An Adaptive Multimeme Algorithm for Designing HIV Multidrug Therapies

Ferrante Neri, Jari Toivanen, Giuseppe Leonardo Cascella, and Yew-Soon Ong

Abstract—This paper proposes a period representation for modeling the multidrug HIV therapies and an Adaptive Multimeme Algorithm (AMmA) for designing the optimal therapy. The period representation offers benefits in terms of flexibility and reduction in dimensionality compared to the binary representation. The AMmA is a memetic algorithm which employs a list of three local searchers adaptively activated by an evolutionary framework. These local searchers, having different features according to the exploration logic and the pivot rule, have the role of exploring the decision space from different and complementary perspectives and, thus, assisting the standard evolutionary operators in the optimization process. Furthermore, the AMmA makes use of an adaptation which dynamically sets the algorithmic parameters in order to prevent stagnation and premature convergence. The numerical results demonstrate that the application of the proposed algorithm leads to very efficient medication schedules which quickly stimulate a strong immune response to HIV. The earlier termination of the medication schedule leads to lesser unpleasant side effects for the patient due to strong antiretroviral therapy. A numerical comparison shows that the AMmA is more efficient than three popular metaheuristics. Finally, a statistical test based on the calculation of the tolerance interval confirms the superiority of the AMmA compared to the other methods for the problem under study.

Index Terms—HIV therapy design, memetic algorithms, adaptive algorithms, nonlinear integer programming.

1 Introduction

In western countries, the life expectancy of patients infected by Human Immunodeficiency Virus (HIV) has increased to tens of years due to improved medical treatments. Nowadays, the most common medication is a Highly Active AntiRetroviral Therapy (HAART), which is a "cocktail" consisting of three or more drugs. These therapies keep the number of viruses low and they help to maintain a high number of CD4+ T-cells, which are a fundamental part of the human immune system. A person is considered to have the Acquired Immune Deficiency Syndrome (AIDS) once the T-cell count falls below a threshold value. As long as the HAART medication is effective, this can be prevented. These treatments have been very successful, but they still have several disadvantages and shortfalls. Some patients develop resistance to a drug in their HAART cocktail and, due to this, it is necessary to change the composition of medication. The medications often have unpleasant side effects which can sometimes be so severe that the medication has to be stopped or changed. Particularly in developing countries, the high cost of HAART restrains their use. Due to these reasons, research on alternative treatments is active. This paper studies methods for constructing multidrug therapies which stimulate such a strong immune response that HIV is suppressed without the need for further medication after the initial treatment period.

Human immunodeficiency viruses can infect many types of cells, including CD4+ T-cells, which are an essential part of the human immune system. Once the viral code is integrated into the host cell's DNA code, it can produce large quantities of viruses. The two most important types of anti-HIV drugs are currently Reverse Transcriptase Inhibitors (RTIs) and Protease Inhibitors (PIs). Usually, the HAART cocktail consists of two or more RTIs and a PI. The reverse transcriptase inhibitors prevent HIV from infecting cells by blocking the integration of the viral code. Protease inhibitors interfere with the replication of HIV, leading to production of defective viruses which are not infectious, that is, they cannot infect cells. In practice, RTIs cannot completely block viruses from infecting cells and, similarly, some infectious viruses are still produced under a PI medication. The maximum efficacy of a drug depends on many factors. A mutation can lead to a drug-resistant virus strain and, thus, to lower efficacy for some medications.

Mathematical models for the pathogenesis of HIV are usually based on systems of differential equations. The models presented in [1], [2], [3], [4], [5], [6], [7] have been used to design dynamical antiviral drug therapies. An immune response can have a profound effect on the long-term progression of HIV. Thus, the models including an immune response like the ones in [1], [2], [5], [6], [7] are better suited to designing long-term treatments. Currently, the immune response to HIV is not well understood; due to this, several different models have been proposed. In particular, a model including the immune system is needed

- *F. Neri and J. Toivanen are with the Department of Mathematical Information Technology, Agora University of Jyväskylä, FI-40014 Jyväskylä, Finland. E-mail: neferran@cc.jyu.fi, tene@mit.jyu.fi.*
- *G.L. Cascella is with the Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Via E. Orabona, 4-70125 Bari, Italy. E-mail: leocascella@ieee.org.*
- *Y.-S. Ong is with the School of Computer Engineering, Nanyang Technological University, Blk N4, 2b-39, Nanyang Avenue, Singapore 639798. E-mail: asysong@ntu.edu.sg.*

Manuscript received 1 Mar. 2006; revised 15 June 2006; accepted 1 Aug. 2006; published online 12 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0036-0306. Digital Object Identifier no. 10.1109/TCBB.2007.070202.

to design medical treatments that stimulate a strong immune response. In this paper, we use the model for the pathogenesis of HIV which was introduced in [1] and then employed in [8], [9], [10].

A large number of papers consider control techniques for planning HIV therapies. The papers [1], [11], [12], [13], [14], [15] consider only RTI medication, while the papers [16], [17] consider only PIs. In [18], [19], [20], [21], all drugs in a HAART medication are lumped as one control variable in the model. The papers [8], [9], [22], [23], [24], [10], [25] design dynamic multidrug therapies employing RTIs and PIs. In these therapies, the dosages of both medications can change independently of each other. This paper also studies this type of dynamic multidrug therapy.

In the considered control approaches, the amount of medication can be either continuous or binary. The latter type is called Structured Treatment Interruption (STI) and has been extensively studied in the medical literature, see [26] and references therein. The main argument for using 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], [11], [27], [12], [22], [13], [14], [23], [28], [16], [24], [17], [7], [25]. Structured treatment interruption schedules have been considered In [1], [8], [10], [19], [20], [21]. Here, STI medications combining RTIs and PIs are studied.

This paper designs STI antiviral multidrug therapies for HIV using an open loop tracking problem which is similar to the one in [9]. Continuously varying medications were constructed in [9] using a state-dependent Riccati equation-based feedback control. The aim here is to construct STI medications which are easier to administer and that have lower risks of developing drug resistance than continuously varying medications. The same mathematical model, but with a different fitness function, was used to optimize STI medications in [8], [10]. These papers used a binary presentation for the medication, while a period presentation is used here. The main advantages of the period presentation are a lower cardinality of optimization problems and more flexibility on imposing constraints for medications. In particular, the evolutionary optimization approach in [10] is closely related to the approach considered here, but, due to different representation, the optimization techniques are not the same.

In [29], a related problem of designing treatment interruption schedules for cancer chemotherapies using evolutionary algorithms is considered. The parameterization for the schedule in [29] is defined by the durations of the treatment and rest periods and then the number of the cycles is constrained to nine. This leads to an optimization problem with 18 real-valued variables. This paper proposes a similar parametrization for STI therapies for HIV. Typical optimal STIs have many short cycles and, thus, the optimization problem has a high dimensionality when using such a parameterization. The considered optimization of multidrug HIV therapies leads to a nonlinear integer programming problem which has a high dimensionality. A further challenge is due to a small range of values for the objective function for a dominant part of the decision space. This leads to a fairly flat fitness landscape [30], [31], [32]. Under these conditions, the problem is hard to solve. A classical optimization method usually gets trapped into a neighborhood of a local minimum with a high objective function value. A simple Evolutionary Algorithm often either stagnates due to the flatness of the landscape or converges prematurely to suboptimal solutions.

In order to overcome these undesired behaviors and to find the optimal HIV multidrug therapy, a Computational Intelligence Approach is proposed here. More specifically, this paper describes an Adaptive Multimeme Algorithm (AMmA) [33], [34], [35] which combines the features of an evolutionary framework and three local searchers which are intelligently activated by means of an adaptive rule. These three local searchers have different features and, thus, bias the optimization process in different ways. The first local searcher is highly explorative since it executes a random search working on all the variables of the candidate solutions at the same time and greedily accepts enhanced solutions. The second local searcher is highly exploitative since it works on one variable at once and accepts a new solution only after having exhaustively explored the neighborhood of the starting solution. The third local searcher has intermediate exporative/exploitative features. It works on two variables and makes use of the logic of the Simulated Annealing (it could therefore also accept solutions with worse performance) in order to jump out from a suboptimal basin of attraction and, thus, detect new promising search directions while trying to exploit the most of the genotype of the starting solution. Moreover, the AMmA employs a dynamic parameter setting for both evolutionary framework and local searchers, with the aim of following the necessity of the evolutionary process preventing the stagnation and the premature convergence and, at the same time, avoiding unnecessary fitness evaluations.

2 HIV Medication Optimization

2.1 HIV Model

We model the pathogenesis of HIV using a system of Ordinary Differential Equations (ODEs) introduced in [1] and employed in [8], [9], [10]. This model has two classes of target cells which can be infected by HIV. The first class is the CD4+ T-cells denoted by T_1 in the model. These cells are a fundamental component of the human immune response system. The second class of cells are denoted by T_2 and their biological cell type is not specified in [1]. The viral load in the model is denoted by V. The model includes an immune response measure denoted by E, which is also not specified biologically. The control variables in the model are the Reverse Transcriptase Inhibitor (RTI) efficacy ϵ_α and the Protease Inhibitor (PI) efficacy ϵ_β. The mathematical properties of the model which have been studied in [1], [8] show that it captures many of the observed behavioral properties of long-term HIV dynamics [2], [36]. The system of ODEs defining the model reads

TABLE 1
The Parameters in the HIV Model

parameter	value	unit	parameter	value	unit
λ_1	10.0	$\frac{\text{cells}}{\text{mm}^3\cdot\text{day}}$	λ_2	31.98×10^{-3}	$\frac{\text{cells}}{\text{mm}^3\cdot\text{day}}$
d_1	0.01	$\frac{1}{\text{day}}$	d_2	0.01	$\frac{1}{\text{day}}$
k_1	8.0×10^{-4}	$\frac{\text{mm}^3}{\text{virions}\cdot\text{day}}$	k_2	0.1	$\frac{\text{mm}^3}{\text{virions}\cdot\text{day}}$
m_1	0.01	$\frac{\text{mm}^3}{\text{cells}\cdot\text{day}}$	m_2	0.01	$\frac{\text{mm}^3}{\text{cells}\cdot\text{day}}$
ρ_1	1	$\frac{\text{virions}}{\text{cells}}$	ρ_2	1	$\frac{\text{virions}}{\text{cells}}$
δ	0.7	$\frac{1}{\text{day}}$	c	13.0	$\frac{1}{\text{day}}$
f	0.34	–	N_T	100.0	$\frac{\text{virions}}{\text{cells}}$
λ_E	1.0×10^{-3}	$\frac{\text{cells}}{\text{mm}^3\cdot\text{day}}$	δ_E	0.1	$\frac{1}{\text{day}}$
b_E	0.3	$\frac{1}{\text{day}}$	d_E	0.25	$\frac{1}{\text{day}}$
K_b	0.1	$\frac{\text{cells}}{\text{mm}^3}$	K_d	0.5	$\frac{\text{cells}}{\text{mm}^3}$

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

The state variables in the model (1) are: 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 the second kind, V is viruses, and E is immune effectors. We use cubic millimeters (mm^3) as a volume unit. The biological meaning of the terms in (1) has been discussed in [1], [8]. Table 1 gives the values of the parameters. These same parameter values have been used in [1], [8], [9], [10].

It is well-known that, usually, after discontinuing HAART medications, the viral load rebounds to a high level; see [37], [38], for example. An exception to this was observed with the "Berlin patient" who interrupted HAART medications twice and, after medications were stopped permanently, the viral load stayed at a low level [39]. This behavior was attributed to a strong immune response stimulated by interruptions in the therapy. The model (1) exhibits a pronounced viral rebound when the immune response is not strong, that is, when the level of the immune effectors E is not high enough. This is clearly demonstrated in the paper [8], while it is not transparent in the numerical results presented in Section 4 due to fairly short interruptions in the medications. Particularly, the fitting of the model parameters using clinical HIV treatment interruption data in the study [40] lead to many patients experiencing viral rebounds to a good match between the data and the states of the model; for more details, see [41].

When no medication is administrated ($\epsilon_\alpha = \epsilon_\beta = 0$), the steady states of the model (1), that is, the states in which the time derivatives are zero, are described and analyzed in [1], [8]. A steady state of particular interest is the so-called "healthy" steady state given by

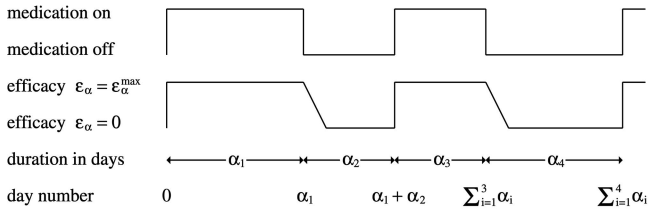

Fig. 1. The beginning of the RTI medication schedule and its efficacy.

$$
\begin{aligned}
&T_1 = 967.839\,\tfrac{\text{cells}}{\text{mm}^3}, \quad T_2 = 0.621\,\tfrac{\text{cells}}{\text{mm}^3}, \quad T_1^* = 0.076\,\tfrac{\text{cells}}{\text{mm}^3},\\
&T_2^* = 0.006\,\tfrac{\text{cells}}{\text{mm}^3}, \quad V = 0.415\,\tfrac{\text{virions}}{\text{mm}^3}, \quad \text{and} \quad E = 353.108\,\tfrac{\text{cells}}{\text{mm}^3}.
\end{aligned}
\tag{2}
$$

The value of T_1 in (2) is close to a thousand cells per cubic millimeter, which corresponds to a person without HIV under the model (1) and given parameters. This is the reason for calling this state "healthy." Furthermore, the viral load V in (2) is fairly low.

Following [1], [8], [9], [10], we have chosen the initial condition of (1) to be the acute infection

$$
\begin{aligned}
&T_1 = 1,000\,\tfrac{\text{cells}}{\text{mm}^3}, \quad T_2 = 3.198\,\tfrac{\text{cells}}{\text{mm}^3}, \quad T_1^* = 0.0\,\tfrac{\text{cells}}{\text{mm}^3},\\
&T_2^* = 0.0\,\tfrac{\text{cells}}{\text{mm}^3}, \quad V = 0.001\,\tfrac{\text{virions}}{\text{mm}^3}, \quad \text{and} \quad E = 0.01\,\tfrac{\text{cells}}{\text{mm}^3}.
\end{aligned}
\tag{3}
$$

In the acute infection (3), the state variables have the same values as that of a healthy person except that there is a small quantity of HIV in the blood.

In order to numerically simulate the pathogenesis of HIV, we have discretized the model (1) in time using a second-order backward differentiation formula (BDF2) [42]. For the numerical results, 30 minute time steps are used, that is, for each day, 48 time steps are performed. The simulation for 750 days requires a few tenths of a second.

2.2 Medications, Their Representation, and Efficacies

We consider Structured Treatment Interruption (STI) schedules for medications. This means than, at any given time, a maximum dose of a medicine is administered to a patient or that medicine is not given. We describe the RTI medication by a binary valued function of time t, denoted by $\alpha(t)$, that is, $\alpha : [0, T] \to \{0, 1\}$, where T is the duration of the considered schedule in days. Similarly, the PI medication is defined by $\beta(t)$.

In order to make schedules easier to follow for patients, we only allow a medication to be started or stopped at one specified time during a day, say, in the morning. This means that it is necessary to decide, for each 24 hour period, if a medication is taken or not. Hence, the RTI medication is described by the binary values $\alpha(0), \alpha(1), \ldots, \alpha(T-1)$. This suggests the representation of the medication by a binary number with T bits. This type of representation was used in [8], [10]. Instead of this, we represent the medication schedule as an integer vector. Let the vector $\alpha = (\alpha_1, \alpha_2, \ldots)$ define the RTI medication schedule. The first element, α_1, gives the duration of the first time on the RTI medication period in days. The second element, α_2, gives the duration of the first time off the RTI medication period. In general, α_{2k-1}, $k = 1, 2, \ldots$, gives the length of the kth time on period in days and α_{2k}, $k = 1, 2, \ldots$, gives the length of the kth off period. Fig. 1 shows an example of the beginning of the RTI

medication schedule. The schedule for the PI medication is defined in the same way by an integer vector $\beta = (\beta_1, \beta_2, \ldots)$.

To our knowledge, this is the first time that the period representation has been used for STI antiviral medication for HIV. A similar representation was used in [29] to optimize cancer medications. This representation has several benefits over a binary representation. It leads to a vast reduction in dimensionality, which can result in much faster convergence in the optimization. Furthermore, it offers an easier way to pose constraints on the lengths of on and off medication periods. We have not considered such constraints here, but there are many reasons why they can be important. For example, without carefully constraining these periods, the probability of unpleasant and even severe side effects, as well as mutations resulting in drug-resistant strains of HIV, can be unacceptably high. Furthermore, allowing the medications to be started or stopped at many specific times during a day does not increase the dimensionality with the period representation as much as it does with the binary representation.

We choose the maximum number of on and off medication periods during the considered time interval $[0, T]$ and we denote it by $L + 1$. We require the last periods α_{L+1} and β_{L+1} to be ends to the day T. Thus, we have the constraints $\sum_{k=1}^{L+1} \alpha_k = T$ and $\sum_{k=1}^{L+1} \beta_k = T$. Due to this, it is possible to compute α_{L+1} and β_{L+1} from the other periods. Hence, we do not need to store them. We allow zero length periods and, from this, it follows that the number of on and off medication periods can be fewer than $L + 1$. We remark that, if L is odd, then the last periods are off medication periods and, if L is even, then they are on periods. In order to simplify the notations in the following, we define a vector x which is obtained by concatenating the vectors $\alpha = (\alpha_1, \ldots, \alpha_L)$ and $\beta = (\beta_1, \ldots, \beta_L)$, that is, $x = (\alpha, \beta)$.

In practice, an RTI medication cannot completely block the integration of the viral code into the target cells and a PI can only partially prevent the replication of viruses by infected cells. This means there exist some maximum efficacies $\epsilon_\alpha^{\max}$ and $\epsilon_\beta^{\max}$ which are less than one for the RTI and PI medications, respectively. We have chosen to use the values $\epsilon_\alpha^{\max} = 0.8$ and $\epsilon_\beta^{\max} = 0.4$ in the numerical experiments. The study [40] fitting the parameters of the model (1) using clinical data shows that, for different patients, these maximum efficacies essentially vary in the whole range from zero to one, with the average values close to 0.5. The paper [1] studying anti-HIV medications used the maximum RTI efficacy 0.8 while no PI medications were employed. The papers [8], [9], [10] used the values 0.7 and 0.3 for the maximum efficacies of RTI and PI, respectively. Numerical experiments in [9] show that a strong immune response allowing discontinuing medication at some point is possible when the total combined efficacy $1 - (1 - \epsilon_\alpha^{\max})(1 - \epsilon_\beta^{\max})$ as defined in [36] is higher than about 0.8. The maximum efficacies 0.7 and 0.3 approximately lead to this limiting value. The value 0.8 and 0.4 used in this paper lead to the combined total efficacy 0.88, which is closer to an average patient susceptible to having a strong immune response according to the fitted parameters in [40]. This is the reason for slightly modifying the more usual maximum efficacies 0.7 and 0.3 in the previous studies.

After starting a medication, it takes a while for the medication to be fully efficacious. For example, this could require a few hours. We neglect this transition period and assume that the RTI and PI medications have the efficacies $\epsilon_\alpha^{\max}$ and $\epsilon_\beta^{\max}$, respectively, immediately after starting to administer them. Typically, nowadays, the RTI and PI medications are taken twice a day. Their efficacies take several hours to decay after the medication has been taken for the last time. We model this as a linear decay from the maximum efficacy to zero efficacy, which requires 24 hours for both medications. The same decay rate has been used in [1], [8], [9]. An example of the RTI efficacy ϵ_α is shown in Fig. 1.

2.3 Optimization Problem

An effective HIV medication can lead to a low viral load, but it cannot completely clear HIV [36]. Many HIV infected people can live in a "healthy" condition for a long period without any HIV medication and without medical problems due to HIV. For these reasons, instead of trying to eradicate HIV our aim is to find multidrug therapies which steer the medical condition to the "healthy" state (2). Once the neighborhood of this state is reached, therapy can be discontinued due to the local asymptotic stability of the state (2).

In control theory, problems in which the aim of the control is to steer the state to a given desired nonzero state are called tracking problems. A usual way to accomplish this is to find controls which minimize a weighted least square distance from the desired state over time. Based on this approach, we formulate, similarly to [9], the optimization of a multidrug therapy schedule as the minimization of a least squares fitness function J given by a sum

$$J(\alpha, \beta) = J(x) = \sum_{i=1}^{3} w_i J_i, \tag{4}$$

where the weight coefficients w_i are $w_1 = 10$ and $w_2 = w_3 = 1$ and the objective functions J_i are

$$\begin{aligned} &J_1 = 10 \int_0^T (E - 353.108)^2 dt, \; J_2 = \int_0^T \alpha^2 \, dt, \\ &\text{and } J_3 = \int_0^T \beta^2 \, dt. \end{aligned} \tag{5}$$

In (5), T is a given time horizon, E is the measure of the immune response in (1), α is the RTI medication schedule, and β is the PI medication schedule. Thus, J_1 measures the immune effectors over the time interval $[0, T]$ while J_2 and J_3 measure the amount of medication. The same fitness function was used in [9] with an additional term for the viral load. Thus, the aim is to steer the immune response to the "healthy" state level and, at the same time, minimize the amount of medications.

The optimization of the HIV multidrug therapy for T days is defined by a constrained nonlinear integer programming problem

$$\min_x J(x) \tag{6}$$

subject to the constraints

$$\sum_{k=1}^{L} x_k \leq T \quad \text{and} \quad \sum_{k=L+1}^{2L} x_k \leq T \tag{7}$$

and the state (1) with the initial condition (3).

Our numerical studies in the following are based on a 750 day period, that is, $T = 750$. This is a sufficiently long time to reach the "healthy" state. Furthermore, based on several experiments, we have chosen the maximum number of on and off medication periods to be 132, that is, $L = 131$. Hence, the last periods are off medication. As our results will show, this number of periods is large enough in order to reach the "healthy" state while, with a much smaller number, the optimization has difficulties in attaining this.

2.4 Features of the Decision Space and the Fitness Landscape

As highlighted above, the period representation of the HIV problem leads to a reduction in dimensionality, that is, a reduction in the cardinality of the decision space compared to the binary representation. In the case of binary representation, the cardinality of the decision space is given by 2^{2T}. In the case of period representation, the cardinality of the decision space can be determined in the following way: Let us consider one medication, for example, the RTI medication; the schedule for this medication is defined by the vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{L+1})$ having the length $L + 1$. The components α_k are integers and they satisfy the inequalities $0 \leq \alpha_k \leq T$. Furthermore, the sum $\sum_{k=1}^{L+1} \alpha_k$ is T. By using an induction proof, one can easily show that the total number of possible different vectors α is given by

$$\zeta = \frac{(T + L)!}{L!\,T!}. \tag{8}$$

The number of different PI medication schedules is the same ζ. Thus, the cardinality of the decision space is given by ζ^2. For our problem, we have $L = 131$ and $T = 750$. It follows that $\zeta = 2.768 \times 10^{159}$ and the cardinality of the decision space is 7.664×10^{318}. The period representation allows the algorithm to work with a decision space having vastly lower cardinality than the binary representation has ($2^{1,500}$); more specifically it is 4.576×10^{132} times lower.

Even though the application of the period representation leads to a significant reduction in dimensionality, it is still very high and the problem is challenging to solve, especially with the fitness function being computationally relatively expensive (each fitness evaluation takes about 0.2 seconds on a PC with a 3 GHz processor). In addition, the fitness landscape presents a further difficulty. It is rather flat [30], [32], [31], meaning that most solutions of the decision space have very similar fitness values. Also, the basin of attraction around the optimal solution is very narrow. Due to these reasons, many optimization algorithms (deterministic, evolutionary, or other kind of metaheuristic) can easily stagnate, leading to a large number of fitness function evaluations without significant improvement in its value or prematurely converging to a suboptimal solution with an unsatisfactory fitness.

This paper proposes an Adaptive Multimeme Algorithm (AMmA) which combines the features of an evolutionary algorithm, a random local searcher, a deterministic local searcher, and a metaheuristic in order to explore the decision space from different and complementary perspectives. These ingredients lead to an algorithm which can subdue the difficult fitness landscape (see [33] and [43]). In addition, the proposed algorithm uses an adaptive rule which measures the fitness diversity of the problem in order to dynamically set the algorithmic parameters and to enable/disable the usage of the local searchers.

3 The Adaptive Multimeme Algorithm

This section gives a description of the algorithm designed to solve the problem in (6). In the following sections, the analysis of each algorithmic component is carried out and the comments related to the algorithmic choices are given. The last subsection of this section shows how the single algorithmic components are intelligently combined in order to solve the optimization problem.

3.1 Initial Sampling

As highlighted above, the solutions are very similar among each other, in terms of fitness values, in a very wide region of the decision space. The initial sampling is therefore a very critical issue for this class of problems. A pseudorandom initial sampling can easily lead to an initial population made up of individuals having a very similar performance, this performance being very low for all of the individuals. Moreover, in the case of an unlucky initial sampling (very probable for this fitness landscape), the flatness of the function could lead to a very slow evolution of the candidate solutions and, thus, the stagnation of the algorithm. Due to the high cardinality of the decision space, it is impossible to perform an initial sampling which covers every region of the decision space.

In order to find a compromise between the necessity of having spread out solutions and the impossibility of handling an initial population with an enormous size due to computational limitations, a Quasi-Random Initial Sampling (QRIS) [44], [45], [46] consisting of the following is proposed here: The range of variability of the genes having positions 1, 2, 3, 132, 133, and 134 is divided into three intervals, $[0, T/3]$, $[T/3 + 1, 2/3 \cdot T]$, and $[2/3 \cdot T + 1, T]$. This division individuates $3^6 = 729$ possible combinations of the intervals for these six genes. The initial sampling of these six genes is obtained by generating pseudorandomly (with uniform distribution) integer numbers within the described intervals in such a way that all possible combinations are covered. Then, 729 candidate solutions are formed from these 729 sets of the six genes by generating the remaining $262 - 6 = 256$ genes pseudorandomly.

Some additional solutions are generated pseudorandomly with uniform distribution. In this way, the first three periods of the therapy for both medications are more spread out in the decision space. The choice of gene positions where the QRIS is applied comes from a physical consideration of the problem. It has been noted that the beginning of the medications is very important in terms of immune response and, therefore, genes 1, 2, 3, 132, 133, and 134 heavily bias the fitness values. Obviously, this QRIS

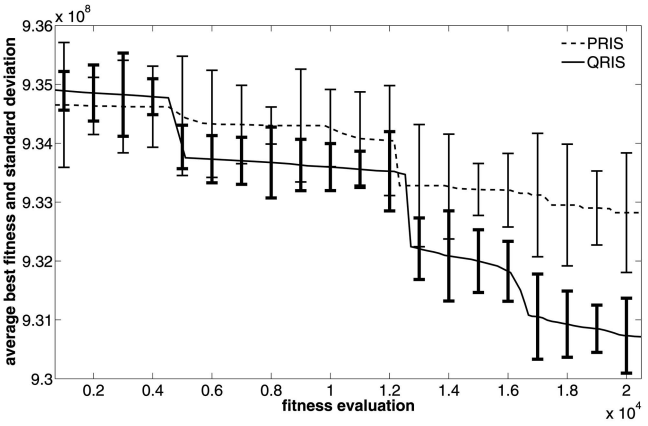

Fig. 2. Comparison between the pseudorandom and quasirandom initial sampling.

does not ensure a full coverage of all regions of the decision space, but it leads to a more robust algorithmic behavior.

In order to show the advantages compared to the classical uniformly distributed pseudorandom sampling and, thus, to justify this algorithmic choice, a numerical test has been performed. Keeping all other algorithmic parameters as constants, the algorithms using QRIS and Pseudo-Random Initial Sampling (PRIS) have been run 20 times each. The runs have been terminated after 20,000 fitness evaluations. At the end of each generation, the average of the best fitness values has been computed for the 20 test runs (Average Best Fitness). For both of the sampling techniques, if the solutions do not respect the constraints in the equations in Section 2.3, the days in which the therapy was exceeded are ignored. Fig. 2 compares the performance of the algorithms using the two sampling strategies. The standard deviation bars are also included in Fig. 2.

The PRIS offers a better performance at the very beginning of the optimization process (see [45]), but it is outperformed after 12,000 fitness evaluations by the QRIS. According to our interpretation, the QRIS generates a set of solutions that are more genotypically spread out than the ones generated by the PRIS. This effect leads to an enhanced efficiency of search in the directions toward the global optimum. As shown in Fig. 2, the PRIS-trend has a constant value for several generations and, thus, there is a risk of prematurely converging, even in the early generations. On the other hand, it is important to note that the initial sampling is only one important part of the algorithm. The QRIS can help the optimization process only during the early generations and, thus, the other algorithmic components also have to be designed carefully in order to find a solution fairly close to the optimum.

3.2 Standard Evolutionary Operators

The set of the solutions undergoes selection and variation operations at each generation. The following algorithmic choices have been made.

3.2.1 Parent Selection

At each generation, a subset of solutions is selected, on the basis of their performance, in order to constitute the set of parent solutions. The ranking parent selection [47], [48] using the stochastic universal sampling algorithm [49] has been chosen. This choice follows from the consideration that, due to the flatness of the fitness landscape, the fitness values of the individuals of the population at a typical generation are very similar to each other. It is therefore necessary to employ a parent selection mechanism which generates selection probabilities that are not based on the actual fitness values (as in the case of fitness proportional selection, see [46]).

TABLE 2
Comparison among Crossover Techniques

	Recombination on the whole chromosome		Recombination on the substrings	
	one-point	two-point	one-point	two-point
η	0.07	0.02	0.19	0.26

3.2.2 Crossover

Crossover operation generates offspring from the parents and it consists of the following: The two selected chromosomes are divided in two substrings of the same length (corresponding to the two types of medication). The two-point crossover technique [50] is then applied to each substring of the chromosomes. The reason for applying the recombination separately on the two substrings is that the application of the recombination operators to the whole chromosome could lead to solutions with worse performance. More specifically, the one-point crossover technique [51] applied to the whole chromosome is not explorative enough since it generates an offspring very similar to its parents. This can lead to a premature convergence. The two-point crossover technique applied to the whole chromosome swaps the middle part of two chromosomes. This middle part could contain the end of the first medication and the beginning of the second one. We noticed that the resulting offspring solutions generated in this way very often have worse performance than their parent solutions.

In order to better understand the behavior of the crossover for this kind of problem and, thus, justify the algorithmic choice, the following numerical test has been designed. A sampling of 1,000 points has been performed pseudorandomly with uniform distribution in the decision space and the fitness function J has been computed for all of them. These points have been pseudorandomly selected with uniform distribution and recombined by the several crossover techniques under examination, generating an offspring population consisting of 2,000 points. The fitness values of the offspring population have also been calculated. We denote by J_{avg} the average fitness value over the 1,000 parent solutions and we define η to be the number of offspring solutions with better fitness than J_{avg} divided by the size of the offspring population (i.e., 2,000). Table 2 shows the effectiveness of the several crossover techniques for the fitness landscape under examination.

As can be seen in Table 2, in all cases, the efficiency of the recombination operator is quite low.

3.2.3 Mutation

In order to explore the decision space more efficiently, some of the generated offspring solutions undergo a mutation

while budget conditions
  Generate an independent random vector made up of positive and negative integer numbers d_k having the same length as x_k;
  Check if $x_k + d_k$ is a feasible solution;
  if $J(x_k + d_k) < J(x_k)$
    $x_{k+1} = x_k + d_k$;
  end-if
end-while

Fig. 3. Localized Random Search pseudocode.

operator (see Section 3.4). The mutation occurs in the following way: Two genes of the chromosome, one belonging to the first substring and one belonging to the second, are selected pseudorandomly (with uniform distribution). For each gene, a small integer number (positive or negative) is generated pseudorandomly from a zero-mean Gaussian distribution and then added to the selected genes. In other words, a normally distributed mutation [52], [53] is employed for integers.

3.2.4 Survivor Selection

When an offspring is generated by recombination and processed by the mutation operator, a population must be selected for the subsequent generation by means of the survivor selection. Since the recombination generates many offspring solutions with worse performance than their parents (see Table 2), an age-based replacement or a generational approach are likely inefficient. This paper proposes an approach which selects, for the subsequent iteration, the best S_{pop} individuals [54], [55] (where S_{pop} is a parameter set adaptively, see Section 3.4) among both parent and offspring solutions.

3.3 Local Searchers: Functioning and Usage

In order to support the evolutionary process, three local searchers with different features are employed. These local searchers explore the decision space from different perspectives [33], [56], [57] and to compete and cooperate [58] for the common goal of determining the global optimum.

3.3.1 Localized Random Search

The Localized Random Search (LRS, see [59]) is a random searcher having the greedy pivot rule [44], which aims at exploring the decision space in every direction. The LRS tries to enhance a starting solution by adding a random vector. Fig. 3 shows the pseudocode of the LRS for the kth iteration (x_k is the candidate solution for the related iteration).

The main features of the LRS are that it executes the minimization working simultaneously on all genes of the chromosome and it executes the update of the candidate solution as soon as a trial is successful (greedy pivot rule). Due to its inner structure, this local searcher turned out to be very efficient for solutions far from the optimum. It assists and cooperates with the evolutionary framework, offering solutions having good performance and a new genotypic structure (the perturbation is executed to all genes). On the other hand, the efficiency of this algorithmic component significantly decreases when the starting solution already has fairly good performance. In this case, the LRS often does not find any solutions with better

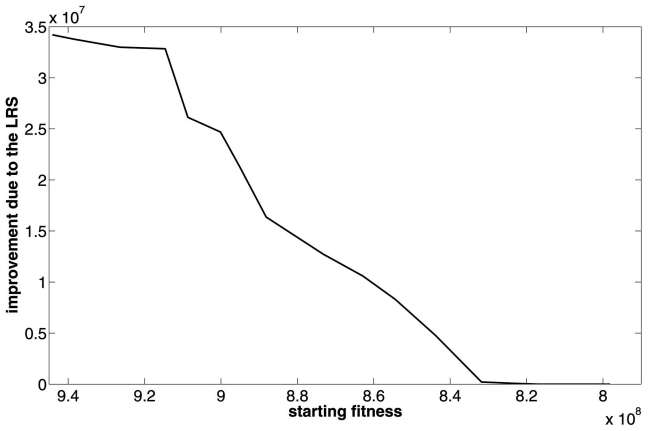

Fig. 4. Effectiveness of the LRS in dependence on the fitness value of the starting solution.

performance than the starting one. In other words, this local searcher is very powerful in the early generations of the AMmA, but it can be inefficient in the neighborhood of the optimum. Since the computational cost of the LRS is defined by the budget imposed (see Fig. 3), a failure of this local searcher would lead to a waste of computational resources.

In order to analyze the effectiveness of the LRS, the following numerical test has been designed: For a fixed budget equal to 500 fitness evaluations, the LRS has been performed on several starting solutions with different fitness values. The improvement of the solution with respect to the corresponding initial solution has been computed. Fig. 4 shows the improvements obtained by the LRS versus the fitness of the starting solution.

3.3.2 Steepest Descent Explorer

The Steepest Descent Explorer (SDE) has completely different features, compared with the LRS, in terms of pivot rule and neighborhood generating function [44]. In addition, unlike the LRS, the SDE is a fully deterministic local searcher. The SDE consists of the following: For a given step size h, a 262×262 direction matrix H is formed by setting its diagonal elements to be h. The SDE perturbs an initial solution by adding to it and by subtracting from it the rows of the direction matrix one by one. This generates a set of 524 new candidate solutions. The SDE selects from these 524 solutions the one with the best performance. If the best solution has higher performance than the initial one, then replacement occurs; otherwise, the SDE fails.

Thus, the SDE is a deterministic local searcher with a steepest descent pivot rule which explores the neighborhood of a candidate solution in order to perform the hill climb (or, more properly, the hill descent due to minimization) within the basin of attraction, thus executing a "lifetime learning" [60] of a candidate solution. In other words, the SDE executes the "exploratory move" of the Hooke-Jeeves algorithm (see [61], [62], [63]) in the case of a discrete decision space. Fig. 5 shows the pseudocode of the SDE for a given h and a given initial solution x_0.

The SDE is a simple and computationally relatively cheap local searcher which could efficiently improve the fitness value of the initial solution. It is important to note

```
Generate the direction matrix H = hI;
for each row of the matrix H
   Calculate x_0 + row(H) and x_0 − row(H);
   Check if x_0 + row(H) (and x_0 − row(H)) is a feasible solution;
end-for
Set x* to be the solution with the best performance among the newly calculated solutions;
if x* has better performance than x_0;
   x_0 = x*;
end-if
```

Fig. 5. Steepest Descent Explorer pseudocode.

that, due to the nature of the problem, candidate solutions with one gene difference can have quite a large difference in the fitness value. This local searcher can be very valuable in assisting the optimization process. On the other hand, due to its inner structure, the SDE does not aim to explore other areas of the decision space (it changes just one gene of the chromosome) and, therefore, its use in the stage of the evolutionary process which aims to find new good genotypes can be inefficient. In addition, the success of this local searcher depends on the position of the candidate solution within the decision space and not on its starting fitness value. In other words, the SDE can fail or succeed in every stage of the optimization process.

3.3.3 Simulated Annealing

The Simulated Annealing (SA) metaheuristic [64], [65] offers a third exploratory perspective in the decision space which can choose a search direction leading to a basin of attraction different from the one where starting point x_0 is. The exploration is performed by using the same mutation operator as was described in the evolutionary framework (see Section 3.2.3).

For the sake of clarity, the pseudocode of the SA for a given current best solution x_{cb} is shown in Fig. 6.

The main reason to employ the SA in the AMmA is that the evolutionary framework should be assisted in finding better solutions which improve the available genotype, but, at the same time, exploring areas of the decision space not yet explored. This local searcher changes two genes by adding or subtracting pseudorandom quantities. It accepts, with a certain probability, solutions with worse performance in order to obtain a global enhancement in a more promising basin of attraction. In addition, the exploratory logic aims, unlike the LRS, to exploit the available genotype and, at the same time, unlike the SDE, to explore a relatively wide area of the decision space. Thus, the SA is included as a local searcher having an intermediate exploratory pressure and efficiency between the LRS and the SDE.

The application of the SA local searcher can be successful, in most of the cases, in the early generations and in the late generations as well. Moreover, due to its structure, the SA can efficiently offer solutions in unexplored basins of attractions and, thus, prevent an undesired premature convergence. The most delicate issue related to the SA is the choice of parameters. The SA has two parameters, which are the budget and the initial temperature $Temp^0$ (see Fig. 6). Even though these parameters should be simultaneously set since the success of the local searcher depends on both, the budget has been fixed to be 500 fitness evaluations (in order to have a constant computational cost for the SA). The temperature is adaptively set following the necessity of the evolutionary process (see Section 3.4). The temperature

```
while budget conditions
Perturb the current best solution by mutation thus generating x_per;
Check if x_per is a feasible solution;
Calculate the fitness value of the perturbed solution;
if J(x_per) < J(x_cb)
   Accept the perturbed solution as a new current best point (x_cb = x_per);
else
   Calculate the probability p = e^((J(x_cb) − J(x_per))/Temp);
   Generate a pseudo-random value u ∈ [0, 1];
   if u < p
      Accept the perturbed solution as a new current best point;
   else
      Keep x_cb as the current best solution;
   end-if
end-if
Reduce Temp;
end-while
```

Fig. 6. Simulated Annealing pseudocode.

$Temp$ is reduced according to a hyperbolic law following the suggestions in [66].

3.4 Adaptation

In order to design a robust multimeme algorithm [56], [33], a set of adaptive rules has been implemented to choose parameters dynamically and to assist in the coordination of the local searchers. As highlighted above, the main difficulty of this problem is the coexistence of a high cardinality decision space and a rather flat fitness landscape (i.e., the values of the fitness function have a small variability in a large part of the decision space). These properties make the problem hard to solve since the algorithm has a high risk of prematurely converging or stagnating.

For dynamically balancing the exploration and exploitation of the algorithm [57], a parameter

$$\psi = 1 - \left| \frac{J_{avg} - J_{best}}{J_{worst} - J_{best}} \right| \tag{9}$$

has been defined, where J_{worst}, J_{best}, and J_{avg} are the worst, best, and average of the fitness function values in the population, respectively. The parameter ψ is a population diversity index which is well-suited for flat fitness landscapes. It measures the population diversity in terms of fitness and it is relative to the range of the fitness values $[J_{best}, J_{worst}]$ in the population. Thus, even when all fitness values are very similar, leading to J_{best} and J_{worst} being close to each other, ψ still gives a well-scaled measure since it uses the relative distance of J_{avg} from J_{best}. The population has high diversity when $\psi \approx 1$ and low diversity when $\psi \approx 0$. A low diversity means that the population is converging. We remark that the absolute diversity measure used in [10], [67], and [68] is inadequate in this case since, according to it, the population diversity would be very low most of the time.

3.4.1 Dynamic Parameter Setting

The coefficient ψ is used to dynamically tune the algorithmic parameters [69], [70], [71]. At each generation, the number of crossovers is adaptively chosen to be

$$N_{cr} = N_{cr}^{f} + N_{cr}^{v} \cdot (1 - \psi), \tag{10}$$

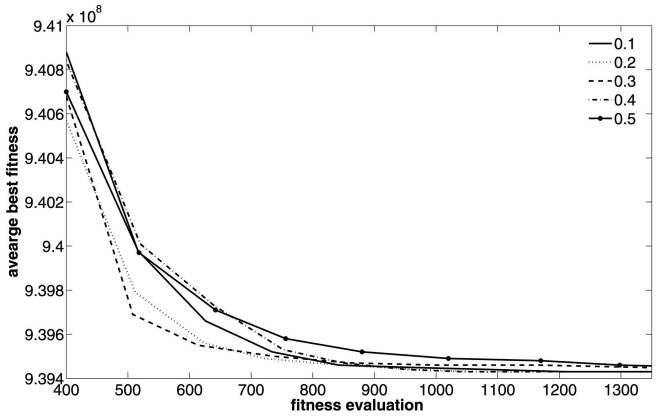

Fig. 7. Algorithmic performance for several values of p_m^{max}.

where N_{cr}^f is the minimum number of crossovers and N_{cr}^v is the maximum number of additional crossovers. When the population diversity is low, that is, $\psi \approx 0$, more offspring are generated in order to increase diversity, while, when $\psi \approx 1$ and the diversity is quite high, it is desirable to exploit more available solutions. In this case, a relatively small number of crossovers is more adequate.

After all crossovers have been performed, the mutation probability is given by

$$p_m = p_m^{\max} \cdot (1 - \psi). \tag{11}$$

The role of this dynamic mutation probability is to increase the explorative pressure in the presence of low population diversity and to decrease it in the presence of high population diversity. The probability p_m gives the proportion of the population undergoing the mutation.

The value p_m^{max} is a parameter that has been tuned by performing a numerical test which analyzes its effect on the algorithmic performance. The AMmA has been run for 1,350 fitness evaluations for the values $0.1, 0.2, 0.3, 0.4, 0.5$, while the other parameters have been unchanged. The AMmA has been run 25 times for each of the previous values. For these five values of p_m^{max}, the best fitness values have been saved at the end of each iteration. These values have then been averaged over the 25 available values (Average Best Fitness, see also Section 3.1). Fig. 7 shows the comparison of the algorithmic performance for the above values of p_m^{max}.

On the basis of the obtained results, p_m^{max} has been set to be 0.3 for this problem.

For the survivor selection, the population size is computed as

$$S_{pop} = S_{pop}^f + S_{pop}^v \cdot (1 - \psi), \tag{12}$$

where S_{pop}^f and S_{pop}^v are the fixed minimum and maximum sizes of the variable population, respectively. The coefficient ψ is then used to dynamically set the population size [72], [73], [71] in order to prevent a premature convergence and stagnation. When the population is highly diverse, a small number of solutions need to be exploited. When $\psi \approx 0$, the population is converging and a larger population size is required to increase the exploration.

3.4.2 Coordination of the Local Searchers

As described above, three local searchers are employed to assist the evolutionary framework. Their parameters and usage are adaptively chosen using the parameter ψ in order to properly coordinate the use of the local searchers by taking into account their differences and the necessities of evolution.

The LRS is activated by the condition $\psi \in [0.1, 0.5]$. This adaptive rule is based on the observation that this local searcher can very efficiently increase the population diversity during the early generations (see Fig. 4). If ψ is lower than 0.5, it means that the population diversity is decreasing and the application of the LRS can introduce a new genotype in the population which can prevent a premature convergence. On the other hand, the condition $\psi < 0.1$ is satisfied in the late generations when the convergence is approaching and the fitness value has already been drastically reduced. In this case, the application of the LRS can be unsuccessful, leading to a waste of computational resources. It is important to remark that the LRS is applied to an individual of the population pseudorandomly (with uniform distribution) selected. This choice is based on the consideration that this local searcher can lead to a significant improvement from any starting solution in the population.

The SDE is activated when $\psi < 0.1$ and it is applied to the solution with best performance. The basic idea behind this adaptive rule is that the SDE has the role of quickly improving the best solution while staying in the same basin of attraction. The condition $\psi < 0.1$ means that the SDE is employed when there are some chances that optimal convergence is approaching. An early application of this local searcher can be inefficient since a high exploitation of solutions having poor fitness values would not lead to significant improvements of the population. In addition, since this local searcher is not explorative, its usage when a high exploration is required would not lead to the introduction of new genotypes. The SDE requires the parameter h (the step size) to be set, which is adaptively chosen to be $\mathrm{ceil}(100 \times \psi)$ at each activation. This adaptive rule means that the exploration of the SDE occurs within a radius between 1 and 10 (if $\psi \approx 0.1h$ is set to 10, if $\psi < 0.01h$ is set to 1) according to the state of convergence of the algorithm. Far from the convergence, the SDE tries to explore in a rather wide area while, near the convergence, it is less explorative and aims to fully exploit solutions which already have good performance in order to "end the game" [60].

The SA is activated when $\psi < 0.2$ and it is applied to the individual with the second best performance in the population. Since an important property of this local searcher is that it could lead to improvements to a starting solution without changing the genotype very much but exploring some neighbor basin of attractions, the activation has been chosen to occur when the convergence is approaching (i.e., when the population is supposed to be within a basin of attraction). The SA has the role of exploiting already good genotypes and, at the same time, exploring the neighborhood. The application of the SA can lead a solution which is worse than the starting one. Due to this reason, it is applied to the second best individual. This gives a chance to enhance a solution with good performance

```
Quasi-Random Initial Sampling;
Fitness evaluation of the initial population;
Calculate ψ = 1 − |(J_avg − J_best)/(J_worst − J_best)|;
while global budget conditions and ψ > 0
  Parent selection by ranking;
  Calculate N_cr = N_cr^f + N_cr^v · (1 − ψ);
  Generate by crossover 2 × N_cr offspring individuals;
  Calculate p_m = p_m^max · (1 − ψ);
  Execute the mutation on the offspring individuals according to the mutation probability p_m;
  Calculate the fitness of the offspring individuals;
  Sort the population made up of parents and offspring according to their fitness values;
  if ψ ∈ [0.1, 0.5]
    Select one individual of the population pseudo-randomly;
    Execute the LRS on the selected individual;
    if the LRS succeeds
      Execute the SDE on the individual enhanced by the LRS;
    end-if
  end-if
  if ψ < 0.1
    Execute the SDE on the individual with best performance;
  end-if
  if ψ < 0.2
    Execute the SA on the individual with the 2^nd best performance;
    if the SA succeeds
      Execute the SDE on the individual enhanced by the SA;
    end-if
  end-if
  Calculate S_pop = S_pop^f + S_pop^v · (1 − ψ);
  Select the S_pop best individuals to the subsequent generation;
  Calculate ψ = 1 − |(J_avg − J_best)/(J_worst − J_best)|;
end-while
```

Fig. 8. Adaptive Multimeme Algorithm pseudocode.

without possibly ruining the genotype of the best solution. The initial temperature $Temp^0$ has to be chosen for this local searcher. It is adaptively set to be $Temp^0 = \|J_{avg} - J_{best}\|$. This means that the probability of accepting a worse solution depends on the state of the convergence. In other words, the algorithm does not accept worse solutions when the convergence has practically occurred.

An additional rule for the SDE has been implemented. When the LRS or the SA has succeeded in enhancing the starting solution, the algorithm tries to further enhance it by the application of the SDE. This choice can be justified by the consideration that, when the LRS and the SA succeed, they return a solution having better performance with a genotype quite different from the starting one and, thus, belonging to a region of the decision space which has not yet been exploited.

3.5 Algorithmic Design

The pseudocode of the AMmA in Fig. 8 describes how the algorithmic components introduced and analyzed separately are combined. There are two stopping criteria for the algorithm which are when a preset budget is exceeded or when $\psi = 0$, i.e., the population is made up of one unique genotype.

4 Numerical Results

4.1 Experimental Setup

For the AMmA, 50 simulation experiments have been executed. Each experiment has been stopped after 85,000 fitness evaluations. At the end of each generation, the best fitness value has been saved. The average over the 50 experiments defines the Average Best Fitness (ABF). Analogously, 50 experiments have been carried out with a Genetic Algorithm (GA) and an Evolution Strategy (ES) in order to perform a comparison of the performance between the AMmA and two classical methods.

4.1.1 Genetic Algorithm

A standard steady state Genetic Algorithm (GA) [74] has been implemented. The GA employs a pseudorandom initial sampling (PRIS), two-point crossover, and swap mutation (see [46]).

4.1.2 Evolution Strategy

An Evolution Strategy (ES) for our discrete problem has been implemented. As a standard ES, this ES does not contain any parent selection and, thus, it considers all populations to be a population of parents [75]. An intermediate recombination which rounds to the nearest integer has been implemented and the Gaussian mutation (rounding to the nearest integer) has been implemented, resorting the $1/5$ success rule [76]. Finally, a $(\mu + \lambda)$ strategy has been chosen.

Table 3 shows the parameter settings for the adaptive multimeme algorithm, the GA, and the ES.

4.1.3 Simulated Annealing

A Simulated Annealing (SA) [64], [65] with the maximum budget fixed to 85,000 fitness evaluations has also been run for the comparison. Also, in this case, 50 runs have been executed. The initial temperature has been set to be 5×10^8 and a hyperbolic reduction law of the temperature has been used [66]. For each of the 50 experiments, the starting

TABLE 3
Parameter Setting for the AMmA, the GA, and the ES

PARAMETER	AMmA	GA	ES
size of initial population	500	500	500
population size for subsequent iterations	dynamic between 80 and 320	160	160
number of crossovers per generation	dynamic between 40 and 160	100	100
mutation probability	dynamic between 0 and 0.3	0.1	1 with 1/5 success rule
fitness evaluations	85000	85000	85000

solution has been generated pseudorandomly with uniform distribution. The average fitness value over the 50 experiments has been calculated after every 100 fitness evaluations. Analogously to the case of the AMmA, GA, and ES, this average value is also called Average Best Fitness (ABF).

4.2 The Results and Their Comparison

Fig. 9 and Fig. 10 show the Reverse Transcriptase Inhibitor (RTI) and Protease Inhibitor (PI) efficacies, respectively, for the most effective HIV therapy schedule found by the AMmA over the 50 experiments carried out. Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, and Fig. 16 show the behavior of the state variables of the model under the best therapy. Fig. 9 and Fig. 10 show that both medications are stopped before 400 days. This early termination of the medical treatment reduces undesired side effects and the possibility of mutations leading to drug-resistant HIV strains. Fig. 11 shows that the number of uninfected CD4+ T-cells stays very high at all times. This means that the immune system is not compromised at any time. The choice of the fitness, the period representation of schedules, and the proposed algorithm all contribute to this satisfactory result.

The previous studies in [8] and [10], based on a binary representation, slightly lower drug efficacies, and different optimization algorithms, lead to medical treatments which were stopped after 590 and 500 days, respectively. Fig. 16 shows the immune response corresponding to the best therapy schedule given by the AMmA and compares it with the results obtained using binary representations and by means of two different optimization algorithms. More specifically, the shown immune responses are obtained by using the Heuristic algorithm proposed in [8] and by using the Adaptive Evolutionary Algorithm with Intelligent Mutation Local Searchers (AEA with IMLSs) proposed in [10]. The therapies given by these algorithms lead to an immune response which reached the "healthy" steady state level in about 600 and 500 days, respectively, while the therapy proposed by the AMmA required less than 400 days to reach the same level. Thus, the results presented appear to be very promising.

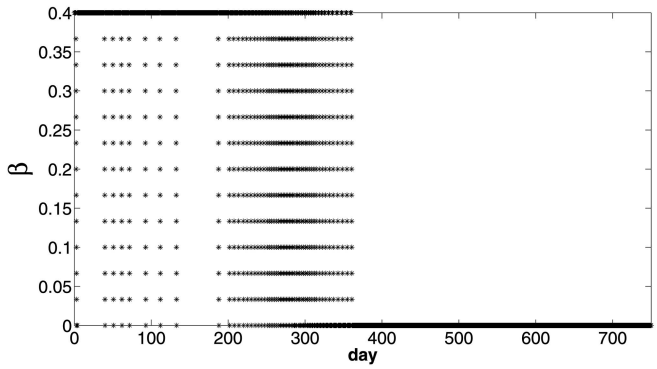

Fig. 10. Protease Inhibitor (PI) medication efficacy.

4.3 Algorithmic Analysis of the Results and Comparison with Other Optimization Algorithms

In order to better explain the behavior of the AMmA, the diagram plotting ψ versus fitness evaluations in the most successful experiment is shown in Fig. 17.

The behavior of the parameter ψ is very oscillatory before the convergence ($\psi = 0$) occurs. Due to the flatness of most of the landscape, ψ takes relatively low values, meaning that the fitness values of the population are very similar to each other and the small oscillations are basically due to the effect of the dynamic parameter setting (mainly, the variable population size [69]). On the other hand, the presence of some higher peaks in the trend of ψ is due to successful local searches. An abrupt increase of ψ corresponds to the introduction of a new individual into the population, which has far better performance than the others. More specifically, the peaks around 10,000 fitness evaluations are due to successful searches by the LRS, while the peaks around 60,000 fitness evaluations are caused by successful searches by the SA.

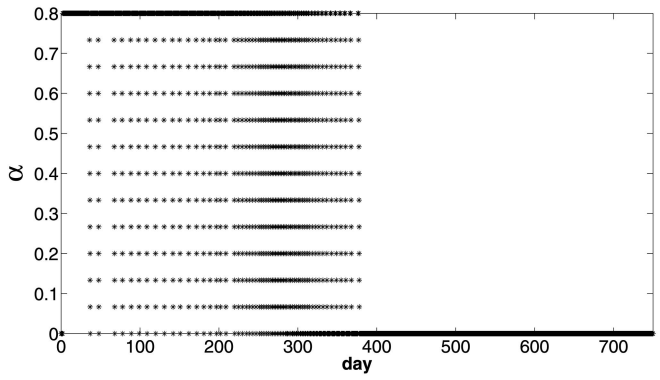

Fig. 9. Reverse Transcriptase Inhibitor (RTI) medication efficacy.

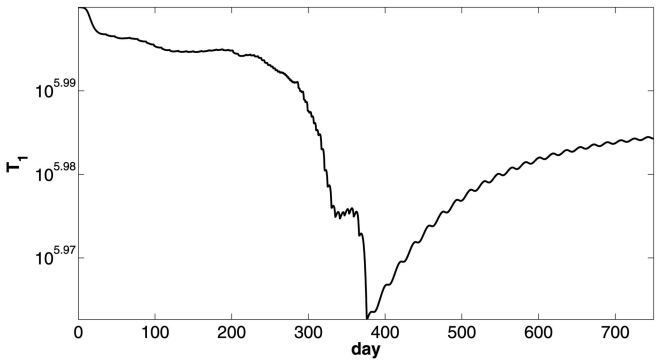

Fig. 11. Uninfected CD4+ T-cells.

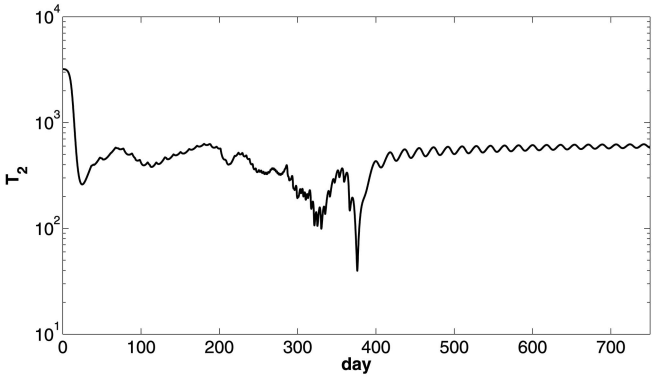

Fig. 12. Uninfected target cells of the second kind.

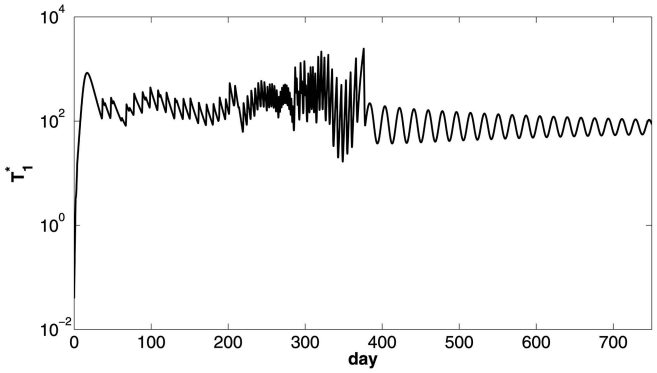

Fig. 13. Infected CD4+ T-cells.

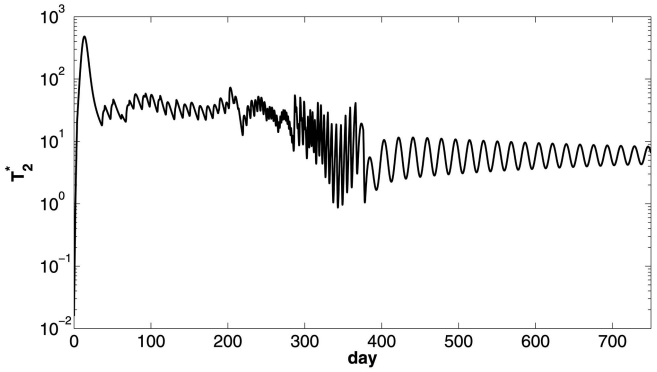

Fig. 14. Infected target cells of the second kind.

Fig. 18 shows the behavior of S_{pop}, N_{cr}, and p_m in the most successful experiment of the AMmA. Since the trends of S_{pop}, N_{cr}, and p_m are proportional, they are represented in Fig. 18 by a unique trace and three differently scaled y-axes.

Table 4 gives, for each algorithm under examination, the fitness J^b obtained by the most successful experiment (over the 50 sample runs), the related single objective functions J_i, the average best fitness at the end of the experiments $< J >$, the fitness of the least successful experiment J^w, and the standard deviation σ divided to the related value of $< J >$.

Regarding the dominance of the given solutions, it can be seen that the best solution obtained by AMmA strictly dominates the best solutions given by the other methods. Concerning the robustness of the algorithms, the value $\sigma/ < J >$ is very small for all algorithms after 85,000 fitness evaluations. This basically means that the four algorithms offer a good performance in terms of robustness.

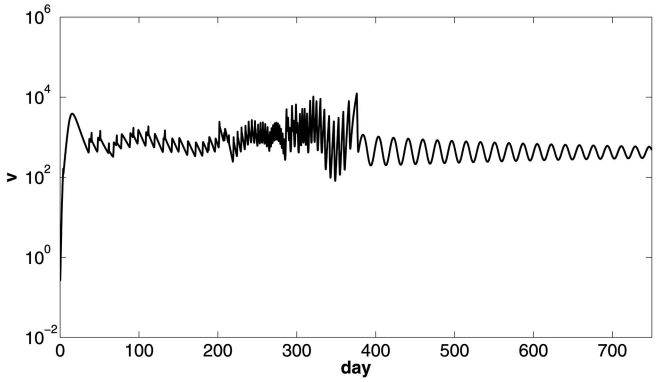

Fig. 15. Viral load.

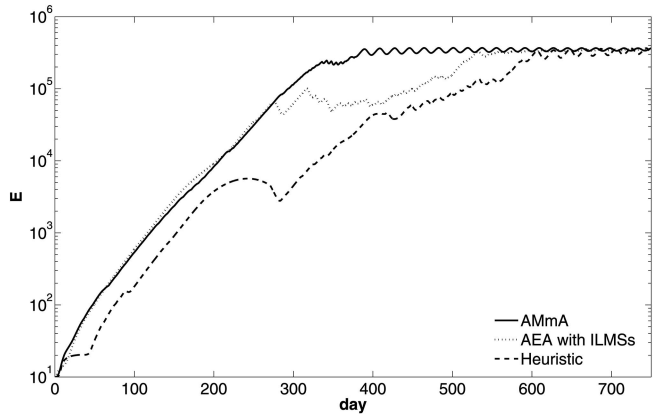

Fig. 16. Immune response.

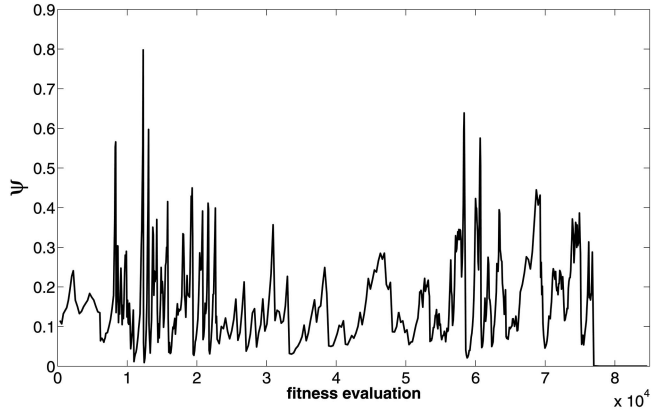

Fig. 17. Behavior of ψ.

Fig. 19 shows the algorithmic performance of the four algorithms under examination. The standard deviation bars (in circles) are also included in the graph.

From this figure and Table 4, it is qualitatively clear that the AMmA leads to significantly better results than the other methods. Moreover, it is interesting to consider the relationship of Fig. 19 and Fig. 17. Although Fig. 17 refers to one experiment while Fig. 19 is based on an average, it is clear that, at around 10,000 fitness evaluations and after 60,000 fitness evaluations, there are steep decreases in the diagram of the algorithmic performance of the AMmA. These decreases correspond to high oscillations in the trend of ψ. According to our interpretation, this phenomenon is due to successful runs of the local searchers, which led to the generation of solutions with better performance than the

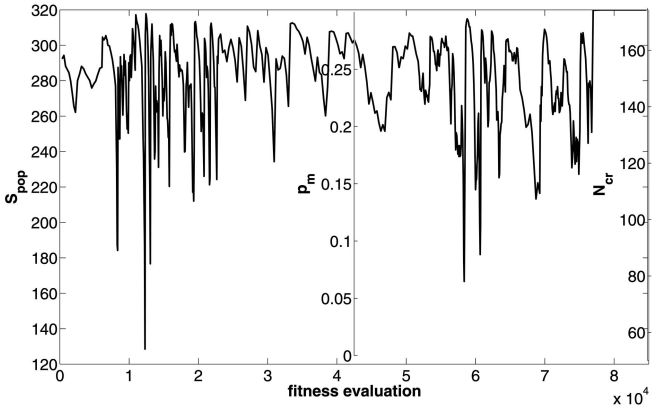

Fig. 18. Behavior of S_{pop}, N_{cr}, and p_m

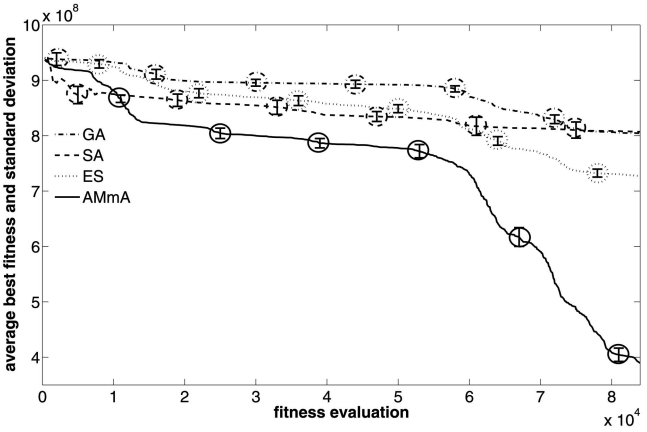

Fig. 19. Comparison of the algorithmic performance of the AMmA with the GA, the ES, and the SA.

rest of the population and, thus, to a temporary increase of the population diversity. In addition, it can be seen that the AMmA presents a similar algorithmic performance compared to the other algorithms for about 60,000 fitness evaluations and, after a certain point, it clearly starts to outperform the other methods. This effect can be seen as the capability of the AMmA in detecting the optimal basin of attraction after having explored other areas of the decision space and thus properly exploiting a promising search direction.

In order to perform a quantitative comparison, the following statistical test has been designed: For each algorithm, the most and least successful results have been saved (see Table 4) and the interval $[J_b, J_w]$ has been considered as a two-sided tolerance interval. Following the procedure given in [77] and [78], 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 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 $\left[J^w, J^b\right]$. This result is valid for all four algorithms. Moreover, considering that J^w for the AMmA is much lower than J^b of the other three algorithms and that the difference between the J^w of the AMmA and the J^b of the other algorithms is much larger than the width of each tolerance interval, it can be concluded that it is highly improbable that the GA, the ES, and the SA could outperform the AMmA.

5 Conclusion

This paper proposes an Adaptive Multimeme Algorithm (AMmA) for designing HIV multidrug therapies. The AMmA is an optimization algorithm consisting of an evolutionary framework having dynamic parameters and three different local searchers which are adaptively employed in order to explore the decision space from complementary perspectives and exploit, in various ways, the available candidate solutions.

The optimal solution given by the AMmA is very satisfactory since it leads to an immune response which reaches a fairly healthy steady state in a significantly shorter time than the previous studies. In addition, the proposed medication contains a relatively low number of medication days and, therefore, helps to avoid harmful side effects and mutations of HIV to drug-resistant strains.

Numerical comparisons show that the AMmA outperforms three other standard methods for this class of problems. In particular, the AMmA proved to have better performance than the other algorithms in avoiding stagnation and premature convergence, thus properly handling the high cardinality of the decision space and the flatness of the fitness landscape.

Acknowledgments

The numerical experiments carried out have been executed using the computers of the Laboratory of Electrical Machines, Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy. The authors wish to thank Professor Luigi Salvatore for his kind support in this and Anna V. Kononova for the precious discussions.

TABLE 4
Numerical Results

METHOD	J_1	J_2	J_3	J^b	$< J >$	J^w	$\sigma / < J >$
GA	8.01238×10^7	527	593	8.0124×10^8	8.0428×10^8	8.3716×10^8	0.0122
SA	7.9533×10^7	710	326	7.9809×10^8	8.0693×10^8	8.2973×10^8	0.0218
ES	7.1909×10^7	484	490	7.1910×10^8	7.2565×10^8	7.4224×10^8	0.0098
AMmA	3.6439×10^7	285	287	3.6768×10^8	3.7742×10^8	3.8863×10^8	0.0148

REFERENCES

[1] B.M. Adams, H.T. Banks, M. Davidian, H.-D. Kwon, H.T. Tran, S.N. Wynne, and E.S. Rosenberg, "HIV Dynamics: Modeling, Data Analysis, and Optimal Treatment Protocols," *J. Computer Applied Math.,* vol. 184, no. 1, pp. 10-49, 2005.

[2] S. Bonhoeffer, M. Rembiszewski, G.M. Ortiz, and D.F. Nixon, "Risks and Benefits of Structured Antiretroviral Drug Therapy Interruptions in HIV-1 Infection," *AIDS,* vol. 14, no. 15, pp. 2313-2322, 2000.

[3] A. Perelson, D. Kirschner, and R. de Boer, "The Dynamics of HIV Infection of CD4+ T Cells," *Math. Bioscience,* vol. 114, pp. 81-125, 1993.

[4] A.S. Perelson and P.W. Nelson, "Mathematical Analysis of HIV-1 Dynamics in Vivo," *SIAM Rev.,* vol. 41, no. 1, pp. 3-44, 1999.

[5] D. Wodarz and M.A. Nowak, "Specific Therapy Regimes Could Lead to Long-Term Immunological Control of HIV," *Proc. Nat'l Academy of Sciences USA,* vol. 96, no. 25, pp. 14464-14469, 1999.

[6] D. Wodarz, "Helper-Dependent vs. Helper-Independent CTL Responses in HIV Infection: Implications for Drug Therapy and Resistance," *J. Theoretical Biology,* vol. 213, no. 3, pp. 447-459, 2001.

[7] D. Wodarz and M.A. Nowak, "Mathematical Models of HIV Pathogenesis and Treatment," *BioEssays,* vol. 24, no. 12, pp. 1178-1187, 2002.

[8] B.M. Adams, H.T. Banks, H.-D. Kwon, and H.T. Tran, "Dynamic Multidrug Therapies for HIV: Optimal and STI Control Approaches," *Math. Bioscience Eng.,* vol. 1, no. 2, pp. 223-241, 2004.

[9] H.T. Banks, H.-D. Kwon, J. Toivanen, and H.T. Tran, "A State Dependent Riccati Equation Based Estimator Approach for HIV Feedback Control," *Optimal Control Applied Methods,* vol. 27, no. 2, pp. 93-121, 2006.

[10] F. Neri, J. Toivanen, and R.A.E. Mäkinen, "An Adaptive Evolutionary Algorithm with Intelligent Mutation Local Searchers for Designing Multidrug Therapies for HIV," *Applied Intelligence,* to appear, 2007.

[11] J. Alvarez-Ramirez, M. Meraz, and J.X. Velasco-Hernandez, "Feedback Control of the Chemotherapy of HIV," *Int'l J. Bifurcation and Chaos,* vol. 10, no. 9, pp. 2207-2219, 2000.

[12] S. Butler, D. Kirschner, and S. Lenhart, "Optimal Control of Chemotherapy Affecting the Infectivity of HIV," *Advances in Math. Population Dynamics: Molecules, Cells, Man,* pp. 104-120, 1997.

[13] R. Culshaw, S. Ruan, and R.J. Spiteri, "Optimal HIV Treatment by Maximising Immune Response," *J. Math. Biology,* vol. 48, no. 5, pp. 545-562, 2004.

[14] K.R. Fister, S. Lenhart, and J.S. McNally, "Optimizing Chemotherapy in an HIV Model," *Electronic J. Differential Equations,* vol. 1998, no. 32, pp. 1-12, 1998.

[15] H. Shim, S.J. Han, C.C. Chung, S. Nam, and J.H. Seo, "Optimal Scheduling of Drug Treatment for HIV Infection: Continuous Dose Control and Receding Horizon Control," *Int'l J. Control Automation Systems,* vol. 1, no. 3, pp. 401-407, 2003.

[16] D. Kirschner, S. Lenhart, and S. Serbin, "Optimal Control of the Chemotherapy of HIV," *J. Math. Biology,* vol. 35, no. 7, pp. 775-792, 1997.

[17] U. Ledzewicz and H. Schättler, "On Optimal Controls for a General Mathematical Model for Chemotherapy of HIV," *Proc. 2002 Am. Control Conf.,* vol. 5, pp. 3454-3459, 2002.

[18] R. Zurakowski, M.J. Messina, S.E. Tuna, and A.R. Teel, "HIV Treatment Scheduling via Robust Nonlinear Model Predictive Control," *Proc. Fifth Asian Control Conf.,* 2004.

[19] R. Zurakowski and A.R. Teel, "Enhancing Immune Response to HIV Infection Using MPC-Based Treatment Scheduling," *Proc. 2003 Am. Control Conf.,* vol. 2, pp. 1182-1187, 2003.

[20] R. Zurakowski, A.R. Teel, and D. Wodarz, "Utilizing Alternate Target Cells in Treating HIV Infection through Scheduled Treatment Interruptions," *Proc. 2004 Am. Control Conf.,* vol. 1, pp. 946-951, 2004.

[21] R. Zurakowski and A.R. Teel, "A Model Predictive Control Based Scheduling Method for HIV Therapy," *J. Theoretical Biology,* vol. 238, no. 2, pp. 368-382, 2006.

[22] M.A.L. Caetano and T. Yoneyama, "Short and Long Period Optimization of Drug Doses in the Treatment of AIDS," *Anais da Academia Brasileira de Ciéncias (Annals of the Brazilian Academy of Sciences),* vol. 74, no. 3, pp. 379-392, 2002.

[23] A.M. Jeffrey, X. Xia, and I.K. Craig, "When to Initiate HIV Therapy: A Control Theoretic Approach," *IEEE Trans. Biomedical Eng.,* vol. 50, no. 11, pp. 1213-1220, 2003.

[24] J.J. Kutch and P. Gurfil, "Optimal Control of HIV Infection with a Continuously-Mutating Viral Population," *Proc. 2002 Am. Control Conf.,* vol. 5, pp. 4033-4038, 2002.

[25] L.M. Wein, S.A. Zenios, and M.A. Nowak, "Dynamic Multidrug Therapies for HIV: A Control Theoretic Approach," *J. Theoretical Biology,* vol. 185, no. 1, pp. 15-29, 1997.

[26] S.H. Bajaria, G. Webb, and D.E. Kirschner, "Predicting Differential Responses to Structured Treatment Interruptions during HAART," *Bull. Math. Biology,* vol. 66, no. 5, pp. 1093-1118, 2004.

[27] M.E. Brandt and G. Chen, "Feedback Control of a Biodynamical Model of HIV-1," *IEEE Trans. Biomedical Eng.,* vol. 48, no. 7, pp. 754-759, 2001.

[28] H.R. Joshi, "Optimal Control of an HIV Immunology Model," *Optimal Control Applied Methods,* vol. 23, no. 4, pp. 199-213, 2002.

[29] M. Villasana and G. Ochoa, "Heuristic Design of Cancer Chemotherapies," *IEEE Trans. Evolutionary Computation,* vol. 8, no. 6, pp. 513-521, 2004.

[30] B. Derrida and L. Peliti, "Evolution in a Flat Fitness Landscape," *Bull. Math. Biology,* vol. 53, pp. 355-382, 1991.

[31] N.F. McPhee and R. Poli, "A Schema Theory Analysis of Mutation Size Biases in Genetic Programming with Linear Representations," *Proc. IEEE Congress on Evolutionary Computation (CEC '01),* pp. 1078-1085, 2001.

[32] P. Martin and R. Poli, "Analysis of the Behavior of Hardware Implementation of GP Using FPGA and Handel-C," Technical Report CSM 357, Univ. of Essex, U.K., 2002.

[33] N. Krasnogor, "Toward Robust Memetic Algorithms," *Recent Advances in Memetic Algorithms,* pp. 185-207, 2004.

[34] P. Merz, "Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies," PhD thesis, Parallel Systems Research Group, Dept. of Electrical Eng. and Computer Science, Univ. of Siegen, 2000.

[35] N. Krasnogor and J. Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues," *IEEE Trans. Evolutionary Computation,* vol. 9, pp. 474-488, 2005.

[36] D.S. Callaway and A.S. Perelson, "HIV-1 Infection and Low Steady State Viral Loads," *Bull. Math. Biology,* vol. 64, no. 1, pp. 29-64, 2002.

[37] T.-W. Chun, R.T. Davey Jr., D. Engel, H.C. Lane, and A.S. Fauci, "AIDS: Re-Emergence of HIV after Stopping Therapy," *Nature,* vol. 401, pp. 874-875, 1999.

[38] F. Garcia, M. Plana, C. Vidal, A. Cruceta, W.A. O'Brien, G. Pantaleo, T. Pumarola, T. Gallart, J.M. Miro, and J.M. Gatell, "Dynamics of Viral Load Rebound and Immunological Changes after Stopping Effective Antiretroviral Therapy," *AIDS,* vol. 13, pp. F79-F86, 1999.

[39] J. Lisziewicz, E. Rosenberg, J. Lieberman, H. Jessen, L. Lopalco, R. Siliciano, B. Walker, and F. Lori, "Control of HIV Despite the Discontinuation of Antiretroviral Therapy," *New England J. Medicine,* vol. 340, no. 21, pp. 1683-1684, 1999.

[40] B.M. Adams, H.T. Banks, M. Davidian, and E.S. Rosenberg, "Model Fitting and Prediction with HIV Treatment Interruption Data," Technical Report CRSC-TR05-40, Center for Research in Scientific Computation, North Carolina State Univ., 2005.

[41] B.M. Adams, "Non-Parametric Parameter Estimation and Clinical Data Fitting with a Model of HIV Infection," PhD dissertation, Dept. of Math., North Carolina State Univ., 2005.

[42] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations, II, Stiff and Differential-Algebraic Problems,* second ed. Springer-Verlag, 1996.

[43] Y.S. Ong, M.H. Lim, N. Zhu, and K.W. Wong, "Classification of Adaptive Memetic Algorithms: A Comparative Study," *IEEE Trans. Systems, Man and Cybernetics—Part B,* vol. 36, no. 1, pp. 141-152, 2006.

[44] W.E. Hart, N. Krasnogor, and J.E. Smith, "Memetic Evolutionary Algorithms," *Recent Advances in Memetic Algorithms,* pp. 3-27, 2004.

[45] H. Maaranen, K. Miettinen, and M. Mäkelä, "Using Quasi Random Sequences in Genetic Algorithms," *Optimization and Inverse Problems in Electromagnetism,* pp. 33-44, 2003.

[46] A.E. Eiben and J.E. Smith, *Introduction to Evolutionary Computation.* Springer-Verlag, 2003.

[47] D. Whitley, "The Genitor Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials Is Best," *Proc. Third Int'l Conf. Genetic Algorithms,* pp. 116-121, 1989.

[48] J.E. Baker, "Adaptive Selection Methods for Genetic Algorithms," *Proc. First Int'l Conf. Genetic Algorithms and Their Applications,* pp. 101-111, 1985.

[49] J.E. Baker, "Reducing Bias and Inefficiency in the Selection Algorithm," *Proc. Second Int'l Conf. Genetic Algorithms,* pp. 14-21, 1987.
[50] W.M. Spears, "The Role of the Mutation and Recombination in Evolutionary Akgorithms," PhD thesis, George Mason Univ., Fairfax, Va., 1998.
[51] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* The MIT Press, Apr. 1992.
[52] L.J. Fogel, *Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming.* Wiley-Interscience, July 1999.
[53] H.P. Schwefel, *System Evolution and Optimum Seeking.* Wiley, 1995.
[54] L.J. Eshelman, "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination," *Foundations of Genetic Algorithms 1 (FOGA-1),* pp. 263-283, 1991.
[55] D.K. Gehlhaar and D.B. Fogel, "Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking," *Proc. Fifth Ann. Conf. Evolutionary Programming,* pp. 419-429, 1996.
[56] N. Krasnogor, B. Blackburne, E. Burke, and J. Hirst, "Multimeme Algorithms for Proteine Structure Prediction," *Proc. Parallel Problem Solving in Nature VII,* 2002.
[57] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flow Shop Scheduling," *IEEE Trans. Evolutionary Computation,* vol. 7, no. 2, pp. 204-223, 2003.
[58] Y.S. Ong and A.J. Keane, "Meta-Lamarkian Learning in Memetic Algorithms," *IEEE Trans. Evolutionary Computation,* vol. 8, no. 2, pp. 99-110, 2004.
[59] J.C. Spall, *Introduction to Stochastic Search and Optimization.* J. Wiley and Sons, 2003.
[60] A.E. Eiben and J.E. Smith, "Hybrid Evolutionary Algorithms," *Introduction to Evolutionary Computing, Hybridisation with Other Techniques: Memetic Algorithms,* 2003.
[61] R. Hooke and T.A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems," *J. ACM,* vol. 8, pp. 212-229, Mar. 1961.
[62] F. Kaupe Jr., "Algorithm 178: Direct Search," *Comm. ACM,* vol. 6, no. 6, pp. 313-314, June 1963.
[63] C.T. Kelley, *Iterative Methods of Optimization,* pp. 212-229. SIAM, 1999.
[64] S. Kirkpatrick, C.D.J. Gelatt, and M.P. Vecchi, "Optimization by Simulated Annealing," *Science,* no. 220, pp. 671-680, 1983.
[65] V. Cerny, "A Thermodynamical Aprroach to the Traveling Salesman Problem," *J. Optimization, Theory and Application,* vol. 45, no. 1, pp. 41-51, 1985.
[66] H. Szu and R. Hartley, "Fast Simulated Annealing," *Physics Letters A,* vol. 122, pp. 157-162, 1987.
[67] A. Caponio, G.L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A Fast Adaptive Memetic Algorithm for On-Line and Off-Line Control Design of PMSM Drives," *IEEE Trans. System Man and Cybernetics—Part B,* vol. 37, no. 1, pp. 28-41, Feb. 2007.
[68] F. Neri, G.L. Cascella, N. Salvatore, A.V. Kononova, and G. Acciani, "Prudent-Daring vs. Tolerant Survivor Selection Schemes in Control Design of Electric Drives," *Proc. Evoworkshops Conf.,* pp. 806-810, 2006.
[69] T. Bäck, A.E. Eiben, and N.A.L.V. der Vaart, "An Empirical Study of Gas without Parameters," *Proc. Sixth Conf. Parallel Problem Solving in Nature,* pp. 315-324, 2000.
[70] J.E. Smith and T. Fogarthy, "Operator and Parameter Adaptation in Genetic Algorithms," *Soft Computing,* vol. 1, no. 2, pp. 81-87, 1997.
[71] A.E. Eiben, R. Hinterding, and Z. Michaelwicz, "Parameter Control," *Evolutionary Computation 2, Advanced Algorithms and Operators,* pp. 170-187, 2000.
[72] T. Bäck, "The Interaction Rate of Mutation Rate, Selection, and Self-Adaptation within a Genetic Algorithm," *Proc. Parallel Problem Solving from Nature (PPSN-II),* pp. 85-94, 1992.
[73] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS—A Genetic Algorithm with Varying Population Size," *Proc. First IEEE Conf. Evolutionary Computation, 1994 and IEEE World Congress on Computational Intelligence,* vol. 1, pp. 73-78, June 1994.
[74] D. Whitley, "The Genitor Algorithm and Selection Pressure. Why Rank-Based Allocation of Reproductive Trials Is Best," *Proc. Third Int'l Conf. Genetic Algorithms,* pp. 116-121, 1989.
[75] H. Schwefel, *Numerical Optimization of Computer Models.* Wiley, 1981.
[76] I. Rechemberg, *Evolutionstrategie: Optimierung Technisher Systeme nach Prinzipien des Biologishen Evolution.* Fromman-Hozlboog Verlag, 1973.
[77] D.B. Owen, *Handbook of Statistical Tables.* Addison-Wesley, 1962.
[78] S.S. Wilks, "Statistical Prediction with Special Reference to the Problem of Tolerance Limit," *Annals of Math. Statistics,* vol. 13, 1942.

Ferrante Neri received the MS degree in electrical engineering from the Technical University of Bari, Italy, in 2002. In 2003, he worked as an assistant researcher with the Department of Electrotechnics and Electronics, Technical University of Bari, Italy. Since 2004, he has been a PhD student in scientific computing and optimization with the University of Jyväskylä, Finland, and in electrotechnical engineering with the Technical University of Bari, Italy. His research interests include computational intelligence and, in particular, memetic algorithms, evolutionary optimization in the presence of uncertainties, and optimization problems having a computationally expensive fitness function. He is a student member of the IEEE.

Jari Toivanen received the MS and PhD degrees in applied mathematics, both from the University of Jyväskylä, Finland, in 1992 and 1997, respectively. He is an Academy Research Fellow and a docent in scientific computing in the Department of Mathematical Information Technology at the University of Jyväskylä and a visiting researcher at the Center for Research in Scientific Computation at North Carolina State University, Raleigh, North Carolina. His research interests include the development of numerical methods for solving (partial) differential equations and related optimization problems.

Giuseppe Leonardo Cascella received the MS degree with honors and the PhD degree in electrical engineering from the Technical University of Bari, Italy, in 2001 and 2005, respectively. He worked with Getrag GmbH Systemtechnik, St. Georgen, Germany, on an automatic transmission. He is currently an assistant researcher with the Technical University of Bari. His research interests include artificial intelligence for electrical drives, parallel computing, and image processing. He is a member of the IEEE.

Yew-Soon Ong received the BS and MS degrees in electrical and electronics engineering from Nanyang Technology University, Singapore, in 1998 and 1999, respectively. He then joined the Computational Engineering and Design Center, University of Southampton, United Kingdom, where he completed the PhD degree in 2002. He is currently an assistant professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests lie in computational intelligence spanning: memetic algorithms, surrogate-assisted evolutionary design, and grid computing. He has been a guest editor of the *IEEE Transactions on Systems, Man, and Cybernetics*, *Genetic Programming*, the *Evolvable Machines Journal*, and has coedited a volume of *Advances in Natural Computation* (Springer). He is currently editing an upcoming book dedicated to evolutionary computation in dynamic and uncertain environments in the Springer Series on Computational Intelligence. He is also member of the editorial board for the *International Journal of Computational Intelligence* and is an Emergent Technologies technical committee member and Evolutionary Computation in Dynamic and Uncertain Environments technical committee member of the IEEE Computational Intelligence Society. He is a member of the IEEE.