Appl Intell (2007) 27: 219–235
DOI 10.1007/s10489-007-0069-8

An adaptive evolutionary algorithm with intelligent mutation local searchers for designing multidrug therapies for HIV

Ferrante Neri · Jari Toivanen · Raino A. E. Mäkinen

Published online: 16 June 2007

Abstract This paper proposes a novel Memetic Algorithm consisting of an Adaptive Evolutionary Algorithm (AEA) with three Intelligent Mutation Local Searchers (IMLSs) for designing optimal multidrug Structured Treatment Interruption (STI) therapies for Human Immunodeficiency Virus (HIV) infection. The AEA is an evolutionary algorithm with a dynamic parameter setting. The adaptive use of the local searchers helps the evolutionary process in the search of a global optimum. The adaptive rule is based on a phenotypical diversity measure of the population. The proposed algorithm has been tested for determining optimal 750-day pharmacological protocols for HIV patients. The pathogenesis of HIV is described by a system of differential equations including a model for an immune response. The multidrug therapies use reverse transcriptase inhibitor and protease inhibitor anti-HIV drugs. The medical protocol designed by the proposed algorithm leads to a strong immune response; the patient reaches a "healthy" state in one and half years and after this the STI medications can be discontinued. A comparison with a specific heuristic method and a standard Genetic Algorithm (GA) has been performed. Unlike the heuristic, the AEA with IMLSs does not impose any restrictions on the therapies in order to reduce the dimension of the problem. Unlike the GA, the AEA with IMLSs can overcome the problem of premature convergence to a suboptimal medical treatment. The results show that the therapies designed by the AEA lead to a "healthy" state faster than with the other methods. The statistical analysis confirms the computational effectiveness of the algorithm.

F. Neri (✉) · J. Toivanen · R.A.E. Mäkinen
Department of Mathematical Information Technology, Agora University of Jyväskylä, 40014 Jyväskylä, Finland
e-mail: neferran@cc.jyu.fi

F. Neri
Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Via E. Orabona, 4, 70125 Bari, Italy
e-mail: neri@deemail.poliba.it

J. Toivanen
Center for Research in Scientific Computation, North Carolina State University, Box 8205, Raleigh, NC 27695-8205, USA

1 Introduction

Medical treatments for Human Immunodeficiency Virus (HIV) have greatly improved during the last two decades. Typically they can prolong the time before the onset of Acquired Immune Deficiency Syndrome (AIDS) for tens of years. Particularly, the prevailing medical practice is to prescribe Highly Active Anti-Retroviral Therapy (HAART) which can keep the viral load low and maintain high CD4+ T-cell counts. This therapy is a combination of three or more drugs which are called "cocktails". Some patients develop resistance to one or more of the drugs in which case it is necessary to change the composition of HAART. Sometimes there are severe side effects from the medications which again lead to stopping or a change of medications. The cost of HAART is often prohibitively high in developing countries. Due to these and other reasons, the search for alternative treatments is very active. This paper studies dynamic multidrug therapies that stimulate an immune response which can suppress HIV by itself after the therapy has ended.

Human immunodeficiency viruses infect CD4+ T-cells, which are an important part of the human immune system, and other target cells. The infected cells produce a large number of viruses. Currently the two most important categories of anti-HIV drugs are Reverse Transcriptase

Inhibitors (RTIs) and Protease Inhibitors (PIs). A typical HAART cocktail consists of one or more RTIs and a PI. The reverse transcriptase inhibitors prevent HIV from infecting cells by blocking the integration of the viral code into the target cells. Protease inhibitors interfere with the replication of viruses by infected cells. Viruses are still produced, but they are non infectious, that is, they are not capable of infecting cells. In practice, RTIs cannot block viruses completely from infecting target cells. Also, some infectious viruses are produced under a PI medication. Each drug has a maximum efficacy which depends on many factors like virus strains present. By varying the dosage it is possible to change the efficacy of the medication between no effect and the maximum efficacy.

Many differential equation models of HIV pathogenesis have been developed. Some of these models used to design dynamical drug treatments are presented in [1–7]; see also the collection [8]. In the long term pathogenesis of HIV an immune response can play an important role. The models in [3, 4] do not contain an immune response while the models in [1, 2, 5–7] do. The immune mechanism responding to HIV is not yet very well understood and due to this the proposed models vary. For our study of designing medical treatments to stimulate a strong immune response it is mandatory to model immune systems as well.

Already tens of papers have been written on using control techniques for planning HIV therapies. The papers [1, 9–13] consider only RTI medication while the papers [14, 15] consider only PIs. In [16–19] all effects of a HAART medication are combined to one control variable in the model. In [20–25] dynamical multidrug therapies based on RTIs and PIs are designed. In these therapies the dosage of both medications can change independently of each other. This paper studies these kinds of dynamical multidrug therapies.

In the considered control approaches the amount of medications can be either continuous or on-off-type. The second type is called Structured Treatment Interruption (STI) and it has been extensively studied in medical literature, see [26] and references therein. The main argument to use STI medications instead of continuously varying dosage is to lower the risk of HIV mutating to strains which are resistant to the current medication regime. Studies of continuously varying medical therapies have been more common, see [1, 7, 9–12, 14, 15, 22, 23, 25, 27, 28]. Structured treatment interruption schedules have been considered in [1, 17–20, 24]. Here STI medications using RTIs and PIs are studied.

This paper considers the design of STI multidrug therapies for HIV using the same model problem as in [20] where the therapies were constructed using a heuristic optimization method. The medical rationale of this model problem is to find STIs which expose the immune system to such a level of HIV and infected cells that it stimulates a strong immune response. This is a delicate task, since too low a level of HIV does not stimulate a response and too high a level impairs it. With a strong response the immune system can subdue the HIV after initial therapy. This was observed for the first time in the case of "Berlin patient" who interrupted HAART medications twice and after stopping medications permanently the viral load stayed low [29]. The heuristic method in [20] restricts the length of treatment periods and off treatment periods to be a multiple of five days and then it performs consecutively the optimization of STI for 30 day subperiods until 750 days is reached. Due to these simplifications the heuristic can lead to suboptimal treatment schedules far from an optimal. In order to increase the robustness and quality of therapies, this paper proposes a computational intelligence algorithm for the original model problem without making any simplified restrictions.

In [24], a Genetic Algorithm (GA) was used to design STI therapies based on a different HIV model when the cardinality of the combinatorial optimization problem was very modest. The algorithm led to satisfactory results in this case. Nevertheless, for a more complex and accurate model and therefore for an objective function having a large number of variables to be optimized, a GA could easily fail. Due to the multimodality of the fitness landscape and the high cardinality of the decision space, a GA would probably stagnate or converge to a suboptimal solution [30–34]. In order to avoid these two undesirable behaviors, a proper tuning of the algorithm's parameters, such as the size of the population and the probability of the mutation, would be required. On the other hand, as happens in many hard to solve problems [30], any static set of parameters having the values fixed during an optimization run seems to be inappropriate for the following two reasons. The first one is that a parameter tuning must be done by running trial simulations; this process, that is in general time consuming, can lead to an unacceptable calculation time for hard to solve problems. The second one is that different values of parameters might be optimal at different stages of the evolutionary process [35–40].

This paper proposes a Memetic approach [30, 41–43] consisting of an Adaptive Evolutionary Algorithm (AEA) with three Intelligent Mutation Local Searchers (IMLS) for designing optimal STI multidrug HIV therapies. The adaptive rules are based on a dynamic measurement of the phenotypical diversity of the population and thus on the state of the phenotypical convergence of the algorithm. The AEA adaptively chooses the size of the population and the probability of mutation. Moreover it makes use of three Intelligent Mutation Local Searchers [41] adaptively executed according to the necessities of the evolutionary process. These three local searchers have different features and they have the role of increasing the population diversity [44], improving the performance of some solutions during their "life-time" [45, 46], executing the "endgame" hill climb [45] to finalize the optimization process [41]. The main idea of the AEA with

IMLSs is to prevent the premature convergence and stagnation by a control based on the fitness values of the population. This control has to dynamically balance the needs of exploration and exploitation taking into account the state of the evolution [47]. The goal of this adaptive control is achieved by means of the combination of two different algorithmic philosophies: the use of adaptive conventional parameters (e.g. dynamic "aggressiveness" of the mutation) and the adaptive use of Intelligent Mutation Local Searchers.

This paper is organized in the following six sections. Section 2 describes a differential model for the pathogenesis of HIV and formulates the problem of designing HIV therapies as an optimization problem. Section 3 presents the heuristic algorithm previously proposed to solve the problem formulated in Sect. 2. Sections 4 and 5 describe and comment on the proposed AEA with IMLSs analyzing the problem of the parameter setting of the algorithm. Section 6 contains the description of the experimental design, the optimized medical therapy, the associated pathogenesis of HIV, and the algorithmic performance of the AEA with IMLSs. A statistical analysis showing the superiority of the proposed algorithm is also presented in Sect. 6. The conclusions in Sect. 7 discuss the numerical results, the efficiency of each algorithmic component and the practical implications on the medical research of the novel computational intelligence algorithm proposed here.

2 HIV model and optimization problem

In this paper, the pathogenesis of HIV is modeled with a system of Ordinary Differential Equations (ODEs) described in [1, 20, 21]. This model is a combination of the two target models in [48] and the immune response model in [2]. It captures many of the observed behavioral properties of long term HIV dynamics described in [2, 48]. The time dependent control variables are the efficacies of the Reverse Transcriptase Inhibitor (RTI) and Protease Inhibitor (PI) medications denoted by ϵ_α and ϵ_β, respectively. The system of ODEs describing the model reads

$$\begin{aligned}
\dot{T}_1 &= \lambda_1 - d_1 T_1 - (1-\epsilon_\alpha) k_1 V T_1, \\
\dot{T}_2 &= \lambda_2 - d_2 T_2 - (1 - f\epsilon_\alpha) k_2 V T_2, \\
\dot{T}_1^* &= (1-\epsilon_\alpha) k_1 V T_1 - \delta T_1^* - m_1 E T_1^*, \\
\dot{T}_2^* &= (1 - f\epsilon_\alpha) k_2 V T_2 - \delta T_2^* - m_2 E T_2^*, \\
\dot{V} &= (1-\epsilon_\beta) N_T \delta (T_1^* + T_2^*) - cV \\
&\quad - [(1-\epsilon_\alpha)\rho_1 k_1 T_1 + (1 - f\epsilon_\alpha)\rho_2 k_2 T_2] V, \\
\dot{E} &= \lambda_E + \left(b_E \frac{T_1^* + T_2^*}{T_1^* + T_2^* + K_b} - d_E \frac{T_1^* + T_2^*}{T_1^* + T_2^* + K_d} \right) E \\
&\quad - \delta_E E.
\end{aligned} \tag{1}$$

Each state variable is associated with a compartment and they have the following meanings: T_1 is uninfected CD4+ T-cells, T_2 is uninfected target cells of second kind, T_1^* is infected T-cells, T_2^* is infected target cells of second kind, V is human immunodeficiency viruses, and E is immune effectors. We use milliliter (ml) as a volume unit. The parameter values are given in Table 1. In practice, HIV medications are not perfectly efficient, that is, some target cells are infected under a RTI medication and some infectious viruses are produced under a PI medication. For the efficacies ϵ_α and ϵ_β this means that they satisfy the inequalities $0 \le \epsilon_\alpha \le \epsilon_\alpha^{\max}$ and $0 \le \epsilon_\beta \le \epsilon_\beta^{\max}$ for some maximum efficacies $\epsilon_\alpha^{\max} < 1$ and $\epsilon_\beta^{\max} < 1$. In the considered model problem, their values are $\epsilon_\alpha^{\max} = 0.7$ and $\epsilon_\beta^{\max} = 0.3$.

Most of the terms in the model (1) have straightforward interpretations [2, 48]. The diagram in Fig. 1 shows the interactions between the compartments. The positive terms λ_1, λ_2, and λ_E in the first, second, and last equation in (1) correspond to the production of new T-cells, type 2 cells, and

Table 1 Parameter Values for the HIV Model

	Value	Unit		Value	Unit
λ_1	10000	cells/ml day	λ_2	31.98	cells/ml day
d_1	0.01	day^{-1}	d_2	0.01	day^{-1}
k_1	8.0×10^{-7}	ml/viruses day	k_2	1.0×10^{-4}	ml/viruses day
m_1	1.0×10^{-5}	ml/cells day	m_2	1.0×10^{-5}	ml/cells day
ρ_1	1	viruses/cells	ρ_2	1	viruses/cells
δ	0.7	day^{-1}	c	13.0	day^{-1}
f	0.34	–	N_T	100.0	viruses/cells
λ_E	1.0	cells/ml day	δ_E	0.1	day^{-1}
b_E	0.3	day^{-1}	d_E	0.25	day^{-1}
K_b	100	cells/ml	K_d	500	cells/ml

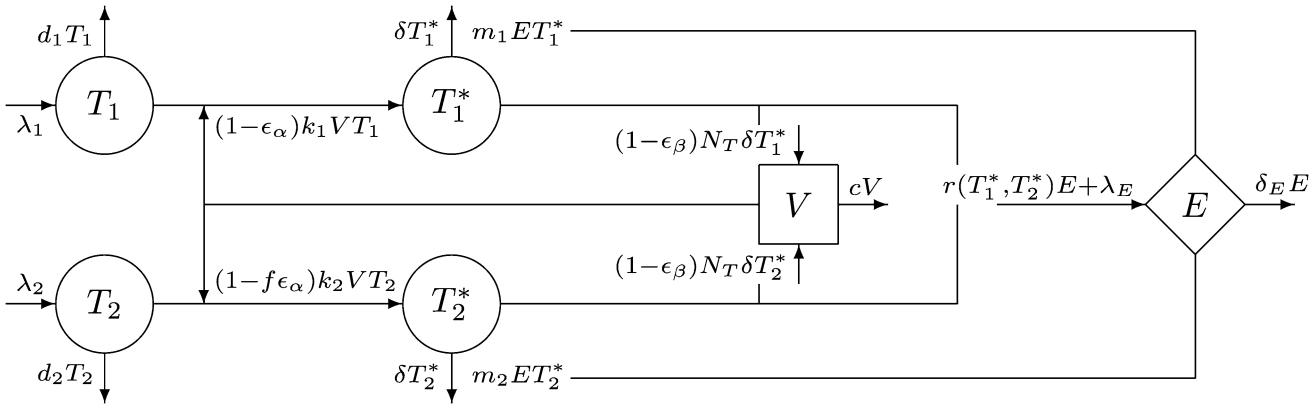

Fig. 1 A diagram of the HIV model, where the compartments are: T_1 uninfected T-cells, T_2 uninfected type 2 cells, T_1^* infected T-cells, T_2^* infected type 2 cells, V viruses, and E immune effectors. The function r is $r(T_1^*, T_2^*) = b_E(T_1^* + T_2^*)/(T_1^* + T_2^* + K_b) - d_E(T_1^* + T_2^*)/(T_1^* + T_2^* + K_d)$

immune effectors, respectively. For example, T-cells are produced by bone marrow. The negative terms containing coefficients d_1, d_2, δ, c, and δ_E present the death/clearance of cells/viruses. The HIV infection of T-cells is described by the terms with $(1-\epsilon_\alpha)k_1VT_1$. In the first equation of (1), the minus sign means that infection decreases the number of healthy cells. The product VT_1 can be considered to present the probability of HIV to encounter T-cells. Other product terms have similar interpretations. The coefficient $(1-\epsilon_\alpha)$ means that higher RTI efficacy reduces the probability of infection. Particularly, a perfectly efficient RTI ($\epsilon_\alpha = 1$) would completely prevent the infection of T-cells. The most complicated term in the model is

$$\left(b_E\frac{T_1^*+T_2^*}{T_1^*+T_2^*+K_b} - d_E\frac{T_1^*+T_2^*}{T_1^*+T_2^*+K_d}\right)E.$$

The first quotient presents the stimulation of immune effector production due to the presence of infected cells while the second quotient describes how a high amount of infected cells impairs the production [2]. The concentration of infected cells has to be in a parameter dependent range for the production rate to be larger than the clearance rate, that is, to have a strengthening immune response. Thus, the anti-HIV therapy cannot be so strong that the amount of infected cells falls below the range in which an immune response occurs. On the other hand, if the medication is too inefficacious then the response is impaired by a high concentration of infected cells.

Without any medication ($\epsilon_\alpha = \epsilon_\beta = 0$) the model (1) has several steady states, that is, such states that all time derivatives $\dot{T}_1$, $\dot{T}_2$, $\dot{T}_1^*$, $\dot{T}_2^*$, $\dot{V}$, and $\dot{E}$ in (1) are zero. For the parameters given in Table 1 they are described and analyzed in [1, 20]. A particularly interesting state is the so-called "healthy" steady state given by

$$\begin{aligned} &T_1 = 967839 \text{ cells/ml}, \qquad T_2 = 621 \text{ cells/ml}, \\ &T_1^* = 76 \text{ cells/ml}, \qquad T_2^* = 6 \text{ cells/ml}, \\ &V = 415 \text{ viruses/ml}, \quad \text{and} \quad E = 353108 \text{ cells/ml}. \end{aligned} \tag{2}$$

Under the model (1) and the given parameters a person without an HIV infection has the T-cell count T_1 equal to a million cells per milliliter. The value of T_1 in (2) is close to that which is the reason for calling this state "healthy". Furthermore, the viral load V in (2) is reasonably low.

A Highly Active Anti-Retroviral Therapy (HAART) can lead to a low viral load, but it cannot completely clear HIV; see [48], for example. Therefore, it is not a realistic goal to try to eradicate HIV using therapy. It can be claimed that a person infected with HIV can live in the "healthy" state (2) for a long time without medical problems due to HIV and without any medication for HIV. Hence, it would be highly desirable to design multidrug therapies which would steer the medical condition towards the "healthy" state. Once a neighborhood of this state is reached the therapy can be discontinued due to the local asymptotic stability of (2).

Our aim is to find effective multidrug therapies by minimizing a sum

$$J = \sum_{i=1}^{4} w_i J_i, \tag{3}$$

where

$$\begin{aligned} J_1 &= \int_0^T V\,dt, \qquad J_2 = \int_0^T E\,dt, \\ J_3 &= \int_0^T \epsilon_\alpha^2\,dt, \qquad J_4 = \int_0^T \epsilon_\beta^2\,dt. \end{aligned} \tag{4}$$

In (4), T is a given time horizon, V is the number of free viruses, E is the measure of the immune response, ϵ_α is the efficacy of RTI, and ϵ_β is the efficacy of PI. Thus, J_1 and

J_2 measure the amount of viruses and immune effectors, respectively, over the time interval $[0, T]$. Similarly, J_3 and J_4 measure the amount of the RTI and PI medications, respectively, over the same time interval. The minimization of J can be seen as a scalarized multiobjective optimization problem [49]. Here the same weights $w_1 = 0.1$, $w_2 = -1000$, $w_3 = w_4 = 20000$ in (3) are used as in [20]. Thus, the aim is to minimize the amount of viruses and medications while trying to maximize the immune response. The goal is to steer the state to the "healthy" state by finding a dynamical medication which minimizes J.

This paper considers Structured Treatment Interruption (STI) schedules for medications. Thus, either a patient receives the maximum dose of a medicine or none at all. For the schedule to be practical the decision to take medication is made for one day intervals. Thus, the RTI medication can be described by a vector α containing binary numbers telling whether the RTI medication is taken on ith day ($\alpha_i = 1$) or not ($\alpha_i = 0$), $i = 0, \ldots, T$. In the same way the PI medication can be defined using a vector β. The efficacies change linearly from one day to the next. These transition periods can be considered as the time required for the drug to be fully absorbed and conversely the time required for the drug to be cleared from the body. Thus, at time t the efficacies are

$$\epsilon_\alpha = [(i + 1 - t)\alpha_i + (t - i)\alpha_{i+1}]\epsilon_\alpha^{\max}$$

and

$$\epsilon_\beta = [(i + 1 - t)\beta_i + (t - i)\beta_{i+1}]\epsilon_\beta^{\max},$$

where $i = \lfloor t \rfloor$, that is, i is the largest integer less than or equal to t.

We denote the spaces for the vectors α and β by

$$D_1^{T+1} = \{0, 1\}^{T+1},$$

where the subscript 1 signifies that the medications are fixed for one day periods and the superscript $T + 1$ means that the medication decision has to be made for $T + 1$ days which also includes the starting day for the medications. Now the optimal control problem for finding a dynamical multidrug therapy defined by the vectors α and β reads

$$\min_{\alpha \in D_1^{T+1}, \beta \in D_1^{T+1}} J(\alpha, \beta) \tag{5}$$

subject to the state equation (1) and a given initial condition which is chosen to be the acute infection

$$\begin{aligned} &T_1 = 10^6 \text{ cells/ml}, \qquad T_2 = 3198 \text{ cells/ml}, \\ &T_1^* = 10^{-4} \text{ cells/ml}, \qquad T_2^* = 10^{-4} \text{ cells/ml}, \\ &V = 1 \text{ viruses/ml}, \qquad E = 10 \text{ cells/ml}. \end{aligned} \tag{6}$$

The time discretization of the state (1) is performed using a second-order backward differentiation formula (BDF2) [50]. For the numerical results 30 minute time steps are used, that is, for each day 48 time steps are performed.

The cardinality of the decision space $D = (D_1^{T+1})^2$ is given by $\text{card}(D) = 2^{2\cdot(T+1)}$, since it can be easily seen that $\text{card}(D_1^{T+1}) = 2^{T+1}$. Thus, the cardinality of the decision space is very high and this can make the solution of the optimization problem very difficult. Moreover, since each evaluation of the objective function J requires the solution of a system of differential equations, it is computationally expensive (each fitness evaluation takes about 0.2 seconds on a PC with a 3 GHz processor). Finding a suitable optimization algorithm is therefore a challenging task because the global optimization process can be very time consuming and moreover there is a quite high risk that, due to the presence of many variables, the algorithm could converge to a suboptimal solution.

3 Background: the heuristic method

The solution to the optimization problem (5) was approximated using the following heuristic method for 750 day period ($T = 750$), proposed in [20]. The first step for the heuristic is to restrict the therapies so that they can change only every five days instead of every day. This means that the optimization problem (5) is replaced by

$$\min_{\alpha \in D_5^{T+1}, \beta \in D_5^{T+1}} J(\alpha, \beta), \tag{7}$$

where D_5^{T+1} is given by

$$D_5^{T+1} = \{(0\,0\,0\,0\,0)^T, (1\,1\,1\,1\,1)^T\}^{\lceil (T+1)/5 \rceil},$$

and $\lceil (T + 1)/5 \rceil = 151$ for $T = 750$. Therefore, the cardinality of the decision space is reduced to $2^{2\lceil (T+1)/5 \rceil}$. Nevertheless the cardinality is still very high and the reduced problem (7) is difficult to solve.

The second step for the heuristic method is to consider a sequence of subperiods instead of the whole period $[0, T]$. More precisely, 750 days are divided into 25 separate 30 day periods. Then, the heuristic optimizes the medications for each subperiod consecutively by using the HIV pathogenesis from the previous subperiods with the optimized medications as medical history. In the following, we describe this procedure more precisely. We denote the vectors defining the RTI and PI medication schedules for the kth subperiod by α^k and β^k, respectively. They both consist of 30 binary numbers belonging to the space

$$D_5^{30} = \{(0\,0\,0\,0\,0)^T, (1\,1\,1\,1\,1)^T\}^6.$$

Now the optimization problem for the kth subperiod reads

$$\min_{\alpha^k \in D_5^{30}, \beta^k \in D_5^{30}} J^{30k}(\alpha, \beta),$$

$$\text{where } \alpha = \begin{pmatrix} \alpha^1 \\ \vdots \\ \alpha^k \\ 0 \end{pmatrix}, \ \beta = \begin{pmatrix} \beta^1 \\ \vdots \\ \beta^k \\ 0 \end{pmatrix} \tag{8}$$

and J^{30k} is the same objective function as in (3) except that the integrations are performed over the time interval $[0, 30k]$ instead of $[0, T]$. In (8), α^i and β^i, $i = 1, \ldots, k-1$, are the optimized medications from the previous subperiods. Thus, each optimization problem finds a medication schedule for a 30 day interval while the medication up to that interval is obtained from the previous optimizations. Each 30 day interval contains six five day intervals. Thus, for one medication there are 2^6 possible on-off-combinations for a 30 day interval and the cardinality of the decision space for both medications together is $(2^6)^2 = 4096$. This cardinality is sufficiently small so that an exhaustive search can be performed to solve the optimization problems in (8) for $k = 1, \ldots, 25$.

The heuristic method computes and compares all solutions belonging to a drastically restricted decision space, since its cardinality is reduced from 2^{1502} to 25×2^{12}, that is, by the factor 1.37×10^{447}. This reduction is motivated by empirical observations made with the specific case under study. Therefore, it can lead to unsatisfactory results in other cases. Under other model parameters corresponding to a different person's physiology the medication for five day periods might not offer enough flexibility for stimulating strong immune response. Also, the subperiod simplification and fairly short subperiods may cause lack of robustness, that is, the heuristic may fail in finding an existing medication schedule stimulating a strong immune response. Thus, the use of the heuristic in a general case is questionable without extensive further study.

4 Adaptive evolutionary algorithm with intelligent mutation local searchers

In order to solve the problem in (5), a computational intelligence approach is proposed in this paper. Instead of making hypotheses reducing the cardinality of the decision space, the optimal solution search is carried out in the original decision space D by means of an intelligent evolutionary algorithm which aims to find the global optimum of $J(\alpha, \beta)$ without performing a large number of objective function evaluations.

In [51] the methods of performing the control of the algorithmic parameters are classified into three categories:

- *deterministic parameter control*: this rule modifies the strategy parameter in a deterministic way without any feedback from the search,
- *adaptive parameter control*: this takes place when there is some form of feedback from the search that serves as inputs to a mechanism used to determine the direction or the magnitude of the change to the strategy parameter,
- *self-adaptive parameter control*: the parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination.

Following the definition given in [51], an Adaptive Evolutionary Algorithm (AEA) with Intelligent Mutation Local Searchers (IMLS) [41] is proposed for performing the minimization of the objective function J in (3).

This AEA with IMLSs consists of the following. The first set of sampling points (α, β) of the decision space D is chosen pseudo-randomly under uniform distribution. This sampling concerns a set of S_{pop} pairs of vectors whose genes are binary numbers which represent the on-off medication for each day of the therapy. In our evolutionary process, each individual is made up then of a pair of the chromosomes α and β.

With the first generation, the fitness function J is calculated for all the individuals of the initial population and the following index is calculated (see [34]):

$$\xi = \min\left\{ \left| \frac{J_{\text{best}} - J_{\text{avg}}}{J_{\text{best}}} \right|, 1 \right\} \tag{9}$$

where J_{best} and J_{avg} are respectively the best and average fitness among the fitness values of the population. The index ξ is a fitness based measurement of the phenotypical diversity of the population and it can be seen as a measurement of the state of the phenotypical convergence of the algorithm (see for details [34] and [52]). If $\xi \approx 1$ there is a high phenotypical diversity and therefore the convergence conditions are far; if $\xi \approx 0$ there is a low phenotypical diversity and means that the convergence is approaching. The index ξ is thus used, as will be shown, to adaptively tune the algorithmic parameters in order to handle the multivariate fitness landscape of J.

At each subsequent generation, the individuals undergo ranking parent selection [53, 54] using the stochastic universal sampling algorithm [55] and the selected individuals undergo the two-point crossover [56]. This leads to $S_{\text{pop}}/2$ crossovers at each generation which are performed in the following way. For the first chromosome of two parent solutions α_{par1} and α_{par2}, two different cutting points are chosen pseudo-randomly and each chromosome is divided in three substrings. The crossover is performed exchanging the middle substring between the chromosomes, as shown in the

following example:

$$\begin{array}{lll} \alpha_{\text{par1}} & 0\,1\,1 \vdots\, 0\,0\,0\,1\,0\,0 \vdots\, 1 & \\ \alpha_{\text{par2}} & 1\,1\,1 \vdots\, 0\,0\,1\,0\,0\,1 \vdots\, 0 & \end{array} \rightarrow \begin{array}{ll} \alpha_{\text{off1}} & 0\,1\,1 \vdots\, 0\,0\,1\,0\,0\,1 \vdots\, 1 \\ \alpha_{\text{off2}} & 1\,1\,1 \vdots\, 0\,0\,0\,1\,0\,0 \vdots\, 0 \end{array}$$

In the same way, the two-point crossover is also applied to the chromosome β. The choice of the two point crossover is due to one empirical consideration based on the physical meaning of the problem. In fact, it has been observed that the best individuals contain many zeros in the tail of the chromosome and many ones in the head of the chromosome. This genotype means that the patient is supposed to take the medicines according to the medications α and β at the beginning of the therapy and to discontinue the therapy as soon as possible (the minimization of J_3 and J_4). It is thus clear that a recombination operator which is explorative in the same way on all the bits of the chromosomes (e.g. the uniform crossover) risks spoiling the genotype of good solutions and on the other hand a one point crossover technique could turn out not explorative enough in such a high cardinality decision space.

When all the crossovers are executed the mutation probability is calculated by means of the following formula:

$$p_m = p_m^{\max} \cdot (1 - \xi). \tag{10}$$

The role of this dynamic mutation probability is to increase the explorative pressure in the presence of low phenotypical diversity ($\xi \approx 0$) and to decrease it in the presence of high phenotypical diversity ($\xi \approx 1$). The value $p_m^{\max}$ is a parameter that has been tuned, as will be shown in Sect. 5, by analyzing its effect on the algorithmic performance.

For each newly generated offspring individual a pseudo-random number between 0 and 1 is generated by means of a uniformly distributed probability function. The mutation occurs on the offspring individual under analysis if this pseudo-random number is smaller than p_m.

The bit-wise mutation operator which changes one 0 to 1 and vice versa has been implemented making use of the *mutation clock* operator. The position of the bit where the mutation is applied depends on the position of the previously mutated bit and on a random number returned by a probability function (for details see [57, 58]). The fitness function J is then calculated for all the offspring individuals and the population made up of both parents and offspring are sorted according to the their fitness value.

In order to explore the neighborhood of the best solutions and, therefore, to increase the search pressure in the directions of the search the AEA makes use also of three Intelligent Mutation Local Searchers (IMLSs) [30, 41] adaptively activated by a criterion based on the value of ξ. The conditions for the use of the IMLSs are the following:

(a) If ($\xi < 0.5$)
the *Increasing Diversity* IMLS is applied to $N^{\text{ID}} = \text{round}(pr^{\text{ID}} \cdot S_{\text{pop}} \cdot (1 - \xi))$ individuals of the population and the solutions returned by the IMLS are inserted in the starting population. pr^{ID} is a parameter to be tuned (see Sect. 5) which represents the maximum proportion of the population undergoing this IMLS.

(b) If ($0.1 < \xi < 0.5$)
the *Greedy Descent* IMLS is applied to one individual of the population pseudo-randomly chosen.

(c) If ($\xi < 0.01$)
the *Steepest Descent* IMLS is applied to the best individual of the population.

The three IMLS are described in the following.

The *Increasing Diversity* IMLS picks out one candidate solution and pseudo-randomly changes 5 bits. The mutated individual is inserted in the population with a probability given by the following formula (see [44]):

$$P(accept) = \begin{cases} 1, & \text{if } \Delta E > 0, \\ e^{k_n \Delta E / (J_{\text{best}} - J_{\text{avg}})}, & \text{otherwise} \end{cases} \tag{11}$$

where $\Delta E = |J_{\text{original}} - J_{\text{mutated}}|$ is the difference between the fitness values of the individual before and after the mutation and k_n is a normalization factor. It is important to notice that the newly mutated solution does not replace the original one but it is simply inserted in the population. The condition $\xi < 0.5$ means that this operator is activated when the phenotypical diversity of the individuals is decreasing and then the phenotypical convergence is approaching. The meaning of this local searcher is that, in the case of decreasing of the phenotypical diversity (it could correspond to a convergence to a suboptimal solution) some good individuals which increase the population diversity are introduced. It can be observed from (11) that the *Increasing Diversity* IMLS could accept solutions which are not as good. This feature makes this local searcher highly explorative and its role can be crucial in a condition of suboptimal convergence. This IMLS is computationally rather cheap (each application costs one fitness evaluation) and it is applied to several individuals of the population taking into account the current phenotypical diversity. The maximum population rate which undergo the *Increasing Diversity* IMLS is given by pr^{ID} (see Sect. 5).

The *Greedy Descent* IMLS picks out one candidate solution and pseudo-randomly changes 2 bits. If the mutation is successful, meaning the mutated individual has better performance than the original one, the newly mutated individual replaces the original one and the local searcher exits from the loop, otherwise 2 other bits are chosen pseudo-randomly and the process is repeated for the original individual. A stop criterion ensures the exit from the loop after a certain number (in our case 50) of unsuccessful trials. This

local searcher aims to help the evolutionary process improving in a Lamarckian logic the performance of some candidate solutions. The greedy ascent pivot rule ensures that the computational cost of this operator will not be excessive. The conditions concerning the coefficient ξ ($0.1 < \xi < 0.5$) have been given according to the following semi-empirical consideration. If $\xi \geq 0.5$ the phenotypical diversity is rather high and the evolutionary operators do not really need the help of a local searcher, that can turn out unnecessary, in consideration of its computational cost compared to the cost of one ordinary mutation. If $0.1 < \xi < 0.5$, the *Greedy Descent* IMLS is really effective because it suggests to the evolutionary process new good search directions. If $\xi \leq 0.1$ the phenotypical diversity is quite low and therefore a convergence, to an optimal or suboptimal solution, is likely approaching. Since due to its inner structure a greedy local searcher could suggest a wrong search direction [59], its use in phenotypical convergence conditions could spoil the right search direction taken by the evolutionary algorithm. It is important to remark that if $\xi \approx 0$, the role of exploring the decision space in order to jump out from a possible suboptimal basin of attraction is entrusted on an aggressive mutation probability, on the *Increasing Diversity* IMLS and, as it will be shown, on a large population size that are for this aim more effective and computationally cheaper than the *Greedy Descent* IMLS. Each application of this IMLS has a computational cost compared between 1 and 50 fitness evaluations.

The *Steepest Descent* IMLS works on the best individual of the population. For a given solution (α, β), each bit is flipped one-by-one and, thus, an auxiliary population of $2 \cdot (T + 1)$ individuals having a genotype differing by one bit (hamming distance equal to one) from the starting solution is obtained. The value of the fitness function J is calculated for all these $2 \cdot (T + 1)$ individuals. If the best individual among the mutated ones has a higher fitness value than the starting solution, the replacement occurs. Otherwise, this operator fails and there is no replacement. The condition $\xi < 0.01$ is given because of two reasons: the first is that this operator is computationally very expensive and the use of it must be done only in extreme conditions, the second is that this operator aims to "end the game" [45] in the optimization process. In fact, if $\xi < 0.01$ the phenotypical convergence is almost reached notwithstanding the aggressive mutation, the *Increasing Diversity* IMLS and the large population size. When the convergence is approaching an evolutionary algorithm is less efficient than at the beginning of the optimization process [30] and therefore, even if expensive (each application of this IMLS costs $2 \cdot (T + 1)$ fitness evaluations), the *Steepest Descent* IMLS can likely be more efficient (and quicker) than several generations of evolutionary algorithm.

The main idea is that local searchers with different features (e.g. different pivot rule and individual of application) should explore the decision space from different perspectives [44, 59–61] following the necessities of the optimization process. By the pivot rule of a local searcher we mean the criteria for accepting an improving point [41]. A local searcher employing a steepest descent pivot rule selects the search direction after having explored the entire neighborhood of the current best point. A local searcher employing a greedy descent pivot rule selects the search direction as soon as an improved neighbor solution has been found.

These IMLSs are supposed to "compete and cooperate" [62] in order to support the evolutionary algorithm in convergence to the global optimum. The IMLSs work on the same population of individuals attempting to enhance the performance one or more solutions and thus "cooperate" in the search for the global optimum. At the same time, the IMLSs have different working principles in detecting the search directions and thus they "compete" in the sense that the individuals enhanced by the local searchers will compete in surviving in the subsequent generation.

It is important to remark that the three IMLSs work on a different amount of bits. The exact number of bits (5, 2, 1) has been set on the basis of preliminary tests and according to the following algorithmic philosophy. The idea is that the *Increasing Diversity* IMLS, which is supposed to be the most explorative operator, works on 5 bits, the *Steepest Descent* IMLS, which is supposed to be the most exploitative operator,works on only 1 bit (it tries to enhance one solution exploiting almost all the genotype), the *Greedy Descent* IMLS, which is supposed to be between the previous ones in terms of exploration/exploitation, works on 2 bits.

The value of the population size is then updated in order to perform the survivor selection. The population size is calculated according the following formula:

$$S_{\text{pop}} = S^{\text{f}}_{\text{pop}} + S^{\text{v}}_{\text{pop}} \cdot (1 - \xi), \tag{12}$$

where $S^{\text{f}}_{\text{pop}}$ and $S^{\text{v}}_{\text{pop}}$ are the fixed minimum and maximum sizes of the variable population, respectively. The coefficient ξ is then used to dynamically set the population size [35, 38, 63] in order to inhibit premature convergence and stagnation. If $\xi \approx 1$ the population is phenotypically highly diverse and, thus, a small number of solutions need to be exploited, if $\xi \approx 0$ the population is converging and a larger population size is required to increase the exploration. The S_{pop} best individuals of the population are thus selected to survive for the subsequent generation [64, 65].

The index ξ is then updated according to (9) for the subsequent generation. The algorithm is stopped when $\xi = 0$ and the *Steepest Descent* IMLS fails for a given number of generations (in our case 10).

As explained in [34], the index ξ is fitness based and thus does not ensure a correct guess on the genotypical convergence of the algorithm. In other words, it would return the

begin-AEA with IMLSs
create initial population;
fitness evaluations of J for all the individuals
calculation of $\xi = \min\left\{1, \left|\frac{J_{best} - f_{avg}}{J_{best}}\right|\right\}$;
while (conditions)
parents selection by ranking;
recombination by the two point crossover;
calculation of the mutation probability $p_m = p_m^{max} \cdot (1 - \xi)$;
mutation;
fitness J evaluations of the offspring;
if $\xi < 0.5$
execute *Increasing Diversity* IMLS on $N^{ID} = \text{round}\left(pr^{ID} \cdot Spop \cdot (1 - \xi)\right)$ individuals;
if $\xi > 0.1$
execute the *Greedy Descent* IMLS on one individual randomly chosen;
end-if
if $\xi < 0.01$
execute the *Steepest Descent* IMLS on the best individual;
end-if
end-if
calculation of $S_{pop} = S_{pop}^f + S_{pop}^v \cdot (1 - \xi)$;
survivor selection of the S_{pop} best individuals;
calculation of $\xi = \min\left\{1, \left|\frac{J_{best} - J_{avg}}{J_{best}}\right|\right\}$;
end-while
end-AEA with IMLSs

Fig. 2 Pseudocode of the AEA with IMLSs

value 0 if all the population is made up of individuals having different genotypes but the same fitness values (plateau areas of the decision space, saddle points etc.). On the other hand, this index requires a low computational cost compared to any genotypical indicator (e.g. measurement of the genotypic distance in fitness sharing) and it is very easy to calculate. Moreover, if $\xi \approx 1$ guessing that the algorithm requires a higher exploitation is always correct since this condition means that the best individual has much better performance than the average. In this case it is desirable to exploit the available genotype by reducing the population size and the proportion of individuals undergoing mutation. On the contrary, the condition $\xi \approx 0$ always means that the population is made up of individuals having very similar performance. In this case the crossover would likely not let a better individual be generated and thus a higher exploration is required using mutation, a large population size and local searchers.

Finally, it is important to remark that a wrong guess on the convergence does not necessarily imply failure of the algorithm. If for example the population is entirely contained in a suboptimal plateau of the fitness landscape, the AEA with IMLSs tries to use the maximum of the explorative resources for escaping this undesired condition and hopefully detect solutions having higher performance. More specifically, a large population size gives the algorithm more available genotypes, an aggressive mutation gives more chances to find a better solution outside the plateau, the *Increasing Diversity* IMLS attempts to explore new genotypes and thus new areas of the decision space.

Figure 2 shows the pseudocode of the AEA with IMLSs.

5 Parameter setting

As highlighted in [30], to implement an adaptive system for parameters control does not necessarily mean that the algorithms contain fewer parameters compared to a standard evolutionary algorithm (as in the case of [63]). On the other hand, the algorithmic performance of an adaptive algorithm is not so sensitive to its parameters [30] unlike the traditional evolutionary algorithms whose success is heavily influenced by a proper parameter setting (e.g. population size, mutation probability etc.). In our case two parameters, that is $p_m^{\max}$ and pr^{ID}, have been tuned by performing test simulations. The parameter under study has been changed while the other parameters have been left unchanged.

For $p_m^{\max}$ in (10), the AEA with IMLSs has been run for 10000 fitness evaluations for the values 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. During these runs pr^{ID} has been kept constant and equal to 0.2. The AEA with IMLSs has been run 5 times for each of the previous values. For these six values of $p_m^{\max}$, the best fitness values have been saved at the end of each

generation. The Average Best Fitness is defined here as the average value over the 5 available simulations for the best fitness values at each generation. Figure 3 shows the comparison of the algorithmic performance for the several values of $p_m^{\max}$ under examination. Since the population size is variable, the algorithmic performance is expressed in terms of fitness evaluations. The results show that, even though the trends are rather similar among each other, the best performance has been obtained for $p_m^{\max} = 0.4$ and this value has therefore been chosen. According to our interpretation, the values $0.1, 0.2, 0.3$ lead to better results in the first generations because the algorithm is more exploitative but this effect is gradually reduced in the following generations when a higher explorative pressure is required. As shown, after 10000 fitness evaluations the worst results are for $p_m^{\max} = 0.1$. In this case, it is quite evident that the algorithm could prematurely converge. With the values $0.5, 0.6$, the improvements are slower than in the case with $p_m^{\max} = 0.4$. This is probably due to an explorative mutation which spoils the genotype of some good candidate solutions.

A similar simulation test has been run in order to tune pr^{ID} ($p_m^{\max}$ has been kept constant and equal to 0.4). The algorithmic performance has been studied for pr^{ID} equal to $0.1, 0.2, 0.3$ as shown in Fig. 4. Also in this case, for each value of pr^{ID}, 5 simulations for 10000 fitness evaluations have been performed. The results show that the algorithm with $pr^{\mathrm{ID}} = 0.1$ is better at the beginning of the optimization process but the choice $pr^{\mathrm{ID}} = 0.2$ seems to be much better after the first generations. Moreover the choice $pr^{\mathrm{ID}} = 0.2$ led to better results than $pr^{\mathrm{ID}} = 0.3$ for all of the simulation. It is important to remark that $pr^{\mathrm{ID}} = 0.3$ means that at most 30% of the population undergoes the *Increasing Diversity* IMLS. This operator is computationally rather cheap but still needs to perform one fitness evaluation each time it is activated.

The results show that the value 0.3 is too high. This is probably because the *Increasing Diversity* IMLS fails too many times and does not produce a sufficient pay-off for the computational resources spent. The value 0.1 leads to an algorithm which is not explorative enough and after a first stage which exploits the initial population diversity it slows down. The value 0.2 for pr^{ID} has therefore been chosen.

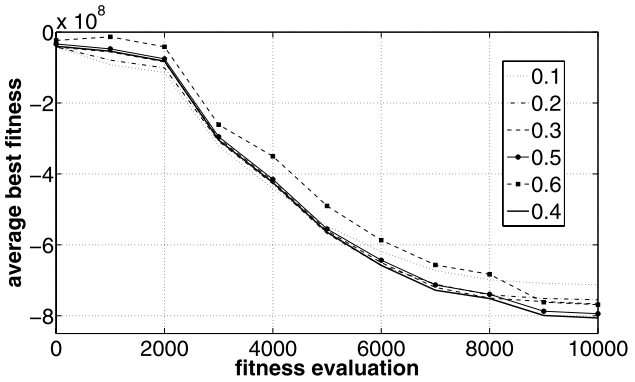

Fig. 3 Algorithmic performance for several values of $p_m^{\max}$

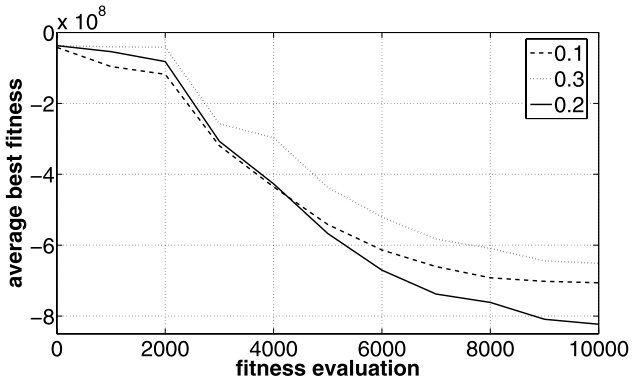

Fig. 4 Algorithmic performance for several values of pr^{ID}

6 Numerical results

Here STI therapies are designed and compared for a 750 day time horizon after an acute HIV infection using the heuristic method described in Sect. 3, the AEA with IMLSs and a standard GA. The algorithmic parameters related to the GA and the AEA with IMLSs are shown in Table 2.

The standard GA makes use of the same parent selection, crossover technique and mutation technique described for the AEA with IMLSs in Sect. 4. Moreover the GA has fixed population size and mutation probability. Each of these values has been set as the mean value of the corresponding range of variability for the AEA with IMLSs (see Table 2). The standard GA does not make use of any IMLS.

For both the standard GA and the AEA with IMLSs 50 experiments have been performed. The heuristic method has been performed only once since it is a deterministic method and the fitness is not noisy. Figure 5 shows the optimized

Table 2 GA and AEA with IMLSs parameter setting

Parameter	GA	AEA with IMLSs
Size of initial population pseudo-randomly generated	1000	1000
Population size for subsequent generations	450	dynamic between 100 and 800
Mutation probability	0.2	dynamic between 0 and 0.4
Fitness evaluations	100 000	100 000

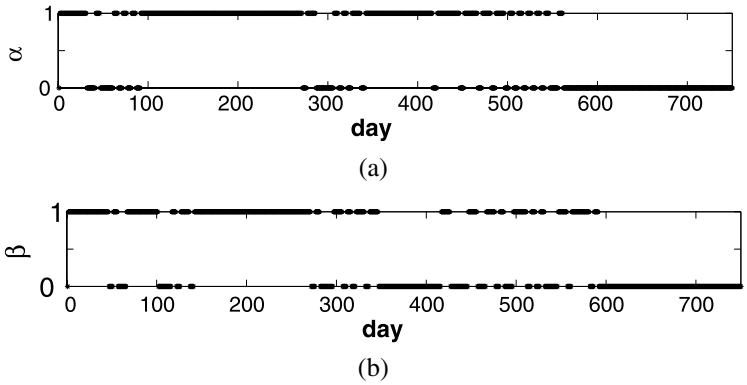

Fig. 5 Optimal STI multidrug medication for HIV obtained by the heuristic method. (**a**) Reverse transcriptase inhibitor (RTI) medication. (**b**) Protease inhibitor (PI) medication

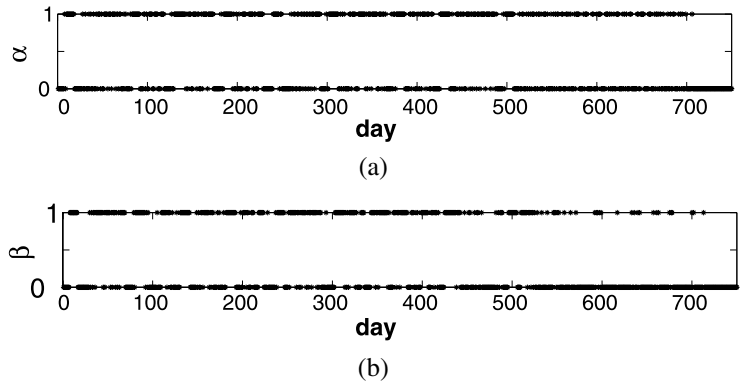

Fig. 6 Optimal STI multidrug medication for HIV obtained by the GA. (**a**) Reverse transcriptase inhibitor (RTI) medication. (**b**) Protease inhibitor (PI) medication

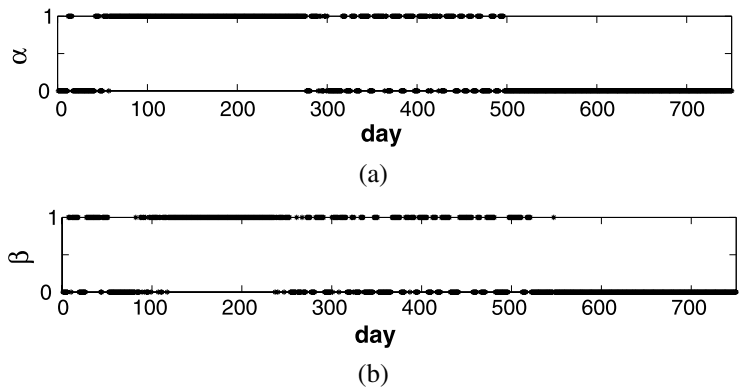

Fig. 7 Optimal STI multidrug medication for HIV obtained by the AEA with IMLSs. (**a**) Reverse transcriptase inhibitor (RTI) medication. (**b**) Protease inhibitor (PI) medication.

medications obtained using the heuristic method. Figures 6 and 7 show the best optimization results over the 50 experiments for the GA, and the AEA with IMLSs, respectively. The upper plots show the vectors α defining the Reverse Transcriptase Inhibitor (RTI) medication and the lower plots show the vectors β the Protease Inhibitor (PI) medication. The therapies obtained using the GA and the AEA with IMLSs start after a few days. The heuristic proposes a therapy starting immediately after the HIV infection which is probably a suboptimal solution caused by the limitation to consider only 30 day subperiods. The medications given by the GA do not have clear medication and rest periods, but the two other medications have some more apparent on and off periods. Another measure of the quality of medications apart from the objective function J is how early medications can be discontinued. For all therapies the last medications are PIs. The medications given by the heuristic method and the AEA with IMLSs are discontinued after 590 and 546 days, respectively, while the medication suggested by the GA requires more than 700 days. Thus, in this sense the AEA with IMLSs gives more effective medication than the heuristic and the solution given by the GA is far from optimal.

Table 3 shows the values of the objective functions J_i and the weighted sum objective function J for the heuristic method and the best results over the 50 experiments for the standard GA and the AEA with IMLSs. The results in Table 3 show that the value of the objective function J is much lower in the case of the AEA with IMLSs. According to the

Table 3 Values of the multi-objective function and its components

	J_1 [10^7]	J_2 [10^8]	J_3	J_4	J [10^{11}]
Heuristic	1.2924	0.7057	204.8215	32.2203	−0.7057
GA	2.5457	0.4430	159.4852	25.1438	−0.4429
AEA with IMLSs	0.9307	1.0135	170.1952	28.3505	−1.0134

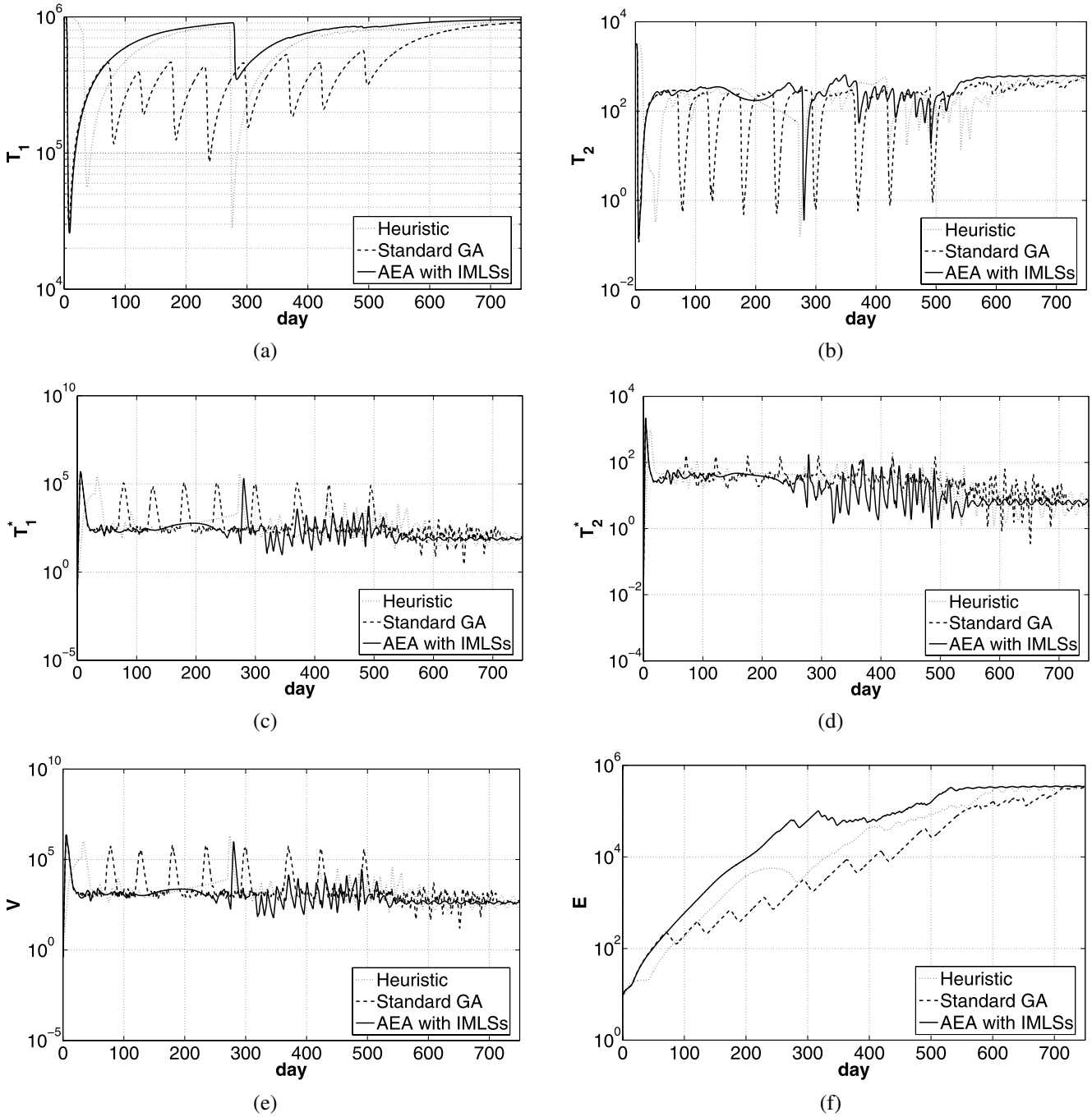

Fig. 8 STI control solutions obtained by the GA, the heuristic method and the AEA. (**a**) Uninfected CD4+ T-cells. (**b**) Uninfected target cells of second kind. (**c**) Infected CD4+ T-cells. (**d**) Infected target cells of second kind. (**e**) Viral load. (**f**) Immune response

weighted sum fitness J, the AEA with IMLSs therefore has clearly outperformed both the heuristic and the GA. Moreover, considering that J_1, J_3 and J_4 have to be minimized and, on the contrary, J_2 has to be maximized as the weight of this objective function is negative, the solution given by the AEA with IMLSs strictly dominates the solution given by the heuristic method. The comparison between the AEA with IMLSs and the GA shows that, although the value of J for the AEA with IMLSs is more than twice as small, this solution does not dominate the one given by the GA. In fact, the solution given by the GA offers slightly better performance in terms of the quantity of medications (J_3 and J_4) that the patient should take but much worse performance than the AEA with IMLSs with respect to viruses and immune effectors. For the considered STI therapies it is much more important that the therapy stimulates a strong immune response quickly than the minimal use of medications. In this sense the fact that the GA proposes a smaller quantity of medications than the AEA with IMLSs is of minor interest.

Figure 8 shows the behavior of the state variables in the model (1) for the optimized STI medications. The therapies proposed by the heuristic method and the AEA with IMLSs causes clear dips in the amount of uninfected T-cells during the first 50 days and then again around the 280th day and the corresponding abrupt increases in the infected T-cells and viruses. For the therapy designed by the GA the behavior of the infected and uninfected T-cells as well as viruses is much more oscillatory due to short breaks in the medications. The last plot in Fig. 8 shows clearly the therapy given by the AEA with IMLSs stimulates a stronger immune response than the two other optimized therapies. In particular, Fig. 8(f) shows that in the AEA with IMLSs designed therapy the amount of immune effectors E reach the neighborhood of the steady state value 353108 cells/ml after about 550 days while for the other therapies designed by the heuristic method and the GA this require around 600 and 700 days, respectively. Thus, the steady state value of E is approximately reached when the medications are discontinued. The therapy proposed by the AEA with IMLSs is thus very beneficial for the patient since it leads to a quick immune response (the healthy state is reached within 550 days) and it allows interruption of the medical treatment in a rather short time (546 days) which reduces possible side effects. In this sense, the medical protocol proposed in this paper is a valuable improvement over that in [20].

In order to analyze the algorithmic behavior of the AEA with IMLSs, the diagram of the coefficient ξ vs fitness evaluation is shown in Fig. 9 in the best case over the 50 experiments. In Fig. 9 also shown is each threshold value for activating the three IMLSs. As expected the trend of ξ is oscillatory and it is shown how the algorithm converges to the final solution when ξ approaches zero. It is important to notice that from the given trend of ξ it is possible to determine

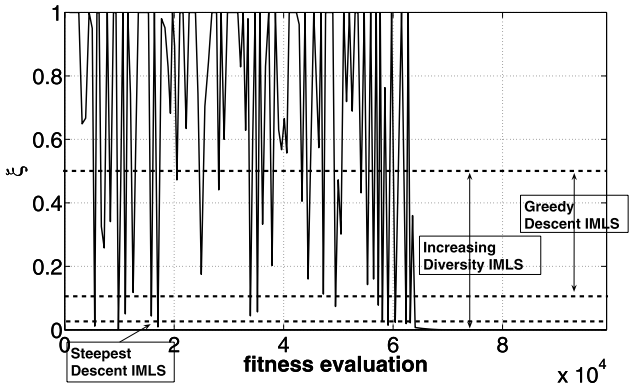

Fig. 9 Behavior of ξ in the most successful experiment

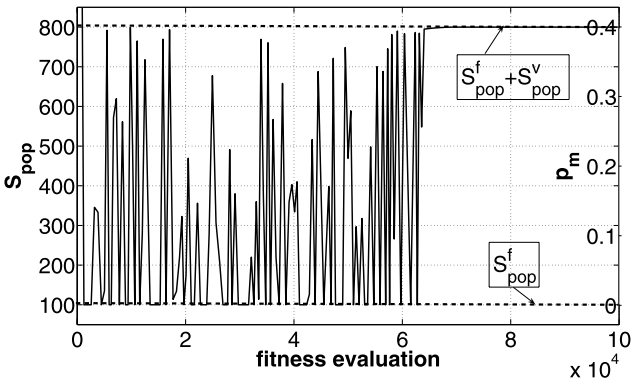

Fig. 10 Behavior of S_{pop} and p_m in the most successful experiment

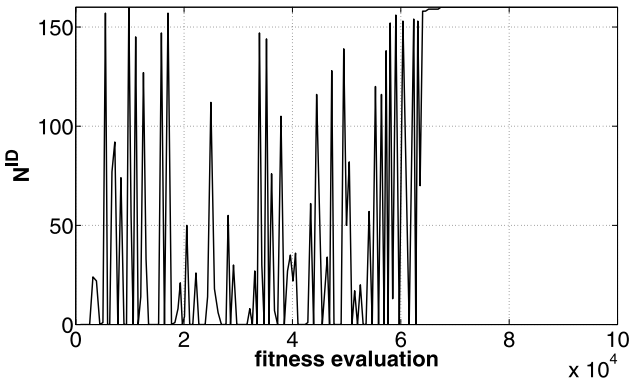

Fig. 11 Behavior of N^{ID} in the most successful experiment

the population size, the mutation probability and the number of individuals undergoing *Increasing Diversity* IMLS. Figure 10 shows the behavior of the population size S_{pop} and the mutation probability p_m for the same generations shown in Fig. 9. Since the trend of S_{pop} and p_m are proportional, they are represented in Fig. 10 by a unique trace and two differently scaled y-axes.

Figure 11 shows the trend, for the same most successful experiment over the 50 carried out, of the number N^{ID} of individuals undergoing *Increasing Diversity* IMLS.

Table 4 Statistical comparison of the performance of the GA and the AEA with IMLSs

Method	$J_{\max}$	$J_{\min}$	$\bar{J}$	σ	$\sigma/\bar{J}$
GA	-4.1454×10^{10}	-4.4290×10^{10}	-4.2806×10^{10}	8.7098×10^{8}	0.0203
AEA with IMLSs	-9.9252×10^{10}	-1.0134×10^{11}	-1.0006×10^{11}	4.5854×10^{8}	0.0046

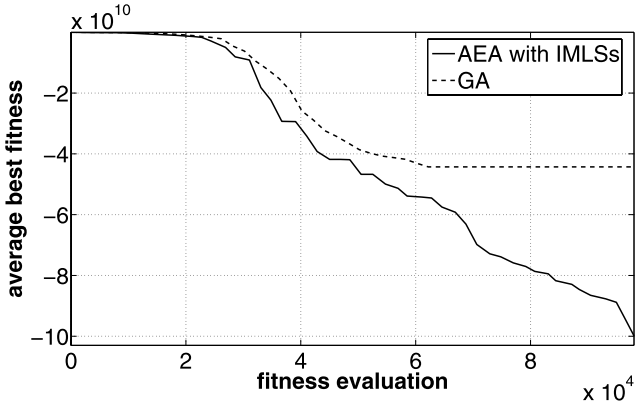

Fig. 12 Comparison of the algorithmic performance between the AEA with IMLSs and the standard GA

As mentioned above, 50 experiments have been performed for both the standard GA and the AEA with IMLSs. Each of these experiments have been stopped after 100 000 fitness evaluations. For each experiment and for both the GA and the AEA with IMLSs the best fitness values at the end of each generation and at the end of the optimization process have been saved. The average best fitness values (see Sect. 5) have been calculated over the 50 available data for each generation. Figure 12 shows and compares the algorithmic performance of the GA and the AEA with IMLSs. Figure 12 shows that for all 50 experiments the standard GA prematurely converged after about 60 000 fitness evaluations to a suboptimal solution. On the contrary, the AEA with IMLSs continued the optimization process without being trapped in a basin of attraction. In order to make a comparison between the GA and the AEA with IMLSs, the best fitness value after 100 000 has been averaged over the 50 experiments thus obtaining a final average optimal $\bar{J}$. The values of standard deviation, absolute (σ) and relative to $\bar{J}$ ($\sigma/\bar{J}$) have also been given in Table 4. The final fitness values have also been reported for the values of the maximum (worst) $J_{\max}$ and minimum (best) $J_{\min}$, over these 50 experiments.

The results show that, according the average best fitness value $\bar{J}$, the AEA with IMLSs outperforms the standard GA. Moreover the value of standard deviation is smaller in the case of the AEA with IMLSs and therefore the proposed algorithm is probably more robust than the standard GA.

According to our interpretation, the smaller value of σ is mainly due to the use of the local searchers and in particular to the *Steepest Descent* IMLS. As highlighted above, the evolutionary algorithms are efficient in finding solutions near the optimum but they are not very efficient in finalizing the optimization process [30]. In our case, the *Steepest Descent* IMLS deterministically finalizes the optimization process (with steepest descent pivot rule) and thus ensures that it will return to the same optimal value for different solutions falling in the same basin of attraction. Therefore, if over different simulations this local searcher processed solutions belonging to the same basin of attraction, it converged every time to solutions that possess the same fitness value. This would explain the small standard deviation in Table 4.

It is also interesting to note that the GA tends to prematurely converge to a set of solutions, similar among each other and having a fitness value $J \approx 4.3 \times 10^{10}$. In fact, as can be seen from Table 4, the value of σ is also relatively small for the GA. This effect is probably due to the presence of a strong basin of attraction in the fitness landscape which attracts the population of the GA to converge on it.

The values of $J_{\min}$ and $J_{\max}$ have been reported in order to compare the performance of the two algorithms in terms of tolerance intervals. Unfortunately the data generated by an optimization method cannot be approximated by a normal (Gaussian) distribution and therefore a general analysis must be carried out. Well known tolerance interval estimates, for a certain confidence level, are the ranges which should contain a certain percentage of random data. If this range is bounded from both the sides we talk about two-sided tolerance interval. If this range is bounded just from one side then we talk about one-sided tolerance interval. In other words, for N random values it is possible to predict, with an established confidence level, the proportion of the M forthcoming random values which will fall within the interval bounded by the smallest and the largest random values among the N values previously sampled (two-sided tolerance interval). Analogously, it is possible to predict, with an established confidence level, the proportion of the M forthcoming random values which will be smaller than the largest value among the N previously sampled or larger than the smallest value among the N previously sampled (one-sided tolerance interval).

In our application, the range is given ($[J_{\min}, J_{\max}]$), the confidence level is established and the corresponding proportion of data can be calculated. Following the procedure given in [66, 67] for a two-sided tolerance interval the proportion γ of a set of data which falls within a given interval with a given confidence level δ has been determined by:

$$\gamma \approx 1 - \frac{a}{n}, \tag{13}$$

where n is the number of available samples and a is the positive root of the equation

$$(1+a)-(1-\delta)\cdot e^{a}=0. \tag{14}$$

In the case of one-sided tolerance interval

$$\gamma \approx 1-\frac{b}{n}, \tag{15}$$

where

$$b=-\ln(1-\delta). \tag{16}$$

In our case, taking into account that $n=50$, it is possible to state that with a confidence level $\delta=0.95$ a proportion $\gamma=0.9086$ of data falls within the interval $[J_{\min}, J_{\max}]$. This value of γ is obviously valid for both the GA and the AEA with IMLSs. In other words, since most of the data would fall within the tolerance intervals and since the $J_{\min}$ for the GA is much larger than the $J_{\max}$ for the AEA with IMLSs, it is possible to state that it is highly improbable that the GA could outperform the AEA with IMLSs even once. Moreover, considering the one-sided tolerance interval, with a confidence level $\delta=0.95$ a proportion $\gamma=0.9418$ of data returned by the AEA with IMLSs will not be larger than $J_{\max}$. Since $J_{\max}$ is much smaller than the optimal value given by the heuristic (see Table 3), it is also quite improbable that the AEA with IMLSs gives worse results than the heuristic.

7 Conclusion

This paper presented an Adaptive Evolutionary Algorithm (AEA) with three Intelligent Mutation Local Searchers (IMLSs) for designing optimal Structured Treatment Interruption (STI) multidrug therapies for HIV. The AEA with IMLSs minimizes a weighted sum objective function based on a dynamic model of the HIV pathogenesis in a human body. The model includes two different types of target cells, infected cells, viruses and an immune response. Furthermore, it describes the effects of the Reverse Transcriptase Inhibitor (RTI) and Protease Inhibitor (PI) therapies which are currently the most commonly used anti-HIV medications. The AEA with IMLSs is able to construct a medical protocol for an HIV patient that stimulates a patient's immune response so much that medical therapy can be discontinued after one and a half years.

The results given by the AEA with IMLSs and a standard GA have been compared with the results presented in the literature which were obtained using a heuristic method. The AEA with IMLSs outperformed both other approaches in terms of the optimality of the solution and therefore led to a more efficient medication schedule. In particular, the medical treatment designed by the proposed algorithm is very promising since it leads to a strong immune response and a "healthy" medical condition after one and half years. Moreover the proposed protocol allows the patient to discontinue the therapy about one month and a half earlier than with the protocol given by the heuristic method and about six months earlier than with the protocol designed by the standard GA. This earlier termination of medications reduces the likelihood of side effects caused by drugs. In addition, the overall amount of medicines in the protocol designed by the AEA with IMLSs is smaller than in the one obtained by the heuristic method in the literature. This leads to a reduced cost of treatment which is particularly helpful in developing countries.

From the optimization point of view the results are also good. The AEA with IMLSs does not impose any restrictions for the structure of the therapies like the heuristic method does based on specific properties of the problem under study. The problems related to stagnation and convergence to a suboptimal solution are solved using the adaptation and the IMLSs. As the numerical results show, due to the size of the decision space and to the behavior of the fitness function, a standard GA usually converges to a suboptimal solution. For this class of problems the results show that the AEA with IMLSs has the following properties. The implemented adaptation can dynamically balance the exploration and exploitation following the needs of the evolutionary process and, therefore, preventing stagnation and premature convergence. The *Increasing Diversity* IMLS and the related adaptive rule can increase the population diversity when it is required, "refreshing" the genotype of the population. The *Greedy Descent* IMLS assists in the optimization to finding new promising search directions using a Lamarckian logic. The *Steepest Descent* IMLS helps the algorithm to find the optimal solution when population is rather near to it which is usually a difficult task for evolutionary algorithms.

Acknowledgements This research was supported by: Joint DMS/NIGMS Initiative to Support Research in the Area of Mathematical Biology; contract/grant number: 1R01GM67299-01. The Academy of Finland grant #204741. The authors wish also to acknowledge Prof. Luigi Salvatore and the laboratory of Electrical Machines, Politecnico di Bari for the technical support in executing the numerical experiments. The authors like to thank the reviewers for several suggestions improving the presentation.

References

1. Adams BM, Banks HT, Davidian M, Kwon H-D, Tran HT, Wynne SN, Rosenberg ES (2005) HIV dynamics: modeling, data analysis, and optimal treatment protocols. J Comput Appl Math 184(1):10–49
2. Bonhoeffer S, Rembiszewski M, Ortiz GM, Nixon DF (2000) Risks and benefits of structured antiretroviral drug therapy interruptions in HIV-1 infection. AIDS 14(15):2313–2322
3. Perelson A, Kirschner D, de Boer R (1993) The dynamics of HIV infection of CD4+ T cells. Math Biosci 114:81–125

4. Perelson AS, Nelson PW (1999) Mathematical analysis of HIV-1 dynamics in vivo. SIAM Rev 41(1):3–44
5. Wodarz D, Nowak MA (1999) Specific therapy regimes could lead to long-term immunological control of HIV. Proc Natl Acad Sci 96(25):14464–14469
6. Wodarz D (2001) Helper-dependent vs. helper-independent CTL responses in HIV infection: implications for drug therapy and resistance. J Theor Biol 213(3):447–459
7. Wodarz D, Nowak MA (2002) Mathematical models of HIV pathogenesis and treatment. BioEssays 24(12):1178–1187
8. Tan W-Y, Wu H (2005) Deterministic and stochastic models of AIDS epidemics and HIV infections with intervention. World Scientific, Hackensack
9. Alvarez-Ramirez J, Meraz M, Velasco-Hernandez JX (2000) Feedback control of the chemotherapy of HIV. Int J Bifur Chaos 10(9):2207–2219
10. Butler S, Kirschner D, Lenhart S (1997) Optimal control of chemotherapy affecting the infectivity of HIV. In: Arino O, Axelrod D, Kimmel M, Langlais M (eds) Advances in mathematical population dynamics: molecules, cells, man. World Scientific, Singapore, pp 104–120
11. Culshaw R, Ruan S, Spiteri RJ (2004) Optimal HIV treatment by maximising immune response. J Math Biol 48(5):545–562
12. Fister KR, Lenhart S, McNally JS (1998) Optimizing chemotherapy in an HIV model. Electron J Differ Equ 1998(32):1–12
13. Shim H, Han SJ, Chung CC, Nam S, Seo JH (2003) Optimal scheduling of drug treatment for HIV infection: continuous dose control and receding horizon control. Int J Control Autom Syst 1(3):401–407
14. Kirschner D, Lenhart S, Serbin S (1997) Optimal control of the chemotherapy of HIV. J Math Biol 35(7):775–792
15. Ledzewicz U, Schättler H (2002) On optimal controls for a general mathematical model for chemotherapy of HIV. In: Proceedings of the 2002 American control conference, vol 5, pp 3454–3459
16. Zurakowski R, Messina MJ, Tuna SE, Teel AR (2004) HIV treatment scheduling via robust nonlinear model predictive control. In: Proceedings of the 5th Asian control conference
17. Zurakowski R, Teel AR (2003) Enhancing immune response to HIV infection using MPC-based treatment scheduling. In: Proceedings of the 2003 American control conference, vol 2, pp 1182–1187
18. Zurakowski R, Teel AR, Wodarz D (2004) Utilizing alternate target cells in treating HIV infection through scheduled treatment interruptions. In: Proceedings of the 2004 American control conference, vol 1, pp 946–951
19. Zurakowski R, Teel AR (2006) A model predictive control based scheduling method for HIV therapy. J Theor Biol 238(2):368–382
20. Adams BM, Banks HT, Kwon H-D, Tran HT (2004) Dynamic multidrug therapies for HIV: optimal and STI control approaches. Math Biosci Eng 1(2):223–241
21. Banks HT, Kwon H-D, Toivanen J, Tran HT (2006) An state dependent Riccati equation based estimator approach for HIV feedback control. Optim Control Appl Methods 27(2):93–121
22. Caetano MAL, Yoneyama T (2002) Short and long period optimization of drug doses in the treatment of AIDS. An Acad Bras Ci 74(3):379–392
23. Jeffrey AM, Xia X, Craig IK (2003) When to initiate HIV therapy: a control theoretic approach. IEEE Trans Biomed Eng 50(11):1213–1220
24. Kutch JJ, Gurfil P (2002) Optimal control of HIV infection with a continuously-mutating viral population. In: Proceedings of the 2002 American control conference, vol 5, pp 4033–4038
25. Wein LM, Zenios SA, Nowak MA (1997) Dynamic multidrug therapies for HIV: a control theoretic approach. J Theor Biol 185(1):15–29
26. Bajaria SH, Webb G, Kirschner DE (2004) Predicting differential responses to structured treatment interruptions during HAART. Bull Math Biol 66(5):1093–1118
27. Brandt ME, Chen G (2001) Feedback control of a biodynamical model of HIV-1. IEEE Trans Biomed Eng 48(7):754–759
28. Joshi HR (2002) Optimal control of an HIV immunology model. Optim Control Appl Methods 23(4):199–213
29. Lisziewicz J, Rosenberg E, Lieberman J, Jessen H, Lopalco L, Siliciano R, Walker B, Lori F (1999) Control of HIV despite the discontinuation of antiretroviral therapy. New Engl J Med 340(21):1683–1684
30. Eiben AE, Smith JE (2003) Introduction to evolutionary computation. Springer, Berlin
31. Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Trans Evol Comput 3(2):124–141
32. Fonlupt C, Preux P, Robilliard D (1997) Preventing premature convergence via cooperating genetic algorithms. In: Ošmera P (ed) Proceedings of the 3rd international Mendel conference on genetic algorithms, optimization problems, fuzzy logic, neural networks, rough sets (MENDEL'97), pp 50–55
33. Ishibuchi H, Yoshida T, Murata T (2003) Balance between genetic search and local search in memetic algorithms for multiobjective permutation flow shop scheduling. IEEE Trans Evol Comput 7(2):204–223
34. Caponio A, Cascella GL, Neri F, Salvatore N, Sumner M (2007) A fast adaptive memetic algorithm for on-line and off-line control design of pmsm drives. IEEE Trans Syst Man Cybern Part B, special issue on memetic algorithms 37(1):28–41
35. Bäck T (1992) The interaction rate of mutation rate, selection, and self-adaptation within a genetic algorithm. In: Proceedings of parallel problem solving from nature (PPSN-II). Elsevier, Amsterdam, pp 85–94
36. Davis L (1989) Adapting operator probabilities in genetic algorithms. In: Schaffer J (ed) Proceedings of the 3rd international conference on genetic algorithms. Kaufmann, Los Altos, pp 61–69
37. Smith JE (2001) Modelling GAs with self-adaptive mutation rates. In: Proceedings of GECCO. Kaufmann, Los Altos, pp 599–606
38. Eiben AE, Hinterding R, Michaelwicz Z (2000) Parameter control. In: Baeck T, Fogel DB, Michaelwicz Z (eds) Evolutionary computation 2, advanced algorithms and operators. Institute of Physics, pp 170–187
39. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
40. Magoulas GD, Plagianakos VP, Vrahatis MN (2004) Neural network-based colonoscopic diagnosis using on-line learning and differential evolution. Appl Soft Comput 4(4):369–379
41. Hart WE, Krasnogor N, Smith JE (2004) Memetic evolutionary algorithms. In: Hart WE, Krasnogor N, Smith JE (eds) Recent advances in memetic algorithms. Springer, Berlin, pp 3–27
42. Salcedo-Sanz S, Bousono-Calzon C (2005) A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications. Appl Intell 22:3
43. Ong YS, Lim MH, Zhu N, Wong KW (2006) Classification of adaptive memetic algorithms: a comparative study. IEEE Trans Syst Man Cybern Part B 36(1):141–152
44. Krasnogor N, Smith J (2000) A memetic algorithm with self-adaptive local search: TSP as a case study. In: Whitley D, Goldberg D, Cantu-Paz E, Spector L, Parmee I, Beyer H-G (eds) Proceedings of the genetic and evolutionary computation conference (GECCO-2000), Las Vegas, NE, USA. Kaufmann, Los Altos, pp 987–994
45. Eiben AE, Smith JE (2003) Hybrid evolutionary algorithms. In: Introduction to evolutionary computing, hybridisation with other

techniques: memetic algorithms. Slides of the Lecture Notes, Chapter 10
46. Houck CR, Joines JA, Kay MG, Wilson JR (1997) Empirical investigation of the benefits of partial lamarckianism. Evol Comput 5(1):31–60
47. Schwefel HP (1995) System evolution and optimum seeking. Wiley, New York
48. Callaway DS, Perelson AS (2002) HIV-1 infection and low steady state viral loads. Bull Math Biol 64(1):29–64
49. Miettinen K (1999) Nonlinear multiobjective optimization. International series in operations research & management science, vol 12. Kluwer Academic, Boston
50. Hairer E, Wanner G (1996) Solving ordinary differential equations. II. Stiff and differential-algebraic problems, 2nd edn. Springer series in computational mathematics, vol 14. Springer, Berlin
51. Smith JE, Fogarthy T (1997) Operator and parameter adaptation in genetic algorithms. Soft Comput 1(2):81–87
52. Neri F, Cascella GL, Salvatore N, Kononova AV, Acciani G (2006) Prudent-daring vs tolerant survivor selection schemes in control design of electric drives. In: Applications of evolutionary computing. Lecture notes in computer science, vol 3907. Springer, Berlin, pp 805–809
53. Whitley D (1989) The genitor algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In: Proceedings of the third international conference on genetic algorithms, pp 116–121
54. Baker JE (1985) Adaptive selection methods for genetic algorithms. In: Proceedings of the first international conference on genetic algorithms and their applications, pp 101–111
55. Baker JE (1987) Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the second international conference on genetic algorithms, pp 14–21
56. Spears WM (1998) The role of the mutation and recombination in evolutionary algorithms. PhD thesis, VA George Mason University, Fairfax
57. Goldberg DE (1989) Genetic algorithms for search, optimization and machine learning. Addison–Wesley, Reading
58. Deb K, Agrawal S (1999) Understanding interactions among genetic algorithm parameters. In: Foundations of the genetic algorithms (FOGA-5), pp 265–286
59. Krasnogor N (2004) Toward robust memetic algorithms. In: Hart WE, Krasnogor N, Smith JE (eds) Recent advances in memetic algorithms, Springer, Berlin, pp 185–207
60. Krasnogor N (1999) Co-evolution of genes and memes in memetic algorithms. In: Wu A (ed) Proceeding of the 1999 genetic and evolutionary computation conference
61. Krasnogor N, Blackburne B, Burke E, Hirst J (2002) Multimeme algorithms for proteine structure prediction. In: Proceeding of parallel problem solving in nature VII. Lecture notes in computer science. Springer, Berlin
62. Ong YS, Keane AJ (2004) Meta-Lamarkian learning in memetic algorithms. IEEE Trans Evol Comput 8(2):99–110
63. Arabas J, Michalewicz Z, Mulawka J (1994) Gavaps—a genetic algorithm with varying population size. In: Proceedings of the first IEEE conference on evolutionary computation, 1994 and IEEE world congress on computational intelligence, vol 1, June 1994, pp 73–78
64. Eshelman LJ (1991) The chc adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Foundations of genetic algorithms 1 (FOGA-1). Kaufmann, Los Altos, pp 263–283
65. Gehlhaar DK, Fogel DB (1996) Tuning evolutionary programming for conformationally flexible molecular docking. In: Fogel TBLJ, Angeline PJ (eds) Proceedings of the 5th annual conference on evolutionary programming. MIT, Cambridge, pp 419–429
66. Owen DB (1962) Handbook of statistical tables. Addison-Wesley, Reading
67. Wilks SS (1942) Statistical prediction with special reference to the problem of tolerance limit. Ann Math Stat 13(4):400–409