

ỨNG DỤNG MẠNG NƠN, LOGIC MỜ VÀ THUẬT TOÁN DI TRUYỀN GIẢI QUYẾT BÀI TOÁN TỐI ƯU HÓA CÔNG THỨC VÀ QUY TRÌNH SẢN XUẤT DƯỢC PHẨM

Hoàng Kiếm⁽¹⁾, Lê Hoài Bắc⁽¹⁾, Đỗ Quang Dương⁽²⁾

⁽¹⁾ Khoa CNTT – Trường Đại Học Khoa Học Tự Nhiên – ĐHQG-HCM

⁽²⁾ Khoa Dược – Trường Đại Học Y Dược Tp. Hồ Chí Minh

(Bài nhận ngày 11 tháng 02 năm 2003)

TÓM TẮT: Bài báo giới thiệu một giải pháp tối ưu hoá công thức, quy trình sản xuất dược phẩm bằng máy tính, giải pháp này là sự kết hợp của các kỹ thuật thông minh nhân tạo: mạng nơron, logic mờ, thuật toán di truyền. Với giải pháp này nhà bào chế thuốc sẽ tiết kiệm được thời gian và công sức hơn so với việc dùng cách truyền thống vì chương trình máy tính xử lý sẽ cho kết quả nhanh chóng và chính xác hơn.

1. Giới thiệu:

Giai đoạn nghiên cứu và phát triển thuốc có liên quan đến hai nội dung chính: thành lập công thức và xây dựng quy trình sản xuất, trong đó xây dựng công thức là nền tảng. Đối với nhà sản xuất thuốc, việc thành lập công thức là việc thường xuyên phải làm, bởi vì mỗi sản phẩm đều có một vòng đời nhất định và nhu cầu cạnh tranh trên thị trường đòi hỏi phải không ngừng cải tiến sản phẩm hiện có hay thay thế sản phẩm mới [1]. Các phương pháp tối ưu hóa truyền thống (toán thống kê, đơn hình (simplex)...) có thể áp dụng với các dữ liệu đơn giản và tuyến tính. Nhưng với những dữ liệu phức tạp và phi tuyến thì các phương pháp này không còn phù hợp.

Phương pháp tối ưu hóa bằng máy tính có nhiều triển vọng thay thế các phương pháp truyền thống trong mục đích tối ưu hóa vì nó không giới hạn số biến độc lập (x); có thể tối ưu đồng thời nhiều biến phụ thuộc (y); phù hợp với các loại dữ liệu phức tạp và phi tuyến [1].

Việc ứng dụng *mạng nơron*, *logic mờ* và *thuật toán di truyền* giải quyết bài toán tối ưu hóa công thức và quy trình phần nào giúp cho các dược sĩ tiết kiệm được thời gian và sức lực hơn so với việc dùng các phương pháp thủ công truyền thống trong quá trình thành lập công thức. Mỗi kỹ thuật giới thiệu đều có những ưu điểm và hạn chế. Tuy nhiên khi kết hợp chúng lại với nhau, ưu điểm của kỹ thuật này sẽ khắc phục hạn chế của kỹ thuật khác [3][5][6][8] - mạng nơron không có khả năng biểu diễn tri thức, trong khi đó logic mờ giải quyết vấn đề này rất tốt - và giải quyết khá tốt bài toán tối ưu hóa công thức và quy trình sản xuất.

2. Mạng nơron và hệ thống neuro-fuzzy [6][7][8][9]

Hệ thống suy diễn mờ và mạng nơron là các kỹ thuật trong các hệ thống thông minh. Mạng nơron học từ dữ liệu thực nghiệm bằng cách điều chỉnh các trọng số trên các liên kết giữa các nút trên các lớp mạng. Hệ thống suy diễn mờ sử dụng những tính toán dựa trên lý thuyết mờ, luật mờ *if...then...else...* và các hàm thành viên. Hệ thống neuro-fuzzy tương tự như hệ thống suy diễn mờ nhưng được tăng cường khả năng học của mạng nơron. Với các kỹ thuật cài đặt khác nhau trong chức năng xử lý, hệ thống neuro-fuzzy được phân loại: GARIC (*Generalized Aproximate Reasoning based Intelligent Control*), FALCON (*A Fuzzy Adaptive Learning Control Network*), ANFIS (*Adaptive Neurofuzzy Inference System*), NEFCON (*Neural Fuzzy Controller*), FUN (*Fuzzy Using Neural Network*), EFuNN (*Evolving Fuzzy*

Neural Networks), dmEFuNN (Dynamic Evolving Fuzzy Neural Networks). Sau đây là hai mô hình Neuro-Fuzzy System được sử dụng khá thông dụng:

Mô hình 1: Mamdani Neuro-Fuzzy System (Hình 1.)

Lớp 1: *nhập* - giá trị biến độc lập được đưa trực tiếp vào nút mạng.

Lớp 2: *mờ hóa* - tính giá trị mờ tương ứng cho các giá trị từ đầu vào thông qua các hàm thành viên. Các tham số của hàm thành viên được cập nhật trong quá trình học của mạng.

Lớp 3: *phát sinh luật* - mỗi nút mạng sẽ tính toán và phát sinh luật, toán tử T-norm dùng ở lớp này.

Lớp 4: *kết hợp luật* - nút mạng ở lớp này có hai chức năng chính: kết hợp các luật có từ lớp trước và xác định giá trị mờ của giá trị kết xuất.

Lớp 5: *tổng hợp và giải mờ* - nút mạng lớp này có chức năng tổng hợp các luật và tính kết quả thực xuất ra.

Mô hình 2: Takagi-Sugeno Neuro-Fuzzy System (Hình 2.)

Lớp 1, lớp 2, lớp 3: xử lý tương tự mô hình Mamdani Neuro-Fuzzy System.

Lớp 4: mỗi nút trong lớp này sẽ nhận giá trị xuất của lớp trước và sau đó tính tỷ số như sau:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2, \dots$$

Khi đó mỗi luật sẽ có thêm một giá trị \bar{w}_i sẽ sử dụng trong việc tính giá trị nút xuất ở lớp sau (giá trị này gọi là *normalized firing trengths*).

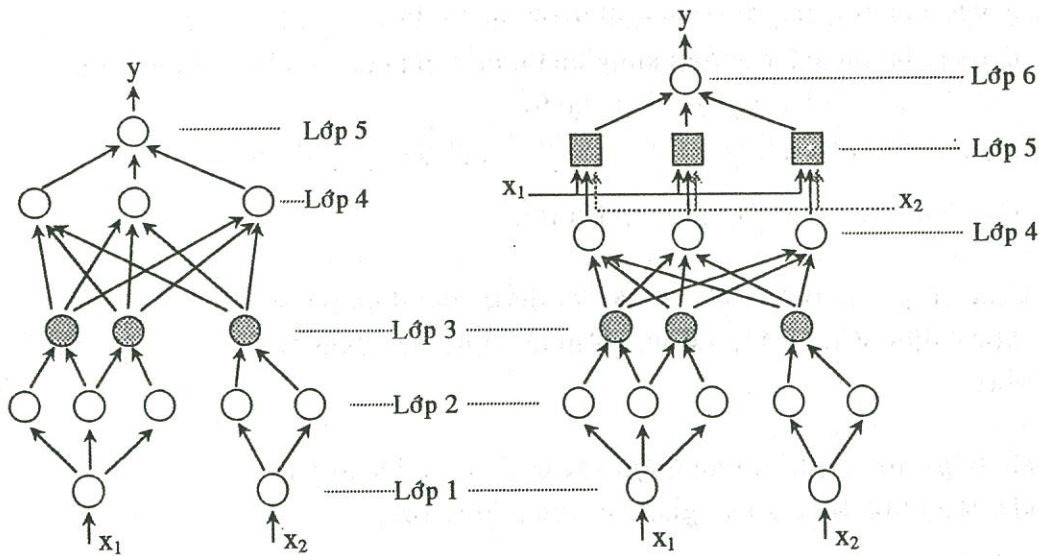
Lớp 5: mỗi nút ở lớp này là nút điều chỉnh với kết quả xuất ở nút cuối, giá trị xuất ở mỗi nút có giá trị:

$$\bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i)$$

tập {p, q, r} là tập các tham số trong lớp này, các tham số này xem như là tham số tổng hợp trong mô hình ANFIS (hay SONFIN). Hàm f tính theo mô hình Sugeno.

Lớp 6: ở lớp này chỉ có một nút tổng hợp kết quả xuất ra từ lớp trước, giá trị xuất ở nút này là tổng các giá trị kết quả xuất ra từ lớp trước:

$$\text{Output} = \sum \bar{w}_i f = \frac{\sum_i w_i f_i}{\sum_i w_i}$$



Hình 1. Mamdani Neuro-Fuzzy System

Hình 2. Takagi-Sugeno Neuro-Fuzzy System

Mạng nơron và neuro-fuzzy systems giải quyết tốt vấn đề thiết lập mô hình nhân quả trong bài toán tối ưu hoá công thức và quy trình. Tuy nhiên, bài toán tối ưu hóa công thức vẫn chưa trọn vẹn nếu chỉ dùng kỹ thuật mạng nơron và logic mờ. Ở giai đoạn này, chỉ mới thiết lập được mô hình nhân quả (hàm thích nghi cho quá trình tối ưu dùng thuật toán di truyền) cho bài toán tối ưu hóa công thức và quy trình.

3. Giải quyết bài toán tối ưu dùng mạng nơron, thuật toán di truyền và logic mờ [1][2]

3.1. Mô hình kết hợp giữa thuật toán di truyền và logic mờ:

Kỹ thuật mạng nơron và logic mờ thiếu khả năng tối ưu hóa, trong khi đó tối ưu hóa là một đặc trưng của thuật toán di truyền. Thuật toán di truyền áp dụng các nguyên tắc của sự tiến hóa trong tự nhiên vào hệ thống trí tuệ nhân tạo. Sự tiến hóa là một quá trình không điều khiển, các sinh vật tốt hơn sẽ thích nghi nhiều hơn với môi trường, sinh vật sống sót sẽ sinh sản... và đó là quá trình chọn lọc và tiến hóa tự nhiên.

Quá trình tối ưu: dùng thuật toán di truyền kết hợp logic mờ với hàm thích nghi dựa vào mô hình nhân quả được thiết lập trong quá trình luyện mạng nơron. Các giá trị thích nghi của từng cá thể là giá trị có được từ hàm thích nghi nhưng đã được xử lý (làm mờ hóa).

Các bước tối ưu hoá công thức quy trình dùng thuật toán di truyền:

Bước 1: khởi tạo quần thể, một phần dùng lại dữ liệu thực nghiệm đưa vào, một phần lấy ngẫu nhiên trong giới hạn khảo sát.

Bước 2: tính giá trị thích nghi từ kết quả dự đoán mạng nơron. Xác định các giá trị y_i . Tính các giá trị $\mu_i(y_i)$ ($\mu_i(y_i)$ là hàm thành viên tương ứng với y_i). Kiểm tra sự tối ưu: đạt/ không đạt

Bước 3: sao chép lại các nhiễm sắc thể dựa vào giá trị thích nghi của chúng và tạo ra những nhiễm sắc thể mới bằng lai ghép, đột biến).

Bước 4: loại bỏ những thành viên không thích nghi trong quần thể.

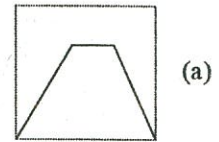
Bước 5: chèn những nhiễm sắc thể mới vào quần thể.

Bước 6: quay lại bước 2 cho đến khi việc tối ưu chưa đạt điều kiện định trước (100%) hay số vòng lặp chưa đạt tối đa.

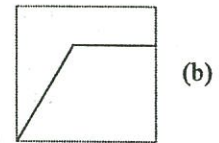
Điều kiện tối ưu (dấu = xảy ra): $\Sigma \mu_i(y_i) \leq \text{số biến phụ thuộc}$

Các dạng hàm mờ dùng trong quá trình tối ưu:

Hàm mái bằng (Flat Tent) (a): giá trị tối ưu được xác định trong khoảng từ Mid1 đến Mid2. Mức tối ưu giảm từ Mid1 đến Min và Mid2 đến Max.



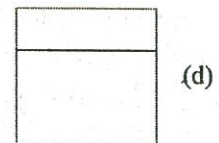
Hàm lên dốc (Up) (b): giá trị tối ưu được xác định trong khoảng từ Mid1 = Mid2 đến Max. Mức tối ưu giảm từ Mid1 đến Min.



Hàm xuống dốc (Down) (c): giá trị tối ưu được xác định trong khoảng từ (Mid1 = Mid2) đến Min. Mức tối ưu giảm từ Mid2 đến Max.

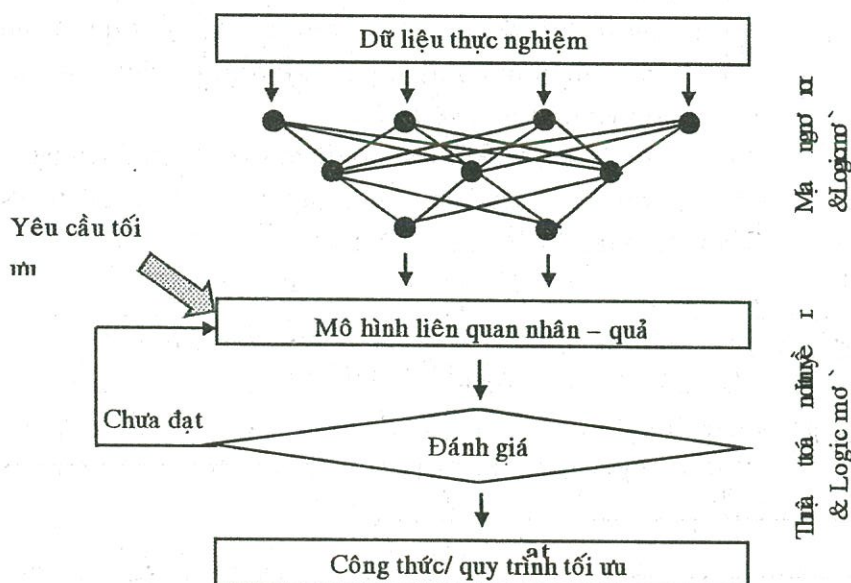


Hàm phẳng (Flat) (d): giá trị tối ưu được xác định một cách linh hoạt là bất kỳ điểm nào trong khoảng từ Min đến Max.



3.2. Kết hợp 3 kỹ thuật AI (Artificial Intelligence) giải quyết bài toán tối ưu hóa công thức và quy trình

Việc kết hợp các kỹ thuật mạng nơron, logic mờ và thuật toán di truyền giải quyết bài toán tối ưu hóa công thức và quy trình có thể được tóm tắt và minh họa như Hình 3:



Hình 3. Mô hình kết hợp kỹ thuật mạng nơron, logic mờ và thuật toán di truyền

Các bước tiến hành tối ưu (tìm giá trị biến độc lập theo biến phụ thuộc):

Bước 1: với dữ liệu thực nghiệm ban đầu, làm dữ liệu cho mạng nơron (hay neuro-fuzzy). Tìm mô hình nhân quả giữa các biến độc lập x và các biến phụ thuộc y . Kiểm tra lại độ chính xác của mô hình nhân quả bằng các bộ dữ liệu thử và các đánh giá thống kê.

Bước 2: xác định các khoảng giá trị các biến phụ thuộc theo yêu cầu người dùng, dùng thuật toán di truyền kết hợp logic mờ tìm ra các giá trị biến độc lập tốt nhất. Ở bước này có sử dụng logic mờ với mục đích là tìm những giá trị biến độc lập sao cho các giá trị biến phụ thuộc gần với giá trị được yêu cầu nhất. Thuật toán di truyền sử dụng mô hình nhân quả ở *Bước 1* làm hàm thích nghi để tìm các giá trị biến độc lập tốt nhất. Quá trình tìm kiếm trong quần thể sẽ lặp lại cho đến khi tìm ra giá trị biến độc lập thỏa các yêu cầu biến phụ thuộc hoặc hết thời gian cho phép. Kết quả xuất ra sẽ là giá trị biến độc lập tốt nhất đáp ứng được yêu cầu đặt ra đối với biến phụ thuộc.

4. Kết quả thực nghiệm và đánh giá

Chúng tôi đã xây dựng một chương trình ứng dụng sử dụng 3 công cụ thông minh nhân tạo: mạng nơron, logic mờ (thuật toán Backprop, ANFIS) và thuật toán di truyền cho phép giải quyết bài toán tối ưu hóa công thức và quy trình sản xuất. Kết quả đạt được bước đầu được trình bày dưới đây.

Công thức Kem Vitamin E – Nghệ được thiết kế và tối ưu hóa bằng phần mềm INForm 3.0 (chương trình tối ưu công thức, quy trình của hãng IntelligenSys (<http://www.intelligensys.co.uk>)) và đã được sản xuất thử nghiệm.

Dữ liệu thực nghiệm:

x_1 : lượng dầu %	y_1 : độ dày mỏng trung bình
x_2 : lượng Cetostearyl Alcohol	y_2 : thời gian tách lớp phút
x_3 : lượng Polawax	y_3 : kích thước tiểu phân
x_4 : lượng tween 80	y_4 : độ Ph
x_5 : lượng spam 80	

điều kiện ràng buộc:

y_1 : TENT (Mid1 = 12; Mid2 = 12.5)	y_3 : DOWN (Mid1 = Mid2 = 0.585)
y_2 : UP (Mid1 = Mid2 = 57)	y_4 : TENT (Mid1 = 6.5; Mid2 = 6.7)

Kết quả tối ưu của INForm 3.0:

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
143.4	90	40	4.9	0.4	12.41	77.07	0.54	6.71

Kết quả sản xuất thực nghiệm (sản xuất thử 2 lần)

INForm 3.0	lần 1	lần 2	Trung bình
12.41	13.98	12.82	13.4
77.07	72	78	75
0.54	0.59	0.55	0.57
6.71	6.32	6.68	6.5

Kiểm tra khả năng dự đoán của chương trình qua kết quả tối ưu của INForm 3.0 và kết quả sản xuất thực nghiệm.

Dùng giá trị biến độc lập của INForm 3.0 kiểm tra khả năng dự đoán của chương trình cài đặt với thực nghiệm.

Kết quả dự đoán:

INForm 3.0	Lần 1	Lần 2	trung bình	BackProp	ANFIS
12.41	13.98	12.82	13.4	12.54	13.423
77.07	72	78	75	78.06	80.38
0.54	0.59	0.55	0.57	0.54	0.803
6.71	6.32	6.68	6.5	6.58	6.801

Kết quả thống kê phân tích phương sai hai yếu tố không lặp cho từng cặp kết quả:

Trung bình và BackProp: $F = 0.429 < F_{0.05} = 10.128$

INForm 3.0 và Trung bình: $F = 0.237 < F_{0.05} = 10.128$

INForm 3.0 và BackProp: $F = 0.956 < F_{0.05} = 10.128$

INForm 3.0 và ANFIS: $F = 2.489 < F_{0.05} = 10.128$

Trung bình và ANFIS: $F = 1.303 < F_{0.05} = 10.128$

Chương trình cài đặt có kết quả dự đoán và kết quả thực nghiệm trung bình của hai lô kem không có sự khác nhau về mặt thống kê.

Kiểm tra kết quả tối ưu của chương trình và của INForm 3.0

So sánh kết quả với kết quả tối ưu bởi INForm 3.0:

	INForm 3.0	BackProp	ANFIS
x_1	143.4	142.65	120
x_2	90	89.56	90
x_3	40	39.764	40
x_4	4.9	5.811	6
x_5	0.4	0.474	0.4
y_1	12.41	12.319	12.33
y_2	77.07	77.767	75
y_3	0.54	0.515	0.52
y_4	6.71	6.641	6.57

Kết quả thống kê, dùng phân tích phương sai hai yếu tố không lặp:

INForm 3.0 và BackProp (giá trị X):	$F = 0.101 < F_{0.05} = 7.708$
INForm 3.0 và BackProp (giá trị Y):	$F = 0.453 < F_{0.05} = 10.128$
INForm 3.0 và ANFIS (giá trị X):	$F = 0.885 < F_{0.05} = 7.708$
INForm 3.0 và ANFIS (giá trị Y):	$F = 1.344 < F_{0.05} = 10.128$

Chương trình cài đặt có kết quả tối ưu và kết quả tối ưu của INForm 3.0 không có sự khác nhau về mặt thống kê (BackProp: $F = 0.453 < F_{0.05} = 10.128$ và ANFIS: $F = 1.344 < F_{0.05} = 10.128$). Tương tự, kết quả của biến độc lập tương ứng cũng không khác nhau về mặt thống kê (BackProp: $F = 0.101 < F_{0.05} = 7.708$; ANFIS: $F = 0.885 < F_{0.05} = 7.708$).

5. Kết luận

Mặc dù lý thuyết mạng nơron, logic mờ và thuật toán di truyền đã được giới thiệu khá lâu, nhưng ứng dụng của nó vẫn được quan tâm và phát triển đến ngày nay; ứng dụng mạng nơron, logic mờ và thuật toán di truyền giải quyết bài toán tối ưu hóa công thức và quy trình sản xuất được phẩm là một ví dụ. Việc kết hợp các kỹ thuật giải quyết khá tốt bài toán đặt ra, giúp cho các dược sĩ tiết kiệm công sức và thời gian hơn so với việc sử dụng phương pháp truyền thống như thống kê, simplex...

A COMBINATION OF NEURAL NETWORKS, FUZZY LOGIC AND GENETIC ALGORITHMS FOR SOLVING THE OPTIMIZATION PROBLEMS IN PHARMACEUTICS

Le Hoai Bac⁽¹⁾, Hoang Kiem⁽¹⁾, Do Quang Duong⁽²⁾

⁽¹⁾ Faculty of Information Technology - University of Natural Sciences – VNU-HCM

⁽²⁾ Faculty of Pharmacy - University of Medicine and Pharmacy Ho Chi Minh City

ABSTRACT: In the pharmaceutical market, all products have a life cycle and out-of-date products should be improved or innovated by new ones. For this reason, design and optimization of formulation are the regular demands. Traditional methods of design and optimization - such as statistics, simplex, ... - have only used for simple and linear data. In the event that data is complicated and non-linear, these methods are not suitably.

This paper presents a solution for optimizing formulation and process of production in pharmaceuticals by combining of AI technics: neural networks, fuzzy logic and genetic algorithms. This solution helps pharmacist save a lot of time and labor because of its accuracy and processing speed.

TÀI LIỆU THAM KHẢO

- [1]. Đặng Văn Giáp, *Thiết kế & tối ưu hóa công thức và quy trình-Áp dụng phần mềm thông minh*, Nhà xuất bản y học, (2002).
- [2]. Hoàng Kiếm và Lê Hoàng Thái, *Thuật giải di truyền: Cách giải tự nhiên các bài toán trên máy tính*, Nhà xuất bản giáo dục, (2000).

- [3]. Nauck D. and Kruse R., *Neuro_Fuzzy Systems for Function Approximation*, Faculty of Computer Science_ Neural and Fuzzy Systems, Otto_von_Guericke_University of Magdeburg, Universitaetsplatz 2, D-39106 Magdeburg, Germany, (1997).
- [4]. Lee H., Chen C. and Huang T., *Learning efficiency improment of backpropagation algorithm by error saturation prevention method*, National Taiwan University of Science and Technology, Department of Electronic Engineering, (2001).
- [5]. Jantzen J., *Neurofuzzy Modelling*, Department of Automation, University of Denmark, report no 98-H-874, (1998).
- [6]. Jang J.R., Sun C. and Mizutani E., *Neuro-fuzzy and Soft Computing*, A Computation Approach To Learning And Machine Intelligence, Prentice Hall, Upper Saddle River, (1997).
- [7]. Kasabov N.K., Kim J.S., Gray A.R. and Watts M.J., *FuNN - A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition*, Department of INForm 3.0ation Science, University of Otago, P.O.Box 56, Dunedin, New Zealand.
- [8]. Bonissone P.P., *Adaptive Neural Fuzzy Inference Systems (ANFIS): Analysis and Applications*, GE CRD, Schenectady, NY USA, (1997).
- [9]. Fullér R., *Neural Fuzzy Systems*, Abo Akademi University, ISBN 951-650-624-0, ISSN 0358-5654, (1995).
- [10]. Schiffmann W., Joost M. and Werner R., *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*, University of Koblenz, Institute of Physics, Rheinau 156075 Koblenz, (1994).