Thuật toán Ro-CS giải bài toán lập lịch với tài nguyên giới hạn

Nguyễn Thế Lộc[1], Đặng Quốc Hữu[2]
[1] Trường Đại học Sư phạm Hà Nội, Việt Nam
[2] Trường Đại học Thương mại, Hà Nội, Việt Nam
Tác giả liên hệ: Đặng Quốc Hữu, huudq@tmu.edu.vn
Ngày nhận bài: 01/05/2022, ngày sửa chữa: 01/06/2022, ngày duyệt đăng: 30/06/2022
Định danh DOI: 10.32913/mic-ict-research-vn.v2022.n1.1055

Tóm tắt: Bài báo đề xuất phương pháp tìm lời giải cho Bài toán MS-RCPSP (Multi Skill-Resource Constrained Project Scheduling Problem). MS-RCPSP đã được chứng minh là bài toán NP-Khó, do vậy cần sử dụng các phương pháp tính toán tiến hóa, cận tối ưu nhằm tìm được lời giải phù hợp trong thời gian chấp nhận được. Thuật toán đề xuất là thuật toán lai ghép giữa thuật toán Cuckoo Search (CS) và kỹ thuật Rotate giúp mở rộng không gian tìm kiếm sau mỗi thế hệ tiến hóa, nhằm tăng khả năng tìm được lời giải tốt hơn. Thuật toán mới gọi là Ro-CS có khả năng áp dụng để tìm lời giải cho bài toán MS-RCPSP. Để kiểm chứng thuật toán Ro-CS, bài báo tiến hành thực nghiệm trên bộ dữ liệu chuẩn iMOPSE, kết quả thực nghiệm được tổng hợp, đánh giá, so sánh, phân tích cho thấy tính hiệu quả của thuật toán đề xuất.

Từ khóa: *Tính toán tiến hóa, thuật toán cận tối ưu, bài toán MS-RCPSP.*

Title: **The Ro-CS Algorithm Solving the MS-RCPSP Problem**

Abstract: This paper proposes a method to solve the MS-RCPSP problem (Multi Skill-Resource Constrained Project Scheduling Problem). MS-RCPSP is an NP-Difficult, so it is necessary to use evolutionary, optimization methods to find a suitable solution in an acceptable time. The proposed algorithm is a hybrid be-tween Cuckoo Search (CS) algorithm and Rotate technique to expand the investi-gation space after each evolutionary generation, to increase the possibility of find-ing a better solution. The new algorithm called Ro-CS is suitable to find a feasible solution for the MS-RCPSP. The proposed algorithm has been experimented on the iMOPSE standard dataset, and the results are synthesized, evaluated, com-pared, and analyzed to denote the effectiveness of the proposed algorithm.

Keywords: *Evolutionary computing, optimization, MS-RCPSP problem.*

I. GIỚI THIỆU

Bài toán lập lịch thực hiện dự án với tài nguyên giới hạn RCPSP (Resource Con-strained Project Scheduling Problem)[1, 2], là bài toán đặt ra mục tiêu tìm phương án lịch biểu để bố trí các tài nguyên cho trước thực hiện và hoàn thành mọi công việc (tác vụ) trong một dự án với thời gian ngắn nhất hoặc chi phí thấp nhất hoặc cân bằng cả hai yếu tố thời gian và chi phí. Thông thường, số lượng tác vụ cần thực hiện của dự án lớn hơn nhiều so với số lượng tài nguyên có thể sử dụng. Bài toán MS-RCPSP (Multi Skill - RCPSP)[1–4] mở rộng từ RCPSP bằng việc bổ sung thêm ràng buộc mới, gần với các dự án trong thực tế hơn, đó là: mỗi tài nguyên có thể có nhiều kỹ năng khác nhau và mỗi kỹ năng có mức/bậc kỹ năng nhất định. Mỗi tác vụ sẽ có yêu cầu về tài nguyên thực hiện cần đáp ứng loại kỹ năng cụ thể và mức kỹ năng nhất định. Khi đó, các tài nguyên có kỹ năng đúng theo yêu cầu của tác vụ và mức kỹ năng lớn hơn hoặc bằng so với yêu cầu sẽ có năng lực để thực hiện tác vụ. MS-RCPSP đã được chứng minh thuộc lớp NP-Khó nên không thể tìm ra lời giải tối ưu trong thời gian đa thức, thay vì đó, nhiều nghiên cứu đã tập trung vào việc tìm các phương pháp giải tiến hóa, cận tối ưu để tìm gia lời giải chấp nhận được và đủ tốt trong thời gian phù hợp.

Phần tiếp theo của bài báo có cấu trúc như sau: Phần II giới thiệu một số công trình nghiên cứu có liên quan đến bài toán MS-RCPSP. Phần III mô tả phát biểu toán học của bài toán. Phần IV trình bày thuật toán đề xuất Ro-CS để tìm lời giải cận tối ưu cho bài toán MS-RCPSP. Phần V mô tả các thực nghiệm để kiểm chứng thuật toán, phân tính, đánh giá chất lượng lời giải. Phần VI tóm tắt những kết quả chính của bài báo và hướng nghiên cứu sẽ tiến hành trong tương lai.

II. NHỮNG CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Bài toán lập lịch với tài nguyên giới hạn và đa kỹ năng MS-RCPSP có nhiều ứng dụng trong khoa học và thực tiễn nên đã được nghiên cứu rộng rãi bởi nhiều nhà khoa học và đã có nhiều công bố khoa học liên quan đến bài toán này. Myszkowski và cộng sự [5] có nhiều công bố về bài toán này, nhóm tác giả đã đề xuất các phương pháp tìm lời giải cho bài toán MS-RCPSP dựa trên các thuật toán như: thuật toán Đàn kiến [4]; thuật toán GA [5, 14] và một số thuật toán khác. Một đóng góp quan trọng của Myszkowski và cộng sự là đã xây dựng và công bố bộ dữ liệu chuẩn iMOPSE [5, 6] sử dụng cho để thực nghiệm các thuật toán cho bài toán này. Hosseinian [7] và cộng sự cũng có nhiều nghiên cứu về bài toán MS-RCPSP. Năm 2018, Hosseinian và Baradaran công bố kết quả nghiên cứu [7] về bài toán Multi-mode Multi-skilled Resource-Constrained Project Scheduling Problem (MMSRCPSP), một biến thể của bài toán MS-RCPSP trong đó bổ sung thêm ràng buộc rằng mỗi tác vụ chỉ có thể được thực thi ở một vài chế độ (mode) định trước và không thể thay đổi mode một khi quá trình thực thi đã được bắt đầu. Để giải quyết bài toán, Hosseinian và cộng sự đã sử dụng thuật toán GA cổ điển kết hợp với phương pháp ra quyết định dựa trên độ đo thông tin Shanon-entropy để lựa chọn cá thể cho thế hệ kế tiếp. Năm 2019 nhóm tác giả này công bố bài báo [8] trong đó sử dụng thuật toán Dandelion Algorithm để giải quyết bài toán MS-RCPSP, trong bài này nhóm tác giả lựa chọn thực nghiệm trên iMOPSE [5, 6, 14].

Javanmard và cộng sự sử dụng 2 thuật toán tiến hóa truyền thống là GA và PSO để lên phương án lịch biểu cho các tài nguyên đa kỹ năng (trên thực tế là các công nhân và kỹ sư) thực hiện một dự án trong ngành hóa chất với mục tiêu tối thiểu hóa tổng chi phí của dự án [9]. Hai thuật toán đề xuất được kiểm chứng trên bộ dữ liệu PSPLIB, vốn không hoàn toàn thích hợp với những bài toán lập lịch có mục tiêu là tối thiểu hóa chi phí.

H. Davari-Ardakani [10] đề xuất một biến thể đa mục tiêu của bài toán MS-RCPSP nhằm tối thiểu hóa hai đại lượng là thời gian và chi phí thực hiện của dự án. Dừng lại ở việc phát biểu bài toán, H. Davari-Ardakani và đồng nghiệp [11] chưa đề xuất được giải pháp mới mà chỉ tiến hành thực nghiệm với những giải pháp đã có như Max-min, vì vậy bài toán có thể coi là vẫn còn để mở.

III. PHÁT BIỂU BÀI TOÁN

Là bài toán mở rộng từ bài toán RCPSP, MS-RCPSP [1–3, 12] bổ sung thêm ràng buộc về kỹ năng của tài nguyên thực hiện: mỗi tài nguyên có nhiều kỹ năng (skill) khác nhau và mỗi kỹ năng có một mức/bậc (level) cụ thể. Để thực hiện được tác vụ, tài nguyên cần đáp ứng về kỹ năng và mức kỹ năng theo yêu cầu của tác vụ. Các ràng buộc của bài toán quy định, tài nguyên có cùng loại kỹ năng và có mức kỹ năng lớn hơn hoặc bằng mức kỹ năng mà tác vụ yêu cầu thì có thể thực hiện được tác vụ đó.

Phát biểu toán học của bài toán MS-RCPSP

Các ký hiệu:

- C_i: các tác vụ cần thực hiện và hoàn thành trước tác vụ i;
- S: tập kỹ năng của tất cả các tài nguyên; S^i: tập các kỹ năng của tài nguyên i, $S^i \subseteq S$;
- S_i: kỹ năng thứ i;
- t_j: thời gian thực hiện tác vụ j;
- L: các tài nguyên có khả năng tái sử dụng được dùng trong dự án;
- L^k: các tài nguyên đáp ứng yêu cầu của tác vụ k; $L^k \subseteq L$;
- L_i: tài nguyên thứ i;
- W: các tác vụ cần thực hiện của dự án;
- W^k: tập tác vụ phù hợp với năng lực của tài nguyên k , $W^k \subseteq W$;
- W_i: tác vụ thứ i ;
- r^i: tập kỹ năng được yêu cầu để thực hiện tác vụ i;
- B_k, E_k: thời điểm bắt đầu và kết thúc tác vụ k ;
- $A_(u,v)^t$: có giá trị bằng 1 cho biết tài nguyên v; đang thực hiện tác vụ u tại thời điểm t và ngược lại;
- h_i: mức kỹ năng của kỹ năng (skill) i;
- g_i: loại kỹ năng i; m: makespan của cá thể (lịch biểu);
- P: lịch biểu thỏa mãn các ràng buộc của bài toán;
- $P_(all)$: tập mọi lịch biểu;
- $f(P)$: hàm mục tiêu, trả về makespan của lịch biểu P;
- z: số lượng tài nguyên có thể sử dụng của dự án;
- n: số lượng tác vụ cần thực hiện của dự án;

Lời giải của bài toán MS-RCPSP làm tìm ra một lịch biểu P để:

$$f(P) \rightarrow min \tag{1}$$

Một lịch biểu đúng cần đáp ứng các ràng buộc sau:

$$S^k \neq \emptyset \quad \forall L_k \in L \tag{2}$$

$$t_j \geq 0 \quad \forall W_j \in W \tag{3}$$

$$E_j \geq 0 \quad \forall W_j \in W \tag{4}$$

$$E_i \leq E_j - t_j \quad \forall W_j \in W, j \neq 1, W_i \in C_j \tag{5}$$

$$\forall W_i \in W^k \;\; \exists S_q \in S^k : g_(S_q) = g_(r^i) \; v \; h_(S_q) \geq h_(r^i) \tag{6}$$

$$\forall L_k \in L, \forall q \in m : \sum_{i=1}^{n} A_{i,k}^q \leq 1 \tag{7}$$

$$\forall W_j \in W \exists ! q \in [0-m], !L_k \in L : A_{j,k}^q = 1; với \; A_{j,k}^q \in \{0;1\} \tag{8}$$

Các ràng buộc về kỹ năng của tài nguyên được yêu cầu để thực hiện một tác vụ được mô tả dưới dạng một ma trận như ví dụ được minh họa trong Hình 1 dưới đây. Trong Hình 1 , ký hiệu: $S_(i.j)$ nghĩa là kỹ năng S_i và mức kỹ năng j.

IV. THUẬT TOÁN ĐỀ XUẤT

Thuật toán 1: fRotate.

```
Dữ liệu vào: currentBest (các thể tốt nhất sau mỗi thế hệ)..
Dữ liệu ra: lịch biểu sau khi được hiệu chỉnh.
begin
    makespan = f(currentBest);
    newbest = currentBest;
    L_b ← lastResource (newbest); //tài nguyên hoàn thành sau cùng
    W_b ← {các task vụ được thực hiện bởi L_b};
    for i = size(W_b^R) downto 1 do
        freepos ← { Vị trí tài nguyên rỗi };
        if freepos > 0 then
            newbest = { ;
              W_i thực hiện tại vị trí freepos;
              Tính toán lại vị trí thực hiện của các task khác;
            }
            newmakespan = f(newbest); // tính toán lại makespan
            if newmakespan < makespan then
                makespan = newmakespan;
                return newbest; // dịch chuyển thành công, trả về kết quả và dừng hàm
            end
        end
    end
    return newbest;
end
```

Là bài toán thuộc lớp NP-Khó, để tìm lời giải cho MS-RCPSP, cần sử dụng các phương pháp tìm lời giải gần đúng. Ro-CS là lai giữa phương pháp gần đúng Cuckoo Search và kỹ thuật Rotate.

1. Thuật toán Cuckoo Search (CS)

Thuật toán Cuckoo Search(CS) [12, 13] được Yang và Deb đưa ra vào năm 2009, đến nay CS đã được nhiều nhà khoa học nghiên cứu và áp dụng cho các bài toán tối ưu khác nhau, trong đó có bài toán MS-RCPSP. Thuật toán này được phát triển dựa trên đặc tính sinh hoạt của loài chim Cuckoo và hành vi bay của một số loài chim và ruồi giấm - gọi là Lévy flight.

CS phù hợp với các bài toán tối ưu tính toán trên dữ liệu lớn bằng cách tạo ra sự cân bằng trong các kỹ thuật tìm kiếm nghiệm toàn cục và tìm kiếm cục bộ nhằm tạo ra các nghiệm tốt hơn ở thế hệ tiếp theo. Mục tiêu của thuật toán là sinh ra những cá thể mới có tiềm năng tốt hơn thay

Thuật toán 2: Ro-CS.

```
Dữ liệu vào: dataset, t_max : số thế hệ tiến hóa
Dữ liệu ra: cá thể tốt nhất và makespan
begin
    Load and Valid dataset;
    t = 0;
    P_all = Initial population;
    f = {Calculate the fitness, makespan, bestnest};
    while t < t_max do
        P_new = {Tạo cá thể mới bằng Lévy Flight} ( theo công thức 10);
        P_i = {Lấy ngẫu nhiên một cá thể từ P_all };
        if f(P_new) < f(Pi) then
            P_i = P_new;
        end
        P^pa = pa% tổ kém nhất từ P_all;
        for j = 1 to size(P^pa) do
            P_new = Tạo cá thể mới bằng Lévy Flight(theo công thức 9);
            if f(P_new) < f(P_j^pa) then
                P_j^pa = P_new;
            end
        end
        P_all = {Thay thế P^pa vào P_all};
        f = {Calculate the fitness, bestnest};
        bestnest = fRotate(bestnest);
        makespan = f(bestnest);
        t = t+1;
    end
    return {bestnest, makespan};
end
```

thế các cá thể (nghiệm) có chất lượng kém. Việc tạo ra cá thể mới áp dụng kỹ thuật Lévy Flight với các phép Tìm kiếm toàn cục (Global Search) và Tìm kiếm cục bộ (Local Search).

Thế hệ tiếp theo tạo ra bằng phép tìm kiếm cục bộ, áp dụng công thức 9:

$$x_i^{t+1} = x_i^t + \alpha s \bigotimes H(pa - e) \bigotimes (x_j^t - x_k^t) \qquad (9)$$

Thế hệ tiếp theo tạo ra bằng phép tìm kiếm toàn cục, áp dụng công thức 10:

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \qquad (10)$$

với:

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, s >> s_0 > 0 \qquad (11)$$

trong đó:

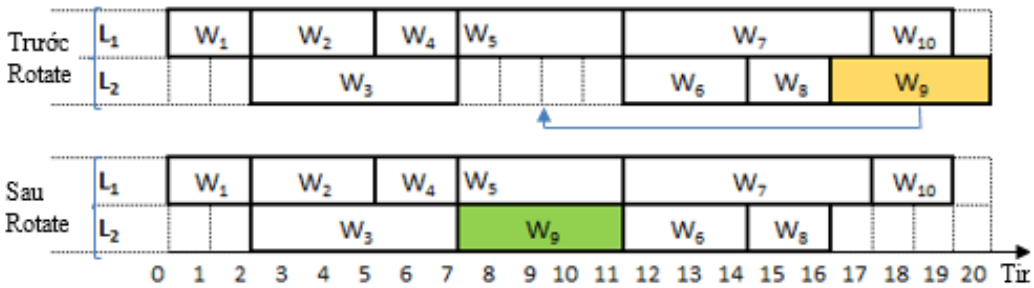
- λ: là một hằng số, $0 < \lambda \leq 2$
- $\Gamma(\lambda)$: là hàm trả lại giá trị (λ-1)!

Thuật toán CS thực hiện bằng cách tiến hóa quần thể qua nhiều thế hệ khác nhau. Ở mỗi thế hệ, CS thực hiện tạo một cá thể mới bằng kỹ thuật tìm kiếm toàn cục (Glob-al Search). Sau khi tính toán, tiến hóa, thuật toán sẽ thay thế một số cá thể kém nhất bằng các cá thể mới, các cá thể

✓ Có thể thiết lập
× Không thể thiết lập

Tài nguyên		Tác vụ			
		$S_{2.2}$	$S_{3.1}$	$S_{2.2}$	$S_{1.1}$
		W_1	W_2	W_3	W_4
$S_{1.3}, S_{2.2}$	L_1	✓	×	✓	✓
$S_{2.1}, S_{3.2}$	L_2	×	✓	×	×
$S_{1.2}, S_{2.1}$	L_3	×	×	×	✓
$S_{2.2}, S_{3.3}$	L_4	✓	✓	✓	×

Hình 1. Tài nguyên thực hiện tác vụ.



Hình 2. Biểu đồ Gantt khi áp dụng phương pháp Rotate.

mới tạo ra bằng kỹ thuật tìm kiếm cục bộ (Local Search). Kết quả của thuật toán là một cá thể (lịch biểu) với thời gian thực hiện tối thiểu.

2. Kỹ thuật Rotate

Rotate là kỹ thuật xoay vòng nhằm hạn chế tối đa khoảng thời gian rỗi của tài nguyên, dẫn đến giảm tổng thời gian thực hiện dự án. Các bước thực hiện cụ thể như sau:

- **Bước 1**: Tìm tài nguyên sử dụng nhiều nhất trong Cbest, gọi là Lb. Đây là tài nguyên kết thúc công việc sau cùng trong các tài nguyên sử dụng để thực hiện dự án.
- **Bước 2**: Duyệt các tác vụ được thực hiện bởi Lb theo thứ tự ngược với thứ tự thực hiện từ phải qua trái trên tài nguyên, tức là tác vụ nào thực hiện sau sẽ xem xét trước.
  W_b^R: tập tác vụ vụ đang được thực hiện bởi L_b
- **Bước 3**: Với mỗi tác vụ $W_i \in W_b^R (i = sizeof(W_b^R) downto 1)$, thực hiện:
  - *Bước 3.1*: Tìm khoảng thời gian rỗi trên tài nguyên
  - *Bước 3.2*: Chuyển thứ tự thực hiện của tác vụ hiện tại về vị trí của tài nguyên rỗi nếu không vi phạm ràng buộc về thứ tự ưu tiên.
  - *Bước 3.3*: Điều chỉnh thời gian bắt đầu và kết thúc thực hiện của các tác vụ có vị trí thực hiện sau vị trí mới của tác vụ hiện tại
  - *Bước 3.4*: Tính toán lại makespan của dự án, nếu tạo ra cá thể tốt hơn thì dừng thuật toán.

Về bản chất, phương pháp Rotate là cách tái sắp xếp thứ tự thực hiện các tác vụ trên tài nguyên hoàn thành công việc sau cùng mà không vi phạm các ràng buộc về thứ tự ưu tiên giữa các tác vụ, đồng thời hạn chế tối đa thời gian tài nguyên rỗi. Việc này giúp giảm tổng thời gian thực hiện dự án đáng kể.

Hình 2 là ví dụ về một dự án với 10 tác vụ và 2 tài nguyên. Trước khi áp dụng kỹ thuật Rotate, thời gian thực hiện tác vụ là 20, sau khi áp dụng kỹ thuật Roate, thời gian thực hiện dự án giảm xuống còn 10.

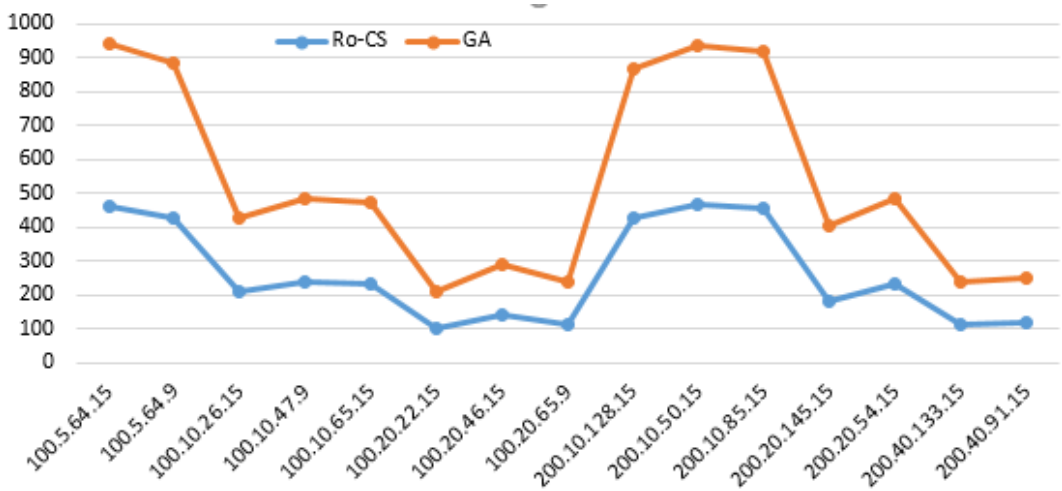
Phương pháp Rotate được trình bày chi tiết trong **Thuật toán 1**.

với:

- f: hàm tính makespan của một lịch biểu
- size: hàm tính số phần tử của một tập/mảng
- lastResource: hàm trả về tài nguyên kết thúc thực hiện sau cùng

Bảng I
KẾT QUẢ THỰC NGHIỆM TRÊN BỘ DỮ LIỆU IMOPSE

Dataset	GA			Ro-CS		
	Best	**Avg**	**Std**	**Best**	**Avg**	**Std**
100.5.64.15	478	487	8.2	462	467	2.5
100.5.64.9	453	457	3.2	430	434	2.1
100.10.26.15	221	224	2.1	209	213	1.9
100.10.47.9	247	253	5.9	237	240	1.3
100.10.65.15	240	244	3.2	233	236	2.2
100.20.22.15	109	115	5.2	101	109	5.2
100.20.46.15	145	150	4.1	143	152	7.2
100.20.65.9	124	131	6.5	115	127	6.3
200.10.128.15	440	446	5.7	428	435	4.1
200.10.50.15	468	471	2.9	465	467	0.5
200.10.85.15	465	469	4	456	458	0.3
200.20.145.15	222	229	6.1	183	192	3.8
200.20.54.15	249	255	5.1	235	240	2.7
200.40.133.15	126	134	7.4	115	127	9
200.40.91.15	133	139	5.8	119	128	7.1



Hình 3. So sánh kết quả thực nghiệm của giá trị BEST.

3. Thuật toán Ro-CS

Áp dụng kỹ thuật Rotate vào thuật toán CS ta được thuật toán 2 Ro-CS dưới đây.

với:

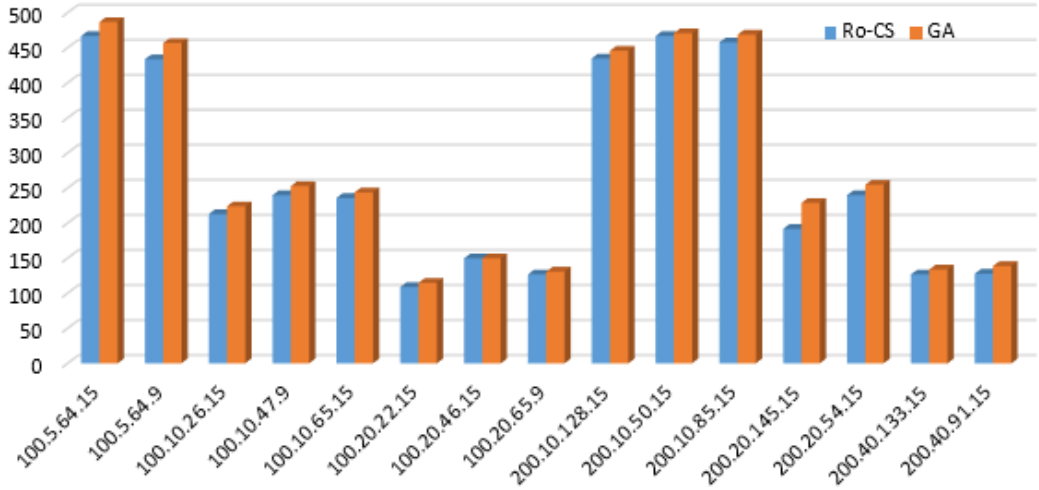
- f: hàm tính makespan của một lịch biểu
- fRotate: là hàm áp dụng kỹ thuật Rotate.

V. KẾT QUẢ THỰC NGHIỆM

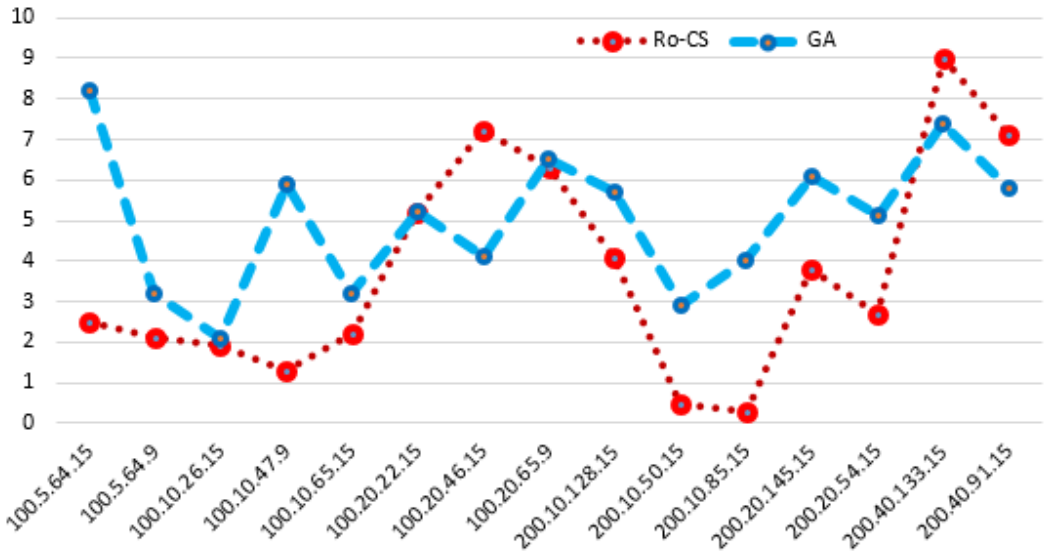
Để kiểm chứng thuật toán đề xuất Ro-CS, bài báo tiến hành thực nghiệm trên bộ dữ liệu iMOPSE [5,6,14] (là bộ dữ liệu chuẩn quốc tế được phát triển riêng cho bài toán MS-RCPSP. Bộ dữ liệu iMOPSE được thể hiện như trong Bảng II dưới đây.

Trong đó:

- Tasks: là số tác vụ cần thực hiện của dự án.
- Resources: là số tài nguyên có thể sử dụng để thực hiện các tác vụ, các tài nguyên này có khả năng tái sử dụng, nghĩa là sau khi hoàn thành một tác vụ, có thể sử dụng tài nguyên đó để thực hiện tác vụ khác.
- Precedence Relations: là tổng số ràng buộc ưu tiên giữa các tác vụ trong bài toán, ràng buộc ưu tiên chỉ ra thứ tự thực hiện của các tác vụ.
- Skills: số kỹ năng của tài nguyên thực hiện, mỗi tài



Hình 4. So sánh kết quả thực nghiệm của giá trị AVG.



Hình 5. So sánh kết quả thực nghiệm của giá trị STD.

nguyên có thể có nhiều kỹ năng khác nhau, mỗi kỹ năng có một mức kỹ năng cụ thể.

1. Tham số thực nghiệm

Thực nghiệm Ro-CS được cài đặt với các tham số đầu vào như sau:

- Dữ liệu: 15 file dữ liệu trong bộ dữ liệu iMOPSE như trong Bảng I.
- Khởi tạo quần thể ban đầu với 100 cá thể;
- Số thế hệ tiến hóa: 50.000;
- Số lần chạy thực nghiệm trên mỗi bộ dữ liệu: 25;
- Môi trường thực nghiệm: Thực nghiệm được tiến hành trên môi trường Matlab 2014
- Máy tính thực nghiệm: CPU Intel Core i5, tốc độ 2.2 GHz, RAM 6GB, hệ điều hành Windows 10.

2. Tham số thực nghiệm

Kết quả thực nghiệm với bộ dữ liệu iMOPSE [5,6] được trình bày trong Bảng I.

Trong Bảng I, dữ liệu cột GA được lấy từ phần thực nghiệm của nhóm tác giải Myszkowsky với công cụ GARunner [14] do chính nhóm tác giải công bố.

Các thực nghiệm được triển khai trên 02 thuật toán GA và Ro-CS sử dụng bộ dữ liệu iMOPSE. Bài báo đã tiến hành thực nghiệm các thuật toán một cách độc lập với các tham số đầu vào cài đặt như trong mục v. Kết quả thực

Bảng II
BỘ DỮ LIỆU IMOPSE

Name	Tasks	Resources	Precedence Relations	Skills
100.5.64.15	100	5	64	15
100.5.64.9	100	5	64	9
100.10.26.15	100	10	26	15
100.10.47.9	100	10	47	9
100.10.65.15	100	10	65	15
100.20.22.15	100	20	22	15
100.20.46.15	100	20	46	15
100.20.65.9	100	20	65	9
200.10.128.15	200	10	128	15
200.10.50.15	200	10	50	15
200.10.85.15	200	10	85	15
200.20.145.15	200	20	145	15
200.20.54.15	200	20	54	9
200.40.133.15	200	40	133	15
200.40.91.15	200	40	91	9

nghiệm được trình bày trong Bảng II chỉ ra thuật toán Ro-CS có chất lượng lời giải tốt hơn thuật toán GA trong mọi trường hợp. Cụ thể:

- Với giá trị tốt nhất (BEST) Ro-CS tốt hơn GA từ 0.6% đến 17.6%. So sánh chi tiết được thể hiện trong Hình 3.
- Với giá trị trung bình (AVG) Ro-CS tốt hơn GA từ 0% đến 16.2%. So sánh chi tiết được thể hiện trong Hình 4.
- Tổng độ lệch chuẩn của GA 75.4, của Ro-CS 56.2, điều này cho thấy Ro-CS hoạt động ổn định hơn. So sánh chi tiết được thể hiện trong Hình 5.

Như vậy, thuật toán mới Ro-CS với việc bổ sung kỹ thuật Rotate mang lại quả tốt hơn so với GA.

VI. KẾT LUẬN

Bài báo này đã trình bày phát biểu toán học cụ thể về bài toán MS-RCPSP thông qua các ràng buộc, đây là bài toán có nhiều ứng dụng trong khoa học và thực tế. Để tìm lời giải cho bài toán MS-RCPSP, bài báo đã đề xuất một thuật toán mới là Ro-CS, đây là thuật toán lai giữa thuật toán di truyền và kỹ thuật Rotate nhằm mở rộng không gian tìm kiếm sau mỗi thế hệ tiến hóa. Để kiểm chứng tính hiệu quả của thuật toán đề xuất, bài báo đã tiến hành thực nghiệm trên bộ dữ liệu iMOPSE (bộ dữ liệu chuẩn được phát triển để tiến hành các thực nghiệm riêng cho bài toán MS-RCPSP). Kết quả thực nghiệm đã được tổng hợp, so sánh với GA trên các giá trị tốt nhất (BEST), giá trị bình quân (AVG) và giá trị độ lệch chuẩn (STD). Thông qua việc so sánh cho thấy, Ro-CS mang lại hiệu quả tốt hơn GA với cùng bài toán và cùng bộ dữ liệu thực nghiệm, cụ thể: giá trị BEST tốt hơn từ 0.6% đến 17.6%, giá trị AVG tốt hơn đến 16.2 và tổng STD tốt hơn 19.2.

Ngoài việc sử dụng tìm lời giải cho bài toán MS-RCPSP, thuật toán Ro-CS còn có khả năng áp dụng tốt cho các bài toán lập lịch khác. Trong thời gian tới, tác giả sẽ tiếp tục nghiên cứu và cải tiến thuật toán dựa trên các phương pháp gần đúng khác, sử dụng bước di chuyển ngẫu nhiên dựa trên xác suất Gauss, Cauchy, ... nhằm nâng cao chất lượng lời giải.

TÀI LIỆU THAM KHẢO

[1] A. Christian, S. Demassey, E. Néron, “Resource Constrained Project Scheduling: Models, Algorithms, Extensions and Applications”, *ISBN 978-1-84821-034-9*, 2008.

[2] R. Klein: “Scheduling of Resource-Constrained Projects”, *Springer Science & Business Media*, Vol. 10, 2012.

[3] Huu Dang Quoc, Loc Nguyen The, Cuong Nguyen Doan, "Naixue Xiong: Effective Evolutionary Algorithm for Solving the Real-Resource-Constrained Scheduling Problem," *Journal of Advanced Transportation*, vol. 2020, Article ID 8897710, 11 pages, 2020

[4] P.B. Myszkowski, M. Skowroński, L. Olech, K. Oślizło, "Hybrid Ant Colony Optimization in solving Multi–Skill Resource–Constrained Project Scheduling Problem", *Soft Computing Journal*, Volume 19, Issue 12, pp.3599–3619, 2015.

[5] P.B. Myszkowski, M. Laszczyk, I. Nikulin, M. Skowro, “iMOPSE: a library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem”, *Soft Computing Journal*, 23: 32397, 2019.

[6] P.B. Myszkowski, M.E. Skowronski, K.Sikora, “A new benchmark dataset for Multi-Skill Resource-Constrained Project Scheduling Problem”, *Computer Science and Information Systems*, ACSIS, Vol. 5, pp. 129–138, 2015. DOI: 10.15439/2015F273.

[7] A.H. Hosseinian, V. Baradaran, "An Evolutionary Algorithm Based on a Hybrid Multi-Attribute Decision Making Method for the Multi-Mode Multi-Skilled Resource-Constrained Project Scheduling Problem.", *Journal of Optimization in Industrial Engineering*, 12.2, pp. 155-178,2019.

[8] A.H. Hosseinian, V. Baradaran, "Detecting communities of workforces for the multi-skill resource-constrained project scheduling problem: A dandelion solution approach.", *Journal of Industrial and Systems Engineering*, pp. 72-99, 12.2019.

[9] S. Javanmard, B. Afshar-Nadjafi, S.T. Niaki, "Preemptive multi-skilled resource investment project scheduling problem: Mathematical modeling and solution approaches.", *Computers & Chemical Engineering*, 96, pp. 55-68, 2017.

[10] H. Najafzad, H. Davari-Ardakani, R. Nemati-Lafmejani, "Multi-skill project scheduling problem under time-of-use electricity tariffs and shift differential payments.", *Energy Journal*, vol. 168, pp. 619-636, Elsevier,2019.

[11] Huu Dang Quoc, Loc Nguyen The, Cuong Nguyen Doan, Toan Phan Thanh, Naixue Xiong, “Intelligent Differential Evolution Scheme for Network Resources in IoT”, *Scientific Programming (IF:1.28, Q3)*, Volume 2020, Article ID 8860384 | 12, 2020. DOI: 10.1155/2020/8860384

[12] X.S. Yang, “Nature-Inspired Metaheuristic Algorithms”, *Luniver Press*, ISBN-13: 978-1-905986-28-6, 2010.

[13] X.S. Yang, S. Deb, "Cuckoo search via Lévy flights", *Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, USA, pp. 210-214, 2009.
[14] Bộ dữ liệu iMOPSE và công cụ GARunner: http://imopse.ii.pwr.wroc.pl/download.html

SƠ LƯỢC VỀ TÁC GIẢ

Nguyễn Thế Lộc

Tốt nghiệp đại học khoa Toán Tin, đại học Sư phạm Hà Nội năm 1993, tốt nghiệp thạc sĩ CNTT tại trường đại học Bách khoa Hà Nội, nhận bằng tiến sỹ tại Viện Nghiên cứu Khoa học Công nghệ Nhật Bản JAIST năm 2007. Hiện đang công tác tại trường Đại học Sư phạm Hà Nội.
Lĩnh vực nghiên cứu: mạng máy tính và truyền thông, xử lý song song và phân tán.
Điện thoại : 0988.765.837.
E-mail: locnt@hnue.edu.vn

Đặng Quốc Hữu

Tốt nghiệp đại học và thạc sĩ khoa CNTT, Đại học Quốc Gia Hà Nội năm 2006 và 2015, nhận bằng Tiến sĩ tại Viện Khoa học và Công nghệ Quân sự năm 2022. Hiện đang công tác tại đại học Thương Mại.
Lĩnh vực nghiên cứu: tính toán tối ưu, tính toán tiến hóa, xử lý song song và phân tán.
Điện thoại : 0988710727.
E-mail: huudq@tmu.edu.vn