

GIẢI PHÁP TOÁN HỌC CHO VIỆC PHÂN PHỐI CHI PHÍ TRONG ĐỘ TIN CẬY PHẦN MỀM

Nguyễn Thị Thuỳ Trang, Phan Thị Ngọc Mai*

Trường Đại học Công nghiệp Thực phẩm TP.HCM

*Email: *maiptn@cntp.edu.vn*

Ngày nhận bài: 05/9/2017; Ngày chấp nhận đăng: 05/12/2017

TÓM TẮT

Bài báo đề xuất giải pháp toán học trong việc phân phối chi phí cho độ tin cậy phần mềm: ứng dụng quy hoạch nguyên nhị phân tìm độ tin cậy lớn nhất của các module mua và ứng dụng quy hoạch phi tuyến tìm độ tin cậy lớn nhất của các module phát triển trong công ty. Nghiên cứu này đã giải quyết bài toán tìm độ tin cậy lớn nhất không vượt quá giới hạn chi phí đã cho, và ngược lại tìm khoảng chi phí nhỏ nhất để phần mềm có độ tin cậy là một giá trị xác định trước.

Từ khoá: Độ tin cậy phần mềm, module phát triển trong công ty, module tích hợp, module mua, phân phối chi phí.

1. GIỚI THIỆU

1.1. Định nghĩa

Độ tin cậy phần mềm [1] là xác suất của hoạt động phần mềm không có lỗi trong một khoảng thời gian nhất định trong một môi trường được chỉ định.

Trong bài báo này, cấu trúc của một phần mềm được tổ chức các module theo cấu trúc cây phân cấp và các module trong phần mềm được tồn tại dưới hai dạng: module đơn và module tích hợp [2].

Module đơn [2] là module được tạo ra từ chính nó. Module này có thể được mua từ bên ngoài thị trường và cũng có thể được phát triển trong công ty.

Module mua là những module trong công ty không có đầy đủ điều kiện để phát triển hoặc chi phí mua có thể tiết kiệm hơn so với việc phát triển trong công ty.

Module tích hợp là một module được tạo thành từ nhiều module đơn hoặc có thể từ các module đơn và module tích hợp khác.

Với lý do phân bổ nguồn tài nguyên hợp lý để tạo ra phần mềm có tính tin cậy cao và tiết kiệm được chi phí (dựa vào nguồn lực hiện có của công ty) nhà quản lý quyết định phân module phát triển trong công ty, phần module mua, và phần module dùng lại.

Một module được xem thích hợp để phát triển trong công ty khi trong công ty có đầy đủ điều kiện để phát triển và việc phát triển trong công ty có thể sẽ tiết kiệm hơn so với việc mua từ bên ngoài. Loại module này bao gồm module đơn và module tích hợp.

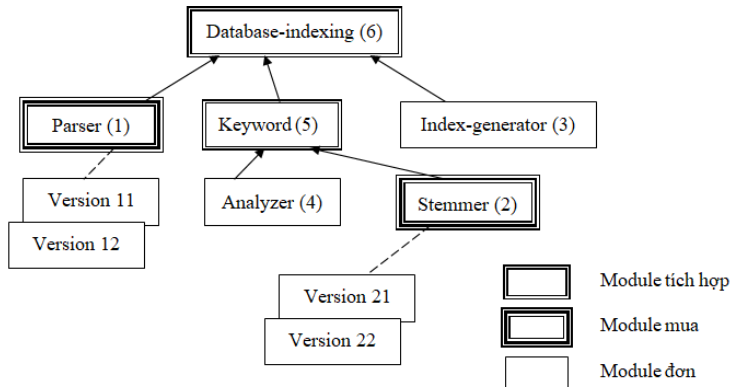
Một module được xem là thích hợp để mua khi có nhiều version trên thị trường và trong công ty không có đầy đủ điều kiện để phát triển hoặc chi phí để mua có thể tiết kiệm hơn so với việc phát triển trong công ty. Loại module này là module đơn.

Một module được xem là thích hợp dùng lại khi trong công ty đã có sẵn (do trong công ty phát triển hoặc đã mua trước đó) và việc dùng lại này rõ ràng không tốn chi phí.

Vấn đề chính trong bài toán này là phân phối chi phí cho độ tin cậy phần mềm. Do đó, các mô hình phát triển phần mềm chỉ bao gồm các module mua và các module phát triển trong công ty, còn phần module dùng lại do không có sự tham gia của nhân tố chi phí cho nên sẽ không được xét đến.

Ví dụ, trong Hình 1 dưới đây [2]:

- Có 4 module đơn: *Index-generator*, *Analyzer*, *Parse* và *Stemmer*.
 - + *Index-generator* và *Analyzer* là hai module đơn phát triển trong công ty.
 - + *Parse* và *Stemmer* là hai module mua, mỗi module có hai version trên thị trường.
- Hai module *Keyword* và *Database-index* là module tích hợp:
 - + Module *Keyword* được tích hợp từ hai module *Analyzer* và *Stemmer*
 - + Module *Database-index* được tích hợp từ ba module: *Parser*, *Keyword*, *Index-generator*.



Hình 1. Phần mềm Database-indexing

1.2. Độ tin cậy của các module

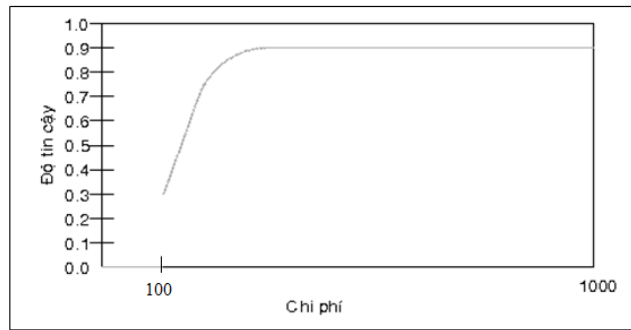
1.2.1. Độ tin cậy của module đơn phát triển trong công ty

Chọn hàm số mũ âm để tính độ tin cậy của các module phát triển trong công ty. Giả sử $x_i^{(0)}$ là chi phí cơ bản cần thiết để phát triển module i trong công ty. Chi phí cơ bản để phát triển module $x_i^{(0)}$ và độ tin cậy ban đầu là $r_i^{(0)}$. Nếu chi phí được tăng lên độ tin cậy có thể được tăng thêm. Độ tin cậy tối đa có thể đạt được cho module i là $r_i^{(\max)}$, độ tin cậy $r_i^{(\max)} = 1$ nhưng với mức độ đúng đắn 100% rất khó xảy ra, do đó $r_i^{(\max)} \leq 1$.

Độ tin cậy của một module i là r_i [2]:

$$r_i = \begin{cases} r_i^{(m)} - (r_i^{(m)} - r_i^{(0)}) e^{-\alpha_i(x_i - x_i^{(0)})} & x_i \geq x_i^{(0)} \\ 0 & x_i < x_i^{(0)} \end{cases} \quad (1)$$

trong đó α_i là một thông số phản ánh độ nhạy của độ tin cậy module mỗi khi có sự thay đổi chi phí. Giá trị α_i lớn sẽ tác động đến việc thay đổi chi phí x_i . Do đó khi $x_i = x_i^{(0)}$ thì $r_i = r_i^{(0)}$ và khi $x_i \rightarrow \infty$ thì $r_i = r_i^{(m)}$.



Hình 2. Độ tin cậy của một module phần mềm

Hình 2 được tham khảo từ *Berman and Cutler* (1998) biểu diễn độ tin cậy của công thức (1): $r_i^{(0)} = 0,3, r_i^{(m)} = 0,9, \alpha_i = 0,05, x_i^{(0)} = 100$. Trong trường hợp này, độ tin cậy bằng 0 khi chi phí nhỏ hơn 100 và 0.3 khi chi phí bằng 100. Độ tin cậy tăng đến giá trị lớn nhất là 0,9 khi $x_i \rightarrow \infty$ [2].

1.2.2. Độ tin cậy của một module mua

Một module i trong tập hợp các module mua được giả định có n_i version trên thị trường ($n_i > 1$).

Cho y_{ij} là một biến nhị phân biểu thị cho việc mua hay không mua version thứ j của module i . Nếu $y_{ij} = 1$ thì version j của module i được mua, ngược lại $y_{ij} = 0$ thì version j của module i không được mua. Với mục tiêu của mô hình là cực đại hóa độ tin cậy của phần mềm được ràng buộc trên tổng ngân sách đã cho (B). Do đó, để tiết kiệm chi phí mỗi module mua chỉ mua duy nhất một version trên thị trường, với điều kiện $\sum_{j=1}^{n_i} y_{ij} = 1$, độ tin cậy của module mua i là r_i :

$$r_i = \sum_{j=1}^{n_i} r_{ij} y_{ij} \quad (2.1)$$

và chi phí để mua module i là:

$$c_i = \sum_{j=1}^{n_i} c_{ij} y_{ij} \quad (2.2)$$

1.2.3. Độ tin cậy của một module tích hợp

Module T_i là một module tích hợp được tạo thành từ các module i_1, i_2, \dots, i_s . Độ tin cậy module T_i phụ thuộc vào độ tin cậy các module con của T_i .

Cho $r_{T_i}^{(m)}$ là độ tin cậy lớn nhất có thể đạt được của module tích hợp T_i . Do đó, độ tin cậy tối đa có thể đạt được của module T_i được tính theo công thức $r_{T_i}^{(\max)} = \prod_{k=1}^s r_{i_k}$. Tuy nhiên, trong quá trình tích hợp các module con có thể xảy ra những lỗi do có sự không tương thích giữa các module với nhau. Do đó, gọi $r_{T_i}^{(0)}$ là độ tin cậy nhỏ nhất có thể có của module T_i (có thể nhỏ hơn hoặc bằng $r_{T_i}^{(\max)}$). Cho q_{T_i} ($0 < q_{T_i} \leq 1$) là một hệ số phản ánh sự tương thích giữa các module. Vì vậy $r_{T_i}^{(0)} = q_{T_i} \prod_{k=1}^s r_{i_k} = q_{T_i} r_{T_i}^{(\max)}$.

Tương tự module đơn được phát triển trong công ty, độ tin cậy của một module tích hợp T_i :

$$R_{T_i} = \begin{cases} r_{T_i}^{(m)} - (r_{T_i}^{(m)} - r_{T_i}^{(0)}) e^{-\alpha_i(x_i - x_i^{(0)})} & x_i \geq x_i^{(0)} \\ 0 & x_i < x_i^{(0)} \end{cases} \quad (3)$$

trong đó $\alpha_i, x_i, x_i^{(0)}$ đã được định nghĩa trong phần trước.

Các số ngẫu nhiên được chọn cho ví dụ [2]:

$$r_{11} = 0,7, c_{11} = 5$$

$$r_{12} = 0,9, c_{12} = 6$$

$$r_{21} = 0,87, c_{21} = 7$$

$$r_{22} = 0,95, c_{22} = 8$$

$$r_3^{(m)} = 0,83, r_3^{(0)} = 0,53, \alpha_3 = 0,3, x_3^{(0)} = 2$$

$$r_4^{(m)} = 0,9, r_4^{(0)} = 0,5, \alpha_4 = 0,4, x_4^{(0)} = 3,5$$

$$q_5 = 0,7, \alpha_5 = 0,25, x_5^{(0)} = 4$$

$$q_6 = 0,8, \alpha_6 = 0,3, x_6^{(0)} = 3$$

Để tính toán độ tin cậy của hệ thống, đầu tiên tính độ tin cậy của các module mua (1) và (2) và các module đơn (3) và (4).

$$r_1 = r_{11}y_{11} + r_{12}y_{12}$$

$$r_2 = r_{21}y_{21} + r_{22}y_{22}$$

$$r_3 = \begin{cases} 0,83 - (0,83 - 0,52)e^{-0,3(x_3 - 2)} & x_3 \geq 2 \\ 0 & x_3 < 2 \end{cases}$$

$$r_4 = \begin{cases} 0,9 - (0,9 - 0,5)e^{-0,4(x_4 - 3,5)} & x_4 \geq 3,5 \\ 0 & x_4 < 3,5 \end{cases}$$

Độ tin cậy của module tích hợp Keyword (5) là:

$$r_5 = \begin{cases} r_5^{(m)} - (r_5^{(m)} - r_5^{(0)}) e^{-0,25(x_5 - 4)} & x_5 \geq 4 \\ 0 & x_5 < 4 \end{cases}$$

trong đó $r_5^{(0)} = r_2r_4$ và $r_5^{(m)} = 0,8r_5^{(m)}$

\hat{R} là độ tin cậy của module tích hợp Database-indexing (6)

$$r_6 = \begin{cases} r_6^{(m)} - (r_6^{(m)} - r_6^{(0)}) e^{-0,3(x_6 - 3)} & x_6 \geq 3 \\ 0 & x_6 < 3 \end{cases}$$

khi đó $r_6^{(0)} = r_1r_3r_5$ và $r_6^{(m)} = 0,8r_6^{(m)}$

Bài toán:

$$\max \hat{R}$$

S.T.

$$c_{11}y_{11} + c_{12}y_{12} + c_{21}y_{21} + c_{22}y_{22} + x_3 + x_4 + x_5 + x_6 \leq B$$

$$y_{11} + y_{12} = 1$$

$$y_{21} + y_{22} = 1$$

$$x_i \geq x_i^{(0)}, i = 3, 4, 5, 6$$

$$y_{11}, y_{12}, y_{21}, y_{22} = 0 \text{ hoặc } 1$$

Thông qua việc sử dụng hàm Solver của công cụ Microsoft Excel. Sau khi giải bài toán thu được các kết quả sau:

Bảng 1. Giải pháp cho những nguồn ngân sách khác nhau

B	y_{11}	y_{12}	y_{21}	y_{22}	x_3	x_4	x_5	x_6	Độ tin cậy Tối ưu
25	1	0	1	0	2,0000	4,0000	4,0000	3,0000	0,11826
26	0	1	1	0	2,0000	4,0000	4,0000	3,0000	0,15205
30	0	1	0	1	3,3816	5,6183	4,0000	3,0000	0,2518
35	0	1	0	1	5,1168	6,9833	4,7556	4,1441	0,3491
40	0	1	0	1	6,3842	7,9627	6,2414	5,4115	0,4269
45	0	1	0	1	7,6511	8,9325	7,7371	6,6785	0,4870
50	0	1	0	1	8,9178	9,8958	9,2410	9,9452	0,5316
55	0	1	0	1	10,1842	10,8547	10,7494	9,2116	0,5639
60	0	1	0	1	11,4505	11,8105	12,2610	10,4778	0,5868
70	0	1	0	1	13,9826	13,7167	15,2906	13,0099	0,6140
80	0	1	0	1	16,5145	15,6189	18,3246	15,5418	0,6270
100	0	1	0	1	21,5779	19,4188	24,3978	20,6053	0,6361
150	0	1	0	1	34,2205	28,9238	39,5821	33,2734	0,63863
200	0	1	0	1	46,7863	37,8185	56,3336	45,0614	0,63868

2. MÔ HÌNH TỔNG QUÁT

Giả sử trong phần mềm tồn tại n module và các module này có thể được mua ở bên ngoài thị trường hoặc được phát triển trong công ty. Cho z_i là một biến nhị phân, khi $z_i = 1$ thì module i là được phát triển trong công ty, ngược lại nếu $z_i = 0$ thì module i được mua từ bên ngoài. Số version của những module i được mua bên ngoài thị trường là n_i và mỗi module mua chỉ mua một version trong số các version của module đó. Từ một module có thể được phát triển trong công ty hoặc được mua từ bên ngoài thị trường, $z_i + \sum_{j=1}^{n_i} y_{ij} = 1$. Gọi r_i là độ tin cậy của module i được phát triển trong công ty với chi phí x_i , r_{ij} , c_{ij} là độ tin cậy và chi phí của một version j của module i . Do đó, đối với bất kỳ một module phần mềm i nào có độ tin cậy R_i được cho bởi:

$$R_i = r_i z_i + \sum_{j=1}^{n_i} r_{ij} y_{ij}$$

Tương tự, gọi C_i là chi phí để thực hiện một module i :

$$C_i = x_i z_i + \sum_{j=1}^{n_i} c_{ij} y_{ij}$$

Trong trường hợp này bài toán được phát biểu như sau:

$$\text{Max } \hat{R} \quad (\text{GP1})$$

S.T.

$$\sum_{i=1}^n \sum_{j=1}^{n_i} c_{ij} y_{ij} + \sum_{i=1}^n x_i z_i \leq B \quad (\text{GP2})$$

$$z_i + \sum_{j=1}^{n_i} y_{ij} = 1 \quad \text{với } i = 1, 2, K, n \quad (\text{GP3})$$

$$z_i, y_{ij} = 0, 1 \quad \text{với } i = 1, K, n; j = 1, K, n_i \quad (\text{GP4})$$

trong đó:

- (GP1) cực đại hoá độ tin cậy.
- (GP2) đảm bảo tổng các khoảng chi tiêu là không vượt ngân sách.
- (GP3) đảm bảo có đúng một module i được phát triển trong công ty hoặc có duy nhất một version được mua trên thị trường cho module i .
- (GP4) đảm bảo các biến y_{ij}, z_i là các biến nhị phân.

3. GIẢI QUYẾT BÀI TOÁN

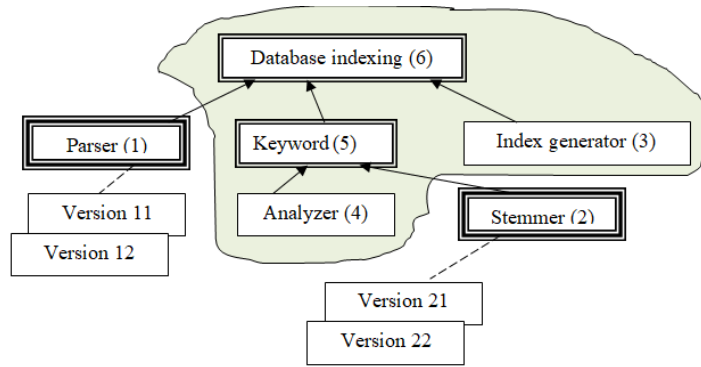
Trong giai đoạn thiết kế phần mềm, nhà quản lý sẽ ước lượng độ tin cậy của phần mềm dựa vào chi phí đã cho. Dựa vào các yếu tố hiện có của công ty, nhà quản lý có thể chủ động xây dựng kế hoạch phân phối chi phí giữa việc mua và phát triển các module một cách hợp lý để tạo ra phần mềm có độ tin cậy mong muốn.

Để giải quyết bài toán, một giải pháp sử dụng phương pháp quy hoạch nguyên nhị phân [3] được đề xuất việc phân phối chi phí cho các module mua, kết hợp với phương pháp quy hoạch phi tuyến giải quyết hàm số mũ nhiều biến [4], để phân phối chi phí cho các module phát triển trong công ty. Thông qua việc kết hợp này, các hàm trong Matlab [5] được dùng để xây dựng hai giải pháp cho phép kết hợp giữa bài toán quy hoạch nguyên và quy hoạch phi tuyến một cách tự động thông qua việc phân hoạch bài toán như sau:

Bước 1: Phân hoạch bài toán thành hai phần: module mua và module phát triển trong công ty.

- Module mua: các biến trong module mua là các biến nguyên nhị phân (chỉ nhận giá trị: 0 hoặc 1) sẽ được thực hiện thông qua bài toán quy hoạch nguyên.
- Phần module phát triển trong công ty: các biến trong module phát triển trong công ty là các biến thực. Hàm mục tiêu là một hàm nhiều biến, các điều kiện ràng buộc là các phương trình phi tuyến, bài toán quy hoạch phi tuyến cũng có thể dễ dàng thực hiện cho phần module phát triển trong công ty.

Bước 2: Kết hợp hai bài toán lại thông qua đó tìm độ tin cậy lớn nhất có thể đạt được của phần mềm sao cho không vượt quá giới hạn ngân sách đã cho.



Hình 3. Sự phân hoạch bài toán

Do vấn đề đặc biệt này, có thể phân hoạch bài toán thành hai bài toán con để giải quyết. Giả sử kinh phí cung cấp cho dự án phần mềm này là B , sẽ trích ra phần B' để mua các module, phần còn lại ($B-B'$) được dùng vào việc phát triển các module trong công ty.

Vấn đề phân phối chi phí giữa module mua và module phát triển trong công ty là vấn đề rất quan trọng trong giải quyết bài toán tối ưu hóa phân phối chi phí cho độ tin cậy phần mềm. Hai vấn đề được thực hiện để tối ưu hoá bài toán là: (1) tìm ra độ tin cậy lớn nhất có thể có để không vượt quá giới hạn ngân sách đã cho; (2) tìm ra chi phí nhỏ nhất của phần mềm có độ tin cậy là một hằng số cho trước. Để giải quyết hai vấn đề nêu trên, cần thực hiện hai bài toán sau:

3.1. Bài toán 1

Tìm ra giải pháp phân phối chi phí để phần mềm có độ tin cậy lớn nhất mà không vượt quá giới hạn ngân sách đã cho, các bước để thực hiện bài toán:

- Bước 1: (Khởi tạo) nhập vào các thông số của module phần mềm:
 - + Số module phần mềm, số module mua, số module đơn phát triển trong công ty.
 - + Số version của mỗi module mua, chi phí và độ tin cậy của từng version.
 - + Chi phí khởi tạo, độ tin cậy lớn nhất, độ tin cậy nhỏ nhất, thông số phản ánh độ nhạy của module đơn phát triển trong công ty.
 - + Chi phí khởi tạo, thông số phản ánh sự tương thích của các module con, thông số phản ánh độ nhạy của module tích hợp, các module con của module tích hợp, chuyển sang bước 2.
- Bước 2: Nhập chi phí để phát triển phần mềm (B). Nếu $B \geq \sum_{i=1}^m c_i^{(\min)} + \sum_{i=m+1}^n x_i^{(0)}$ thì yêu cầu nhập lại B , ngược lại chuyển sang bước 3.
- Bước 3: Nhập tổng chi phí để mua các module mua B' . Nếu $\sum_{i=1}^m c_i^{(\min)} \leq B' \leq \sum_{i=1}^m c_i^{(\max)}$ và $B - B' \geq \sum_{i=m+1}^n x_i^{(0)}$ thì yêu cầu nhập lại B' , ngược lại chuyển sang bước 4.
- Bước 4: Tính độ tin cậy và chi phí cho từng module mua, chuyển sang bước 5.
- Bước 5: Thiết lập mối quan hệ, những điều kiện ràng buộc giữa các module đơn và module tích hợp và chuyển sang bước 6.

- Bước 6: Tìm độ tin cậy lớn nhất của các module trong phần mềm và chi phí của từng module ứng với độ tin cậy đó. Nếu độ tin cậy không thỏa mãn yêu cầu thì chuyển sang bước 7, ngược lại chuyển sang bước 8.
- Bước 7: Nhập giai đoạn hiệu chỉnh. Khi $Y = 0$ thì chuyển sang bước 1 và thực hiện lại bài toán, khi $Y = 1$ thì chuyển sang bước 2, khi $Y = 2$ thì chuyển sang bước 3.
- Bước 8: (*Kết thúc*) Xuất ra kết quả: độ tin cậy của phần mềm, độ tin cậy của các module, chi phí của các module.

3.2. Bài toán 2

Tìm chi phí nhỏ nhất để phần mềm có độ tin cậy lớn hơn hoặc bằng độ tin cậy đã định sẵn, các bước để thực hiện bài toán:

- Bước 1: (*Khởi tạo*) nhập vào các thông số:
 - + Số module phần mềm, số module mua, số module đơn phát triển trong công ty.
 - + Số version của mỗi module mua, chi phí và độ tin cậy của từng version.
 - + Chi phí khởi tạo, độ tin cậy lớn nhất, độ tin cậy nhỏ nhất, thông số phản ánh độ nhạy của module đơn phát triển trong công ty.
 - + Chi phí khởi tạo, thông số phản ánh sự tương thích của các module con, thông số phản ánh độ nhạy của module tích hợp, các module con của module tích hợp, chuyển sang bước 2.
- Bước 2: Nhập tổng chi phí để mua các module (B'). Nếu $\sum_{i=1}^m c_i^{(\min)} < B' < \sum_{i=1}^m c_i^{(\max)}$ thì chuyển sang bước 4, ngược lại yêu cầu nhập lại.
- Bước 3: Nhập độ tin cậy phần mềm, chuyển sang bước 4.
- Bước 4: Tính ra độ tin cậy và chi phí cho từng module mua, chuyển sang bước 5.
- Bước 5: Thiết lập mối quan hệ và các ràng buộc giữa các module đơn và module tích hợp, chuyển sang bước 6.
- Bước 6: Tìm chi phí nhỏ nhất của các module trong phần mềm và độ tin cậy của các module ứng với chi phí đó. Nếu tìm ra chi phí thỏa mãn điều kiện thì chuyển sang bước 8, ngược lại chuyển sang bước 7.
- Bước 7: Nhập giai đoạn hiệu chỉnh. Khi $Y = 0$ thì chuyển sang bước 1 và thực hiện lại bài toán, khi $Y = 1$ thì chuyển sang bước 2, khi $Y = 2$ thì chuyển sang bước 3.
- Bước 8: (*Kết thúc*) Xuất ra kết quả: tổng chi phí của phần mềm, độ tin cậy của các module, chi phí của các module.

4. THỬ NGHIỆM

Trong nghiên cứu này, ngôn ngữ được chọn để viết và kiểm nghiệm đó là ngôn ngữ MATLAB [4]. MATLAB cung cấp một công cụ tính toán toán học và lập trình bậc cao dễ sử dụng, hiệu quả và thân thiện với người dùng.

Bảng 2. Giải pháp cho những nguồn ngân sách khác nhau cho bài toán 1

B	B'	$y_{11}y_{12}$ $y_{21}y_{22}$	$[x_3 \dots x_6]$	$[r_3 \dots r_5]$	Độ tin cậy tối ưu (r_6)
25	12	1010	[2,0000, 4,0000, 4,0000, 3,0000]	0,5300, 0,5725, 0,3985	0,1183
26	13	0110	[2,0000, 4,0000, 4,0000, 3,0000]	0,5300, 0,5725, 0,3985	0,1521
30	14	0101	[3,3822, 5,6178, 4,0000, 3,0000]	0,6318, 0,7285, 0,5537	0,2519
35	14	0101	[5,0354, 6,9367, 4,8520, 4,1759]	0,7093, 0,7988, 0,6362	0,3491
40	14	0101	[6,1630, 7,9788, 6,3025, 5,5557]	0,7093, 0,7988, 0,6362	0,4267
45	14	0101	[7,0692, 9,0706, 7,9018, 6,9584]	0,7093, 0,7988, 0,6362	0,4863
50	14	0101	[8,9192, 9,8961, 9,2542, 7,9305]	0,7924, 0,8690, 0,7812	0,5317
55	14	0101	[10,186, 10,8559, 10,8019, 9,1562]	0,8043, 0,8789, 0,8045	0,5639
60	14	0101	[11,4559, 11,8187, 12,3789, 10,3465]	0,8124, 0,8856, 0,8206	0,5868
80	14	0101	[16,6108, 15,6124, 18,6015, 15,1754]	0,8263, 0,8969, 0,8476	0,6270
100	14	0101	[16,0927, 30,7139, 23,2710, 15,9224]	0,8256, 0,9000, 0,8536	0,6317
150	14	0101	[35,6548, 65,0458, 16,9591, 18,3404]	0,8300, 0,9000, 0,8483	0,6324
200	14	0101	[49,2676, 89,9183, 22,2423, 24,5718]	0,8300, 0,9000, 0,8532	0,6372

Bảng 3. Giải pháp cho những nguồn ngân sách khác nhau cho bài toán 2

Độ tin cậy phần mềm (r_6)	B'	$y_{11}y_{12}$ $y_{21}y_{22}$	$[r_3 \dots r_5]$	$[x_3 \dots x_6]$	Tổng chi phí tối ưu (B)
0,1183	12	1010	[0,5300, 0,5727, 0,3986]	[2,0000, 4,0013, 4,0000, 3,0000]	25,0013
0,2	12	1010	[0,6687, 0,7621, 0,5304,]	[4,0674, 6,1620, 4,0000, 3,0947]	29,3241
0,25	13	0110	[0,6608, 0,7550, 0,5255]	[3,9086, 6,0367, 4,0000, 3,0000]	29,9453
0,3	14	0101	[0,6847, 0,7764, 0,5901]	[4,4172, 6,4370, 4,0000, 3,4445]	32,2987
0,35	14	0101	[0,7127, 0,8011, 0,6355]	[5,1295, 6,9932, 4,7705, 4,1570]	35,0502
0,4	14	0101	[0,7372, 0,8224, 0,6787]	[5,9117, 7,5989, 5,6860, 4,9391]	38,1356
0,45	14	0101	[0,7596, 0,8415, 0,7193]	[6,8300, 8,3047, 6,7667, 5,8568]	41,7582
0,5	14	0101	[0,7801, 0,8588, 0,7577]	[7,9797, 9,1827, 8,1267, 7,0065]	46,2957
0,55	14	0101	[0,7992, 0,8747, 0,7942]	[9,5842, 10,4007, 10,0340, 8,6112]	52,6301
0,6	14	0101	[0,8170, 0,8894, 0,8290]	[12,4556, 12,5681, 13,4632, 11,4833]	63,9701

Bảng 4. Giải pháp cho những nguồn ngân sách khác nhau cho bài toán 1 có 22 module

B	B'	y_{ij}	$[x_5 \dots x_{22}]$	$[r_5 \dots r_{21}]$	Độ tin cậy tối ưu (r_{22})
145	26	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 3,00, 5,00,7,00, 8,00, 6,00, 5,00, 7,50, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00,10,00]	[0,30, 0,56, 0,49, 0,50, 0,35, 0,55, 0,46, 0,35, 0,21, 0,13, 0,05, 0,27, 0,02, 0,09, 0,26, 0,04, 0,09]	0,0035
200	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 9,52, 12,40,7,00, 16,20, 14,37, 11,67, 15,16, 6,50, 6,00, 6,50, 8,00, 6,50, 6,90, 8,00, 7,28, 9,00, 10,00]	[0,30, 0,93, 0,98, 0,50, 0,93, 0,86, 0,96, 0,94, 0,24, 0,22, 0,17, 0,89, 0,08, 0,80, 0,81, 0,75, 0,75]	0,55
250	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 16,23, 24,49, 7,00, 19,06, 20,23, 14,21, 18,50, 6,50, 6,00, 6,50, 9,31, 6,50, 11,09, 11,54, 11,67, 11,53, 10,64]	[0,30, 0,96, 0,99, 0,50, 0,96, 0,89, 0,98, 0,96, 0,24, 0,22, 0,17, 0,93, 0,08, 0,89, 0,87, 0,86, 0,83]	0,71
300	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 17,80, 28,33, 7,00, 23,14, 26,21, 21,02, 20,11, 6,50, 6,00, 6,50, 16,52, 6,50,14,53, 15,69, 14,51, 5,29, 15,35]	[0,30, 0,96, 0,99, 0,50, 0,97, 0,90, 0,99, 0,97, 0,24, 0,22, 0,17, 0,95, 0,08, 0,91, 0,89, 0,90, 0,85]	0,77
400	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 34,63, 49,65, 7,00, 29,58, 34,46, 19,64, 31,98, 6,50, 6,00, 6,50, 20,00, 6,50, 21,01, 23,57, 22,67, 22,33, 18,97]	[0,30, 0,96, 0,99, 0,50, 0,97, 0,90, 0,99, 0,97, 0,24, 0,22, 0,17, 0,95, 0,08, 0,92, 0,89, 0,91, 0,86]	0,7856
500	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 46,04, 62,22, 7,00, 35,48, 41,33, 23,38, 38,70, 6,50, 6,00, 6,50, 37,30, 6,50, 20,67, 32,59, 24,98, 33,11, 32,70]	[0,30, 0,96, 0,99, 0,50, 0,97, 0,90, 0,99, 0,97, 0,24, 0,22, 0,17, 0,95, 0,08, 0,92, 0,89, 0,91, 0,86]	0,7882

Bảng 5. Giải pháp cho những nguồn ngân sách khác nhau cho bài toán 2 có 22 module

Độ tin cậy phần mềm (r_{22})	B'	y_{ij}	$[x_5 \dots x_{22}]$	$[r_5 \dots r_{21}]$	Tổng chi phí (B)
0,0035	26	1 0 0 0 1 0 1 0 0 1 0 0 0	[4,00, 3,00, 5,00, 7,00, 8,00, 6,00, 5,00, 7,50, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00,10,00]	[0,30, 0,56, 0,49, 0,50, 0,35, 0,55, 0,46, 0,35, 0,21, 0,13, 0,05, 0,27, 0,02, 0,09, 0,26, 0,04, 0,09]	135,00
0,1	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 4,20, 7,84, 7,00, 10,32, 6,34, 6,82, 9,77, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00, 30,33]	[0,30, 0,71, 0,87, 0,50, 0,69, 0,58, 0,76, 0,72, 0,24, 0,22, 0,17, 0,61, 0,08, 0,40, 0,49, 0,29, 0,34]	154,78
0,2	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 5,21, 8,76, 7,00, 11,43, 7,77, 7,74, 10,78, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00, 38,21]	[0,30, 0,79, 0,91, 0,50, 0,78, 0,68, 0,84, 0,80, 0,24, 0,22, 0,17, 0,71, 0,08, 0,53, 0,60, 0,42, 0,47]	161,19

Độ tin cậy phần mềm (r_{22})	B'	y_{ij}	$[x_5 \dots x_{22}]$	$[r_5 \dots r_{21}]$	Tổng chi phí (B)
0,3	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 6,15, 9,59, 7,00, 12,49, 9,17, 8,59, 11,72, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00, 38,66]	[0,30, 0,85, 0,94, 0,50, 0,84, 0,74, 0,88, 0,86, 0,24, 0,22, 0,17, 0,78, 0,08, 0,62, 0,68, 0,53, 0,57]	167,23
0,5	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 8,59, 11,63, 7,00, 15,24, 12,89, 10,77, 14,16, 6,50, 6,00, 6,50, 8,00, 6,50, 6,00, 8,00, 7,00, 9,00, 43,05]	[0,30, 0,92, 0,97, 0,50, 0,92, 0,84, 0,95, 0,93, 0,24, 0,22, 0,17, 0,87, 0,08, 0,76, 0,79, 0,70, 0,72]	182,78
0,7	35	0 0 0 1 0 1 0 0 1 0 0 0 1	[4,00, 12,54, 14,84, 7,00, 19,75, 19,14, 14,30, 18,11, 6,50, 6,00, 6,50, 9,02, 6,50, 10,57, 11,01, 10,86, 10,85, 47,95]	[0,30, 0,95, 0,98, 0,50, 0,96, 0,88, 0,98, 0,96, 0,24, 0,22, 0,17, 0,92, 0,08, 0,87, 0,86, 0,85, 0,82]	222,50

5. KẾT LUẬN

Tự động hóa quá trình phân phối chi phí để đánh giá độ tin cậy phần mềm là một bài toán mở, nhiều phương pháp đã được đưa ra để giải quyết. Phương pháp kết hợp quy hoạch nguyên và quy hoạch phi tuyến của hàm nhiều biến là một giải pháp được đề xuất để giải quyết vấn đề này. Đứng trên góc độ của một công trình nghiên cứu, bài báo đã đưa ra hai giải pháp nhằm cung cấp thêm một cách thức để giải quyết bài toán tối ưu hoá phân phối chi phí cho độ tin cậy phần mềm: Một là tìm ra độ tin cậy lớn nhất có thể có để không vượt quá giới hạn ngân sách đã cho; Hai là tìm ra chi phí nhỏ nhất của phần mềm có độ tin cậy là một hằng số cho trước.

Tuy nhiên, để có thể xây dựng một giải pháp hữu hiệu hơn cho bài toán này, cần thực hiện các nghiên cứu mở rộng như: Xây dựng các mô hình khác mô hình số mũ để giải quyết việc phân phối chi phí cho độ tin cậy phần mềm; Xây dựng hàm quy hoạch phi tuyến tồn tại cả biến nguyên và biến thực giải quyết bài toán một cách tối ưu nhất; Lấy dữ liệu thực từ các hãng phần mềm ứng dụng vào mô hình.

TÀI LIỆU THAM KHẢO

1. Hoang Pham - Software reliability, Springer – Verlag Singapore Pte.Ltd, 2000.
2. Berman O., Cutler M. - Cost allocation for software reliability, in: Recent advances in reliability and quality engineering, vol.2, Series on Quality, Reliability & Engineering Statistics, Editor Hoang Pham, Word Scientific, 2001.
3. John W. Chinneck - Practical optimization: A gentle introduction, Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada.
4. Jonathan F. Bard - Practical bilevel optimization: Algorithms and applications, Springer, 1999.
5. Nguyễn Nhật Lệ, Phan Mạnh Dân - Giải bài toán tối ưu hoá ứng dụng bằng MATLAB-MAPLE tối ưu hoá tĩnh và điều khiển tối ưu, NXB Khoa học Kỹ thuật, Hà Nội, 2005.

ABSTRACT

**MATHEMATICAL SOLUTIONS TO THE COST ALLOCATION
OF SOFTWARE RELIABILITY**

Nguyen Thi Thuy Trang, Phan Thi Ngoc Mai*
Ho Chi Minh city University of Food Industry
*Email: *maiptn@cntp.edu.vn*

In this paper, two mathematical solutions to the cost allocation of software reliability are proposed, including application of binary integer programming to find the greatest reliability of the purchased modules, and use of nonlinear programming to find the greatest reliability of the module development in the company. This study finds the greatest possible reliability of the software without exceeding the given cost limit and the smallest cost so that the software reliability is a predetermined value.

Keywords: Software reliability, in-house developed module, integration module, purchased module, cost allocation.