PHẦN MỀM TÍNH TOÁN KHOA HỌC RST2ANU
GIẢI BÀI TOÁN TỐI ƯU TOÀN CỤC

Scientific computing software RST2ANU **for the global optimization problem**

Nguyễn Hải Thanh, Đặng Xuân Hà

SUMMARY

This paper presents RST2ANU software version 1.0 built by the authors to solve the global optimization problem. The software is based on the controlled random search algorithm incorporating simulated annealing as proposed by Mohan and Nguyen Hai Thanh. Written in Microsoft Visual C++ environment with protection against illegal copy, RST2ANU software provides friendly user interface allowing convenient input, manipulation and store of data. It has been used for realistic optimization problems in agricultural fields like resource use planning and management, determination of optimal farm household investment structure and crop pattern transformation.

Key words: Global optimization, controlled random search, scientific computing software.

TÓM TẮT

Bài báo này giới thiệu phần mềm RST2ANU phiên bản 1.0 được các tác giả xây dựng nhằm giải bài toán tối ưu phi tuyến toàn cục dựa trên thuật giải tìm kiếm ngẫu nhiên có điều khiển kết hợp với thuật toán mô phỏng quá trình tôi của vật liệu của C. Mohan và Nguyễn Hải Thanh. Phần mềm được viết bằng công cụ lập trình Microsoft Visual C++ 6.0, có giao diện thân thiện cho phép người dùng nhập liệu, xử lý và lưu trữ kết quả một cách thuận tiện cũng như có khả năng chống sao chép. Phần mềm RST2ANU đã được sử dụng để giải các bài toán tối ưu thực tế trong một số lĩnh vực nông nghiệp như quy hoạch sử dụng và quản lý tài nguyên, xác định cơ cấu đầu tư nông hộ, chuyển đổi cơ cấu cây trồng.

Từ khoá: Tối ưu toàn cục, tìm kiếm ngẫu nhiên có điều khiển, phần mềm tính toán khoa học.

1. ĐẶT VẤN ĐỀ

Tối ưu hoá là một trong những lĩnh vực kinh điển của toán học có ảnh hưởng đến hầu hết các lĩnh vực, trong đó có nông nghiệp. Trong thực tế, việc tìm ra giải pháp tối ưu cho một vấn đề nào đó chiếm một vai trò hết sức quan trọng. Phương án tối ưu là những phương án tốt nhất, tiết kiệm chi phí, tài nguyên, sức lực mà lại cho hiệu quả cao. Mô hình tối ưu tổng quát, hay bài toán tối ưu tổng quát, có dạng $F(X) \rightarrow Min\ (Max)$ *với* $X \in D \subset R^n$.

F ở đây có thể là một hàm vô hướng hay hàm véc tơ, tuyến tính hay phi tuyến. Trong trường hợp F là hàm vô hướng thì ta có mô hình tối ưu đơn mục tiêu, còn nếu F là hàm véc tơ thì có mô hình tối ưu đa mục tiêu. D được gọi là miền ràng buộc hay miền phương án khả thi, thường được biểu diễn bởi các đẳng thức và / hoặc các bất đẳng thức.

Dạng chính tắc của bài toán tối ưu toàn cục được biểu diễn như sau (Bazarra M. S. và Shetty C. M., 1984; Mohan C. và Nguyễn Hải Thanh, 1999; 2001):

Min (Max) $f(X)$, $X = (x_1, x_2, \ldots, x_n) \in R^n$, *với các điều kiện ràng buộc*

(i) $g_j(X) \le 0, \quad j = 1, 2, \ldots, k,$

(ii) $g_j(X) = 0, \quad j = k+1, k+2, \ldots, m,$

Trong các bài toán thực tế có thể bổ sung thêm các ràng buộc

(iii) $a_i \le x_i \le b_i, \quad i = 1, 2, \ldots, n.$

Trong trường hợp hàm mục tiêu f(X) hay có ít nhất một trong các hàm ràng buộc $g_j(X)$, j = 1, 2, …, m, là hàm phi tuyến, chúng ta có bài toán tối ưu phi tuyến. Khi tất cả các toạ độ x_i đều bắt buộc nhận các giá trị nguyên, i = 1, 2, …, n, thì ta có bài toán tối ưu nguyên. Còn nếu

chỉ có một số toạ độ (nhưng không phải tất cả các toạ độ) bắt buộc nhận giá trị nguyên thì ta có bài toán tối ưu hỗn hợp nguyên.

Ký hiệu D là miền các phương án (miền ràng buộc) cho bởi các ràng buộc (i), (ii) và / hoặc (iii) thì bài toán tối ưu trên đây có thể viết gọn hơn như sau: $f(X) \rightarrow Min\ (Max)\ với\ X \in D$.

Lúc này, đối với bài toán cực tiểu hoá, $X^* \in D$ được gọi là phương án tối ưu toàn cục nếu $\forall\ X \in D$ ta luôn có: $f(X^*) \leq f(X)$. Trong trường hợp $f(X^*) \leq f(X)$ chỉ đúng với $\forall X \in D$ trong một lân cận nào đó của X^* thì X^* được gọi là phương án tối ưu địa phương. Một cách tương tự, có thể định nghĩa khái niệm phương án tối ưu toàn cục / địa phương cho bài toán cực đại hoá. Nếu chúng ta chỉ quan tâm tới việc tìm kiếm phương án tối ưu toàn cục thì có bài toán tối ưu toàn cục.

Các phương pháp giải bài toán tối ưu toàn cục phi tuyến đơn mục tiêu được phân ra thành hai lớp (Bazarra M. S. và Shetty C. M., 1984; Nguyễn Hải Thanh và cs, 1998; 2003; 2005): phương pháp tất định và phương pháp ngẫu nhiên (deterministic and stochastic methods). Phương pháp tất định sử dụng các tính chất giải tích của hàm mục tiêu và các hàm ràng buộc. Một số dạng bài toán tối ưu toàn cục với những tính chất giải tích nhất định của hàm mục tiêu và các hàm ràng buộc có thể giải được bằng các phương pháp tất định thích hợp, chẳng hạn như phương pháp quy hoạch toàn phương, quy hoạch tách, qui hoạch lồi, quy hoạch d.c… Trong các trường hợp đó phương án tối ưu toàn cục có thể tìm được sau một số hữu hạn bước tính toán với độ chính xác chọn trước.

Tuy nhiên, đối với nhiều lớp bài toán tối ưu toàn cục phương pháp tất định tỏ ra không có hiệu quả. Trong khi đó, các phương pháp ngẫu nhiên như: phương pháp đa khởi tạo (*multistart*), mô phỏng tôi (*simulated annealing*), thuật giải di truyền (*genetic algorithm*)… có thể áp dụng để giải các bài toán tối ưu toàn cục dạng bất kỳ, không đòi hỏi các tính chất đặc biệt của hàm mục tiêu hay các hàm ràng buộc. Các phương pháp ngẫu nhiên đặc biệt tỏ ra có hiệu quả đối với các bài toán tối ưu phi tuyến nguyên và hỗn hợp nguyên. Tuy nhiên, các phương pháp này thường chỉ cho phương án "gần" tối ưu khá tốt sau một số hữu hạn bước mà không kiểm soát được độ chính xác của phương án tìm được.

Như vậy, hiện tại có nhiều phương pháp tối ưu toàn cục được đề xuất. Tuy nhiên chưa có một phương pháp nào tỏ ra hữu hiệu cho mọi bài toán tối ưu, đặc biệt là các bài toán tối ưu với biến nguyên hay hỗn hợp nguyên. Hơn nữa, các phương pháp tối ưu cần phải được lập trình để đóng gói thành các phần mềm thân thiện đối với người sử dụng. Đây là một đòi hỏi rất thực tế của các kĩ sư, các nhà khoa học, các doanh nghiệp trong nhiều lĩnh vực công nghiệp cũng như nông nghiệp. Trong bài báo này, chúng tôi trình bày một phần mềm tính toán khoa học (RST2ANU) có thể đáp ứng được phần nào các đòi hỏi nêu trên đối với người sử dụng để giải các bài toán tối ưu phi tuyến toàn cục với các biến liên tục, nguyên hoặc hỗn hợp nguyên. Phần mềm này được xây dựng dựa trên phương pháp tìm kiếm ngẫu nhiên có điều khiển (cùng tên gọi RST2ANU) do Mohan và Nguyễn Hải Thanh đề xuất. Đây là một phương pháp tối ưu đã được chạy kiểm thử trên hàng trăm bài toán mẫu và nhiều bài toán thực tế với độ tin cậy rất cao và tốc độ tính toán chấp nhận được.

2. THUẬT GIẢI TÌM KIẾM NGẪU NHIÊN CÓ ĐIỀU KHIỂN RST2ANU

Thuật giải RST2ANU là thuật giải lặp, bao gồm hai pha, pha địa phương và pha toàn cục (Mohan C. và Nguyễn Hải Thanh, 1999; 2001).

Trong pha toàn cục, một số lượng thích hợp đủ lớn các phương án chấp nhận được được phát sinh ra một cách ngẫu nhiên và lưu trữ trong mảng có tên A. Đánh dấu hai điểm có giá trị hàm mục tiêu lớn nhất và nhỏ nhất tương ứng là M và L.

Trong pha địa phương, các phương án được xử lý nhằm thu được giá trị ước lượng tốt hơn của hàm mục tiêu. Trong pha này, thuật giải xác định X* là điểm được nội suy bậc hai dựa trên phương án L và hai phương án khác được chọn ngẫu nhiên trong mảng A. Nếu như X* chấp

nhận được thì với f(X*) ≤ f(M), M sẽ được thay thế bởi X* trong mảng A, còn với f(X*)>f(M) M sẽ được thay thế bởi X* với xác suất ***p= exp(-β(f(X*)-f(M))/(f(X*)-f(L)))***, trong đó β >0 là tham số được lựa chọn thích hợp. Nếu X* không phải là phương án chấp nhận được, bỏ qua X* và chọn hai phương án khác trong A một cách ngẫu nhiên rồi cùng với L tiếp tục sinh ra phương án mới. Quá trình cứ thế tiếp diễn như vậy cho tới khi tập hợp các phương án trong A sẽ có xu hướng co cụm lại xung quanh một phương án tối ưu toàn cục.

Sơ đồ thuật giải RST2AN được thể hiện trên hình 1. Các ký hiệu trên sơ đồ được giải thích như sau:

- *n, f(X), g(j), a_i, b_i, ...* là các đầu vào.
- *A = RandomNSolution (N)* phát sinh N phương án ngẫu nhiên chấp nhận được, đồng thời tính giá trị của hàm mục tiêu và trả về kết quả cho mảng A. Như vậy, mảng A chứa luôn cả giá trị hàm mục tiêu tương ứng với từng phương án.
- *Arrange(A)* sắp xếp mảng A theo thứ tự tăng dần của hàm mục tiêu.
- *Max(A), Min(A)* trả về phương án có giá trị hàm mục tiêu lớn nhất và nhỏ nhất trong A.
- *Clustered(A, eps1, eps2)* cho biết mảng A đã hội tụ theo hàm mục tiêu hay chưa.
- Nếu (f(M) – f(L))/FM) < eps1 thì mảng A hội tụ, ngược lại chưa hội tụ, với FM = f(M) nếu f(M) > eps2, ngược lại FM = 1.
- *NewSolution()* trả về một phương án mới được suy ra từ 3 điểm: L và hai điểm được chọn ngẫu nhiên khác trong mảng A theo phương pháp nội suy.
- *Feas(X)* nhận giá trị TRUE nếu X chấp nhận được, ngược lại nhận giá trị FALSE
- *Random(0,1)* trả về giá trị ngẫu nhiên nằm trong khoảng (0,1).
- *Replace(A, M, X*)* thay thế M trong A bởi X* kèm theo cả giá trị hàm mục tiêu sao cho không cần phải sắp xếp lại mảng A mà vẫn đảm bảo các điểm được sắp xếp theo thứ tự giá trị hàm mục tiêu tăng dần.
- *Kết thúc 1:* Số lần tìm kiếm liên tiếp mà không cải thiện được giá trị hàm mục tiêu vượt quá số lần cho phép. Thuật giải dừng với giá trị tốt nhất của hàm mục tiêu tìm được là FL tương ứng với phương án L.
- *Kết thúc 2:* Phương án tối ưu toàn cục đã đạt được là L với giá trị hàm mục tiêu là FL.
- *Kết thúc 3:* Số lần nội suy liên tiếp mà không tìm được phương án thay thế M trong A vượt quá số lần cho phép. Thuật giải dừng với giá trị tốt nhất của hàm mục tiêu tìm được là FL tương ứng với phương án L.
- *Kết thúc 4:* Số lần lặp vượt quá số lần cho phép. Thuật giải dừng với giá trị tốt nhất của hàm mục tiêu tìm được là FL tương ứng với phương án L.

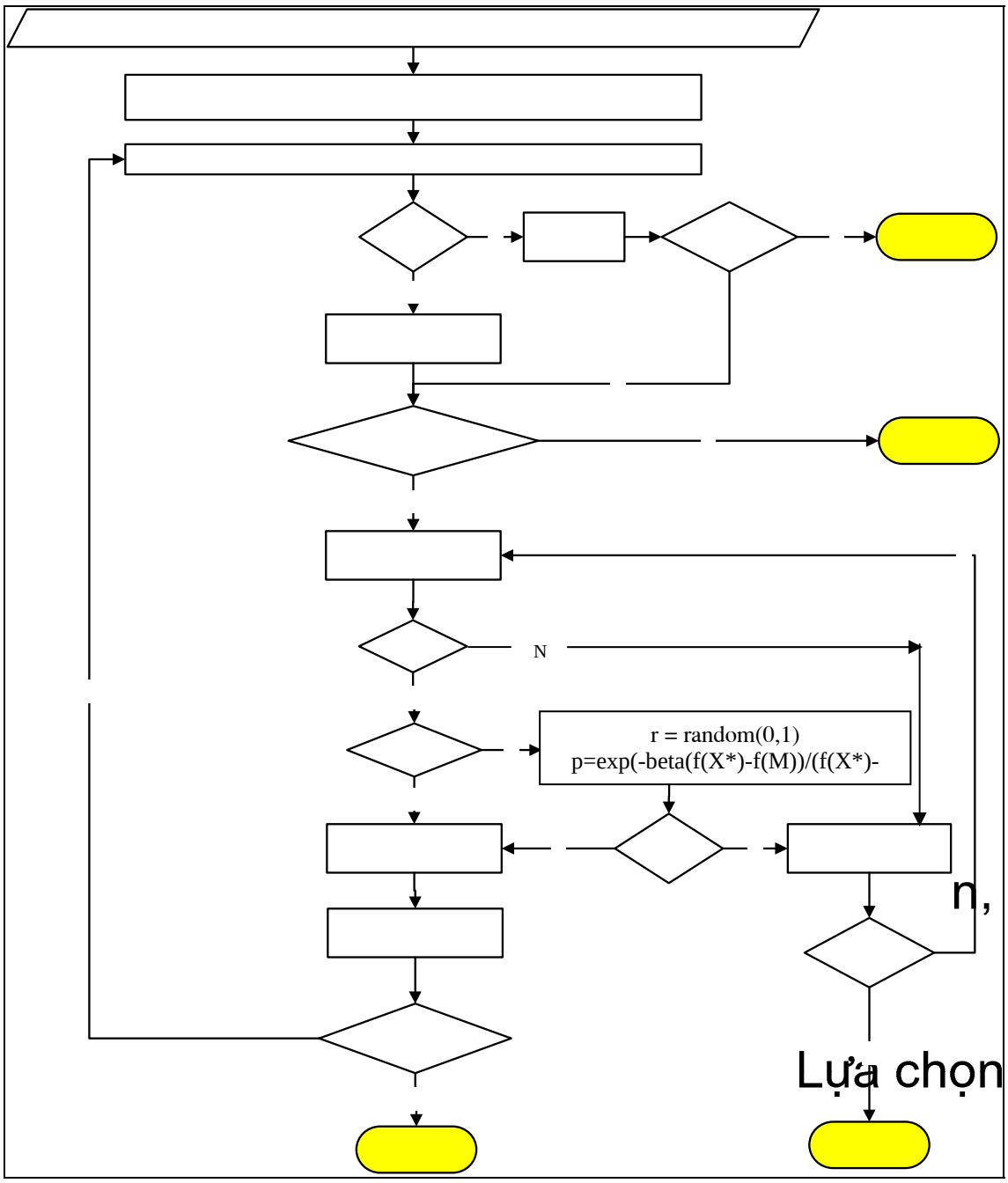
3. XÂY DỰNG PHẦN MỀM RST2ANU

Phần mềm RST2ANU phiên bản 1.0 được xây dựng nhằm giải bài toán tối ưu toàn cục phi tuyến với thuật giải nêu trên (Nguyễn Hải Thanh và cs, 2002; 2003). Công cụ được sử dụng là Microsoft Visual C++ 6.0. Các tài liệu chuyên khảo của Đỗ Xuân Lôi (1999) và Roger Pressman (1997) cũng đã được tham khảo trong quá trình xây dựng phần mềm.

Để tiến hành việc nhận dạng và tính giá trị hàm, tiện ích evaluateExpression được sinh ra từ AnaGram parser generator, một sản phẩm của Parsifal Software đã được sử dụng. evaluateExpression cho phép lượng giá một hay nhiều biểu thức liên quan được nhập vào thông qua các xâu ký tự, mỗi xâu có thể là một hay nhiều biểu thức được viết cách nhau bởi dấu chấm phảy (;).

Trong thực tế, nhiều bài toán có ràng buộc khá phức tạp, đặc biệt là các bài toán chứa ràng buộc dạng đẳng thức. Nếu chỉ sử dụng phương pháp gieo ngẫu nhiên để tạo ra các phương án thì khả năng thu được phương án chấp nhận được là khá thấp. Vì vậy, chương trình cho phép đưa vào các luật phát sinh phương án nhằm tạo ra các phương án đủ tốt đồng thời thoả mãn các ràng buộc.

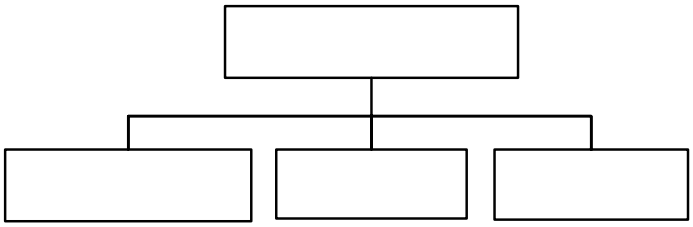
Phần mềm được xây dựng với mục đích cho phép người dùng nhập vào bài toán tối ưu toàn cục một cách dễ dàng, kết quả được lưu ra tệp dữ liệu.



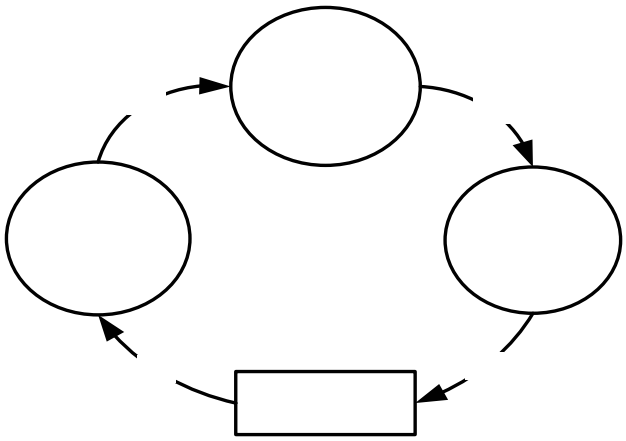
Hình 1. Sơ đồ thuật giải RST2ANU

Thiết kế chương trình

Phân cấp chức năng (hình 2)

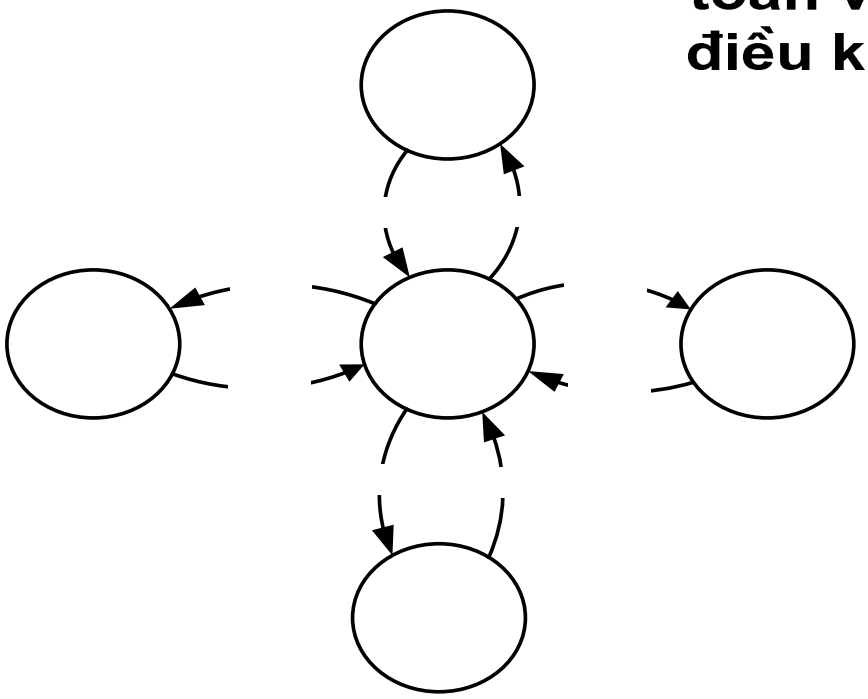


Hình 2. Phân cấp chức năng



Hình 3. Biểu đồ luồng dữ liệu mức khung cảnh

Nhập/phân tích bài toán và các tham số điều khiển giải thuật



Hình 4. Biểu đồ luồng dữ liệu mức đỉnh

Chú thích cho các chức năng

1-2

(1) Chức năng nhập dữ liệu, phân tích và chuẩn hoá dữ liệu cho bài toán:
- Hàm F(X) là một xâu ký tự (xem phần cú pháp).
- Các luật áp dụng trong quá trình phát sinh phương án là một xâu ký tự.
- Các ràng buộc cũng là một xâu ký tự.
- Số biến số và các điều kiện khác.

(2) Chức năng giải bài toán thực hiện giải bài toán theo thuật giải RST2ANU.

(2.1) Chức năng chính điều khiển thuật giải.

Nhập/phân tích dạng bài toán

(2.2) Phát sinh phương án với điều kiện xác định của biến.
(2.3) Kiểm tra tính chấp nhận được của phương án bao gồm 2 khâu:
- Áp đặt các luật lên phương án.
- Kiểm tra tính chấp nhận được của phương án.
(2.4) Nội suy phương án mới theo phương pháp nội suy bậc hai.
(2.5) Tính giá trị hàm mục tiêu ứng với phương án đã cho.
(3) Chức năng ghi nhận kết quả lưu kết quả ra tệp dữ liệu.

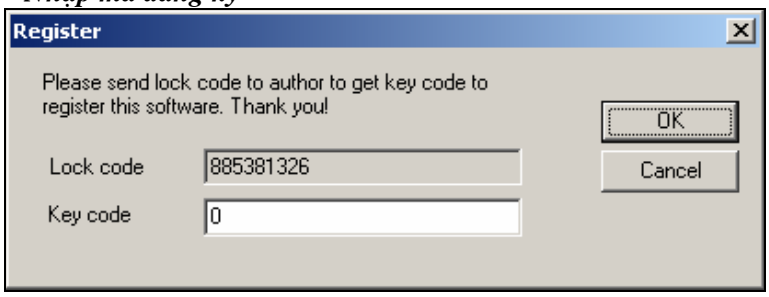
Chú thích cho luồng dữ liệu

(0-1) Dữ liệu bài toán và dữ liệu điều khiển thuật giải.
(3-0) Kết quả giải bài toán bằng thuật giải RST2ANU.
(1-2) Dữ liệu bài toán và các điều kiện khác sau khi được phân tích.
(2-3) Kết quả giải bài toán theo thuật giải RST2ANU.
(2.1-2.2) Yêu cầu phát sinh phương án.
(2.2-2.1) Phương án trả về hoặc báo hiệu không tìm được phương án thoả điều kiện.
(2.1-2.3) Phương án.
(2.3-2.1) Chấp nhận được hay không chấp nhận được, phương án trả về (sau khi đã bị biến đổi theo luật do người dùng nhập vào).
(2.1-2.4) Yêu cầu nội suy phương án mới.
(2.4-2.1) Phương án nội suy.
(2.1-2.5) Phương án.
(2.5-2.1) Giá trị hàm mục tiêu.

4. MỘT SỐ GIAO DIỆN SỬ DỤNG CỦA PHẦN MỀM RST2ANU

Khởi động chương trình: Chương trình được gói gọn trong một file chạy duy nhất mang tên rst2anu1.0.exe. Khi bắt đầu khởi động chương trình, người dùng sẽ được hỏi mã đăng ký sử dụng chương trình.

Nhập mã đăng ký



Hình 5. Nhập mã đăng ký

Mỗi người dùng sẽ được cấp một mã đăng ký và phải có mã đăng ký mới sử dụng được chương trình, do đó chương trình không thể bị sao chép. Sau khi nhập mã đăng ký (hình 5), các lần chạy sau, người dùng không cần phải nhập mã đăng ký nữa.

Ví dụ. Xét bài toán tối ưu phi tuyến toàn cục hỗn hợp nguyên

$$z = x_1^{0,6} + x_2^{0,6} + x_3^{0,4} + 2x_4 + 5x_5 - 4x_3 - x_6' \rightarrow \text{Min}$$

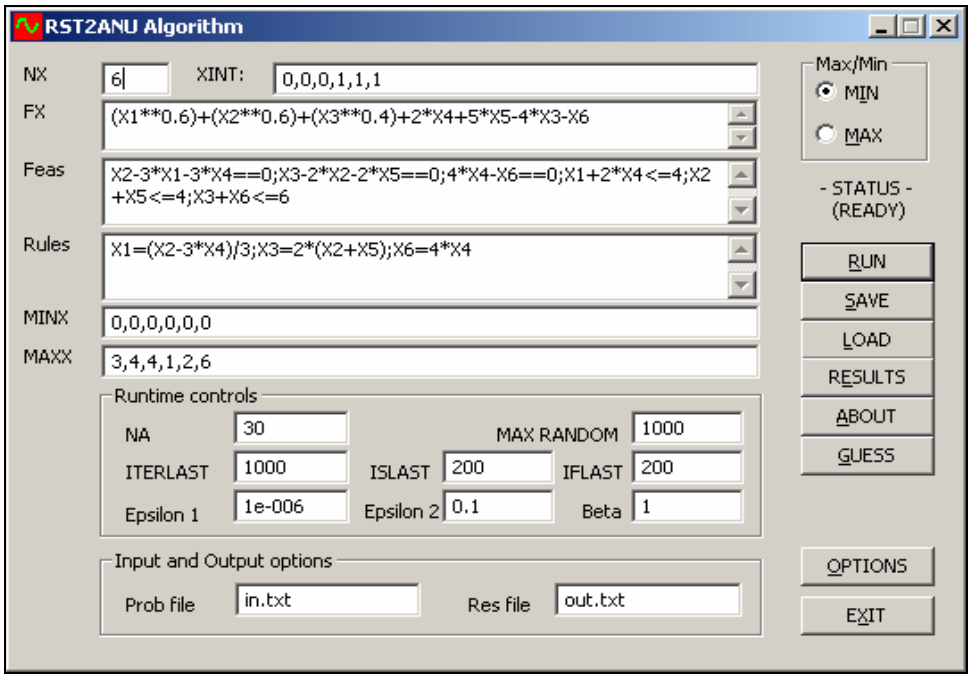
với các ràng buộc:

$$\begin{cases} x_2 - 3x_1 - 3x_4 = 0;\ x_3 - 2x_2 - 2x_5 = 0; \\ 4x_4 - x_6 = 0;\ x_1 + 2x_4 \le 4; \\ x_2 + x_5 \le 4;\ x_3 + x_6 \le 6; \\ x_1 \le 3;\ x_2 \le 4;\ x_3 \le 4;\ x_4 \le 1;\ x_5 \le 2;\ x_6 \le 6; \end{cases}$$

$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$; x_4, x_5, x_6 là các biến nguyên.

Nhập / lưu bài toán và các dữ kiện

Giao diện chương trình sau khi khởi động thành công như sau (hình 6):



Hình 6. Giao diện chương trình sau khi nhập bài toán

Thông qua giao diện này, người dùng có thể nhập bài toán một cách dễ dàng:

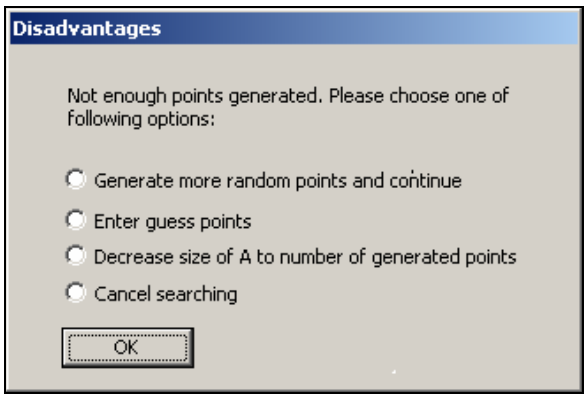
- NX là số biến của bài toán.
- XINT xác định biến nguyên và biến không nguyên. Như trong hình trên, XINT = 0,0,0,1,1,1 cho biết 3 biến đầu là biến thực, ba biến sau là biến nguyên.
- FX là xâu xác định hàm ràng buộc, được nhập theo cú pháp của evaluateExpression. Các biến được viết bằng ký hiệu "X" có kèm theo chỉ số. Ví dụ, X1 là biến thứ nhất, X5 là biến thứ 5.
- Nếu bài toán tối ưu là bài toán tìm cực tiểu thì lựa chọn ô MIN và ngược lại chọn ô MAX với bài toán tìm cực đại.
- Feas xâu cho biết các hàm ràng buộc, được nhập cách nhau bởi dấu chấm phẩy hoặc xuống dòng. Các xâu này cũng tuân theo cú pháp của evaluateExpression.
- Rules là các xâu chỉ ra các luật. ở đây, một luật có thể coi như là một lệnh gán giá trị của một biến bởi giá trị của một biểu thức các biến khác.
- MINX là mảng xác định cận dưới cho các biến, các giá trị viết cách nhau bởi dấu (,).
- MAXX là mảng xác định cận trên cho các biến, các giá trị viết cách nhau bởi dấu (,).
- NA là kích thước của mảng A.
- MAX RANDOM là số lần cố gắng tối đa để tìm một phương án chấp nhận được bằng phương pháp ngẫu nhiên.
- ITERLAST, ISLAST, IFLAST là các giới hạn về số vòng lặp, số lần thất bại trong việc cải thiện giá trị hàm mục tiêu, số lần thất bại trong việc nội suy phương án mới chấp nhận được.
- Epsilon1, epsilon2 là các số dương đủ nhỏ nhằm xác định tiêu chuẩn co cụm của mảng A theo thuật giải.
- Beta là hằng số sử dụng trong công thức tính xác xuất thay thế một phương án tốt hơn trong mảng A bởi một phương án tồi hơn.
- Prob file và Res file là các tệp đầu vào và tệp kết quả. Có thể soạn sẵn tệp bài toán đầu vào rồi nạp bài toán. Cũng có thể lưu một bài toán đã nhập ra tệp.

Chạy chương trình

Sau khi nhập bài toán hay nạp bài toán từ tệp, có thể chạy chương trình bằng cách kích chuột vào nút RUN. Trong khi chạy chương trình, ô trạng thái ở phía trên nút RUN sẽ xuất hiện dòng chữ SEARCHING. Khi thuật giải chạy xong thì ô trạng thái sẽ trở về READY cho biết đã sẵn sàng cho các bài toán tiếp theo. Trong quá trình chạy chương trình, nếu không phát sinh đủ số phương án chấp nhận được theo kích thước của mảng A, thuật giải sẽ tạm dừng và hỏi người dùng cách giải quyết. Người dùng có thể có một trong các lựa chọn sau:

- Yêu cầu tiếp tục tìm thêm phương án.
- Sử dụng số lượng phương án đã tìm được để giải quyết bài toán.
- Nhập thêm các phương án dự đoán (chương trình sẽ kiểm tra các phương án này xem có chấp nhận được không)
- Kết thúc thuật giải.

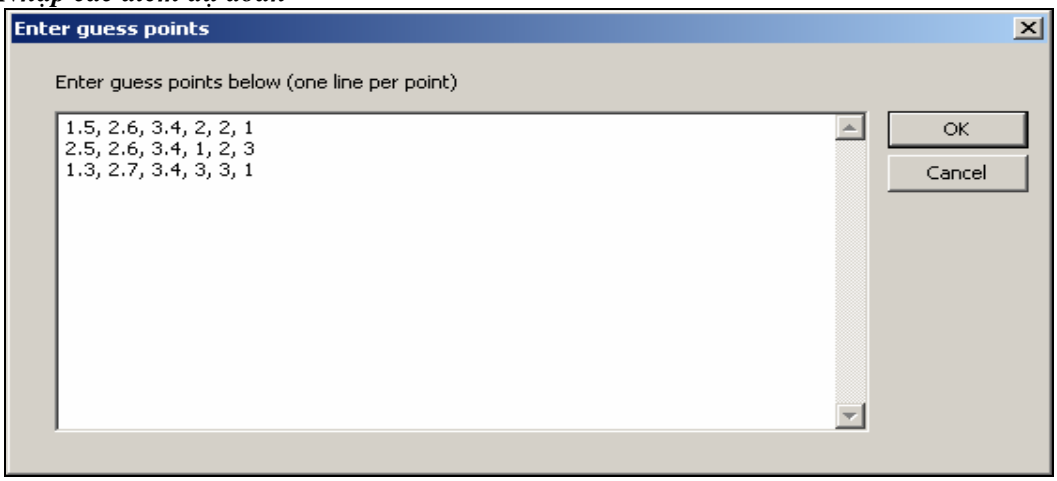
Xem hình 7 để biết thêm về tình huống không phát sinh đủ phương án



Hình 7. Tình huống phát sinh không đủ phương án

Người dùng cũng có thể nhập các điểm dự đoán trước khi chạy thuật giải. Kích chuột vào nút GUESS, giao diện nhập các điểm dự đoán cho phép nhập các phương án dự đoán vào các dòng, mỗi dòng là một phương án dự đoán.

Nhập các điểm dự đoán



Hình 8. Nhập các điểm dự đoán

Xem kết quả

Sau khi chạy xong chương trình, kết quả chạy sẽ được lưu ra file văn bản, bao gồm phương án tối ưu, giá trị hàm mục tiêu, mảng A,… có cấu trúc như trên hình 9.

```
**************************RESULTS*********************************
fx=(X1**0.6)+(X2**0.6)+(X3**0.4)+2*X4+5*X5-4*X3-X6
ITER = 158/1000; ISMAX = 76/200; IFMAX = 1001/1000
X*     = 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000
f(X*) = -11.959130

============================ARRAY A================================
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666667 2.000000 4.000000 0.000000 0.000000 0.000000 -11.959130
 0.666321 1.998962 3.997925 0.000000 0.000000 0.000000 -11.951906
 0.663472 1.990417 3.980834 0.000000 0.000000 0.000000 -11.892427
 0.662414 1.987243 3.974487 0.000000 0.000000 0.000000 -11.870339
 0.659851 1.979553 3.959105 0.000000 0.000000 0.000000 -11.816829
 0.640848 1.922544 3.845088 0.000000 0.000000 0.000000 -11.420634
 0.640116 1.920347 3.840693 0.000000 0.000000 0.000000 -11.405380
```

Hình 9. Cấu trúc file kết quả

Bảo vệ, chống sao chép phần mềm

Để đảm bảo phần mềm viết ra không bị sao chép trái phép, chúng tôi sử dụng phương pháp bảo vệ phần mềm bởi mã đăng ký sử dụng. Căn cứ trên các thông tin thu thập từ máy tính của người dùng (chỉ với mục đích bảo vệ phần mềm), chúng tôi tiến hành mã hoá và tạo ra mã đăng ký cho từng người dùng trên từng máy tính (Nguyễn Hải Thanh và cs, 2003; 2005).

5. KẾT LUẬN

Phần mềm tính toán khoa học RST2ANU đã được thiết kế và xây dựng có thể sử dụng để giải quyết nhiều mô hình tối ưu phát sinh trong lĩnh vực nông nghiệp, hỗ trợ cho giảng dạy và nghiên cứu khoa học nông nghiệp cũng như trong các lĩnh vực khác.

Phần mềm này là thân thiện với người sử dụng và đã được đóng gói tránh sao chép, có thể được phổ cập có bản quyền một cách rộng rãi. Việc tạo ra các giao diện thân thiện cho phép dễ dàng nhập các hàm mục tiêu và ràng buộc của nhiều dạng bài toán tối ưu phi tuyến là một vấn đề khá phức tạp đã được giải quyết thành công trong phần mềm này.

Trong tình hình hiện tại, khi các phần mềm tối ưu phi tuyến không có sẵn trên thị trường trong và ngoài nước, phần mềm RST2ANU nên được triển khai sử dụng để giải quyết các bài toán tối ưu trong các lĩnh vực khác nhau, bao gồm cả các bài toán nguyên và hỗn hợp nguyên (Nguyễn Hải Thanh, 1997; 1998). Các nghiên cứu cần tiếp tục được triển khai và được hỗ trợ về mặt tài chính để tích hợp RST2ANU vào các gói phần mềm trong điều khiển tự động hóa hay trong các hệ hỗ trợ ra quyết định.

Tài liệu tham khảo

Bazarra M. S. and Shetty C. M. (1984), *Nonlinear programming: Theory and Algorithms,* John Wiley & Sons, New York.

Đỗ Xuân Lôi (1999), Cấu trúc dữ liệu và giải thuật, Nxb Thống kê.

Mohan C. and Nguyen Hai Thanh (1999), "A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems", Computational Optimization and Applications, Vol. 14, pp. 103-132.

Mohan C. and Nguyen Hai Thanh (2001), "An interactive satisficing method for solving multiobjective mixed fuzzy-stochastic programming problems", International Journal for Fuzzy Sets and Systems, Vol. 117, No.1, pp. 61-79.

Pressman R. S. (1996), Software Engineering, McGraw-Hill.

Nguyễn Hải Thanh (1997), "Một số mô hình tối ưu dùng trong nông nghiệp", Kết quả nghiên cứu khoa học Trường ĐHNN I Hà Nội, Quyển 3, trang 228-236, Nxb Nông nghiệp.

Nguyễn Hải Thanh, Trần Thị Huyền, Lê Thị Duyên (2002), "Xây dựng phần mềm tối ưu đa mục tiêu giải các bài toán thực tế", Tóm tắt báo cáo Hội nghị Toán học toàn Việt Nam lần thứ 6, trang 146, Huế, 9/2002.

Nguyễn Hải Thanh (2003), "Xây dựng hệ phần mềm máy tính phục vụ giảng dạy và nghiên cứu khoa học nông nghiệp", Báo cáo tổng kết đề tài cấp Bộ, Bộ Giáo dục và Đào tạo, mã số B2001-32-23.

Nguyen Hai Thanh (1998), "Optimization in fuzzy-stochastic environment and its applications in industry and economics", Proceedings of VJFUZZY'98: Vietnam-Japan biletral symposium on fuzzy systems and applications, pp. 342-349, 30/9 – 2/10/1998.

Nguyễn Hải Thanh (chủ biên), Đỗ Thị Mơ, Đặng Xuân Hà và các tác giả khác (2005), Tin học ứng dụng trong ngành nông nghiệp, Nxb Khoa học và Kỹ thuật.