International Journal of Innovative Computing, Information and Control
Volume **9**, Number **7**, July **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 2821–2838

OPTIMIZED DRUG SCHEDULING FOR CANCER CHEMOTHERAPY USING IMPROVED IMMUNE ALGORITHM

Jinn-Tsong Tsai[1], Wen-Hsien Ho[2] and Yao-Mei Chen[3,4,*]

[1]Department of Computer Science
National Pingtung University of Education
No. 4-18, Min-Sheng Road, Pingtung 900, Taiwan
jttsai@mail.npue.edu.tw

[2]Department of Healthcare Administration and Medical Informatics
[3]School of Nursing
Kaohsiung Medical University
No. 100, Shi-Chuan 1st Road, Kaohsiung 807, Taiwan
whho@kmu.edu.tw; *Corresponding author: ymchen@kmu.edu.tw

[4]Department of Nursing
Kaohsiung Municipal Hsiao-Kang Hospital
No. 482, Shan-Ming Road, Kaohsiung 812, Taiwan

Received May 2012; revised September 2012

Abstract. *Taguchi immune algorithm (TIA) is proposed for optimizing multi-dose drug schedules, treatment periods, and drug toxicities in cancer chemotherapy. The objective of the algorithm is to find the drug schedules that minimize the number of tumor cells for a given treatment period under the constraints of drug concentration and drug toxicity. The algorithm developed in this study is applicable for scheduling multi-dose drug cancer chemotherapy for a fixed treatment period. Cancer chemotherapy drug schedules generated for different treatment durations and different drug toxicities are also analyzed. The TIA is used to solve multi-dose drug scheduling problems classified as high dimensional, multimodal, and nonlinear optimization problems. The TIA combines the use of Taguchi method for exploiting the optimal solution in micro-space with the use of artificial immune algorithm (AIA) for exploring the optimal feasible region in macro-space. Therefore, the advantage of the TIA is the capability of both exploration and exploitation. Applications of the TIA to solving the problem of multi-dose drug scheduling for cancer chemotherapy confirm its effectiveness for optimizing drug schedules. The experimental simulation results show that the TIA can help physicians select efficient drug schedules for cancer chemotherapy. The simulations also show that cumulative drug toxicity is an important factor in the reduction of tumor cells.*

Keywords: Drug scheduling problem, Cancer chemotherapy, Immune algorithm, Taguchi method

1. Introduction.

Systematic medical treatment is currently the standard cancer chemotherapy regimen. The aim of cancer treatment is to decrease cancerous cells, even if spread of tumors in the body gradually continues. However, because of their high toxicity, drugs used for chemotherapy can damage normal cells in tissues or organs throughout the body as they circulate in the bloodstream. Therefore, two objectives must be considered when selecting drug dosages prescribed for cancer patients undergoing chemotherapy. One is to consistently limit the maximum concentration of a drug in the body. The other is to limit the cumulative toxicity of a drug to the maximal tolerable level during the treatment period. The constraint of drug toxicity is a major concern when developing a cancer chemotherapy model. Many mathematical models have been developed for predicting tumor growth and for limiting the toxicity of cancer chemotherapy drugs [1-17]. A low number of tumor cells remaining after a fixed treatment period can be used as an indicator of an effective cancer chemotherapy regimen whereas a high number of drug-resistant cells remaining after the treatment can be used as an indicator of chemotherapeutic failure. Therefore, the drug delivery method, the drug resistance of the patient, and the metabolism and toxicity of the drug are important considerations when planning cancer chemotherapy. The first drug schedules for cancer treatment that considered constraints such as the drug resistance of the patient and the toxicity of the anticancer drug were based on models first developed by Martin [5,6]. However, experiments by Liang et al. [15-17] later showed two major problems with these models: the unreasonable timing of the first treatment and the application of three-point constraints, which do not improve the efficiency of drugs used for cancer treatment. The model developed by Liang et al. [15-17] considered the ability of the body to recover from the effects of anticancer drugs and successively overcame the two limitations of the Martin model. In addition to exploring models for scheduling delivery of drugs of varying toxicity and metabolism, researchers have proposed various approaches to solving the problem of optimizing schedules for delivering cancer chemotherapy drugs. For example, Martin [6] used an established numerical solution technique known as control parameterization to solve drug scheduling problems encountered in non-linear programming. Bojkov et al. [18] employed an intuitive direct search optimization procedure to solve the problems. Luus et al. [19] obtained improved solutions to drug scheduling problems by using direct search optimization based on random numbers and search region contraction, which avoids the need to narrow the search space according to intuition alone. Tan et al. [12] proposed paladin-distributed evolutionary algorithms solving drug scheduling optimization problem. Floares et al. [14] applied neural network methodology to optimizing solutions for complex cancer chemotherapy problems. Liang et al. [15-17] proposed an adaptive elitist population-based genetic algorithm for constructing drug scheduling models, which is a multimodal problem involving several discontinuous subregions. Based on the above developments, recent artificial intelligence methods and evolutionary algorithms have attempted to construct drug scheduling models for cancer chemotherapy by using differential equations under the constraints of drug delivery, drug concentration, and cumulative drug toxicity.

This study developed and evaluated a method of optimizing drug schedules for cancer chemotherapy that combines the modified drug scheduling models of cancer chemotherapy developed by Liang et al. [15-17] with the Taguchi immune algorithm (TIA) developed earlier by the lead author in Tsai et al. [20-22]. The purpose was to maximize the efficiency of a drug schedule for a specified period of cancer chemotherapy treatment given the specific conditions of the patient. To enhance the exploration and exploitation capabilities of the TIA, the use of clonal proliferation [23] incorporated into hypermutation [24] of several antibody diversifications is combined with the use of Taguchi recombination for local search. The systematic reasoning capability of the Taguchi method [25-30] is exploited for use in selecting improved genes during the recombination operation. Therefore, clonal proliferation within hypermutation combined with Taguchi method obtains a TIA with better robustness and faster convergence compared with the conventional artificial immune algorithm (AIA).

The paper is organized as follows. Section 2 gives the relevant works and methods. The TIA for solving cancer chemotherapy drug scheduling problems is described in Section 3. In Section 4, the experimental results and discussions are provided. Finally, Section 5 offers some conclusions.

2. **Relevant Works and Methods.** In recent years, many mathematical cancer models have been developed for predicting tumor growth and for minimizing drug toxicity during cancer chemotherapy. The most important works and methods are described below. Among the most important works is Martin [5,6], who used the following differential equations to describe drug scheduling models for cancer chemotherapy treatment:

$$\frac{dx_1}{dt} = -\lambda x_1 + k(x_2 - \beta)H(x_2 - \beta), \tag{2.1}$$

$$\frac{dx_2}{dt} = u - \gamma x_2, \tag{2.2}$$

$$\frac{dx_3}{dt} = x_2, \tag{2.3}$$

where the initial state $x^T(0) = [\ln(100)\ 0\ 0]$. The variable x_1 is a transformed variable that is inversely related to the tumor mass N, where $N = 10^{12}\exp(-x_1)$ cells and the initial tumor cell population is set to 10^{10} cells. The variable x_2 is the drug concentration in the body in drug units (D). The variable x_3 is the cumulative drug toxicity in the body. In Equation (2.1), which describes the net change in the tumor cell population per unit of time, $-\lambda x_1$, the first on the right side of the equal sign, represents the increase in cells due to cell proliferation, and the second term, $k(x_2 - \beta)H(x_2 - \beta)$, represents the decrease in cells resulting from use of the drug. Parameter λ, $\lambda = 9.9 \times 10^{-4}$ day^{-1}, is a constant related to the growth function. The k, $k = 8.4 \times 10^{-3}$ day$^{-1}D^{-1}$, is the proportion of tumor cells killed per unit of chemotherapy drug delivered per unit of time. The H is the Heaviside unit function, shown in Equation (2.4), where β is a threshold drug concentration level equal to 10 D. The implication of the function in Equation (2.4) is that the drug is inefficient if the number of tumor cells killed is smaller than the number of tumor cells that reproduce.

$$H(x_2 - \beta) = \begin{cases} 1 & \text{if } x_2 \geq \beta \\ 0 & \text{if } x_2 < \beta \end{cases} \tag{2.4}$$

Equation (2.2) depicts the net increase in the drug concentration at the cancer site. The u and γ are the rate of drug delivery and the biochemical character parameter of the drug, respectively, where $u \geq 0$ and $\gamma = 0.27$ day^{-1}. Equation (2.3) describes the cumulative drug toxicity integral to the drug concentration over the period of exposure. Performance index I, which indicates the maximum effectiveness of the cancer treatment, is defined as

$$I = x_1(t_f), \tag{2.5}$$

where the final time $t_f = 84$ days. The control optimization is performed subject to constraints on drug delivery, $u \geq 0$, on drug concentration, $x_2 \leq 50$, and on cumulative drug toxicity, $x_3 \leq 2.1 \times 10^3$.

Earlier works by Skipper [1], Crowther [31], and Goldie and Coldman [32] proposed that a major cause of failed chemotherapy is drug resistance since drug-resistant cells can increase as the tumor burden increases. They therefore proposed the following three-point constraint on cancer chemotherapy to achieve a tumor size reduction of least 50 percent every three weeks and to minimize the potential emergence of drug-resistant cells: $x_1(21) \geq \ln(200)$, $x_1(42) \geq \ln(400)$, and $x_1(63) \geq \ln(800)$ [5,6,12,14,18,19]. Under this three-point constraint, the performance index I obtained by Martin [6] was 16.836, which corresponded to a final tumor size of $N = 4.878 \times 10^4$ cells. In following studies, Bojkov et al. [18] obtained a value of 17.223 ($N = 3.31 \times 10^4$ cells), Luus et al. [19] obtained 17.476 ($N = 2.57 \times 10^4$ cells), Tan et al. [12] obtained 17.472 ($N = 2.58 \times 10^4$ cells), and Floares et al. [14] obtained 18.22 ($N = 1.22 \times 10^4$ cells).

However, some researchers have proposed that the three-point constraint cannot improve the overall efficiency of cancer treatment [12,15-17,33,34]. Such studies simplify the optimal drug scheduling problem by excluding the three-point constraint (i.e., $x_1(21) \geq \ln(200)$, $x_1(42) \geq \ln(400)$, and $x_1(63) \geq \ln(800)$) and considering only the drug toxicity constraint. This line of research includes Carrasco and Banga [33], who obtained a performance index I of 17.742, which corresponds to a final tumor size of $N = 1.97 \times 10^4$ cells, and Luus [34] and Tan et al. [12], who obtained an index of 17.993, which corresponds to $N = 1.534 \times 10^4$ cells.

An important underlying problem in the cancer drug scheduling model developed by Martin [5,6], which is represented by Equations (2.1)-(2.3), was considered in works by Liang et al. [15-17]. Equation (2.1) describes the efficiency of a drug schedule for a given treatment period. Equation (2.2) obtains the change in the concentration of a drug in the body. In Equation (2.3), however, variable x_3 never decreases throughout the entire cancer treatment because drug concentration x_2 on the right side of the equation is never smaller than 0. Equation (2.3) does not consider metabolism of the drug in the body. The cumulative drug toxicity x_3 decreases because drug concentration x_2 decreases through metabolism and through clearance by the liver and kidney in the body and tends to decrease to 0. To correct these deficiencies and to obtain an accurate description of the metabolic process of cumulative drug toxicity x_3 in the body, Equation (2.3) was modified as follows.

$$\frac{dx_3}{dt} = x_2 - \eta x_3, \tag{2.6}$$

where η is a positive constant representing the rate at which drug toxicity is eliminated. On the right side of Equation (2.6), the first term x_2 describes the cumulative increase in drug toxicity x_3 due to drug concentration x_2, and the second term $-\eta x_3$ describes the cumulative decrease in the toxicity of a drug as it is metabolized in the body. Therefore, the drug scheduling model is modified by replacing Equation (2.3) with Equation (2.6). Additionally, based on the value of 0.4 for parameter η and on the analyses of maximal cumulative drug toxicity, the new constraint on x_3 is $x_3 \leq 100$ rather than $x_3 \leq 2.1 \times 10^3$.

3. **TIA for Solving Drug Scheduling Problems.** This section introduces the TIA and applies it to solve the problem of optimizing multi-dose drug schedules. Experiments performed in earlier applications of TIA in optimization problems [20-22] confirm its outstanding performance in terms of efficiency and solution quality. The details of such an application are as follows.

1) Antibody (variable) representation

Modeling a drug schedule is a high dimensional and multimodal optimization problem. Several schemes for representing drug dose variables have been studied. One example is Tan et al. [12], who used a pair-wise variable representation to reduce the complexity of variables. For example, a drug schedule starting from day 8 at a drug dosage level of 10.5 D was represented by $(10.5, 8)$. This variable representation is used for the treatment period as the given doses do not change frequently. In clinical practice, however, the drug schedule is generally a repeated policy, which requires a more complex pair-wise variable representation. A cycle-wise variable representation was proposed by Liang et al. [15-17] to describe drug doses in the drug scheduling problem. The cycle-wise variable representation includes two parts: a front part for expressing drug doses in the initial days of the treatment, and a cyclic part that describes the number of cycles after the initial treatment. For example, the cycle-wise variable representation $\left\{57.05, 27 \times \overline{21.5, 2 \times \overline{0}}\right\}$ indicates a drug dose of 57.05 D delivered on the first day followed by 27 drug doses of 21.5 D delivered once every three days thereafter. Since cycle-wise variable representation has proven suitable and sufficiently flexible for drug scheduling problems, the concept of the cycle-wise variable representation was applied here for use in arranging drug schedules.

2) Initial population

For each drug dose schedule in the cycle-wise variable representation, the real coding representation used to represent the continuous numeric variable in the drug scheduling problem encodes each antibody as a vector of floating-point numbers that have the same length as the vector for the design variables. For example, the drug schedule for the 84-day treatment period is defined as $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (78 \times \overline{u_5})\}$. The vector of the design variables is $(u_1, u_2, u_3, u_4, u_5)$. Therefore, a general vector $u = (u_1, u_2, \ldots, u_i, \ldots, u_n)$ is used as an antibody (the design variables) to represent a solution to the drug scheduling problem. The initialization procedure uses the following algorithm to produce p_s antibodies, where p_s denotes the antibody population size.

Algorithm 1:

Step 1: Generate a uniformly distributed random value β, where $\beta \in [0, 1]$.

Step 2: Let $u_i = u_i^L + \beta(u_i^U - u_i^L)$, where u_i^L and u_i^U are the lower and upper bounds of u_i, respectively. Repeat n times, and produce a vector $(u_1, u_2, \ldots, u_n)$.

Step 3: Repeat the above two steps p_s times, and produce p_s initial feasible solutions.

3) Clonal proliferation within hypermutation

Based on the biological immune principles, the selection and mutation events in B-cell clonal proliferation processes allow these lymphocytes to increase their receptor diversity and to improve their capability to recognize selective antigens (i.e., the events increase their affinities with selective antigens). In this study, the drug scheduling optimization problem uses convex combination [35,36] as the hypermutation mechanism. Antibodies that join in clonal proliferation are selected from the antibody population pool according to the clonal selection rate p_c, where p_c is the probability of joining the clonal proliferation of an antibody. Each gene in each antibody performs a hypermutation of a convex combination at hypermutation rate p_h. When solving the drug scheduling problem, for a given antibody $u = (u_1, u_2, \ldots, u_i, u_j, u_k, \ldots, u_n)$, if the element u_i is determined to perform the hypermutation and the other u_k is randomly selected to join in, the resulting offspring antibody becomes $u' = (u_1, u_2, \ldots, u'_i, u_j, u_k, \ldots, u_n)$, where the new gene u'_i is $u'_i = (1 - \beta)u_i + \beta u_k$, β is a random value, and $\beta \in [0, 1]$. For example, the drug schedule is defined to be $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (78 \times \overline{u_5})\} = \left\{56.5, (3 \times \overline{12.8}), 3.2, 9.5, (78 \times \overline{10.5})\right\}$ such that the given antibody is $u = (56.5, 12.8, 3.2, 9.5, 10.5)$. If the element 12.8 ($u_2$) is selected to perform the hypermutation, the other element 9.5 (u_4) is randomly selected to join in, and $\beta = 0.3$. The resulting antibody becomes $u' = (56.5, 11.81, 3.2, 9.5, 10.5)$, where the new gene 11.81 (u'_2) is derived from $(1 - 0.3) \times 12.8 + 0.3 \times 9.5$.

4) Recombination by Taguchi method

The orthogonal arrays of the Taguchi method can be used to study a large number of design variables with a small number of experiments and can screen factors that have important effects on the performance of the design. The design variables (parameters) are called factors, and parameter settings are called levels. The details of the Taguchi method can be found in works by Phadke [25], Montgomery [26], and Park [27]. The two-level orthogonal array here has Q factors, where Q is the number of design factors (variables) and each factor has two levels. To establish an orthogonal array of Q factors with two levels, let $L_n(2^{n-1})$ represent $n - 1$ columns and n individual experiments corresponding to the n rows, where $n = 2^k$, k is a positive integer ($k > 1$), and $Q \leq n - 1$. If $Q < n-1$, only the front Q columns are used while the other $n - 1 - Q$ columns are ignored. For example, if the array has six factors with two levels for each factor, six columns are needed to allocate the factors, so $L_8(2^7)$ is sufficient for this purpose since it has seven columns.

Additionally, the better combinations of design variables are determined by integrating the orthogonal array and the signal-to-noise ratio of the Taguchi method. An important underlying concept of this method is to maximize the signal-to-noise ratio (η) performance measure by using the orthogonal array to run a partial set of experiments. The η refers to the mean-square-deviation of the objective function. The Taguchi definitions for the larger-the-better characteristic and for the smaller-the-better characteristic are $\eta = -10\log\left(\frac{1}{n}\sum_{t=1}^{n}\frac{1}{y_t^2}\right)$ and $\eta = -10\log\left(\frac{1}{n}\sum_{t=1}^{n}y_t^2\right)$, respectively which are measured in decibels, where $\{y_1, y_2, \ldots, y_n\}$ denotes a set of characteristics. Further details can be found in earlier works presented by Phadke [25], Montgomery [26], and Park [27].

Since only the degree of η is described in the orthogonal array experiments performed in this study, the above equations were modified as $\eta_i = (y_i)^2$ or $(1/y_i)^2$ to maximize the objective function (larger-the-better) or to minimize the objective function (smaller-the-better), respectively. Let y_i denote the evaluation value of the objective function of experiment i, where $i = 1, 2, \ldots, n$, and n is the number of experiments. The effects of the various factors (variables or genes) can be defined as follows:

$$E_{fl} = \text{ sum of } \eta_i \text{ for factor } f \text{ at level } l, \tag{3.1}$$

where i is the experiment number, f is the factor name, and l is the level number.

The somatic recombination concept of a partial crossover between two randomly selected light chains [37,38] is applied in the recombination procedure. Since the objective is to propose an effective problem-solving algorithm rather than to model a biological phenomenon, the somatic recombination concept is modified by using two antibodies instead of two light chains. That is, one antibody donates its genes to the other one. Additionally, since the random donation mechanism used in the conventional evolutionary algorithm performs poorly in optimization problems, improving the design of the donation mechanism is crucial improving for the overall performance of the evolutionary algorithm.

The main objective of these matrix experiments is to determine whether, at each locus, a gene donated by one antibody replaces a gene of another antibody. One antibody donates its genes to the other since the E_{fl} has the highest value in the experimental region. If $E_{f1} > E_{f2}$, level 1 is the best level for factor $f \in [1, Q]$ in a two-level problem (i.e., the gene of locus f of one antibody is donated to locus f of the other antibody); otherwise, the donation mechanism at locus f is not activated. Activation of the donation mechanism for each gene at each locus f also obtains the new antibody. Thus, the systematic reasoning ability of the Taguchi method ensures that this new antibody has the best or close-to-best evaluation value of the objective function among those of 2^Q combinations of factor level, where 2^Q is the total number of experiments needed for all combinations of factor levels.

The Taguchi method solves drug scheduling optimization problems by performing a matrix experiment for systematic donation of genes from one antibody to another. When the donation mechanism is activated, matrix experiments are performed by the orthogonal array. Antibodies generated by replication and by clonal proliferation are randomly selected two at a time from the antibody pool. The detailed steps for each experiment in the matrix experiments are as follows.

Algorithm 2:

Step 1: Set $j = 1$. Generate two sets U_1 and U_2, each of which has Q design factors (variables), and then use the first Q columns of the orthogonal array $L_n(2^{n-1})$ to allocate the Q design factors, where $n \geq Q + 1$.

Step 2: Form subsets U_1 and U_2 for use as level 1 and level 2, respectively, by assigning two antibodies randomly chosen from the population pool generated by the replication and the clonal proliferation procedures.

Step 3: Assign the level 1 value obtained from U_1 and the level 2 value obtained from U_2 to level cells of the j experiment in the orthogonal array.

Step 4: Calculate the fitness value and signal-to-noise ratio for the new antibody.

Step 5: If $j > n$, then go to Step 6. Otherwise, $j = j + 1$. Return to Step 3, and continue through Step 5.

Step 6: Calculate the effects of the various factors (E_{f1} and E_{f2}), where $f = 1, 2, \ldots, Q$.

Step 7: If $E_{f1} > E_{f2}$, the gene of locus f of the new antibody is from U_1. Otherwise, the gene is from U_2, where $f = 1, 2, \ldots, Q$. Repeat the procedure for each gene at each locus to obtain the new antibody.

5) Mutation

Like the hypermutation mechanism, the mutation operation also uses convex combination to solve the drug scheduling optimization problem [35,36]. Each gene in a single antibody undergoes a convex combination mutation at mutation rate p_m. For a given antibody vector $u = (u_1, u_2, \ldots, u_i, u_j, u_k, \ldots, u_n)$, if element u_i is selected to perform the mutation and if the other u_k is randomly selected to join in, the resulting offspring antibody vector becomes $u' = (u_1, u_2, \ldots, u'_i, u_j, u_k, \ldots, u_n)$, and the new gene u'_i is $u'_i = (1 - \beta)u_i + \beta u_k$, where β is a random value and where $\beta \in [0, 1]$.

6) Penalty function

The penalty function applies to the constraints of drug concentration and cumulative drug toxicity in the drug scheduling problem. The function converts a constrained optimization problem into an unconstrained one by penalizing unfeasible individuals in the population. Compared with the evolutionary algorithm with constraints, the evolutionary method with penalty function is superior for approaching feasible regions of the search space because it navigates through unfeasible regions by reducing the penalty on feasible regions. That is, it limits the search paths to feasible regions. Therefore, for finding feasible individuals in a wide design parameter space in the constrained optimization problem, the evolutionary algorithm with penalty function is better than the evolutionary algorithm with constraints. To clarify this point, it is important to note the distinction between feasible and unfeasible individuals. Each unfeasible individual violates multiple constraints in the range $[1, R]$, where R is the number of design constraints. The higher this index is, the larger the penalty should be. Thus, penalty value P is defined as follows:

$$P = w_p \sum_{j=1}^{R} |w_L(L_j - y_j) + w_U(y_j - U_j)|, \tag{3.2}$$

where y_j is a calculated value depending on the function of the design constraint; U_j and L_j are the upper and lower bounds, respectively, of the function of the design constraint; w_p is a value used to distinguish between feasible and unfeasible individuals; and w_L and w_U denote the weights. Additionally, if $y_j < L_j$, then $w_L = 1$ and $w_U = 0$; if $y_j > U_j$, then $w_L = 0$ and $w_U = 1$; if $L_j \leq y_j \leq U_j$, then $w_L = 0$ and $w_U = 0$.

7) Steps of the TIA

The steps for establishing the TIA, which are based on artificial immune principles and on the Taguchi method, are as follows.

Step 1: Set parameters: antibody population size p_s, replication rate p_r, clonal selection rate p_c, hypermutation rate p_h, and mutation rate p_m.

Step 2: Use Algorithm 1 to generate the initial antibody population, and use the performance index and penalty function to calculate the fitness value of each antibody.

Step 3: Perform selection by roulette wheel approach [39].

Step 4: Perform clonal proliferation within the hypermutation. From the antibody population pool, select the antibody to join in clonal proliferation according to clonal selection rate p_c. For each gene in a single antibody, perform hypermutation of convex combination at hypermutation rate p_h.

Step 5: Reproduce antibodies at replication rate p_r. Apply Taguchi method to recombining antibodies from the replication with antibodies from the clonal proliferation.

Step 6: Perform Algorithm 2 (using matrix experiments and signal-to-noise ratios) to generate improved offspring.

Step 7: Repeat Step 6 until the loop number $\left(\frac{1}{4} \times p_s \times p_r\right)$ has been met.

Step 8: The antibody population from the Taguchi method is generated.

Step 9: Perform the mutation operation at mutation rate p_m.

Step 10: New antibody population is generated.

Step 11: For the antibody population of the last generation and for the new antibody population of the current generation, sort the fitness values in increasing order.

Step 12: Select the best p_s antibodies as the new antibody population of the next generation.

Step 13: If the stopping criterion has been met, go to Step 14. Otherwise, return to Step 3, and continue through Step 13.

Step 14: Display the best antibody and fitness value.

4. **Experimental Results and Discussion.** The models used for cancer chemotherapy drug scheduling in this study were modifications of the models developed earlier by Liang et al. [15-17]. However, the same parameter values were used. The differential equations, parameters and constraints are as follows:

$$\frac{dx_1}{dt} = -\lambda x_1 + k(x_2 - \beta)H(x_2 - \beta), \tag{4.1}$$

$$\frac{dx_2}{dt} = u - \gamma x_2, \tag{4.2}$$

$$\frac{dx_3}{dt} = x_2 - \eta x_3, \tag{4.3}$$

where the initial state $x^T(0) = [\ln(100)\ 0\ 0]$. The parameters are $\lambda = 9.9 \times 10^{-4}$ day^{-1}, $k = 8.4 \times 10^{-3}$ day$^{-1}D^{-1}$, $\beta = 10\ D$, $\gamma = 0.27$ day^{-1}, $\eta = 0.4$ day^{-1}, and

$$H(x_2 - \beta) = \begin{cases} 1 & \text{if } x_2 \geq \beta \\ 0 & \text{if } x_2 < \beta \end{cases}. \tag{4.4}$$

The control optimization is performed subject to constraints on drug delivery, $u \geq 0$, on drug concentration, $x_2 \leq 50$, and on cumulative drug toxicity, $x_3 \leq 100$. For each cancer treatment, the performance index I to be maximized is

$$I = x_1(t_f), \tag{4.5}$$

where t_f is the final time. Here, t_f is defined as 84 days. This study compared multi-dose drug schedules for treatment periods of 84, 70, 60, 50, and 49 days. Fitness function F, which is used to convert the optimization problem from a constrained problem to an unconstrained problem, is defined as

$$F = \frac{1}{I} + P, \tag{4.6}$$

where P is the penalty value in Equation (3.2). Equation (4.6) represents the smaller-the-better strategy of finding the optimal drug schedule. According to this equation, a smaller fitness value implies a larger performance index I and $P = 0$ (i.e., $x_2 \leq 50$ and $x_3 \leq 100$).

To assist physicians in designing clinical cancer chemotherapy drug treatments based on specific patient conditions, this study explored drug schedules for different treatment periods. Drug scheduling patterns can generally be classified as continuous (drug delivery on every day of the treatment period) or repeated (drug deliveries separated by specified durations during the treatment period) [15,16]. Details of the application of these scheduling patterns in this study are given below.

Two continuous drug scheduling patterns were used in the 84-day treatment period:

1. $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (78 \times \overline{u_5})\}$, daily drug delivery at five different doses.
2. $\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, (70 \times \overline{u_{15}})\}$, daily drug delivery at 15 different doses.

Eight different repeated-type scheduling patterns were used in the 84-day treatment period:

1. $\left\{u_1, (3 \times \overline{u_2}), 0, u_3, (39 \times \overline{0, u_4})\right\}$, four different doses delivered on 44 days (1 day off after each delivery except for front 4 days).
2. $\{u_1, 0, u_2, 0, u_3, 0, u_4, (38 \times \overline{0, u_5}), u_6\}$, six different doses delivered on 43 days (1 day off after each delivery).
3. $\left\{u_1, (3 \times \overline{u_2}), 0, 0, \left(26 \times \overline{u_3, (2 \times \overline{0})}\right)\right\}$, three different doses delivered on 30 days (2 days off after each delivery except for front 4 days).
4. $\left\{u_1, (2 \times \overline{0}), u_2, (2 \times \overline{0}), u_3, \left(25 \times \overline{(2 \times \overline{0}), u_4}\right), 0, u_5\right\}$, five different doses delivered on 29 days (2 days off after each delivery).
5. $\left\{u_1, (3 \times \overline{u_2}), \left(20 \times \overline{(3 \times \overline{0}), u_3}\right)\right\}$, three different doses delivered on 24 days (3 days off after each delivery except for front 4 days).
6. $\left\{u_1, (3 \times \overline{0}), u_2, (3 \times \overline{0}), u_3, \left(18 \times \overline{(3 \times \overline{0}), u_4}\right), 0, 0, u_5\right\}$, five different doses delivered on 22 days (3 days off after each delivery).
7. $\left\{u_1, (4 \times \overline{0}), u_2, (4 \times \overline{0}), u_3, \left(14 \times \overline{(4 \times \overline{0}), u_4}\right), 0, 0, u_5\right\}$, five different doses delivered on 18 days (4 days off after each delivery).
8. $\left\{u_1, \left(13 \times \overline{(5 \times \overline{0}), u_2}\right), (4 \times \overline{0}), u_3\right\}$, three different doses delivered on 15 days (5 days off after each delivery).

This study used TIA with a Runge-Kutta method in Matlab [40] to construct cancer chemotherapy drug scheduling models, which are high dimensional, multimodal, and nonlinear optimization problems described using differential equations with constraints. Optimization was performed in both continuous and repeated cancer drug delivery schedules. Optimizing the main parameters of evolutionary environments continues to be an active area of research in the computer modeling literature. Studies have shown how the performance of a genetic algorithm (GA) [41,42] can be improved by adapting the main parameters used in other methods [43,44]. For example, Chou et al. [45] showed how the performance of a GA can be improved by applying experimental design method to optimize its evolutionary parameters. Therefore, this study applied the experimental design method to adjusting the evolutionary parameters. The computational experiments for the TIA were performed using the following evolutionary parameters: antibody population size p_s of 100, replication rate p_r of 0.8, mutation rate p_m of 0.2, clonal selection rate p_c of 0.2, and hypermutation rate p_h of 0.8. The stopping criterion for the TIA was a fitness value that did not decrease for at least 50 successive generations out of 300 generations. Some drug scheduling problems were also performed in 30 independent runs to test the stability of the solutions. Each of the continuous and repeated drug scheduling problems involved 3 to 15 variables (factors), and one column was used to allocate each factor in an orthogonal array to perform recombination in the TIA. Therefore, 3 to 15 columns were used in each array. The 7-column $L_8(2^7)$ orthogonal array was used if 7 or fewer variables were involved. When performing the matrix experiments, the columns located in the front part of the orthogonal array were selected first, and the unselected columns were ignored. Therefore, the $L_{16}(2^{15})$ orthogonal array was used because it met the criteria of more than 7 variables and fewer than or equal to 15 variables.

For each drug schedule, Table 1 shows the simulation results obtained by the TIA, including the number of drug deliveries, the performance index, and the corresponding final number of tumor cells. For continuous-type drug scheduling, the best performance index values reported for other approaches are $x_1 = 17.742$ [33], $x_1 = 17.993$ [12,34], $x_1 = 18.03$ [46], and $x_1 = 18.22$ [14] for final tumor cell numbers of 1.97×10^4, 1.534×10^4, 1.478×10^4, and 1.22×10^4 cells, respectively. Comparison of the simulation results shows that the best performance index obtained by the TIA ($x_1 = 24.72$; $N = 18$ cells) is better than those obtained by the above methods. For repeated-type drug scheduling, the best performance index values obtained by the approach reported in Liang et al. [16] for policies 2-4 are $x_1 = 24.331$ ($N = 27$ cells), $x_1 = 23.806$ ($N = 46$ cells), and $x_1 = 21.376$ ($N = 521$ cells), respectively. In contrast, those obtained by TIA for policies 1, 3, and 5 (Table 1) are $x_1 = 24.526$ ($N = 22$ cells), $x_1 = 23.841$ ($N = 44$ cells), and $x_1 = 21.377$ ($N = 520$ cells), respectively. Again, the simulation results obtained by the TIA are consistently superior to those obtained by Liang et al. [16].

Figures 1 and 2 show the results for the 84-day continuous-type drug schedules, including changes in values for control variables (u), numbers of tumor cells (N), drug concentrations (x_2), and cumulative drug toxicities (x_3). Figures 3-7 show the representative results for the 84-day repeated-type drug schedules. Table 1 shows that the final

Table 1. Comparison of drug schedules obtained by the TIA in terms of numbers of drug deliveries, performance indexes, and corresponding final number of tumor cells

Continuous-type drug schedules for 84-day treatment periods				
No.	Drug schedule	No. of drug deliveries	Index (x_1)	Final no. of cells
1	$\{57.05, (3 \times \overline{13.5}), 2.309, 10.633, (78 \times \overline{10.8})\}$	84	24.72	18
2	$\{56.95, 8.89, 14.56, 15.42, 5.01, 4.25, 9.47, 20.93, 3.38,$ $13.16, 1.9, 21.09, 8.15, 10.6, (70 \times \overline{10.8})\}$	84	24.629	20
Repeated-type drug schedules for 84-day treatment periods				
No.	Drug schedule	No. of drug deliveries	Index (x_1)	Final no. of cells
1	$\{57.05, (3 \times \overline{13.5}), 0, 15.79, (39 \times \overline{0, 21.49})\}$	44	24.526	22
2	$\{55.18, 0, 24.68, 0, 23.26, 0, 19.15, (38 \times \overline{0, 21.09}), 6.59\}$	43	24.0607	36
3	$\left\{57.03, (3 \times \overline{13.32}), 0, 0, (26 \times \overline{31.36, (2 \times \overline{0})})\right\}$	30	23.8414	44
4	$\{56.17, (2 \times \overline{0}), 31.45, (2 \times \overline{0}), 31.93,$ $(25 \times (\overline{2 \times \overline{0}), 31.14}), 0, 16.3\}$	29	23.4927	63
5	$\left\{57.05, (3 \times \overline{13.502}), (20 \times (\overline{3 \times \overline{0}), 37.68})\right\}$	24	21.377	520
6	$\{56.52, (3 \times \overline{0}), 37.85, (3 \times \overline{0}), 37.68,$ $(18 \times (\overline{3 \times \overline{0}), 37.47}), 0, 0, 31.8\}$	22	21.012	749
7	$\{57.05, (4 \times \overline{0}), 42.26, (4 \times \overline{0}), 42.26,$ $(14 \times (\overline{4} \times \overline{0}), 42.23), 0, 0, 31.69\}$	18	18.8082	6787
8	$\left\{57.05, (13 \times (\overline{5 \times \overline{0}), 45.76}), 4 \times \overline{0}, 42.26\right\}$	15	16.704	55658

number of tumor cells gradually decreases as the number of drug deliveries increases during the 84-day treatment period. The data for drug schedules 1 and 2 in Table 1 and in Figures 1-4 also show that the final number of tumor cells is still small after completing the continuous and repeated versions of drug schedules 1 and 2 but the cumulative drug toxicity approaches the maximal value of 100 on most treatment days. If the treatment goal is applying the drug schedule that is most efficient for reducing the tumor cells to

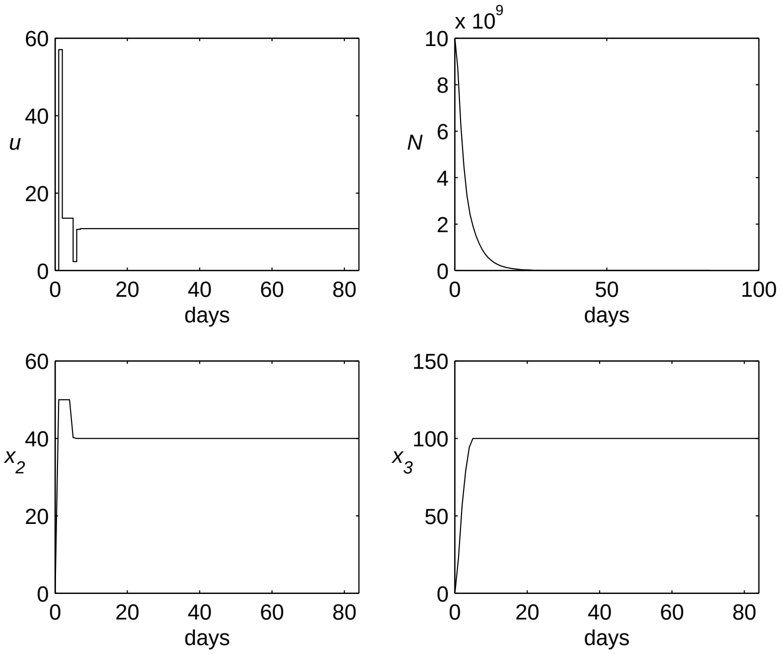

Figure 1. Changes in values for u, N, x_2, and x_3 in continuous-type drug schedule 1

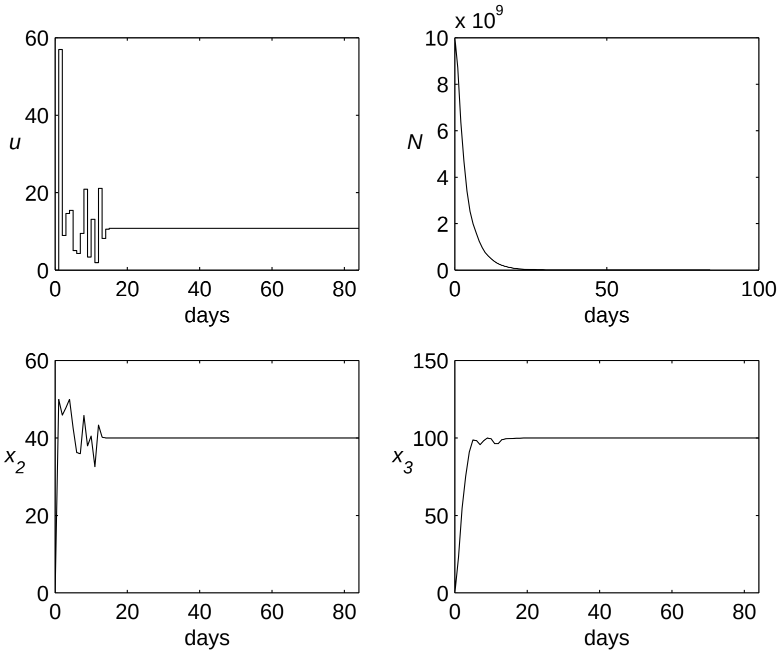

Figure 2. Changes in values for u, N, x_2, and x_3 in continuous-type drug schedule 2

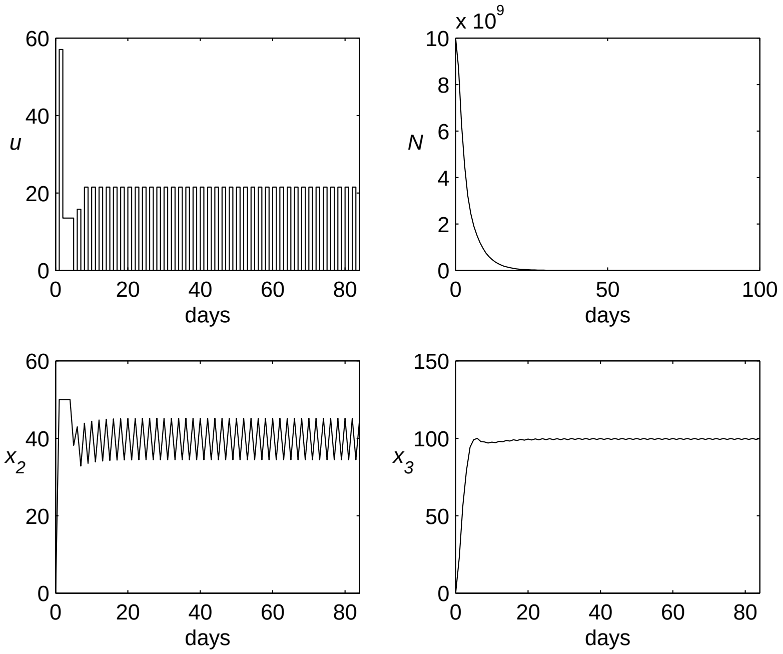

Figure 3. Changes in values for u, N, x_2, and x_3 in repeated-type drug schedule 1

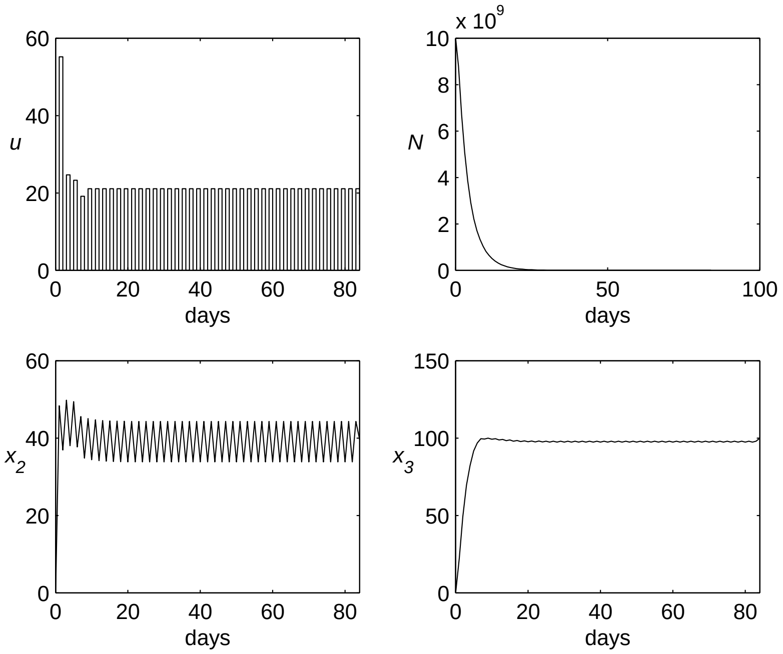

Figure 4. Changes in values for u, N, x_2, and x_3 in repeated-type drug schedule 2

a number approaching zero, the continuous-type drug schedule 1 should be selected even at the cost of increased cumulative drug toxicity. However, if the treatment objective is to reduce the tumor cells to a specified number or to limit cumulative drug toxicity to a specified value, other drug schedules may be preferable, depending on the condition of the patient.

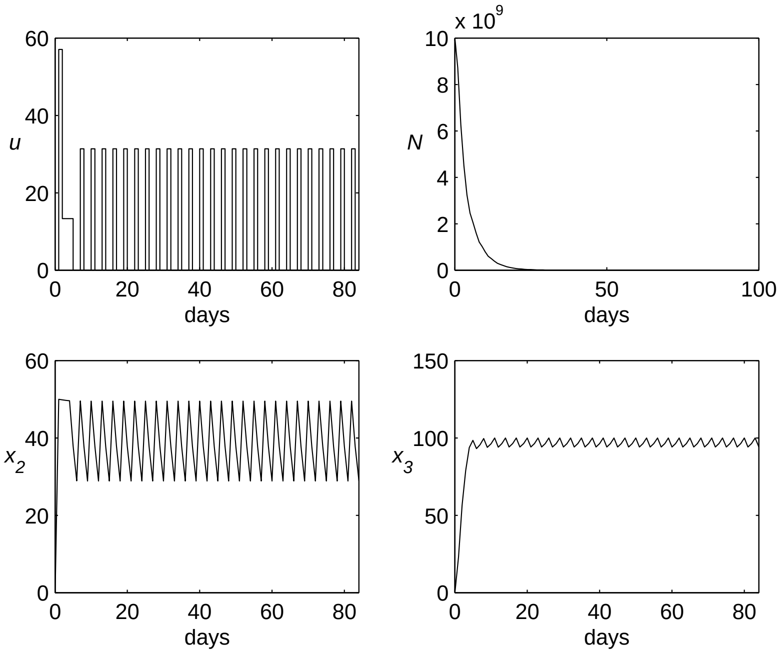

FIGURE 5. Changes in values for u, N, x_2, and x_3 in repeated-type drug schedule 3

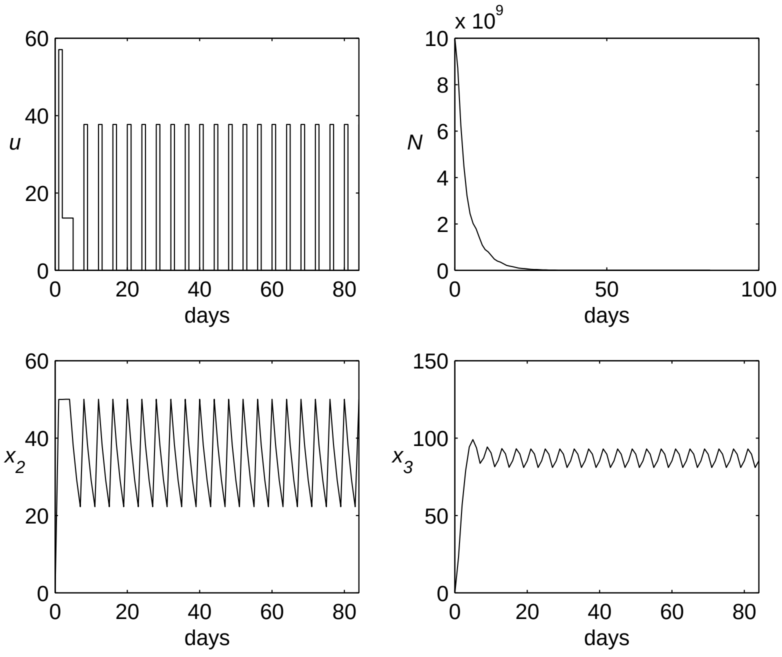

FIGURE 6. Changes in values for u, N, x_2, and x_3 in repeated-type drug schedule 5

Efforts to adjust the drug scheduling model include many clinical trials by Liang et al. [16,17]. A randomized phase II study performed by an oncologist on their research team compared a gemcitabine-cisplatin combination with a gemcitabine-etoposide. They then prospectively collected all drug administration-related data, including dosage, tumor response and toxicity observed in further clinical experiments. Their modified model was

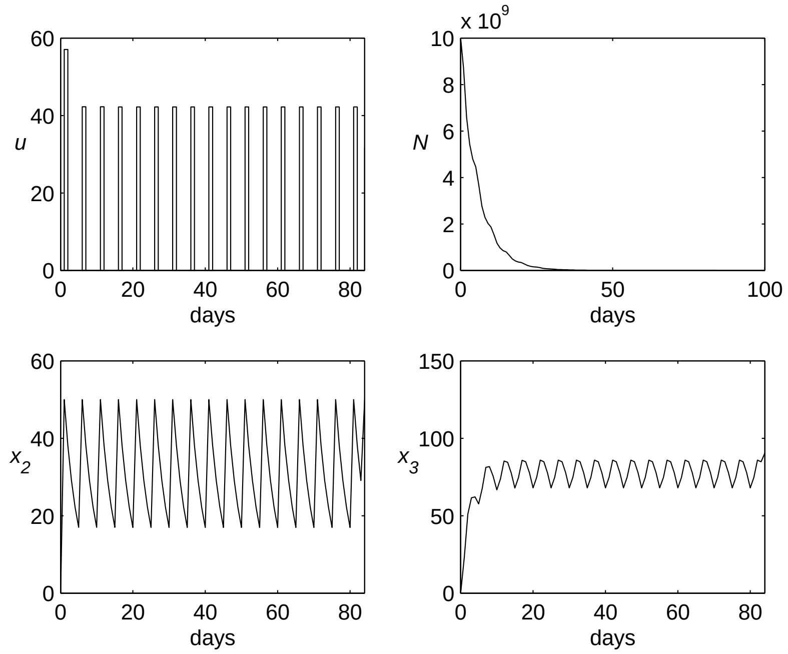

Figure 7. Changes in values for u, N, x_2, and x_3 in repeated-type drug schedule 7

used to test the drug scheduling obtained by the TIA. In the current study, the TIA obtains superior simulation results.

Table 2. Performance of the TIA in solving drug scheduling problems in 30 independent runs

No.	Drug schedule	Best index (x_1)	Mean x_1	Standard deviation in x_1
1	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (78 \times \overline{u_5})\}$	24.72	24.39778	0.37362
2	$\{u_1, (3 \times \overline{u_2}), 0, u_3, (39 \times \overline{0, u_4})\}$	24.526	24.25229	0.27907
3	$\{u_1, (3 \times \overline{u_2}), 0, 0, (26 \times \overline{u_3, (2 \times \overline{0})})\}$	23.8414	23.70808	0.34278
4	$\{u_1, (3 \times \overline{u_2}), (20 \times \overline{(3 \times \overline{0}), u_3})\}$	21.377	20.90986	0.21507

The systematic reasoning mechanism of the orthogonal array with signal-to-noise ratio in the TIA represents an advance in micro space algorithms because it accelerates convergence to the global solution. By contributing to macro real-number space, the AIA enhances the performance of the TIA. Table 2 shows the good performance of the TIA in solving some drug scheduling problems. Notably, the standard deviation in x_1 is quite small in all drug schedules, which indicates that the TIA is a robust method that can obtain stable solutions. Therefore, we conclude that use of the TIA for solving drug scheduling optimization problems is feasible.

Notably, this study and most relevant studies in the literature (e.g., [6,12-19,33,34]), only consider an 84-day treatment period. Therefore, the question is whether the drug scheduling models, parameters, and constraints are appropriate for treatment periods of other durations. In the continuous-type drug scheduling problem considered here, i.e., $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (p \times \overline{u_5})\}$, where p is a positive integer for matching the defined treatment days, treatment periods of 84, 70, 60, 50, and 49 days and constraints of $x_2 \leq 50$ and $x_3 \leq 100$ have been discussed. For these periods, Tables 3 and 4 show the

Table 3. Computational results for continuous-type drug scheduling problems for varying treatment periods of 84, 70, 60, 50, and 49 days

No.	Continuous-type drug schedule $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (p \times \overline{u_5})\}$	Period (days)	Index (x_1)	Final no. of cells
1	$\{57.05, (3 \times \overline{13.5}), 2.309, 10.633, (78 \times \overline{10.8})\}$	84	24.72	18
2	$\{56.64, (3 \times \overline{12.55}), 6.97, 8.48, (64 \times \overline{10.8})\}$	70	21.49064	464
3	$\{56.78, (3 \times \overline{12.98}), 4.98, 9.37, (54 \times \overline{10.8})\}$	60	19.18160	4672
4	$\{55.94, (3 \times \overline{13.62}), 2.92, 10.29, (44 \times \overline{10.8})\}$	50	16.83826	48667
5	$\{56.33, (3 \times \overline{11.84}), 8.33, 8.85, (43 \times \overline{10.77})\}$	49	16.53774	65728

Table 4. Solution performance of the TIA in 30 independent runs of continuous-type drug scheduling problems for different periods

No.	Continuous-type drug schedule $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (p \times \overline{u_5})\}$	Period (days)	Best index (x_1)	Mean x_1	Standard deviation in x_1
1	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (78 \times \overline{u_5})\}$	84	24.72	24.39778	0.37362
2	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (64 \times \overline{u_5})\}$	70	21.49064	21.28579	0.21035
3	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (54 \times \overline{u_5})\}$	60	19.18160	19.06981	0.12201
4	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (44 \times \overline{u_5})\}$	50	16.83826	16.75063	0.14531
5	$\{u_1, (3 \times \overline{u_2}), u_3, u_4, (43 \times \overline{u_5})\}$	49	16.53774	16.49567	0.13463

computational results and the overall performance of the TIA. Table 3 shows that the final number of tumor cells gradually increases as the day of treatment period decreases. Therefore, given the same constraints in drug concentration and cumulative drug toxicity, changing the chemotherapeutic drug dose cannot substantially reduce tumor cells over a treatment period shorter than 84 days. On most treatment days, all drug schedules achieve a cumulative drug toxicity of 100, which approaches the maximal limit. Therefore, in the continuous-type drug scheduling problem, $\{u_1, (3 \times \overline{u_2}), u_3, u_4, (p \times \overline{u_5})\}$, an 84-day treatment period is reasonable under the constraint of a cumulative drug toxicity of $x_3 \leq 100$. The TIA was further applied to solving these problems in 30 independent runs to test whether it obtains stable solutions. Table 4 shows that, for each drug schedule, the standard deviation in x_1 is quite small, which confirms that the obtained solutions are stable and that the drug schedules are feasible.

Other issues that arise are the appropriate concentration and cumulative toxicity of the drug when the treatment objective is to reduce the tumor cells to a number approaching zero and whether the patient can endure the optimal drug concentration and cumulative drug toxicity. The following two drug schedules obtained by the TIA, which achieve a final number of tumor cells approaching zero, are given for reference. Where drug concentration (x_2) and cumulative drug toxicity (x_3) are defined as $x_2 \leq 50$ and $x_3 \leq 125$, respectively: $\left\{57.05, (3 \times \overline{13.5}), 12.441, 14.304, (78 \times \overline{13.495})\right\}$ where the corresponding final number of tumor cells is 0, and $\left\{57.05, (3 \times \overline{13.5}), 0, 23.805, (39 \times \overline{0, 23.801})\right\}$ where the corresponding final number of tumor cells is 1. These two drug schedules are the most efficient policies for a patient that can endure a drug concentration (x_2) and cumulative drug toxicity (x_3) of 50 and 125, respectively, on most treatment days.

The evolutionary algorithm of the TIA has potential applications for solving many problems including continuous- and repeated-type drug schedules used for cancer treatment. Because it applies a systematic reasoning approach, the TIA can search a huge space without prior knowledge. Currently, the TIA is also one of the best optimization methods. The many uses of TIA for solving optimization problems in engineering design include the use of real-coded encoding to design two-dimensional recursive filters [22].

The TIA can also use real-coded and symbol encoding to optimize global numerical problems and to solve job-shop scheduling problems, respectively [21]. A real-coded TIA has also been used to design digital IIR filters [20]. These diverse applications confirm its applicability in many other task domains.

In terms of clinical treatment, drug scheduling policies can generally be classified as continuous or repeated policies. Table 1 shows the two continuous-type and the eight repeated-type drug scheduling policies applied in this study. When the treatment objective is to reduce tumor size with minimum toxicity, repeated-type drug scheduling policies 5-8 are suitable because cumulative drug toxicity is decreased to as low as 60. For other patients, continuous-type drug scheduling policies 1-2 and repeated-type policies 1-2 may be more efficient but with a high toxicity. These data show that, based on the unique conditions of the individual patient, the TIA finds the most efficient drug scheduling policy for cancer treatment.

5. Conclusions.

In the study, modified versions of the drug scheduling models presented by Liang et al. [15-17] were evaluated for effectiveness in optimizing both continuous-type and repeated-type multi-dose drug schedules for cancer chemotherapy. Optimization problems involving different treatment periods and different cumulative drug toxicities were also considered. The significant and novel contributions are addressed as follows. 1) The performance tests confirmed the efficiency and robustness of the proposed TIA, which combines the Taguchi method and the AIA, in solving optimal drug scheduling problems. 2) For an 84-day treatment period, the simulation results show that the final number of tumor cells gradually decreases as the number of drug deliveries increases. Cumulative drug toxicity values, however, approach the maximal value of 100 on most treatment days. 3) The data show that, for all treatment periods (70, 60, 50, and 49 days), the final number of tumor cells gradually increases as the day of treatment period decreases. For treatment periods shorter than 84 days, the final number of tumor cells cannot be substantially changed by increasing the chemotherapeutic drug dose, and the cumulative drug toxicity on most treatment days is 100, which is the maximal limit. 4) An interesting finding of this study is that tumor cells can be almost completely eliminated in patients who can endure high cumulative drug toxicity. Two optimal drug schedules were generated for patients who can endure cumulative drug toxicity (x_3) of 125.

Because this study confirmed the potential contribution of drug scheduling models in the quest for a cancer cure, future works by the authors will consider whether the model can be improved by considering different conditions during the treatment period and by including drug toxicity data obtained from clinical experience and trials.

Acknowledgments. The authors thank G. C. Hong and C. C. Wu for their assistance in implementing the optimization experiments. This work was partially supported by the National Science Council, Taiwan, under grant numbers NSC95-2221-E-153-002, NSC 99-2320-B-037-026-MY2, NSC100-2221-E-153-001, and NSC 101-2320-B-037-022.

REFERENCES

[1] H. E. Skipper, Adjuvant chemotherapy, *Cancer*, vol.41, pp.936-940, 1978.

[2] J. M. Murray, Optimal control of a cancer chemotherapy problem with general growth and loss functions, *Math. Biosci.*, vol.38, pp.273-287, 1990.

[3] J. M. Murray, Some optimal control problems in cancer chemotherapy with a toxicity limit, *Math. Biosci.*, vol.100, pp.49-68, 1990.

[4] R. B. Martin, M. E. Fisher, R. F. Minchin and K. L. Teo, A mathematical model of cancer chemotherapy with an optimal selection of parameters, *Math. Bio.*, vol.99, pp.205-230, 1990.

[5] R. B. Martin, *Optimal Control of Drug Administration in Cancer Chemotherapy*, Ph.D. Thesis, School Comput. Inf. Sci., Univ. Western Australia, Perth, Australia, 1991.

[6] R. B. Martin, Optimal control drug scheduling of cancer chemotherapy, *Automatica*, vol.28, pp.1113-1123, 1992.

[7] R. B. Martin and K. L. Teo, A worst-case optimal parameter selection model of cancer chemotherapy, *IEEE Trans. on Biomedical Engineering*, vol.39, pp.1081-1085, 1992.

[8] R. B. Martin and K. L. Teo, *Optimal Control of Drug Administration in Cancer Chemotherapy*, World Scientific, Singapore, 1994.

[9] M. Marusic, Z. Bajzer, J. P. Freyer and S. Vuk-Pavlovic, Analysis of growth of multicellular tumour spheroids by mathematical models, *Cell Prolif.*, vol.27, pp.73-94, 1994.

[10] T. E. Wheldon, *Mathematical Models in Cancer Research*, Adam Hilger, Philadelphia, PA, 1998.

[11] D. Merkle, M. Middendorf and H. Schmeck, Ant colony optimization for resource-constrained project scheduling, *IEEE Trans. Evol. Comput.*, vol.6, pp.333-346, 2002.

[12] K. C. Tan, E. F. Khor, J. Cai, C. M. Heng and T. H. Lee, Automating the drug scheduling of cancer chemotherapy via evolutionary computation, *Artif. Intell. Med.*, vol.25, pp.169-185, 2002.

[13] K. C. Tan, A. Tay and J. Cai, Design and implementation of a distributed evolutionary computing software, *IEEE Trans. on System, Man, and Cybernetics – Part C: Applications and Reviews*, vol.33, pp.325-338, 2003.

[14] A. Floares, C. Floares, M. Cucu and L. Lazar, Adaptive neural networks control of drug dosage regimens in cancer chemotherapy, *Proc. of the International Joint Conference on Neural Networks*, pp.154-159, 2003.

[15] Y. Liang, K. S. Leung and T. S. K. Mok, Evolutionary drug scheduling model in cancer chemotherapy, *Lecture Notes in Computer Science*, vol.3103, pp.1126-1137, 2004.

[16] Y. Liang, K. S. Leung and T. S. K. Mok, A novel evolutionary drug scheduling model in cancer chemotherapy, *IEEE Trans. on Inf. Technol. Biomed.*, vol.10, pp.237-245, 2006.

[17] Y. Liang, K. S. Leung and T. S. K. Mok, Evolutionary drug scheduling models with different toxicity metabolism in cancer chemotherapy, *Applied Soft Computing*, vol.8, pp.140-149, 2008.

[18] B. Bojkov, R. Hansel and R. Luus, Application of direct search optimization to optimal control problems, *Hung. J. Ind. Chem.*, vol.21, pp.177-185, 1993.

[19] R. Luus, F. Hartig and F. J. Keil, Optimal drug scheduling of cancer chemotherapy by direct search optimization, *Hung. J. Ind. Chem.*, vol.23, pp.55-58, 1995.

[20] J. T. Tsai and J. H. Chou, Design of optimal digital IIR filters by using an improved immune algorithm, *IEEE Trans. on Signal Processing*, vol.54, pp.4582-4596, 2006.

[21] J. T. Tsai, W. H. Ho, J. H. Chou and T. K. Liu, Improved immune algorithm for global numerical optimization and job-shop scheduling problems, *Applied Mathematics and Computation*, vol.194, no.2, pp.406-424, 2007.

[22] J. T. Tsai, W. H. Ho and J. H. Chou, Design of two-dimensional recursive filters by using Taguchi-based immune algorithm, *IET Signal Processing*, vol.2, no.2, pp.110-117, 2008.

[23] N. K. Jerne, The immune system, *Sci. Amer.*, vol.229, pp.52-60, 1973.

[24] R. S. Harris, Q. Kong and N. Maizels, Somatic hypermutation and the three R's: Repair, replication and recombination, *Mutation Research*, vol.436, pp.157-178, 1999.

[25] M. S. Phadke, *Quality Engineering Using Robust Design*, Prentice-Hall, New Jersey, 1989.

[26] D. C. Montgomery, *Design and Analysis of Experiments*, Wiley, New York, 1991.

[27] S. H. Park, *Robust Design and Analysis for Quality Engineering*, Chapman and Hall, London, 1996.

[28] W. H. Ho, J. H. Chou and C. Y. Guo, Parameter identification of chaotic systems using improved differential evolution algorithm, *Nonlinear Dynamics*, vol.61, pp.29-41, 2010.

[29] W. H. Ho and C. S. Chang, Genetic-algorithm-based artificial neural network modeling for platelet transfusion requirements on acute myeloblastic leukemia patients, *Expert Systems with Applications*, vol.38, pp.6319-6323, 2011.

[30] W. H. Ho, J. X. Chen, I N. Lee and H. C. Su, An ANFIS-based model for predicting adequacy of vancomycin regimen using improved genetic algorithm, *Expert Systems with Applications*, vol.38, pp.13050-13056, 2011.

[31] D. Crowther, Rational approach to the chemotherapy of human malignant disease – II, *Br. Med. Journal*, vol.4, pp.216-218, 1974.

[32] J. H. Goldie and A. J. Coldman, A mathematical model for relating the drug sensitivity of tumors of their spontaneous mutation rate, *Can. Treat. Rep.*, vol.63, pp.1727-1733, 1978.

[33] E. F. Carrasco and J. R. Banga, Dynamic optimization of batch reactors using adaptive stochastic algorithms, *Industrial and Engineering Chemistry Research*, vol.36, pp.2252-2261, 1997.

[34] R. Luus, Comment on Dynamic optimization of batch reactors using adaptive stochastic algorithms, *Industrial and Engineering Chemistry Research*, vol.37, pp.305, 1998.

[35] M. Bazaraa, J. Jarvis and H. Sherali, *Linear Programming and Network Flows*, John Wiley and Sons, New York, 1990.

[36] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley and Sons, New York, 1997.

[37] G. C. Luh, C. H. Chueh and W. W. Liu, MOIA: Multi-objective immune algorithm, *Engineering Optimization*, vol.35, pp.143-164, 2003.

[38] G. C. Luh and C. H. Chueh, Multi-objective optimal design of truss structure with immune algorithm, *Computers and Structure*, vol.82, pp.829-844, 2004.

[39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Massachusetts, 1989.

[40] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994.

[41] J. T. Tsai, J. H. Chou and T. K. Liu, Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm, *IEEE Trans. on Neural Networks*, vol.17, pp.69-80, 2006.

[42] J. T. Tsai, T. K. Liu and J. H. Chou, Hybrid Taguchi-genetic algorithm for global numerical optimization, *IEEE Trans. on Evolutionary Computation*, vol.8, pp.365-377, 2004.

[43] L. Davis, Adapting operator probabilities in genetic algorithms, *Proc. of the 1989 International Conference on Genetic Algorithms*, San Mateo, pp.61-69, 1989.

[44] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, *IEEE Trans. on System, Man, and Cybernetics*, vol.16, pp.122-128, 1986.

[45] J. H. Chou, S. H. Chen and J. J. Li, Application of Taguchi-genetic method to design optimal grey-fuzzy controller of a constant turning force system, *J. of Materials Processing Technology*, vol.105, pp.333-343, 2000.

[46] J. R. Banga and E. F. Carrasco, Rebuttal to the comments of rein luus on dynamic optimization of batch reactors using adaptive stochastic algorithms, *Ind. Eng. Chem. Res.*, vol.37, pp.306-307, 1998.