Chương 1: Thuật toán tiến hóa EA

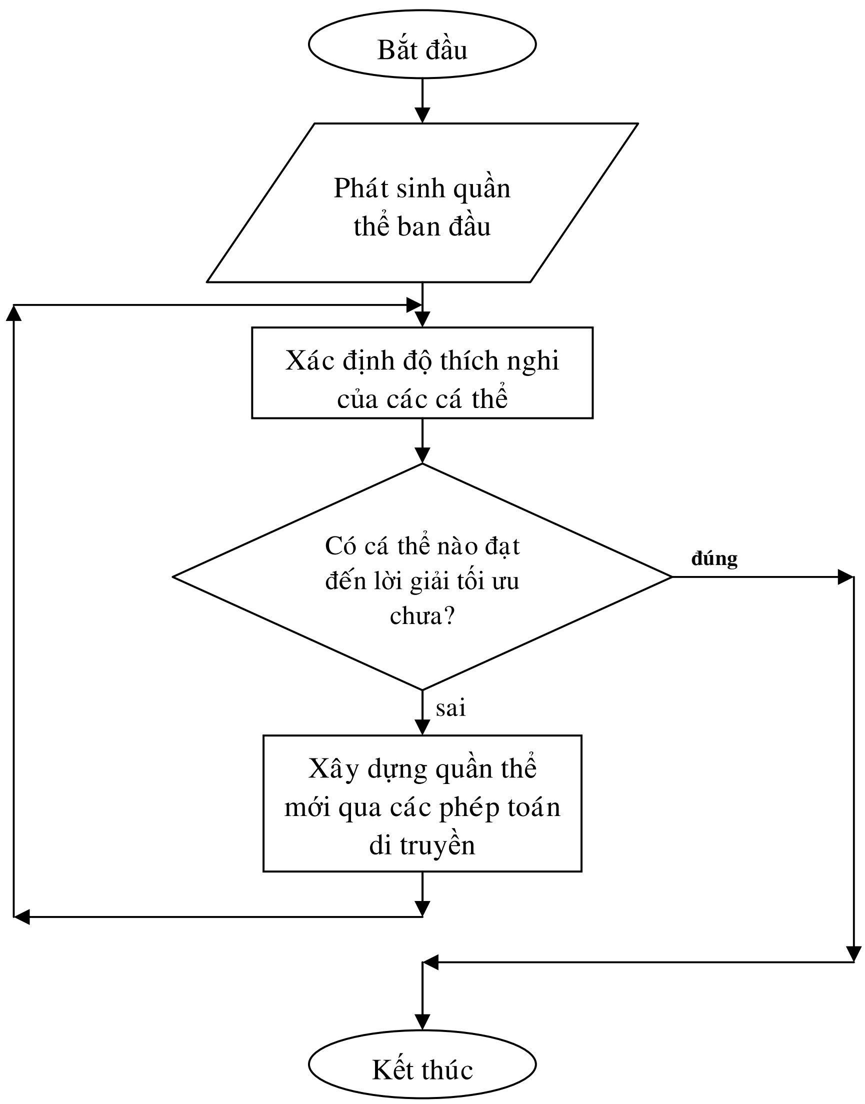
Trong chương này xin giới thiệu tổng quan về hai phần:

- Trước tiên xin sơ lược qua về thuật giải di truyền.
- Tiếp theo nói về Thuật giải tiến hóa EA.

<u>1.1 Thuật toán di truyền GA:</u> [2][4][5][8]

Xin sơ lược về thuật giải di truyền:

- Thuật giải di truyền (GA=Genetic Algorithms), do John Henry Holland đề xướng vào giữa thập niên 70, thế kỷ XX.
- Dùng để tìm kiếm, chọn lựa các giải pháp tối ưu để giải quyết các bài toán khác nhau dựa trên cơ chế chọn lọc tự nhiên của ngành di truyền học.



Hình 1.1 Lưu đồ thuật toán di truyền (GA)

<u>**Giải thích:**</u>

1. <u>*Bước 1:*</u> Phát sinh quần thể ban đầu.

   Dựa vào kích thước (popsize) cho trước, phát sinh ngẫu nhiên ra số cá thể của quần thể ban đầu.

2. <u>*Bước 2:*</u> Xác định độ thích nghi của các cá thể.

   Tùy vào hàm mục tiêu của bài toán cụ thể, ta sẽ có những hàm để đánh giá độ thích nghi của các cá thể.

3. <u>*Bước 3:*</u> Kiểm tra xem có cá thể nào đạt đến lời giải tối ưu (hoặc sau một số lần lặp tạo sinh được xác định) chưa?

   Nếu đạt yêu cầu thì dừng.

   Ngược lại qua bước 4.

4. <u>*Bước 4:*</u> Xây dựng lại quần thể mới dựa vào các phép toán di truyền.
   - Lai ghép.
   - Đột biến.
   - Chọn lọc.

5. <u>*Bước 5:*</u> lặp lại bước 2.

<u>Nhận xét:</u>

- <u>Ưu điểm:</u>
  - Ứng dụng mạnh mẽ trong việc tìm kiếm các giải pháp tối ưu.
  - Ứng dụng rộng rãi trong nhiều lĩnh vực.
  - GA làm việc trên một tập các điểm, chứ không phải duy nhất một điểm.
  - GA chỉ sử dụng hàm mục tiêu chứ không sử dụng bất kỳ một tri thức phụ nào.
  - GA sử dụng các công thức tính xác suất, mà không dùng các công thức tính toán tuyệt đối.

- Hạn chế:
  - Nếu chọn mô hình biểu diễn không thích nghi, GA sẽ hội tụ sớm, dẫn tới bài toán chỉ tối ưu cục bộ.
  - Nếu số lần lặp tạo sinh (gen) ít, có thể không dẫn đến kết quả tối ưu.
  - Do GA chỉ sử dụng hàm mục tiêu, không sử dụng tri thức khác nên khi gặp một bài toán phức tạp, sẽ dẫn đến mô hình bài toán không thích nghi.

1.2 Thuật toán tiến hóa EA: [4][8][9]

Ta đã biết thuật giải di truyền cổ điển ban đầu với những ứng dụng rộng rãi của nó. Tuy nhiên, khi ứng dụng nó vào thực tế, đôi khi gặp những bài toán đòi hỏi một cách biểu diễn lời giải thích hợp. Nếu không, thuật giải di truyền khó có thể cho lời giải tốt. Từ đó các biến thể của nó xuất hiện, dựa trên việc là tận dụng các tri thức của bài toán, thông qua ba hướng:

(1) Xây dựng một cấu trúc dữ liệu hợp lý, sao cho việc xây dựng các phép toán di truyền được tự nhiên và hiệu quả nhất.

(2) Sử dụng phương pháp đã và đang được sử dụng để giải bài toán này và kết hợp chúng với thuật giải di truyền.

(3) Tận dụng cả hai cách trên.

Trong phạm vi luận văn trình bày này: chương 2 xin được chọn hướng thứ nhất "Xây dựng một cấu trúc dữ liệu hợp lý, sao cho việc xây dựng các phép toán di truyền được tự nhiên và hiệu quả nhất", chương 4 xin chọn hướng thứ ba: dựa vào tri thức cụ thể của bài toán để xây dựng một mô hình biểu diễn, với các phép toán di truyền sao cho tự nhiên và hiệu quả nhất. Đồng thời kết hợp với các phương pháp đã sử dụng trước đó.

Thuật toán EA chính là sự phát triển từ thuật toán di truyền cổ điển với các biến thể của nó.

<u>* Thuật toán tiến hóa EA:</u>

Bắt đầu

t=0

Khởi tạo P(t), Lượng giá P(t)

Lặp cho đến khi thỏa điều kiện dừng

t=t+1

Lai ghép 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)

Hết lặp

Kết thúc

Trong đó, ***Chọn lọc*** gồm hai bước:

Sắp xếp quần thể theo thứ tự độ thích nghi giảm dần.

Loại bỏ các cá thể cuối dãy chỉ giữ lại n cá thể tốt nhất. (Trường hợp quần thể có kích thước cố định – trong chương trình thể nghiệm, chọn theo trường hợp này).

- Thuật toán EA gồm năm thành phần chính:
    1. Cách biểu diễn di truyền cho lời giải của bài toán.
    2. Cách khởi tạo quần thể ban đầu.
    3. Một hàm lượng giá đóng vai trò môi trường, đánh giá các lời giải theo mức độ "thích nghi" của chúng.
    4. Các phép toán di truyền.
    5. Các tham số khác: kích thước quần thể, xác suất áp dụng các phép toán di truyền, số lần lặp,...

Chương 2: Tối ưu hàm nhiều biến rời rạc dạng tổng quát

Tối ưu hóa là một trong những vấn đề - bài toán kinh điển trong tất cả mọi lĩnh vực của cuộc sống từ nhu cầu đơn giản của từng cá nhân đến các nhu cầu phức tạp của các tổ chức thương mại, chính trị, xã hội.

Một cách hình thức ta có thể định nghĩa bài toán tối ưu hóa như sau:

Định nghĩa 1.1 (giải bài toán tối ưu tổng quát):

> Cho I là không gian các giải pháp của một bài toán P cho trước.
>
> Tìm giải pháp $a^* \in I$ sao cho $\forall a \in I$, $\psi(a) \leq \psi(a^*)$
>
> Trong đó, $\psi: I \rightarrow R$ là một hàm gán một giá trị thực cho một phần tử thuộc tập I. Giá trị này là một cách số hóa một số tiêu chuẩn nào đó mà bài toán P đòi hỏi.

Như vậy, mục tiêu cuối cùng của bài toán tối ưu là tìm kiếm lời giải tối ưu. Tìm kiếm đó cần cân đối hai mục tiêu (có vẻ mâu thuẫn nhau): khai thác những lời giải tốt nhất và khảo sát không gian tìm kiếm.

Tuy nhiên, các bài toán tối ưu trên thực tế lại hiếm khi đòi hỏi một điểm tối ưu tuyệt đối (chặt chẽ theo định nghĩa trong toán học) mà chỉ đòi hỏi một điểm tối ưu đủ tốt theo một tiêu chuẩn nào đó.

Cho đến nay, đã có khá nhiều phương pháp giải quyết bài toán tối ưu (thông qua các bài toán tối ưu hàm số), nhưng nhìn chung các phương pháp chỉ dừng lại ở những lớp bài toán với thông tin rõ ràng hoặc với các thông tin bổ trợ khác.

Hơn nữa, việc tìm ra điểm tối ưu tuyệt đối nhiều lúc không thể thực hiện được do bài toán đặt ra quá phức tạp. Chẳng hạn trong sản xuất kinh doanh, người ta thường tìm cách tối thiểu chi phí sản xuất. Và dĩ nhiên, họ chỉ cần một giải pháp mà theo đó, chi phí giảm đến một mức độ nào đó là đủ chứ không nhất thiết phải thực sự thấp nhất. Đây chính là điều kiện rất thuận lợi để áp dụng thuật giải di truyền để giải bài toán tối ưu hàm nhiều biến tổng quát.

Trong chương này xin trình bày về:

- Bài toán tối ưu tổng quát và phân loại chúng.
- Tìm hiểu bài toán nhiều biến rời rạc tổng quát.
- Với ví dụ minh họa: bài toán vận tải với hai phương pháp giải quyết:
    - Véc tơ
    - Ma trận.

Dựa trên việc "Xây dựng một cấu trúc dữ liệu hợp lý, sao cho việc xây dựng các phép toán di truyền được tự nhiên và hiệu quả nhất".

2.1 Đặt vấn đề:

2.1.1 Bài toán tối ưu tổng quát:

Bài toán tối ưu tổng quát được phát biểu như sau:

Cực đại hóa (hay cực tiểu hóa) hàm:

$$f(x) \rightarrow \max (\min) \qquad (1.1)$$

Với các điều kiện:

$$g_i(x)\ (\leq,=,\geq)\ b_i,\ i = \overline{1,m} \qquad (1.2)$$

$$x \in X \subset R^n \qquad (1.3)$$

Bài toán (1.1), (1.3) được gọi là một qui hoạch, hàm f(x) được gọi là hàm mục tiêu, các hàm $g_i(x)(\leq,=,\geq)\ b_i,\ i = \overline{1,m}$ được gọi là các hàm ràng buộc, mỗi đẳng thức hoặc bất đẳng thức trong hệ (1.2) được gọi là một ràng buộc.

Tập hợp:

$$D = \{\ x \in X \mid g_i(x)(\leq,=,\geq)\ b_i,\ i = \overline{1,m}\ \} \qquad (1.4)$$

D được gọi là miền giá trị. Mỗi điểm $x = \{x_1,x_2,..,x_n\} \in D$ được gọi là một phương án (hay một lời giải chấp nhận được). Một phương án $x^* \in D$ phải đạt giá trị cực đại hay giá trị cực tiểu.

Cụ thể là: $x^* \in D$ đạt được các giá trị sau:

- $f(x^*) \geq f(x), \forall x \in D$ đối với bài toán tìm max.
- $f(x^*) \leq f(x), \forall x \in D$ đối với bài toán tìm min.

Đây được gọi là phương pháp tối ưu (hay còn được gọi là lời giải tối ưu).

Khi đó, giá trị $f(x^*)$ được gọi là giá trị tối ưu của bài toán.

2.1.2 Phân loại các bài toán tối ưu:

Một trong những phương pháp hiển nhiên nhất để giải các bài toán đặt ra là phương pháp “vét cạn”.

Nội dung của phương pháp "vét cạn":

> Tính giá trị hàm mục tiêu f(x) trên tất cả các phương án, sau đó so sánh các giá trị tính được để tìm ra phương án tối ưu và giá trị tối ưu của bài toán.

Tuy nhiên cách giải quyết này khó có thể thực hiện được, ngay cả khi kích thước của bài toán (số biến n và số ràng buộc m) là không lớn. Trong thực tế, tập hợp D thường có một số lượng rất lớn các phần tử, trong nhiều trường hợp, số lượng các phần tử của tập D là vô hạn.

Vì vậy, phương pháp "vét cạn" là không khả thi. Chúng ta cần phải có những nghiên cứu trước về mặt lý thuyết để có thể tách ra từ bài toán tổng quát, những lớp bài toán "dễ giải hơn".

Các nghiên cứu lý thuyết đó thường là:

- Nghiên cứu các tính chất và các thành phần bài toán (hàm mục tiêu, các hàm ràng buộc, các biến số, các hệ số,..).
- Các điều kiện tồn tại lời giải chấp nhận được.
- Các điều kiện cần và đủ của cực trị.
- Tính chất của các đối tượng nghiên cứu.

Các tính chất của các thành phần của bài toán và đối tượng nghiên cứu giúp ích cho ta rất nhiều trong công việc phân loại các bài toán.

Một bài toán tối ưu (qui hoạch toán học) có thể được phân ra thành các loại như sau:

- Loại 1: Qui hoạch tuyến tính (QHTT) nếu hàm mục tiêu f(x) và tất cả các hàm ràng buộc $g_i(x)(\leq,=,\geq)$ b_i, $i=\overline{1,m}$ là tuyến tính. Một trường hợp riêng quan trọng của qui hoạch tuyến tính là bài toán vận tải.
- Loại 2: Qui hoạch tham số (QHTS) nếu các hệ số trong biểu thức của hàm mục tiêu và các ràng buộc phụ thuộc vào tham số.

- Loại 3: Qui hoạch động (QHĐ) nếu đối tượng xét là các quá trình có nhiều giai đoạn nói chung, hay các quá trình phát triển theo thời gian nói riêng.
- Loại 4: Qui hoạch phi tuyến (QHPT) nếu f(x) hoặc có ít nhất một trong các hàm $g_i(x)$ là phi tuyến hoặc cả hai trường hợp đó cùng xảy ra.
- Loại 5: Qui hoạch rời rạc (QHRR) nếu miền ràng buộc D là tập rời rạc. Trong trường hợp riêng, khi các biến chỉ nhận giá trị nguyên, ta có qui hoạch nguyên (QHN). Một trường hợp riêng của QHN là qui hoạch biến bool khi các biến số chỉ nhận giá trị 0 hoặc 1.
- Loại 6: Qui hoạch đa mục tiêu (QHĐMT) nếu trên cùng một miền ràng buộc ta xét các hàm mục tiêu khác nhau.

Trong các loại bài toán trên, ta quan tâm tìm hiểu loại 5 (bài toán qui hoạch nguyên) và loại 6 (qui hoạch đa mục tiêu). Từ đó đưa ra một lớp bài toán đa mục tiêu, nhiều ràng buộc tổng quát.

2.2 Bài toán nhiều biến rời rạc nhiều ràng buộc tổng quát: [3][4][7][9]

Trong chương trước ta nghiên cứu về tổng quan Thuật giải di truyền (Genetic Algorithms - GA) và Thuật giải tiến hóa EA. Trong phần này, ta nghiên cứu về tối ưu hàm nhiều biến rời rạc, với phát biểu bài toán như sau:

Tìm $X = (x_0,x_1,x_2,..,x_{n-1}) \in Zn$, sao cho:

f(X) đạt giá trị tối ưu (cực đại hay cực tiểu).

thỏa:

$min_1 \leq g_1(X) \leq max_1$

$min_2 \leq g_2(X) \leq max_2$

.....

$min_n \leq g_n(X) \leq max_n$

2.3 Cách giải quyết:

Không gian tìm kiếm sẽ là tập các hoán vị của n biến. Bất cứ một hoán vị nào của n biến này cũng là một lời giải chấp nhận được. Do đó kích thước của không gian tìm kiếm là n! (rất lớn). Ta áp dụng thuật giải di truyền để giải quyết lớp bài toán này.

2.3.1 Biểu diễn nhiễm sắc thể:

Thông thường có nhiều cách biểu diễn một nhiễm sắc thể:

- Biểu diễn bằng chuỗi nhị phân 0,1: Mỗi gen của nhiễm sắc thể được mã hóa nhờ số lượng bit (0,1) nào đó. Cách biểu diễn này có nhược điểm là độ chính xác không cao (các phần tử được truy cập là các số nguyên), muốn tăng độ chính xác phải tăng số lượng bit biểu diễn do đó dẫn đến làm chậm thuật toán, tính chính xác bị mất khi tăng kích cỡ miền vì chiều dài nhị phân cho trước là cố định.
- Biểu diễn bằng số thập phân: Mỗi nhiễm sắc thể được mã hóa là một véc tơ số dấu phẩy động với cùng chiều dài của véc tơ lời giải. Cách biểu diễn này khắc phục được các nhược điểm của biểu diễn nhị phân, độ chính xác tùy thuộc vào khả năng của máy (số chữ số thập phân sau dấu phẩy), có khả năng biểu diễn được các miền rộng lớn.
- Biểu diễn bằng chữ cái
- Biểu diễn bằng kết hợp chữ và số.
- ...

Trong bài toán này, ta dùng cách biểu diễn véc tơ số nguyên

chrom = $(x_0,x_1,x_2,..,x_{n-1}) \in Zn$

Tùy theo nội dung, yêu cầu và những tri thức bài toán mà ta chọn t_size = sizeof(x_i) có giá trị khác nhau (1 byte, 2 byte,.., n byte) để thích hợp với bài toán.

2.3.2 <u>Hàm thích nghi:</u>

- Miền khả thi:

  $\omega = \{ X \in Zn : min_i \leq g_i(X) \leq max_i, \forall i = 0,1,..,n-1 \}$

- Hàm đánh giá:

  * Trường hợp bài toán tối ưu là cực tiểu hàm đánh giá f(X) thì hàm thích nghi như sau:

$$fitness(X) = \begin{cases} C_{\max} - f(X), & f(X) < C_{\max} \quad and \quad X \in \omega \\ 0 & f(X) >= C_{\max} \quad or \quad X \notin \omega \end{cases}$$

  Trong đó C_{max} là một tham số đầu vào. Có thể lấy C_{max} là giá trị f lớn nhất trong quần thể hiện tại, hoặc lớn nhất sau k vòng lặp.

  * Trường hợp bài toán tối ưu là cực đại hàm đánh giá f(X) thì hàm thích nghi như sau:

$$fitness(X) = \begin{cases} f(X) - C_{\min}, & f(X) > C_{\min} \quad and \quad X \in \omega \\ 0 & f(X) <= C_{\min} \quad or \quad X \notin \omega \end{cases}$$

  Trong đó C_{min} là một tham số đầu vào. Có thể lấy C_{min} là giá trị f nhỏ nhất trong quần thể hiện tại, hoặc nhỏ nhất sau k vòng lặp.

2.3.3 <u>Toán tử chọn cá thể:</u>

- Tính tổng thích nghi (total fitness) của tất cả các thành viên trong quần thể:

  total fitness $= \sum_{i=1}^{N} fitness(X_i)$

  trong đó, N là số nhiễm sắc thể trong quần thể,

  X_i là nhiễm sắc thể thứ i trong quần thể.

- Tính vị trí chọn từng nhiễm sắc thể:

  Pos_i = fitness(X_i) / total fitness

  Trong đó, pos_i là vị trí chọn nhiễm sắc thể thứ i, i = 0,..,N-1

- Phát sinh một số q ngẫu nhiên trong khoảng (0,1):

  + Nếu $q < pos_0$ thì nhiễm sắc thể thứ 0 được chọn.

  + Nếu $pos_{i-1} < q < pos_i$, i = 1,2,..,N-1 thì nhiễm sắc thể thứ i được chọn.

2.3.4 <u>Toán tử lai ghép:</u>

- Giả sử có 2 nhiễm sắc thể sau:

  chrom1 = $(x_0,x_1,x_2,..,x_{n-1})$

  chrom2 = $(y_0,y_1,y_2,..,y_{n-1})$

- Xác định vị trí lai ghép (i) bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].
- Hoán đổi các thành phần của chrom1 và chrom2 từ vị trí lai ghép thứ i cho đến cuối.
- Khi đó từ 2 nhiễm sắc thể ban đầu:

  chrom1 = $(x_0,x_1,x_2,..,x_i,..,x_{n-1})$

  chrom2 = $(y_0,y_1,y_2,..,y_i,..y_{n-1})$

  ta được 2 nhiễm sắc thể mới:

  chrom1 = $(x_0,x_1,x_2,..,y_i,..,y_{n-1})$

  chrom2 = $(y_0,y_1,y_2,..,x_i,..,x_{n-1})$

2.3.5 <u>Toán tử đột biến:</u>

- Giả sử có nhiễm sắc thể sau:

  chrom = $(x_0,x_1,x_2,..,x_{n-1})$

- Xác định vị trí đột biến (i) bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].
- Thay phần tử x_i bằng một số nguyên $y \in Z$.
- Khi đó từ nhiễm sắc thể ban đầu:

  chrom = $(x_0,x_1,x_2,..,x_i,..,x_{n-1})$

Ta được nhiễm sắc thể mới:

chrom = $(x_0,x_1,x_2,..,y,..,x_{n-1})$

2.3.6 Thuật giải:

Bước 1: Tạo ngẫu nhiên quần thể ($P_{t=0}$), với t=0.

Bước 2: Tính giá trị thích nghi (fitness) của từng nhiễm sắc thể.

Bước 3: Lặp cho đến khi thỏa điều kiện dừng:

- Chọn k cặp nhiễm sắc thể từ P_t.
- Lai ghép các nhiễm sắc thể trong S.
- Đột biến các nhiễm sắc thể trong S.
- Chọn lọc các nhiễm sắc thể có giá trị thích nghi cao nhất.
- Thay P_t bằng P'_t.
- t = t+1.

Hết lặp.

2.4 Ví dụ bài toán minh họa: [1][3][4]

2.4.1 Đặt vấn đề:

Ta có nhiều kho hàng chứa vật liệu xây dựng cung cấp cho nhiều công trình đang thi công xây dựng. Bài toán được đặt ra là: Ta phải hoạch định lộ trình từ các nguồn cung cấp (kho hàng) đến các đích (công trình xây dựng) như thế nào, để chi phí vận chuyển là nhỏ nhất?

Vấn đề cạnh tranh ngày nay là làm thế nào để hạ được giá thành sản phẩm của đơn vị mình. Một trong vấn đề then chốt (theo các chuyên gia về kinh tế đánh giá) để hạ giá thành sản phẩm là: "giảm chi phí vận chuyển". Nhất là tình hình hiện nay, giá xăng dầu đang là vấn đề nóng bỏng trên thế giới. Vì thế bài toán giảm chi phí vận chuyển là một bài toán đáng được quan tâm.

Giả sử ta xem vật liệu xây dựng là một mặt hàng duy nhất.

Bài toán vận tải được phát biểu như sau:

Ký hiệu:

n: số nguồn cung cấp (kho hàng).

k: số đích yêu cầu (công trình xây dựng).

cost(ij): chi phí vận chuyển 1 đơn vị hàng hóa từ nguồn i đến đích j

x_{ij}: số lượng đơn vị hàng hóa được vận chuyển từ nguồn i đến đích j

dest(j): số lượng của nhu cầu tại đích j, với j = 1, 2, .., k

sour(i): số lượng cung cấp tại nguồn i, với i = 1, 2, .., n

Bài toán vận tải được định nghĩa như sau:

$$Min\sum_{i=1}^{n}\sum_{j=1}^{k} f_{ij}(X_{ij}) \text{ (P)}$$

Trong đó: $f_{ij}(x_{ij}) = cost(ij)*x_{ij}$

với các ràng buộc:

(1) $\sum_{j=1}^{k} X_{ij} <= sour(i), \qquad i = 1,2,..,n$

(2) $\sum_{i=1}^{n} X_{ij} >= dest(j), \qquad j = 1,2,..k$

(3) $\sum_{i=1}^{n} sour(i) >= \sum_{j=1}^{k} dest(j)$

Với: $x_{ij} >= 0$; i=1,2,..,n và j=1,2,..,k

Giải thích:

Ràng buộc thứ nhất: Tổng lượng hàng chuyên chở từ một nguồn không thể vượt quá khả năng cung cấp của nó.

Ràng buộc thứ hai: Tổng lượng hàng chuyên chở đến một đích phải hợp với nhu cầu của nó.

Bài toán trên đòi hỏi tổng các nguồn cung cấp ít nhất phải bằng tổng nhu cầu đích.

Nếu tổng của chúng bằng nhau thì gọi là bài toán vận tải tuyến tính cân bằng.

Bài toán này ta xét với điều kiện tuyến tính cân bằng.

Bài toán này ta xem tất cả sour(i) và dest(j) là các số nguyên: x_{ij} là các số nguyên.

2.4.2 *Ví dụ cụ thể:*

Giả sử xét bài toán với 3 nguồn và 4 đích như sau:

Các nguồn: sour(1)=15, sour(2)=25 và sour(3)=5

Các đích: dest(1)=5, dest(2)=15, dest(3)=15 và dest(4)=10

Bảng 2.1 Bảng giá cost của bài toán vận tải ví dụ minh họa

	1	2	3	4
1	10	0	20	11
2	12	7	9	20
3	0	14	16	18

Dòng i
Cột j

2.5 Các bước giải quyết bài toán theo véc tơ:

Nội dung chủ yếu của phần này là áp dụng thuật toán tiến hóa EA (theo hướng tiếp cận thứ nhất: "Xây dựng một cấu trúc dữ liệu hợp lý, sao cho việc xây dựng các phép toán di truyền được tự nhiên và hiệu quả nhất") để giải bài toán vận tải.

2.5.1 Cách biểu diễn di truyền cho lời giải của bài toán.

Biểu diễn lời giải: là một véc tơ có giá trị nguyên dương, gồm q phần tử có giá trị phân biệt (với q=k*n).

$(x_1,x_2,x_3,x_4,..,x_{q-1},x_q)$ với $x_i \in [1,k*n]$, i=1, 2, .., q

Ví dụ: với q=12, ta có véc tơ biểu diễn sau:

(10,8,5,3,1,11,4,12,7,6,9,2)

2.5.2 Cách khởi tạo quần thể ban đầu.

Thông qua thủ tục khởi tạo: với các ràng buộc. Xin trình bày ở phần sau.

2.5.3 Một hàm lượng giá đóng vai trò môi trường, đánh giá các lời giải theo mức độ "thích nghi" của chúng.

Hàm đánh giá:

$$\sum_{j=1}^{k}\sum_{i=1}^{n} V_{ij} * \cos t[i][j]$$

2.5.4 Các phép toán di truyền.

Toán tử di truyền:

Đột biến nghịch đảo:

$(x_1,x_2,..,x_q) \Rightarrow (x_q,x_{q-1},..,x_1)$ với q=k*n

Đột biến:

$(x_1,x_2,x_i,..,x_j,x_{q-1},x_q) \Rightarrow (x_1,x_2,x_j,..,x_i,x_{q-1},x_q)$ vị trí i,j

Toán tử lai ghép:

B1: Tạo ra một bản sao của cá thể cha thứ hai.

B2: Chọn một phần tùy ý của cá thể cha thứ nhất.

B3: Tạo ra một thay đổi tối thiểu trong cá thể con cháu cần thiết để đạt được một mẫu được chọn.

Ví dụ:

Cá thể cha thứ nhất: (1,2,3,4,5,6,7,8,9,10,11,12)

Cá thể cha thứ hai: (7,3,1,11,4,12,5,2,10,9,6,8)

Phần được chọn là: (4,5,6,7)

Cá thể con cháu sẽ là: (3,1,11,4,5,6,7,12,2,10,9,8)

2.5.5 Các tham số khác: kích thước quần thể, xác suất áp dụng các phép toán di truyền, số lần lặp,...

Kích thước quần thể: popsize=30

Số lần lặp: gen=50

2.5.6 Thủ tục khởi tạo một cá thể:

Vào: array dest[k], sour[n];

Ra: array (v_{ij}), v_{ij} >= 0 với mọi i, j;

$$\sum_{j=1}^{k} V_{ij} = dest[i], \qquad i = 1,2,..,n$$

$$\sum_{i=1}^{n} V_{ij} = sour[j], \qquad j = 1,2,..k$$

Procedure Khoi_Tao;

Begin

Thiết lập dãy số chưa duyệt từ 1 đến k*n;

Lặp

Chọn ngẫu nhiên 1 số chưa duyệt q ∈ [1,k*n] và duyệt nó:

Hàng i = [(q-1)/k+1];

Cột j = [(q-1) mod k +1];

Val = min(sour[i],dest[j]); Vij=Val;

Sour[i]=sour[i]-Val;dest[j]=dest[j]-Val;

Cho đến khi tất cả các số được duyệt;

End;

Với dãy số nguyên (10,8,5,3,1,11,4,12,7,6,9,2)

Đầu tiên ta khởi tạo một ma trận phân phối 4 X 5 như sau:

	5	15	15	10
15				
25				
5				

Với x_1=10, ta có:

Hàng i=3, cột j=2

Val=min(sour[3],dest[2])=5

V[3][2]=5

	5	10	15	10
15				
25				
0		5		

Tương tự như vậy với x_2, x_3,.., x_{12} ta có ma trận phân phối V_{ij} như sau:

	0	0	0	0
0	0	0	15	0
0	5	10	0	10
0	0	5	0	0

2.5.7 Thuật giải EA cho bài toán vận tải:

Bắt đầu

t=0

Khởi tạo P(t)

Lượng giá P(t)

Lặp cho đến khi thỏa điều kiện dừng

t=t+1

Lai ghép 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)

Hết lặp

Kết thúc

Trong đó, ***Chọn lọc*** gồm hai bước:

Sắp xếp quần thể theo thứ tự độ thích nghi giảm dần

Loại bỏ các cá thể cuối dãy chỉ giữ lại n cá thể tốt nhất. (Trường hợp quần thể có kích thước cố định)

2.5.8 Kết quả thử nghiệm

Lời giải với các x_{ij} nguyên, với chi phí min là: 315

Bảng 2.2 Bảng kết quả của bài toán vận tải ví dụ minh họa

Sour(i): hàng

Dest(j): cột

	5	15	15	10
15	0	5	0	10
25	0	10	15	0
5	5	0	0	0

2.6 Các bước giải quyết bài toán theo ma trận:

2.6.1 Cách biểu diễn: di truyền cho lời giải của bài toán theo ma trận.

Là cấu trúc ma trận $V=(v_{ij})$, với i:1,2,..,k và j: 1,2,..,n

Với ràng buộc:

- $v_{ij} \geq 0 \ \forall_i=1,2,..,k$ và $\forall_j=1,2,..,n$
- $\sum_{i=1}^{k} v_{ij}$ =dest[j] với m=1,..,n
- $\sum_{j=1}^{n} v_{ij}$ =source[i] với i=1,..k

Trong đó:

i: dòng, j: cột,

k: số đích nhu cầu cần cung cấp,

n: số nguồn cung cấp,

m: cột đang xét.

Khởi tạo quần thể ban đầu: bằng cách sử dụng lại thủ tục khởi tạo ở mục 2.5.6 nêu trên.

2.6.2 Hàm lượng giá:

$$Eval(v_{ij})=\sum_{i=1}^{k}\sum_{j=1}^{n} v_{ij} .cost[j][m]$$

2.6.3 Các phép toán di truyền:

2.6.3.1 Đột biến:

Giả sử:

$\{i_1,i_2,..,i_p\}$ là 1 tập con của $\{1,2,..,k\}$

và $\{j_1,j_2,..,j_q\}$ là tập con của $\{1,2,..,n\}$

sao cho $2 \leq p \leq k$ và $2 \leq q \leq n$.

Ta diễn tả sự đột biến là ma trận $V=(v_{ij})$ (k x n) sinh ra 1 ma trận (p x q) $W=(w_{ij})$ từ V bằng cách sau:

- Một thành phần $v_{ij} \in V$ là thành phần của W $\Leftrightarrow$ i $\in \{i_1,i_2,..,i_p\}$ và j $\in \{j_1,j_2,..,j_q\}$ (nếu $i=i_r$ và $j=j_s$ thì thành phần v_{ij} được đặt tại dòng r và cột s của ma trận W).
- Bây giờ ta thiết lập $sour_w[i]$ và $dest_w[j]$ $(1\leq i\leq p, 1 \leq j \leq q)$ đối với ma trận W:

  $Source_w[i]=\sum v_{ij}$, với j $\in \{j_1,j_2,..,j_q\}$ và $1 \leq i \leq p$

  $Destw[j] = \sum v_{ij}$, với i $\in \{i_1,i_2,..,i_p\}$ và $1 \leq j \leq q$

Ta khởi tạo giá trị đối với ma trận W sao cho tất cả các ràng buộc thỏa mãn.

Ví dụ: Cho bài toán vận tải 4 nguồn cung cấp (sources) và 5 đích nhu cầu (destinations) sau:

sour[1] = 8, sour[2] = 4, sour[3] = 12, sour[4] = 6

dest[1] = 3, dest[2] = 5, dest[3] = 10, dest[4] = 7, dest[5] = 5

Giả sử cha được chọn là:

	3	5	10	7	5
8	0	0	5	0	3
4	0	4	0	0	0
12	0	0	5	7	0
6	3	1	0	0	2

Chọn ngẫu nhiên 2 dòng {2,4} và 3 cột {2,3,5} ta được ma trận con W:

4	0	0
1	0	2

Có sour[1]=4, sour[2]=3, dest[1]=5, dest[2]=0, dest[3]=2, sau khi khởi tạo lại ta được ma trận W có giá trị sau:

2	0	2
3	0	0

Vì vậy ma trận con của ma trận V là:

	3	5	10	7	5
8	0	0	5	0	3
4	0	2	0	0	2
12	0	0	5	7	0
6	3	3	0	0	0

2.6.3.2 Phép lai:

Lấy cặp ma trận V1 và V2, V1 = (v^1_{ij}) và V2 = (v^2_{ij}) tạo ra 2 ma trận trung gian là DIV = (div_{ij}) và REM = (rem_{ij}) được định nghĩa sau:

$div_{ij} = [(v^1_{ij}+ v^2_{ij}) \text{ div } 2]$

$rem_{ij} = [(v^1_{ij}+ v^2_{ij}) \text{ mod } 2]$

Ma trận REM có tính chất đặc biệt: số lượng bit 1 ở mỗi dòng và mỗi cột đều là số chẵn. Chúng ta dùng tính chất này để chuyển ma trận REM thành 2 ma trận REM_1 và REM_2 như sau:

$REM = REM_1 + REM_2$

$sour_{REM1}[i] = sour_{REM2}[i] = sour_{REM}[i]/2$ với i=1,..,k

$dest_{REM1}[j] = dest_{REM2}[j] = dest_{REM}[j]/2$ với j=1,..,n

và 2 con được sản xuất là:

$V_3 = DIV + REM_1$

$V_4 = DIV + REM_2$

Ví dụ: theo ví dụ đã nêu ở trên. Lấy 2 cha là:

V_1

1	0	0	7	0
0	4	0	0	0
2	1	4	0	5
0	0	6	0	0

V_2

0	0	5	0	3
0	4	0	0	0
0	0	5	7	0
3	1	0	0	2

Ma trận DIV và REM là:

DIV

0	0	2	3	1
0	4	0	0	0
1	0	4	3	2
1	0	3	0	1

REM

1	0	1	1	1
0	0	0	0	0
0	1	1	1	1
1	1	0	0	0

Ma trận REM được tách thành 2 ma trận REM_1 và REM_2, như sau:

REM_1

0	0	1	0	1
0	0	0	0	0
0	1	0	1	0
1	0	0	0	0

REM_2

1	0	0	1	0
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0

Cuối cùng 2 ma trận con là:

V_3

0	0	3	3	2
0	4	0	0	0
1	1	4	4	2
2	0	3	0	1

V_4

1	0	2	4	1
0	4	0	0	0
1	0	5	3	3
1	1	3	0	1

Chương 3: Tìm hiểu bài toán đa mục tiêu với nhiều ràng buộc

[4][7][9][10][11][12][13][14][15]

Trong chương này, xin trình bày về:

- Tìm hiểu về đa mục tiêu.
- Các định nghĩa và một số phương pháp giải quyết.
- Một số khó khăn của tối ưu đa mục tiêu.

3.1 Giới thiệu: Multiobjective evolution algorithms (MEAs)

Đối với nhiều bài toán khi ra quyết định trong thế giới thực, có một nhu cầu là tối ưu hóa đồng thời nhiều mục tiêu.

Ví dụ: Việc thiết kế một chiếc ô tô, có thể có nhiều mục tiêu cần đạt là: giá thành nhỏ nhất, tốc độ nhanh nhất, tiêu thụ nhiên liệu ít nhất và sang trọng nhất. Những mục tiêu này rõ ràng mâu thuẫn (conflicting) nhau và các kỹ thuật đơn mục tiêu ở phần trước không thể giải quyết được những bài toán này.

MEAs có thể sản sinh ra một tập giải pháp tiềm năng về một vài khả năng phán đoán tốt nhất.

Những bài toán tối ưu đa mục tiêu này cần những kỹ thuật riêng biệt, mà những kỹ thuật này rất khác với các kỹ thuật tối ưu hóa chuẩn, đối với tối ưu hóa một mục tiêu duy nhất. Rõ ràng là nếu có hai mục tiêu cần tối ưu hóa, ta có thể tìm một lời giải tốt nhất tương xứng với mục tiêu thứ nhất, còn lời giải kia là tốt nhất đối với mục tiêu thứ hai.

3.2 Một số định nghĩa:

Định nghĩa 1: Để không làm mất tính tổng quát, ta có vấn đề tối ưu hóa đa mục tiêu với m biến x quyết định và n mục tiêu y như sau:

Tìm Max của $y=f(x)=(f_1(x_1,..,x_m),..,f_n(x_1,..,x_m))$

Trong đó: $x=(x_1,..,x_m) \in X$ (2.1)

$y=(y_1,..,y_n) \in Y$

và:

x được gọi là tham số véc tơ quyết định, X là không gian tham số.

y là véc tơ mục tiêu, Y là không gian mục tiêu.

Một véc tơ quyết định a $\in$ X được nói là thống trị một véc tơ quyết định b $\in$ X (cũng có thể viết là a $\succ$ b) nếu và chỉ nếu:

$$\begin{aligned} & \forall i \in \{1,..,n\} & : \quad & f_i(a) \geq f_i(b) \\ \wedge \quad & \exists j \in \{1,..,n\} & : \quad & f_j(a) > f_j(b) \end{aligned} \qquad (2.2)$$

Thêm nữa, chúng ta nói rằng a bao phủ (covers) b (a $\succeq$ b) nếu và chỉ nếu a $\succ$ b hoặc f(a)=f(b).

Dựa vào quy ước này, chúng ta có thể định nghĩa những giải pháp không bị thống trị (nondominated), tối ưu Pareto (Pareto-optimal) như sau:

Định nghĩa 2: Cho a $\in$ X là một véc tơ quyết định tùy ý.

(a) Véc tơ quyết định a được nói là không bị thống trị (nondominated) đối với một tập X' $\subseteq$ X nếu và chỉ nếu không có véc tơ nào trong X' thống trị a, ta có:

$$\nexists a' \in X' : a' \succ a \qquad (2.3)$$

(b) Véc tơ quyết định a được gọi là Pareto-optimal nếu và chỉ nếu a là không bị thống trị đối với toàn bộ không gian tham số X.

Nếu tập X' không xác định rõ ràng, toàn bộ không gian tham số X sẽ được bao hàm (implied).

Những véc tơ tham số không bị thống trị không thể được cải thiện trong bất cứ mục tiêu nào mà không có gây ra một sự giảm sút (degradation) trong ít nhất một trong những mục tiêu. Chúng đại diện cho ý tưởng giải pháp tối ưu toàn cục. Chú ý rằng một tập Pareto-optimal không cần thiết chứa đựng tất cả giải pháp Pareto-optimal trong X. Tập các véc tơ mục tiêu f(a'), a' $\in$ X', tương ứng với một tập véc tơ tham số Pareto-optimal a' $\in$ X' được gọi là "ngưỡng tối ưu Pareto" ("Pareto-optimal front" or "Pareto front").

3.3 Mô tả bài toán:

Để cho tổng quát, ở đây ta cho là đi tìm max của tất cả các hàm mục tiêu.

Tìm max $f_1(x)$,..,max $f_n(x)$ với $x \in D \subset R^k$,

thỏa các ràng buộc sau:

$g_1(x) \le a_1, .., g_m(x) \le a_m$.

Vấn đề khó khăn của bài toán là các hàm $f_1(x)$,..,$f_n(x)$ không cùng đồng thời đạt max.

Sẽ tiện hơn nếu phân loại tất cả những lời giải mạnh cho bài toán tối ưu hóa đa mục tiêu thành những lời giải bị thống trị và không bị thống trị (hay Pareto-optimal). Vì lời giải x sẽ bị thống trị nếu tại đó tồn tại lời giải khả thi y không kém x trên mọi tọa độ, nghĩa là, đối với mọi hàm mục tiêu f_i(i=1,..,n):

$$f_i(x) \le f_i(y) \text{ với mọi } 1 \le i \le n$$

Nếu lời giải không bị thống trị bởi bất cứ lời giải khả thi nào khác, ta gọi nó là lời giải không bị thống trị (Pareto-optimal). Tất cả các lời giải không bị thống trị có lẽ cho ta một vài lợi ích, lý tưởng thì hệ thống có thể báo cáo lại tập của mọi điểm không bị thống trị.

Có một số phương pháp cổ điển về tối ưu hóa đa mục tiêu. Những phương pháp này bao gồm:

3.3.1 Phương pháp trọng số hóa các mục tiêu: ở đó các hàm mục tiêu f_i được tổ hợp thành một hàm mục tiêu chung F:

$$F(x) = \underset{x \in D}{Max} = \sum_{i=1}^{n} w_i f_i(x)$$

thỏa $g_j(x) \le a_j$, j=1..m

Trong đó các trọng $w_i \in [0..1]$, với w_i là mức độ quan trọng của hàm mục tiêu $f_i(x)$, và $\sum_{i=1}^{k} w_i = 1$

Cách 1: w_i do người dùng quyết định dẫn đến việc giải bán tự động.

Cách 2: w_i phát sinh tự động dẫn đến việc giải tự động.

Các véc tơ trọng khác nhau cho các lời giải không bị thống trị khác nhau.

3.3.2 Phương pháp hàm khoảng cách: tổ hợp nhiều hàm mục tiêu vào một hàm cơ sở của cấp cần có của véc tơ y.

$$F(x) = \left(\sum_{i=1}^{k} \left| f_i(x) - y_i \right|^r \right)^{1/r}$$

thường r=2 (không gian mêtric Euclide).

3.3.3 Phương pháp dùng hàm mục tiêu có thể xem là ràng buộc:

$$\underset{x \in D}{Max} f_k(x)$$

Thỏa $f_j(x) \leq \varepsilon_j$ với $\forall j$ khác k và $g_i(x) \leq a_i$ với i=1..m

Phương pháp này giải bán tự động, f_k và ε_j do người dùng quyết định.

Trong đề tài này, ta dựa vào phương pháp thứ ba này, để giải quyết những bài toán ứng dụng ở phần sau.

3.4 Những khó khăn trong tối ưu đa mục tiêu:

Trong việc mở rộng những ý tưởng EAs đơn mục tiêu để xử lý các trường hợp đa mục tiêu, có hai vấn đề chính được đặt ra:

Một là: Làm thế nào để đạt tới việc gán giá trị thích nghi và việc chọn lọc để hướng dẫn việc tìm kiếm hướng đến tập tối ưu Pareto (Pareto-optimal).

Hai là: Làm thế nào để duy trì một quần thể khác nhau để ngăn chặn việc hội tụ sớm và đạt được một sự phân phối, lan rộng tốt "trade-off front".

Chú ý rằng hàm mục tiêu tự nó không định tính được như là hàm thích nghi từ khi nó là véc tơ được lượng giá và thích nghi thành một giá trị vô hướng. Những tiếp cận khác nhau để liên hệ hàm thích nghi với hàm mục tiêu có thể được phân loại với sự quan tâm đến kết quả trước hết. Vấn đề thứ hai là thường được giải quyết bởi việc đưa ra tối ưu hóa (elitism) và sự tái kết hợp trung gian. Sự tối ưu hóa là một cách để bảo đảm rằng những cá thể tốt thì được chọn (bởi sự đột biến hoặc tái tạo), đơn giản bởi vì việc lưu trữ chúng riêng biệt trong một tập bên ngoài, những phần tử được chọn lọc.

Sự tái kết hợp trung gian, nói cách khác, trung bình những véc tơ của thế hệ trước để sản sinh một thế hệ sau, như sau:

$$x'_j=\alpha x_{j1}^{g}+(1-\alpha)x_{j2}^{g} \quad ,j,j_1,j_2 \in \{1,..,\mu\} \qquad (3.4)$$

$$x_i^{g+1} = x'_j + N(0,\Sigma) \quad ,i=1,..,\lambda \qquad ,j \in \{1,..,\mu\}$$

Sự tái kết hợp số học là một trường hợp đặc biệt của tái kết hợp trung gian, với α=0.5.

Trong đó:

μ: kích thước cá thể cha được chọn

λ: kích thước của quần thể

$N(0,\Sigma)$: một véc tơ của những số ngẫu nhiên Gaussian được phân phối như nhau giữa 0 và ma trận "co-variance" (đồng biến).

Chương 4: Ứng dụng EA cho lập phương án thi công trong xây dựng

Mở đầu:

Trong các chương trước ta nghiên cứu về Thuật giải tiến hóa EA để tối ưu hàm nhiều biến rời rạc, đồng thời ta cũng tìm hiểu về đa mục tiêu và các phương pháp để giải quyết. Trong đó kết hợp đồng thời giữa Thuật giải tiến hóa và ý tưởng của mô phỏng luyện thép.

Như ta đã biết, trong thực tế việc tìm kiếm tối ưu rất phức tạp đa dạng, đơn cử như 2 ứng dụng về lập tiến độ thi công trong xây dựng mà ta sẽ thử nghiệm trong chương này.

- Ví dụ minh họa về bài toán thi công xây dựng.
- Ứng dụng 1: Ứng dụng thực tế trong lập tiến độ thi công cầu Sơn Mãn.
- Ứng dụng 2: Ứng dụng thực tế trong lập tiến độ thi công Trụ sở Công ty tư vấn XD Bến Tre.

4.1 Giới thiệu bài toán ví dụ minh họa:

4.1.1 Đặt bài toán:

Tìm $X = (x_0, x_1, .., x_{n-1}) \in Z^n$, sao cho:

F(X) đạt được giá trị tối ưu (cực tiểu hoặc cực đại)

thỏa:

$min_1 \leq g_1(X) \leq max_1$

$min_2 \leq g_2(X) \leq max_2$

………………………

$min_m \leq g_m(X) \leq max_m$

4.1.2 Giải quyết bài toán:

Đây là bài toán với nhiều ràng buộc phức tạp, nên ta sẽ áp dụng thuật giải tiến hóa đã trình bày ở các chương trước để giải bài toán nêu trên.

4.1.2.1 Biểu diễn nhiễm sắc thể:

- Chrom=$(x_0, x_1, x_2, .., x_{n-1}) \in Z^n$.

4.1.2.2 Hàm thích nghi:

- Miền khả thi:

$$\Omega = \{X \in Z^n : min_i \leq g_i(X) \leq max_i, \forall i = 0,1,..,m-1\}$$

- Bài toán tối ưu là cực tiểu hàm đánh giá f(x) thì hàm thích nghi như sau:

$$\text{Fitness}(X)=\begin{cases} C_{\max} - f(X), & f(X) < C_{\max} \quad \textit{và} \quad X \in \Omega \\ 0, & \textit{khác} \end{cases}$$

Trong đó, C_{max} là một tham số đầu vào. Có thể lấy C_{max} là giá trị f lớn nhất trong quần thể hiện tại, hoặc lớn nhất sau k lần lặp.

4.1.2.3 Toán tử chọn cá thể:

- Tính tổng thích nghi (total fitness) của tất cả các thành viên trong quần thể:

  Total finess = $\sum_{i=1}^{N} fitness(X_i)$

  Trong đó, N là số nhiễm sắc thể trong quần thể và X_i là nhiễm sắc thể thứ i trong quần thể.

- Tính vị trí chọn từng nhiễm sắc thể:

  Pos_i = fitness(X_i) / total fitness

  Trong đó, pos_i là vị trí chọn nhiễm sắc thể thứ i, i=0,..,N-1.

- Phát sinh một số q ngẫu nhiên trong khoảng (0,1)
  - Nếu $q<pos_0$ thì nhiễm sắc thể thứ 0 được chọn.
  - Nếu $pos_{i-1} < q < pos_i$, i=1,2,..N-1 thì nhiễm sắc thể thứ i được chọn.

4.1.2.4 Toán tử lai ghép:

- Giả sử có 2 nhiễm sắc thể sau:

  Chrom1 = $(x_0,x_1,..,x_{n-1})$

  Chrom2 = $(y_0,y_1,..,y_{n-1})$

- Xác định vị trí lai ghép (i) bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].

- Hoán đổi các thành phần của chrom1 và chrom2 từ vị trí lai ghép thứ i cho đến cuối.

- Khi đó từ 2 nhiễm sắc thể ban đầu:

  Chrom1 = $(x_0,x_1,..,\mathit{x_i},..,\mathit{x_{n-1}})$

  Chrom2 = $(y_0,y_1,..,y_i,..,y_{n-1})$

  Ta được 2 nhiễm sắc thể mới:

  Chrom1 = $(x_0,x_1,..,y_i,..,y_{n-1})$

Chrom2 = $(y_0,y_1,..,x_i,..,x_{n-1})$

4.1.2.5 Toán tử đột biến:

- Giả sử ta có nhiễm sắc thể sau:

  Chrom = $(x_0,x_1,..,x_{n-1})$

- Xác định vị trí đột biến (i) bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].
- Thay phần tử x_i bằng một số nguyên ngẫu nhiên $y \in Z$.
- Khi đó từ nhiễm sắc thể ban đầu:

  Chrom = $(x_0,x_1,..,x_i,..,x_{n-1})$

  Ta được nhiễm sắc thể mới:

  Chrom = $(x_0,x_1,..,y,..,x_{n-1})$

4.1.2.6 Thuật giải:

- *Bước 1:* Tạo ngẫu nhiên quần thể (P_t=0).
- *Bước 2:* Tính giá trị thích nghi (fitness) của từng nhiễm sắc thể.
- *Bước 3:* Lặp cho đến khi thỏa điều kiện dừng:
  - Chọn k cặp nhiễm sắc thể từ P_t

    S = Select(P_t,k)

  - Lai ghép các nhiễm sắc thể trong S

    C = Crossover(S)

  - Đột biến các nhiễm sắc thể có giá trị thích nghi cao nhất

    M = Mutation(S)

  - Chọn lọc các nhiễm sắc thể có giá trị thích nghi cao nhất

    P'_t = Selection($S \cup C \cup M \cup P_t$)

  - Thay Pt bằng P'_t
  - Tăng t lên 1
- Cuối lặp.

4.1.3 Bài toán ví dụ minh họa:

4.1.3.1 Phát biểu bài toán:

- *Cho một sơ đồ mạng có n công việc và m sự kiện với các thông tin sau:*
  - Các công việc i có quan hệ logic với nhau (thứ tự trước sau).
  - Mỗi công việc i tại mỗi thời điểm sử dụng q dạng tài nguyên khác nhau

$$Q_k(i), k=0,..,q-1.$$

  - Thời gian hoàn tất công việc i ($t_{HT}(i)$) có thể dao động trong đoạn cho trước: [$Tmin_{HT}(i)$, $Tmax_{HT}(i)$].

$$t_{HT}(i) \in [Tmin_{HT}(i), Tmax_{HT}(i)], \forall i=0,1,..,n-1.$$

- *Yêu cầu:*

  - Tìm thời gian hoàn tất của các công việc ($t_{HT}(i)$).
  - Tìm các thời điểm bắt đầu ($t_{BĐ}(i)$), thời điểm kết thúc ($t_{KT}(i)$) của các công việc.

Sao cho:

  - Thời gian găng (t_{Gang}) là nhỏ nhất.
  - Tổng từng loại tài nguyên (T_{Qk}) sử dụng tại từng thời điểm là nhỏ nhất.

Đồng thời thỏa các điều kiện sau:

  - Đảm bảo thứ tự trước sau giữa các công việc:

$$order(i,j), \forall i,j=0,1,..,n-1 \text{ và } i \neq j.$$

  - $t_{HT}(i) \in t_{HT}(i) \in [Tmin_{HT}(i), Tmax_{HT}(i)], \forall i=0,1,..,n-1.$
  - $T_{Qk} \in [Qmin_k, Qmax_k], \forall k=0,..,q-1.$

Trong đó: $T_{Qk} = \sum_{i=0}^{n-1} Q_k(i)$

4.1.3.2 Các bước giải quyết bài toán:

Đây là bài toán đa mục tiêu với nhiều ràng buộc phức tạp, nên ta sẽ áp dụng thuật giải tiến hóa để giải bài toán nêu trên. Ta giải quyết thông qua hai giai đoạn:

- Giai đoạn 1:
  - Xác định $T = (t_0,t_1,..,t_{n-1})$
  - Sao cho: t_{Gang} min
  - Thỏa các ràng buộc:
    - $T_i \in [Tmin_{HT}(i),Tmax_{HT}(i)]$, $\forall i = 0,1,2,..,n-1$.
    - $\forall i = 0,1,2,..,n-1$, $\exists(t_{BĐ}(i),t_{KT}(i))$: $T_{Qk} \in [Qmin_k,Qmax_k]$, $\forall k=0,1,2..,q-1$.
    - order(i,j), $\forall i,j = 0,1,2,..n-1$ và $i \neq j$.

- Giai đoạn 2:
  - Ứng với $T = (t_0,t_1,..,t_{n-1})$ tìm được ở giai đoạn 1.
  - Xác định $(t_{BĐ}(i),t_{KT}(i))$, $\forall i = 0,1,2,..,n-1$.
  - Thỏa các ràng buộc:
    - $\forall i = 0,1,2,..,n-1$, $\exists(t_{BĐ}(i),t_{KT}(i))$: $T_{Qk} \in [Qmin_k,Qmax_k]$, $\forall k=0,1,2..,q-1$.
    - order(i,j), $\forall i,j = 0,1,2,..n-1$ và $i \neq j$.
- Giải quyết giai đoạn 1: Dùng thuật giải tiến hóa để giải quyết giai đoạn này.

- Biểu diễn nhiễm sắc thể:
  - Chrom = $(t_0,t_1,t_2,..,t_{n-1})$
  - Trong đó: $t_i \in [Tmin_{HT}(i),Tmax_{HT}(i)]$, $\forall i = 0,1,2,..,n-1$.
- Tính giá trị thích nghi cho từng nhiễm sắc thể:
  - Fitness(chrom) = f (chrom)
  - Trong đó:
    - ✓ f(chrom) = T_{Gang}(chrom), T_{Gang}(chrom) là hàm tính tGang cho chrom.
    - ✓ $\Omega = \{chrom= (t_0,t_1,t_2,..,t_{n-1}): g_1,g_2,g_3\}$

      Với:

      g_1: $Tmin_{HT}(i) \le t_i \le Tmax_{HT}(i)$, $\forall i = 0,1,2,..,n-1$.

      g_2: $\forall i=0,1,2,..,n-1$, $\exists(t_{BĐ}(i),t_{KT}(i))$:$Qmin_k \le T_{Qk} \le Qmax_k$, $\forall k=0,1,2..,q-1$.

      g_3: order(i,j), $\forall i,j = 0,1,2,..n-1$ và $i \neq j$.
- Toán tử chọn cá thể: Giả sử quần thể hiện hành có μ cá thể, $P_1(t)$ là kết quả tái sinh từ P(t) được thực hiện như sau:
  - (1) Đầu tiên ta sắp tăng dần theo f(chrom)
  - (2) Chép μ - s ($s < \mu$) cá thể tốt nhất từ P(t)
  - (3) Chép s/2 cá thể tốt nhất của P(t) vào $P_1(t)$
  - (4) Khởi tạo ngẫu nhiên s/2 cá thể còn lại cho $P_1(1)$

  Với bước (3), ta bảo đảm các cá thể thích nghi nhất có khả năng sinh sản cao.

  Bước (4) làm cho quần thể thêm phong phú.

- <u>Toán tử lai ghép:</u>
  - Giả sử có 2 nhiễm sắc thể sau:

    chrom1 = $(t_0,t_1,t_2,..,t_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t'_{n-1})$
  - Xác định vị trí lai ghép i bằng cách phát sinh một số ngẫu nhiên trong đoạn [0,n-1].
  - Hoán đổi các thành phần của chrom1 và chrom2 từ vị trí lai ghép thứ i cho đến cuối.
  - Khi đó từ 2 nhiễm sắc thể ban đầu:

    chrom1 = $(t_0,t_1,t_2,..,t_i,..,t_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t'_i,..,t'_{n-1})$

  Ta được 2 nhiễm sắc thể mới:

    chrom1 = $(t_0,t_1,t_2,..,t'_i,..,t'_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t_i,..,t_{n-1})$
- <u>Toán tử đột biến:</u>
  - <u>*Đột biến loại một:*</u>
    - Giả sử có 2 nhiễm sắc thể sau:

      chrom1 = $(t_0,t_1,t_2,..,t_{n-1})$
    - Xác định vị trí đột biến i bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].
    - Thay phần tử t_i bằng một số ngẫu nhiên $t' \in [Tmin_{HT}(i),Tmax_{HT}(i)]$.
    - Khi đó từ nhiễm sắc thể ban đầu:

      chrom2 = $(t_0,t_1,t_2,..,t_i,..,t_{n-1})$

    Ta được 2 nhiễm sắc thể mới:

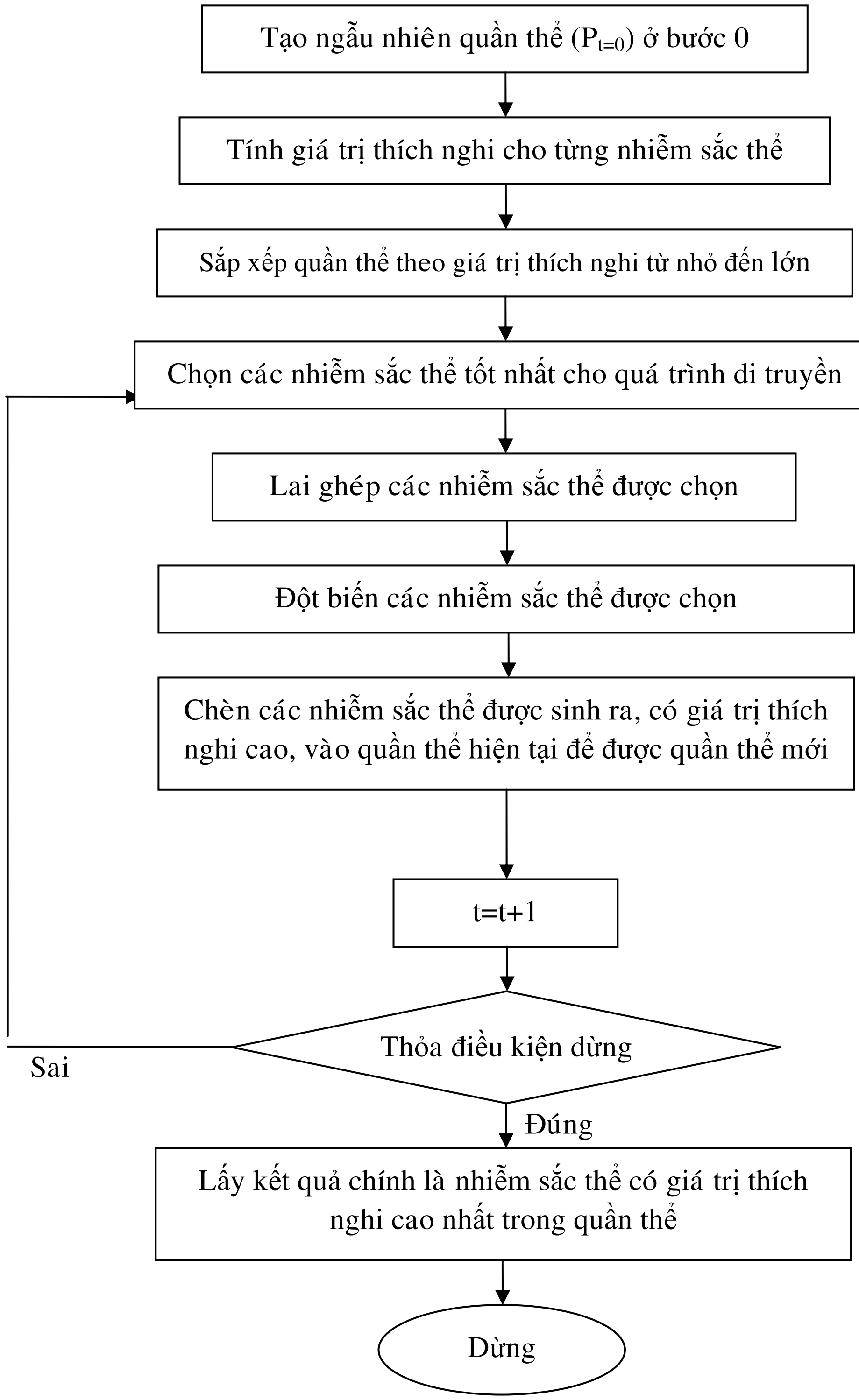
      chrom1 = $(t_0,t_1,t_2,..,t',..,t_{n-1})$

- *Đột biến loại hai:*
  - Tương tự như loại 1.
  - Điểm khác biệt so với đột biến loại 1 là: thay phần tử t_i bằng trung bình của $Tmin_{HT}(i)$ và $Tmax_{HT}(i)$

    $t' = (Tmin_{HT}(i) + Tmax_{HT}(i))/2$
- *Đột biến loại ba:* Tương tự như loại 1.
  - Điểm khác biệt so với đột biến loại 1 là: thay phần tử t_i bằng $Tmin_{HT}(i)$.

    $t' = Tmin_{HT}(i)$

- Sơ đồ giải thuật:



Hình 4.1 Lưu đồ thuật toán EA cho lập phương án thi công trong xây dựng.

Giải quyết giai đoạn 2: Dùng thuật giải tiến hóa để giải quyết.

- Biểu diễn nhiễm sắc thể:

  Chrom = $(t_0,t_1,t_2,..,t_p)$

  Trong đó, các t_i với $\forall i = 0,1,..,p$ là các thời điểm bắt đầu của các công việc không là công việc găng.

- Tính giá trị thích nghi cho từng nhiễm sắc thể:

  $$\text{Fitness(X)}=\begin{cases} C_{\max} - f(X), & f(X) < C_{\max} \quad và \quad X \in \Omega \\ 0, & khác \end{cases}$$

  Trong đó:

  - F(chrom)= $Max\left(\sum_{k=0}^{q-1} TQ_k\right)$

    Với: $T_{Qk}= \sum_{i=0,w(i)\in\Delta}^{m-1} Q_k(w(i))$, $\forall k=0,1,..,q$.

    Trong đó, q là số loại tài nguyên.

    m là số công việc đang thực hiện tại thời điểm đang xét.

    Δ là tập các công việc w(i) đang thực hiện tại thời điểm đang xét.

  - C_{max}= Max(f(chrom)), $\forall$chrom $\in$ quần thể hiện tại.
  - Ω={ chrom = $(t_0,t_1,t_2,..,t_p)$: g_1,g_2}

    Với g_1: $Qmin_k \le T_{Qk} \le Qmax_k$, $\forall k=0,..,q-1$

    g_2: order(i,j), $\forall i,j=0,1,..,n-1$ và $i\neq j$.

- Toán tử chọn cá thể: Giả sử quần thể hiện hành có μ cá thể, $P_1(t)$ là kết tái sinh từ P(t) được thực hiện như sau:

  - (1) Đầu tiên ta sắp tăng dần theo f(chrom)
  - (2) Chép μ - s (s < μ) cá thể tốt nhất từ P(t)
  - (3) Chép s/2 cá thể tốt nhất của P(t) vào $P_1(t)$

- (4) Khởi tạo ngẫu nhiên s/2 cá thể còn lại cho $P_1(1)$

Với bước (3), ta bảo đảm các cá thể thích nghi nhất có khả năng sinh sản cao.

Bước (4) làm cho quần thể thêm phong phú.

- Toán tử lai ghép:
  - Giả sử có 2 nhiễm sắc thể sau:

    chrom1 = $(t_0,t_1,t_2,..,t_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t'_{n-1})$

  - Xác định vị trí lai ghép i bằng cách phát sinh một số ngẫu nhiên trong đoạn [0,n-1].
  - Hoán đổi các thành phần của chrom1 và chrom2 từ vị trí lai ghép thứ i cho đến cuối.
  - Khi đó từ 2 nhiễm sắc thể ban đầu:

    chrom1 = $(t_0,t_1,t_2,..,t_i,..,t_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t'_i,..,t'_{n-1})$

  Ta được 2 nhiễm sắc thể mới:

    chrom1 = $(t_0,t_1,t_2,..,t'_i,..,t'_{n-1})$

    chrom2 = $(t'_0,t'_1,t'_2,..,t_i,..,t_{n-1})$

- Toán tử đột biến:
  - Đột biến loại một:
    - Giả sử có 2 nhiễm sắc thể sau:

      chrom1 = $(t_0,t_1,t_2,..,t_{n-1})$

    - Xác định vị trí đột biến i bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].

- Thay phần tử t_i bằng một số ngẫu nhiên $t' \in [Tmin_{HT}(i), Tmax_{HT}(i)]$.
- Khi đó từ nhiễm sắc thể ban đầu:

  chrom2 = $(t_0,t_1,t_2,..,t_i,..,t_{n-1})$

  Ta được 2 nhiễm sắc thể mới:

  chrom1 = $(t_0,t_1,t_2,..,t',..,t_{n-1})$

- <u>Đột biến loại hai:</u>
- Tương tự như loại 1.
- Điểm khác biệt so với đột biến loại 1 là: thay phần tử t_i bằng trung bình của $Tmin_{HT}(i)$ và $Tmax_{HT}(i)$

  $t' = (Tmin_{HT}(i) + Tmax_{HT}(i))/2$

- <u>Đột biến loại ba:</u>
- Tương tự như loại 1.
- Điểm khác biệt so với đột biến loại 1 là: thay phần tử t_i bằng $Tmin_{HT}(i)$.

  $t' = Tmin_{HT}(i)$

- <u>Các thông số sơ đồ của sơ đồ mạng:</u>

  Gọi $t_{KhS}(i)$, $t_{KS}(i)$, $t_{KhM}(i)$, $t_{KM}(i)$, $t_{HT}(i)$ lần lượt là thời điểm khởi sớm, kết sớm, khởi muộn, kết muộn và thời gian hoàn tất thứ i.

  Gọi tGang là thời gian găng.

  Gọi order(i,j) nói lên công việc i phải được thực hiện trước công việc j.

  - Tính khởi sớm và kết sớm của các công việc:
    - Khởi sớm và kết sớm của các công việc bắt đầu (các công việc không có công việc trước nó).

      $t_{KhS}(i)=0$

      $t_{KS}(i)=t_{KhS}(i)+t_{HT}(i)$

- Khởi sớm và kết sớm của các công việc khác (các công việc có công việc trước nó).

  $t_{KhS}(i)=Max(t_{KS}(j))$ với order(j,i), $j \neq i$

  $t_{KS}(i)=t_{KhS}(i)+t_{HT}(i)$

- Tính tGang:

  tGang=Max($t_{KS}(i)$) với $\forall i=0,1,..,n-1$.

- Tính khởi muộn và kết muộn của các công việc:

  - Khởi muộn và kết muộn của các công việc sau cùng (các công việc không có công việc sau nó).

    $t_{KM}(i)=t_{Gang}$

    $t_{KhM}(i)=t_{KM}(i)-t_{HT}(i)$

  - Khởi muộn và kết muộn của các công việc khác (các công việc có công việc sau nó).

    $t_{KM}(i)=Min(t_{KhS}(j))$ với order(i,j), $j \neq i$

    $t_{KhM}(i)=t_{KM}(i)-t_{HT}(i)$

4.1.3.3 Số liệu cụ thể:

Cho bài toán ví dụ sau:

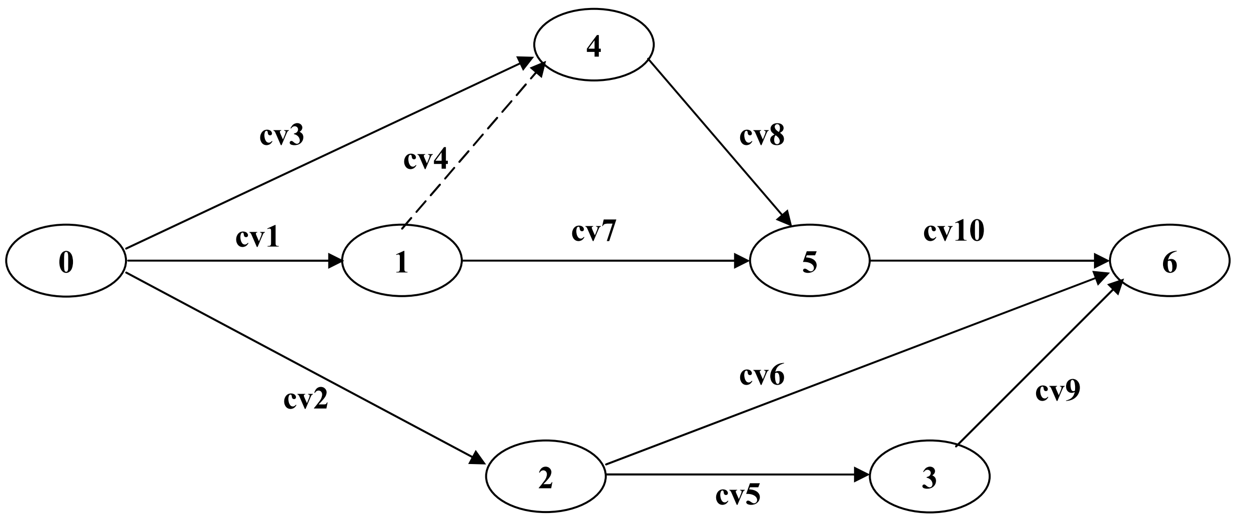
Có 10 công việc (được nêu trong bảng 4.1).

Thời gian được tính bằng ngày.

Số lao động, kinh phí, vật tư tính trên một ngày.

Bảng 4.1 Bảng các công việc của bài toán thi công.

STT	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn tất tối thiểu	Thời gian hoàn tất tối đa	Lao động	Kinh phí	Vật tư
1	cv 0-1	0	1	10	22	10	2	1
2	cv 0-2	0	2	9	25	12	4	2
3	cv 0-4	0	4	5	15	25	7	4
4	cv 1-4	1	4	0	0	0	0	0
5	cv 2-3	2	3	5	17	30	9	0
6	cv 2-6	2	6	5	19	26	7	3
7	cv 1-5	1	5	6	20	15	6	4
8	cv 4-5	4	5	7	22	10	5	2
9	cv 3-6	3	6	7	19	7	4	3
10	cv 5-6	5	6	10	25	5	3	1



Hình 4.2 Sơ đồ quan hệ giữa các công việc

Yêu cầu:

- ✓ Tìm thời gian hoàn tất của các công việc (trong khoảng thời gian hoàn tất tối thiểu và thời gian hoàn tất tối đa tương ứng).
- ✓ Tìm các thời điểm bắt đầu, thời điểm kết thúc của các công việc.

Sao cho:

- ✓ Thời gian găng là nhỏ nhất.
- ✓ Đảm bảo quan hệ trước sau giữa các công việc.
- ✓ Tổng lao động sử dụng trong ngày, tổng kinh phí sử dụng trong ngày và tổng thiết bị sử dụng trong ngày nằm trong khoảng cho phép, theo bảng sau:

Bảng 4.2 Bảng các ràng buộc

Tên ràng buộc	Giá trị tối thiểu	Giá trị tối đa
Lao động	5	38
Kinh phí	2	17
Vật tư	1	9

Áp dụng theo thuật giải tiến hóa đã trình bày, với các thông số sau:

	EA1	EA2
popSize	200	200
pCross	0.75	0.75
pMu	0.02	0.02

Ta tìm được thời gian Gang tối ưu là = 32.

Bảng 4.3 Bảng kết quả phân bố các công việc

STT	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn tất	Thời điểm bắt đầu	Thời điểm kết thúc	Lao động	Kinh phí	Vật tư
1	cv 0-1	0	1	10	0	10	10	2	1
2	cv 0-2	0	2	9	11	20	12	4	2
3	cv 0-4	0	4	10	0	10	25	7	4
4	cv 1-4	1	4	0	10	10	0	0	0
5	cv 2-3	2	3	10	10	20	30	9	0
6	cv 2-6	2	6	5	20	25	26	7	3
7	cv 1-5	1	5	6	26	32	15	6	4
8	cv 4-5	4	5	10	10	20	7	4	3
9	cv 3-6	3	6	7	25	32	10	5	2
10	cv 5-6	5	6	12	20	32	5	3	1

4.1.3.4 Kết luận

Trong các phần trên trình bày việc áp dụng Thuật giải tiến hóa để giải quyết bài toán tối ưu đa mục tiêu, nhiều ràng buộc và nhiều biến tổng quát. Tiến tới, chỉ ra tính khả thi của phương pháp trong bài toán ví dụ: Lập sơ đồ mạng biểu diễn mối quan hệ của m công việc sao cho thời gian găng nhỏ nhất mà vẫn thỏa mãn các ràng buộc:

- ✓ Đảm bảo quan hệ trước sau giữa các công việc.
- ✓ Đảm bảo mức độ cho phép về tổng lao động, tổng kinh phí, tổng thiết bị sử dụng trong ngày.

4.2 Ứng dụng thực tế trong lập tiến độ thi công cầu Sơn Mãn

4.2.1 Mở đầu:

Các chương trước ta đã tìm hiểu về cơ sở lý thuyết của thuật giải di truyền và biến thể của nó, sơ đồ mạng để lập phương án thi công trong xây dựng. Chương này ta ứng dụng vào bài toán cụ thể "lập tiến độ thi công cầu sơn Mãn".

Cầu Sơn Mãn nằm trên tuyến đường sắt Hà Nội – Lào Cai tại km 291+167 trong một địa hình không trở ngại nhiều cho việc thi công xây dựng. Tuyến đường nằm bên tả ngạn sông Hồng, cách bờ sông không xa. Đường quốc lộ 1B Hà Nội – Lào Cai nằm ở phía Đông Bắc, có đường mòn đi men theo bờ suối, khi đắp sửa sơ bộ, ô tô có thể chở vật tư, thiết bị đến tận chân công trình. Đơn vị thi công xây dựng cầu trực thuộc xí nghiệp xây dựng công trình 202. Đơn vị thi công đã đưa ra bảng liệt kê các công việc cần thiết để hoàn thành cầu Sơn Mãn.

Bài toán đặt ra là làm sao phân bố các công việc thi công cầu Sơn Mãn sao cho:

- Thời gian hoàn tất các công việc là nhỏ nhất (Tgang).
- Số công tại mỗi thời điểm là ít nhất.

Với việc bảo đảm thứ tự trước sau của các công việc.

Và phải cho biết các yêu cầu sau:

- Tìm thời gian hoàn tất của các công việc.
- Tìm các thời điểm bắt đầu và kết thúc của từng công việc.

Bảng 4.4 Bảng các công việc cầu Sơn Mãn

STT	WorkName	FirstEvent	LastEvent	MinCompleteTime	MaxCompleteTime	Worker
1	cv 0-1	0	1	30	35	6
2	cv 0-3	0	3	30	40	11
3	cv 1-2	1	2	0	0	0
4	cv 1-3	1	3	15	25	10
5	cv 2-3	2	3	10	25	10
6	cv 3-4	3	4	3	10	16
7	cv 3-5	3	5	2	5	6
8	cv 4-5	4	5	10	15	30
9	cv 5-6	5	6	1	5	18
10	cv 5-7	5	7	1	3	9
11	cv 6-7	6	7	0	0	0
12	cv 7-8	7	8	2	5	30
13	cv 8-9	8	9	2	5	9
14	cv 8-10	8	10	2	5	3
15	cv 8-11	8	11	5	9	4
16	cv 8-13	8	13	5	11	18
17	cv 9-10	9	10	0	0	0
18	cv 9-12	9	12	3	8	16
19	cv 10-11	10	11	0	0	0
20	cv 11-12	11	12	0	0	0
21	cv 11-13	11	13	3	9	9
22	cv 12-14	12	14	2	5	10
23	cv 12-14	12	14	5	10	20
24	cv 13-14	13	14	0	0	0
25	cv 13-15	13	15	5	10	30
26	cv 14-15	14	15	2	7	6
27	cv 14-15	14	15	0	0	0
28	cv 15-16	15	16	5	10	4
29	cv 15-17	15	17	1	3	30
30	cv 15-22	15	22	5	14	12
31	cv 16-18	16	18	2	5	7
32	cv 17-19	17	19	5	10	4
33	cv 17-19	17	19	2	8	2
34	cv 17-20	17	20	10	13	22
35	cv 18-19	18	19	0	0	0
36	cv 18-21	18	21	5	10	15
37	cv 19-22	19	22	2	5	9
38	cv 20-22	20	22	0	0	0
39	cv 20-24	20	24	2	5	1
40	cv 21-25	21	25	10	19	25
41	cv 21-30	21	30	0	0	0
42	cv 22-23	22	23	6	15	30
43	cv 23-24	23	24	0	0	0
44	cv 23-26	23	26	5	10	4
45	cv 24-27	24	27	5	15	15

46	cv 25-38	25	38	20	30	10
47	cv 26-28	26	28	1	5	10
48	cv 27-29	27	29	2	5	9
49	cv 28-30	28	30	3	8	30
50	cv 29-31	29	31	3	7	4
51	cv 30-31	30	31	0	0	0
52	cv 30-36	30	36	5	16	25
53	cv 31-32	31	32	1	4	11
54	cv 32-33	32	33	1	5	10
55	cv 33-34	33	34	5	10	10
56	cv 34-35	34	35	2	5	10
57	cv 35-37	35	37	1	3	10
58	cv 36-38	36	38	10	15	20
59	cv 37-38	37	38	5	15	12

4.2.2 Mô hình bài toán:

4.2.2.1 Phân tích bài toán:

Bài toán trên được biểu diễn dưới dạng sơ đồ mạng, với các công việc bảo đảm có thứ tự trước sau.

Với hai mục tiêu:

4.2.2.1.1 ***Mục tiêu thứ nhất:*** Thời gian hoàn tất cả công trình là ít nhất

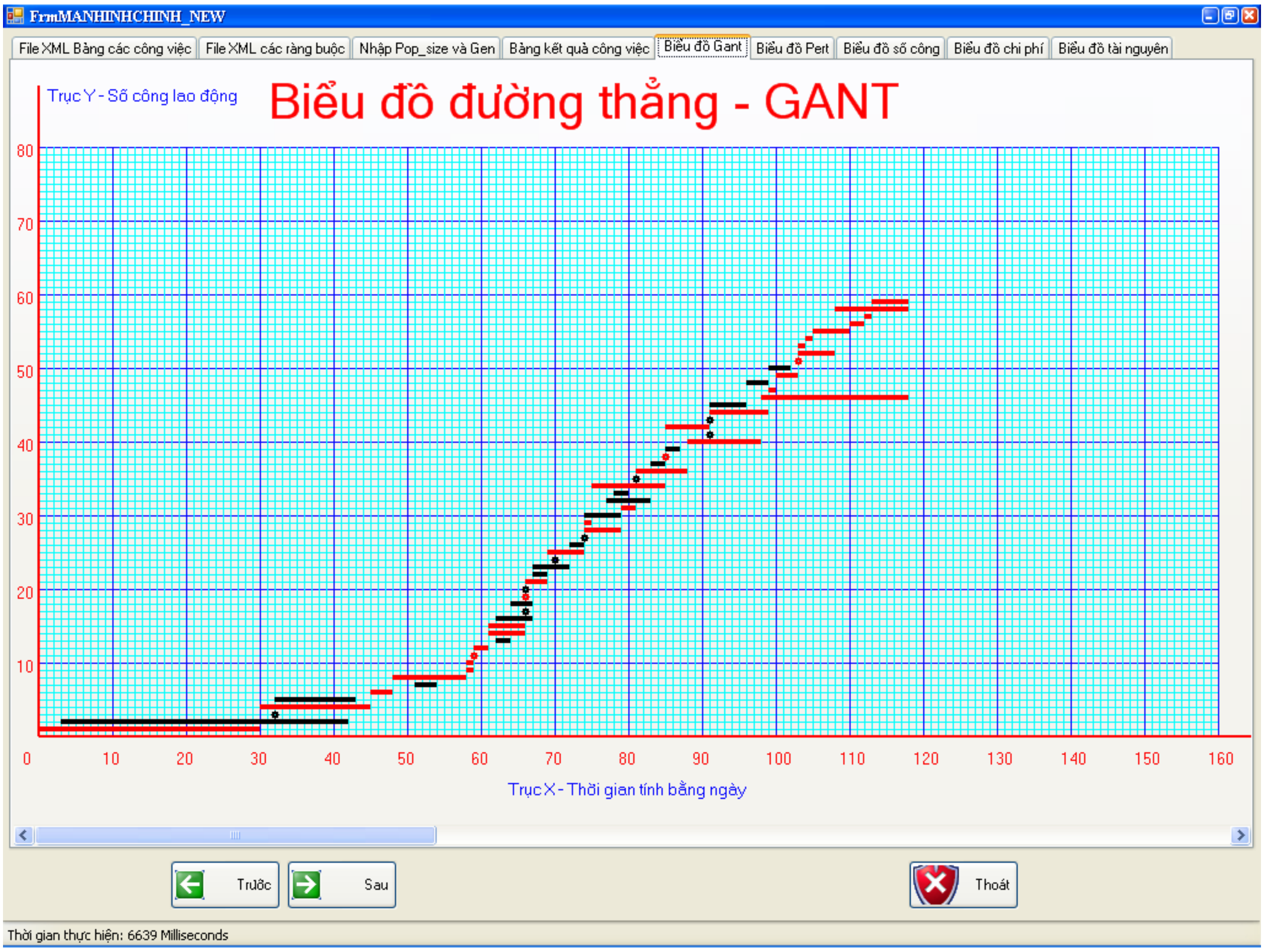
$$F1=Tgang = \sum_{i=0}^{n-1} t_i$$

Sao cho Tgang là nhỏ nhất (min).

Với n là số công việc.

Với $t_i \in [Tmin_{HT}(i), Tmax_{HT}(i)]$.

Và $Tmin_{HT}(i), Tmax_{HT}(i)$ là số liệu đã cho ở bảng 4.4.



<u>Hình 4.3</u> Biểu diễn Tgăng

4.2.2.1.2 ***Mục tiêu thứ hai:*** Số công tại mỗi thời điểm là ít nhất.

Ứng với mục tiêu thứ nhất, ta có:

$F2 = Max(TQ_1, TQ_2, .., TQ_k)$, với k=Tgang.

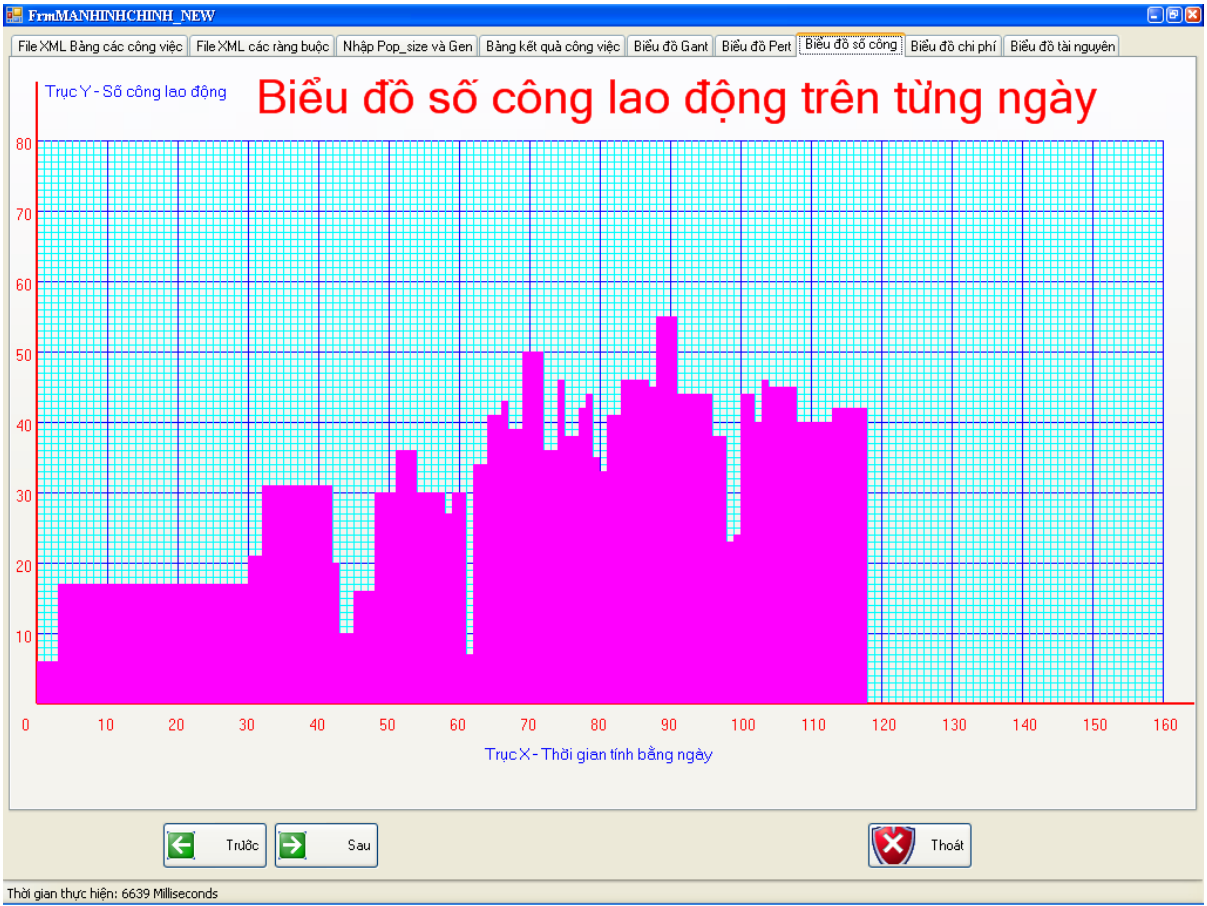
Với $TQ_k = \sum_{\substack{i=1 \\ w(i)\in\Delta}}^{m} Q_k(w(i))$

Trong đó:

m: là số công việc tại thời điểm đang xét thứ k.

Δ: là tập các công việc w(i) tại thời điểm đang xét k.

Q_k: là số công lao động của công việc w(i), được nêu trong bảng 4.4



Hình 4.4 – Biểu diễn số công lao động trên từng ngày

Với các ràng buộc sau:

$\Omega = \{g_1, g_2, g_3\}$

- g_1: $Tmin_{HT}(i) <= t_i <= Tmax_{HT}(i)$, $\forall i=0,1,2,..,n-1$
- g_2: $\forall i=0,1,2,..,n-1$, $\exists (T_{BĐ}(i), T_{KT}(i))$: $TQ_k \in [Qmin_k, Qmax_k]$.
- g_3: order(i,j), i=0,1,2,..,n-1 và $i \neq j$.

4.2.2.2 *Giải quyết bài toán:*

Vấn đề đặt ra là ta phải trả lời hai yêu cầu sau:

- Tìm thời gian hoàn tất của các công việc.
- Tìm các thời điểm bắt đầu và kết thúc của từng công việc.

Do đó bài toán ta phân làm hai phần:

- Phần 1: Xác định T1 = $(t_0,t_1,t_2,..,t_{n-1})$, với n là số công việc.

  Sao cho: t_{gang} min.

  Với t_i là thời gian hoàn tất công việc thứ i, $\forall i=0,1,2,..,n-1$.

  Thỏa mãn các ràng buộc:

  - g_1: $Tmin_{HT}(i) <= t_i <= Tmax_{HT}(i)$, $\forall i=0,1,2,..,n-1$
  - g_2: $\forall i=0,1,2,..,n-1$, $\exists (T_{BĐ}(i),T_{KT}(i))$: $TQ_k \in [Qmin_k,Qmax_k]$.
  - g_3: order(i,j), i=0,1,2,..,n-1 và $i \neq j$.
- Phần 2: Xác định T2 = $(t_{BĐ0},t_{BĐ1},t_{BĐ2},..,t_{BĐn-1})$, ứng với T1 = $(t_0,t_1,t_2,..,t_{n-1})$.

  Với $t_{BĐi}$ là thời điểm bắt đầu công việc thứ i, $\forall i=0,1,2,..,n-1$.

  Thỏa mãn các ràng buộc:

  - g_2: $\forall i=0,1,2,..,n-1$, $\exists (T_{BĐ}(i),T_{KT}(i))$: $TQ_k \in [Qmin_k,Qmax_k]$.
  - g_3: order(i,j), i=0,1,2,..,n-1 và $i \neq j$.

Do độ phức tạp của bài toán, ta chọn thuật giải tiến hóa (EA) kết hợp với ý tưởng mô phỏng luyện thép để giải quyết bài toán này.

Trong đó:

- Giai đoạn 1: dùng thuật giải tiến hóa EA và kết hợp ý tưởng mô phỏng luyện thép.
- Giai đoạn 2: dùng ý tưởng mô phỏng luyện thép.

Tuy nhiên trong quá trình thực hiện, ta thực hiện đồng thời cả hai giai đoạn.

4.2.2.3 Các bước thực hiện:

- Biểu diễn nhiễm sắc thể:
    - Chrom = $(t_0,t_1,t_2,..,t_{n-1})$, ứng với giai đoạn 1: tìm thời gian hoàn tất của tất cả các công việc.
        - Trong đó: $t_i \in [Tmin_{HT}(i),Tmax_{HT}(i)]$, $\forall i = 0,1,2,..,n-1$.
        - n là số công việc.
- Hàm thích nghi:
    - Fitness(chrom) = f (chrom)
    - Trong đó:
  - ✓ f(chrom) = T_{Gang}(chrom), T_{Gang}(chrom) là hàm tính tGang cho chrom.
  - ✓ $\Omega = \{chrom= (t_0,t_1,t_2,..,t_{n-1}): g_1,g_2,g_3\}$

    Với:

    g_1: $Tmin_{HT}(i) \leq t_i \leq Tmax_{HT}(i)$, $\forall i = 0,1,2,..,n-1$.

    g_2: $\forall i=0,1,2,..,n-1$, $\exists(t_{BĐ}(i),t_{KT}(i))$:$Qmin_k \leq T_{Qk} \leq Qmax_k$, $\forall k=0,1,2..,q-1$.

    g_3: order(i,j), $\forall i,j = 0,1,2,..n-1$ và $i \neq j$.
- Toán tử chọn lọc (select):

  Giả sử quần thể hiện hành có μ cá thể, $P_1(t)$ là kết tái sinh từ P(t) được thực hiện như sau:

    - (1) Đầu tiên ta sắp tăng dần theo f(chrom) và Qmax (số công lớn nhất).
    - (2) Chép μ - s (s < μ) cá thể tốt nhất từ P(t).

  s: là số cá thể được chọn để lai ghép và đột biến.

- Toán tử lai ghép (crossover):

  Với s là số cá thể được lai ghép, lần lượt ta ghép từng cặp nhiễm sắc thể. Giả sử ta có 2 nhiễm sắc thể:

  Chrom1 = $(x_0,x_1,x_2,..,x_{n-1})$.

  Chrom2 = $(y_0,y_1,y_2,..,y_{n-1})$

  Ta phát sinh một số ngẫu nhiên $i \in [1,n-1]$ cho biết vị trí trong mỗi cá thể cha mẹ được lai ghép.

  Chrom1 = $(x_0,x_1,x_2,..,x_i,..,x_{n-1})$.

  Chrom2 = $(y_0,y_1,y_2,..,y_i,..,y_{n-1})$

  Hai cá thể con sẽ là:

  Chrom1 = $(x_0,x_1,x_2,.., y_i,..,y_{n-1})$.

  Chrom2 = $(y_0,y_1,y_2,.., x_i,..,x_{n-1})$.

- Toán tử đột biến (mutation):

  Do đặc thù của bài toán, toán tử đột biến là được quan tâm nhiều nhất, bởi vì các ràng buộc đã phân tích ở trên. Ta chọn phương án đột biến sau:

  - Giả sử có 1 nhiễm sắc thể sau:

    chrom1 = $(t_0,t_1,t_2,..,t_{n-1})$

  - Xác định vị trí đột biến i bằng cách phát sinh một số nguyên ngẫu nhiên trong đoạn [0,n-1].
  - Thay phần tử t_i bằng $t' = Tmin_{HT}(i)$.
  - Khi đó từ nhiễm sắc thể ban đầu:

    Chrom1 = $(t_0,t_1,t_2,..,t_i,..,t_{n-1})$

  Ta được 1 nhiễm sắc thể mới:

    Chrom2 = $(t_0,t_1,t_2,..,t',..,t_{n-1})$

- Thuật giải:

  Bước 1: Tạo ngẫu nhiên quần thể (P_t=0).

  Bước 2: Tính giá trị thích nghi (fitness) của từng nhiễm sắc thể.

  Bước 3: Lặp cho đến khi thỏa điều kiện dừng:

  Chọn k cặp nhiễm sắc thể từ P_t: S = Select(P_t,k)

  Lai ghép các nhiễm sắc thể trong S: C = Crossover(S)

  Đột biến các nhiễm sắc thể có giá trị thích nghi cao nhất

  M = Mutation(S)

  Chọn lọc các nhiễm sắc thể có giá trị thích nghi cao nhất

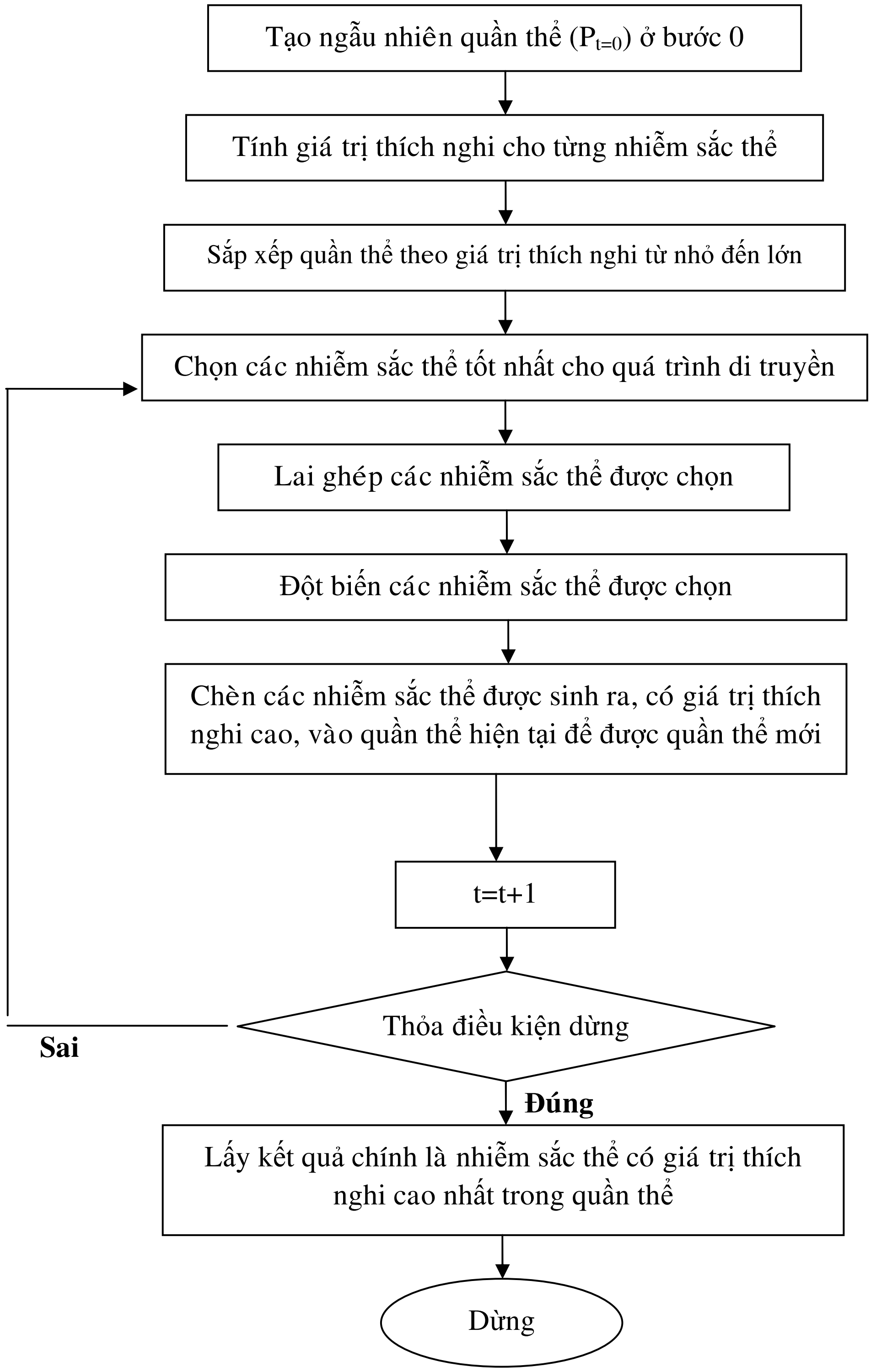
  $P'_t = \text{Selection}(S \cup C \cup M \cup P_t)$

  Thay Pt bằng P'_t

  Tăng t lên 1

  Cuối lặp.

- <u>Sơ đồ thuật giải:</u>



<u>Hình 4.5</u> – Sơ đồ thuật giải bài toán thi công xây dựng

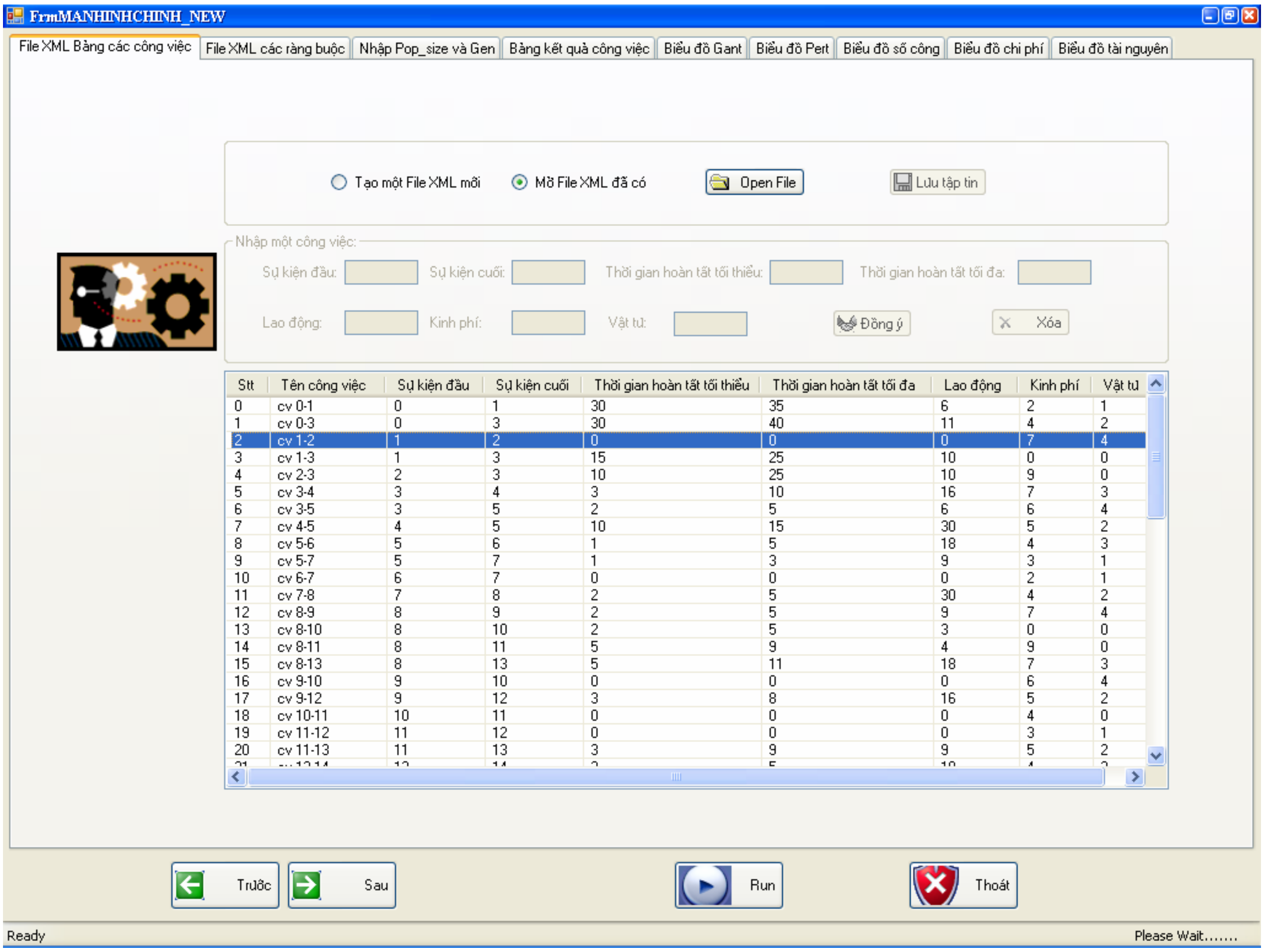
4.2.3 <u>Thử nghiệm cài đặt:</u>

4.2.3.1 Giới thiệu:

- Môi trường: Chương trình được viết trên hệ điều hành Window XP.
- Ngôn ngữ cài đặt: dùng C# của bộ Microsoft Visual 2005 (Ver 8.0).
- Dữ liệu lưu trữ: dùng tập tin XML được hỗ trợ bởi Microsoft Visual 2005.

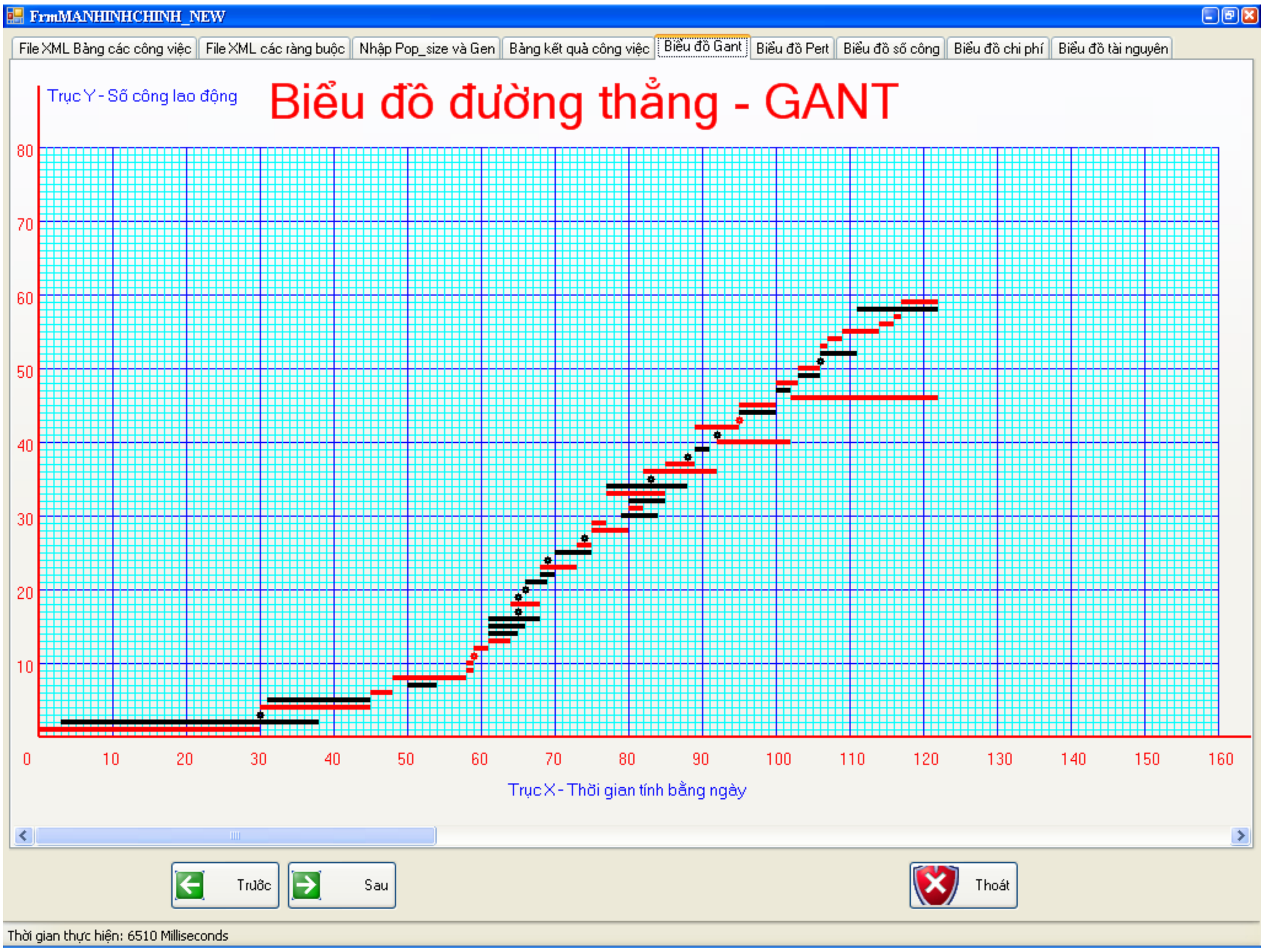
4.2.3.2 Kết quả đạt được:

- Màn hình nhập liệu:



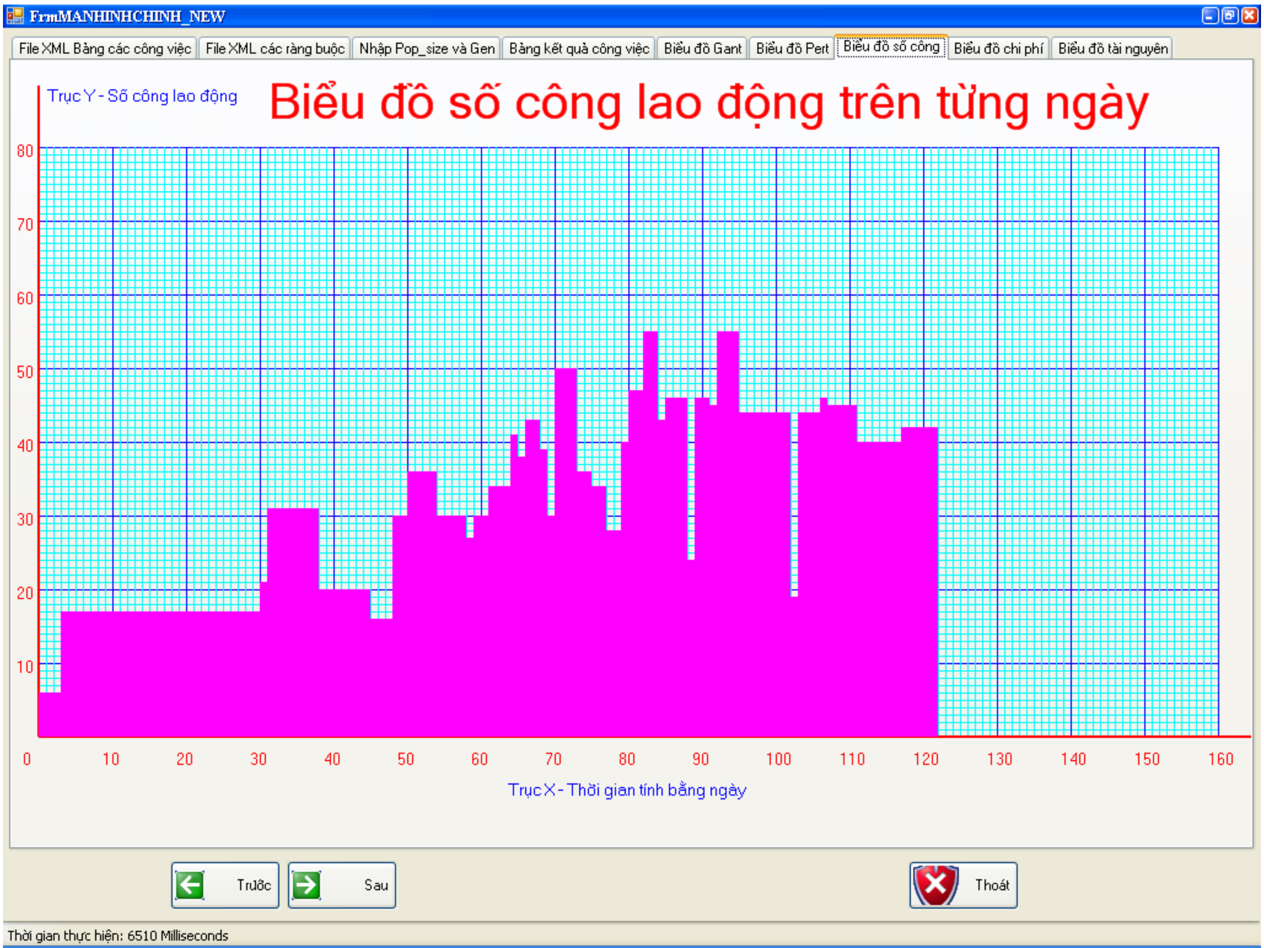
<u>Hình 4.6</u> – Màn hình nhập liệu từ file dữ liệu.

- Màn hình kết quả sơ đồ đường thẳng (GANT):



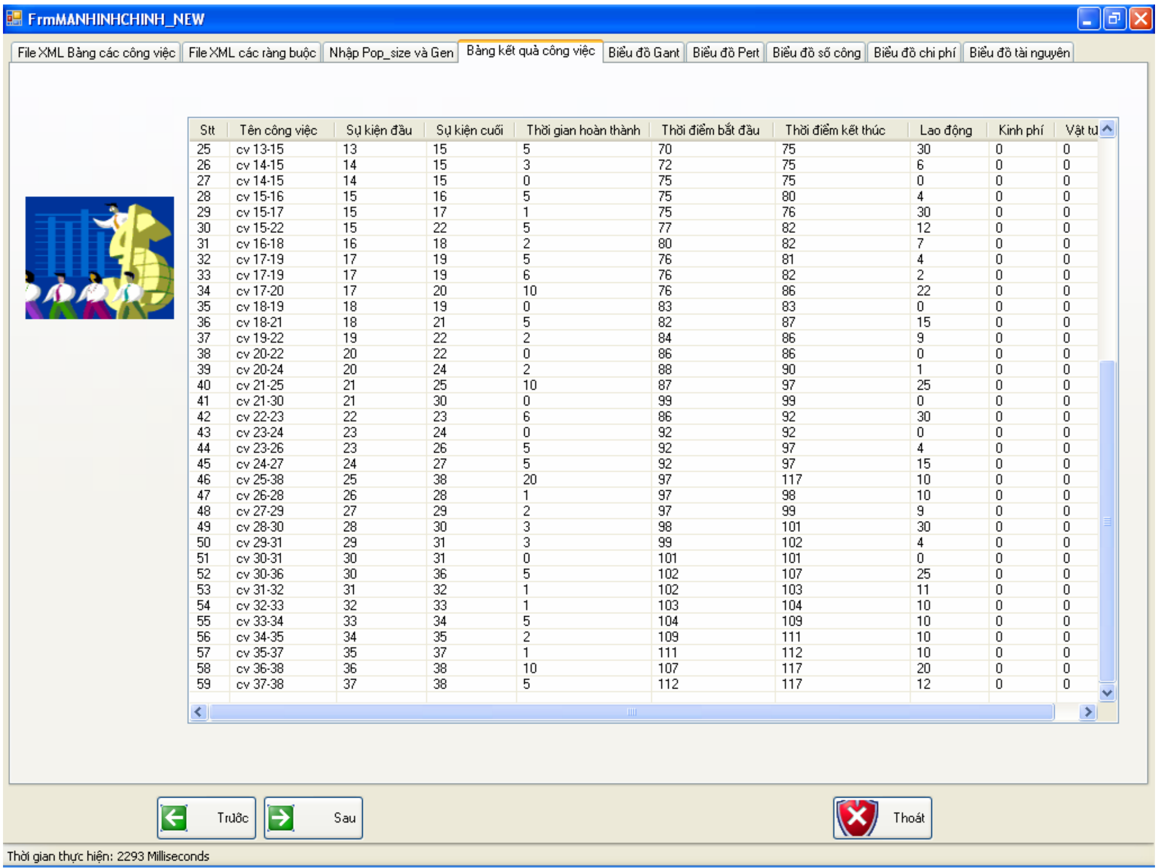
Hình 4.7 – Màn hình kết quả biểu diễn đường GANT.

- Màn hình kết quả số công lao động tính trên từng ngày.



Hình 4.8 – Màn hình kết quả số công lao động tính trên từng ngày.

- Màn hình bảng kết quả các công việc.



Stt	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn thành	Thời điểm bắt đầu	Thời điểm kết thúc	Lao động	Kinh phí	Vật tư
25	cv 13-15	13	15	5	70	75	30	0	0
26	cv 14-15	14	15	3	72	75	6	0	0
27	cv 14-15	14	15	0	75	75	0	0	0
28	cv 15-16	15	16	5	75	80	4	0	0
29	cv 15-17	15	17	1	75	76	30	0	0
30	cv 15-22	15	22	5	77	82	12	0	0
31	cv 16-18	16	18	2	80	82	7	0	0
32	cv 17-19	17	19	5	76	81	4	0	0
33	cv 17-19	17	19	6	76	82	2	0	0
34	cv 17-20	17	20	10	76	86	22	0	0
35	cv 18-19	18	19	0	83	83	0	0	0
36	cv 18-21	18	21	5	82	87	15	0	0
37	cv 19-22	19	22	2	84	86	9	0	0
38	cv 20-22	20	22	0	86	86	0	0	0
39	cv 20-24	20	24	2	88	90	1	0	0
40	cv 21-25	21	25	10	87	97	25	0	0
41	cv 21-30	21	30	0	99	99	0	0	0
42	cv 22-23	22	23	6	86	92	30	0	0
43	cv 23-24	23	24	0	92	92	0	0	0
44	cv 23-26	23	26	5	92	97	4	0	0
45	cv 24-27	24	27	5	92	97	15	0	0
46	cv 25-38	25	38	20	97	117	10	0	0
47	cv 26-28	26	28	1	97	98	10	0	0
48	cv 27-29	27	29	2	97	99	9	0	0
49	cv 28-30	28	30	3	98	101	30	0	0
50	cv 29-31	29	31	3	99	102	4	0	0
51	cv 30-31	30	31	0	101	101	0	0	0
52	cv 30-36	30	36	5	102	107	25	0	0
53	cv 31-32	31	32	1	102	103	11	0	0
54	cv 32-33	32	33	1	103	104	10	0	0
55	cv 33-34	33	34	5	104	109	10	0	0
56	cv 34-35	34	35	2	109	111	10	0	0
57	cv 35-37	35	37	1	111	112	10	0	0
58	cv 36-38	36	38	10	107	117	20	0	0
59	cv 37-38	37	38	5	112	117	12	0	0

Hình 4.9 – Màn hình kết quả các công việc.

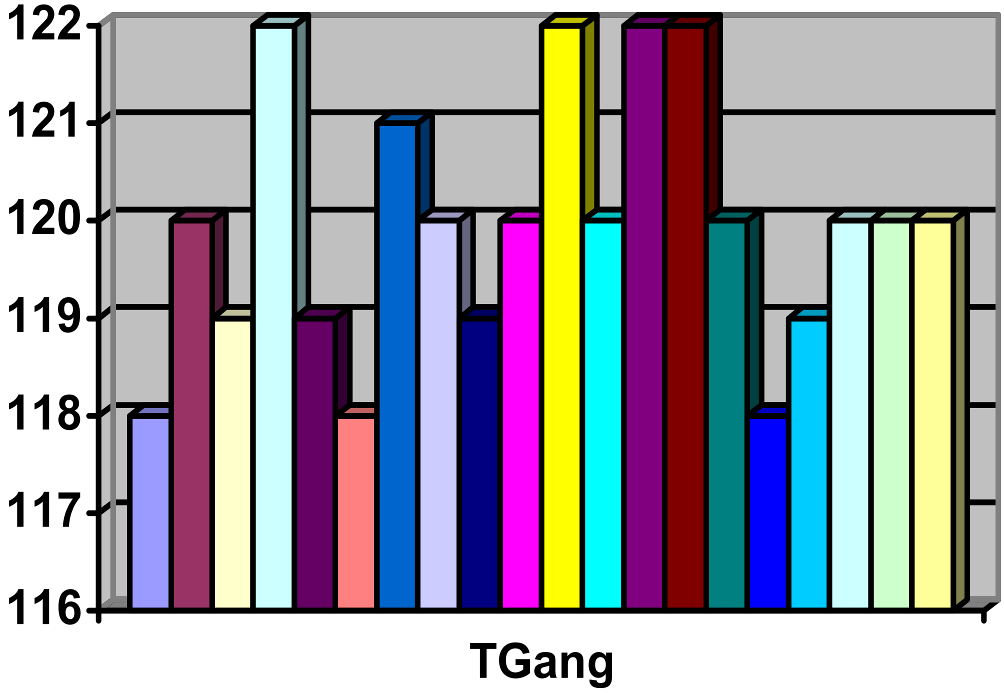
4.2.3.3 Bảng so sánh:

- Với số liệu của bảng 4.4
- Sau 20 lần thử nghiệm.
- Popsize=50.
- Số lần tạo sinh Gen=50.

Ta có bảng so sánh sau:

Bảng 4.5 Bảng so sánh các thử nghiệm.

Lần thứ	TGăng	Số công tối đa trong 1 ngày
1	118	55
2	120	55
3	119	55
4	122	55
5	119	55
6	118	55
7	121	55
8	120	55
9	119	55
10	120	55
11	122	55
12	120	55
13	122	55
14	122	55
15	120	55
16	118	55
17	119	55
18	120	55
19	120	55
20	120	55



Hình 4.10 – Biểu đồ biểu diễn Tgang của 20 lần thể nghiệm

4.2.4 Kết luận:

Sau 20 lần thử nghiệm, kết quả bảng so sánh ta thấy:

- Thấp nhất: TGăng = 118, với số lần: 3 lần.
- Cao nhất: TGăng = 122, với số lần: 4 lần.
- Trung bình: TGăng = 120, với số lần: 8 lần.

Với kết quả nêu trên, ta có thể chấp nhận được giải pháp lựa chọn "thuật giải tiến hóa (EA) kết hợp với ý tưởng mở phỏng luyện thép để giải quyết bài toán này".

4.3 Ứng dụng thực tế trong lập tiến độ thi công Trụ sở Công ty tư vấn XD Bến Tre:

4.3.1 Giới thiệu:

- Quy mô công trình:

  Công trình có diện tích xây dựng 22m x 14m, 6 tầng (có thêm tầng lững và tầng mái), khung bê tông cốt thép, móng cọc ép, một số móng nhỏ dùng cừ tràm, tường gạch sơn nước. Biện pháp thi công là kết hợp cơ giới và thủ công.

- Khối lượng công tác:

  Với mỗi công tác ta xác định khối lượng dựa trên các bản vẽ kiến trúc và kết cấu có sẵn.

- Lập tiến độ:

  Trên cơ sở một số công tác chính của công trình như thi công phần móng, thi công phần khung bê tông cốt thép, xây tường, lắp cửa, trát tường, sơn tường, lát gạch và thi công cầu thang. Về tài nguyên cho mỗi công tác: ta chỉ quan tâm chính đến số công lao động của mỗi công việc.

4.3.2 Bảng công việc: với 294 hạng mục công việc.

Bảng 4.6 Bảng công việc của Trụ sở Bến Tre.

STT	WorkName	FirstEvent	LastEvent	MinCompleteTime	MaxCompleteTime	Worker
1	cv 0-1	0	1	12	16	17
2	cv 1-2	1	2	3	5	2
3	cv 2-3	2	3	6	8	4
4	cv 2-4	2	4	3	5	9
5	cv 2-5	2	5	7	10	10
6	cv 3-5	3	5	0	0	0
7	cv 4-5	4	5	0	0	0
8	cv 5-6	5	6	2	3	7
9	cv 6-7	6	7	2	3	13
10	cv 7-8	7	8	2	3	8
11	cv 8-9	8	9	3	5	10
12	cv 8-10	8	10	3	5	5
13	cv 8-19	8	19	2	3	14

14	cv 8-20	8	20	2	5	11
15	cv 9-10	9	10	0	0	0
16	cv 10-11	10	11	2	3	17
17	cv 11-12	11	12	5	8	9
18	cv 12-13	12	13	5	8	9
19	cv 13-14	13	14	1	2	6
20	cv 13-15	13	15	2	3	7
21	cv 14-15	14	15	0	0	0
22	cv 15-16	15	16	1	2	4
23	cv 16-17	16	17	1	2	6
24	cv 17-18	17	18	6	9	20
25	cv 18-19	18	19	0	0	0
26	cv 19-20	19	20	0	0	0
27	cv 20-21	20	21	2	3	12
28	cv 21-22	21	22	2	3	10
29	cv 21-23	21	23	1	2	15
30	cv 21-68	21	68	2	2	0
31	cv 22-23	22	23	0	0	0
32	cv 23-24	23	24	1	2	17
33	cv 24-25	24	25	4	8	20
34	cv 24-26	24	26	2	3	16
35	cv 24-70	24	70	2	2	0
36	cv 25-26	25	26	0	0	0
37	cv 26-27	26	27	2	4	22
39	cv 27-28	27	28	2	3	12
38	cv 27-29	27	29	1	2	19
40	cv 27-72	27	72	14	14	0
41	cv 28-29	28	29	0	0	0
42	cv 29-30	29	30	2	5	11
43	cv 30-31	30	31	2	3	12
44	cv 30-74	30	74	2	2	0
45	cv 31-32	31	32	1	2	11
46	cv 32-33	32	33	1	2	14
47	cv 33-34	33	34	1	2	5
48	cv 33-35	33	35	1	2	12
49	cv 33-76	33	76	14	14	0
50	cv 34-35	34	35	0	0	0
51	cv 35-36	35	36	1	2	5
52	cv 36-37	36	37	4	8	18
53	cv 36-78	36	78	2	2	0
54	cv 37-38	37	38	3	7	18
55	cv 38-39	38	39	2	5	20
56	cv 39-40	39	40	2	3	12
57	cv 39-80	39	80	14	14	0
58	cv 40-41	40	41	1	2	18
59	cv 41-42	41	42	2	3	11
60	cv 42-43	42	43	4	7	18
61	cv 42-82	42	82	2	2	0

62	cv 43-44	43	44	3	7	16
63	cv 44-45	44	45	2	5	20
64	cv 45-46	45	46	2	3	11
65	cv 45-84	45	84	14	14	0
66	cv 46-47	46	47	1	2	18
67	cv 47-48	47	48	1	2	19
68	cv 48-49	48	49	4	7	18
69	cv 48-86	48	86	2	2	0
70	cv 49-50	49	50	3	7	16
71	cv 50-51	50	51	2	4	20
72	cv 51-52	51	52	2	3	11
73	cv 51-88	51	88	14	14	0
74	cv 52-53	52	53	1	2	18
75	cv 53-54	53	54	1	2	18
76	cv 54-55	54	55	4	7	18
77	cv 54-90	54	90	2	2	0
78	cv 55-56	55	56	4	7	20
79	cv 56-57	56	57	4	7	20
80	cv 57-58	57	58	2	3	12
81	cv 57-92	57	92	14	14	0
82	cv 58-59	58	59	2	3	12
83	cv 59-60	59	60	1	2	18
84	cv 60-61	60	61	1	2	4
85	cv 60-94	60	94	2	2	0
86	cv 61-62	61	62	3	7	15
87	cv 61-63	61	63	2	3	15
88	cv 62-63	62	63	0	0	0
89	cv 63-64	63	64	2	3	16
90	cv 64-65	64	65	1	2	6
91	cv 64-66	64	66	1	2	9
92	cv 64-96	64	96	14	14	0
93	cv 65-66	65	66	0	0	0
94	cv 66-67	66	67	1	1	6
95	cv 67-98	67	98	2	2	0
96	cv 68-69	68	69	2	3	14
97	cv 69-73	69	73	0	0	0
98	cv 70-71	70	71	1	2	10
99	cv 71-73	71	73	0	0	0
100	cv 72-73	72	73	4	7	20
101	cv 73-100	73	100	30	35	30
102	cv 73-172	73	172	1	2	20
103	cv 73-197	73	197	1	2	8
104	cv 74-75	74	75	2	3	12
105	cv 75-77	75	77	0	0	0
106	cv 76-77	76	77	2	3	12
107	cv 77-101	77	101	19	25	20
108	cv 78-79	78	79	1	2	5
109	cv 79-81	79	81	0	0	0

110	cv 80-81	80	81	4	7	20
111	cv 81-102	81	102	1	2	23
112	cv 81-173	81	173	1	2	17
113	cv 81-198	81	198	1	2	8
114	cv 82-83	82	83	2	3	12
115	cv 83-85	83	85	0	0	0
116	cv 84-85	84	85	4	7	20
117	cv 85-103	85	103	18	25	22
118	cv 85-174	85	174	1	2	16
119	cv 85-199	85	199	1	2	8
120	cv 86-87	86	87	2	3	11
121	cv 87-89	87	89	0	0	0
122	cv 88-89	88	89	4	7	20
123	cv 89-104	89	104	20	27	20
124	cv 89-175	89	175	1	2	17
125	cv 89-200	89	200	1	2	8
126	cv 90-91	90	91	2	3	11
127	cv 91-93	91	93	0	0	0
128	cv 92-93	92	93	4	7	20
129	cv 93-105	93	105	20	25	20
130	cv 93-176	93	176	1	2	18
131	cv 93-201	93	201	1	2	8
132	cv 94-95	94	95	2	3	12
133	cv 95-97	95	97	0	0	0
134	cv 96-97	96	97	2	5	20
135	cv 97-106	97	106	15	20	25
136	cv 98-99	98	99	1	2	6
137	cv 99-228	99	228	0	0	0
138	cv 100-107	100	107	3	3	0
139	cv 100-128	100	128	2	5	15
140	cv 100-165	100	165	1	2	11
141	cv 101-109	101	109	3	3	0
142	cv 101-129	101	129	2	5	16
143	cv 101-166	101	166	2	3	18
144	cv 102-111	102	111	3	3	0
145	cv 102-130	102	130	1	2	12
146	cv 102-167	102	167	1	2	16
147	cv 103-113	103	113	3	3	0
148	cv 103-131	103	131	2	5	16
149	cv 103-168	103	168	2	3	17
150	cv 104-115	104	115	3	3	0
151	cv 104-132	104	132	2	5	16
152	cv 104-169	104	169	2	3	18
153	cv 105-117	105	117	3	3	0
154	cv 105-133	105	133	2	5	16
155	cv 105-170	105	170	2	3	20
156	cv 106-119	106	119	3	3	0
157	cv 106-134	106	134	1	2	16

158	cv 106-171	106	171	2	3	11
159	cv 107-108	107	108	4	7	20
160	cv 108-135	108	135	4	4	0
161	cv 108-156	108	156	1	2	19
162	cv 109-110	109	110	3	5	18
163	cv 110-137	110	137	4	4	0
164	cv 110-158	110	158	2	5	20
165	cv 111-112	111	112	1	2	4
166	cv 112-139	112	139	4	4	0
167	cv 112-159	112	159	1	2	15
168	cv 113-114	113	114	3	6	16
169	cv 114-141	114	141	4	4	0
170	cv 114-160	114	160	3	5	21
171	cv 115-116	115	116	3	5	17
172	cv 116-143	116	143	4	4	0
173	cv 116-161	116	161	3	5	20
174	cv 117-118	117	118	3	6	15
175	cv 118-145	118	145	4	4	0
176	cv 118-162	118	162	3	5	20
177	cv 119-120	119	120	3	5	15
178	cv 119-121	119	121	3	5	15
179	cv 120-147	120	147	4	4	0
180	cv 120-163	120	163	3	5	15
181	cv 120-164	120	164	1	2	11
182	cv 121-122	121	122	3	7	15
183	cv 121-149	121	149	4	4	0
184	cv 122-123	122	123	3	6	17
185	cv 122-150	122	150	0	0	0
186	cv 123-124	123	124	3	6	16
187	cv 123-151	123	151	0	0	0
188	cv 124-125	124	125	1	2	4
189	cv 124-152	124	152	0	0	0
190	cv 125-126	125	126	3	5	18
191	cv 125-153	125	153	0	0	0
192	cv 126-127	126	127	4	8	20
193	cv 126-154	126	154	0	0	0
194	cv 127-155	127	155	0	0	0
195	cv 128-222	128	222	0	0	0
196	cv 129-223	129	223	0	0	0
197	cv 130-224	130	224	0	0	0
198	cv 131-225	131	225	0	0	0
199	cv 132-226	132	226	0	0	0
200	cv 133-227	133	227	0	0	0
201	cv 134-228	134	228	0	0	0
202	cv 135-136	135	136	3	6	16
203	cv 136-222	136	222	0	0	0
204	cv 137-138	137	138	1	3	15
205	cv 138-223	138	223	0	0	0

206	cv 139-140	139	140	1	1	4
207	cv 140-224	140	224	0	0	0
208	cv 141-142	141	142	2	3	15
209	cv 142-225	142	225	0	0	0
210	cv 143-144	143	144	2	3	16
211	cv 144-226	144	226	0	0	0
212	cv 145-146	145	146	2	3	14
213	cv 146-227	146	227	0	0	0
214	cv 147-148	147	148	2	3	14
215	cv 148-228	148	228	0	0	0
216	cv 149-150	149	150	2	3	14
217	cv 150-151	150	151	2	3	14
218	cv 151-152	151	152	2	3	16
219	cv 152-153	152	153	2	3	15
220	cv 153-154	153	154	1	1	4
221	cv 154-155	154	155	2	3	15
222	cv 155-228	155	228	3	6	16
223	cv 156-157	156	157	3	6	20
224	cv 157-222	157	222	0	0	0
225	cv 158-223	158	223	0	0	0
226	cv 159-224	159	224	0	0	0
227	cv 160-225	160	225	0	0	0
228	cv 161-226	161	226	0	0	0
229	cv 162-227	162	227	0	0	0
230	cv 163-228	163	228	0	0	0
231	cv 164-228	164	228	0	0	0
232	cv 165-222	165	222	0	0	0
233	cv 166-223	166	223	0	0	0
234	cv 167-224	167	224	0	0	0
235	cv 168-225	168	225	0	0	0
236	cv 169-226	169	226	0	0	0
237	cv 170-227	170	227	0	0	0
238	cv 171-228	171	228	0	0	0
239	cv 172-177	172	177	1	2	12
240	cv 173-178	173	178	1	2	19
241	cv 174-179	174	179	1	2	11
242	cv 175-180	175	180	1	2	19
243	cv 176-181	176	181	1	2	12
244	cv 177-182	177	182	1	2	12
245	cv 178-183	178	183	1	2	10
246	cv 179-184	179	184	1	2	10
247	cv 180-185	180	185	1	2	10
248	cv 181-186	181	186	1	2	10
249	cv 182-187	182	187	14	14	0
250	cv 183-189	183	189	14	14	0
251	cv 184-191	184	191	14	14	0
252	cv 185-193	185	193	14	14	0
253	cv 186-195	186	195	14	14	0

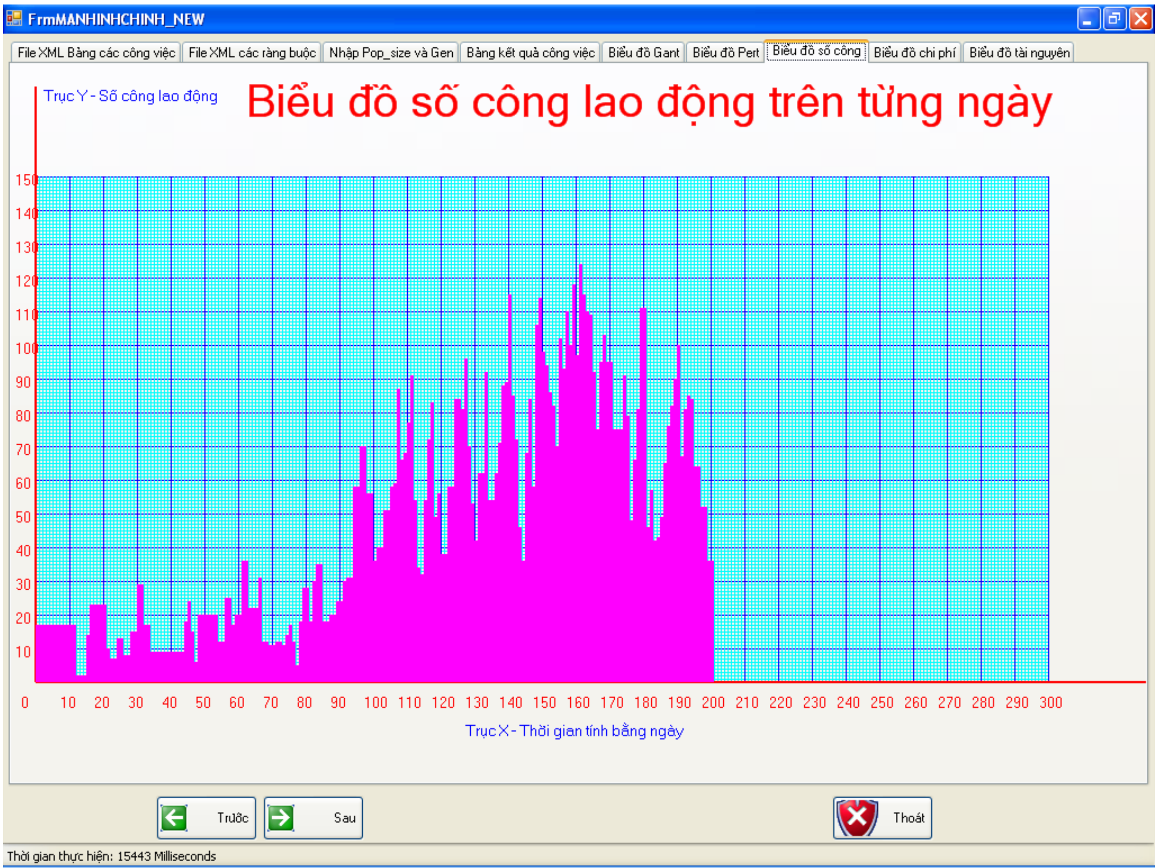
254	cv 187-188	187	188	1	2	20
255	cv 188-222	188	222	0	0	0
256	cv 189-190	189	190	1	2	17
257	cv 190-224	190	224	0	0	0
258	cv 191-192	191	192	1	2	16
259	cv 192-225	192	225	0	0	0
260	cv 193-194	193	194	1	2	17
261	cv 194-226	194	226	0	0	0
262	cv 195-196	195	196	1	2	18
263	cv 196-227	196	227	0	0	0
264	cv 197-202	197	202	1	1	6
265	cv 198-203	198	203	1	2	11
266	cv 199-204	199	204	1	2	11
267	cv 200-205	200	205	1	2	11
268	cv 201-206	201	206	1	2	11
269	cv 202-207	202	207	1	2	22
270	cv 203-208	203	208	2	3	13
271	cv 204-209	204	209	2	3	13
272	cv 205-210	205	210	2	3	13
273	cv 206-211	206	211	2	3	13
274	cv 207-212	207	212	14	14	0
275	cv 208-214	208	214	14	14	0
276	cv 209-216	209	216	14	14	0
277	cv 210-218	210	218	14	14	0
278	cv 211-220	211	220	14	14	0
279	cv 212-213	212	213	1	2	8
280	cv 213-222	213	222	0	0	0
281	cv 214-215	214	215	1	2	8
282	cv 215-224	215	224	0	0	0
283	cv 216-217	216	217	1	2	8
284	cv 217-225	217	225	0	0	0
285	cv 218-219	218	219	1	2	8
286	cv 219-226	219	226	0	0	0
287	cv 220-221	220	221	1	2	8
288	cv 221-227	221	227	0	0	0
289	cv 222-228	222	228	0	0	0
290	cv 223-228	223	228	0	0	0
291	cv 224-228	224	228	0	0	0
292	cv 225-228	225	228	0	0	0
293	cv 226-227	226	227	0	0	0
294	cv 227-228	227	228	0	0	0

4.3.3 Giải quyết bài toán:

Ta giải quyết như bài toán Cầu Sơn Mãn.

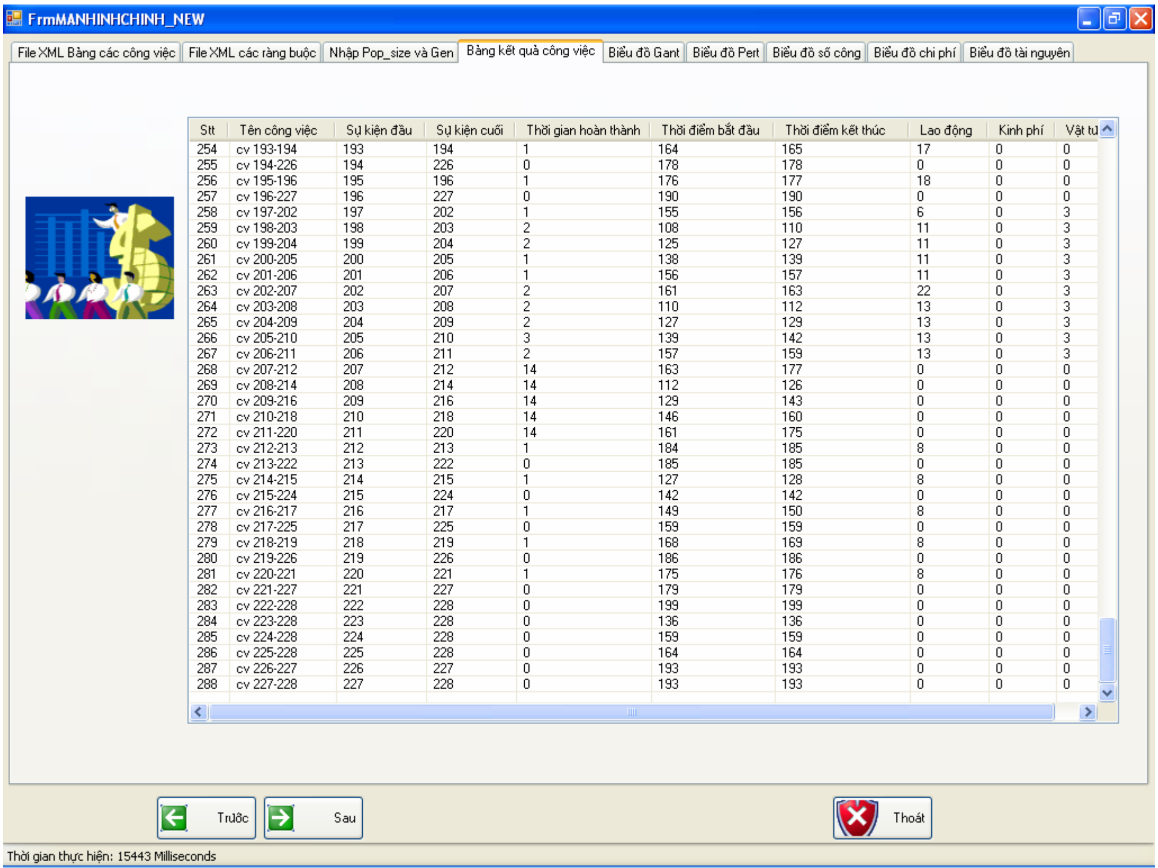
4.3.4 Kết quả đạt được:

- Màn hình kết quả số công lao động tính trên từng ngày.



Hình 4.11 – Màn hình kết quả số công lao động tính trên từng ngày.

- Màn hình bảng kết quả các công việc.



Stt	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn thành	Thời điểm bắt đầu	Thời điểm kết thúc	Lao động	Kinh phí	Vật tư
254	cv 193-194	193	194	1	164	165	17	0	0
255	cv 194-226	194	226	0	178	178	0	0	0
256	cv 195-196	195	196	1	176	177	18	0	0
257	cv 196-227	196	227	0	190	190	0	0	0
258	cv 197-202	197	202	1	155	156	6	0	3
259	cv 198-203	198	203	2	108	110	11	0	3
260	cv 199-204	199	204	2	125	127	11	0	3
261	cv 200-205	200	205	1	138	139	11	0	3
262	cv 201-206	201	206	1	156	157	11	0	3
263	cv 202-207	202	207	2	161	163	22	0	3
264	cv 203-208	203	208	2	110	112	13	0	3
265	cv 204-209	204	209	2	127	129	13	0	3
266	cv 205-210	205	210	3	139	142	13	0	3
267	cv 206-211	206	211	2	157	159	13	0	3
268	cv 207-212	207	212	14	163	177	0	0	0
269	cv 208-214	208	214	14	112	126	0	0	0
270	cv 209-216	209	216	14	129	143	0	0	0
271	cv 210-218	210	218	14	146	160	0	0	0
272	cv 211-220	211	220	14	161	175	0	0	0
273	cv 212-213	212	213	1	184	185	8	0	0
274	cv 213-222	213	222	0	185	185	0	0	0
275	cv 214-215	214	215	1	127	128	8	0	0
276	cv 215-224	215	224	0	142	142	0	0	0
277	cv 216-217	216	217	1	149	150	8	0	0
278	cv 217-225	217	225	0	159	159	0	0	0
279	cv 218-219	218	219	1	168	169	8	0	0
280	cv 219-226	219	226	0	186	186	0	0	0
281	cv 220-221	220	221	1	175	176	8	0	0
282	cv 221-227	221	227	0	179	179	0	0	0
283	cv 222-228	222	228	0	199	199	0	0	0
284	cv 223-228	223	228	0	136	136	0	0	0
285	cv 224-228	224	228	0	159	159	0	0	0
286	cv 225-228	225	228	0	164	164	0	0	0
287	cv 226-227	226	227	0	193	193	0	0	0
288	cv 227-228	227	228	0	193	193	0	0	0

Hình 4.12 – Màn hình kết quả các công việc.

4.4 Hướng phát triển:

- Dùng cho việc tính toán sơ bộ chi phí trong việc tham gia đấu thầu dự án.
- Dùng để ước tính công trình này thực hiện trong khoảng thời gian bao lâu là hợp lý trong việc thanh tra xây dựng.
- Dùng để làm đầu vào cho Microsoft Project.
- Ứng dụng có thể phát triển cho các bài toán lập lịch biểu: nhất là bài toán sắp xếp thời khóa biểu cho các Phòng Giáo Vụ hoặc Đào Tạo khi phải vừa thỏa cho từng Lớp, cho từng Giảng Viên và đồng thời phù hợp tất cả các phòng học (đa mục tiêu).
- Ứng dụng cho việc thiết kế cơ khí, máy móc với đa mục tiêu đã được trình bày ở chương 3.
- Hy vọng phát triển thêm phần đa kết quả, trong đó kết hợp tính toán song song hoặc tính toán lưới.

Tài liệu tham khảo

Tiếng Việt

[1] TRƯƠNG MỸ DUNG (1994), "Toán Kỹ thuật lượng tính trong Quản lý", *Giáo trình ĐH Kinh tế, Tp.HCM*.

[2] Hoàng Kiếm và Lê Hoàng Thái (2000), *Thuật giải di truyền Cách giải tự nhiên Các bài toán trên máy tính*, NXB Giáo Dục.

[3] Phạm Hoàng Vương (2000), *Ứng dụng thuật giải di truyền giải một lớp bài toán tối ưu*, Luận án thạc sĩ khoa học, Trường Đại Học Khoa Học Tự Nhiên TP. Hồ Chí Minh.

[4] Nguyễn Đình Thúc (2001), *Trí tuệ nhân tạo Lập trình tiến hóa Cấu trúc dữ liệu + Thuật giải di truyền = Chương trình tiến hóa*, NXB Giáo dục.

[5] Hoàng Kiếm (2001), *Giải một bài toán trên máy tính như thế nào Tập hai*, NXB Giáo Dục.

[6] Nguyễn Thanh Tùng (2004), *Áp dụng thuật giải di truyền giải quyết bài toán "Lập tiến độ thi công trong xây dựng"*, Đồ án tốt nghiệp chuyên ngành Công Nghệ Tri Thức, Đại Học Khoa Học Tự Nhiên TP. HCM.

[7] Trần Văn Vinh (2002), *Phát triển một số thuật toán giải bài toán tối ưu đa mục tiêu và ứng dụng*, Luận án thạc sĩ khoa học CNTT, Trường Đại Học Khoa Học Tự Nhiên TP. HCM.

[8] Lê Hoàng Thái (2004), *Xây dựng, phát triển, ứng dụng một số mô hình kết hợp giữa Mạng Nơron, Logic Mờ và Thuật Giải Di Truyền*, Luận án tiến sĩ toán học.

[9] Nguyễn Thanh Trung (2006), *Ứng dụng giải thuật di truyền đa mục tiêu trong việc thiết kế chi tiết máy*, Luận án thạc sĩ khoa học CNTT.

Tiếng Anh

[10] Ivo F. Sbalzariniy, Sibylle M̊ullery and Petros Koumoutsakosyz (2000), *Multiobjective optimization using evolutionary algorithms*, Center for Turbulence Research Proceedings of the Summer Program.

[11] Carlos A. Coello Coello, Ricardo L.B. (2002), *Evolutionary Multiobjective Optimization using a Cultural Algorithm*, Department of Computer Science, Tulane University.

[12] Carlos A. Coello Coello, Gregorio T. P. (2001), *Multiobjective optimization using a Micro-genetic algorithm*, Proceedings of the genetic and evolutionary computation conference (GECCO 2001), pp 274-282.

[13] Eckart Z., Marco L., Stefan B. (2002), SPEA2: *Improving the Strength Pareto Evolutionary Algorithm for Multi-Objective Optimization*, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

[14] Johan A. (2003), *Aplication of a Multi-Object Genetic Algorithm to Engineering Design Problems*, Department of Mechanical Engineering. Linkoping University.

[15] Ossyczka Y., Mrzyglod M. (2005), *Evolutionary Optimization of Mechanical Structures in Computer Simulated Environment*, 6th World Congresses of Structural and Multidisciplinary Optimization. Brasil.

Phụ lục A: Ứng dụng XML trong Visual studio 2005 để lưu trữ dữ liệu

Chương I: Giới thiệu sơ lược về XML

I.Khái niệm XML:

- Extensible Markup Language.
- Là ngôn ngữ hình thức thuộc dạng đánh dấu với:
  - Bộ từ vựng và các từ: Các thẻ đánh dấu.
  - Bộ cú pháp: Qui tắc kết hợp các thẻ đánh dấu
- Tuân theo định chuẩn chung: Định chuẩn XML (W3C)
  - Bởi World Wide Web Consortium (W3C)
  - http://www.w3.org/TR/2000/REC-xml-20001006
  - http://www.w3.org/TR/REC-xml-names/

II.Khái niệm tài liệu XML:

- Dạng biểu diễn thông tin được đề xuất của Công nghệ XML.
- Văn bản có cấu trúc, được tạo lập dựa trên một ngôn ngữ XML nào đó.

III.XML và vấn đề lưu trữ thông tin:

Có 3 cách ứng dụng chính của XML để lưu trữ dữ liệu bên trong hệ thống tin học:

1. Chỉ sử dụng các tập tin XML để lưu trữ dữ liệu.
2. Một số dữ liệu lưu trữ dưới dạng tập tin XML, một số khác lưu trữ bên trong cơ sở dữ liệu.
3. Lưu trữ toàn bộ bên trong cơ sở dữ liệu, tài liệu XML khi đó được nhúng vào nội dung các bảng dữ liệu.

Cách 1:

- *Ưu điểm:*
  - Không cần sự hổ trợ của các hệ quản trị cơ sở dữ liệu, do đó dễ cài đặt và triển khai phần mềm.
- *Khuyết điểm:*
  - Tính hiệu quả không cao khi khối lượng dữ liệu lớn.
- *Nhận xét:*
  - Các phần mềm trò chơi, các phần mềm với khối lượng dữ liệu nhỏ và vừa rất thích hợp với XML theo cách 1 này.
  - Các phần mềm về quản lý không thích hợp cho cách 1 này.
  - Rất thích hợp cho các ứng dụng trên môi trường tin học không có (hoặc chưa có) hệ quản trị cơ sở dữ liệu như: điện thoại di động, máy công cụ,..

Cách 2, 3:

- *Ưu điểm:* Có thể kết hợp tốt ưu điểm của cả 2 mô hình lưu trữ thông tin: XML, cơ sở dữ liệu.
- *Khuyết điểm:*
  - Cần có sự cân nhắc và quyết định đúng đắn loại thông tin nào sẽ dùng hình thức lưu trữ nào.
- *Nhận xét:*
  - Cách 2 là cách phổ biến nhất hiện nay.
  - Cấu hình của hệ thống tin học (phân hệ ứng dụng) là loại thông tin thường được chọn để lưu trữ theo dạng tài liệu XML.

IV.Ví dụ 1 trang XML dạng Text:

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	
<booklist>	
<book ***genre="Science" format="Hardcover"***>	
<author>	phần tử (element)
<firstname>**Richard**</firstname>	
<lastname>**Feynman**</lastname>	
</author>	
<title>**The character of Physical Law**</title>	
</book>	
<book ***genre="Fiction"***>	
<author>	phần tử (element)
<firstname>**R.K.**</firstname>	
<lastname>**Narayan**</lastname>	
</author>	
<title>**Waiting for the Mahatma**</title>	
<published>**1981**</published>	
</book>	
</booklist>	

Biểu diễn thuộc tính:

genre="Science" ***format="Hardcover"***	***Tên_thuộc_tính = 'Giá trị'***
genre="Fiction"	

Biểu diễn một thẻ và nội dung:

<table>
<tr><td><firstname>R.K.</firstname></td><td rowspan="4"><Tên thẻ> Nội dung </Tên thẻ></td></tr>
<tr><td><lastname>Narayan</lastname></td></tr>
<tr><td><title>Waiting for the Mahatma</title></td></tr>
<tr><td><published>1981</published></td></tr>
</table>

V. Tài liệu XML có 3 thành phần:

- Thẻ khai báo (XML Processing Instruction)
- Khai báo loại tài liệu (Document Type Declaration)
- Thể hiện tài liệu (Document Instance)

1. Thẻ khai báo (XML Processing Instruction)

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

- Thông tin version của định chuẩn XML.
- Loại mã hóa các ký tự trong tài liệu :
  - UTF-8: Unicode utf-8.
  - UTF-16: Unicode utf-16
- Standalone:
  - "yes": Liên kết với tài liệu khác.
  - "no": Không liên kết với tài liệu khác.

2. Có 2 loại khai báo tài liệu

- Loại khai báo tài liệu Internal:

```
<!DOCTYPE CustomerOrder [
 <!-- internal DTD goes here! -->
]>
```

(DTD: Document Type Definition)

- Tham chiếu External:

```
<!DOCTYPE CustomerOrder SYSTEM
"http://www.myco.com/CustOrder.dtd">
```

3. Thể hiện tài liệu (Document Instance: The Markup)

- Phần tử Gốc (Root Element)
  - Bắt buộc phải có trong mỗi tài liệu XML.
  - Phải có tên giống trong phần khai báo loại tài liệu.
- Phần tử con (Elements)
  - Có thể chứa các phần tử con khác.
  - Có thể có các thuộc tính của chúng.
  - Có hoặc không có giá trị.
- Các thuộc tính (Attributes)
  - Được gán cho các phần tử.
  - Cung cấp thêm thông tin cho phần tử.

4. Việc đánh dấu thẻ XML (Tag-Sets)

- Bắt đầu với <*someTag*> và kết với </*someTag*>
  - Các thẻ rỗng có dạng: <*someTag* />
- Một số ngoại lệ (Exceptions):
  - Thẻ khai báo: <?xml ... ?>
  - Lời chú thích: <!-- *some comment* -->

- Khai báo loại tài liệu:
    - <! DOCTYPE [...]>

5. Ví dụ: một tài liệu XML tham chiếu external

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<! DOCTYPE CustomerOrder
SYSTEM "http://www.myco.com/dtd/order.dtd" >
<CustomerOrder>
        <Customer>
                <Person>
                        <FName> Olaf </FName>
                        <LName> Smith </LName>
                </Person>
                <Address AddrType = "shipping">
                        91 Park So, New York, NY 10018 </Address>
                <Address AddrType = "billing">
                        Hauptstrasse 55, D-81671 Munich        </Address>
        </Customer>
        <Orders>
                <OrderNo> 10 </OrderNo>
                <ProductNo> 100 </ProductNo>
                <ProductNo> 200 </ProductNo>
        </Orders>
        <!-- More <Customer>s ... -->
</CustomerOrder>
```

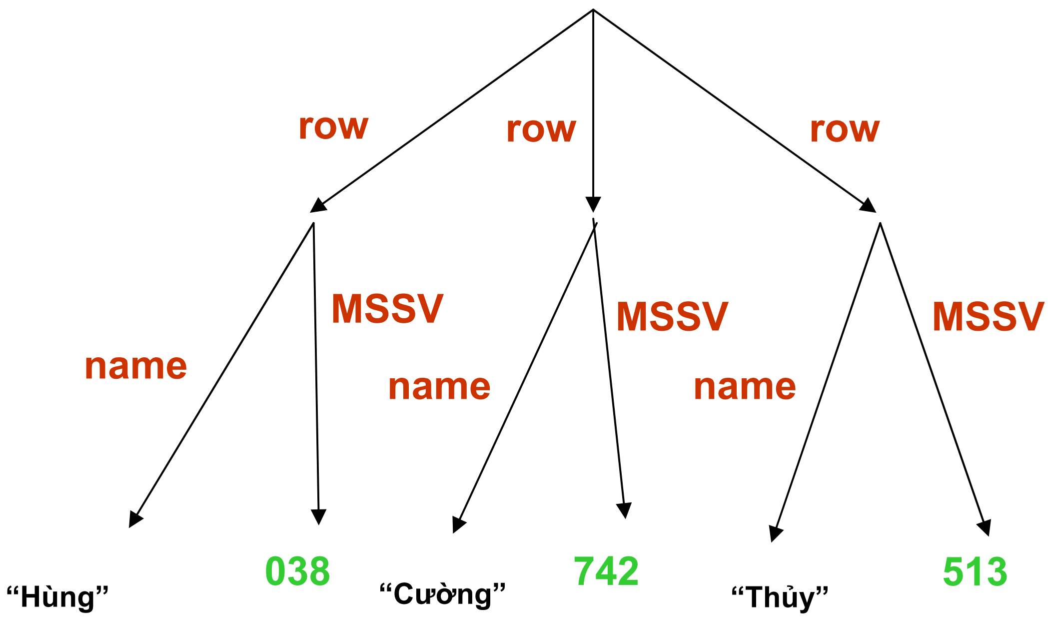
Hình 5.1 Một tài liệu XML từ trang web

"http:/www.myco.com/dtd/order.dtd"

VI. XML và Relational Data

Name	Mã số sinh viên
Hùng	038
Cường	742
Thủy	513

Relation



XML

Chương II: Ứng dụng XML vào Bài toán lập tiến độ thi công trong xây dựng

I.Giới thiệu về dữ liệu của "Bài toán lập tiến độ thi công trong xây dựng"

Cho bài toán ví dụ sau: gồm 10 công việc, thời gian được tính bằng ngày, số lao động, kinh phí, vật tư trên một ngày.

Ta có bảng công việc sau:

STT	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn tất tối thiểu	Thời gian hoàn tất tối đa	Lao động	Kinh phí	Vật tư
1	Cv 0-1	0	1	10	22	10	2	1
2	Cv 0-2	0	2	9	25	12	4	2
3	Cv 0-4	0	4	5	15	25	7	4
4	Cv 1-4	1	4	0	0	0	0	0
5	Cv 2-3	2	3	5	17	30	9	0
6	Cv 2-6	2	6	6	19	26	7	3
7	Cv 1-5	1	5	10	20	15	6	4
8	Cv 4-5	4	5	8	22	10	5	3
9	Cv 3-6	3	6	7	19	7	4	2
10	Cv 5-6	5	6	10	25	5	3	1

Ta có bảng ràng buộc như sau:

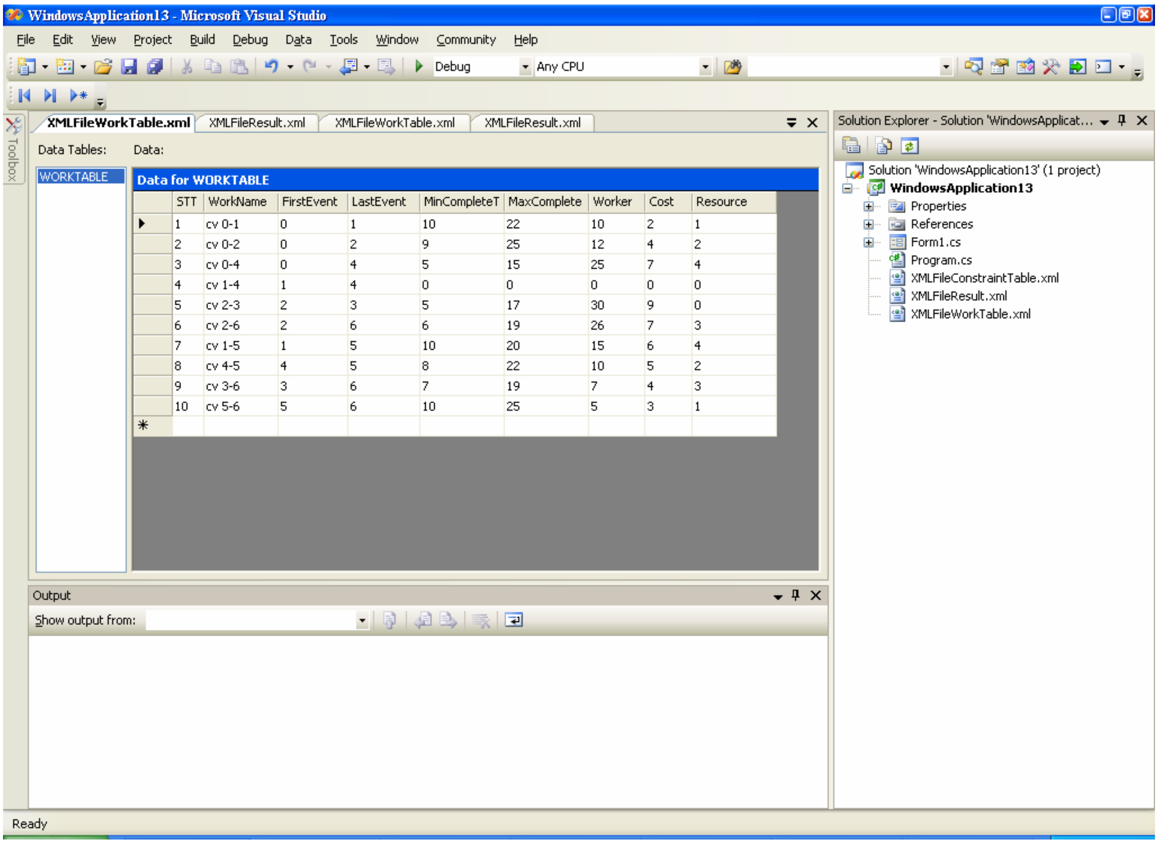
Tên ràng buộc	Giá trị tối thiểu	Giá trị tối đa
Lao động	5	38
Kinh phí	2	17
Vật tư	1	9

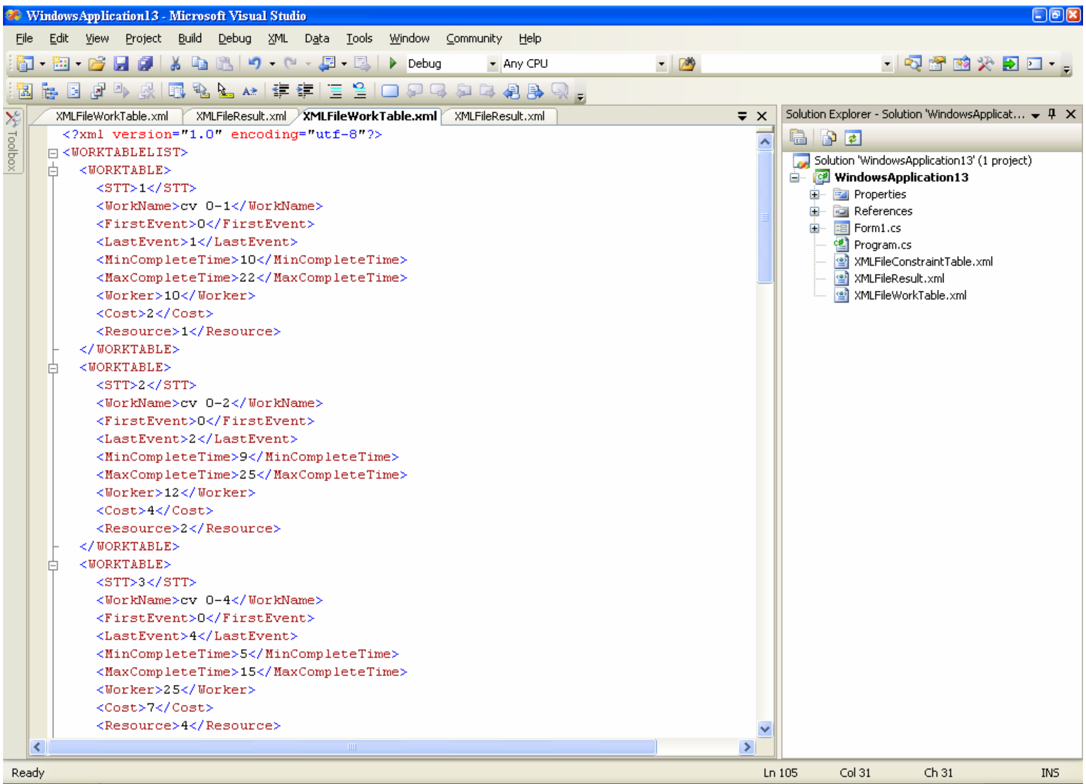
Ta có bảng kết quả công việc sau:

STT	Tên công việc	Sự kiện đầu	Sự kiện cuối	Thời gian hoàn tất	Thời điểm bắt đầu	Thời điểm kết thúc	Lao động	Kinh phí	Vật tư
1	Cv 0-1	0	1	10	0	10	10	2	1
2	Cv 0-2	0	2	9	10	20	12	4	2
3	Cv 0-4	0	4	10	0	10	25	7	4
4	Cv 1-4	1	4	0	10	10	0	0	0
5	Cv 2-3	2	3	10	10	20	30	9	0
6	Cv 2-6	2	6	5	20	25	26	7	3
7	Cv 1-5	1	5	6	26	32	15	6	4
8	Cv 4-5	4	5	7	25	32	10	5	3
9	Cv 3-6	3	6	10	10	20	7	4	2
10	Cv 5-6	5	6	12	20	32	5	3	1

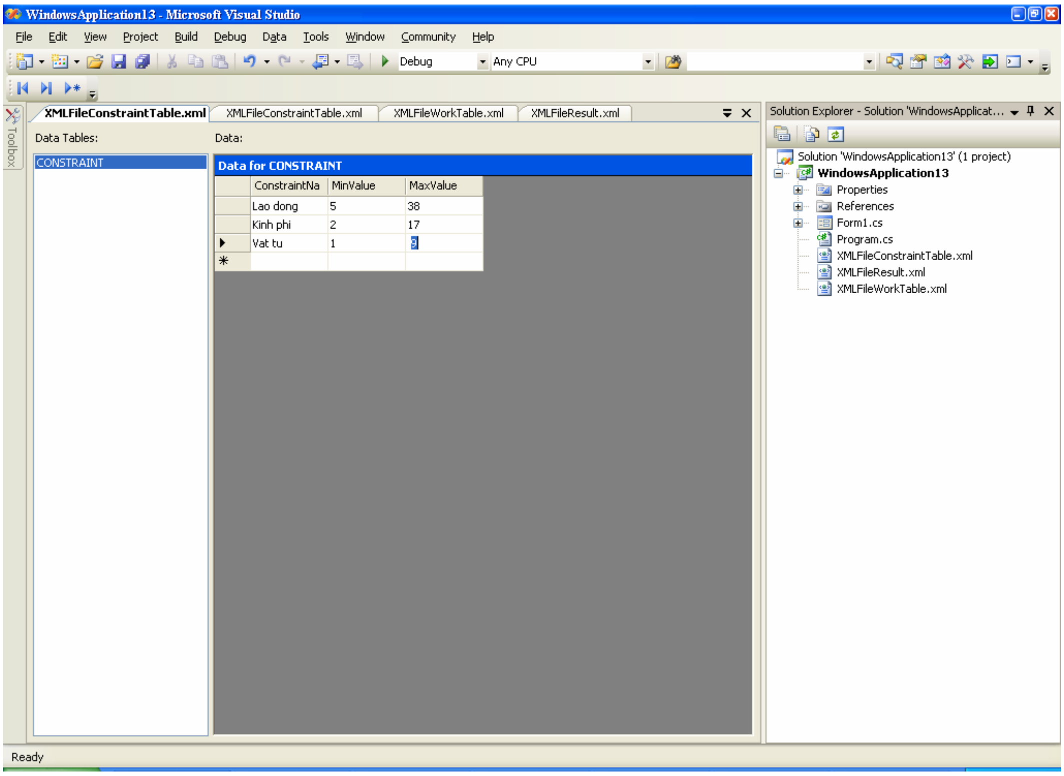
II. Dữ liệu của "Bài toán lập tiến độ thi công trong xây dựng" bằng XML trên C# 2005

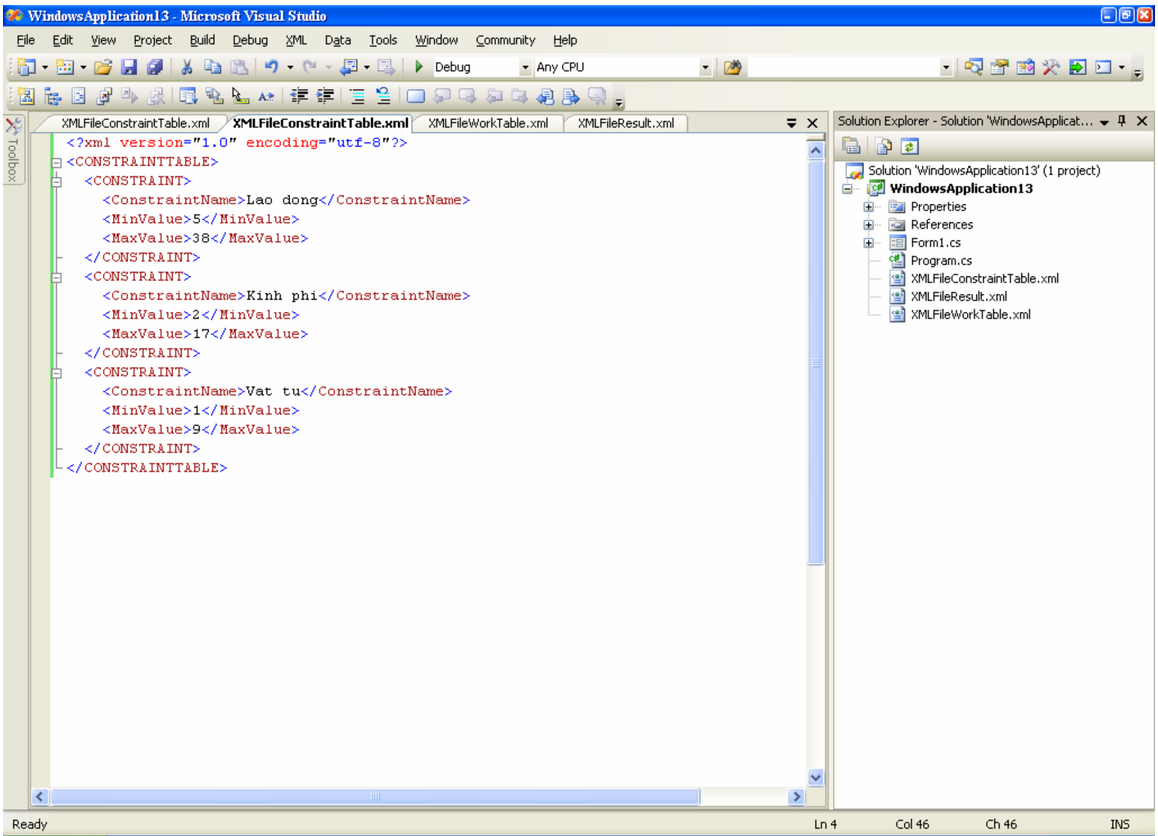
Đối với bảng công việc:


WindowsApplication13 - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Window Community Help
XMLFileWorkTable.xml XMLFileResult.xml XMLFileWorkTable.xml XMLFileResult.xml
Data Tables: Data:
WORKTABLE
Data for WORKTABLE
STT WorkName FirstEvent LastEvent MinCompleteT MaxComplete Worker Cost Resource
1 cv 0-1 0 1 10 22 10 2 1
2 cv 0-2 0 2 9 25 12 4 2
3 cv 0-4 0 4 5 15 25 7 4
4 cv 1-4 1 4 0 0 0 0 0
5 cv 2-3 2 3 5 17 30 9 0
6 cv 2-6 2 6 6 19 26 7 3
7 cv 1-5 1 5 10 20 15 6 4
8 cv 4-5 4 5 8 22 10 5 2
9 cv 3-6 3 6 7 19 7 4 3
10 cv 5-6 5 6 10 25 5 3 1
Solution Explorer - Solution 'WindowsApplicat...
Solution 'WindowsApplication13' (1 project)
WindowsApplication13
Properties
References
Form1.cs
Program.cs
XMLFileConstraintTable.xml
XMLFileResult.xml
XMLFileWorkTable.xml
Output
Show output from:
Ready

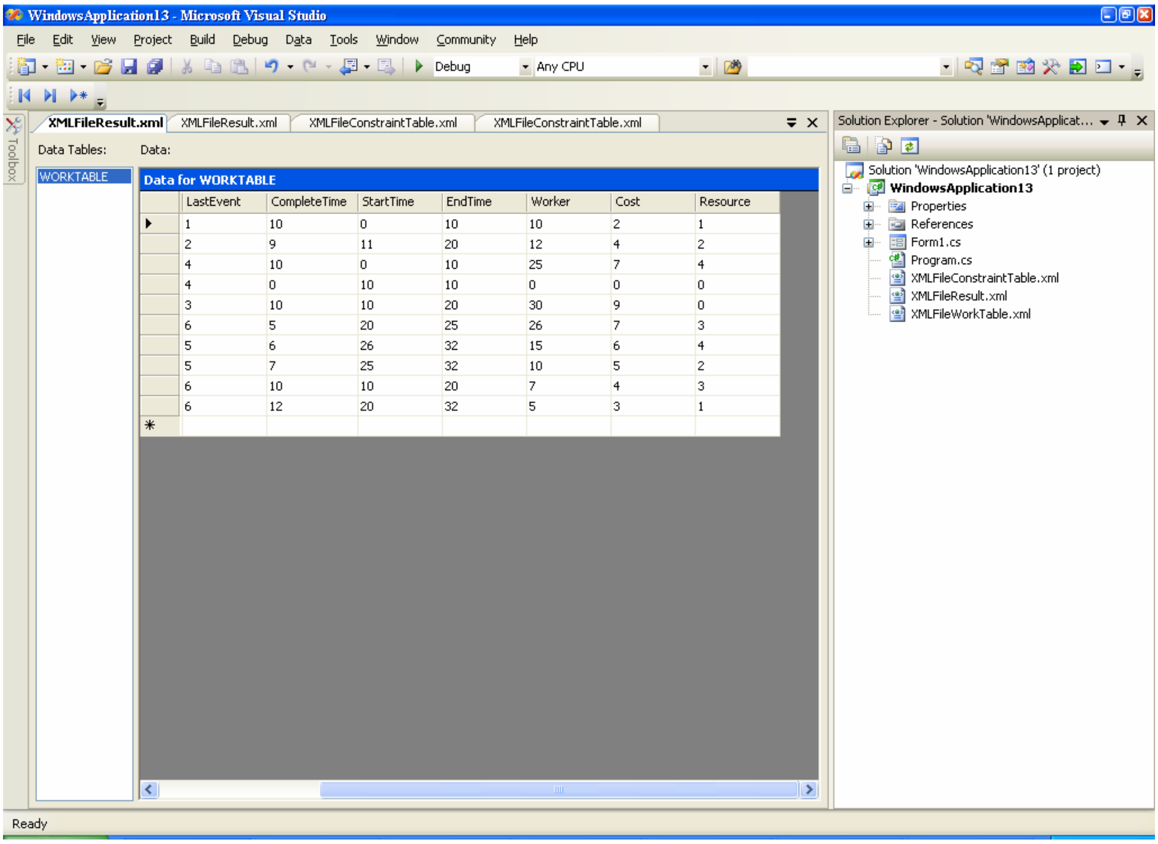

WindowsApplication13 - Microsoft Visual Studio
File Edit View Project Build Debug XML Data Tools Window Community Help
XMLFileWorkTable.xml XMLFileResult.xml XMLFileWorkTable.xml XMLFileResult.xml
<?xml version="1.0" encoding="utf-8"?>
<WORKTABLELIST>
<WORKTABLE>
<STT>1</STT>
<WorkName>cv 0-1</WorkName>
<FirstEvent>0</FirstEvent>
<LastEvent>1</LastEvent>
<MinCompleteTime>10</MinCompleteTime>
<MaxCompleteTime>22</MaxCompleteTime>
<Worker>10</Worker>
<Cost>2</Cost>
<Resource>1</Resource>
</WORKTABLE>
<WORKTABLE>
<STT>2</STT>
<WorkName>cv 0-2</WorkName>
<FirstEvent>0</FirstEvent>
<LastEvent>2</LastEvent>
<MinCompleteTime>9</MinCompleteTime>
<MaxCompleteTime>25</MaxCompleteTime>
<Worker>12</Worker>
<Cost>4</Cost>
<Resource>2</Resource>
</WORKTABLE>
<WORKTABLE>
<STT>3</STT>
<WorkName>cv 0-4</WorkName>
<FirstEvent>0</FirstEvent>
<LastEvent>4</LastEvent>
<MinCompleteTime>5</MinCompleteTime>
<MaxCompleteTime>15</MaxCompleteTime>
<Worker>25</Worker>
<Cost>7</Cost>
<Resource>4</Resource>
Solution Explorer - Solution 'WindowsApplicat...
Solution 'WindowsApplication13' (1 project)
WindowsApplication13
Properties
References
Form1.cs
Program.cs
XMLFileConstraintTable.xml
XMLFileResult.xml
XMLFileWorkTable.xml
Ready
Ln 105 Col 31 Ch 31 INS

Bảng ràng buộc:

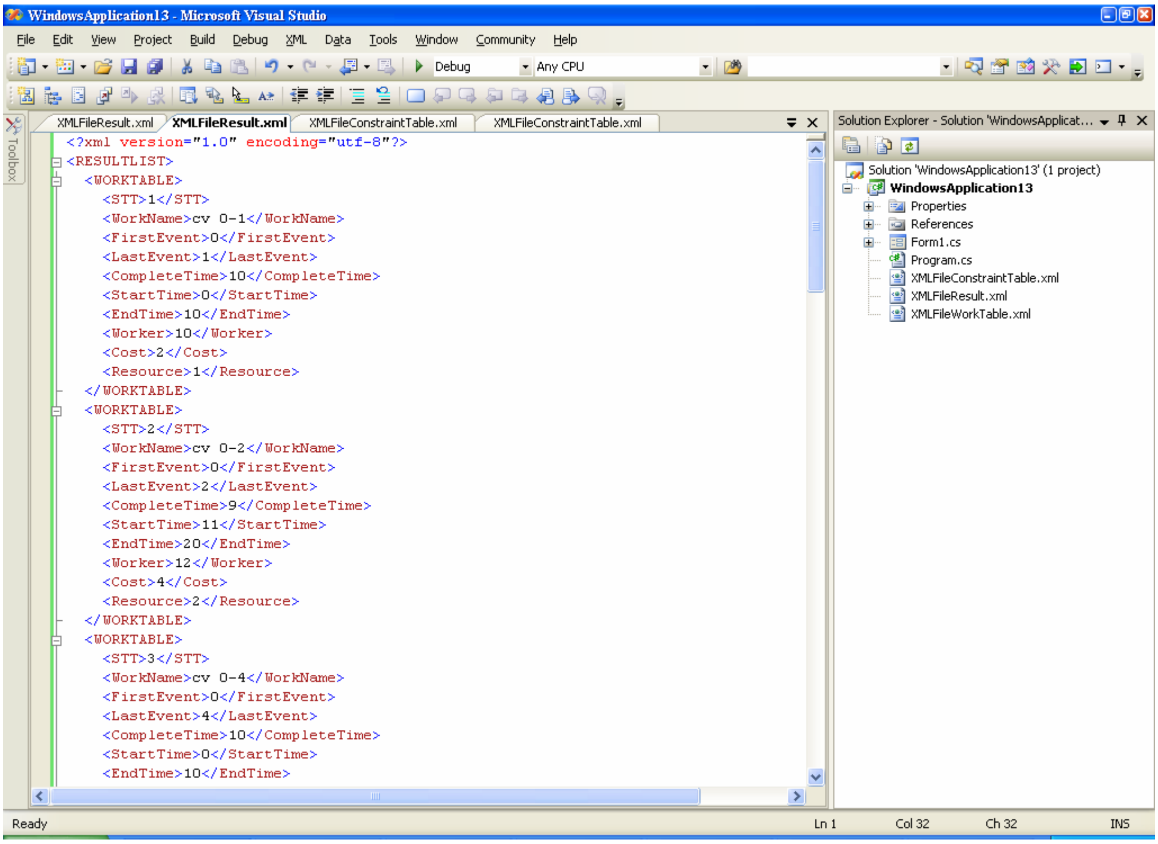




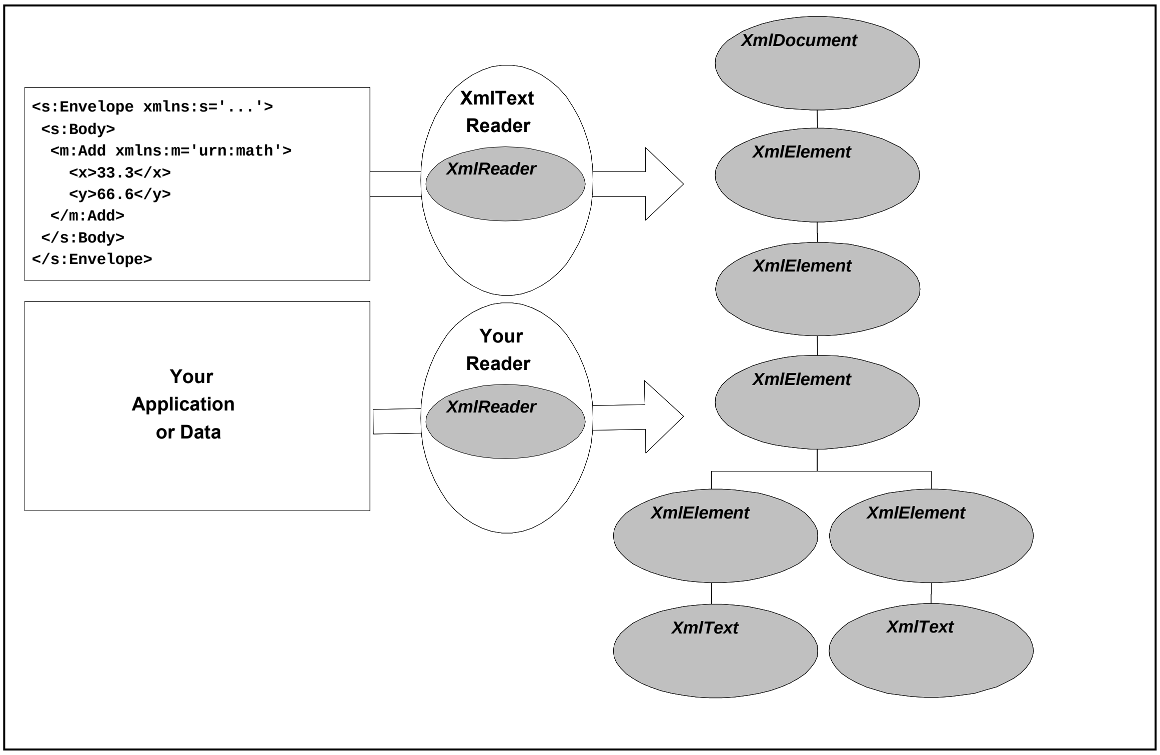
Bảng kết quả công việc:



LastEvent	CompleteTime	StartTime	EndTime	Worker	Cost	Resource
1	10	0	10	10	2	1
2	9	11	20	12	4	2
4	10	0	10	25	7	4
4	0	10	10	0	0	0
3	10	10	20	30	9	0
6	5	20	25	26	7	3
5	6	26	32	15	6	4
5	7	25	32	10	5	2
6	10	10	20	7	4	3
6	12	20	32	5	3	1



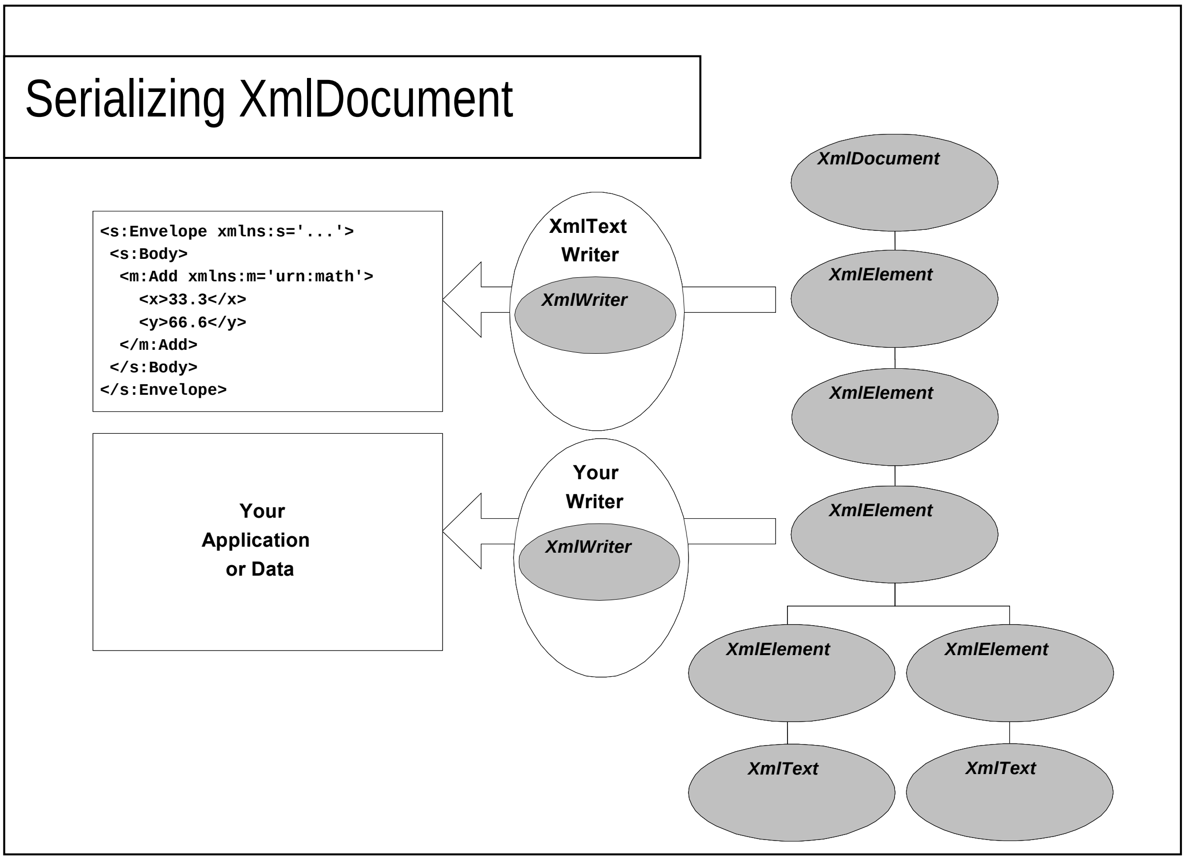
Loading XmlDocument



III. Các hàm xử lý dữ liệu của "Bài toán lập tiến độ thi công trong xây dựng" bằng XML trên C# 2005

1. Hàm đọc bảng công việc từ dữ liệu file *.xml vào bộ nhớ:

```
public bool Doc(string Ten_tap_tin)
{
    bool Kq = true;
    XmlDocument Tai_lieu = new XmlDocument();
    try
    {
        Tai_lieu.Load(Ten_tap_tin);
        XmlElement Goc = Tai_lieu.DocumentElement;
        foreach (XmlElement Nut in Goc.ChildNodes)
            Work_Table.Add(new XL_WORK(Nut));
    }
    catch (Exception Loi)
    {
        Kq = false;
    }
    return Kq;
}
```



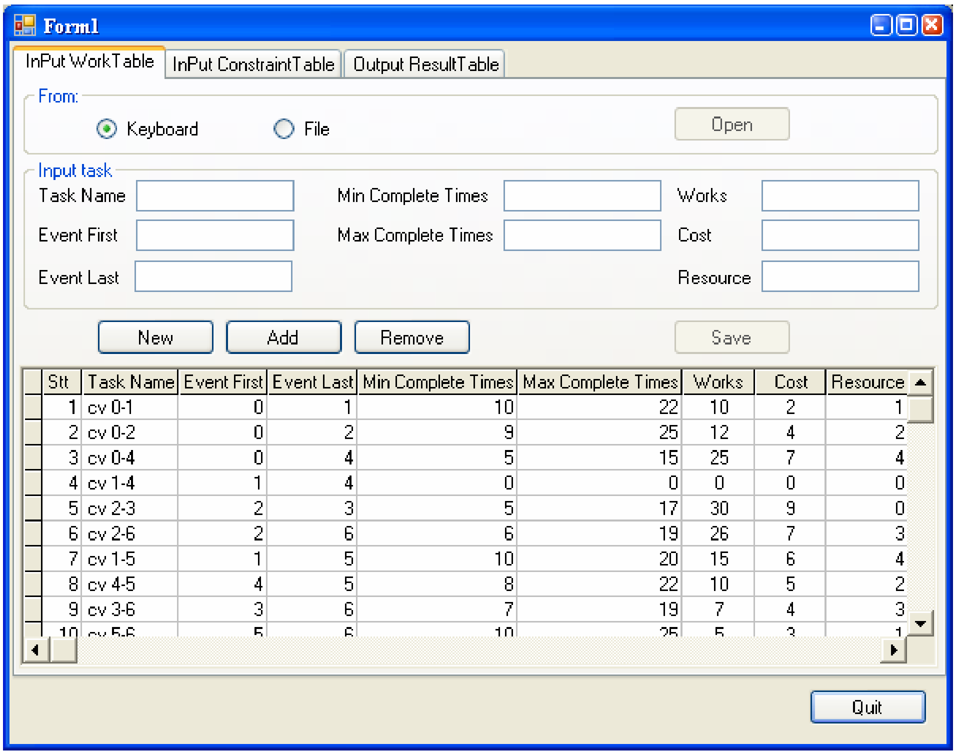
2. Hàm ghi bảng công việc từ bộ nhớ vào file *.xml:

```
public bool Ghi(string Ten_tap_tin)
{    bool Kq = true;     int i = 1;
    XmlDocument Tai_lieu = new XmlDocument();
    try
    {   XmlElement Root = Tai_lieu.CreateElement ("WORKTABLELIST");


        foreach (XL_WORK W in Work_Table)
        {
            XmlElement Nut = W.AppendNode(Tai_lieu, i++);
            Root.AppendChild(Nut);
```

```
        }
        Tai_lieu.AppendChild(Root);
        Tai_lieu.Save(Ten_tap_tin);
    }
    catch (Exception Loi)
    {       Kq = false;
    }
    return Kq;
}
```

<u>Kết quả trên màn hình:</u>



Stt	Task Name	Event First	Event Last	Min Complete Times	Max Complete Times	Works	Cost	Resource
1	cv 0-1	0	1	10	22	10	2	1
2	cv 0-2	0	2	9	25	12	4	2
3	cv 0-4	0	4	5	15	25	7	4
4	cv 1-4	1	4	0	0	0	0	0
5	cv 2-3	2	3	5	17	30	9	0
6	cv 2-6	2	6	6	19	26	7	3
7	cv 1-5	1	5	10	20	15	6	4
8	cv 4-5	4	5	8	22	10	5	2
9	cv 3-6	3	6	7	19	7	4	3
10	cv 5-6	5	6	10	25	5	3	1

3. Hàm đọc bảng ràng buộc từ dữ liệu file *.xml vào bộ nhớ:

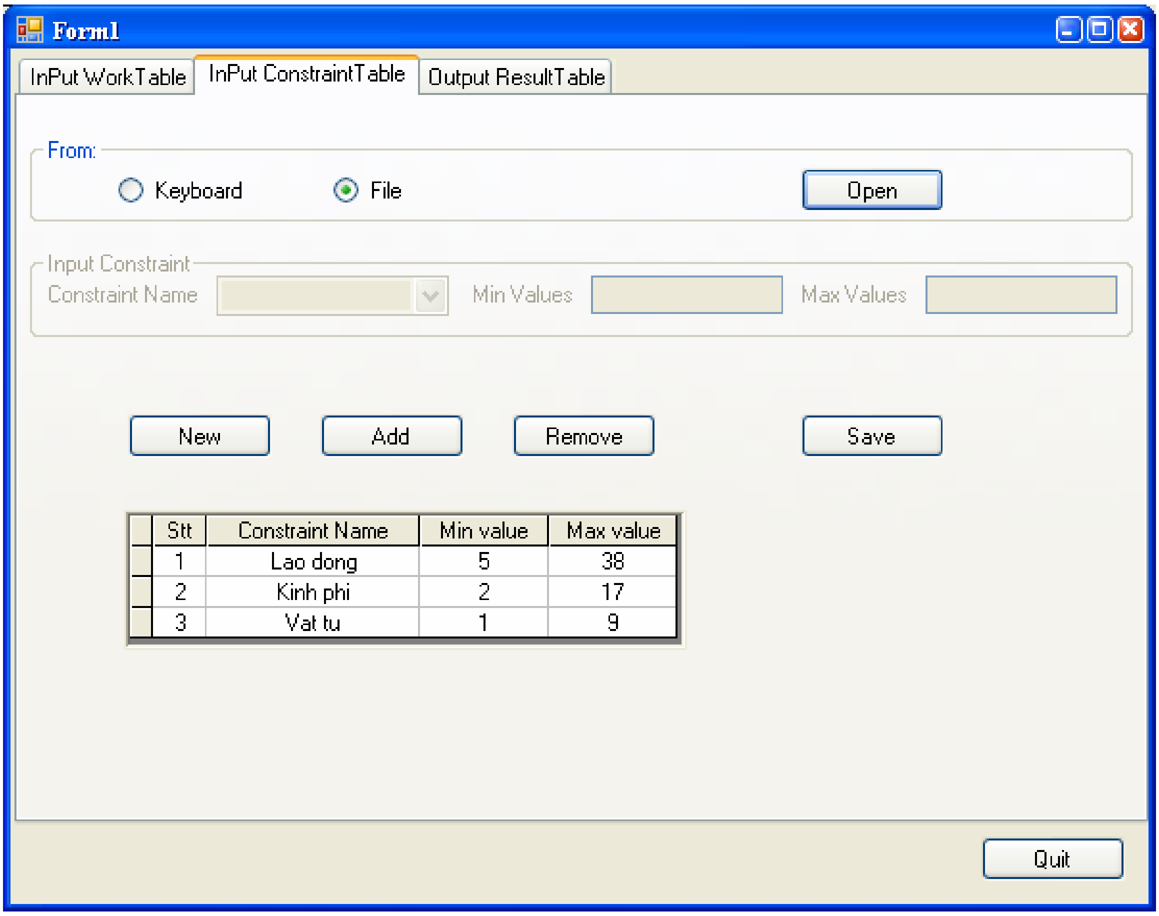
```
public bool Doc(string Ten_tap_tin)
{
    bool Kq = true;
    XmlDocument Tai_lieu = new XmlDocument();
    try
    {
        Tai_lieu.Load(Ten_tap_tin);
        XmlElement Goc = Tai_lieu.DocumentElement;
        foreach (XmlElement Nut in Goc.ChildNodes)
                    Constraint_Table.Add(new XL_CONSTRAINT (Nut));
    }
    catch (Exception Loi)
    {
        Kq = false;
    }
    return Kq;
}
```

4. Hàm ghi bảng ràng buộc từ bộ nhớ vào file *.xml:

```
public bool Ghi(string Ten_tap_tin)
{
    bool Kq = true;
    int i = 1;
    XmlDocument Tai_lieu = new XmlDocument();
```

```
    try
    {
            XmlElement         Root          =         Tai_lieu.CreateElement
("CONSTRAINTTABLE");

        foreach (XL_CONSTRAINT C in Constraint_Table)
        {
            XmlElement Nut = C.AppendNode(Tai_lieu, i++);
            Root.AppendChild(Nut);
        }
        Tai_lieu.AppendChild(Root);
        Tai_lieu.Save(Ten_tap_tin);
    }
    catch (Exception Loi)
    {
        Kq = false;
    }
    return Kq;
}
```



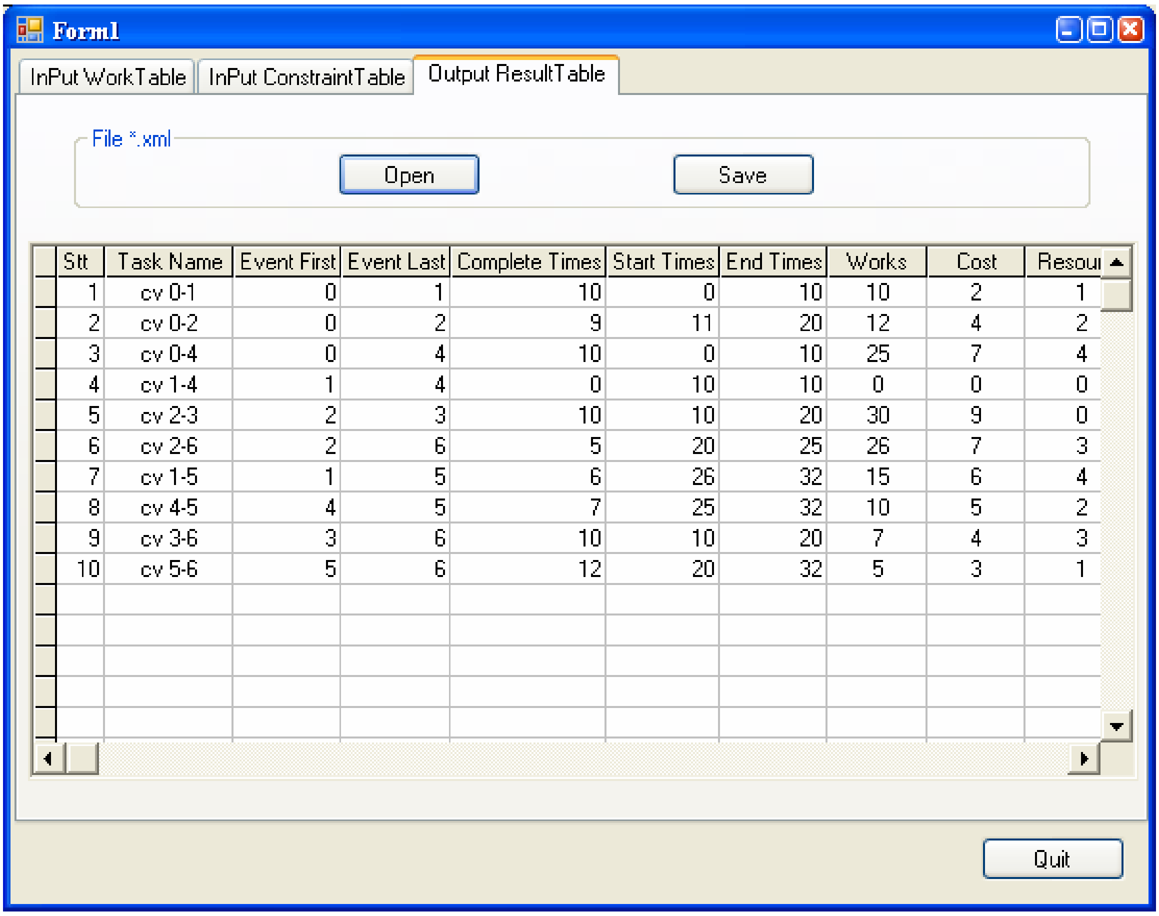
5. Hàm đọc bảng kết quả từ dữ liệu file *.xml vào bộ nhớ:

```
public bool Doc(string Ten_tap_tin)
{
    bool Kq = true;
    XmlDocument Tai_lieu = new XmlDocument();
    try
    {
        Tai_lieu.Load(Ten_tap_tin);
        XmlElement Goc = Tai_lieu.DocumentElement;
        foreach (XmlElement Nut in Goc.ChildNodes)
                    Result_Table.Add(new XL_RESULT (Nut));
    }
```

```
    catch (Exception Loi)
    {        Kq = false;
    }
    return Kq;
}
```

6. Hàm ghi bảng kết quả từ bộ nhớ vào file *.xml:

```
public bool Ghi(string Ten_tap_tin)
{    bool Kq = true;    int i = 1;
    XmlDocument Tai_lieu = new XmlDocument();
    try
    {   XmlElement Root = Tai_lieu.CreateElement ("RESULTTABLE");
        foreach (XL_RESULT R in Result_Table)
        {
            XmlElement Nut = R.AppendNode(Tai_lieu, i++);
            Root.AppendChild(Nut);
        }
        Tai_lieu.AppendChild(Root);
        Tai_lieu.Save(Ten_tap_tin);
    }
    catch (Exception Loi)
    {
        Kq = false;
    }
    return Kq;
}
```



Stt	Task Name	Event First	Event Last	Complete Times	Start Times	End Times	Works	Cost	Resou
1	cv 0-1	0	1	10	0	10	10	2	1
2	cv 0-2	0	2	9	11	20	12	4	2
3	cv 0-4	0	4	10	0	10	25	7	4
4	cv 1-4	1	4	0	10	10	0	0	0
5	cv 2-3	2	3	10	10	20	30	9	0
6	cv 2-6	2	6	5	20	25	26	7	3
7	cv 1-5	1	5	6	26	32	15	6	4
8	cv 4-5	4	5	7	25	32	10	5	2
9	cv 3-6	3	6	10	10	20	7	4	3
10	cv 5-6	5	6	12	20	32	5	3	1

<u>Kết luận:</u>

- Ứng dụng của XML rất rộng rãi: trao đổi trên phạm vi rộng (internet), biểu diễn thông tin, trao đổi thông tin và lưu trữ thông tin.
- Tương lai nó có thể thay thế cho các hệ quản trị cơ sở dữ liệu: Access, SQL Server,...
- Đặc biệt là ứng dụng lưu trữ dữ liệu trong điện thoại di động, máy điều khiển công cụ,..

Tài liệu tham khảo: của phụ lục A

[1] Nguyễn Tiến Huy, Công Nghệ XML và ứng dụng tập 1-2-3, Giáo Trình ĐH KHTN, 8/2005.

[2] Simon Robinson, Professional C#, Wrox Press Ltd, 2001.

Các trang Online:

- XPath tutorials
  - http://www.w3schools.com/xpath/
  - http://www.zvon.org/xxl/XPathTutorial/General/examples.html
- XQuery tutorials
  - http://www.w3schools.com/xquery/default.asp
  - http://www.db.ucsd.edu/people/yannis/XQueryTutorial.htm
- XML reading
  - http://www.rpbourret.com/xml/XMLAndDatabases.htm

Phụ lục B: Các phương pháp kinh điển

I. <u>Phương pháp leo đồi:</u>

Phương pháp leo đồi dùng kỹ thuật lặp, kỹ thuật được áp dụng cho một điểm duy nhất (điểm hiện hành trong không gian tìm kiếm). Trong mỗi lần lặp, một điểm mới được chọn từ lân cận của điểm hiện hành. Nếu điểm mới cung cấp giá trị tốt hơn của hàm mục tiêu, điểm mới sẽ thành điểm hiện hành. Nếu không, một lân cận khác của điểm hiện hành sẽ được chọn và quá trình được lặp lại. Phương pháp này dừng khi không cải thiện được lời giải.

Có nhiều phiên bản khác nhau của thuật giải leo đồi, thủ tục sau trình bày một dạng đơn giản của phương pháp leo đồi.

<u>Algorithm Thủ tục leo đồi</u>

Begin

    T:=0;

    Repeat

        Cucbo:=false;

        Chọn ngẫu nhiên một điểm a $\in$ I;

        Repeat

            Chọn ngẫu nhiên n điểm $a_1, a_2, .., a_n \in$ I quanh lân cận a;

            Tính $\psi(a_1), \psi(a_2), .., \psi(a_n)$;

            If(max{$\psi(a_1), \psi(a_2), .., \psi(a_n)$}< $\psi(a)$) then

                Cucbo:=true;

            Else   A:=a' sao cho $\psi(a')= \max\{\psi(a_1), \psi(a_2), .., \psi(a_n)\}$;

        Until cucbo;

        T:=t+1;

    Until ($t>t_{max}$)

End;

Trong thủ tục trên, thành công hay thất bại tùy thuộc nhiều vào thời điểm khởi tạo. Đối với bài toán tối ưu cục bộ, cơ hội tiến đến được tối ưu toàn cục rất mong manh. Để tăng cơ hội thành công, phương pháp leo đồi thường được thực thi cho một số (lớn) các điểm khởi đầu khác nhau (những điểm này không cần chọn ngẫu nhiên – một tập hợp các điểm khởi đầu của một lần thực thi phụ thuộc vào kết quả những lần chạy trước đó).

II. <u>Phương pháp mô phỏng luyện thép:</u>

Kỹ thuật mô phỏng luyện thép đã loại hầu hết những bất lợi những điểm bất lợi của phương pháp leo đồi: lời giải không còn phụ thuộc vào điểm khởi đầu nữa và (thường là) gần với điểm tối ưu. Ý tưởng chính của phương pháp dựa vào tham số điều khiển "nhiệt độ" T. Nói chung, nhiệt độ T càng thấp thì cơ hội nhận điểm mới càng nhỏ. Trong khi thực thi thuật giải, nhiệt độ T của hệ thống được hạ thấp theo từng bước. Giải thuật dừng đối với một số giá trị nhỏ nào đó của T mà với giá trị đó thì gần như không còn thay đổi nào được chấp nhận nữa.

Giải thuật sau trình bày một dạng đơn giản của phương pháp mô phỏng luyện thép:

<u>Algorithm Thủ tục luyện thép</u>

Begin

    T:=0;

    Khởi tạo nhiệt độ T;

    Chọn ngẫu nhiên một điểm a$\in$ I;

    Repeat

        Repeat

            Chọn a'$\in$ I trong lân cận của a;

            If (ψ(a')>ψ(a)) then

```
                    A:=a';
                Else
                    If(random[0,1]<exp{(ψ(a')-ψ(a))/T}) then
                        A:=a';
            Until(thỏa tiêu chuẩn dừng 1)
            T:=g(T,t);(g(T,t) là hàm giảm T theo số bước t)
            T:=t+1;
        Until(thỏa tiêu chuẩn dừng 2)
    End;
```

Trong thủ tục này, một điểm mới được chọn làm điểm hiện hành với xác suất $p=\exp\{(\psi(a')-\psi(a))/T\}$ với nhiệt độ T. Chính điểm mới này là cải tiến của leo đồi để có thể thoát khỏi cực tiểu cục bộ. Tuy nhiên thời gian chạy của thủ tục mở phỏng luyện thép rất lâu. Có ý kiến cho rằng: “Trong các kỹ thuật leo đồi, con kăngguru có thể hy vọng tìm được đỉnh núi gần chỗ nó khởi hành. Không bảo đảm rằng đó là đỉnh Everest, hay là một đỉnh cao nhất. Trong mô phỏng luyện thép, con kăngguru được ăn uống và hy vọng, một cách ngẫu nhiên, trong một thời gian dài, nó có thể đến được đỉnh Everest”.

III. <u>Phương pháp đơn hình tuyến tính:</u>

Phương pháp đơn hình tuyến tính được G. Dantzig giới thiệu vào năm 1947 để giải các bài toán quy hoạch tuyến tính. Từ đó đến nay phương pháp này được xem là thực sự hiệu quả để giải các bài toán có thể đưa về quy hoạch tuyến tính trong thực tế.

Xét bài toán quy hoạch tuyến tính ở dạng chuẩn (cách tiếp cận Vacek Chvátal và Robert J. Vanderbei).

$$\min \sum_{j=1}^{n} c_j x_j ,$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{i=1..m},$$

$$x_j \ge 0 \ \text{ j=1..n}$$

Việc đầu tiên ta đặt hàm mục tiêu là:

$$z = \sum_{j=1}^{n} c_j x_j$$

kế đến ta đưa biến phụ

$$w_i = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad \text{i=1..m}$$

vào bộ biến

$(x_1,..,x_n,w_1,..,w_m)=(x_1,..,x_n,x_{n+1},..,x_{n+m})$

Khi đó bài toán trở thành

$$z = \sum_{j=1}^{n} c_j x_j$$

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j \qquad \text{i=1..m}$$

Nội dung của thuật toán đơn hình là chuyển từ một trạng thái hiện tại đang xét sang một trạng thái khác với giá trị mục tiêu tốt hơn. Mỗi trạng thái có n biến cơ sở và m biến không cơ sở (biến phụ). Gọi B là tập các chỉ số tương ứng với các biến không cơ sở {n+1,n+2,..,n+m} và N là tập chỉ số của biến cơ sở. Ở trạng thái xuất phát thì N={1,2,..,n} và B={n+1,..,n+m}, nhưng chúng sẽ thay đổi sau mỗi bước. Ở mỗi bước, trạng thái đều có dạng:

$$z = \bar{z} + \sum_{j \in N} \bar{c}_j x_j$$

$$x_i = \bar{b}_i - \sum_{j=1}^{n} \bar{a}_{ij} x_j \qquad \text{với } i \in B$$

Ở đây dấu gạch trên đầu ký tự để chỉ rằng đại lượng này thay đổi qua các bước.

Ở mỗi bước lặp, đúng một biến từ không cơ sở trở thành biến cơ sở, được gọi là biến vào, và đúng một biến ra. Biến vào được chọn trong các biến có hệ số hàm mục tiêu (tức hệ số trong hàm mục tiêu) âm để làm giảm hàm mục tiêu. Nếu không có hệ số mục tiêu âm thì nghiệm nhận được ở bước lặp đó là tối ưu. Nếu có nhiều hệ số mục tiêu âm ta được phép chọn lựa. Cách thường dùng là chọn biến có hệ số (âm) nhỏ nhất để hy vọng làm giảm hàm mục tiêu nhiều nhất.

Biến ra được chọn để bảo đảm tính không âm của các biến. Giả sử biến vào đã được chọn là x_k, tức là giá trị của nó trở thành dương. Khi đó các biến đang là cơ sở sẽ bị đổi và bằng

$$x_i = \overline{b}_i - \overline{a}_{ik}\overline{x}_k \quad \text{với } i \in B$$

x_k được phép lớn đến mức mọi $x_i \geq 0$, $i \in B$. Tức là:

$$\frac{1}{x_k} \geq \frac{\overline{a}_{ik}}{b_i} \qquad \text{với } i \in B$$

hoặc tương đương:

$$x_k = \left(\max_{i \in B} \frac{\overline{a}_{ik}}{\overline{b}_i} \right)^{-1}$$

Ở đây ta quy ước $\frac{0}{0} = 0$ và ta xét sau trường hợp không có tỉ số $\frac{\overline{a}_{ik}}{\overline{b}_i}$ nào dương.

Vậy quy tắc chọn biến ra là chọn biến có chỉ số $l \in B$ mà $\frac{\overline{a}_{lk}}{\overline{b}_k} = \left(\max_{i \in B} \frac{\overline{a}_{ik}}{\overline{b}_i} \right)$.

Sau khi chọn biến vào và biến ra, việc chuyển trạng thái đang xét sang trạng thái mới là nhờ các phép toán hàng. Toàn bộ việc làm này gọi là phép xoay quanh chốt. Vì có thể có nhiều biến vào và biến ra có thể lấy đều bảo đảm giảm hàm mục tiêu và các biến vẫn không âm, ta sẽ thấy có các quy tắc cụ thể để tránh sự không xác định đó, được gọi là quy tắc xoay.

Trở lại với trường hợp không có tỉ số $\frac{\overline{a}_{ik}}{\overline{b}_i}$ với i ∈ B nào dương. Tỉ số này gặp phải khi tìm biến ra, sau khi đã xác định biến x_k là biến vào, tức là tăng từ 0 lên một số dương. Nhớ rằng lúc đó các biến cơ sở là:

$x_i = \overline{b}_i - \overline{a}_{ik}\overline{x}_k$ với i ∈ B

Mà $\overline{b}_i$ và $-a_{ik}$ là cùng dấu (vì $\frac{\overline{a}_{ik}}{\overline{b}_i} \le 0$) và là không âm. Do đó mọi biến cơ sở x_i không thể từ âm trở thành âm. Vậy biến vào có thể lấy giá trị lớn tùy ý để hàm mục tiêu tiến đến -∞. Lúc này ta nói là hàm mục tiêu không giới nội (dưới), hoặc bài toán không giới nội. Như vậy, trường hợp này sẽ không có nghiệm tối ưu.

<u>Tóm lại ta có thể xem phương pháp này gồm các bước sau:</u>

<u>Bước 1:</u> Thêm vào các biến bù để tạo trạng thái xuất phát và hàm mục tiêu.

<u>Bước 2:</u> Chọn biến vào. Nếu không chọn được (tức là không có hệ số mục tiêu âm) thì kết thúc với nghiệm nhận được tối ưu. Ngược lại, sang Bước 3.

<u>Bước 3:</u> Chọn biến ra x_k sao cho $x_k = \left(\max_{i\in B}\frac{\overline{a}_{ik}}{\overline{b}_i}\right)^{-1}$ và x_k>0.

Nếu không chọn được (tức không có tỉ số $\frac{\overline{a}_{ik}}{\overline{b}_i}$ nào dương) thì dừng, bài toán không có nghiệm tối ưu. Ngược lại sang bước 4.

<u>Bước 4:</u> Thay đổi các biến cơ sở theo công thức:

$$x_i = \overline{b}_i - \overline{a}_{ik}\overline{x}_k \quad \text{với } i\in B$$

Để tạo từ vựng mới và quay lại bước 2.

Phương pháp đơn hình thực sự hiệu quả để giải bài toán quy hoạch tuyến tính nhưng lại không thể áp dụng cho trường hợp phi tuyến. Để giải quyết vấn đề phi tuyến ta sẽ giải quyết ở chương sau.

IV. Phương pháp đơn hình phi tuyến:

Nelder-Mead đã đề ra thuật toán để giải quyết bài toán cực tiểu hàm n biến xác định trong không gian R^n:

Min f(x): $x \in R^n$

Thuật toán gồm các bước sau:

Bước 1: Tạo một không gian gồm n+1 điểm $x_1, x_2, .., x_{n+1}$ trong không gian R^n. Tính giá trị hàm mục tiêu f(x) tại n+1 đỉnh của đơn hình.

Bước 2: Tìm giá trị lớn nhất của f(x) và đỉnh của đơn hình đạt giá trị này, giá trị nhỏ nhất của f(x) và đỉnh của đơn hình đạt giá trị này, ta ký hiệu chúng là:

$$f_{max}, x_{max}, f_{min}, x_{min}$$

Bước 3: Nếu $|f\text{max} - f\text{min}| \le \varepsilon$ thì f_{min} và x_{min} là lời giải tối ưu của bài toán, kết thúc thuật toán. Ngược lại chuyển qua bước 4.

Bước 4: Tính điểm trọng tâm của đơn hình theo công thức sau:

$$x_s = \sum_{j=1}^{n+1} \frac{|f(x_j)|}{\sum |f(x_j)|} x_j$$

Bước 5: Chiếu đối xứng của x_{max} qua x_s ta được x_R:

$$x_R = (1+\alpha)x_S + \alpha x_{\max}$$

trong đó $\alpha \in [0,1]$ và có thể lấy bằng 1. Tính f(R)=$f(x_R)$.

Bước 6: Nếu $f_R \le f_{min}$ thì sang bước 7, trái lại sang bước 8.

Bước 7: Ta triển khai tiếp x_R để được x_E theo công thức:

$$x_E = x_S + 2(x_R - x_S) \text{ và tính } f(E)=f(x_E).$$

Nếu $f(x_E)<f_{min}$ thì thay x_{max} cũ bởi x_E, ta được một đơn hình mới và quay lại bước 2. ngược lại thì thay x_{max} cũ bởi x_R, ta được một đơn hình và quay lại bước 2.

Bước 8: Nếu $f_R<f_{max}$ thì thay x_{max} cũ bởi x_R, ta được một đơn hình mới và quay lại bước 2. Ngược lại thì sang bước 9.

Bước 9: Tính: $x_K = \frac{1}{2}x_{max} + \frac{1}{2}x_S$ và tính $f(K)=f(x_K)$.

Nếu $f_K<f_{max}$ thì thay x_{max} cũ bởi x_K, ta được một đơn hình mới và quay lại bước 2.

Ngược lại thì thu hẹp đơn hình theo công thức:

$$x_j = \frac{1}{2}x_j + \frac{1}{2}x_{min} \quad \text{với } \forall j \ne \min$$

Quay lại bước 2.

Phương pháp này thường áp dụng trong các bài toán kỹ thuật khi hàm f(x) có cấu trúc rất phức tạp, khó có thể tìm được tối ưu toàn cục. Trong thực hành, ta nên chọn đơn hình xuất phát ở bước 1 có (n+1) đỉnh ngẫu nhiên. Với các đơn hình xuất phát khác nhau phương pháp này có thể dẫn đến kết quả khác nhau. Do đó ta có thể thực hiện nhiều lần rồi chọn kết quả tốt nhất của các lần thực hiện đó. Phương pháp này tuy không phải lúc nào cũng cho lời giải tối ưu nhưng nếu không gian R^n là "không gian lồi" (không gian mà hai điểm nối nhau bất kỳ trong không gian này, thì tất cả các điểm nằm trên đường đó phải nằm trong không gian này) thì ta được kết quả là tối ưu.