RESEARCH PAPER

Application of a Mixed Optimization Strategy in the Design of a Pharmaceutical Solid Formulation at Laboratory Scale

Mariela Salazar de Saavedra[1,*] and
Ivan I. Saavedra Cuadra[2]

[1]*Mencion de Tecnologia Industrial Farmaceutica,
Facultad de Farmacia, Universidad Central de Venezuela,
Apartado Postal 40109, Caracas 1041, Venezuela*
[2]*Instituto de Mecanica de Fluidos, Facultad de Ingenieria,
Universidad Central de Venezuela, Apartado Postal 47762,
Caracas 1041A, Venezuela*

ABSTRACT

The objective of this work was to develop an optimization strategy for the design of pharmaceutical formulations. The mixed strategy was used to optimize a dry powder blend containing 500 mg of alpha methyl dopa to be filled into hard gelatin capsules. The experimental plan consisted of assessing blend flow and dissolution rate using formulations manufactured at small laboratory scale, selecting the optimum formulation, and confirming the data. Two optimization techniques were used in the solid pharmaceutical product design: a genetic algorithm (GA) and a downhill simplex technique. The genetic algorithm used in this work was implemented in an interactive form. Data for each generation of formulations were introduced to the computer with the corresponding values of a fitness function, which was determined in experimental form for each individual formulation. The fitness function used to evaluate product performance (capsule) was defined in terms of the dissolution rate multiplied by a weight function that penalizes those formulations with flow index outside a predefined range. The formulation design contained variable concentrations and types of lubricants/glidants. There were 64 combinations of seven agents with discrete ranges of concentrations codified into a 16-bit chromosome. Crossing and mutation

*Corresponding author.

operations were implemented with relatively high probabilities, for generations with a relatively small number of individuals, due to the restrictions imposed by the experimental cost. The mixed formulation strategy based on genetic algorithms and downhill simplex was used to obtain sequentially improved formulations based on two desired targets: in vitro dissolution rate and flow properties. The basic downhill simplex method was used to obtain an optimal formulation on the regression response surface obtained from the genetic algorithm data. The results obtained in this work clearly illustrate the potential of the proposed mixed optimization strategy to obtain optimal formulations.

Key Words: *Capsule formulation; Downhill simplex; Genetic algorithms; Mixed optimization techniques*

INTRODUCTION

Common pharmaceutical formulation techniques currently may still have a trial-and-error nature due to the absence of mathematical models that suitably represent the relationships between formulation characteristics and product performance. By considering the properties of pharmaceutical formulations as functions of the proportions of different ingredients, a numerical optimization technique can be applied to obtain better formulations sequentially. An objective function that measures the aptitude of a formulation for a particular set of values of the independent variables must be defined for the optimization procedure. In the case of a pharmaceutical formulation constituted by multiple ingredients, this leads to a problem of multidimensional optimization. A particular difficulty is encountered when attempting to apply traditional gradient methods since it is not possible to have a simple analytical model to represent the behavior of the formulation. In this case, it is necessary to evaluate the aptitude of the formulation by means of laboratory experiments, and it is not possible in general to obtain the derivatives of the objective function analytically. Their evaluation would then have to be implemented by means of multiple evaluations of the function. In the case of pharmaceutical formulations, this constitutes an expensive sequence of experiments or laboratory tests for each new formulation.

The objective of this work was to apply a numerical search procedure to obtain successively improved formulas based on two optimization methods: genetic algorithms (GAs) and downhill simplex optimization technique. Most computational strategies for a multidimensional minimization are based on extensions of one-dimensional minimization algorithms, which require evaluation of the objective function and its derivatives. The simplex method does not require evaluation of derivatives. Consequently, it is preferred for minimization of a function with a value that is obtained as an outcome of a physical experiment. In general, the use of an optimization technique is based on function evaluations without derivative results in a method that is not very efficient in terms of the function evaluations that it requires. However, these function evaluations are considerably less than those usually required for the application of gradient methods, either by construction of a surface response model or by numerical evaluation of the derivatives. Therefore, in relation to time and material expenses required for the execution of the optimization procedure, the simplex method is considered adequate. This search method has additional advantages for pharmaceutical product formulation since allowable ranges of the independent variables or secondary objective functions can be imposed as restrictions in a very simple way.

In the original simplex technique, the function is evaluated at the new point, and the highest point is discarded. In terms of product development, this means that a new formulation is prepared, and its properties are evaluated for each point in the sequential procedure. For pharmaceutical formulations for which time effects are important, this means that laboratory tests have to be performed for each new formulation, thus resulting in a time-consuming sequence. Consequently, it is desirable to obtain more than one candidate formulation at each step, thereby reducing the total time for product development. Genetic algorithms are search methods based on generations of simultaneous evaluations that can be applied to a wide range of problems. They have the main advantage of not needing more information than the value of an aptitude function that measures the success (fitness) of a certain design. For these reasons, in this work a combined strategy was proposed that was divided into two parts: (a) exploration and sequential improvement using a genetic search and (b) final optimization using the simplex method. A description of the basic numerical search methods follows.

NUMERICAL SEARCH METHODS

Downhill Simplex Method

The sequential simplex optimization was originally developed by Nelder and Mead (1) and well described by Press et al. (2). The method requires only function evaluations, not derivatives. The results from previous experiments are used to define the variable levels for the next experiment in the search for an optimum. This technique has been successfully applied to optimize a capsule formulation (3).

This optimization procedure requires a comparison between the objective function values at a limited number of search points in an n-dimension search space. The algorithm is based on a geometric figure called the *simplex*, consisting of $n+1$ points or vertices, together with all interconnecting line segments. In two dimensions, the simplex is a triangle, and in three dimensions, it is a tetrahedron.

The algorithm starts by defining the initial simplex using the best estimate of the solution. Then, it takes a series of steps in which the vertex of the simplex corresponding to the“worst” case (relative to the other vertices of the simplex) is moved, or reflected, through the opposite face of the simplex, thus moving downhill through an n-dimensional topography, until it encounters a minimum. After each reflection, the result at the new vertex is compared with those of the old vertex. If the new vertex is “better,” the algorithm expands the simplex by moving the vertex further out (away from the face it was reflected about). If it is worse, it contracts the simplex. If no improvement is provided by the contraction, the algorithm resets the simplex to its last value and then shrinks it about the best vertex.

For a two-variable study, there are three different responses: the best response X_B, the worst response X_W, and the second-best response X_S. Three operations are then used to move the simplex along the response surface toward an optimum in Nelder and Mead’s simplex technique: reflection, expansion, and contraction.

Reflection

Reflection away from the worst response X_W takes place through the centroid X_0 of the remaining vertices (X_S and X_B). Mathematically, the reflected vertex is given by

$$X_R = (1+a)X_0 - aX_W$$

where X_R is the reflected vertex, X_0 is the centroid of the remanent vertices except for X_W, which is the vertex with the worst response, and a is the reflection coefficient ($a > 0$), typically set to 1. The centroid X_0 is calculated as follows:

$$X_0 = \frac{1}{n}\sum_{i=1}^{n} X_i$$

where n is the number of variables. If the response vertex at X_R lies between the responses at vertex X_S and X_B, then the vertex X_R is retained, and the new simplex (X_S, X_B, X_R) is evaluated.

Expansion

If the response during reflection at vertex X_R is better than the response at vertex X_B, then expansion of the simplex is attempted. The expansion operation is used in the modified sequential simplex technique to help the formulator accelerate the process of locating an optimum formulation. The point X_E is given mathematically by

$$X_E = bX_R + (1-b)X_0$$

where b is the expansion coefficient ($b > 1$), typically set at 2. If the response at vertex X_E is better than that at X_R, then X_R is replaced by X_E, and the new simplex (X_S, X_B, X_E) is evaluated. On the other hand, if the response at X_E is poorer than at X_R, the expansion process is considered unsuccessful. The vertex X_R is then retained, and reflections begin again.

Contraction

If during the reflection process, the response at X_R is worse than all other responses in the simplex except for that at vertex X_W, we replace vertex X_W with vertex X_R and contract. The contraction operation not only accelerates the process of locating the optimum, but also allows the figure to conform better to the response surface, especially if there are sharp ridges or valleys. Mathematically, the vertex generated by contraction X_C .is given by

$$X_C = CX_W + (1 - c)X_0$$

where c is the coefficient of the contraction ($0 \leq c \leq 1$), typically set to 0.5. If during the reflection process, the response at vertex X_R is worse than all other responses in the simplex, including vertex X_W, we retain X_W and locate the contracted vertex using the same equation as above. If the contraction from the original or reflected simplex produces a response at vertex X_C that is better than that at X_W, then the vertex X_W is replaced by vertex X_C, and the reflection process proceeds. However, if the contracted vertex produces a response that is worse than that at X_W, the contraction process is considered a failure. Then, all X_i are replaced by $(X_i + X_B)/2$, and the reflection process is restarted. The reflection, expansion, and contraction operations are performed until an optimum response is located.

Genetic Algorithms

Genetic algorithms (GAs) are adaptive methods that solve problems of search and optimization by imitating the genetic process of living organisms. Throughout the generations, populations in nature evolve according to the principles of natural selection and survival of the fittest, postulated by Darwin in 1859 (4). Similarly, in a genetic algorithm, selection and reproduction allow evolution of problem solutions toward optimal values, depending on a suitable codification. The basic principles of genetic algorithms were established by Holland (5), and they are described well in several texts (6–9). Genetic algorithms can successfully treat a great variety of problems from different areas, including those in which other methods find difficulties.

A genetic algorithm is an iterative procedure based on a constant-size population of individuals, each represented by a finite string of symbols, known as the *genome*, encoding a possible solution in a given problem space. This space, referred to as the *search space*, comprises all possible solutions to the problem at hand. Generally, a genetic algorithm is applied to spaces that are too large to be searched exhaustively. The fundamentals of the technique for the evolutive exploration of such space are given below.

Solution Representation (Codification)

A set of possible solutions for the optimization problem is considered a population of individuals. Coordinates of an individual in the search space are represented by *chromosomes*, in essence a set of character strings. A *gene* is a subsection of a chromosome that encodes the value of a single parameter being optimized. To tackle a problem using a GA, candidate solutions must be encoded in a suitable form. In the traditional GA, solutions are represented by binary bit strings (chromosomes). While integer and decision variables are easily encoded in this form, the representation of continuous control variables is not so simple. In general, the only option available is to approximate them (rescaled as necessary) by equivalent integer variables. The accuracy with which an optimum solution can be resolved then depends on the encoded bit length of these integers, leading to an inevitable compromise between precision and execution time.

Fitness Function

The degree of adaptation of an individual to its environment is specified by its fitness. In biological terms, the set of parameters representing a particular chromosome is denominated a *phenotype*. The phenotype contains the required information to construct an organism. The adaptation of an individual to the problem depends on the evaluation of the genotype, which can be inferred from the phenotype. In other words, it can be computed from the chromosome, using the evaluation (fitness) function. A fitness function or quality criterion must be specifically designed for each problem. Given a particular chromosome, the fitness function assigns a real number to it, which represents its level of adaptation to the problem.

Selection

A standard genetic algorithm proceeds as follows: An initial population of individuals is generated at random or heuristically. At an evolutionary step, known as a *generation*, every individual in the current population is decoded, and its fitness function is evaluated. To form a new population (the next generation), individuals are selected according to their fitness. Many selection procedures are currently in use, one of the simplest being Holland's original fitness-proportionate selection, by which individuals are selected with a probability proportional to their relative fitness. This ensures that the expected number of times an individual is chosen is approximately proportional to its relative performance in the population. Thus, high-fitness individuals stand a better chance of reproducing, while low-fitness ones are more likely to disappear.

Reproduction

Selection alone cannot introduce any new individuals into the population; that is, it cannot find new points in the search space. Therefore, genetic operators must be applied to the selected individuals during the reproductive phase. The best-known genetically inspired operators are crossover and mutation. *Crossover* is performed with some probability between two selected individuals, called *parents*, by exchanging parts of their genomes (i.e., encoding) to form two new individuals, called *offspring*. In its simplest form, substrings are exchanged after a randomly selected crossover point. This operator tends to enable the evolutionary process to move toward "promising" regions of the search space. The mutation operator is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. This is performed by flipping bits at random with some small probability.

Crossover

The simplest form of crossover (one-point crossover) proceeds as follows. First, the entire population is paired off at random to give sets of potential parents. Second, pairs of solutions are chosen to undergo crossover with some predefined probability. If the simulated weighted coin toss rejects crossover for a pair, then both solutions remain in the population unchanged. However, if it is approved, then two new solutions are created by exchanging all the bits following a randomly selected locus on the strings. For example, if crossover after position 5 is proposed between solutions

1	0	0	1	1	1	0	1

and

1	1	1	1	0	0	0	0

the resulting offspring are

1	0	0	1	1	0	0	0

and

1	1	1	1	0	1	0	1

Mutation

The mutation operator is applied to each child on an individual basis. It consists of the random alteration of each gene component from the chromosome. The purpose of the mutation stage is to provide insurance against the irrevocable loss of genetic information and hence to maintain diversity within the population. This last fact is of capital importance to ensure the convergence of genetic algorithms. For instance, if every solution in the population has 0 as the value of a particular bit, then no amount of crossover will produce a solution with a 1 there instead. In traditional GAs, every bit of every solution is potentially susceptible to mutation. Each bit is subjected to a simulated weighted coin toss with a probability of mutation, which is usually very low (on the order of 0.01 or less). If mutation is approved, the bit changes value from 0 to 1 or from 1 to 0.

Genetic algorithms are stochastic iterative processes that are not guaranteed to converge in the usual sense. GA does not use derivative information; it just needs to be supplied with a fitness value for each member of each population. Thus, the evaluation of the problem functions is essentially a "black box" operation as far as the GA is concerned. The termination condition may then be specified as some fixed, maximal number of generations or as the attainment of an acceptable fitness level.

The standard genetic algorithm in pseudocode format is shown as follows:

```
Begin GA
g: $= 0$ {generation counter }
Initialize population $P(g)$
Evaluate population $P(g)$ {i.e., compute fitness
   values}
while not done do
g: $= g + 1$
Select $P(g)$ from $P(g - 1)$
Crossover $P(g)$
Mutate $P(g)$
Evaluate $P(g)$
end while
end GA
```

Rowe (10) recently raised the possibility of the application of genetic algorithms for pharmaceutical product formulation. In his article, he assured that, at the present time, commercially available software packages that apply this optimization procedure for pharmaceutical product formulation do not exist.

EXPERIMENTAL

Materials

All the materials used in this work conformed to USP and/or NF requirements. Alpha methyl dope was used as the active ingredient. The following ingredients were evaluated as lubricant/glidant agents: sodium lauryl sulfate (Henkel, Inc.) carbowax 4000 (Riedel-De Haen, Ag Seelze Hannover), carbowax 6000 (Riedel-De Haen, Ag Seelze Hannover), aerosil, talc (J. T. Baker Chemical), magnesium stearate, and stearic acid.

Sample Preparation

Since the objective of this work was to optimize the formulation of a dry powder blend to be filled into a hard gelatin capsule at a small laboratory scale, each formulation consisted of a batch of 20 capsules. Each capsule contained 500 mg of alpha methyl dope and a lubricant/glidant agent or a mixture of them as excipients.

Manufacturing Method

The following general procedure was used to manufacture all the formulations. Alpha methyl dope was hand screened through a 45-mesh screen. All lubricant/glidant agents were hand screened through an 80-mesh screen. Powder mixtures, each batch weighing 20 gr, were prepared by placing the drug (alpha methyl dope) and the lubricant/glidant agent(s) into a cylindrical glass jar (100-ml capacity) and mixed manually for 1 min. The blend was filled manually into 20 size 0 hard gelatin capsules.

Evaluation of Sample Flow Properties

Blend flow was determined using packing properties such as bulk and tapped powder density and derived parameters. An important parameter of interest in predicting flow properties of a powder is a quantity known as the percentage compressibility C. It can be obtained from bulk and tapped density determinations as follows:

$$C = \frac{\rho_t - \rho_b}{\rho_t} \times 100$$

where ρ_t is the tapped density, and ρ_b is the bulk density. In theory, the more compressible a bed of particulate is, the less able to flow the powder will be. Carr (11) defines a material having a C value of less than 20% to 21% as being a free-flowing material.

Powder bulk densities were determined by weighing 10 gr of the powder (loose packed) in a 25-ml graduated cylinder. Tapped densities were determined after 500 taps on a Vanderkamp tap density tester (Vankel Industries, Chatam, NJ)

Dissolution Rate Determination

Dissolution was determined for three capsules of each formulation in a Hanson dissolution testing apparatus according to the capsule method described in the USP 23 (12). The dissolution medium was 900 ml of hydrochloric acid 0.1 N at 37°C. The stirring speed was 50 rpm. After 20 min, 5-ml samples were withdrawn. The samples were diluted up to 50 ml with hydrochloric acid 0.1 N, and their concentrations were measured on a Beckmann DU 640 spectrophotometer at a wavelength of 280 nm as specified in the USP 23. The percentage of drug dissolved at 20 min was calculated, and a mean value for three capsules was obtained.

IMPLEMENTATION OF THE MIXED OPTIMIZATION

Formulation of the Design Problem

In this work, alpha methyl dope was selected as a model drug due to its good dissolution rate when it is administered without the addition of any other excipients. On the other hand, this active ingredient has poor flow properties. In the industrial manufacture of capsules, the powder flow properties of the mass to encapsulate are of particular importance since for filling uniformity it is necessary that the solid layer reforms so quickly that a uniform dose can be encapsulated. Consequently, one of the most important potential ingredients to be used in its capsule formulation is the lubricant/glidant agent. Since in general the most effective of these agents are hydrophobic in nature, the film cover that forms around particles when the agent is added causes an increase of capsule disintegration time and a reduction of the rate of dissolution of the active ingredient. Since adding additional lubricant to the capsule may improve its ease of manufacturing but may adversely affect dissolution, the formulation properties must be evaluated before filling in terms of their behavior of flow; later, in vitro tests must be done to predict their effectiveness following oral administration of the active ingredient (dissolution tests).

It is well established that the release of poorly soluble hydrophobic drugs from capsules can be improved significantly by the creation of a hydrophilic surface by mixing of the hydrophobic drug with a small amount of a hydrophilic excipient (13). Consequently, it is necessary to add the right type and concentration of standard lubricant/glidant ingredients (hydrophilic and hydrophobic) to allow both capsule properties (dissolution and flow) to be optimized. Seven agents were evaluated for their effect on the flow and dissolution properties of alpha methyl dope. The mixed optimization strategy proposed for the formulation of this active ingredient was then aimed initially to improve the flow properties of alpha methyl dope without reducing its rate of dissolution. Results from flow and dissolution rate determinations were used in the calculation of the fitness function and were introduced to the computer to follow the search using a genetic algorithm to explore the response surface.

Codification of the Independent Variables

The ingredient's specifications for each formulation were codified using 6 bits to indicate the combination of 0, 1, 2, or 3 additives from a list of 7 lubricants/glidants. The remaining 10 bits were used to specify the concentrations of additives in discretized ranges. Then, for the total chromosome length of 16 bits, the search space to explore had a size of $2^{16} = 65{,}536$ individuals. Given the restrictions imposed by the availability of the active ingredient, a population size of 16 individuals was selected as adequate for the size of our chromosome.

Probabilities of Crossing and Mutation

Previous to the use of this searching scheme with experimental data, numerical tests of the basic algorithm with hypothetical fitness functions were made. The best results were obtained with relatively high probabilities of crossing (between 0.9 and 1) and mutation (up to 0.3) and imposing variations in the crossing and mutation probability depending on the relative position of the parents in the population. In this way, the descendants of less apt parents would have greater opportunity to improve by mutation. Competition was also allowed between the new generation and the parent generation to avoid a decrease in the average fitness function due to the small population size. The original batch computer code was modified to introduce to the computer the values of the fitness function from each individual (formulation) interactively. The fitness function was calculated from the formulation flow index and dissolution properties, determined in experimental form, by means of a spreadsheet interface. This also allows intervention of the expert user to reject an off-range point by assigning a value of zero to its fitness function value.

Fitness Function

The *rate of dissolution* (defined as the percentage of dissolved drug after 20 min less than 80%) affected by a penalty function depending on the flow index value was used as the fitness function. Although percentage compressibility determination from bulk and tapped density is a well-established technique for assessment of flow properties, there is relatively large variability in its results. According to Carr (11), a value of percentage of compressibility between 5 and 16 is considered suitable as an index of good flow. A reduction factor applied to the dissolution rate was used as a way (penalty function) to penalize formulations with a flow index outside this range. The factor was chosen so that a reduction of only 5% corresponded to the ends of the interval, and the fitness was highly reduced for those formulations with a flow index that moved farther away. The penalty function P is expressed as follows:

$$P = \frac{6}{6 + \left[\ln\left(\frac{C}{8.66}\right)\right]^2}$$

where C is the flow index. The fitness function is then expressed as

$$F = PQ$$

where Q is the dissolution rate. The relationship between the penalty function and the flow index is shown on Fig. 1.

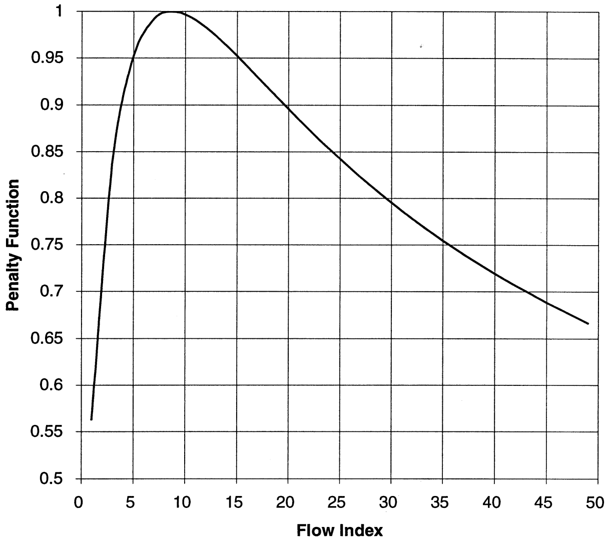

Figure 1. Relationship between the penalty function and flow index.

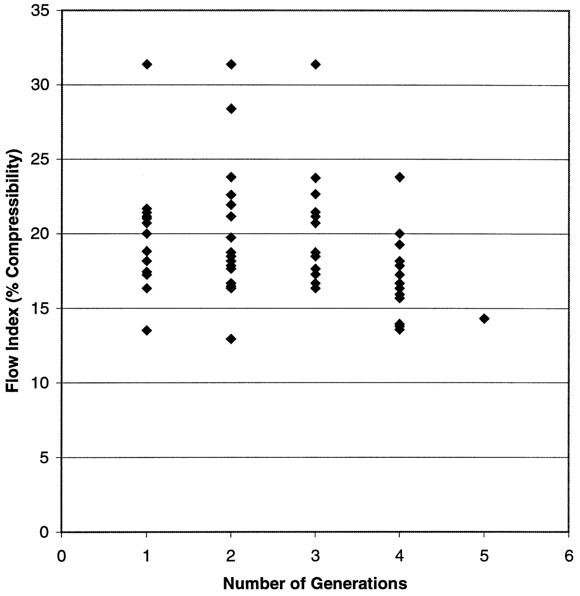

Figure 2. Evolution of the values of the flow index.

Final Optimization

The selection of the best compromise for the considered design problem is an expert choice since this is a multiobjective optimization problem. Consequently, the direct application of the numerical techniques helps to identify trade-off among conflicting objectives, but cannot give single optimal solutions. Typically, the genetic search would end when several equal formulations with an optimal combination of flow properties and dissolution rate were obtained. The preliminary numerical experiments with hypothetical fitness functions indicate that at least 16 generations are required to obtain convergence in that sense. For this reason, a mixed approach is followed by which the experimental genetic search is stopped when there are enough acceptable formulations to choose a fixed set of ingredients and to construct a regression model on those variables. Then, a continuous variable optimization procedure is applied using the regression model.

In the present implementation, after the application of the genetic algorithm in the search for a good range of the capsule formulations, the accumulated data were fitted to a polynomial regression model for each product property (flow properties and dissolution rate). After this point in the experimentation, the basic downhill simplex method was used to obtain an optimal formulation on the response surface of the regression model. The objective function to minimize was selected as the negative of the fitness function. The simplex method was selected for this design stage due to its suitability for fast implementation, and the ability to handle the required constraints, by assigning artificially large values to the objective function at off-range points.

RESULTS

Evolution of Flow and Dissolution Properties

The genetic algorithm implemented in this work uses 16 formulas per generation. A set of starting point formulations for the final optimization procedure was obtained after 4 generations (64 formulations). The behavior of the genetic algorithm in 4 generations can be seen in Figs. 2, 3, and 4. Figure 2 shows how the flow index (%C) approaches the desired range. In Fig. 3, the evolution of the percentage of drug dissolved after 20 min (tests of dissolution) can be observed.

Figure 4 shows the evolution of the fitness function. The values shown in all figures are from the new experimental measurements obtained at each generation, and some decreasing values of the fitness function are observed in Fig. 4. However, due to competition among the parent and child generations, after selection, the average fitness is always a nondecreasing function, and a clear increase can be

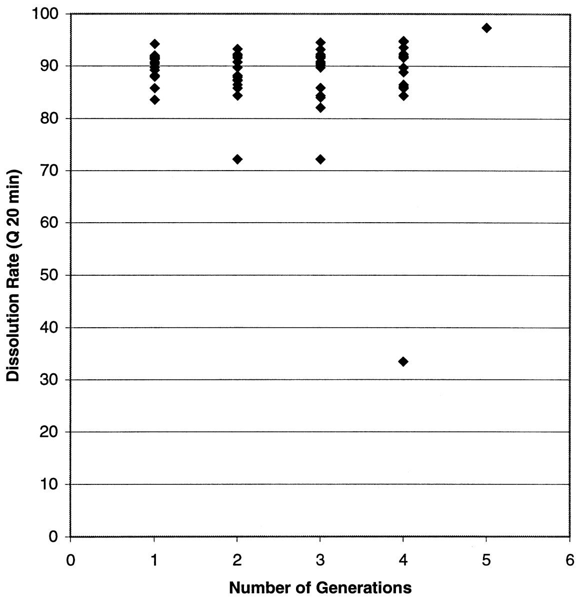

Figure 3. Evolution of the values of the percentage of drug dissolved in 20 min $Q_{20}\%$.

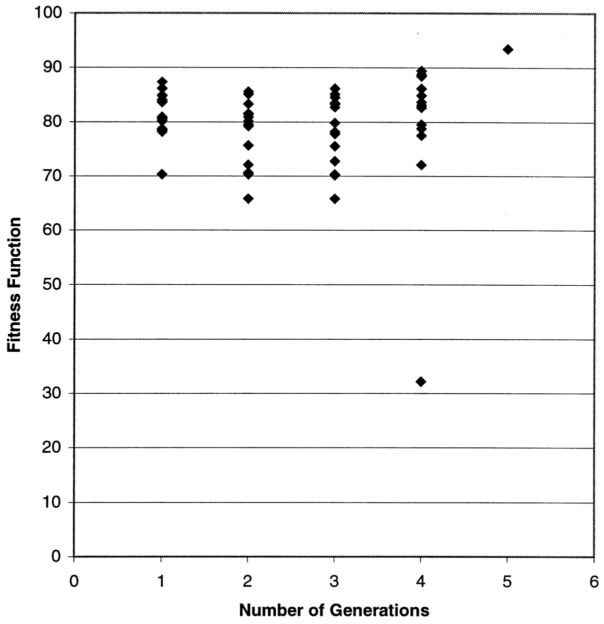

Figure 4. Evolution of the fitness function values.

observed at the fourth generation. At this point, formulations with good properties of flow and dissolution are obtained. The value of the fitness function achieves a global maximum value of 89.70% (percentage drug dissolved).

Regression Model

The model corresponding to the flow index $C(s,p,m)$ is described by the following equation:

$$\begin{aligned} C(s,p,m) = {} & 19.46223906 + 0.54949969\,s \\ & + 1.28085761\,p - 44.03089\,m \\ & + 45.4850716\,sm - 5.0136334\,msp \\ & + 59.91838\,m^2 - 58.250006\,sm^2 \\ & - 1.5048716\,mp \end{aligned}$$

where s, p, and m are the concentrations of sodium lauryl sulfate, carbowax 6000, and magnesium stearate, respectively. The standard error of this regression is 3.28.

The model corresponding to the dissolution rate $Q(s,p,m)$ is described by the following equation:

$$\begin{aligned} Q(s,p,m) = {} & 93.180166 - 3.975536 - 3.5441426p \\ & - 35.09326\,m + 5.8711211\,mp \\ & - 1.691754\,sp + 1.3642834\,p^2 \\ & + 0.8443732\,s^2 + 23.733612\,sm \\ & + 49.150968\,m^2 - 28.04339\,sm^2 \end{aligned}$$

with a standard error of 3.13.

Application of the Simplex Method

According to USP 23 (12), the specifications established for a solid formulation of alpha methyl dope indicate that, to fulfill the rate of dissolution specifications, no less than 80% of the amount of declared drug should dissolve after 20 min in HCl 0.1 N dissolution medium. Since analytical unconstrained optimization would yield unfeasible values of the dissolution rate, the value of Q_{20} was restricted to a range between 80% and 100%. The optimum formulation was selected to contain the active ingredient and a mixture of three lubricant/glidant agents: two of hydrophilic nature (1.76% sodium lauryl sulfate and 4.02% carbowax 6000) and one of hydrophobic nature (0.51% magnesium stearate).

Verification

A follow-up experiment was performed to confirm the predicted optimal combination of components given by the downhill simplex optimization technique. This single experimental point is depicted

Table 1

Predicted Value and Measured Value of Rate of Dissolution and the Flow Index of the Optimum Formulation

Measured Response	Predicted Value	Measured Value
Flow index (%C)	11.83	14.3
Dissolution rate (Q_{20} min)	99.97	97.39

in Figs. 2–4 as generation number 5. The above formulation has a good value of flow index (percentage of compressibility $C = 14.30\%$) and excellent dissolution rate Q_{20} (97.39% dissolved at 20 min). By comparison, it is important to point out that a capsule formulation containing only the active ingredient (alpha methyl dope) shows a value of the percentage of compressibility C of 29.43% (poor flow) and a percentage of dissolved drug after 20 min of 89.70%. The agreement found between the observed results for both properties (flow and dissolution rate) with the predicted ones indicates that this computer optimization mixed formulation technique looks promising for designing capsules (Table 1).

CONCLUSIONS AND RECOMMENDATIONS

The use of a mixed optimization technique that combines the flexibility of genetic algorithms for experimental search and the well-established convergence of the downhill simplex for optimization was illustrated. The ability of the method to obtain good capsule formulations was confirmed. The interactive implementation allows combining the ability of the evolutive numerical algorithm to obtain promising formulations with the experimental confirmation of their properties. It also adds the flexibility of intervention by the expert user.

This scheme allows the evaluation of several formulations simultaneously, which enables faster searching for new formulations during the early formulation stages than classical response surface methodology and with considerably smaller expense of materials. At such times, the active ingredient for evaluation of a formulation under production conditions usually is available in small quantities. The application of a genetic search for exploration also enables selection of the type of ingredients in addition to their concentrations, which in general cannot be accomplished directly with traditional optimization methods. The effectiveness of the genetic algorithm in finding good starting points for final optimization was demonstrated.

Based on these abilities, the potential application of genetic algorithms for exploratory analysis in dosage form design, associated with the downhill simplex technique for final optimization, can be recommended for capsule formulation. The results obtained in this work are preliminary in the sense that their application in capsule formulation was simplified and reduced to optimize flow properties and dissolution rate due to causal factors such as type and concentration of lubricant/glidant ingredients. However, these results provide a base to carry out applications to more complex problems of pharmaceutical formulation design, such as tablets formulations with more ingredients.

ACKNOWLEDGMENT

We give special thanks to the CDCH of Universidad Central de Venezuela for grant 06,175,97, with additional funding provided by the Vicerrectorado Academico, which allowed the upgrade of the computer used in the project. We would like to thank Cruz Guerrero de Diz for her participation in obtaining data used in the study.

REFERENCES

1. Nelder, J.A.; Mead, R. Computer J. **1965**, *7*, 308–313.
2. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T. *Numerical Recipes in C*; Cambridge University Press: Cambridge, UK, 1990.
3. Shek, E.; Ghani, M.; Jones, R.E. J. Pharm. Sci. **1980**, *69* (10), 1135–1142.
4. Darwin, C. *On the Origin of Species by Means of Natural Selection*; Murray: London, 1859.
5. Holland, J. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, 1975.
6. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Reading, MA, 1989.
7. Davis, L. *Handbook of Genetic Algorithms*; Van Nostrand Reinhold: New York, 1991.
8. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer Verlag: Berlin, 1992.

9. Reeves, C. *Modern Heuristic Techniques for Combinatorial Problems*; Blackwell Scientific Publications: Oxford, UK, 1993.
10. Rowe, R. Pharm. Technol. **1997**, *March*, 178–188.
11. Carr, R.L. Brot. Chem. Eng. **1970**, *15*, 1541–1549.
12. U.S. Pharmacopeial Convention. *United States Pharmacopeia XXIII*; Mack Publishing Co.: Easton, PA, 1995; 994–995.
13. Lerk, C.F.; Lagas, M.; Fell, J.T.; Nauta, P. J. Pharm. Sci. **1978**, *67* (7), 935–939.