Agent Systems*, page 104, Washington, DC, USA, 1998. IEEE Computer Society.

[4] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99:7280–7287, May 2002.

[5] Averill M. Law and David M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 1999.

[6] Brian Heath, Raymond Hill, and Frank Ciarallo. A survey of agent-based modeling practices (january 1998 to july 2008). *Journal of Artificial Societies and Social Simulation*, 12(4):9, 2009.

[7] Emma Norling, Liz Sonenberg, and Ralph Rönnquist. Enhancing multi-

agent based simulation with human-like decision making strategies. In *MABS*, pages 214–228, 2000.

[8] Eliot R. Smith and Frederica R. Conrey. Agent-based modeling: A new approach for theory building in social psychology. *Pers Soc Psychol Rev*, 11(1):87–104, 2007.

[9] Joshua M. Epstein. Modelling to contain pandemics. *Nature*, 460(7256):687, August 2009.

[10] Peter A. Johnson and Renee Sieber. *Planning Support Systems Best Practice and New Methods*, volume 95 of *The GeoJournal Library*, chapter Agent-Based Modelling: A Dynamic Scenario Planning Approach to Tourism PSS, pages 211–226. Springer Netherlands, 2009.

[11] Charles M. Macal and Michael J. North. Tutorial on agent-based modeling and simulation part 2: how to model with agents. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 73–83. Winter Simulation Conference, 2006.

[12] Walton M. Hancock and Paul F. Walter. The use of computer simulation to develop hospital systems. *SIGSIM Simul. Dig.*, 10(4):28–32, 1979.

[13] Charles E Saunders, Paul K Makens, and Larry J Leblanc. Modeling emergency department operations using advanced computer simulation systems. *Annals of Emergency Medicine*, 18(2):134 – 140, 1989.

[14] J. B. Jun, S. H. Jacobson, J. R. Swisher, and Correspondence. Application

of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society*, pages 109–123, February 1999.

[15] Anke K. Hutzschenreuter, Peter A. N. Bosman, Ilona Blonk-Altena, Jan van Aarle, and Han La Poutré. Agent-based patient admission scheduling in hospitals. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 45–52, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[16] Spencer S. Jones and R. Scott Evans. An agent based simulation tool for scheduling emergency department physicians. In *AMIA Annual Symposium proceedings, AMIA Symposium*, pages 338–342, 2008.

[17] Marek Laskowski and Shamir Mukhi. Agent-based simulation of emergency departments with patient diversion. In *eHealth*, pages 25–37, 2008.

[18] Lu Wang. An agent-based simulation for workflow in emergency department. In *Systems and Information Engineering Design Symposium, 2009. SIEDS '09.*, pages 19 –23, 24-24 2009.

[19] Hayden Stainsby, Manel Taboada, and Emilio Luque. Towards an agent-based simulation of hospital emergency departments. In *SCC '09: Proceedings of the 2009 IEEE International Conference on Services Computing*, pages 536–539, Washington, DC, USA, 2009. IEEE Computer Society.

[20] T. Ruohonen, P. Neittaanmaki, and J. Teittinen. Simulation model for improving the operation of the emergency department of special health

care. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 453 –458, 3-6 2006.

[21] Mohamed A. Ahmed and Talal M. Alkhamis. Simulation optimization for an emergency department healthcare unit in kuwait. *European Journal of Operational Research*, 198(3):936 – 942, 2009.

[22] Anke K. Hutzschenreuter, Peter A. Bosman, and Han Poutré. Evolutionary multiobjective optimization for dynamic hospital resource management. In *EMO '09: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, pages 320–334, Berlin, Heidelberg, 2009. Springer-Verlag.

[23] Jan A. Persson, Paul Davidsson, Stefan J. Johansson, and Fredrik Wernstedt. Combining agent-based approaches and classical optimization techniques. In *EUMAS*, pages 260–269, 2005.

[24] Giuseppe Narzisi, Venkatesh Mysore, and Bud Mishra. Multi-objective evolutionary optimization of agent-based models: An application to emergency response planning. In *Computational Intelligence*, pages 228–232, 2006.

[25] Lucas. Pun. *Introduction to optimization practice*. Wiley New York, USA, 1969.

[26] Thomas Weise. Global optimization algorithms theory and application, 2008.

[27] Francis Ysidro Edgeworth. *Mathematical Psychics*. Number edgeworth1881 in History of Economic Thought Books. McMaster University Archive for the History of Economic Thought, 1881.

[28] Bauchet Pierre. Pareto (vilfredo) - cours d'ÃĪconomie politique. *Revue ÃĽconomique*, 16(5):811–812, 1965.

[29] David E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning*. Addison-Wesley, 1989.

[30] C.A. Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28 – 36, feb. 2006.

[31] Leandro Nunes de Castro. *Fundamentals of Natural Computing (Chapman & Hall/Crc Computer and Information Sciences)*. Chapman & Hall/CRC, 2006.