Villanueva Micó, RJ.; Hidalgo, J.; Cervigon, C.; Villanueva-Oller, J.; Cortés, J. (2019). Calibration of an agent-based simulation model to the data of women infected by Human Papillomavirus with uncertainty. Applied Soft Computing. 80:546-556. https://doi.org/10.1016/j.asoc.2019.04.015

Calibration of an agent-based simulation model to the data of women infected by Human Papillomavirus with uncertainty

Rafael-J. Villanueva[a], J. Ignacio Hidalgo[b], Carlos Cervigón[c], Javier Villanueva-Oller[d], Juan-Carlos Cortés[a]

[a] *Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Valencia, Spain*
[b] *Departamento de Arquitectura de Computadores y Automática, Universidad Complutense de Madrid, Madrid, Spain*
[c] *Dpto. Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid, Madrid, Spain*
[d] *Dpto. CC. Computación, Arquitectura de la Computación, Lenguajes y Sistemas Informáticos y Estadística e Investigación Operativa, Universidad Rey Juan Carlos, Madrid, Spain*

Abstract

Recently, the transmission dynamics of the Human Papillomavirus (HPV) has been studied. In previous works, we have designed and implemented a computational model (agent-based simulation model) where the contagion of the HPV is described on a network of lifetime sexual partners. The run of a single simulation of this computational model, composed of a network with 500 000 nodes, takes about one hour and a half. In addition to set an adequate model, finding out the model parameters that best fit the proposed model to the available data of prevalence is a crucial goal. Taking into account that the necessary number of simulations to perform the calibration of the model may be very high, the aforementioned goal may become unaffordable. In this paper, we present a procedure to fit the proposed HPV model to the available data and the design of an asynchronous version of the Particle Swarm Optimization (PSO) algorithm adapted to the distributed computing environment. In the process, the number of particles used in PSO should be set carefully looking for a compromise between quality of the solutions and computation time. Another feature of the procedure presented here is that we want to capture the intrinsic uncertainty in the data (data come from a survey) when calibrating the model. To do so, we also propose the design of an algorithm to select the model parameter sets obtained during the calibration that best capture the data uncertainty.

Key words: HPV Transmission Dynamics, Computational Random Network Model, Model Calibration, Particle Swarm Optimization, Uncertainty Quantification, Agent-based simulation modeling.

1. Introduction

Human papillomaviruses (HPV) include more than 100 genotypes of viruses that infect cutaneous, genital and respiratory epithelia of humans. HPV is the most common sexually transmitted infection (STI) in the world. It is transmitted via vaginal, anal or oral sex with someone who has the virus [1]. In general, it is asymptomatic, inoffensive and disappears spontaneously. However, if it persists, it may develop anogenital warts, secondarily juvenile onset of recurrent respiratory papillomatosis, cervical cancer, other anogenital cancers and head and neck cancers [2].

There are two main types of HPV, high risk (HR) that are directly related to the origin of cancers and low risk (LR) related with anogenital warts and mucocutaneous lesions [3]. To fight against the infection, a vaccine has been developed. It protects against the HPV types 6, 11, 16, 18, 31, 33, 45, 52 and 58, responsible of 90% of genital warts and 90% of cancers [4]. In [5, 6] we have proposed a network computational model (agent-based simulation model) of lifetime sexual partners (LSP) to study the HPV dynamics.

In [5], we have performed a calibration of the agent-based simulation model. This calibration allowed to reproduce the Australian scenario [6], where girls aged between 12 and 13 years old were vaccinated and a catch-up vaccination on women aged 14 to 26 years old during two years was done. Two years after the vaccine was introduced, the proportion of diagnosed genital warts declined by a 59% in vaccine eligible young women aged 12 to 26 years in 2007, and by 39% in heterosexual men of the same age.

However, the calibration of this kind of complex random computational models is an open problem and several challenging issues about fitting computational models to data with uncertainty have to be addressed:

- the model is not deterministic and, for the same set of parameters, different runs of the computational model may return different outputs, and, consequently, one realization may fulfill the fitting requirements while another realization does not. Thus, the objective function is not the typical differentiable and closed-form function;
- the determination of an appropriate measure of goodness-of-fit is still needed;

- it is necessary to find model parameters in such a way that their values agree with the figures reported in medical studies, and the method be reliable and reproducible;
- the adaptation of the optimization algorithms to these above issues and the available resources is of special interest;
- the determination of the best parallel implementation of the optimization algorithm, in terms of quality of the solutions and computational efficiency.

In this paper, we present a procedure to fit this network computational model to the available data taking into account all the above issues, that is: first, determining the appropriate number of particles for PSO balancing the quality of the solutions and the computation time; second, designing an asynchronous PSO algorithm on a distributed computing environment to calibrate the model; third, designing a selection algorithm inspired by PSO that allows us to select model parameters that best capture data uncertainty.

The idea of using asynchronous PSO algorithms appears, among others, in the papers [7, 8]. In the latter, the authors show that in most cases, asynchronous updates save considerable time while not significantly impacting the probability of finding a solution.

We must say that we deal with a problem in epidemiology with high uncertainty. Therefore, we expect to obtain reasonable solutions explaining the HPV prevalence in women but not necessarily the optimal. Also, in the proposed algorithms, the computation time is important but not critical, because once we have obtained the calibration we will simulate possible public health scenarios with the obtained parameters. Furthermore, from a more theoretical point of view, some issues about accuracy and performance of the proposed procedure could be improved. Hence, our main goal is to provide a reliable answer to an epidemiological that, to the best of our knowledge has not been solved yet.

The rest of the paper is organized as follows. In Section 2 we describe the computational network model, we show the data to be fitted by the model and, using medical literature, we determine appropriate bounds for the model parameters. In Section 3, we describe the available computers and the distributed computing environment *Sisifo* we have used to perform the model simulations. Section 4 is devoted to the description of the Random Particle Swarm Optimization (rPSO) algorithm [9] and its asynchronous adaptation to the *Sisifo* distributed computing environment (arPSO). In

this section, we introduce a measure of goodness-of-fit that allows us to make specific changes in arPSO aimed to improve its performance.

Section 5 starts proposing a test able to determine the optimal number of particles in the arPSO, in terms of computational efficiency. Then, the calibration procedure is described. Afterwards, we propose a selection algorithm, inspired in PSO and applied to select the best 30 sets of model parameters that best capture data uncertainty. Conclusions are given in Section 6.

2. Description of the model

In papers [5, 6], we described how to build lifetime sexual partners (LSP) networks. These networks are built following the Spanish demographic structure [10] and the LSP structure provide by the survey about sexual habits in Spain [11], for age groups 18 to 29, 30 to 39 and 40 to 65, and collected in Table 1. In this table, figures express the proportion of subjects with none, one, two, three to four, five to nine, and ten or more sexual partners during his/her whole life. For instance, for the age group 30 to 39 around 21% of the males reported 3 or 4 LSPs (see 5th column, 3rd row in Table 1). Although the original work in [11] was done only for people older than 18, due to ethical issues of the study, we will assume that people aged 14 to 17 follows a similar pattern as ages 18 to 29.

Table 1: Proportion of males and females per number of LSP age group [11]. Note that the sum of the rows are 1.

MALES

Age	0 LSP	1 LSP	2 LSP	$3-4$ LSP	$5-9$ LSP	LSP ≥ 10
$14-29$	0.107	0.207	0.131	0.225	0.168	0.162
$30-39$	0.027	0.225	0.128	0.210	0.170	0.240
$40-65$	0.019	0.268	0.140	0.193	0.163	0.217

FEMALES

Age	0 LSP	1 LSP	2 LSP	$3-4$ LSP	$5-9$ LSP	LSP ≥ 10
$14-29$	0.138	0.43	0.186	0.158	0.056	0.032
$30-39$	0.029	0.501	0.168	0.177	0.077	0.048
$40-65$	0.017	0.652	0.138	0.118	0.039	0.036

There are many possible combinations to create networks matching the demography and the LSP distribution described above. As a consequence, the generated networks will contain uncertainty stemming from the buiding process itself and the different shapes these networks may have. Then,

over the LSP network, where each node represents a subject, we describe the dynamics of the HPV [5, 6]. Initially, each node i of the network, is characterized (labeled) by the following information:

- Gender: male, female or men who have sex with men (MSM). Females who have sex with females are not considered because the HPV prevalence is almost inexistent.
- Age Group. We consider the four age groups defined in the [11] to assign LSPs to the node:
  - Group 1: 14 to 17.
  - Group 2: 18 to 29.
  - Group 3: 30 to 39.
  - Group 4: 40 to 65.
- Age of the individual (node) in months.
- Number of expected LSPs for this node, LSP_i, i.e. number of connections of the node in the network.
- Probability of transmission of LR viruses in a sexual intercourse if one of the individuals is infected of LR.
- Probability of transmission of HR viruses in a sexual intercourse if one of the individuals is infected of HR.
- Infected by LR virus, yes or not.
- Time since infection by LR virus.
- Infected by HR virus: yes or not.
- Time since infection by HR virus.

In order to properly describe the transmission dynamics of HPV on the LSP networks, it is also necessary to include 11 model parameters that will govern the aforementioned HPV dynamics. Most of these parameters have been studied in the literature and there are some estimations we have to take into account:

- The average number of men LSP, that is, the average number of connections of all nodes, k. It is around 8 [12]. For this parameter, the search will be performed in the interval $(7, 10)$.

- Global probabilities in order to determine if the existence of a LSP implies sexual intercourses in the current time step per age group, i.e., P_g^1, P_g^2, P_g^3, and P_g^4.

- The average time T_{HR}^{cls}, an individual infected by a high risk HPV clears the infection and recovers. The time for clearing the infections due to HPV HR, for both men and women: in [13], the authors say that the average duration of HPV 16/18 infection is 1.2 years; in [14], the average duration of HPV 16 is 12.19 months $(7.16 - 18.17)$; in [15] the duration of HPV HR infection is between 6.5 months to 11.8 months. Thus, we consider the interval of $(0.8, 1.2)$ years.

- The average time T_{LR}^{cls} an individual infected by a low risk HPV clears the infection and recovers. The time for clearing the infections due to HPV LR, for both men and women: in [13], the mean duration of HPV 6/11 infection is 0.7 years; in [14], the mean duration of HPV infection is 7.52 months $(6.80 - 8.61)$ for any HPV; in [15] the duration of HPV LR infection varies depending the locations, in average, between 6.2 months to 11.7 months. Therefore, we consider an average clearing of any LR in the interval $(0.5, 1)$ years.

- The probability that a woman infected of high risk HPV transmits it to her partner in a sexual intercourse, P_{HR}^w.

- The probability that a woman infected of low risk HPV transmits it to her partner in a sexual intercourse, P_{LR}^w.

- The probability that a man infected of high risk HPV transmits it to his partner in a sexual intercourse, P_{HR}^m.

- The probability that a man infected of low risk HPV transmits it to his partner in a sexual intercourse, P_{LR}^m.

In [13], the authors estimated the probability of HPV infection transmission per partnership and by type and, as in [2], this probability is higher in the transmission from males to females (0.8) than in transmission from females to males (0.7). Given that these data come from estimations (not surveys) and after some runs of our model, we are going to be more flexible. Then, let us consider the probability interval $(0.2, 0.6)$ for LR transmission and $(0.5, 1.0)$ for HR transmission, given that, women transmit less than men.

Now, we are able to define the dynamics of HPV. Hence, we have implemented in C++ a simulator that, given the above described model parameters, builds a LSP network and performs a realization of the transmission dynamics of HPV HR and HPV LR. Algorithm 1 describes the HPV dynamics implemented in the simulator. For each node of the network we check if it is infected, if it clears the infection and if it will infect its connected nodes.

It is important to note that the simulation of the contagion in the network is made through the transmission parameters, with certain probabilities. Then, in order to see if a contagion has been carried out by a sexual intercourse, we simulate this event by generating random numbers and checking if they are less than the corresponding thresholds given by the model transmission parameters. Therefore, randomness is included into the model in a natural way producing uncertainties on the model output that have to be quantified. Thus, there are two sources of uncertainty: the LSP network building and the simulation of HPV dynamics. Thus, our goal is to find the model parameters in such a way that the computational model output is close, in a way we will describe later, to the data in Table 2. In this table, we recall the data presented in [2] related to the percentage of women HR- and LR-infected per group ages $18-29$ and $18-64$.

A set of parameters of the agent-based simulation model, $\overrightarrow{p_i}$, is represented by a vector of 11 parameter values:

$$\overrightarrow{p_i} = (k, P_g^1, P_g^2, P_g^3, P_g^4, T_{HR}^{cls}, T_{LR}^{cls}, P_{HR}^w, P_{LR}^w, P_{HR}^m, P_{LR}^m) \tag{1}$$

and then we evaluate the solution by running the simulation and observing the final disposition of the network.

3. Distributed computing environment

All the realizations were performed on two computers with 64 cores on 8 Xeon Sandy Bridge E5-4620 processors running at 2,2 Ghz, with 16 MB of cache memory and 512 GB RAM memory. The operating system is Ubuntu Server 16.04.3 LTS.

We also have deployed a distributed computing environment called *Sisifo*. *Sisifo* is a client-server based system designed to allow a problem to be solved using distributed computation. *Sisifo* is able to assign tasks to a set of personal computers (clients), wait for the tasks to be completed and collect the results for further analysis. *Sisifo* is made with simplicity as main goal, giving as a result a system that requires almost no maintenance, needs very little configuration time, and can be deployed in just a couple of hours.

```
   input  : Initial Network
   output: Network after simulation of M months
 1 for t ← 1 to #Months do
 2     for i ← 1 to #Nodes do
 3         Update Age of Node i;
 4         if Node is infected by LR-HPV then
 5             if Time since infection ≥ Time of clearance for LR then
 6                 Node i clears the LR infection;
 7                 Reset to 0 the time since LR infection;
 8             else
 9                 Time since LR infection is increased in 1 month
10             end
11         foreach LSP_j of Node i do
12             if LSP_j is infected of LR-HPV and
                GenerateRandom(0,1) < Probability of LR transmission
                then
13                 Node i gets infected of LR-HPV;
14             end
15         end
16         if Node is infected by HR-HPV then
17             if Time since infection ≥ Time of clearance for HR then
18                 Node i clears the HR infection;
19                 Reset to 0 the time since HR infection;
20             else
21                 Time since HR infection is increased in 1 month
22             end
23         foreach LSP_j of Node i do
24             if LSP_j is infected of HR-HPV and
                GenerateRandom(0,1) < Probability of HR
                transmission then
25                 Node i gets infected of HR-HPV;
26             end
27         end
28     end
29 end
```

Algorithm 1: Pseudocode of the dynamics of the HPV implemented in the C++ simulator.

Table 2: Prevalence of HR- and LR-infected women per age groups [2]. Mean and 95% confidence intervals. Co-infection is possible and it is considered in both, HR and LR.

Women	HR-infected	LR-infected
$18-29$ y.o.	24.10%, $[21.33\%, 26.98\%]$	6.36%, $[4.71\%, 8.07\%]$
$18-64$ y.o.	16.23%,$[14.52\%, 17.97\%]$	4.41%, $[3.42\%, 5.45\%]$

The *Sisifo* server keeps listening for requests of the clients. The Server has stored one or more executors, a set of problems to be solved in the *Problem files* folder, and the solutions returned by the clients in the *Result files* folder.

The *Sisifo* Client is a program stored in one or several PCs that connects to the server and asks for a *work packet*. This work packet is composed of two elements: a text file containing the model parameter values (problem) and the simulator executable file. Once the work packet is received, the Client executes a realization using the model parameters stored in the text file. When the realization finishes, a solution file is generated, returned to the server and dropped in the *Results files* folder. More details about how *Sisifo* works can be found in [16].

In our case, the *Sisifo* clients are going to be located in each one of the 64 cores of the Sandy Bridge computer. The *Sisifo* server is located in a regular PC running under Windows 7 OS.

4. Asynchronous Random Particle Swarm Optimization (arPSO)

4.1. Introducing arPSO into the distributed computing environment

Using Python3 [17] and *Mathematica* [18], we have implemented an asynchronous version of Random Particle Swarm Optimization (arPSO) algorithm adapted to the *Sisifo* computing environment. First, we need to recall, in the Algorithm 2, the rPSO algorithm appearing in [9]. For our problem, a solution (particle) is a vector, $\overrightarrow{p_i}$ composed of a set of values for the 11 parameters of the model. Then a run of the simulator under this configuration of the parameters is made and the state of the network is obtained and compared with the data obtained from the surveys.

The Algorithm 2 can be adapted to *Sisifo* computing environment if, using the *Sisifo Server*, the run of the model for each particle is distributed among the *Sisifo Clients*.

However, in a typical PSO procedure, including the rPSO, the set of particles is updated once the fitnesses of all the particles (a whole generation) have been calculated. This means that, while all the fitnesses have not been

input : N: number of particles
input : ITMAX maximum number of iterations
output: F_i: Best fitness
output: $\overrightarrow{p}_{global}^{best}$: Best Solution

// Initialization
for $i \leftarrow 1$ **to** N **do**
    $\overrightarrow{p_i} \leftarrow$ GenerateRandomVector11 $(k, P_g^1, P_g^2, P_g^3, P_g^4, T_{HR}^{cls}, T_{LR}^{cls}, P_{HR}^w, P_{LR}^w, P_{HR}^m, P_{LR}^m)$;
    $\overrightarrow{v_i} \leftarrow$ GenerateRandomVector11 (0,0.1);
    $F_i \leftarrow$ Evaluate $(\overrightarrow{p_i})$;
    $\overrightarrow{p_i}^{best} \leftarrow \overrightarrow{p_i}$;
    if $i = 1$ **then** $\overrightarrow{p}_{global}^{best} \leftarrow \overrightarrow{p_1}$;
    else if $\overrightarrow{p_i}^{best}$ *has better fitness than* $\overrightarrow{p}_{global}^{best}$ **then**
        $\overrightarrow{p}_{global}^{best} \leftarrow \overrightarrow{p_i}^{best}$;
end
// Main loop
for $t \leftarrow 1$ **to** $ITMAX$ **do**
    for $i \leftarrow 1$ **to** N **do**
        // Velocity update
        $\omega \leftarrow$ GenerateRandomNumber $[\frac{1}{4}, \frac{3}{4}]$;
        $\psi_i^1 \leftarrow$ GenerateRandomNumber $(0, 1.5)$;
        $\psi_i^2 \leftarrow$ GenerateRandomNumber $(0, 1.5)$;
        $\overrightarrow{v_i} \leftarrow \omega \cdot \overrightarrow{v_i} + \psi_i^1 \cdot (\overrightarrow{p_i}^{best} - \overrightarrow{p_i}) + \psi_i^2 \cdot (\overrightarrow{p}_{global}^{best} - \overrightarrow{p_i})$;
        // Particle update
        $\overrightarrow{p_i} = \overrightarrow{p_i} + \overrightarrow{v_i}$;
        $F_i \leftarrow$ Evaluate $(\overrightarrow{p_i})$;
        // Local best update
        if $\overrightarrow{p_i}$ *has better fitness than* $\overrightarrow{p_i}^{best}$ **then**
            $\overrightarrow{p_i}^{best} \leftarrow \overrightarrow{p_i}$
        end
    end
    // Update the global best as the particle with the best fitness so far
    $\overrightarrow{p}_{global}^{best} \leftarrow$ best fitness of $\{\overrightarrow{p}_{global}^{best}, \overrightarrow{p_1}^{best}, \ldots, \overrightarrow{p_N}^{best}\}$;
end

Algorithm 2: rPSO pseudocode [9]. GenerateRandomVector11 generates a random vector with 11 elements and GenerateRandomNumber generates a random number. Note that the global best is updated once a whole generation has been evaluated and this global best is used in the update of the next generation.

evaluated and an iteration (loop $12-25$) of Algorithm 2 is not completely finished, the global best is not updated, the particles cannot be updated and new evaluations cannot be performed. Then, in every iteration of rPSO, scenarios where some *Sisifo* clients have finished their evaluations and are idle while other *Sisifo* clients are still performing their evaluations are usual. In these scenarios, we have an under-use of the *Sisifo* system.

In order to avoid the under-use system drawback, we propose the implementation of an asynchronous version of rPSO in such a way that, when the fitness of a particle has been evaluated, this particle is updated (lines $14-19$ in Algorithm 2) without waiting for the evaluation of the remainder particles, considering the current existing global best and its individual best particles. To do so, we modify rPSO algorithm parallelizing the loop $12-25$ and sharing the updates of the global best particle.

The asynchronous rPSO with the inclusion of the above features and others is described in Algorithm 3. In the Figure 1 we show how and where we set the arPSO in the *Sisifo* environment.

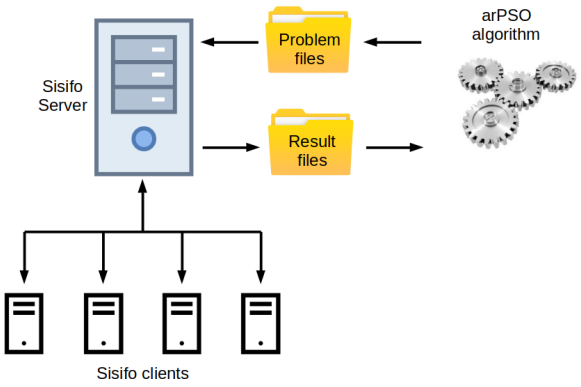

Figure 1: Introduction of the asynchronous rPSO (arPSO) algorithm in the *Sisifo* environment. arPSO manages: (1) the generation of new problems and put them into the *Problem files* folder and (2) the reading and processing of the solution files located in the *Result files* folder.

When the procedure of Algorithm 3 starts running, with the initialization of the particles (loop $1-6$), we create their corresponding problem files in the *Problem files* folder. The *Sisifo Server* detects new problem files and distribute them among free *Sisifo Clients*. These clients carry out the realizations. When a *Sisifo* client ends its task, a file with results is generated and sent to the *Sisifo Server* that drops it on the *Result files* folder. Every time a new file with results appears (line 9) in the *Results files* folder, the arPSO reads the data from the file with results and calculates the fitness.

Then, updates the local best, the global best and the velocity taking into account one of the existing global best, selected randomly, and its individual best (line 161), updates the particle (lines $23-29$) and with the new particle creates a new problem file in the *Problem files* folder. And so on.

4.2. Fitness function and some features included in the arPSO

The fitness evaluation $\mathcal{F}$ performs as follows:

- we start the model and, after a warm-up time of 400 simulated months, we assume a stabilization of the model output;
- we take the model output from month 401 to 500 for the subpopulations in the data of Table 2, i.e., percentage of women HR- and LR-infected per group ages $18-29$ and $18-64$;
- then, for each subpopulation, we calculate the maximum and the minimum of these portions of the model output and we see if these maximum and minimum are inside the corresponding data 95% confidence interval;
- if this happens, the fitness is zero;
- otherwise, the fitness is the sum of the distances from these maxima and minima to the corresponding 95% confidence interval of the data.

In Figure 2, we can see an example of how the fitness function works. We run the computational model. In the month 400 (green vertical line) the model (red line) is stable and from month 401 until month 500 we can compare with the data 95% confidence interval given by the horizontal blue lines. In this case, the error is close to zero, because the only model output outside the confidence interval, but very near, is the corresponding to the lower right graph.

Remark 4.1. *At this point, we want to remark that to evaluate the fitness function, we do not need the model output for all the subpopulations. Only is necessary the output corresponding to HR- and LR-infected women in the group ages* $18-29$ *and* $18-64$*. Then, when a realization is carried out, we retrieve and analyze from the results file, the data corresponding to the above subpopulations from the month* 401 *to* 500*, write them in a row and this row is stored as the result of the realization to be used later. Also, in order to compare properly the model output row with the data, we build the vector whose components are average data, percentile* 2.5 *and percentile* 97.5*,*

repeating 100 *times the* 4 *values of each we have in the Table 2 and write them in a row.*

Next we describe some new features included in our version of the arPSO algorithm. These changes have been devised to explore the parameter space deeper:

1. We include the possibility to discard the new particle and generate a new one randomly with 10% probability, lines $31 - 32$ of Algorithm 3. If it is not discarded, with 10% probability we apply a mutation to the new particle a mutation (lines $34 - 35$).
2. The definition of the fitness function may provide the same fitness values for different particles. Therefore, we are going to store all the particles with the same best fitness and in line 20 we chose p_{global}^{best} randomly among the stored particles.
3. As we mentioned before, physicians around the world have published estimations of most of the model parameters [12, 13, 14, 15]. It is clear that we have to respect their estimation in our fitting procedure avoiding that some model parameters overpass the estimation intervals. Furthermore, finding model parameters in the range of the estimations gives credibility to our model. Therefore, when a model parameter is less than 1% closer to an extreme of the interval provided by the physicians' estimations, we discard this value and it is substituted by a random value inside the interval of bonds for the parameter (line 28).

It is worth noting that the above points allow us to explore extensively the whole space of parameters avoiding the accumulation of particles close to the borders, as we see later in Figure 5.3.

5. Experiments, calibration and results

5.1. Selecting an optimal number of particles to calibrate the HPV network computational model with rPSO

In this section we are going to determine the optimal number of particles for our calibration problem. We can not affirm that *the higher the number of particles, the better the quality of the solution.* Moreover, we also detected that, due to our asynchronous parallel implementation, sometimes the use of more processors does not mean lower execution times. We would need more experiments to detect the correct cause of the delays. However, in

this case is more practical to investigate wich is the optimal combination of particles to achieve the best quality with the lowest number of processors.

To perform this test, we are going to build HPV network models with $50,000$ nodes. We run 5 repetitions of the same problem with different number of particles. Times are shown in seconds. Thus, the question is, what is the quality of the solutions for different number of particles? Table 3 shows relevant information regarding the quality of the solutions for different number of particles and 5 runs of each configuration. Each run carried out 1200 particle evaluations. We measure the quality of the solutions on Table 3 by counting the number of fitness values equal to zero in the 5 runs (*Total # 0s*) and on average (*Avg #0s*). In addition, we also recorded the average iteration that the first zero appears (*Avg First*), from the last population of particles. Fitness of the best, worst and average, averaged over five runs (*Best at end*, *Worst at end* and *Avg at end*). And, regarding executions time, we show on Table 3 information about the average total execution time for 1200 evaluations (*Total Time*), the worst execution time for one particle (*Worst t*) and the best execution time for one particle (*Best t*). Red color indicates the worst configuration, bold letters indicate the best and blue color the second best configuration.

Table 3: Analysis of the quality and execution times of different rPSO configurations. Results of 1200 evaluations for different number of particles on the rPSO process (# Part).

# Part.	Total #0s	Avg #0s	Avg First	Best at end	Worst at end	Avg at end	Total Time	Worst t	Best t
25	18	3.60	**531.00**	**0.015486**	**0.429090**	**0.122573**	8422.00	**148.60**	**107.00**
30	**28**	**5.60**	594.00	0.003785	0.445587	0.131193	**6908.00**	186.60	112.00
35	8	1.60	834.80	0.010211	0.576688	0.149215	7105.00	250.60	119.60
40	16	3.20	692.80	0.003332	0.524417	0.132628	8808.60	389.40	144.60
45	14	2.80	790.40	0.006458	0.639775	0.149670	10004.00	502.00	177.20
50	11	2.20	796.00	0.005168	0.658020	0.139089	11124.40	648.40	220.00
55	2	0.40	1171.20	0.004330	0.518894	0.137088	11905.80	775.00	207.40
60	12	2.40	765.00	0.002153	0.546259	0.132464	13787.80	857.00	245.40
64	5	1.00	985.40	0.002763	0.551228	0.140977	18770.00	1041.20	379.20

As we can see, 25 and 30 particles are the best configurations, since we obtained the higher number of zeros in total and on average and with the lower executions times. Although the results of 25 particles show a very good run with 18 zeros and also a very good execution time, the configuration with 30 particles should be selected bearing in mind both quality and exe-

cution time. Results on total number of 0s and total time were statistically significant with p value of 0.1 after an ANOVA analysis.

5.2. Calibration procedure

Now, we are going to perform the model calibration using the *Sisifo* environment with the arPSO algorithm and $N = 30$ particles. The HPV network models have 500 000 nodes. In order to guarantee the reproducibility of the realizations, we are going to include in the problem files seeds for the generation of the random numbers during the calibration process. Thus, with the same seed we guarantee the building of the same network and the same model output.

We assume that, initially, data of prevalence from Table 2 are not only for women but also for men. Then, we label women and men as infected randomly following these prevalence data. Also, we start running the realization and the first 400 months are used as a warm-up period to stabilize the distribution of infected men and women. Thus, the goal is to calibrate the model parameters in such a way that the model output related to women HR and LR prevalence minimize the fitting function defined in Section 4.2.

We have performed 20 different calibrations using arPSO, each one with around 500 realizations and 30 particles. A total of $10,487$ realizations of the model were performed with an equivalent sequential total computation time of 690 days. More information about the computation time of the realizations can be seen in Figure 3.

5.3. Improving the exploration of the arPSO

As we explained in Section 4.2, we established a procedure respect to the intervals for the parameter values proposed by physicians. Some algorithms such as differential evolution [19] allow to overpass these limits in the search of a good combination of parameters. However, this is not a good idea when implementing our parallel arPSO. First, remember that our parallel version is asynchronous and the updating of the particles is made once every particle is evaluated because parameters out of the defined bounds have not a medical meaning. Therefore, when a model parameter is less than 1% closer to an extreme of the interval provided by the estimations, we discard this value and it is substituted by a random value inside its interval. This also allows to make a deeper and more efficient exploration of the search space.

Figures in Table 5.3 show the exploration performed in some of the parameters of the model (time of clearance and contagion parameters). Figures represent the values of the parameters in the $10,487$ realizations performed

during the 20 executions of arPSO algorithm. X axis represents the realization number and Y axis the value of each parameter. We can see that we find points on most of the search space.

5.4. Selecting the 30 realizations that best capture the data uncertainty

Our goal, now, is to find a procedure to select 30 among the $10,487$ realizations of the model in such a way that the means and the 95% confidence intervals of these 30 realizations are as close as possible of the corresponding means and the 95% confidence intervals of the data in Table 2. We decided to select 30 because, as we can see in Table 3, the total computation time is the best for 30 particles running in parallel in the Sandy Bridge computers.

input : N: number of particles
input : ITMAX maximum number of iterations
input : $[b_1, B_1], [b_2, B_2], \ldots [b_{11}, B_{11}]$: lower and upper bounds of parameters
output: F^{best}: Best fitness
output: $\overrightarrow{p}^{best}_{global}$: Best Solution

```
// Initialization of the particles and velocities.
   Then, they are sent to free Sisifo clients
```
for $i \leftarrow 1$ **to** N **do**
  $\overrightarrow{p_i} \leftarrow$ `GenerateRandomVector11` $(k, P_g^1, P_g^2, P_g^3, P_g^4, T_{HR}^{cls}, T_{LR}^{cls}, P_{HR}^w, P_{LR}^w, P_{HR}^m, P_{LR}^m)$;
  $\overrightarrow{v_i} \leftarrow$ `GenerateRandomVector11` (0,0.1);
  if $i = 1$ **then** $P_{global}^{best} = \{\overrightarrow{p_1}\}$ with fitness $+\infty$;
  Create the Problem File with $\overrightarrow{p_i}$ and send it to a free Sisifo client to be evaluated;
end
```
// Main loop
```
count = 0 ;
while *count* < *ITMAX* **do**
```
  // When the server receives a Result File from a
     client ...
```
  if *a client returns a Result File for particle* $\overrightarrow{p}$ **then**
```
    // Analize the problem file and calculate the
       fitness
```
    $F \leftarrow$ `Evaluate` $(\overrightarrow{p})$;
```
    // Update the local best
```
    if $\overrightarrow{p}$ *has better fitness than* $\overrightarrow{p}^{best}$ **then** $\overrightarrow{p}^{best} \leftarrow \overrightarrow{p}$;
```
    // Update the global best
```
    if $\overrightarrow{p}$ *has the same fitness as* P_{global}^{best} **then**
      add $\overrightarrow{p}$ to P_{global}^{best}
    else if $\overrightarrow{p}$ *has better fitness than* P_{global}^{best} **then**
      $P_{global}^{best} = \{\overrightarrow{p}\}$
    end
```
    // Generate a new particle
```
    $\omega \leftarrow$ `GenerateRandomVector11` $[\frac{1}{4}, \frac{3}{4}]$;
    $\psi^1 \leftarrow$ `GenerateRandomVector11` $(0, 1.5)$;
    $\psi^2 \leftarrow$ `GenerateRandomVector11` $(0, 1.5)$;
    Select $\overrightarrow{p}^{best}_{global}$ randomly among the elements of P_{global}^{best};
    $\overrightarrow{v} \leftarrow \omega \cdot \overrightarrow{v} + \psi^1 \cdot (\overrightarrow{p}^{best} - \overrightarrow{p}) + \psi^2 \cdot (\overrightarrow{p}^{best}_{global} - \overrightarrow{p})$;
    $\overrightarrow{p} = \overrightarrow{p} + \overrightarrow{v}$;
```
    // Generation of a new random particle
```
    if *GenerateRandomNumber(0,1)*< 0.1 **then**
      `GenerateNewParticle` $(\overrightarrow{p})$;
```
    // Mutation
```
    else if *GenerateRandomNumber(0,1)*< 0.1 **then**
      `Mutate` $(\overrightarrow{p})$;
    end
```
    // Preventing overpassing
```
    `PreventOverpassing` $(\overrightarrow{p})$;
```
    // Send the problem to a free client
```
    Create the Problem File with $\overrightarrow{p}$ and send it to a free Sisifo client to be evaluated;
    count ++;
  end
end

Algorithm 3: arPSO, the rPSO pseudocode adapted to the *Sisifo* distributed computing environment with some features included.

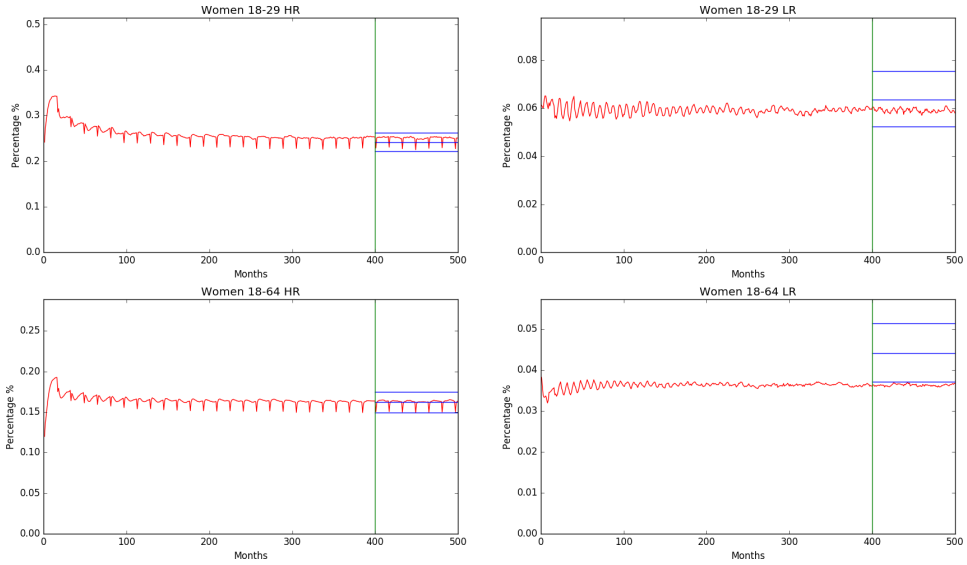

Figure 2: Example of how the fitness function works. In the month 400 (green vertical line) the model (red line) is stable and from month 401 until month 500 we can see if the model output lies inside the data 95% confidence interval given by the horizontal blue lines.

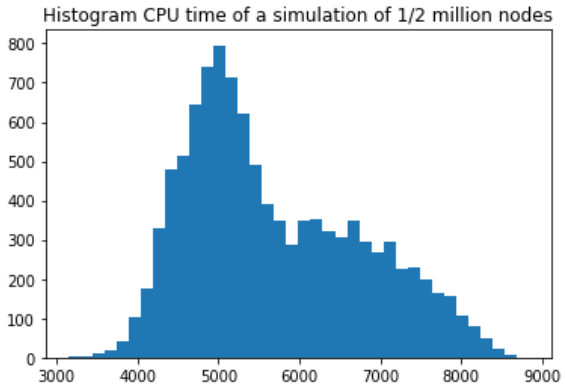

Figure 3: Histogram of the computation time of each one of the $10,487$ realizations of the model using networks of $500,000$ nodes. The average computation time is 5687.9 seconds, around one hour and a half.

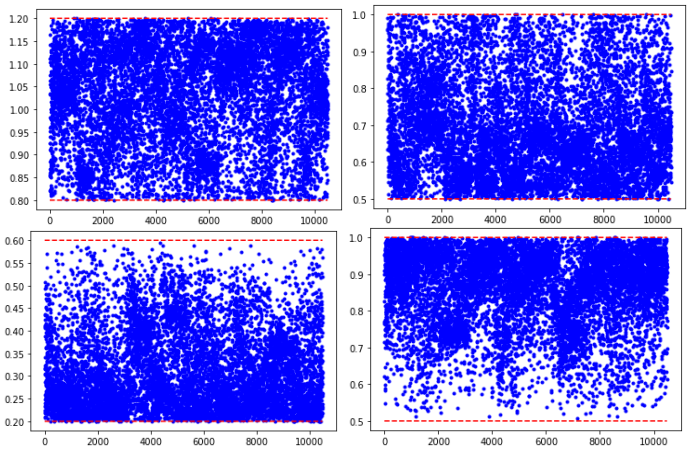

Figure 4: Upper figures, from left to right: Time of clearance of HPV high risk and low risk, both for men and women. Lower figures: on the left, probability to transmit low risk HPV if the infected is a woman; on the right, probability to transmit high risk HPV if the infected is a man. Red dashed lines correspond to the bounds of each parameter. We can see that most of the search space is explored in all the cases.

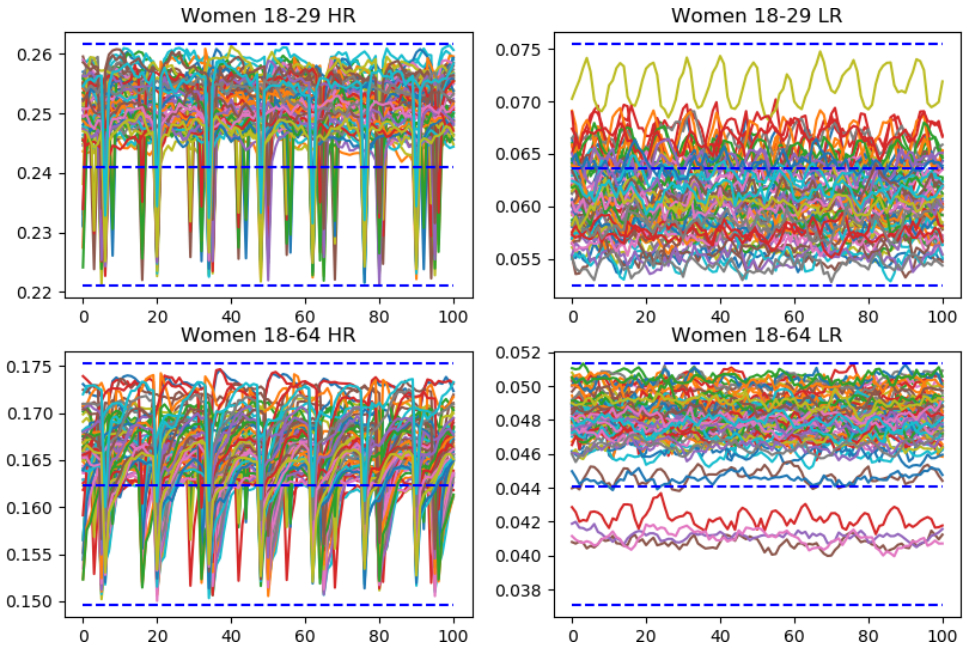

Figure 5: Plot of the 70 model realizations with error 0 from month 401 to 500. Note that all of them lie inside the 95% confidence intervals of the data represented by the blue horizontal dashed lines.

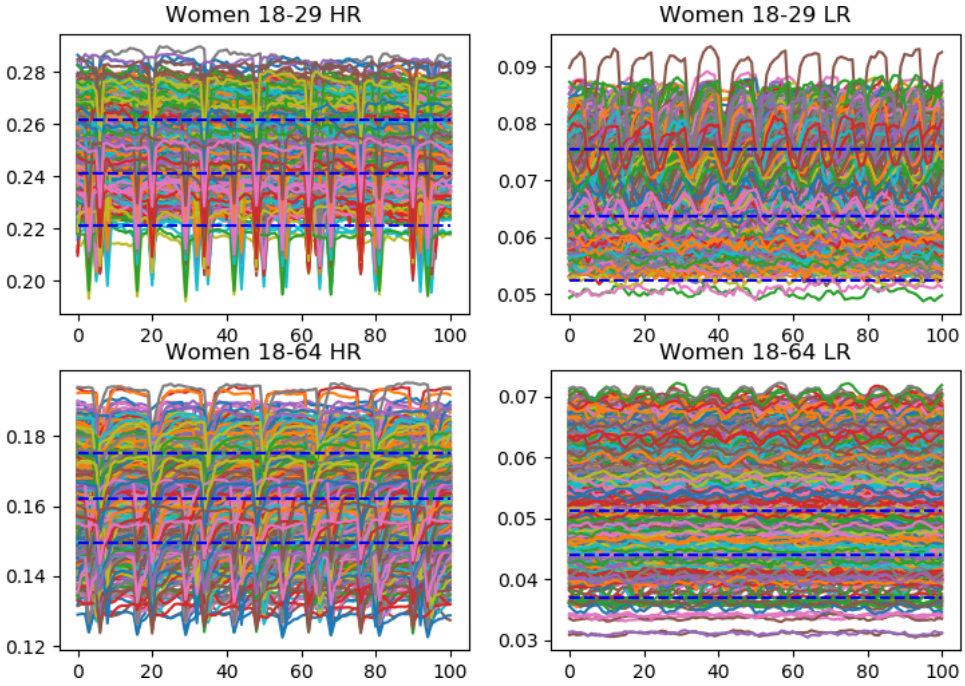

Figure 6: Plot of the 1217 model realizations with error less than 0.03 from month 401 to 500. These realizations cover the 95% confidence intervals of the data, represented by the blue horizontal dashed lines.

Thus, we check the number of possible realizations depending on their error. Then, there are 360 realizations with error less than 0.01, 740 with error less than 0.02, 1217 with error less than 0.03, 1716 with error less than 0.04 and 2294 with error less than 0.05. In Figure 6, we draw the 1217 model realizations of with error less than 0.03. Note that the realizations cover completely the 95% confidence intervals of the data.

Nevertheless, it would be interesting to reduce the number of eligible realizations to much less than $10,487$. In Figure 5 we can see the 70 realizations with error 0, that is, the realizations that lie inside the 95% confidence intervals of the data, represented by the blue horizontal dashed lines. If we select 30 among these 70, it is clear that some percentiles of the model are far from the corresponding percentiles of the data, for instance, the lower parts of the left figures. Therefore, we need to consider realizations with errors greater than zero without forgetting the objective to reduce the number of eligible realizations.

In order to determine the 30 realizations that capture in the best way the

mean and the 95% confidence intervals of the data in Table 2, we propose the PSO-inspired Algorithm 4. Let E be a subset of the $10,487$ realizations (model outputs) to be obtained by the 20 executions of arPSO algorithm previously performed and $E(i)$ the vector corresponding to realization i in the set E. Let $card(E) = M$ be the number of elements of E. In our experiments, if we call E to the set of realizations with error equal to zero, then $card(E) = 70$. For E being the set of realizations with error less that 0.03, $card(E) = 1217$. Thus, the problem now consists of selecting the set of 30 solutions from some specific set E, which best fit the mean, percentile 2.5 and percentile 97.5 of the data:

$$\overrightarrow{S_i} = (E(i_1), E(i_2) \ldots, E(i_{30})). \tag{2}$$

The search process is inpired in the PSO algorithm, but instead of updating $\overrightarrow{S_i}$ with the velocity of the particle as in Algorithm 2 and Algorithm 3, we use a super-set of components (realizations) P_i. This super-set includes the solutions of $\overrightarrow{S_i}$, the best solution found for this particle, $\overrightarrow{S_i}^{best}$, and the best global solution, $\overrightarrow{S}^{best}_{global}$ (see lines $16-18$ at Algorithm 4). We also consider 10% of randomness and 10% of mutation when updating new particles. In this case, mutation consists of changing some of the indexes in the current particle by other randomly chosen, avoiding repetitions. And now the evaluation process is:

INPUT: Set of 30 indexes $I = \{i_1, \ldots, i_{30}\}$, $1 \leq i_j \leq M$, $j = 1, \ldots, 30$.

Step 1. Select the realizations $E(i_1), \ldots, E(i_{30})$ and calculate the mean, percentile 2.5 and percentile 97.5 of them.

Step 2. Calculate the root mean squate of the difference between the mean, percentile 2.5 and percentile 97.5 of the 30 realizations and the data (see Remark 4.1), and sum them up.

Algorithm 4, in the tests we have run, lasts around 20 minutes for 1 million of evaluations of the fitness function in the Sandy Bridge computer, returning accurate solutions. We have performed runs for 30, 40, 50 and 60 particles, being E the realization sets with errors less than 0.01, 0.02, 0.03, 0.04 and 0.05. The lowest error has been 0.1107 for the following realizations

$$\begin{array}{c} 14, 18, 19, 40, 59, 76, 111, 116, 121, 127, 166, 181, 182, 184, 197, 234, \\ 309, 345, 380, 404, 407, 704, 705, 742, 776, 864, 887, 1069, 1082, 1092, \end{array} \tag{3}$$

Table 4: Mean and the 95% confidence interval of the model parameters corresponding to the 30 selected realizations from (3).

Model parameter	Mean	95% confidence interval
Average LSP men	7.89	$[7.05, 9.17]$
Average time for clearing HR HPV	1.08	$[0.98, 1.17]$
Average time for clearing LR HPV	0.60	$[0.52, 0.71]$
Probability a woman transmits LR	0.24	$[0.21, 0.29]$
Probability a man transmits LR	0.30	$[0.23, 0.38]$
Probability a woman transmits HR	0.75	$[0.59, 0.89]$
Probability a man transmits HR	0.90	$[0.77, 0.98]$

among the 1217 of the set of realizations with error less than 0.03. In Figure 7 we draw the 30 selected realizations and in Figure 8 we can see the graphical result of the calibration and how resemble the means and 95% confidence intervals.

The mean and the 95% confidence interval of the model parameters corresponding to the 30 selected realizations from (3) are given in the Table 4. Note that the obtained model parameters are in accordance to the medical parameters appearing in the literature and detailed in Section 2.

input : E with M realizations; N number of particles
input : ITMAX maximum number of iterations
input : CI = 95% Confidence Interval
input : *DataSet* =Table 2
output: $\overrightarrow{S^{best}_{global}}$: 30 realizations from E that best fit *DataSet* with CI

```
// Initialization of the particles and the local and
   global best
```
for $i \leftarrow 1$ **to** N **do**
  $\overrightarrow{S_i} \leftarrow$ `RandomSelect` (30, E);
  $F_i \leftarrow$ `EvaluateSet` ($\overrightarrow{S_i}$, CI, *DataSet*);
  $\overrightarrow{S_i^{best}} \leftarrow \overrightarrow{S_i}$;
  if $i = 1$ **then**
    $\overrightarrow{S}^{best}_{global} \leftarrow \overrightarrow{S_1}$
  else if $\overrightarrow{S_i}$ *has better fitness than* $\overrightarrow{S}^{best}_{global}$ **then**
    $\overrightarrow{S}^{best}_{global} \leftarrow \overrightarrow{S_i}$
  end
end
`// Main loop`
for $t \leftarrow 1$ **to** $ITMAX$ **do**
  for $i \leftarrow 1$ **to** N **do**
    `// Update new particle`
    $P_i \leftarrow \overrightarrow{S_i} \cup \overrightarrow{S_i^{best}} \cup \overrightarrow{S^{best}_{global}}$;
    $P_i \leftarrow$ `RemoveRepetitions` (P_i);
    $\overrightarrow{S_i} \leftarrow$ `RandomSelect` (30, P_i);
    `// Random generation of a new particle`
    if `GenerateRandom(0,1)` < 0.1 **then**
      $\overrightarrow{S_i} \leftarrow$ `RandomSelect` (30, E)
    `// Mutation`
    else if `GenerateRandom(0,1)` < 0.1 **then**
      $\overrightarrow{S_i} \leftarrow$`Mutate` ($\overrightarrow{S_i}$);
    end
    $F_i \leftarrow$ `EvaluateSet` ($\overrightarrow{S_i}$, CI, *DataSet*);
    `// Update local best`
    if $\overrightarrow{S_i}$ *better than* $\overrightarrow{S_i}^{best}$ **then**
      $\overrightarrow{S_i}^{best} \leftarrow \overrightarrow{S_i}$;
    end
    `// Update global best`
    if $\overrightarrow{S_i}^{best}$ *better than* $\overrightarrow{S}^{best}_{global}$ **then**
      $\overrightarrow{S}^{best}_{global} \leftarrow \overrightarrow{S_i}^{best}$;
    end
  end
end

Algorithm 4: PSO-inspired algorithm for selection.

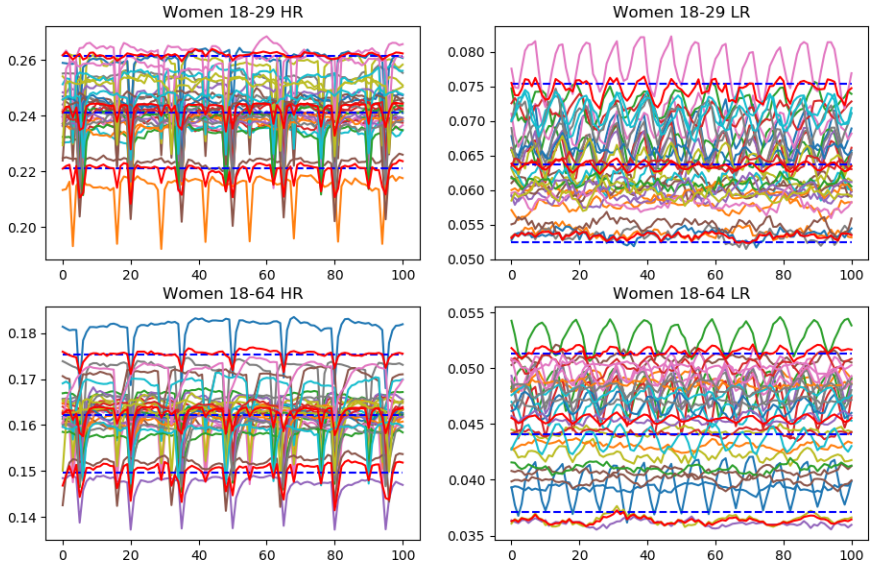

Figure 7: Drawing of the 30 selected model realizations from month 401 to 500. These realizations are the ones whose means and 95% confidence intervals resemble the most the data in Table 2, represented by the blue horizontal dashed lines.

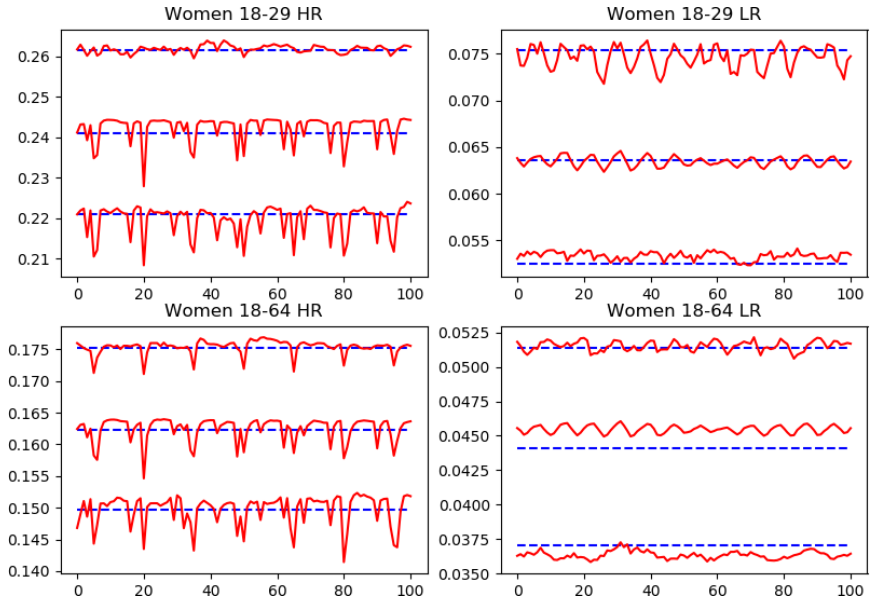

Figure 8: Means and 95% confidence intervals of the 30 selected realizations (in red) compared to the means and 95% confidence intervals of the data (blue).

6. Conclusion

The goal of this paper has been the design of a reproducible and reliable procedure that allows to calibrate a complex network computational model that simulates the transmission dynamics of the Human Papillomavirus (HPV), with the aim to be extensible to other computational models with randomness in the building and uncertainty in the data.

Due to the large amount of computations, we have to use a computed distributed environment and to design an asynchronous PSO algorithm, called arPSO, specifically adapted to this environment. Also, we perform a test to determine the appropriate number of particles to improve the computational efficiency obtaining good quality solutions as well.

Once the calibration has been done, we have to select the model parameters that best capture the data uncertainty. To achieve this goal, we design an *ad-hoc* PSO-inpired algorithm.

In future works, we will use the sets of parameters corresponding to the realizations (3) (and their corresponding seeds) to perform, among others, the following simulations of interest in Public Health,

- to study the decline of genital warts cases in the long-term in Spain,
- to simulate the effect of the tourism on the contagion of HPV in Spain,
- to simulate different strategies of vaccination of girls and boys aimed at erradicating the high risk HPV.

We expect that the above simulations can be performed in a reasonable time because of the selection of only 30 sets of model parameters that fit and explain the data uncertainty, and will help us to predict the evolution of HPV in the above scenarios.

Acknowledgements

This work has been supported by the Spanish Ministerio de Economia y Competitividad grant MTM2017-89664-P.

References

[1] What is HPV? [online, cited Dec 20th, 2016].

[2] X. Castellsagué, T. Iftner, E. Roura, J. A. Vidart, S. K. Kjaer, F. X. Bosch, N. Muñoz, S. Palacios, M. S. M. Rodriguez, L. Serradell, L. Torcel-Pagnon, J. C. and, Prevalence and genotype distribution of human papillomavirus infection of the cervix in Spain: The CLEOPATRE study, Journal of Medical Virology 84 (6) (2012) 947–956. doi:10.1002/jmv.23282.
URL https://doi.org/10.1002%2Fjmv.23282

[3] K. J. Syrjänen, S. M. Syrjänen, Papillomavirus infections in human pathology, Wiley, 2000.

[4] Official Site for GARDASIL9 [online, cited Feb 7th, 2018].

[5] L. Acedo, C. Burgos, J.-I. Hidalgo, V. Sánchez-Alonso, R.-J. Villanueva, J. Villanueva-Oller, Calibrating a large network model describing the transmission dynamics of the human papillomavirus using a particle swarm optimization algorithm in a distributed computing environment, The International Journal of High Performance Computing Applications (2017) 109434201769786doi:10.1177/1094342017697862.
URL https://doi.org/10.1177/1094342017697862

[6] J. Díez-Domingo, V. Sánchez-Alonso, R.-J. Villanueva, L. Acedo, J.-A. Moraño, J. Villanueva-Oller, Random network models to predict the long-term impact of HPV vaccination on genital warts, Viruses 9 (10) (2017) 300. doi:10.3390/v9100300.
URL https://doi.org/10.3390/v9100300

[7] J. Rada-Vilela, M. Zhang, W. Seah, Random asynchronous pso, in: Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, IEEE, 2011, pp. 220–225.

[8] K. Holladay, K. Pickens, G. Miller, The effect of evaluation time variance on asynchronous particle swarm optimization, in: Evolutionary Computation (CEC), 2017 IEEE Congress on, IEEE, 2017, pp. 161–168.

[9] N. Khemka, C. Jacob, Exploratory toolkit for evolutionary and swarm-based optimization, The Mathematica Journal 11 (3) (2010) 376–391. doi:10.3888/tmj.11.3-5.
URL https://doi.org/10.3888%2Ftmj.11.3-5

[10] Portal estadístico de la Generalitat Valenciana (Statistical portal of the government of the Community of Valencia) [online] (2013) [cited Dec 20th, 2016].

[11] Encuesta de salud y hábitos sexuales 2003 (Health and sexual habits survey). Instituto Nacional de Estadística [online] (2003) [cited Dec 20th, 2016].

[12] Estudio de conducta sexual entre homosexuales (Study of sexual behavior among homosexuals), Tech. rep., Durex (2002).

[13] E. H. Elbasha, E. J. Dasbach, R. P. Insinga, Model for assessing human papillomavirus vaccination strategies, Emerging Infectious Diseases 13 (1) (2007) 28–41. doi:10.3201/eid1301.060438.
URL https://doi.org/10.3201%2Feid1301.060438

[14] A. R. Giuliano, J.-H. Lee, W. Fulp, L. L. Villa, E. Lazcano, M. R. Papenfuss, M. Abrahamsen, J. Salmeron, G. M. Anic, D. E. Rollison, D. Smith, Incidence and clearance of genital human papillomavirus infection in men (HIM): a cohort study, The Lancet 377 (9769) (2011) 932–940. doi:10.1016/s0140-6736(10)62342-2.
URL https://doi.org/10.1016/s0140-6736(10)62342-2

[15] A. G. Nyitray, M. Chang, L. L. Villa, R. J. C. da Silva, M. L. Baggio, M. Abrahamsen, M. Papenfuss, M. Quiterio, J. Salmerón, E. Lazcano-Ponce, A. R. Giuliano, The natural history of genital human papillomavirus among HIV-negative men having sex with men and men having sex with women, Journal of Infectious Diseases 212 (2) (2015) 202–212. doi:10.1093/infdis/jiv061.
URL https://doi.org/10.1093/infdis/jiv061

[16] J. Villanueva-Oller, L. Acedo, J. A. Moraño, A. Sánchez-Sánchez, Epidemic random network simulations in a distributed computing environment, Abstract and Applied Analysis 2013 (2013) 1–10. doi:10.1155/2013/462801.
URL https://doi.org/10.1155%2F2013%2F462801

[17] Pyhton Software Foundation [online, cited Feb 7th, 2018].

[18] Wolfram Mathematica: Modern Technical Computing [online, cited Feb 7th, 2018].

[19] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization 11 (4) (1997) 341–359.