

PHÂN LỚP VÀ TRÁNH XUNG ĐỘT TRONG BÀI TOÁN LẬP KẾ HOẠCH VỚI THÔNG TIN KHÔNG ĐẦY ĐỦ

NGUYỄN QUỐC ANH, PHẠM HỒNG HẠNH, HỒ SỸ LỢI

Abstract. This paper describes a new algorithm for planing with incomplete information and conflicts. The given planing problem has two optimazation criteria: maximize the utility and minimize the conflicts of the plan. In order to achieve the first optimization goal by utility with incomplete data we build a clustering algorithm based on a fuzzy comparison method for intervals. To minimize the conflicts while keeping a light utility, we apply genetic algorithm. The experiments show that a good balance is achieved by using a dual algorithm with a flexible order of maximizing utility and minimizing conflicts.

1. GIỚI THIỆU

Bài toán lập kế hoạch là bài toán kinh điển được sự quan tâm đặc biệt bởi các ứng dụng rộng rãi của nó. Trong môi trường bất định, bài toán lập kế hoạch cho số lớn các hoạt động yêu cầu phải xử lý tối ưu với thông tin không đầy đủ, tránh xung đột giữa các hoạt động, đồng thời phải giải quyết vấn đề bùng nổ tổ hợp. Đây là mục tiêu rất khó thực hiện.

Bài này đưa ra cách giải quyết bài toán lập kế hoạch các hoạt động không rõ kết quả. Với những tham số đầu vào đặc trưng cho mỗi hoạt động là tập giá trị mờ, tập các hoạt động xung đột, chúng tôi sử dụng thuật toán kép để tìm nghiệm tối ưu, đồng thời tránh xung đột nhằm đem lại độ thuận lợi cao nhất cho kế hoạch.

Các kết quả thực nghiệm thu được cho thấy thời gian thực hiện của phương pháp tương đối ngắn và chất lượng của kế hoạch khá tốt.

2. BÀI TOÁN LẬP KẾ HOẠCH CHO CÁC HOẠT ĐỘNG KHÔNG RÕ KẾT QUẢ

2.1. Phát biểu bài toán

Từ một tập các hoạt động mà mỗi hoạt động đặc trưng bởi bốn yếu tố:

- + Độ ưu tiên về thời gian xảy ra.
- + Độ ích lợi riêng của từng hoạt động.
(Độ ích lợi riêng của một hoạt động là chỉ số ích lợi nó mang cho toàn bộ kế hoạch, chưa tính đến các yếu tố xung đột và yếu tố thời gian).
- + Tập tên các hoạt động sẽ xảy ra xung đột nếu xếp cạnh nó.
(Xung đột là một hiện tượng làm giảm chất lượng của kế hoạch, xảy ra khi hai hoặc nhiều hoạt động nào đó xếp cạnh nhau. Ví dụ: hai loại thuốc uống liền nhau có thể gây hậu quả xấu).
- + Thời gian để thực hiện hoạt động đó.

Thiết lập kế hoạch tối ưu để thực hiện tiến trình liên tiếp của tập hoạt động trên.

2.2. Dạng toán học

Cho tập hoạt động $M = \{m_1, m_2, \dots, m_n\}$ với

$$m_i : \{U_i, [a_i, b_i], X_{m_i}, \Delta t_i\} \quad (i = 1, \dots, n).$$

Trong đó:

- U_i : độ ích lợi riêng của hoạt động i ,

- $[a_i, b_i]$: khoảng thời gian ưu tiên tiến hành hoạt động i ,
- $X_{m_i} = \{m_{k_1}^i, K, m_{k_i}^i\}$: tập các hoạt động xung đột với $m_i, m_{k_i}^i \in M$,
- Δt_i : khoảng thời gian để tiến hành hoạt động.

Từ tập M thiết lập kế hoạch để thực hiện tiến trình gồm các hoạt động kế tiếp.

Tổng độ ích lợi của kế hoạch tính theo:

$$U^* = \sum_{i=1}^n (U_i + A(a_i, b_i, T_i) - B(i)).$$

Trong đó:

- U^* : tổng độ ích lợi của kế hoạch,
- $A(a_i, b_i, T_i)$: độ ích lợi thêm (động) của hoạt động m_i với thời điểm bắt đầu T_i ,
- U_i : độ ích lợi riêng (tĩnh) của hoạt động m_i ,
- $T_i = \sum_{k=1}^p \Delta t^k$, với các Δt^k , $k = 1, \dots, p$ là các khoảng thời gian tiến hành các hoạt động xảy ra trước hoạt động m_i , trong kế hoạch đã cho,
- $B(i) = H_1(U_i + U_{i-1}) + H_2(U_i + U_{i-1})$ với

$$\begin{cases} H_1 = 0 & \text{nếu } m_{i-1} \in X_{m_i} \\ H_1 \neq 0 & \text{nếu } m_{i-1} \notin X_{m_i} \\ H_2 = 0 & \text{nếu } m_{i+1} \in X_{m_i} \\ H_2 \neq 0 & \text{nếu } m_{i+1} \notin X_{m_i} \end{cases}$$

3. PHƯƠNG HƯỚNG GIẢI QUYẾT MỚI

Phân tích các phương pháp được dùng phổ biến cho việc lập kế hoạch chúng ta nhận thấy các phương pháp dùng phỏng đoán và nhận định nhảy để đưa ra tiến trình [4, 8] là những cách thức có độ rủi ro lớn khi bị giới hạn thời gian hoạt động [13, 15]. Ngoài ra những kiểm chứng sau khi nhận định nhảy hoặc phỏng đoán đôi khi làm cho quá trình lập kế hoạch bị chậm đi đáng kể. Những phương pháp sử dụng hàm đánh giá [2, 14] thường không đủ mạnh trong những môi trường tồn tại nhiều bất định [3, 7]. Do đó, trong cách giải quyết được đưa ra, chúng tôi tạo một thuật toán kép. Tùy thuộc việc đánh giá tác hại của xung đột với toàn bộ ích lợi của kế hoạch, giải thuật phân lớp hoặc giải thuật chống xung đột sẽ là giải pháp chính và áp dụng trước. Qua đó, áp dụng đánh giá trực tiếp và loại các nhánh xấu ngay từ đầu, góp phần giảm thời gian tính toán để thỏa mãn ràng buộc về thời gian.

Chúng tôi xây dựng giải thuật phân lớp dựa trên lý thuyết đánh giá các khoảng mờ [5, 6, 10, 11, 12] để tạo ra các phân lớp ưu tiên và áp dụng thuật toán Gen để tránh xung đột. Nhờ việc kết hợp mềm dẻo hai giải thuật này theo phương pháp trình bày trong [16] số nhánh tính toán thực tế đã giảm bớt, và do vậy giảm thời gian tính toán một cách đáng kể. Phương pháp đã nêu và tiến trình của bài toán đã cho được mô tả qua sơ đồ hình 1.

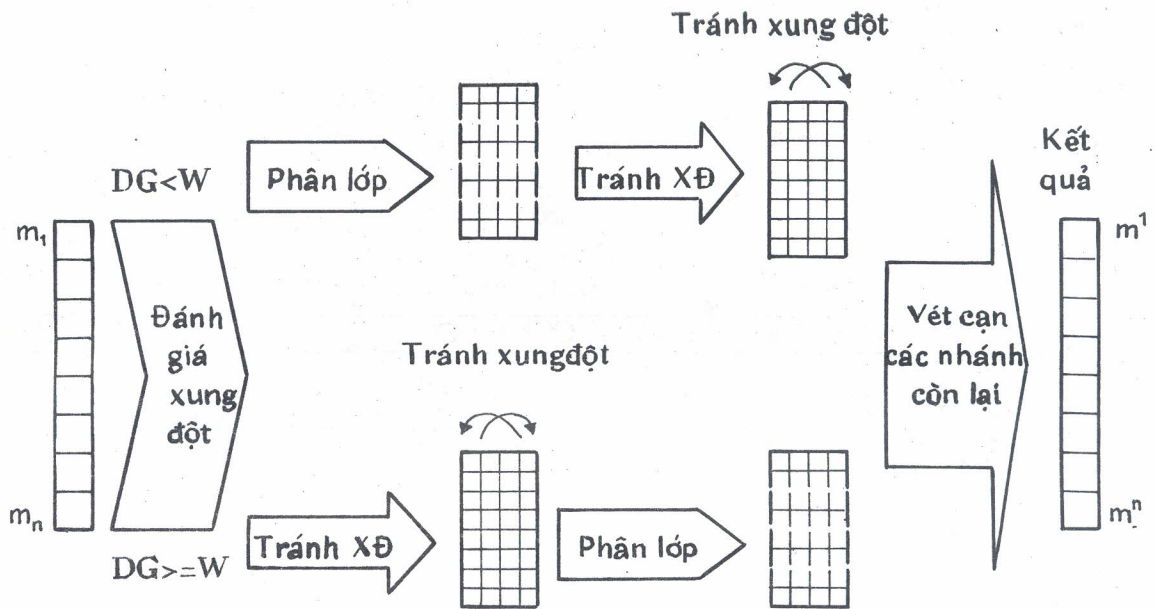
Như vậy phương án lập kế hoạch mà chúng tôi đề nghị có thể chia thành 3 giai đoạn như sau:

1. Đánh giá xung đột.
2. Phân lớp.
3. Tránh xung đột.

4. ĐÁNH GIÁ XUNG ĐỘT

Tạo hàm đánh giá mức xung đột như sau:

Tính trung bình cộng số các hoạt động xung đột tại mỗi hoạt động



Hình 1. Sơ đồ các bước của giải thuật

$$DG = \frac{1}{n} \sum_{i=1}^n FN(X_{m_i}),$$

trong đó: $FN(X_{m_i}) = [X_{m_i}]$ là số phần tử của tập X_{m_i} , n là số phần tử của tập các hoạt động.

Xét hai trường hợp:

1)
$$DG = \frac{1}{n} \sum_{i=1}^n FN(X_{m_i}) \geq w.$$

Áp dụng giải thuật chống xung đột trước, sau đó áp dụng phân lớp.

2)
$$DG = \frac{1}{n} \sum_{i=1}^n FN(X_{m_i}) < w.$$

Áp dụng phân lớp trước, sau đó áp dụng giải thuật chống xung đột.

Với w là ngưỡng đánh giá được cho trước và có thể được thay đổi tùy theo từng bài toán cụ thể.

Ngưỡng đánh giá thường được sử dụng là: $\geq 0,75$ ứng với mật độ xung đột lớn, $< 0,75$ ứng với mật độ xung đột không lớn.

Ta có thể khởi động w với giá trị 0,75, đánh giá và so sánh các kết quả U^* để tìm một ngưỡng đánh giá tốt hơn cho từng bài toán.

5. PHÂN LỚP

1. Khái niệm

Dựa vào tập hoạt động $M = \{m_1, m_2, \dots, m_m\}$:

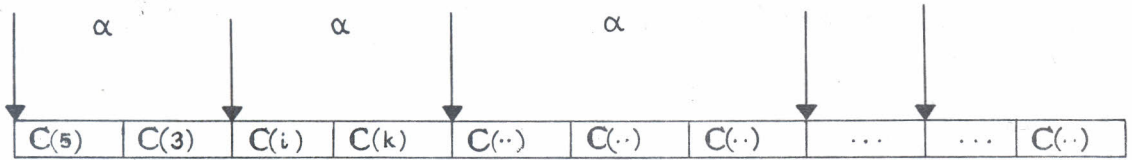
$$m_i : \{U_i, [a_i, b_i], X_{m_i}, \Delta t_i\} \quad (i = 1, \dots, n).$$

Lấy đoạn bất kỳ $[a^*, b^*]$ là một trong các khoảng $[a_i, b_i]$ ($i = 1, \dots, n$).

Tính $C(i)$ theo $[a_i, b_i]$ và $[a^*, b^*]$, trong đó $C(i)$ là hàm so sánh 2 khoảng $[a_i, b_i]$ và $[a^*, b^*]$ sử dụng cách đánh giá mờ mà sẽ được giải thích ở phần sau.

Như vậy ta có tập các giá trị so sánh $C = \{C(1), C(2), \dots, C(n)\}$.

Thực hiện phân lớp theo bước cắt α :



Ở đây: $\alpha = z \frac{C(n) - C(1)}{n}$, $z = \{1, 2, 3, \dots\}$ là hệ số chọn trước. Số lượng các lớp phụ thuộc vào số α .

Ta có cách cắt lớp như sau:

$$L_k = \{C(k) : \max C(k) - \min C(k) \leq \alpha \quad (i = 1, \dots, n) \text{ với } L_k \text{ là một lớp.}$$

2. Cơ sở để phân lớp

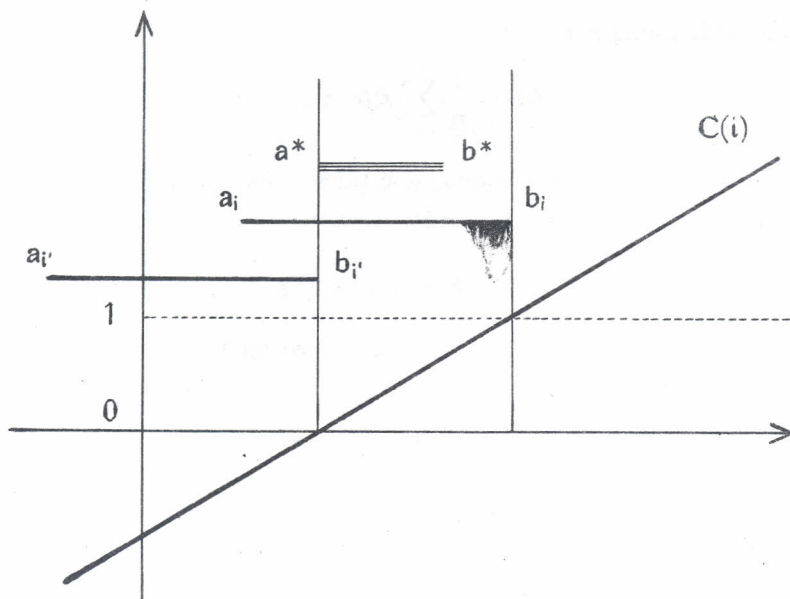
Các lớp được cắt ra dựa trên các giá trị so sánh. Cơ sở để tính các giá trị này là phương pháp so sánh mờ trong [9, 10, 12].

Áp dụng phương pháp so sánh mờ [9, 12].

Hàm so sánh $[a_i, b_i]$ với $[a^*, b^*]$ được tính như sau

$$C(i) = 2 \frac{[a^* - b_i]}{[\Delta^* + \Delta_i]}$$

Đồ thị thể hiện đồ biến thiên của độ so sánh theo phương pháp mờ phụ thuộc vào vị trí tương đối của $[a_i, b_i]$ với $[a^*, b^*]$ được biểu diễn trong hình 2.



Hình 2. Đồ thị $C(i)$

Với đồ thị trên ta nhận thấy:

- Khi $(a^* + b^*) = a_i + b_i$ thì $C(i) = 1$.
- Khi $a^* = b_i$ thì $C(i) = 0$.
- $C(i)$ biến thiên từ $-\infty$ đến $+\infty$.

6. TRÁNH XUNG ĐỘT

Bài toán lập kế hoạch đã nêu đặc trưng bởi hai điều kiện tối ưu hóa không phụ thuộc lẫn nhau là đạt độ ích lợi lớn nhất và xung đột nhỏ nhất.

Trong phần phân lớp, phương pháp mờ đã được dùng chỉ để đánh giá và lựa chọn các hoạt động theo thời gian, vấn đề giải quyết xung đột giữa các hoạt động tiến hành kế tiếp nhau chưa được đặt ra. Do vậy, trong quá trình hạn chế các xung đột, chúng ta không nên làm thay đổi quá nhiều trình tự thực hiện các hoạt động đã được chỉ ra sau bước phân lớp, nghĩa là phải đạt được sự cân bằng giữa tiến trình hoạt động tốt và số khả năng xung đột thấp.

Để đạt được sự cân bằng này, ta thiết lập một hàm đánh giá như sau:

$$G_r = aU_r - bV_r,$$

trong đó U là số đo độ thuận lợi của phương án P_r đã cho và V_r là số xung đột trong P_r . Như vậy G là hàm đánh giá chung và mục tiêu là sắp xếp các hoạt động ξ để G gần đến G_{\max} . Để đạt được mục đích này, chúng ta xây dựng một thuật toán biến thể từ thuật toán di truyền (thuật toán Gen - Genetic Algorithm) [1, 7].

Thuật toán Gen di truyền dựa theo các đặc tính di truyền của tự nhiên là sự kế thừa và tiến hóa (theo lý thuyết của Darwinna) của các cá thể để tồn tại và phát triển. Sử dụng thuật toán Gen cho phép tìm kiếm những khả năng tương đối tốt trong không gian tìm kiếm với chi phí thời gian không cao mà tránh được tối ưu cục bộ.

Đầu vào của bước tránh xung đột là một thứ tự các hoạt động sẽ được thực hiện ([hd[1], dh[2], ... hd[n]). Như vậy nếu coi mỗi phương án lập kế hoạch là một Gen thì chúng sẽ là dãy thứ tự của n bit (ở đây ta hiểu khái niệm bit là một đơn vị cơ bản mang thông tin) ứng với thứ tự thực hiện các hoạt động. Chúng ta nhận thấy là các giá trị có thể của mỗi bit là số tự nhiên từ 1 đến n và trong mỗi cá thể không tồn tại hai bit nào mang cùng một giá trị. Một Gen gọi là không xung đột nếu các bit biểu diễn các hoạt động không xung đột với nhau. Ví dụ trong trường hợp $n = 5$ ta có thể có Gen: 2 4 5 1 3 tương ứng với phương án $\{2, 4, 5, 1, 3\}$.

Áp dụng phương pháp thuật toán Gen đã nêu ta có sơ đồ giải thuật biến thể như sau:

Bước 1. Tạo lập tập các nghiệm (quần thể).

Ở bước này từ nghiệm ban đầu ta tạo ra một quần thể gồm pop-size cá thể (phương án) ban đầu (trường hợp đơn giản nhất là các cá thể tạo ra giống cá thể tương ứng với nghiệm ban đầu).

Bước 2. Lựa chọn các cá thể tốt theo tiêu chuẩn G_i .

Ta tìm những cá thể đáp ứng một tiêu chuẩn đã chỉ ra, tập hợp vào quần thể. Những cá thể khác sẽ bị đào thải. Có thể chọn theo phương án cá thể i được chọn nếu $G_i > G$ trung bình đồng thời tìm G_{\max} bằng cách lưu trữ giá trị G lớn nhất tại thời điểm đó.

Bước 3. Lai ghép.

Ta sẽ tiến hành tìm những cặp cá thể cho lai ghép với nhau tạo ra cặp cá thể mới thay thế cho cặp cá thể cũ. Quá trình lai ghép được thực hiện tại những bit theo phương pháp không làm thay đổi quá nhiều vị trí các bit.

Chọn từ 2 cá thể đem lai tạo những đoạn bit của chúng. Đó là những đoạn bit liên tục không chứa xung đột ở trong. Các đoạn được ghép nối nếu các bit ở phần ghép nối không xung đột với nhau. Trình tự thực hiện như sau: Giả sử bit đang xét là bit thứ k của cá thể được phân tích (k ban đầu = 1).

- (i) Nếu $k = 1$ hoặc bit $k - 1$ không thuộc dãy bit cố định (dãy bit không cần phải thay đổi vì không chứa xung đột) hoặc bit k không xung đột với bit $k - 1$ thì chọn k vào dãy các bit cố định.
- (ii) Nếu bit k xung đột với bit $k - 1$ và bit $k - 1$ thuộc dãy bit cố định thì bit k không thuộc dãy bit cố định.
- (iii) $k = k + 1$. Nếu $k \leq n$ thì quay lại (i).

Sự lai tạo có thể được tiến hành bằng cách tạo ra cá thể (gen) 1' qua sao chép các bit cố định của gen 1, còn mỗi bit k chưa được xác định của gen 1' sẽ lấy giá trị của bit k' của gen 2.

Tương tự ta cũng tạo được cá thể 2'. Với k' là vị trí của bit đầu tiên gặp trong gen 2 mà không có trong gen 1.

Thay thế gen 1 bằng gen 1' và gen 2 bằng gen 2'.

Bước 4. Đột biến.

Đây là quá trình ta sử dụng phương pháp gây đột biến nhằm tạo ra những cá thể có cấu tạo và tính chất mới khác biệt với cá thể khác.

Ta có thể sử dụng phương pháp gây đột biến sau:

Xác định một hệ số đột biến db ($0 \leq db \leq 100$). Dùng một hàm ngẫu nhiên $F(x)$ có mô tả sau

$$F(x) = 1 \text{ với khả năng } x\%$$

$$F(x) = 0 \text{ với khả năng } (100 - x)\%$$

Chỉ những bit có $F(db) = 1$ mới tham gia đột biến. Xác định tiếp một hệ số đột biến Gen dbg ($0 \leq dbg \leq 5$) nhằm tìm những cặp Gen có $F(dbg) = 1$ thì đổi giá trị chúng cho nhau.

Bước 5. Nếu số thế hệ chưa vượt qua ngưỡng dừng thì quay lại bước 2.

Các hệ số sử dụng trong thuật toán như: pop_size , db , dbg , $ngưỡng\ thế\ hệ$ cần được xác định qua thực nghiệm.

7. ĐỘ PHỨC TẠP THUẬT TOÁN VÀ MỘT SỐ KẾT QUẢ THÍ NGHIỆM

1. Độ phức tạp thuật toán

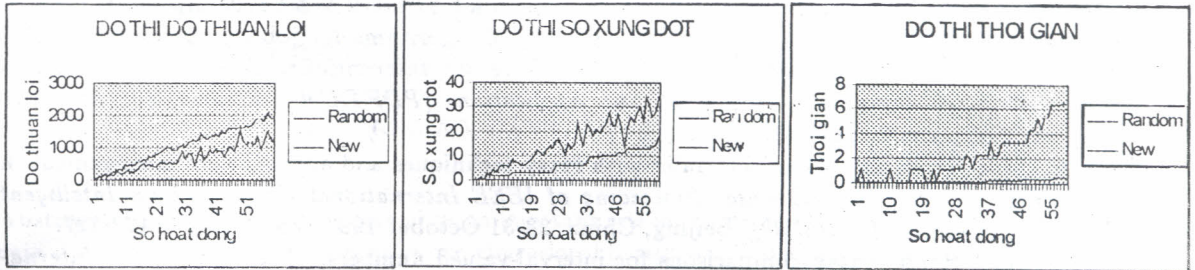
Độ phức tạp thuật toán của phương pháp như sau:

Công việc / Trường hợp	Xấu nhất	Trung bình	Tốt nhất
Sắp xếp	n^2	$n \log(n)$	$n \log(n)$
Phân lớp	n	n	n
Tránh xung đột	n^2	n^2	n^2
Duyệt các lớp	$n! * n$	$n * (n/2)!$	n
Tổng cộng	$n! * n + 2n^2 + n$	$n * (n/2)! + n^2 + n \log(n) + n$	$n^2 + n \log(n) + 2n$

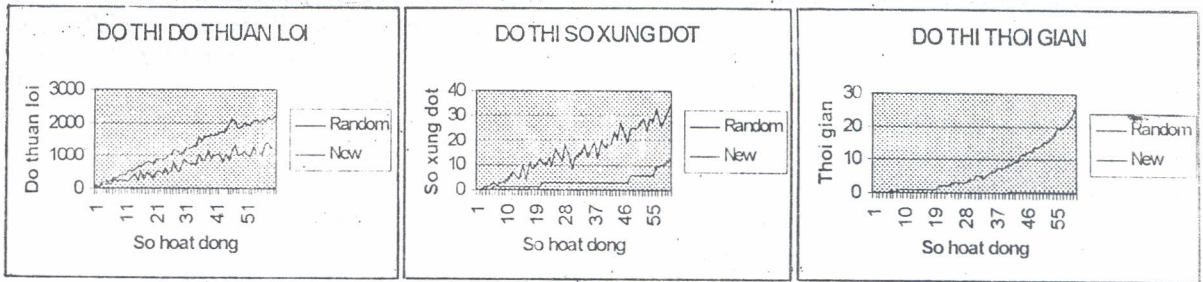
2. Một số kết quả thí nghiệm

Dưới đây là những đồ thị thể hiện kết quả thí nghiệm.

Trường hợp 1: Ngưỡng thể hệ = 20, số hoạt động = 60, $W = 0,75$, $DG \sim 0,5$



Trường hợp 2: Ngưỡng thể hệ = 80, số hoạt động = 60, $W = 0,75$, $DG \sim 0,5$



8. KẾT LUẬN

Trên đây chúng tôi đã trình bày một phương pháp mới lập kế hoạch trong môi trường bất định và chứa đựng xung đột. Phương pháp tạo lớp dựa vào so sánh mờ được kết hợp với thuật toán Gen trong một giải thuật kép nhằm đưa ra kế hoạch có độ thuận lợi cao nhất có thể. Thực nghiệm đã chứng tỏ phương pháp có thể tạo kế hoạch với độ tối ưu khá cao và thời gian tương đối ngắn.

Việc nghiên cứu phát triển hàm đánh giá mức độ xung đột nhằm tạo sự kết hợp tốt hơn nữa giữa quá trình tạo lớp và quá trình tránh xung đột sẽ là hướng phát triển trong tương lai của phương pháp.

TÀI LIỆU THAM KHẢO

- [1] David E. Goldvers, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [2] E. Yampratoom, Using Simulation-based to Plan in an Uncertain and Temporally Complex World, Brown University, Department of Computer Science Technical Report CS-91/41/1994.
- [3] J. S. Penberthy, "The Zeno Temporal Planer. Planing with Continuous Change", PhD thesis, University of Washington, 1993.
- [4] Michael P. Wellman and David V. Pynadath, Plan Recognition under Ucertainty, Air Force Office of Scientific Research, F49620-95-1-0365/1996.
- [5] N. Honda, F. Sugimoto, M. Tanaka, and S. Aida, *Decision Suport System Using Fuzzy Reasoning and Evolution*, Articial Intelligence in Economics and Managements, Elsevier Science Publisher B.V./1986.
- [6] Nguyễn Hải Châu, Ứng dụng luật điều khiển mờ trong quá trình quyết định đường đi trên mạng, *Tạp chí Tin học và Điều khiển học* 13 (1997) 21-35.
- [7] Nicholas Kushmerisk, Dan Weld, Steve Hanks, Adam Carlson, and Denise Draper, The Buridan Planer, University of Washington, 1993.

- [8] Oren Etzioni, Steve Hanks, Daniel Weld, Denise Fraper, Neal Lesh, and Mike Williamson, An Approach to planing with incomplete information (extended abstract), Department of Computer Science and Engineering, RS-35 University of Washington, Seattle, WA 98195/April 20, 1992.
- [9] Pham Hong Hanh and Simonenko V., Adaptation of algorithms for job-resource assigment in heterogeneous distributed systems, *Proceeding of the 3rd International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'96)*, Sunnyvale, USA, 9-11 August, 1996, Vol. 2, 825-846.
- [10] Pham Hong Hanh, Intelligent scheduler with fuzzy techniques and evolutionary management in heterogeneous computing system, *Proceeding of IEEE International Conference on Intelligent Processing Systems (ICIPS'97)*, Beijing, China, 28-31 October 1997, Vol. 1 (1997) 122-127.
- [11] Pham Hong Hanh, Fuzzy comparisons for interval-valued numbers, *Proceeding of 6th International Conference on Fuzzy Theory and Technology (Ft and T'98)*, North Caroline, USA, 24-28 October 1998, 147-151.
- [12] Pham Hong Hanh, Nguyen Quoc Anh, Fuzzy complete continuous comparisons for planing action with incomplete information in custon management, *Proceeding of the Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications*, Ha Long, Vietnam, September 1998, 209-218.
- [13] Thomas L. Dean, Lloy Greenwald, and Leslie Pack Kaelbling, Time-Critical Planing and Scheduling Research at Brown Iniversity, Brown Unversity, Department of Computer Science Technical Report CS-91/41/1994.
- [14] Thomas Wagner and Alan Garvey, Leveraging Uncertainty in Design to Criteria Scheduling, Computer Science Department University of Massachusetts and Computer Science Department Pacific Lutheran University.
- [15] Vũ Ngọc Phàn, Các hệ thống tự động: Bất định và điều khiển, *Tạp chí Tin học và Điều khiển học* **13** (1997) 11-24.
- [16] Zbigniew Michalewicz, *Genetic Algorithms + Data Structures = Evolutionprogram*, Third, Revised and Extended Edition, Springer, 1996.

Nhận bài ngày 9-6-1999

Nhận lại sau khi sửa ngày 10-7-2000

Khoa Công nghệ thông tin,
Trường Đại học Khoa học tự nhiên - ĐHQG Hà Nội.