ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TT & TT

ĐÀO NGỌC TUẤT

KẾT HỢP GIẢI THUẬT DI TRUYỀN VÀ LOGIC MỜ GIẢI BÀI TOÁN TỐI ƯU ĐA MỤC TIÊU

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Thái Nguyên, tháng 10 - 2012

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TT& TT

ĐÀO NGỌC TUẤT

KẾT HỢP GIẢI THUẬT DI TRUYỀN VÀ LOGIC MỜ GIẢI BÀI TOÁN TỐI ƯU ĐA MỤC TIÊU

Chuyên ngành: Khoa học máy tính
Mã số: 60 48 01

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. Vũ Mạnh Xuân

Thái Nguyên, tháng 10 -2012

MỞ ĐẦU

Nhiều bài toán tối ưu trong thực tế là bài toán tối ưu đa mục tiêu, đặc biệt là trong thiết kế. Chẳng hạn người ta muốn thiết kế sản phẩm sao cho chi phí thấp, tiết kiệm nguyên liệu nhưng chất lượng tốt, hoặc thiết kế một bể chứa nước với yêu cầu dung lượng lớn mà chi phí thấp…. Đã có nhiều nhà toán học và tin học đã nghiên cứu về vấn đề này và đưa ra nhiều phương pháp giải khác nhau. Một số những phương pháp thường được sử dụng để giải bài toán đa mục tiêu như: phương pháp nhượng bộ dần, phương pháp tìm nghiệm có khoảng cách ngắn nhất đến nghiệm lý tưởng, phương pháp trọng số,….

Được sự đồng ý của Hội đồng Khoa học khoa Công Nghệ Thông Tin, cùng sự hướng dẫn của thầy giáo **Vũ Mạnh Xuân**, em chọn đề tài khóa luận của mình nhằm nghiên cứu một phương pháp tiếp cận khác để giải bài toán tối ưu đa mục tiêu là sử dụng giải thuật di truyền kết hợp logic mờ.

Mục đích nghiên cứu: tìm hiểu một số phương pháp giải bài toán tối ưu đa mục tiêu, giải thuật di truyền và logic mờ, trên cơ sở đó sử dụng giải thuật di truyền kết hợp với logic mờ để giải bài toán tối ưu đa mục tiêu.

Nội dung của đề tài: gồm 3 chương

1) Chương 1. Giải thuật di truyền
2) Chương 2. Logic mờ
3) Chương 3. Kết hợp giải thuật di truyền và logic mờ giải bài toán tối ưu đa mục tiêu

Để tiến hành nghiên cứu đề tài này, em đã sử dụng phối hợp một số phương pháp như: phương pháp nghiên cứu tài liệu (nghiên cứu tài liệu về các giải thuật di truyền, bài toán tối ưu đa mục tiêu, logic mờ, ngôn ngữ lập trình matlab 7.0); phương pháp lấy ý kiến chuyên gia (giáo viên hướng dẫn, tham khảo trên mạng).

Khi thực hiện đề tài này, bước đầu em đã tìm hiểu được một số phương pháp giải bài toán tối ưu đa mục tiêu, một số vấn đề về giải thuật di truyền, logic mờ. Kết quả là đã đề xuất một kỹ thuật kết hợp giải thuật di truyền với logic mờ để giải bài toán tối ưu đa mục tiêu và đã lập trình thử nghiệm trên một số bài toán cụ thể.

Chương 1: GIẢI THUẬT DI TRUYỀN

Chương này giới thiệu những vấn đề khái quát về giải thuật di truyền (GA) làm cơ sở cho việc ứng dụng giải bài toán tối ưu đa mục tiêu.[2], [6]

1.1. Khái quát chung

Giải thuật di truyền GA(GENETIC ALGORITHM) do D.E. Goldberg đề xuất, sau đó được L. Davis và Z. Michalevicz phát triển, đây cũng chính là một trong các thuật toán tiến hóa. Thuật toán tiến hóa là các chương trình máy tính có dùng các thuật toán tìm kiếm, tối ưu hóa dựa trên nguyên lý tiến hóa tự nhiên.

Giải thuật di truyền được hình thành dựa trên quan niệm: quá trình tiến hóa tự nhiên là quá trình hoàn hảo và hợp lý nhất, tự quá trình này đã mang tính tối ưu. Quan niệm này là một tiên đề đúng, không chứng minh được nhưng phù hợp với thực tế khách quan. Tính tối ưu của quá trình tiến hóa thể hiện ở đặc điểm, thế hệ sau bao giờ cũng tốt hơn (phát triển hơn, hoàn thiện hơn) thế hệ trước. Tiến hóa tự nhiên được duy trì nhờ hai quá trình cơ bản là sinh sản và chọn lọc tự nhiên, trong suốt quá trình tiến hóa tự nhiên, các thế hệ mới luôn được sinh ra để bổ sung thay thế thế hệ cũ. Cá thể nào phát triển hơn, thích ứng hơn với môi trường sẽ tồn tại, cá thể nào không thích ứng được với môi trường sẽ bị đào thải. Sự thay đổi của môi trường là động lực thúc đẩy quá trình tiến hóa, ngược lại tiến hóa cũng tác động trở lại góp phần thay đổi môi trường.

Giải thuật di truyền (GA-Genetic Algorithms) là giải thuật tìm kiếm, chọn lựa các giải pháp tối ưu để giải quyết các bài toán thực tế khác nhau, dựa trên cơ chế chọn lọc của tự nhiên: từ tập lời giải ban đầu, thông qua nhiều bước tiến hoá, hình thành tập lời giải mới phù hợp hơn, và cuối cùng dẫn đến lời giải tối ưu toàn cục.

Trong tự nhiên, mỗi cá thể muốn tồn tại và phát triển phải thích nghi với môi trường, cá thể nào thích nghi hơn thì tồn tại, cá thể nào kém thích nghi thì bị tiêu diệt. Trong mỗi cá thể, các gen liên kết với nhau theo cấu trúc dạng chuỗi, gọi là nhiễm sắc thể (NST). Mỗi NST đặc trưng cho mỗi loài và quyết định sự sống còn của cá thể đó. Do môi trường tự nhiên luôn biến đổi nên cấu trúc NST cũng thay đổi để thích nghi với môi trường và thế hệ sau luôn thích nghi hơn thế hệ trước. Cấu trúc này có được do sự trao đổi thông tin có tính ngẫu nhiên với môi trường bên ngoài hoặc giữa các NST với nhau.

1.2. Các vấn đề cơ bản của giải thuật di truyền

1.2.1. Mã hóa

Việc mô tả di truyền cho lời giải cho bài toán gồm hai phần cơ bản:

+ Xây dựng cấu trúc gen cho mỗi lời giải của bài toán để từ mỗi lời giải ta có thể mã hoá thành một NST (chuỗi các gen).

+ Giải mã các NST để nhận được lời giải.

Đây là vấn đề cần giải quyết trước khi giải bài toán với GA. Tuỳ thuộc vào nội dung của mỗi bài toán mà ta có cách mã hoá khác nhau.

Sau đây là phương pháp mã hoá hay được sử dụng:

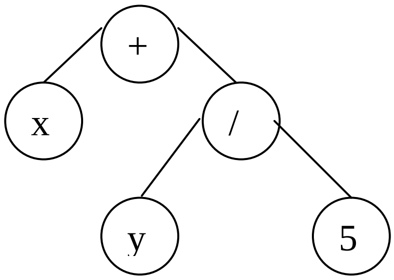
Mã hoá dạng chuỗi nhị phân: đây là phương pháp thông dụng và cơ bản nhất được sử dụng ngay từ bước ban đầu khi nghiên cứu GA. Trong phương pháp này mỗi NST là một chuỗi các bit 0 và 1.

Mã hoá thứ tự: được sử dụng trong bài toán có sắp xếp thứ tự. Ở đây mỗi NST là một chuỗi các số nguyên thể hiện thứ tự phân bố lời giải của bài toán.

Mã hoá theo giá trị: được sử dụng trong các bài toán mà mỗi lời giải là tập các giá trị (ví dụ tập số thực). Trong phương pháp này, mỗi NST là một chuỗi các giá trị có mối quan hệ tương ứng với bài toán.

Mã hoá dạng cây: được sử dụng chủ yếu trong các biểu thức toán học, trong phương pháp mã hoá này mỗi NST là một cây của một nhóm đối tượng nào đó.

Ví dụ: Biểu thức sau x+(y / 5) được mã hoá thành:



Mã hoá số thực : Mỗi NST được mã hoá là một véc tơ trong không gian R^m chẳng hạn $X = (a_1, a_2, ..., a_m)$ với các $a_i \in R$. Cách mã hoá này thường tự nhiên đối với các bài toán tối ưu số và được phát triển rất mạnh trong thời gian gần đây.

1.2.2. Tạo lập lời giải ban đầu (khởi tạo quần thể)

Tập lời giải ban đầu thường được khởi tạo ngẫu nhiên từ miền xác định của các lời giải. Cách tạo lập tập lời giải ban đầu phụ thuộc rất nhiều vào cách mã hoá NST.

Với phương pháp mã hoá nhị phân: xây dựng NST bằng cách tạo ngẫu nhiên chuỗi các bit 0 hoặc 1.

Với phương pháp mã hoá thứ tự: xây dựng NST ban đầu bằng cách hoán vị ngẫu nhiên các thứ tự.

Với phương pháp mã hoá theo giá trị: tạo ngẫu nhiên từng giá trị trong miền xác định của lời giải để tạo ra chuỗi NST ban đầu.

Với mã hoá số thực: tạo ngẫu nhiên N véc tơ thực trong R^m.

1.2.3. Xây dựng hàm phù hợp

Hàm phù hợp đánh giá khả năng phù hợp của tập lời giải theo yêu cầu bài toán. Hàm này được xây dựng cho từng bài toán với yêu cầu cụ thể. Thông thường trong các bài toán tối ưu hàm này chính là hàm mục tiêu của bài toán.

1.2.4. Các toán tử di truyền

a. *Toán tử chọn lọc*

Trong quá trình thực hiện của giải thuật di truyền, sau mỗi lần tiến hoá ta chỉ giữ lại các cá thể có độ phù hợp cao còn các cá thể phù hợp thấp bị loại bỏ. Toán tử chọn lọc thường giữ lại 50% các cá thể phù hợp nhất. Tuy nhiên người ta cũng phát triển nhiều sơ đồ chọn khác nhau nhằm là tăng tính đa dạng của quần thể, tránh sự hội tụ sớm.

b. *Toán tử lai ghép* là toán tử di truyền cơ bản trong GA, tiến trình lai ghép như sau :

Bước 1: Tạo ra tập NST để tạo sinh từ quần thể bằng cách chọn ngẫu nhiên N NST từ M NST (M là kích cỡ quần thể).

Có nhiều cách chọn:

Chọn ngẫu nhiên theo thứ tự: lặp N lần việc tạo ngẫu nhiên ra một số nguyên i thuộc khoảng [1, M] để chọn NST thứ i.

Chọn theo trọng số: tạo trọng số tích luỹ cho M NST theo công thức:

$$p_i = \frac{i}{\sum_{k=1}^{M} k} \qquad \text{(với bài toán tìm min)}$$

$$p_i = \frac{M - i + 1}{\sum_{k=1}^{M} k} \qquad \text{(với bài toán tìm max)}$$

Sau khi có trọng số tích luỹ cho NST, ta lần lượt tạo các xác suất ngẫu nhiên r và duyệt từ NST đầu tiên đến khi gặp NST có trọng số tích luỹ lớn hơn r thì chọn nó.

Bước 2: Sau khi chọn được N NST, lần lượt lấy ra từng cặp NST để lai ghép tạo ra hai NST mới. Một số dạng toán tử lai ghép hay dùng là :

Lai ghép 1 điểm: chọn ngẫu nhiên một vị trí sau đó hoán vị phần đứng sau vị trí vừa chọn giữa hai NST cha và mẹ để nhận được hai NST con.

Lai ghép hai điểm: chọn ngẫu nhiên hai vị trí trong một NST, sau đó hoán vị các giá trị đứng giữa hai điểm đã chọn của hai NST cha mẹ để nhận được hai NST con.

Lai ghép mặt nạ: tạo một mặt nạ ngẫu nhiên có số bit bằng chiều dài của NST. Ta sẽ hoán vị các giá trị của hai NST cha và mẹ ở những vị trí tương ứng với vị trí bit 1 của mặt nạ.

c. *Toán tử đột biến*: Toán tử đột biến được xây dựng để tránh việc nhận được giá trị tối ưu cục bộ. Đột biến gây ra thay đổi ngẫu nhiên trên từng bit của NST để tạo ra một NST mới.

d. *Tạo sinh:* Chọn các cá thể từ quần thể hiện thời làm quần thể mới cho lần lặp kế tiếp.

1.3. Thuật toán di truyền

Giải thuật di truyền giải một bài toán cần có các thành phần sau:

1. Một cấu trúc dữ liệu biểu diễn không gian lời giải của bài toán
2. Cách khởi tạo quần thể ban đầu
3. Hàm định nghĩa độ thích nghi eval(), đóng vai trò môi trường
4. Các phép toán di truyền (phép lai, phép đột biến, phép tái sinh và phép chọn)
5. Các tham số được giải thuật di truyền sử dụng (kích thước quần thể, xác suất lai, đột biến...)

Sơ đồ cấu trúc giải thuật di truyền tổng quát như sau:

Begin

1. t:=0; Khởi tạo P(t);
2. Tính độ thích nghi cho các cá thể thuộc P(t);
3. Khi (điều kiện dừng chưa thỏa mãn) lặp

   t:=t+1;

   Tái sinh P'(t) từ P(t)

   Lai Q(t) từ P(t-1);

   Đột biến R(t) từ P(t-1);

   Chọn lọc P(t) từ $P(t-1) \cup Q(t) \cup R(t) \cup P'(t)$

Kết thúc lặp

Cá thể tốt nhất P(t) là lời giải cần tìm

End.

1.4. Giải thuật di truyền mã hóa số thực

1.4.1. Mã hóa RCGA

Trong phần này ta quan tâm tới giải thuật di truyền mã hóa số thực (RCGA) để giải các bài toán tối ưu giá trị thực trong không gian R^n và không có các ràng buộc đặc biệt.

Một cách tổng quát, bài toán tối ưu số thực có thể xem là một cặp (S,f), trong đó $S \subseteq R^n$ và $f : S \rightarrow R$ là một hàm n biến. Bài toán đặt ra là tìm véctơ $x=(x_1,x_2,\ldots , x_n) \in S$ sao cho f(x) đạt giá trị cực tiểu trên S. Nghĩa là với mọi $y \in S$ phải có $f(x) \leq f(y)$. Hàm f ở đây có thể không liên tục nhưng cần bị chặn trên S (đối với các bài toán tìm cực đại có thể chuyển về cực tiểu một cách đơn giản).

Trong GA mã hoá số thực, mỗi cá thể được biểu diễn như một véc tơ thực n chiều: $b = (x_1 , x_2, \ldots , x_n)$, $x_i \in R$.

Như vậy một quần thể kích cỡ m là một tập hợp có m véctơ trong R^n. Ta cũng có thể xem một quần thể kích cỡ m như một ma trận thực cấp (mxn), đây là cách mã hoá tự nhiên và thuận tiện trong việc thực hiện các toán tử tiến hóa. Sau đây ta sẽ xem xét cụ thể hơn các toán tử này trong giải thuật di truyền mã hoá số thực.

1.4.2. Các toán tử của RCGA

<u>a. Toán tử lai ghép</u>

GA mã hoá số thực cũng áp dụng các toán tử lai ghép như GA cổ điển bao gồm lai ghép 1 điểm, lai ghép nhiều điểm, lai ghép mặt nạ. Ngoài ra, do cách mã hóa quần thể, người ta còn nghiên cứu và đề xuất nhiều dạng khác nhau của toán tử lai ghép trong RCGA. Dưới đây là một số dạng toán tử lai ghép thường dùng với giả thiết cặp cá thể cha mẹ đã chọn để tiến hành lai ghép là $X = (x_1, x_2, \ldots, x_m)$ và $Y = (y_1, y_2, \ldots, y_m)$.

- *Lai số học (Arithmetic Crossover)*

Phép lai này chọn một số thực a (0<a<1); các con X' và Y' được tính bởi:

$$x'_i = a*x_i + (1-a)*y_i ;$$

$y'_i = a*y_i + (1-a)*x_i$

- *Lai ghép Heuristic*

Giả sử với cặp bố mẹ (X, Y) đã chọn, trong đó cá thể X có độ thích nghi (giá trị hàm mục tiêu) tốt hơn cá thể Y thì toán tử này tạo một con duy nhất X' từ cặp X, Y bởi:

$$x'_i = \alpha*(x_i - y_i) + x_i \qquad \text{với} \quad 0 < \alpha < 1$$

b. Toán tử đột biến

Toán tử đột biến trong RCGA được giới thiệu đa dạng hơn trong GA cổ điển. Sau đây sẽ giới thiệu một số dạng điển hình:

Đột biến đều: Với một gen i được chọn ngẫu nhiên để đột biến từ cá thể $b = (x_1, x_2, ..., x_N)$, thành phần x_i được thay thế bởi một số ngẫu nhiên trong khoảng xác định $[l_i, u_i]$ của x_i.

Đột biến biên: Từ cá thể cha đã chọn đột biến x và vị trí chọn đột biến k, thành phần thứ k (x_k) của x được thay bởi l_k hay u_k trong đó $[l_k, u_k]$ là khoảng xác định của x_k. Trong những bài toán mà biên của các biến không lớn và giải pháp cần tìm nằm gần biên thì phép đột biến này tỏ ra rất hữu ích.

Đột biến không đều:

Giả sử t_{max} là một số cực đại định nghĩa trước, thành phần x_i được thay thế bởi một trong 2 giá trị tính theo các công thức sau :

$$x'_i = x_i + \Delta(t, b_i - x_i) \qquad x''_i = x_i - \Delta(t, x_i - a_i)$$

Việc chọn giá trị nào được tiến hành tuỳ theo giá trị ngẫu nhiên khởi tạo với xác suất 1/2. Biến ngẫu nhiên $\Delta(t, x)$ xác định một bước đột biến trong khoảng [0, x] theo công thức sau :

$$\Delta(t, x) = x.(1-\lambda)^{(1-t/t_{max})^{\tau}}$$

Trong công thức này, λ thường là số ngẫu nhiên phân bố đều trong khoảng đơn vị. Tham số τ xác định ảnh hưởng của lần tạo sinh thứ t trên phân bố của đột biến trong miền [0, x].

1.5. Mô hình Markov của giải thuật di truyền.