Research Article

Epidemic Random Network Simulations in a Distributed Computing Environment

J. Villanueva-Oller,[1] L. Acedo,[2] J. A. Moraño,[2] and A. Sánchez-Sánchez[2]

[1] *Centro de Estudios Superiores Felipe II, Aranjuez, 28300 Madrid, Spain*
[2] *Instituto de Matemática Multidisciplinar, Edificio 8G 2° Floor, Universitat Politècnica de Valéncia, 46022 Valencia, Spain*

Correspondence should be addressed to L. Acedo; luiacrod@imm.upv.es

We discuss a computational system following the paradigm of distributed computing, which will allow us to simulate the epidemic propagation in random networks with large number of nodes up to one million. This paradigm consists of a server that delivers tasks to be carried out by client computers. When the task is finished, the client sends the obtained results to the server to be stored until all tasks are finished and then ready to be analysed. Finally, we show that this technique allows us to disclose the emergence of seasonal patterns in the respiratory syncytial virus transmission dynamics which do not appear neither in smaller systems nor in continuous systems.

1. Introduction

Networks have become a paradigm of paramount importance in the analysis of many complex systems. Recent applications include disciplines as varied as evolutionary biology [1], the structure of on-line social networks [2], transportation and economics networks [3], or neural networks for the storage and retrieval of memories [4]. Social networks have been ascertained from real data and used to study the social pandemics of smoking [5] and obesity [6]. In the field of epidemiology, the debate about targeted or mass vaccination in the control of smallpox has also been addressed within the context of network models [7, 8]. Following this idea, an outburst of interest in cellular automata models [9] for epidemic spread has been generalised among researchers in epidemiology. Among many others, we have seen in the literature some applications to hepatitis B transmission [10], immune system [11], plagues which devastate some crops [12], and HIV pandemic [13]. Most cellular automata models are defined in euclidean lattice substrates with the individuals occupying the sites of the lattice. Nevertheless, in many situations, only the topology of the network is relevant for the epidemic spreading. More refined models using random networks as the basic substrate have also been considered [14–16], where the authors describe algorithms for simulating the spread of diseases in large networks.

Networks also provide an alternative to the traditional differential equations approach inaugurated by the seminal work of Kermack and McKendrick [17], Edelstein-Keshet [18], and Murray [19]. Differential equations are a powerful and well-known mathematical tool for studying the dynamics of any system, and, consequently, it is not surprising that they have dominated the research in epidemiology for many years. Typically, in these models, we consider the fraction of susceptible (S), infected (I), and recovered (R) individuals and propose a compartmental model for the transitions between these states. The resulting model has been widely studied [19, 20] but, albeit it is a good approximation in some cases, it is clear that it cannot be the final word in the epidemiology of any real disease. The continuous approach cannot, by its own nature, distinguish among individuals and, consequently, the effects of age, sex, previous illnesses, and any other parameters influencing the propagation of the epidemic under study are difficult to implement.

In order to avoid some of the fundamental problems of continuous models, network models emerged. A network is

a set of nodes representing individuals. Labels or properties may be assigned to each node, such as age, sex, and state respect to the disease (susceptibility, infection, recovery, latency, etc.). Nodes are connected by ties that represent disease transmission paths. Once the network model and the disease evolution rules are stated, it is possible to simulate the evolution of the disease on the network nodes over time and to study its spread on the population.

The science of networks provides several standard alternatives for implementing the network substrate. The most traditional one is based upon the pioneering work of Bollobas [21], the so-called "random graphs," where connections among the pairs of subjects are created with the same probability. Alternative models are the scale-free networks [22] or the small-world networks of Watts [23]. The small-world phenomenon (i.e., every pair of nodes is connected through a path which crosses a small amount of neighbours) is found in many social networks linked by friendship, collaboration, or other social binds.

Nevertheless, the studies of real social networks are restricted to a relatively small number of individuals usually not larger than 10,000 individuals [5], but pandemics involve large number of people in the range of millions. So, the development of a distributed computing solution for simulating pandemics in very large networks is a necessary challenge to be dealt with in epidemiology.

In this paper, we describe how we tackled the problem by using two computational systems which follow the paradigm of distributed computing that allowed us to estimate the parameters in random network epidemic models, depending on the amount of tasks to be carried out: one of them, Sisifo, designed by us to work in intranets, is simpler, uses less resources, and has a quicker development, implementation, and deployment; the other is the well-known Berkeley Open Architecture for Network Computing (BOINC) platform [24]. The main difference among Sisifo and BOINC concerns the security issues and the possibility of widespread distributed computing in personal computers of clients connected to Internet, in the case of BOINC, in contrast with the limitation of Sisifo to a computer intranet where security issues are not so important. In particular, BOINC implements public-key encryption against virus attack in which we did not consider in Sisifo.

Both computational paradigms consist of a client/server structure where the server delivers tasks to be carried out by client computers, and when the task is finished, the client sends the obtained results to the server to be stored until all tasks are finished and ready to be analysed. In our case, every task consists of a network model simulation with a set of parameters and the results are the number of persons of each age in each disease state at any time instant.

We must also remark that in this problem, as in many similar problems arisen in statistical physics, it is integral to the computational process to evaluate averages over many realizations over a single substrate and also over many substrates. This is particularly interesting in the simulation of nonequilibrium processes such as diffusion and reactions as well as epidemic propagation [25]. In our case, the first average refers to the evolution of the disease in a given configuration of the random network. However, as we must also consider different network configurations, in order to take into account the variability of the social structure, a second average over these configurations is also necessary. This is the so-called second average.

In order to achieve this, these double average simulations are usually carried out sequentially in a single computer. In this paper, we show two methods on how to parallelize this calculation and perform the averages directly on the server.

The paper is organized as follows. In Section 2, we describe our two proposals for a distributed computing system: the first one is based on intranets, while the second one uses the BOINC platform. Both of them are capable of managing the simulation of random networks up to one million nodes. An application to the simulation of the seasonal respiratory syncytial virus (RSV) pandemic is given in Section 3. In Section 4 we show that seasonality emerges spontaneously as an amplification of intrinsic fluctuations in large networks without any external forcing. The paper ends with some conclusions and a plan for further study in Section 5.

2. Distributed Computing

Distributed computing is the use of distributed systems to solve complex computational problems by dividing them into independent tasks (each of them can be done separately in one computer). The partial solutions are returned to the server and joined (somehow) to obtain the complete solution. In most cases, the more computers we have, the faster problems are solved. The only exception could be the situation in which the communications are slow or the number of communications per second necessary to deal with the problem could increase above the computing time in a single processor.

In 1993-94, the message-passing interface standard was developed to tap on the incipient multiple processor capabilities available at the time [26]. In recent years, a different form of distributed computing has become increasingly popular, the so-called desktop grid, that allows to take advantage of the resources of a set of computers (e.g., part-time idle computer rooms in the universities) joining them together into a computing network. One of the most known projects using this platform is SETI, dedicated to search for extraterrestrial life by analysing radio signals [27]. The software that makes possible to develop this type of distributed computing projects is BOINC [24]. However, the BOINC setting up is complex, demands advanced knowledge of system administration, software, and web development in a deployment process that may last several months [28]. BOINC is a solution that does not require a supercomputer, with many processors on the same motherboard, as MPI because it relies on the distributed computing over many independent personal computers connected throught the Internet.

2.1. Sisifo: Our Design Proposal. In the context described in the previous section, we have developed Sisifo [29]. Sisifo
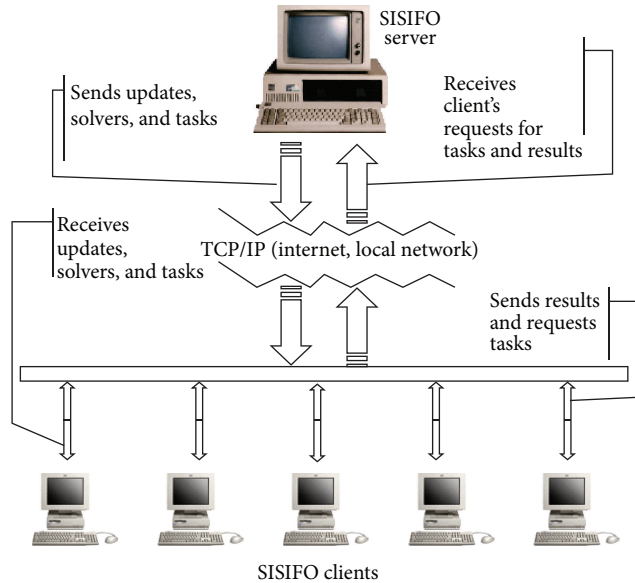
Figure 1: Operation scheme of Sisifo.

Table 1: An example of task file for the Sisifo project.

Task_ID	Prueba10372
RAM_needed	223672
Simulation_days	2555
No_realisations	1
No_nodes	1000000
Node_average_degree_(k)	29
Average_from_infected_to_recovered_(days)	10
Average_from_recovered_to_susceptible_(days)	198
Parameter_b	0.00311
Initial_percentage_of_susceptibles	0.99
Initial_percentage_of_infected	0.01
Initial_percentage_of_recovered	0

is our proposal for a distributed computing system, quite simpler than BOINC, requiring less resources, and providing a significantly quicker development, implementation, and deployment. Although our first objective is to use Sisifo to estimate parameters in epidemic models, of course it can also be used to solve other kinds of problems if they match the requirements for distributed computing. Taking into account that we have previous experience in the development of client/server systems with centralized databases and TCP/IP communication [30], we have used that background to build Sisifo. The operation scheme of Sisifo is shown in Figure 1.

To implement the Sisifo system, we need the following strategy: the problem we want to solve (e.g., simulation of the propagation of an epidemic in different conditions) should be divided into independent tasks, in the sense that they can be computed in a client separately from the others. Tasks are coded in text files and they indicate all necessary data to start the running of a simulation once received by the clients. An example of task for the epidemic model to be discussed in the next section is given in Table 1.

In this file, we detail the number of days of evolution for the epidemic we are simulating, the number of nodes of the network, and the average connectivity degree for the random network as well as the transition probabilities for the susceptible-infected-recovered-susceptible (SIRS) model. Initial conditions for the percentage of susceptibles and infected individuals are also provided.

Each client computer has a Sisifo client installed which requests tasks to the server. The server provides to the client a package containing the solver (the computer program which runs the simulation) and the task to be solved. The task is then marked as pending by the server. Once the client has received the package, it runs the solver with the data provided by the task and waits for the solver to finish. Then, the client takes the obtained results and transmits them back to the server. The results are received by the server and stored and the related task is marked as done. Then, the client requests a new task and the cycle starts again until all tasks have been solved.

Taking into account that we use client computers under our supervision, it is not necessary to include in Sisifo the management of users, statistics, and security control which BOINC does.

The minimum server requirements are as follows:

(i) storage of tasks and their state (undone, pending, and done);

(ii) storage of the solver. In fact, two solvers: one for MS-Windows and one for Linux;

(iii) storage of the client updates (if required, also for MS-Windows and Linux);

(iv) admission of client requests via TCP/IP in a certain (configurable) port. Several Sisifo servers may be installed in the server computer, each one listening to its own port for requests from the clients;

(v) processing clients requests. The client identifies itself providing its version and platform (MS-Windows or Linux) and demands a new task to the server or returns the results to the server;

(vi) answer the clients by, at least,

  (a) providing the pair solver+task, taking into account the operating system of the client,

  (b) providing an updated version of the client software when the current client version is older than the one stored in the server,

  (c) accepting the results at the end of a task execution and informing if the transmission has been completed successfully in order to retransmit in case of errors,

  (d) managing the time when the processing task expires. The server marks the transmission date of each package (solver + task), and in case of an excessive delay (configurable) occurs in the resolution reception; this leads the server to mark the task as undone and resend it to another client,

(e) creating a record with solved tasks and time used,

(f) showing in real-time basic information about connected clients, served tasks, and solved tasks,

(g) checking the validity of the results, because otherwise the results should be rejected and sent to be computed again,

(h) verifying data communication integrity, that is, via CRC.

Moreover, the minimum client requirements are as follows:

(i) use the minimum resources as possible,

(ii) create a TCP/IP connection to the server to do requests,

(iii) process the server messages, including

(a) the solver preparation and its execution with the corresponding task,

(b) the verification of a proper execution of the solver and the correct transmission of the results to the server.

Furthermore, the solver does not have special requirements to be executed by Sisifo apart from some conventions with input/output data format.

Note that emphasis has been put in the simplicity and development time because the network is under our supervision. Of course, this development may be not suitable for a wide network of unreliable clients.

2.2. Distributed Computing throughout Internet. Every task corresponds to the evolution of the disease for a given set of parameters for the epidemic model as shown in Table 1 and a given network configuration.

Although the first experiment with Sisifo gave us satisfactory results in a very short period of time, during the computations, a lot of tasks were unsuccessful because with our first set of tasks, the total number of infected individuals became zero, and therefore, nobody could be infected. We needed a deeper search into the solution space to find what parameters were able to produce a network where the disease did not die out. This required processing a much higher number of tasks, but the intranet where Sisifo was running did not have enough power to cope with all computations in a reasonable time and we wondered if we could boost this.

For this reason, we decided to try out BOINC. BOINC is an open source software actively maintained and used by the scientific community to manage projects of distributed computing as the abovementioned SETI@home [31, 32], ROSETA [33], or Climate Prediction [34]. BOINC protects against several types of attacks and the distribution of viruses using digital signatures based on public-key encryption; its server architecture is highly scalable and the core client is available for most common platforms. These are some of its main features that assure the correct transmission and reception of the task results [35].

To do that, we requested the help of the Falua project [36, 37], an initiative supported by the CES Felipe II of Aranjuez (campus of the Universidad Complutense de Madrid) which provides ad hoc BOINC deployment and computing power for small to medium computation problems. They adapted our solver to the BOINC platform and opened the possibility of foreign collaboration from the BOINC community.

With the increased computation power of the many volunteers of the BOINC community, we were able to simulate more than 145000 tasks in a short time. With this number of tasks processed, a reliable picture of the phase diagram of the model in the plane k-b, where k is the average degree of connectivity of the network and b is the transmission probability, was obtained. The processing of this large number of tasks is attainable because we have sustained peak performance over 370 Gigaflops in contrast with the 225 Gigaflops (at best and during short periods of time) of the Sisifo project, where only 80 computers were dedicated to the project.

3. Epidemic Transmission Dynamics in a Random Network Model

Seasonal pandemics are a serious public health problem. In particular, respiratory infections typically peak in their incidence once a year during the autumn-winter period. These epidemics' effects are particularly acute in young children and the elderly whose immune systems are either underdeveloped or deteriorated.

Among them, the respiratory syncytial virus (RSV) is one of the most severe causes of respiratory disease epidemics in many countries. RSV is a single-stranded RNA virus discovered more than fifty years ago in a child with bronchiolitis [38]. This virus is the cause of a seasonal epidemic in many countries all around the world. Only in Spain, there are around 15,000–20,000 visits to primary care due to RSV every year. Also, up to 18% of the pneumonia hospitalisations of individuals older than 65 are due to RSV [38]. This epidemic is also a major concern in immunocompromised patients at any age [38–40].

Its coincidence with other seasonal epidemics such as influenza and rotavirus produces a large number of hospitalisations year after year, saturating the National Health System. In particular, the cost of pediatric hospitalisation for the Health System of the Community of Valencia (Spain) [41] has been estimated to be 3.5 million euros per year [42] without taking into account indirect costs [43], with a cohort of newborns of around 45,000 children.

Epidemiological continuous models dealing with RSV have been proposed in [43–45], for instance. To our knowledge, only a network model of RSV has been studied in [15, 46].

Alternatively, other network substrates could be used as small-world or scale-free networks. However, respiratory infectious diseases are transmitted by random encounters among people in their social environment including public transportation and schools. For these reasons, a random network model seems reasonable and we have found that

it simulates the epidemic propagation at least as efficiently as traditional continuous differential models. On the other hand, scale-free networks only seem realistic for the propagation of sexually transmitted diseases [47, 48], where a few individuals with a large number of connections are the main hubs for the transmission of the disease. Transmission of respiratory infectious diseases is mainly mediated by random close spatial encounters among people and they do not rely heavily on stable social links (family, friends, work, etc.); for this reason, the random network model seems suitable as a first approach.

3.1. Population Model.

For a realistic simulation of a disease which affects, with different degrees of severity, all age groups of society, we need to incorporate a reasonable population model. We have retrieved mortality rates for Valencia from the Valencian Institute of Statistics [49] and simulated a Forster-McKendrick dynamic model with constant population [20]. The contribution of the disease itself to death rates is not taken into account because it is marginal in comparison with other causes [50]. This is implemented in the random network by the following algorithm.

(i) Every individual in the network has an age assigned to him/her following the Forster-McKendrick model. The age is increased in every time-step (our unit of time is 1 day).

(ii) Every time step we check whether the subject dies or survives until the next time-step. This is calculated by generating a pseudorandom number and comparing it with the mortality rate per day corresponding to the age of the subject. This is performed in a daily basis because the natural scale of the problem, that is, the average duration of the disease, is 10 days [15].

(iii) If a subject dies, he or she is replaced by a susceptible newborn. This way, the population remains constant by definition.

A warming-up period is allowed for the population pyramid to stabilize and the epidemic propagation is simulated afterwards. The results for the Autonomous Region of Valencia are shown in Figure 2.

3.2. Random Network.

Random networks are characterised by the number of individuals or nodes N, usually large (in our case a million nodes), and the average number of contacts of every individual, k (called the degree of this node). Consequently, the number of ties in the network is given by $N \times k/2$. These links are randomly assigned to pairs of individuals with the obvious rule that, at most, only a tie can connect two individuals. The degrees of the nodes in the resulting random network follow a Poisson distribution with mean k. We must take into account that random networks do not capture some structures found in real networks such as degree heterogeneity (different social groups could be connected with different average degrees) or communities (some social groups form connections mainly within the same group). However, a random network provides a first
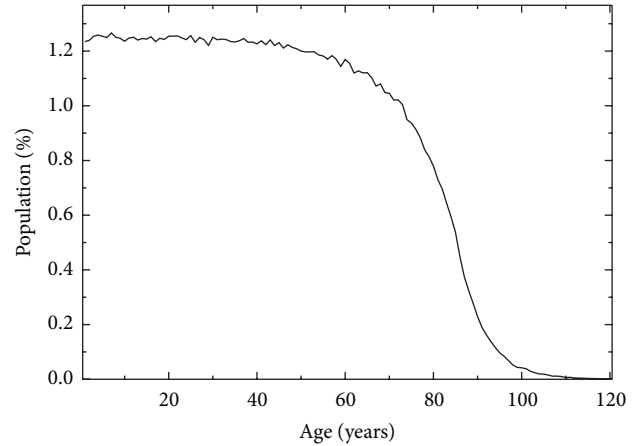

Figure 2: Percentage of persons with a given age in years obtained from the Foerster-McKendrick model after the warming-up period is performed.

approach to the modelling of epidemic transmission in complex systems and it constitutes a good starting point for testing simulation techniques.

An initial state in which 1% of the individuals are infected is chosen as the initial state (99% of the individuals are susceptible and no recovered individuals are found at the beginning). The age of the nodes is chosen from the warmed-up population pyramid in Figure 2. Then, the evolution algorithm of the RSV model proceeds as follows.

(1) Infected individuals recover following an exponential distribution of mean 1∕10, because 10 days is the average time to recover from the illness.

(2) Recovered individuals become susceptible again following an exponential distribution of mean γ per time step, where $1/\gamma$ days is the average time an individual remains immune against re-infection. Weber et al. proposed a model in which $\gamma = 1/200$ [44], however we will find a slightly different value after fitting hospitalization data for RSV in Valencia. In order to perform this fitting we will use the BOINC implementation of the model.

(3) The main difference with respect to the standard continuous model is found in the infection procedure: every susceptible individual can only be infected by infected individuals connected through existing ties with him or her. This occurs with a probability b per time step (one day in our case) in each contact, that is, every infected individual can infect a neighbouring susceptible individual with probability b per unit time. To simulate this process we select a random number, x, for every link connecting an infected with a susceptible individual. If $x < b$ the susceptible individual becomes infected. The same node cannot be infected twice, so if it becomes infected we stop the checking algorithm. The network is stored by means of the efficient algorithm of adjacent lists typically

used in graph theory [51]. By using this algorithm the links are stored in a list of $(k + 2)N/2$ integers.

(4) The age of the individuals is increased in one week on every time-step, that is, our time-step is one week because the duration of the infectious disease (the average time before recovering from the infection) is 10 days and we need a sufficiently small time-step for the model to follow the recovery process.

Then, for each node, for every day, we simulate the disease evolution following the above rules, obtaining the number of susceptible, infected, and recovered for all days at any age. By drawing these data, we can visualise the behaviour of the disease spread over time.

The third step of the algorithm, corresponding to the propagation of the infection, is extremely time consuming: the computer program must check the evolution of the state of every individual on a very large set by analysing the propagation of the infection through the ties with every infected individual in its neighbourhood. The use of adjacent lists allows us to efficiently store and identify the neighbourhood of a site.

Note that parameters k and b should be estimated by fitting the random network model with real data. The above description of the model leads to the fitting of two parameters, but note that b may depend on age, gender, or other situations, in which case the number of parameters will increase.

In order to estimate the unknown parameters, we should choose a feasible parameter space and build a mesh, where, for each mesh point (k, b), a simulation, following the disease rules described above, should be carried out and the results have to be compared to determine its closeness to the real data in the least square sense. The number of simulations depends on the mesh-point size, but in order to explore sufficiently the chosen parameter space, this number must be necessarily high, and here is where the computing distributed environment can help us to obtain the best k and b.

3.3. An Experiment with Sisifo. Two part-time people have developed, implemented, and tested the client and the server in less than three weeks using open source tools. The server has been installed in a personal computer Pentium IV with 1 GB of RAM.

The solver has to be developed specifically for each experiment. For this one in particular, we considered an SIRS random network model applied to the transmission dynamics RSV. The solver implementation for RSV in order to estimate the parameters b (transmission rate) and k (average node degree) has taken ten weeks. Then, we prepared 60 120 tasks for

  (i) a million nodes, divided into age groups following a Foster-McKendrick constant demographic model as described in Section 3.1. In this model individuals within the same year of age constitute an age-group,

 (ii) each node that is labelled by age and state respect to RSV,

(iii) a Poisson nodes distribution with mean $k = 5, 6, \ldots,$ 124,

 (iv) a transmission rate b from 0 to $1/200$ with steps of 10^{-5},

  (v) an average time of infection of 10 days,

 (vi) an average time of immunity after the infection of 200 days according to Weber et al. [44],

(vii) an initial situation where all the population is susceptible, but for 1% of infected individuals.

After that, we started the computing process, installing Sisifo server with all the tasks and solver in the server PC. The number of simultaneous clients has been varied from 10 (usually) up to 150 (during week-ends). Performance achieved the equivalent of more than three years of computing time in just five weeks. To calculate this performance, we must notice that the 60 120 tasks lasted an average of 25 minutes each on every Dual Core 3 GHz computer.

4. BOINC: Continuing the Experiment

The 145 099 tasks were generated by combining k (in the range 25–60), the transmission probability in a person-to-person contact ($0.00216 \leq b \leq 0.005208$ with 10^{-5} jumps), and the average immunity time after infection, 80 days $\leq 1/\gamma \leq 200$ days, with jumps of 3 days. We wanted to consider an interval range for γ because there is no consensus on what is the good value for the recovering rate [43, 44]. These important data for the epidemiology of RSV can be derived from our model fitting to the number of hospitalisations due to RSV infection [42, 43].

In this case, we restricted the intervals for b and k because our main goal is to find the best solution fitting hospitalisation data, varying also the average immunity time after infection, γ. From the experiments with Sisifo, we know that the best solutions always lie in this region. In order to achieve a more detailed exploration of this region, we also increased the number of tasks. Each task corresponds to a different set of parameters of the model as explained in Section 2.1. For each set of parameters, we perform a simulation over 10 different realizations of the random network.

Each task needed an average of 80 minutes for completion and this would have needed more than twenty years of computing time in a single computer dedicated to the task, but using the BOINC distributed computing, we only needed 3 weeks. When the BOINC project was made available to the international community, we got an average of 800 computers connected on peak performance. This made 8500 successful tasks completed each day.

In Figure 3, we have plotted the number of tasks that were carried out successfully each day in the first phase of the BOINC experiment.

The best fit was obtained for the following values: $b =$ 0.00267 (a mean of 267 per 100 000 successful contacts per day), $k = 54$ (a mean of 54 contacts, successful or not, per individual per day), and $s = 4.979$ (4.979% of the weekly infected children under one year old are hospitalized). The immunity period was found to be $1/\gamma = 188$ days (very close to the prediction of continuous models [44]); this value was obtained as a consequence of the best fit of hospitalization
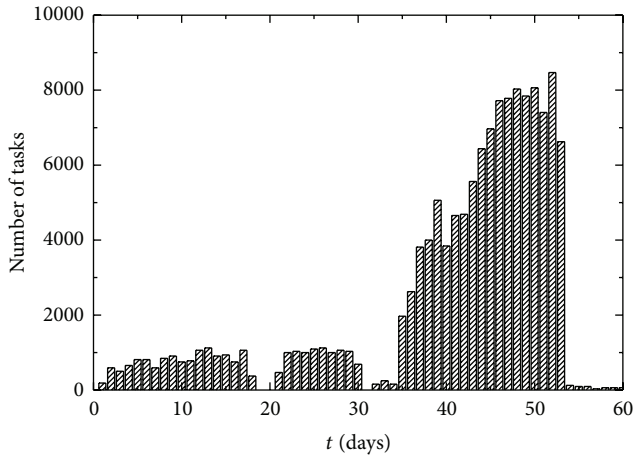
FIGURE 3: The number of successful tasks completed versus time (in days) for the first phase of the BOINC experiment.
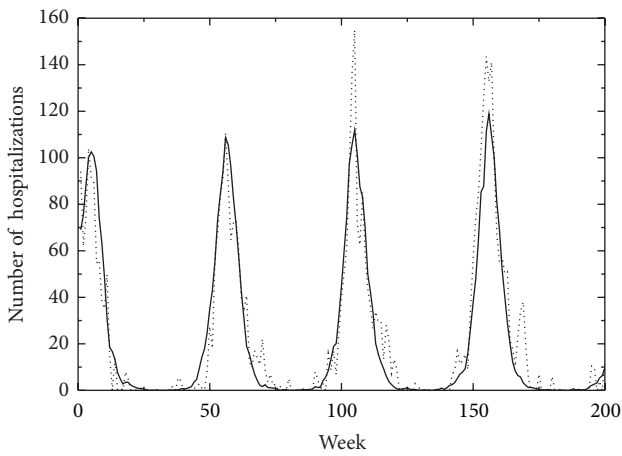

FIGURE 4: The number of hospitalisations of children under 1 year of age in the Spanish region of Valencia: real data (dotted line) and fitting corresponding to the random network model (solid line). The period of time goes from January 2001 to December 2004.

FIGURE 5: The probability for the emergence of a seasonal pattern in a network with N nodes and the parameters for RSV propagation fitted to the Spanish region of Valencia data. Circles are simulation data; the continuous line is an exponential fit.

data in contrast to previous works, where γ was a fixed parameter. The mean quadratic error (MSE = 11.38) is even better than the one obtained by fitting the continuous models [43]. The fitting is depicted in Figure 4. Notice that the parameter s, the fraction of infected children under one year old who become hospitalised as a consequence of RSV infection, is also a fitting parameter. We considered a fraction s of the children aged one year old or less and compared it with the real data to find the optimum result.

The interesting fact about the results depicted in Figure 4 is that the oscillatory behaviour was obtained without resorting to external forcing as usually considered in continuous models [43–45]. Further analysis of the phase diagram of the random network model has already been given [46]. However, it is difficult to attribute the origin of the seasonal behaviour to purely intrinsic dynamics. This is an open problem and many authors believe that the seasonal outbreaks are
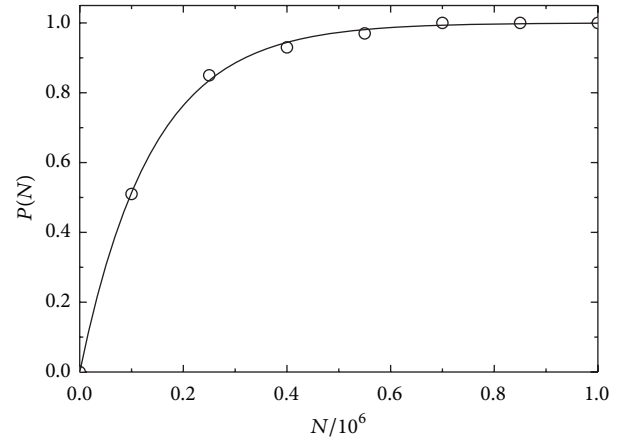
provoked by a slight change in the transmission rate correlated with atmospheric factors [52, 53], although Dushoff et al. have also suggested that dynamical resonance can account for seasonal behaviour of respiratory pandemics (including influenza) without any detectable change in the transmission rate [54]. We have shown that without a seasonal external forcing the model can fit the data for hospitalisations of children under 1 year of age.

Our distributed computing solutions allow us to address another important epidemiological problem. It is useful to determine the size of the population capable of sustaining the seasonal behaviour for a long period of time. By using the parameters corresponding to the disease as fitted in Figure 4, we have reduced the population size and calculated the probability that the seasonal behaviour emerges. Otherwise, the epidemic becomes extinct. We consider several population sizes: 100000, 250000, 400000, 550000, 700000, 850000, and 1000000. For every population size, we build 100 random networks with the parameters corresponding to the fitting of the seasonal epidemic and observe the emergence of oscillatory behaviour in any of these realizations. The fraction of realizations in which a seasonal pattern is observed gives us the probability $P(N)$. Results are plotted in Figure 5.

A good fitting is found by the simple exponential function which we propose heuristically:

$$P(N) = 1 - e^{-N/M}, \tag{1}$$

where M depends on the parameters of the model, but we have not considered this dependence because we are interested only in a particular set of parameters, that is, those corresponding to the fitting of the seasonal epidemic. In this particular case $M = 138\,480$, so we can conclude that a population similar to 1000000 nodes is necessary to sustain the oscillatory behaviour. This result presupposes an ideally isolated population and does not take into account the fact that the virus can be reintroduced after extinction by migratory displacement of nodes.

5. Conclusion

In this paper, we have discussed two computing distributed techniques in order to accelerate the simulation of epidemic models in very large random networks. The first one is an intranet solution (Sisifo) which is simpler to implement and to operate, while the second one is based on the BOINC software and has the advantage of being distributed throughout the whole Internet with the help of volunteers.

Both of them have their advantages and disadvantages and should be chosen depending on the problem and the availability of resources. Sisifo is an intranet solution for distributed computing independently developed for this project, where its main advantages are its fast deployment and easy management. Being an intranet solution, we can control every computer in the project and avoid the complications of security issues. However, we must have a large number of sufficiently efficient computers at our disposal. BOINC allows us to distribute the computation throughout the whole internet, but its deployment is far more complicated and the resources availability is highly variable. For this reason, Sisifo could be a useful solution in many problems which do not require so large computational capabilities as those attained with BOINC. We have shown this by studying the fitting problem for our epidemiological model entirely with Sisifo, using only BOINC for a more accurate calculation involving an additional fitting parameter, γ, the probability, per unit time, for a recovered individual to become susceptible again.

By using this distributed architecture, we have been able to tackle the problem of double statistical average in networks: we consider a set of different random networks characterized by the connectivity, k, and for each of them, a simulation of the propagation of the epidemic is performed. Then, an average over the propagation of the disease in a random network with a given k can be given. This is a fundamental statistical problem in many models arisen in statistical physics and our solution provides an efficient way to implement a computational solution even for very large systems. In order to test its efficiency, we have considered epidemic propagation in random networks as an example.

Random networks provide an alternative to continuous differential models in the simulation of disease transmission dynamics. However, they demand a high amount of computations in parameter estimation, simulations, and so forth.

To test the feasibility of this approach, we performed an experiment in a random network of a million nodes in order to simulate the spread of RSV in the region of Valencia (Spain). Then, computations were carried out to estimate the RSV parameters b, to determine the probability of successful transmission contacts, k, and to determine the average number of individual daily contacts, and γ, the immunity period after infection, as well as the fraction of infected children under one year of age who became hospitalised, s, in a matter of a few weeks by using a distributed computing network, instead of requiring several years for a single PC.

In these simulations, we have shown the emergence of seasonal patterns in the incidence of infections which do not require any external forcing. This could provide a social network explanation to seasonal patterns without the need of invoking weather effects. Moreover, we have shown that a population around one million persons is required for the RSV pandemic not to become extinct in the long run. This result was obtained by considering 100 realizations of the random network for the epidemic model parameters corresponding to the fitting solution of the real hospitalisation data. The stationary state achieved for an initial 1% of infected individuals could be extinction or seasonal behaviour, the fraction $P(N)$ of realizations in which seasonal behaviour is observed to rise progressively from a 50% around $N = 150000$ to 100% at $N = 1000000$.

Now, once the model has been parametrised, we are ready to design specific or targeted vaccination or prophylactic strategies and evaluate their effectiveness.

The next step in our research will consist of the inclusion of an optimization system using genetic algorithms (optimized for the distributed computing environment), for instance, in order to reduce the amount of computations so as to get the optimum and then be capable of facing the fitting of network models with more than two or three parameters to be estimated, as well as to increase the number of nodes. The simulation method discussed in this paper can also be applied to many other similar problems. Currently, we are considering meningococcal C disease spread and vaccination strategies as well as a network model of the brain [55]. Migration and diffusion can also be considered to take into account spatial dynamics effects [56]. Results for these projects will be published elsewhere.
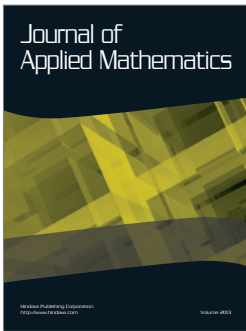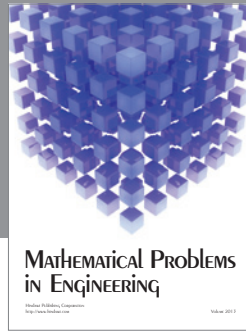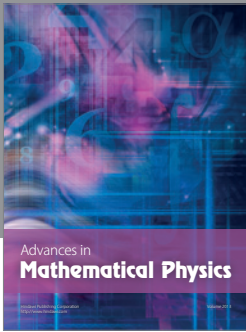
Acknowledgments

References

[1] S. R. Proulx, D. E. L. Promislow, and P. C. Phillips, "Network thinking in ecology and evolution," *Trends in Ecology and Evolution*, vol. 20, no. 6, pp. 345–353, 2005.

[2] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of Facebook networks," *Physica A*, vol. 391, pp. 4165–4180, 2012.

[3] M. J. van der Leij, *The economics of networks: theory and empirics [Ph.D. thesis]*, Tinbergen Institute Research Series, Amsterdam, the Netherlands, 2006, http://repub.eur.nl/res/pub/8212/.

[4] Y. Bar-Yam, *Dynamics of Complex Systems*, Addison-Wesley, Reading, Mass, USA, 1997.

[5] N. A. Christakis and J. H. Fowler, "The collective dynamics of smoking in a large social network," *New England Journal of Medicine*, vol. 358, no. 21, pp. 2249–2258, 2008.

[6] N. A. Christakis and J. H. Fowler, "The spread of obesity in a large social network over 32 years," *New England Journal of Medicine*, vol. 357, no. 4, pp. 370–379, 2007.

[7] M. E. Halloran, I. M. Longini Jr., A. Nizam, and Y. Yang, "Containing bioterrorist smallpox," *Science*, vol. 298, no. 5597, pp. 1428–1432, 2002.

[8] J. Koopman, "Epidemiology: controlling smallpox," *Science*, vol. 298, no. 5597, pp. 1342–1344, 2002.

[9] S. Wolfram, "Cellular automata and complexity: collected papers," Stephen Wolfram, 1994, http://www.stephenwolfram.com/publications/books/ca-reprint/.

[10] E. Ahmed and H. N. Agiza, "On modeling epidemics. Including latency, incubation and variable susceptibility," *Physica A*, vol. 253, no. 1–4, pp. 347–352, 1998.

[11] R. M. Z. dos Santos, "Immune responses: getting close to experimental results with cellular automata models," in *Annual Reviews of Computational Physics VI*, D. Stauffer, Ed., 1999.

[12] M. L. Martins, G. Ceotto, S. G. Alves, C. C. B. Bufon, J. M. Silva, and F. F. Laranjeira, "Cellular automata model for citrus variegated chlorosis," *Physical Review E*, vol. 62, no. 5, pp. 7024–7030, 2000.

[13] U. Hershberg, Y. Louzoun, H. Atlan, and S. Solomon, "HIV time hierarchy: winning the war while, loosing all the battles," *Physica A*, vol. 289, no. 1-2, pp. 178–190, 2001.

[14] G. Witten and G. Poulter, "Simulations of infectious diseases on networks," *Computers in Biology and Medicine*, vol. 37, no. 2, pp. 195–205, 2007.

[15] L. Acedo, J.-A. Moraño, and J. Díez-Domingo, "Cost analysis of a vaccination strategy for respiratory syncytial virus (RSV) in a network model," *Mathematical and Computer Modelling*, vol. 52, no. 7-8, pp. 1016–1022, 2010.

[16] C. L. Barrett, K. R. Bisset, S. G. Eubank, X. Feng, and M. V. Marathe, "EpiSimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks," in *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing (SC '08)*, pp. 37:1–37:12, IEEE Press, Piscataway, NJ, USA, 2008, http://portal.acm.org/citation.cfm?id=1413370.1413408 .

[17] W. O. Kermack and A. G. McKendrick, "Contributions to the mathematical theory of epidemics-I," *Bulletin of Mathematical Biology*, vol. 53, no. 1-2, pp. 33–55, 1991.

[18] L. Edelstein-Keshet, *Mathematical Models in Biology*, SIAM, Philadelphia, Pa, USA, 2005.

[19] J. D. Murray, *Mathematical Biology: I. An Introduction*, Springer, Berlin, Germany, 2002.

[20] H. W. Hethcote, "Mathematics of infectious diseases," *SIAM Review*, vol. 42, no. 4, pp. 599–653, 2000.

[21] B. Bollobas, *Random Graphs*, Cambridge University Press, Cambridge, UK, 2nd edition, 2001.

[22] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[23] D. J. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*, Princeton University Press, Princeton, NJ, USA, 2003.

[24] BOINC, "Open-sources of tware for volunteer computing and grid computing," 2011, http://boinc.berkeley.edu/.

[25] D. ben-Avraham and S. Havlin, *Diffusion and Reactions in Fractal and Disordered Systems*, Cambridge University Press, Cambridge, UK, 2000.

[26] P. S. Pacheco, *Programming Parallel with MPI*, Morgan Kaufmann, San Francisco, Calif, USA, 1997.

[27] B. McConnell and C. Toporek, *Beyond Contact: A Guide to SETI and Communicating with Alien Civilizations*, O'Reilly Media, Sebastopol, Calif, USA, 2001.

[28] "Over view of BOINC," 2011, http://boinc.berkeley.edu/trac/wiki/BoincIntro .

[29] "For further details and to download Sisifo software under request visit," http://sisifo.imm.upv.es/.

[30] J. Villanueva-Oller, R. J. Villanueva, and S. Díez, "CASANDRA: a prototype implementation of a system of network progressive transmission of medical digital images," *Computer Methods and Programs in Biomedicine*, vol. 85, no. 2, pp. 152–164, 2007.

[31] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky, "SETI@HOME-massively distributed computing for SETI," *Computing in Science and Engineering*, vol. 3, no. 1, pp. 78–83, 2001.

[32] "SETI@home," 2011, http://setiathome.berkeley.edu/.

[33] "ROSETTA@home," 2011, http://boinc.bakerlab.org/.

[34] "Climate prediction," 2011, http://climateprediction.net/.

[35] "The Respiratory Syncytial Virus BOINC project," http://falua.cesfelipesegundo.com/VRS.

[36] D. Martin, J. Villanueva-Oller, I. Hidalgo, M. Alberquilla, and I. Contreras, *Anales de Ingenieria Tecnica Informatica de Sistemas 3, CES Felipe II*, Virtual supercomputers and distributed computing with BOINC, Berkeley, Calif, USA, 2010.

[37] J. Villanueva-Oller, D. Martin, I. Hidalgo, M. Alberquilla, and I. Contreras, "El proyecto Falua: computacion distribuida mediante BOINC en el campus de Aranjuez de la UCM (Falua project: distributed computing using BOINC in the Aranjuez campus of UCM)," in *Proceedings of the 21th Congreso Espanol de Informatica*, CEDI 2010, Ed., Jornadas de Paralelismo, Valencia, Spain, September 2010.

[38] C. Hall, *Textbook of Pediatric Infectious Diseases*, Respiratory Syncytial Virus and Human Metapneumovirus, Saunders, Philadelphia, Pa, USA, 5th edition, 2004.

[39] C. B. Hall, K. R. Powell, and N. E. MacDonald, "Respiratory syncytial viral infection in children with compromised immune function," *New England Journal of Medicine*, vol. 315, no. 2, pp. 77–81, 1986.

[40] A. R. Falsey and E. E. Walsh, "Respiratory syncytial virus infection in adults," *Clinical Microbiology Reviews*, vol. 13, no. 3, pp. 371–384, 2000.

[41] http://en.wikipedia.org/wiki/Valencian_Community.

[42] J. Díez Domingo and M. Ridao López, "Incidencia y costes de la hospitalización por bronquiolitis y de las infecciones por virus respiratorio sincitial en la Comunidad Valenciana," *Anales de Pediatria*, vol. 65, no. 4, pp. 325–330, 2006.

[43] L. Acedo, J. Díez-Domingo, J.-A. Moraño, and R.-J. Villanueva, "Mathematical modelling of respiratory syncytial virus (RSV): vaccination strategies and budget applications," *Epidemiology and Infection*, vol. 138, no. 6, pp. 853–860, 2010.

[44] A. Weber, M. Weber, and P. Milligan, "Modeling epidemics caused by respiratory syncytial virus (RSV)," *Mathematical Biosciences*, vol. 172, no. 2, pp. 95–113, 2001.

[45] L. White, J. Mandl, M. Gomes et al., "Understanding the transmission dynamics of respiratory syncytial virus using multiple time series and nested models," *Mathematical Biosciences*, vol. 209, no. 1, pp. 222–239, 2007.

[46] L. Acedo, J.-A. Moraño, R.-J. Villanueva, J. Villanueva-Oller, and J. Díez-Domingo, "Using random networks to study the dynamics of respiratory syncytial virus (RSV) in the Spanish

region of Valencia," *Mathematical and Computer Modelling*, vol. 54, no. 7-8, pp. 1650–1654, 2011.

[47] A. Schneeberger, R. Nat, C. H. Mercer et al., "Scale-free networks and sexually transmitted diseases: a description of observed patterns of sexual contacts in Britain and Zimbabwe," *Sexually Transmitted Diseases*, vol. 31, no. 6, pp. 380–387, 2004.

[48] J. Lou and T. Ruggeri, "The dynamics of spreading and immune strategies of sexually transmitted diseases on scale-free network," *Journal of Mathematical Analysis and Applications*, vol. 365, no. 1, pp. 210–219, 2010.

[49] "Instituto Valenciano de Estadýstica," 2011, http://www.ive.es/.

[50] D. M. Fleming, R. S. Pannell, and K. W. Cross, "Mortality in children from influenza and respiratory syncytial virus," *Journal of Epidemiology and Community Health*, vol. 59, no. 7, pp. 586–590, 2005.

[51] S. Sahni, *Concepts in Discrete Mathematics*, The Camelot, Somerset, UK, 1985.

[52] T. J. Meerhoff, J. W. Paget, J. L. Kimpen, and F. Schellevis, "Variation of respiratory syncytial virus and the relation with meteorological factors in different winter seasons," *Pediatric Infectious Disease Journal*, vol. 28, no. 10, pp. 860–866, 2009.

[53] R. C. Welliver, "Temperature, humidity, and ultraviolet B radiation predict community respiratory syncytial virus activity," *Pediatric Infectious Disease Journal*, vol. 26, supplement 11, pp. S29–S35, 2007.

[54] J. Dushoff, J. B. Plotkin, S. A. Levin, and D. J. D. Earn, "Dynamical resonance can account for seasonality of influenza epidemics," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 48, pp. 16915–16916, 2004.

[55] L. Acedo, J. Villanueva-Oller, J. A. Morano, and R. -J. Villanueva, "The Neurona@Home project: simulating a large- scale cellular automata brain in a distributed computing environment," in *Proceedings 12th Granada Seminar: Physics, Computation and the Mind*, 2013, http://falua.cesfelipesegundo.com/Neurona/ .

[56] J. Arino, J. R. Davis, D. Hartley, R. Jordan, J. M. Miller, and P. van den Driessche, "A multi-species epidemic model with spatial dynamics," *Mathematical Medicine and Biology*, vol. 22, no. 2, pp. 129–142, 2005.

Advances in
Mathematical Physics

The Scientific World Journal

Mathematical Problems in Engineering

Advances in
Decision Sciences

International Journal of
Combinatorics

Journal of
Applied Mathematics

Abstract and
Applied Analysis

Discrete Dynamics in Nature and Society

Journal of Function Spaces and Applications

International Journal of
Mathematics and Mathematical Sciences

Advances in
Operations Research

International Journal of
Differential Equations

Journal of
Probability and Statistics

International Journal of
Stochastic Analysis

ISRN Mathematical Analysis

ISRN Discrete Mathematics

ISRN Algebra

ISRN Applied Mathematics

ISRN Geometry