Many human tumors cannot easily be avoided. In most cases a prophylactic vaccination prevents the tumor growth. In particular the Triplex vaccine prevented mammary carcinoma formation using a Chronic schedule, but it is not known if this schedule is minimal. A computational model named SimTriplex was able to reproduce in silico the in vivo experiments. The combination of SimTriplex and Genetic Algorithms (GA), produced a minimal vaccination schedule capable to guarantee in silico survival. This process required the use of a High Performance Computing (HPC) infrastructure for several days. In this paper we show another optimization procedure based on simulated annealing approach used to get a faster algorithm response.

Keywords: Simulated annealing; optimization; cancer; vaccine.

1. Introduction
Human tumors directly caused by exogenous carcinogens, e.g., tobacco or solar exposure, can be prevented by the elimination of the exposure to the carcinogenic agent. Exogenous tumors are only a subset of human tumors. Many human tumors are indeed caused by endogenous factors or situations that cannot be readily circumvented.
Many clinical attempts stimulating the ability of the immune system to recognize and fight cancer (immunotherapy) have been investigated. An evaluation of the preclinical results of vaccines in mouse models showed in particular a progressive loss of vaccine efficacy directly connected to tumor development. In most cases vaccination before the challenge (prophylactic vaccination) prevented tumor growth, whereas vaccination after the challenge (therapeutic vaccination) demonstrated less effective.

The Triplex vaccine\textsuperscript{7,8} represents a clear example of such immunopreventive approaches. It has been designed to improve the efficacy of existing immunopreventive treatments against mammary carcinoma and demonstrated to be effective in preventing the carcinoma in situ (CIS) formation in HER-2/neu mice (transgenic mice genetically engineered to over-express the “Human Epidermal growth factor Receptor 2” which is a protein giving higher aggressiveness in breast cancers) using the appropriate administration schedule (Chronic schedule).\textsuperscript{9}

However it is not known if Chronic schedule is really the minimal set of vaccinations capable of assuring complete, long-term protection against mammary carcinoma. Shorter heuristic protocols failed in fulfilling this difficult job, but between the Chronic and the shorter schedules there is still an enormous number of possibilities which remain yet unexplored.

This question motivated the development of a computational model named SimTriplex, able to reproduce in silico the same behavior of the in vivo experiments.\textsuperscript{10} The model was then used as a virtual laboratory to find an optimal schedule using a genetic algorithms (GA)\textsuperscript{11} as optimal search methodology. The GA, which required the use of an HPC infrastructure for two days, produced a 37-injections schedule (versus 60 of the Chronic schedule) capable to guarantee in silico survival.\textsuperscript{12}

In a near future the possibility for clinicians to have mathematical — computational tools to aid the definition of personalized schedules could become reality. For this purpose clinicians will need models that can be used in their natural environments, i.e., hospitals, where the computer power facilities are usually one order of magnitude less than in research environments. This raises the problem of producing models which could be used in clinical environments.

Models themselves entitle continuous cyclical refinements by means of interdisciplinary efforts (Fig. 1) and additional inclusions of biological knowledge up to natural-scale. This will increase both the complexity and the computational effort required by models which will probably counterbalance the growth of computational power of computers.

From these considerations a novel approach for a faster “optimal protocol finder” is really advisable. In particular, we devised that the new computational approach had to be affordable even on today consumer market computers, since HPC use is not something that a clinician will find user-friendly. A good candidate for this aim is Simulated Annealing (SA), which is a global optimization algorithm, widely tested and known for its computational speed and ability to achieve optimal solutions.
Fig. 1. Cyclical refinement in defining models that can also be used by other disciplines than biology. Data provided from initial wet-biology experiments is used to develop the first prototype of the model. The model is then refined until it is able to reproduce provided data. Model results are then used to answer or predict specific biological questions about the experiments. The predictions therefore need to be checked with new in vivo experiments. Outcomes from last experiments are then adopted to further refine/improve the model.

The paper is organized as follows. In Sec. 2, we give a brief introduction to Simulated Annealing. In Sec. 3, we define our optimization problem. Section 4 shows the implementation of the algorithm and results are shown in Sec. 5. Section 6 is devoted to conclusions and final remarks.

2. The SA Algorithm

The groundbreaking paper by Kirkpatrick\(^2\) opened the path to a deep analogy between Statistical Mechanics (the behavior of systems with many degrees of freedom in thermal equilibrium at a finite temperature) and Combinatorial Optimization (the method of finding the minimum, if any, of a given function with respect to many parameters). There is a close similarity; indeed, between the procedure of annealing in solids and the required framework for optimization of complex systems, such as the Immune System we are dealing with in this work.

2.1. Annealing in solids

In condensed matter physics a solid can be heated up by increasing the temperature of the heat bath where it is placed to a maximum value such that all particles of the solid arrange themselves in the liquid phase. This first procedure is followed by a second one where the temperature is very slowly cooled down in such a way that all particles are allowed to arrange themselves in the lowest energy ground state.
of the corresponding lattice, provided the starting temperature is sufficiently high and the cooling is carried out slowly enough.

The process is summarized as follows: at each temperature \( T \), the solid is allowed to reach a so-called state of thermal equilibrium, which by definition is achieved when the probability for the solid of being in a state with energy \( E \) is given by the Boltzmann Distribution (a function or probability measure which describes the distribution of the states of a complex system, i.e., the velocities of particles of a gas:\(^\text{13}\)):

\[
P\{E = E\} = \frac{1}{Z(T)} \cdot e^{-E/k_B T},
\]

where \( Z(T) \) is a normalization factor, the partition function, depending on the temperature \( T \) and \( k_B \) is the Boltzmann constant. The factor \( e^{-E/k_B T} \) is the Boltzmann factor. As the temperature decreases, the Boltzmann distribution concentrates in the states with lowest energy and finally, when the temperature approaches zero, only the minimum energy states have a non-zero probability of being occupied.

### 2.2. Simulated annealing

In order to mimic the evolution of the complex systems towards equilibrium once the temperature is fixed, Metropolis\(^\text{1} \) designed a Monte Carlo method commonly referred to as the Metropolis criterion. The method applies small random perturbations to the position of the particles of the solid modifying its configuration. If the difference in energy, \( \Delta E \), between the current and the perturbed configuration is negative, the new configuration indicates a state with lower energy and it is assumed as the new one. Otherwise the probability of acceptance of the new configuration is given by the Boltzmann factor. After a large number of perturbations the probability distribution of the states approaches the Boltzmann distribution.

The Metropolis algorithm can also be used to generate sequences of configurations of a system in a combinatorial optimization problem assuming that the cost function \( C \) and the control parameter \( c \) take the roles of energy and temperature in the physical annealing, respectively.

If we now regard the SA algorithm as a sequence of Metropolis algorithms evaluated at a sequence of decreasing values of the control parameter, i.e., the temperature, we can give a generalization of the method as follows: a generation mechanism is defined so that, given a configuration \( i \), another configuration \( j \) can be obtained by choosing at random an element in the neighborhood of \( i \).

If \( \Delta C_{ij} = C(j) - C(i) \leq 0 \), then the probability for configuration \( j \) to be the next configuration in the sequence is given by 1, and if \( \Delta C_{ij} > 0 \) by \( e^{-\Delta C_{ij}/c} \) (that is nothing else than the Metropolis criterion).

Thus, there is a nonzero probability of choosing a configuration with higher energy than the current one. This process is continued until equilibrium is reached,
i.e., until the probability distribution, $P$, of the configurations approaches the Boltzmann distribution, which translates now as:

$$P\{\text{configuration} = i\} = \frac{1}{Q(c)} \cdot e^{-C(i)/c},$$

(2)

where $Q(c)$ is a normalization constant depending on the control parameter $c$, being the equivalent of the aforementioned partition function.

The control parameter is then lowered in steps, with the system being allowed to approach equilibrium for each step by generating a sequence of configurations in the way we already illustrated. The algorithm terminates for some small value of $c$ where virtually no deteriorations are accepted anymore.

The final frozen configuration is then assumed as solution of the problem at hand.

The global view of the algorithm is shown in the following pseudocode:

**PROCEDURE SIMULATED ANNEALING**

```pseudocode
s = s0;
T = T0;
while (not yet frozen) do
    while (not yet in equilibrium) do
        s' = random state in $N(s)$  (step 1)
        $\Delta c = c(s') - c(s);
        if($\Delta c < 0$)
            then s = s';
        if($\Delta c > 0$)
            then s = s' with probability $e^{-\Delta c/T}$;  (step 2)
        T = reduce(T);
    return(s);
```

where $T$ is the temperature in the algorithm, $s$ and $s'$ are generic configurations of the system, $N(s)$ and $c(s)$ are the neighborhood and the energy of a configuration $s$, respectively.

The simulated annealing algorithm obtains a global minimum if, after a (possibly large) number of transitions $K$ the following relation holds:

$$P\{X(K) \in S_{opt}\} = 1,$$

(3)

where $S_{opt}$ is the set of globally minimal configurations. It can be shown that the previous relation holds asymptotically under certain conditions.$^3$

3. The Optimal Vaccination Schedule Search Problem

The SimTriplex model$^{10}$ has been developed in our previous research to simulate the behavior of the Triplex vaccine. It mimics interactions of immune cells of vaccinated
as well as naive mice showing \textit{in silico} excellent agreements with \textit{in vivo} experiments on HER-2/neu mice.

As previously said, an optimal schedule maintains its efficacy with a minimum number of vaccine administrations. As in standard drug administration, the vaccine has to be effective for a high percentage of patients. In lack of quantitative methods, this is usually achieved using medical consensus, i.e., a public statement on a particular aspect of medical knowledge available at the time it was written, and that is generally agreed upon as the evidence-based, state-of-the-art (or state-of-science) knowledge by a representative group of experts in that area. Our goal is therefore to have a quantitative approach, using simulators and optimization techniques,\textsuperscript{14–16} that can help biologists in designing vaccine protocols.

It is worth to mention here the fundamental definitions of the optimization problem we will deal with. Further mathematical specifications will appear elsewhere.\textsuperscript{12}

Let us consider a time interval $[0, T]$, in which we study the action of the vaccine on a set of virtual mice $S$. This can be, for example, the time-length of the \textit{in vivo} experiment. We then discretize the given time interval in $N - 1$ equally spaced subintervals of width $\Delta t$, i.e., $\{t_1 = 0, t_2, \ldots, t_i, \ldots, t_N = T\}$. The time interval $\Delta t$ corresponds to the time of possible vaccine administrations, e.g., every 8 hours.

Let $x = \{x_1, x_2, \ldots, x_i, \ldots, x_N\}$ be a binary vector representing the sequence of vaccine schedule where $x_i = 0/1$ means respectively administration/no administration of the same quantity of vaccine at time $t_i$. The number of vaccine administrations is given by $n = \sum_{i=1}^N x_i$. The search space $D$ for this problem has therefore cardinality $2^N$. For $T = 400$ days, and $\Delta t = 24$ hours the cardinality is $2^{400}$ which prevents any chance of an exhaustive search.

Anyway, one wet biologist’s requirement is that vaccine administrations can be performed only twice a week (Monday and Thursday) and this is already considered a very intensive vaccination schedule from an immunological point of view.

Luckily, this greatly reduces the cardinality of the search space $D$, from $2^{400}$ ($\sim 10^{120}$) to $2^{114}$ ($\sim 10^{34}$).

The time of the carcinoma in situ (CIS) formation is defined by $\tau(x, \lambda_j)$, which is a function of the vaccination schedule $x$ administered to the mouse $j \in S$ and a parameter $\lambda_j$ which represents the biological diversity. The vaccine will be obviously effective if $\tau \geq T$. The function $\tau$ depends on the action of the stimulated immune system. It cannot be expressed in an analytical form but can be computed by other means, e.g., through SimTriplex simulator.

Thus the optimization problem is to find $\bar{x}$ such that the two following conditions are satisfied:

$$\tau(\bar{x}, \lambda_j) = \max_{x \in D}(\tau(x, \lambda_j)) \quad (4)$$

and

$$n(\bar{x}) = \min_{x \in D}(n(x)) \quad (5)$$

that means we want to maximize the survival time, $\tau$, of each mouse of the sample, using the minimum number of injections, $n$, as well.
At this point, we deal with a multi-objective discrete and unconstrained optimization problem.

Another observation is now in order. As briefly pointed out in Sec. 1, the combination of SimTriplex simulator with a GA algorithm approach lead to an optimal vaccination schedule, capable to guarantee \textit{in silico} survival.\textsuperscript{11} We further specify here that the optimal search strategy was carefully biologically driven: considering that the Chronic schedule proved to be effective in tumor control, the optimal search tried to find a protocol with a minimum number of vaccine administrations able to reproduce, \textit{in silico}, the chronic time evolution of cancer cells. This requirement leads us now to place a threshold on the maximum number of cancer cells allowed, that we express as follows:

\begin{equation}
\begin{aligned}
M_1(x) &\leq \gamma_1, \quad t \in [0,T_{in}], \\
M_2(x) &\leq \gamma_2, \quad t \in [T_{in},T],
\end{aligned}
\end{equation}

where $M_1(x)$ and $M_2(x)$ are the maximum number of cancer cells in $[0,T_{in}]$ (cellular-mediated controlled phase) and in $[T_{in},T]$ (humoral-mediated controlled phase) respectively, and $T_{in} \sim T/3$, while $\gamma_1$ and $\gamma_2$ represent cancer cells threshold in $[0,T_{in}]$ and in $[T_{in},T]$, respectively.

This final requirement turns our optimization problem into a constrained one.

Due to biological variability the optimal schedule found for mouse $j$ is able to prevent CIS formation only in a small fraction ($\sim 20\%$) of mice of the set $S$. So the schedule found for a single mouse is not an immunological effective one as it does not protect a high percentage of the treated mice. To solve this problem, we reformulate the optimization problem in this way: let $\{j_1, j_2, \ldots, j_m\} \subset S$, with $m = 8$, a random chosen subset of \textit{in silico} mice.

\begin{equation}
\begin{cases}
\tau(x, \lambda_{j_1}) = \max(\tau(x, \lambda_{j_1})) \\
\tau(x, \lambda_{j_2}) = \max(\tau(x, \lambda_{j_2})) \\
\quad \vdots \\
\tau(x, \lambda_{j_m}) = \max(\tau(x, \lambda_{j_m})) \\
n(x) = \min(n(x))
\end{cases}
\end{equation}

subject to:

\begin{align*}
M_1(x) &\leq \gamma_1, \quad t \in [0,T_{in}] \\
M_2(x) &\leq \gamma_2, \quad t \in [T_{in},T].
\end{align*}

We modified this last formulation of the problem grouping all the $\tau(x, \lambda_{j_h})$ \quad ($h = 1, \ldots, m$) by a proper statistical indicator. We chose the harmonic mean $H$ of the survivals:

\[ H(x, \lambda_{j_1}, \ldots, \lambda_{j_m}) = \frac{m}{\sum_{i=1}^{m} \frac{1}{\tau(x, \lambda_{j_i})}} \]

since it is very frequently used when statistic measurements of time are involved.
Therefore, the system (7) translates as:

\[
\begin{align*}
H(\bar{x}, \lambda_{j1}, \ldots, \lambda_{jm}) &= \max \{ H(x, \lambda_{j1}, \ldots, \lambda_{jm}) \} \\
n(\bar{x}) &= \min (n(x))
\end{align*}
\]

subject to:

\[
\begin{align*}
M_1(x) &\leq \gamma_1, t \in [0, T_{in}] \\
M_2(x) &\leq \gamma_2, t \in [T_{in}, T].
\end{align*}
\]

The next section will show in detail how to fit all these conditions into the SA algorithm framework.

**Alternative approaches**

Protocol optimization is a hot topic in any drug therapy. While the general problem is the same, i.e., find a protocol which is effective on the maximum number of patients with the minimum number of administrations, the approaches used to solve this problem depend on a wide variety of aspects. The first problem is linked to the allowed time for drug administration. From the mathematical point of view, time is a continuous variable. Classical methods referring to the control theory and to the optimal control will find the optimal protocol as a sequence of points in the continuous axis of time.

The theory of optimal control deals with the problem of finding a control law for a given system in such a way that a certain optimality criterion is reached. We must point out here that these methods require the formulation of a cost functional, i.e., a mathematical model which includes a function of state and control variables usually defined as a set of Ordinary Differential Equations (ODEs). The optimal control is defined by some other differential equations describing the paths of the control variables that minimize the cost functional. Control problems also include extra boundary conditions.

Optimal control problems typically are nonlinear problems and it is necessary to employ numerical methods to solve them.

In the early years of optimal control, the use of indirect methods was the mostly used approach for solving optimal control problems. In an indirect method, the calculus of variations is employed to obtain the first-order optimality conditions. The advantage of the indirect method is that the resulting solution is readily verified to be an optimal trajectory. The disadvantage is that the boundary-value problem is usually extremely difficult to solve.

Presently the direct method is preferred. In this case the state and/or control are approximated using an appropriate function approximation (e.g., polynomial approximation). Simultaneously, the cost functional is approximated as a cost function.

However, as previously stated, optimal control methods (either using indirect or direct approach) need a mathematical formulation of the cost function and cannot be applied to discrete models such as agent based models.
Moreover in many cases (like the one we treat in this paper) vaccine administrations cannot be performed at any time but there exists a finite set of allowed administration times. In this case the problem is a combinatorial one and the appropriate tools come from combinatorial optimization techniques.\footnote{1}

If the model is developed by discrete techniques such as agent based models or cellular automata, the best methods one can apply are represented by stochastic optimization techniques, such as genetic algorithms, simulated annealing and so on and so forth. The main advantage of these methods is represented by their flexibility, which allows their application on a wide range of problems. The main disadvantage probably is a consequence of their biggest advantage. Since they are not designed for specific problems, they are usually slow converging algorithms. If specific knowledge about the problem is available, it can be put inside the optimization algorithm gaining faster convergence speed.

4. Implementation

Let us fit now the general framework exposed in Sec. 3 into our actual problem of optimal vaccine schedule searching via the SA algorithm.

First of all, we match important entities in SA with their counterparts in vaccine protocols: the first have already been described in Sec. 2. Here, we add the relevant concepts of a vaccine administration protocol: the number of injections, the mean survival age of the selected population of mice and the time distribution of injections of a given protocol.

As outlined in Sec. 3, any candidate protocol can be described by a bit-string, i.e., a binary vector \( \mathbf{x} \) of cardinality \( V = 114 \), where the bits position represents the administration time \( t_i \) and the bit value 1/0 represent vaccine or no-vaccine administration at that time.

In order to allow a better understanding of our SA implementation, we express in more detail the total number \( n \) of vaccine administrations and the total number \( M \) of possible schedules \( x_l^n \), \( (l = 1, \ldots, M) \) with \( n \) vaccine administrations, respectively as:

\[
n = \sum_{j=1}^{V} x_l^n(j)
\]

and

\[
M = \frac{V!}{[n!(V-n)!]}.\]

Now, we can say that a proper counterpart of the SA temperature is \( n \), the number of vaccine administrations, since the temperature in SA is slowly and constantly decreased as much as possible and this is the same requirement for the number of vaccine administrations.

In our problem, a reasonable measure of the energy should be an appropriate mean value of the sample survival time. As we want to maximize survival times, this
would lead to finding a maximum (i.e., maximum energy) while SA is designed for finding a minimum. Otherwise stated, our problem belongs to the class of min-max optimization problems but SA is designed to run towards two minimums: that of the control parameter (lowered by definition) and that of the objective function. How do we cope with this issue?

Taking into account that energy is always a positive quantity, we define the energy $E_l$ of a protocol $x_l^n$ proportional to the inverse of the harmonic mean $H$ (see Eq. (8)) of the survival times $\tau_i$ of the $m$-sample mice:

$$E = H = \frac{m}{\sum_{i=1}^{m} \frac{1}{\tau_i}}$$

where $\tau_i \equiv \tau(x, \lambda_j)$.

A definition of the semi-equilibrium in our implementation is now in order. In SA semi-equilibrium at a certain temperature $T$ is the state where the solid has an energy distribution expressed by Boltzmann distribution (thermal equilibrium).

As far as regard to the definition of semi-equilibrium for optimization over a subset of mice, we say it is obtained by a configuration $(n^*, x_l^{n*}, E_l^{n*})$, such that the sum of the survivals of the subset is not less than $(375 \cdot m)$ days (we recall that the in silico experiment lasts for 400-days).

This means allowing minimal survival of all the subset of mice for at least 375 days without exceeding the thresholds $M_1, M_2$ or, for example, requiring full survival under the thresholds for $m-1$ mice and a survival for at least 600 time-steps under the thresholds for the other mouse. Obviously, other combinations for the cumulative survival may arise as well.

The SA algorithm for vaccine protocol optimization is, therefore, defined in the following way:

1. start from a randomly chosen initial vaccine distribution and find the initial semi-equilibrium configuration $n_0, x_l^{n_0}, E_l^{n_0}$ using the Metropolis algorithm, that finds a semi-equilibrium configuration by a finite sequence $x_l^n, l = 1, \ldots, < \lambda$ (where $\lambda$ is a predefined maximum number of iterations) with bits perturbed randomly and $x_l^{n+1}$ accepted stochastically according to the energy variation and temperature $T$;  
2. decrease the number of injections of 1 unit; 
3. find a semi-equilibrium configuration $x_i, E_i$ according to Metropolis algorithm, (as described in point 1); 
4. cycle on point 2

The algorithm stops when, once the algorithm control parameter, i.e., the number of vaccine administrations, is decreased from $n$ to $n-1$, the Metropolis algorithm is not able to find a semi-equilibrium configuration, i.e., an acceptable value of survivals, in $\lambda$ iterations. The accepted protocol is the last found at temperature $n$. 
As previously remarked, we introduced, for each mouse \(i\) of the sample set, two safety thresholds \(M_1, M_2\) on the number of cancer cells. These thresholds bind to the \textit{in silico} observed number of cancer cells when a Chronic protocol is administered. If, at any time \(t\), the number of cancer cells is greater than any of the safety thresholds the survival time of that mouse \(i\) is \(\tau_i = t\).

5. Computational Results

To obtain comparable results with those obtained by the GA, we tested the SA algorithm using the same 8-mice subset and the same set of 200 mice used in previous experiments. The results for SA have been obtained using the Cometa Grid infrastructure technologies.

For these tests the max number \(\lambda\) of iterations of the Metropolis Algorithm was set to 2000.

The mean immune system behavior induced by the three 37-injections protocols (GA, SA) is shown in Figs. 2 and 3. Apart some slight variations in the number of cytotoxic T cells, the two protocols give rise to a very similar IS response.

![Graphs](image-url)

**Fig. 2.** Mean behavior of B cells (up, left), antibodies (up, right), T helper cells (down, left) and cytotoxic T cells (down, right) using the 37-injection protocol found by the Genetic Algorithm. Day 0 corresponds to the beginning of the \textit{in vivo} experiment, i.e., the sixth week after the mice were born.
Another important factor that should be taken into account is represented by the mean number of cancer cells and antigens which occur during the simulation.

The results exhibited in Figs. 4 and 5 clearly show that the SA algorithm avoids the overcoming of the threshold $M_1$ in the first period of the simulation, whereas
Simulated Annealing and Optimal Protocols

Fig. 5. Mean behavior of cancer cells (left) and antibodies (right) using the 37-injection protocol found by the SA algorithm. Day 0 corresponds to the beginning of the in vivo experiment, i.e., the sixth week after the mice were born. The broken line represents the aforementioned cancer cells thresholds.

<table>
<thead>
<tr>
<th>Injections #</th>
<th>Survival (%)</th>
<th>Time (s)</th>
<th>CPUs #</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>87</td>
<td>259200 s</td>
<td>32</td>
<td>8294400 s</td>
</tr>
<tr>
<td>SA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>83.5</td>
<td>10431 s</td>
<td>8</td>
<td>83448 s</td>
</tr>
<tr>
<td>39</td>
<td>83.5</td>
<td>10433 s</td>
<td>8</td>
<td>83464 s</td>
</tr>
<tr>
<td>38</td>
<td>82</td>
<td>10435 s</td>
<td>8</td>
<td>83480 s</td>
</tr>
<tr>
<td>37</td>
<td>80</td>
<td>10437 s</td>
<td>8</td>
<td>83496 s</td>
</tr>
</tbody>
</table>

the genetic algorithm did not succeed in this job. At the end of the simulation the protocol found by the SA overcomes $M_2$ and becomes the worst due to the lack of injections in the last period.

We tested all the protocols found by the SA with less than 40 injections on the 200-mice set to calculate the survival percentage.

As shown in Table 1, the GA 37-injections protocol has a survival of 87%, whereas the SA 37-injections protocol gives a survival of 80%.

The most relevant result is the great improvement in terms of CPU-time moving from the GA to the SA driven algorithm. As we can see from Table 1 the computational time scaled from 259,200 sec in the GA case down to 10,437 sec for the SA case, even if the SA algorithm ran simply over an 8-nodes cluster while the GA required a 32-nodes one. Since the number of CPUs is different, we calculated simply a comparable time in terms of a single CPU that could be used to run both protocols. The improvement is clear: from 8294,400 sec in GA to 83,496 sec in SA driven algorithm.
This result is really valuable in terms of what we stated at the beginning of this paper: computational tools for clinicians should be achievable even on today consumer market computers, even because, on the other hand, an HPC infrastructure (as well as the expertise needed for its use) is not easily at disposal.

6. Conclusions and Future Work

We have presented an original approach for the search of optimal schedules using computational models.

The use of the Simulated Annealing paradigm, with some biological experience on effects of cancer vaccines, produced a ≈ 10^2 times faster algorithm that can be easily used even without having an HPC infrastructure.

Our protocol assured also fairly comparable survival rates with those obtained by the previous GA-based procedure.

Future work will be focused on improving the algorithm with the aim of achieving (i) higher survival levels, equal or even better than those obtained by the GA and (ii), at the same time, an even faster algorithm. We think that the introduction of a biology driven heuristic may help to achieve these improvements. Efforts in this direction are in progress and will be published in due course.

Acknowledgments

This work was supported under the EC contract FP6-2004-IST-4, No. 028069 (ImmunoGrid).

This work makes use of results produced by the PI2S2 Project managed by the Consorzio COMETA, a project co-funded by the Italian Ministry of University and Research (MIUR) within the Piano Operativo Nazionale “Ricerca Scientifica, Sviluppo Tecnologico, Alta Formazione” (PON 2000–2006). More information is available at: http://www.pi2s2.it and http://www.consorzio-cometa.it.

References


