

THUẬT TOÁN TIẾN HÓA GIẢI BÀI TOÁN LẬP LỊCH DỰ ÁN VỚI TÀI NGUYÊN GIỚI HẠN (MS-RCPSP) VÀ ỨNG DỤNG TRONG VIỆC LẬP KẾ HOẠCH SẢN XUẤT THÔNG MINH

TS. Đặng Quốc Hữu^{1*}, ThS. Huỳnh Hoàng Tân², ThS. Nguyễn Tài Tiệp²

¹Trường Đại học Thương Mại

¹Trường Đại học Công nghệ Đồng Nai

*Tác giả liên hệ: Đặng Quốc Hữu, huudq@tmu.edu.vn

THÔNG TIN CHUNG

Ngày nhận bài: 31/05/2023

Ngày nhận bài sửa: 14/06/2023

Ngày duyệt đăng: 24/06/2023

TỪ KHÓA

Bài toán MS-RCPSP;

Tính toán tối ưu;

Thuật toán tiến hóa.

TÓM TẮT

Bài báo trình bày phương pháp lập kế hoạch sản xuất thông minh bằng việc ứng dụng bài toán lập lịch thực hiện dự án với tài nguyên giới hạn MS-RCPSP đa kỹ năng (Multi Skill-Resource Constrained Project Scheduling Problem). Bài toán này đã được chứng minh là bài toán thuộc lớp bài toán kho (NP-Hard), do vậy không thể tìm được lời giải chính xác trong thời gian đa thức mà cần sử dụng một số giải thuật cận tối ưu để tìm nghiệm tốt nhất trong khoảng thời gian cho phép. Bài báo này nghiên cứu và đề xuất phương pháp giải bài toán MS-RCPSP dựa trên thuật toán Memetic kết hợp với phương pháp tái thiết lập tài nguyên (Reassignment), thuật toán mới gọi là MEM-RES, có khả năng tìm kiếm được các nghiệm tốt hơn sau mỗi thế hệ tiến hóa. Bài báo thực hiện chạy thử nghiệm thuật toán mới với bộ dữ liệu sản xuất thực tế chuyên may công nghiệp TNG. Kết quả cho thấy, thuật toán đề xuất phù hợp với bài toán và đạt được hiệu quả cao. Lời giải của bài toán này là một lịch biểu thể hiện việc sắp xếp các tài nguyên thực hiện từng công việc của dự án theo thời gian cho đến khi dự án hoàn thành. Do vậy, lời giải có thể áp dụng được trong việc lập kế hoạch sản xuất tự động thay cho cách thức lập kế hoạch truyền thống như hiện nay trong các nhà máy sản xuất, đặc biệt với mô hình sản xuất theo dây chuyền.

ABSTRACT

This paper introduces an intelligent production planning approach that addresses the challenges of scheduling project execution with limited resources in the MS-RCPSP (Multi Skill-Resource Constrained Project Scheduling Problem). Due to its classification as an NP-Difficult problem, finding an exact solution within a polynomial time frame is impossible. However, the paper proposes a method called MEM-RES, which combines the Memetic algorithm with the Reassignment method to find approximate solutions efficiently. The algorithm iteratively improves the solutions with each evolutionary generation. The proposed algorithm has experimented using actual production data from the TNG industrial sewing line. The results demonstrate the algorithm's suitability for the problem and its ability to achieve high efficiency. The solution obtained through this approach provides a schedule that organizes resources for executing each project's tasks over time until finished. Consequently, this solution can be made the automatic production planning, replacing the traditional methods currently employed in production factories, especially those utilizing a production line model.

1. GIỚI THIỆU

Hiện nay, trong các doanh nghiệp sản xuất, hầu hết việc lập kế hoạch sản xuất vẫn được tiến hành theo kinh nghiệm hoặc thủ công dẫn đến nhiều hạn chế trong việc tính toán định lượng cụ thể cũng như phân bổ nhân công, tài nguyên, cơ sở vật chất trong quá trình sản xuất. Trong lĩnh vực may công nghiệp, hoạt động sản xuất thành phẩm may mặc được thực hiện qua nhiều bước theo quy trình của chuyền may công nghiệp. Để sản xuất ra một thành phẩm, chuyền may được thực hiện qua nhiều công đoạn, có những sản phẩm lên đến vài trăm công đoạn khác nhau. Với cách điều phối thủ công, sẽ khó tối ưu nên thời gian hoàn thành các đơn hàng thường bị kéo dài.

Bài báo này sẽ trình bày việc áp dụng bài toán MS-RCPSP (Klein, 2012; Huu Dang Quoc L. N., 2022; Huu Dang Quoc L. N., 2020) trong việc lập kế hoạch điều phối sản xuất tự động dựa trên việc tìm ra các phương án lịch biểu để sắp xếp các tài nguyên (công nhân) thực hiện các công việc phù hợp với trình độ, kỹ năng của tài nguyên đảm bảo thời gian thực hiện dự án (hợp đồng) là tối thiểu. Trong thực tế sản xuất, mỗi dự án có nhiều công việc (tác vụ) cần thực hiện và số lượng tài nguyên thực hiện thường ít hơn nhiều số tác vụ cần thực hiện, các tài nguyên cũng có các kỹ năng và mức kỹ năng khác nhau, do vậy, cần phải có các phương án bố trí tài nguyên thực hiện các tác vụ phù hợp để nâng cao hiệu quả sản xuất. MS-RCPSP là bài toán thuộc lớp NP-Hard, do vậy không thể tìm được nghiệm trong thời gian đa thức mà cần sử dụng một số phương pháp tìm nghiệm gần đúng để tìm lời giải phù hợp. Một trong các ràng buộc quan trọng của bài toán này là mỗi tác vụ yêu cầu tài nguyên thực hiện cần đáp ứng đúng loại kỹ năng và mức kỹ năng nhất định, theo đó, một tài nguyên chỉ có thể thực hiện được tác vụ nếu có loại kỹ năng

phù hợp và mức kỹ năng lớn hơn hoặc bằng mức kỹ năng mà tác vụ yêu cầu. Để tìm lời giải tốt cho bài toán MS-RCPSP, bài báo đề xuất thuật toán MEM-RES, đây là thuật toán được phát triển từ giải thuật Memetic kết hợp với phương pháp tái thiết lập tài nguyên (Reassignment). Lời giải của bài toán MS-RCPSP khi áp dụng MEM-RES là một phương án lịch biểu chỉ ra việc bố trí tài nguyên thực hiện tác vụ của dự án. Lịch biểu này có thể dùng trong điều phối nhân sự thực hiện các dự án trong thực tế sản xuất, giúp giảm thiểu việc lập kế hoạch sản xuất thủ công.

Đóng góp chính của bài báo gồm:

- Đề xuất một metaheuristic để tìm lời giải cho bài toán MS-RCPSP. Thuật toán mới gọi là MEM-RES, đây là thuật toán lai giữa Memetic và phương pháp tái thiết lập tài nguyên thực hiện dự án.
- Đề xuất phương pháp số hóa dữ liệu sản xuất thực tế, áp dụng cho dữ liệu chuyền may công nghiệp của công ty TNG, dữ liệu số hóa được chuyển thành các bộ dữ liệu chuẩn, phù hợp với đầu vào của bài toán MS-RCPSP.
- Tiến hành thực nghiệm thuật toán đề xuất với bộ dữ liệu thực tế được số hóa từ dữ liệu chuyền may công nghiệp TNG. Dữ liệu thực nghiệm được tổng hợp và so sánh với dữ liệu thực tế về vận hành sản xuất và so sánh với thuật toán khác. Do vậy, có thể sử dụng để phục vụ cho việc lập kế hoạch điều phối sản xuất thông minh trong chuyền may công nghiệp gắn với các hợp đồng sản xuất thực tế.

Bài báo được trình bày tiếp theo với cấu trúc như sau:

- Mục 2 giới thiệu về những nghiên cứu của trước đây của các nhà khoa học về bài toán MS-RCPSP và thuật toán để tìm lời giải cho bài toán này. Phần này cũng giới thiệu về thuật toán Memetic, là thuật toán cơ sở để

phát triển thuật toán giải được đề xuất trong Mục 4 của bài báo.

- Mục 3 trình bày về các ký hiệu toán học được sử dụng để trình bày các ràng buộc của bài toán MS-RCPSP, sau đó phát biểu bài toán bằng các ràng buộc toán học cụ thể.
- Mục 4 giới thiệu về thuật toán chính của bài báo này để giải bài toán MS-RCPSP, phương pháp giải đề xuất được phát triển từ thuật toán Memetic kết hợp với phương pháp tái thiết lập tài nguyên sau mỗi thế hệ tiến hóa.
- Mục 5 giới thiệu về phương pháp số hóa dữ liệu thực tế và áp dụng cho dữ liệu chuyển may công nghiệp. Kết quả số hóa tạo ra các bộ dữ liệu chuẩn, làm đầu vào cho việc chạy thực nghiệm thuật toán đề xuất.
- Mục 6 của bài báo tổng hợp lại nội dung chính của bài và đề ra một số hướng nghiên cứu tiếp theo.

2. NHỮNG CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

MS-RCPSP (Klein, 2012; Huu Dang Quoc L. N., 2022; Huu Dang Quoc L. N., 2020) là một lớp con của bài toán lập lịch trình dự án với tài nguyên giới hạn (RCPSP: Resource-Constrained Project Scheduling Problem) (Klein, 2012; Saad, Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling, 2021) và đã được chứng minh thuộc lớp NP-Hard, nghĩa là không thể tìm được lời giải trong thời gian đa thức. Do đó, cần sử dụng các thuật toán tiến hóa, cận tối ưu để giải bài toán này. Các thuật toán tiến hóa có khả năng tìm được nghiệm tốt cho lớp bài toán khó trong thời gian chấp nhận được, các thuật toán này thường thực hiện bằng việc tạo quần thể và tính toán tiến hóa quần thể qua nhiều thế hệ khác nhau nhằm tìm ra các cá thể tốt, trong quá trình tiến hóa, thuật toán sẽ đánh giá và ghi nhận các giá trị liên

quan của thế hệ trước, và kế thừa cho các thế hệ tiếp theo.

2.1. Các phương pháp giải bài toán lập lịch thực hiện dự án với tài nguyên giới hạn

Nhiều nhà khoa học đã nghiên cứu và công bố rộng rãi các phương pháp khác nhau để tìm ra lời giải cho các bài toán RCPSP và MS-RCPSP. Các phương pháp này, chủ yếu dựa trên các thuật toán tính toán tiến hóa như: thuật toán di truyền (GA: Genetic Algorithm) (Asadujjaman, 2021; Dakhli, 2020; J. Lin, 2020; Saad, Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling, 2021), thuật toán tối ưu hóa bầy đàn (PSO: Particle Swarm Optimization) (Ansari, 2021; Huu Dang Quoc L. N., 2022), thuật toán tiến hóa vi phân (DE: Differential Evolution) (Bhat, 2021; Castro-Gutierrez, 2021; Emary, 2020), thuật toán tìm kiếm chim Cuckoo (CS: Cuckoo Search) (Yang X.S, 2009; X.S., 2010),... Bảng 1 tóm tắt một số công trình tiêu biểu đã được công bố trước đây.

Nhiều nhà khoa học đã nghiên cứu bài toán MS-RCPSP và đề xuất các phương pháp giải khác nhau. Trong (Dakhli, 2020), tác giả đã đề xuất thuật toán đa mục tiêu kết hợp giữa GA với tối ưu hóa Pareto để tìm lời giải, tác giả cũng sử dụng phương pháp tìm kiếm cục bộ để mở rộng không gian tìm kiếm. Curi et al. (Curi, 2020) đã giới thiệu một phương pháp kết hợp thuật toán GA với nhiều heuristic khác nhau trong thuật toán MOGA-MH để tìm lời giải cho MS-RCPSP. Thuật toán đề xuất MOGA-MH được cải tiến bằng cách áp dụng kỹ thuật tìm kiếm vùng lân cận để tạo quần thể ban đầu tốt hơn và bốn kỹ thuật khác để tìm kiếm các lời giải phù hợp. Trong (Xu, 2020), tác giả đã trình bày thuật toán lai giữa GA và phương pháp cấp phát lại tài nguyên gọi là HGA-RR.

Bảng 1. Tổng hợp một số nghiên cứu về họ bài toán RCPSP

TT.	Thuật toán	Bài toán	Các tài liệu tham khảo
1	GA	RCPSP	(Saad, Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling, 2021; R. Kolisch, 1997; S. Sanaei, 2020)
2		NPV-RCPSP	(Asadujjaman, 2021)
3	PSO	MS-RCPSP	(J. Lin, 2020; Dakhli, 2020; Curi, 2020; Xu, 2020; Zhou, 2022)
4		RCPSP	(J. Cui, 2020; Stiti, 2019)
5		MS-RCPSP	(Gholami, 2020; Gholami, 2020; Ansari, 2021)
6	DE	RCPSP	(J. Wu, 2018; Y. Li, 2020)
7		MS-RCPSP	(Rostami, 2020; Emary, 2020; Bhat, 2021; Castro-Gutierrez, 2021)
8	CS	MS-RCPSP	(Maghsoudlou, 2017)

Một số tác giả nghiên cứu giải bài toán MS-RCPSP đa mục tiêu (tối ưu về chi phí và thời gian thực hiện dự án). Tác giả (Pang, 2021) đã nghiên cứu và đề xuất thuật toán MOEA-AEA, đây là thuật toán tìm kiếm đa mục tiêu dựa trên GA kết hợp với kỹ thuật lưu trữ thích nghi có ưu tiên nhằm tăng tính đa dạng của quần thể, từ đó nâng cao hiệu quả lời giải. Trong nghiên cứu (Zhou, 2022), các tác giả đã sử dụng lần lượt ba phương pháp để tìm lời giải đa mục tiêu: đầu tiên, quần thể được tạo ra bằng cách sử dụng phương pháp ưu tiên, tiếp theo, nhóm tác giả ứng dụng thuật toán NSGA-II nâng cao khả năng tìm kiếm lời giải, cuối cùng, các tác giả áp dụng kỹ thuật tìm kiếm cục bộ nhằm nâng cao hiệu quả của lời giải.

Thuật toán GA lai ghép với một số phương pháp khác để tìm lời giải cho họ bài toán RCPSP như tìm kiếm cục bộ, sử dụng các toán tử ưu tiên trong khởi tạo quần thể,... Trong (Asadujjaman, 2021), các tác giả sử dụng GA tích hợp với giải thuật Immune để giải bài toán NPV-Based RCPSP. Thuật toán sử dụng hai kỹ thuật bổ sung để nâng cao hiệu quả: tìm kiếm cục bộ và “forward-backward”. Trong (Saad, Quantum-Inspired Genetic Algorithm for Resource-

Constrained Project-Scheduling, 2021), các tác giả giới thiệu thuật toán QIGA được phát triển từ GA. Trong thuật toán này, các tác giả đề xuất phương pháp khởi tạo quần thể và cập nhật quần thể bằng kỹ thuật “quantum gates” và “superposition”, thuật toán đã được kiểm thử với bộ dữ liệu PSLIB (R. Kolisch, 1997). Các tác giả (S. Sanaei, 2020) cũng sử dụng GA kết hợp với kỹ thuật tìm kiếm cục bộ để giải bài toán MS-RCPSP.

Một trong các phương pháp hiệu quả để giải bài toán MS-RCPSP là sử dụng thuật toán tối ưu bầy đàn PSO (Particle Swarm Optimization). Trong (Ansari, 2021), tác giả đã đề xuất thuật toán lai giữa PSO và GA, trong đó trọng số quán tính của PSO được cập nhật động để cải thiện khả năng tìm kiếm. PSO cũng được nhóm tác giả (Stiti, 2019) sử dụng để tìm lời giải cho bài toán RCPSP, trong đó, tác giả đã đề xuất kỹ thuật “Valid Particle Generator” để phát hiện các lời giải tốt, đồng thời áp dụng các phương pháp thích nghi để cải thiện chất lượng của thuật toán. Các tác giả (J. Cui, 2020) nghiên cứu và đề xuất thuật toán PSO kết hợp với tìm kiếm cục bộ để giải bài toán RCPSP, thuật toán có tên HQPSO mang lại hiệu quả tốt. Các tác giả trong

(Gholami, 2020) đã giới thiệu thuật toán đa mục tiêu phát triển từ PSO, cải tiến được thực hiện để hiệu chỉnh hệ số quán tính bằng phương pháp thích nghi và sử dụng toán tử đột biến chuyên biệt để tăng cường hiệu quả tìm kiếm lời giải. Trong (Rathore, 2020), thuật toán PSO đã được cải thiện để tìm lời giải cho bài toán MS-RCPSP bằng cách sửa đổi hàm cập nhật vị trí cá thể kết hợp một toán tử đột biến riêng.

Ngoài GA và PSO, thuật toán Tiến hóa vi phân DE (Differential Evolution) cũng thường được nghiên cứu và sử dụng để giải các bài toán lập lịch với tài nguyên giới hạn. Đây là một giải thuật tiến hóa dựa trên thông tin định hướng để tạo ra các cá thể ở thế hệ tiếp theo bằng cách sử dụng các toán tử lai ghép, đột biến và chọn lọc. Đặc trưng của thuật toán DE là sử dụng thông tin định hướng trong toán tử đột biến để làm đột biến các cá thể của quần thể hiện tại. Trong bài báo (Y. Li, 2020), các tác giả đã trình bày thuật toán tiến hóa vi phân cải tiến (IDEA) để giải bài toán RCPSP. Thuật toán đề xuất được cải tiến qua hai giai đoạn: tạo quần thể định hướng ban đầu và điều chỉnh phương pháp chọn lọc cá thể để tăng tính hội tụ. Trong công trình (J. Wu, 2018), các tác giả tập trung vào việc giải quyết chiến lược đột biến riêng lẻ để cải thiện khả năng khám phá không gian nghiệm và cải tiến khả năng hội tụ của thuật toán. Tác giả đã giới thiệu thuật toán mới gọi là NBOLDE, việc cải tiến được thực hiện dựa việc sử dụng toán tử đột biến lân cận và phương pháp đánh giá và học tập kinh nghiệm của cá thể đối lập. Ngoài ra, NBOLDE cũng đưa vào phương pháp tính toán các tham số và các trọng số để hỗ trợ việc tìm kiếm lân cận, trong quá trình đột biến để tìm lời giải tốt hơn, tác giả phát triển một kỹ thuật đột biến gọi là “DE/current to best”, thuật toán đã được kiểm chứng và cho thấy hiệu quả tốt. Trong (Bhat, 2021), tác giả đã đề xuất nâng cao hiệu quả của quá trình tiến hóa bằng cách sử dụng các hệ số động áp dụng trong quá trình đột biến của cá thể. Trong (Emary, 2020), các trọng số quán tính được cập nhật động để cải thiện quá trình tìm kiếm. Trong (Rostami, 2020), các tác giả đề xuất

một thuật toán lai kết hợp DE với thuật toán Bee (BA). Trong công trình (Castro-Gutierrez, 2021), DE được sử dụng kết hợp với đột biến Gradient Descent (DEGD) để nâng cao hiệu quả lời giải.

Thông thường, các thuật toán đề xuất sẽ được chạy thực nghiệm để kiểm thử với các bộ dữ liệu chuẩn và đối chiếu so sánh với các thuật toán khác nhằm làm nổi bật tính hiệu quả. Hai bộ dữ liệu thường được sử dụng trong việc thực nghiệm các thuật toán để giải các bài toán họ RCPSP là PSLIB (R. Kolisch, 1997) và iMOPSE (Myszkowski P. B., 2022; P.B. Myszkowski, 2019).

2.2. Thuật toán Memetic

Memetic (MA: Memetic Algorithm) (Afsar, 2022; Seo, 2022; Wilson, 2022) là thuật toán tối ưu hóa kết hợp giữa phương pháp tiến hóa và tìm kiếm cục bộ để giải các bài toán khó. Thuật toán MA có khả năng khắc phục được một số hạn chế của thuật toán tiến hóa truyền thống bằng cách lưu và sử dụng “tri thức” của các thế hệ tiến hóa trước đó để định hướng quá trình tìm kiếm cá thể tốt ở các thế hệ tiếp theo.

Trong thuật toán Memetic, mỗi cá thể là một lời giải của bài toán, nó thỏa mãn các ràng buộc mà bài toán đưa ra. Để tạo ra một cá thể, mỗi thế hệ tiến hóa sẽ thực hiện các bước: chọn lọc, lai ghép và đột biến. Các bước tính toán của MA tương tự như GA, tuy nhiên, điểm cải tiến của MA là bổ sung phương pháp tìm kiếm cục bộ (Local Search) để hiệu chỉnh các cá thể sau các bước tính toán. Phương pháp tìm kiếm cục bộ có khả năng mở rộng không gian tìm kiếm sang các cá thể lân cận với cá thể hiện tại nên có thể tìm và tạo ra được các cá thể tốt hơn, điều này giúp thuật toán hội tụ nhanh hơn và chất lượng giải pháp được cải thiện so với thuật toán GA truyền thống.

Thuật toán Memetic thực hiện các bước sau:

- *Bước 1. Khởi tạo (Population Initialization):* Tạo một quần thể ban đầu với một số lượng cá thể xác định.
- *Bước 2. Tính toán và đánh giá:* tính toán giá trị của mỗi cá thể trong quần thể dựa trên hàm mục tiêu của bài toán.

- **Bước 3. Chọn lọc (Selection):** chọn các cá thể phù hợp để lai ghép nhằm tạo ra các cá thể mới ở thế hệ tiếp theo.
- **Bước 4. Lai ghép (Crossover):** kết hợp các cá thể trước để tạo ra cá thể mới. Quá trình này sẽ sử dụng thông tin của nhiều cá thể tham gia lai ghép để tạo ra cá thể mới, cá thể mới thừa kế tính chất của các cá thể tạo ra nó.
- **Bước 5. Đột biến (Mutation):** bổ sung thêm các thuộc tính ngẫu nhiên vào cá thể mới sau khi lai ghép.
- **Bước 6. Tìm kiếm cục bộ (Local search):** thực hiện các tính toán với các cá thể các lân cận để tìm kiếm cá thể tốt hơn.
- **Bước 7. Ghi nhận (Acceptance):** Ghi nhận cá thể của thế hệ tiến hóa hiện tại nếu nó tốt hơn thế hệ trước đó.
- **Bước 8. Kết thúc (Termination):** Kết thúc thuật toán nếu tìm được lời giải thỏa đáng

hoặc đáp ứng tiêu chí dừng của thuật toán.

- Lặp lại các bước từ 2 đến 8 để tạo thế hệ tiếp theo.

3. PHÁT BIỂU BÀI TOÁN MS-RCPSP

MS-RCPSP là lớp bài toán con của bài toán toán RCPSP (Resource Constrained Project Scheduling Problem) (Asadujjaman, 2021; Huu Dang Quoc L. N., 2022; Huu Dang Quoc L. N., 2020; Klein, 2012). Sự khác biệt của bài toán này là đưa thêm ràng buộc mới phù hợp hơn với các dự án trong thực tế, đó là ràng buộc về kỹ năng của tài nguyên thực hiện. Theo đó, tài nguyên thực hiện dự án có thể có nhiều loại kỹ năng (skill), mỗi kỹ năng có một mức trình độ/bậc kỹ năng (level) cụ thể. Để thực hiện được một tác vụ, tài nguyên cần đáp ứng về loại kỹ năng và mức kỹ năng lớn hơn hoặc bằng mức kỹ năng mà tác vụ yêu cầu.

Bảng 2. Các ký hiệu sử dụng để phát biểu bài toán

C_i	tập tác vụ (task) cần thực hiện trước tác vụ
S	tập tất cả các kỹ năng của 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	tập các tài nguyên
L^k	tập tài nguyên có thể thực hiện tác vụ k ; $L^k \subseteq L$
L_i	tài nguyên thứ i
W	tập các tác vụ của dự án
W^k	tập các tác vụ có thể thực hiện bởi tài nguyên k , $W^k \subseteq W$
W_i	tác vụ thứ i
r^i	tập các kỹ năng được yêu cầu để thực hiện tác vụ i . Một tài nguyên có thể thực hiện được tác vụ i nếu có kỹ năng giống với kỹ năng yêu cầu của tác vụ i và có mức kỹ năng lớn hơn hoặc bằng mức kỹ năng yêu cầu
B_k, E_k	thời gian bắt đầu và thời gian kết thúc thực hiện tác vụ k
$A_{u,v}^t$	biến xác định tài nguyên v có đang thực hiện tác vụ u tại thời điểm t hay không; 1: có, 0: không
h_i	mức kỹ năng của kỹ năng (skill) i
g_i	loại kỹ năng i
m	makespan của lịch biểu
P	một lịch biểu khả thi của bài toán
P_{all}	tập tất cả các lịch biểu
$f(P)$:	hàm mục tiêu, để tính makespan của P
n	số tác vụ
z	số tài nguyên

Dựa trên các ký hiệu trong Bảng 2, bài toán MS-RCPSP được phát biểu dưới dạng các ràng buộc toán học như sau.

Mục tiêu của bài toán là tìm ra một lịch biểu(cá thể) P để:

$$f(P) \rightarrow \min$$

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

$$- S^k \neq \emptyset \quad \forall L_k \in L \quad (1)$$

$$- t_j \geq 0 \quad \forall W_j \in W \quad (2)$$

$$- E_j \geq 0 \quad \forall W_j \in W \quad (3)$$

$$- E_i \leq E_j - t_j \quad \forall W_j \in W, j \neq i, W_i \in C_j \quad (4)$$

$$- \forall W_i \in W^k \exists S_q \in S^k : g_{S_q} = g_{r,i} \text{ và } h_{S_q} \geq h_{r,i} \quad (5)$$

$$- \forall L_k \in L, \forall q \in m : \sum_{i=1}^n A_{i,k}^q \leq 1 \quad (6)$$

$$- \forall W_j \in W \exists ! q \in [0, m], ! L_k \in L : A_{j,k}^q = 1; \text{ với } A_{j,k}^q \in \{0; 1\} \quad (7)$$

Mô tả các ràng buộc:

- Ràng buộc (1): mỗi tài nguyên phải có tối thiểu một kỹ năng nào đó.
- Ràng buộc (2,3): thời gian thực hiện của tác vụ bất kỳ tối thiểu phải bằng 0 (thực tế thì mọi tác vụ thật, có thời gian thực hiện tối thiểu luôn > 0 , trường hợp bằng 0 để minh họa cho 2 tác vụ giả thể hiện thời điểm bắt đầu và kết thúc của dự án)
- Ràng buộc (4): tác vụ cha (task i) phải kết thúc trước thời điểm tác vụ con (task j) bắt đầu. Thời điểm tác vụ i kết thúc ký hiệu là E_i , thời điểm tác vụ con j bắt đầu là $E_j - t_j$ (thời điểm kết thúc trừ đi thời gian thực hiện).
- Ràng buộc (5) nghĩa là: với mọi tác vụ $i \in W^k$ (tập các tác vụ mà tài nguyên k thực hiện được), luôn tồn tại kỹ năng $S \in S^k$ (tập skill của tài nguyên k) sao cho $g_S = g_{S,i}$: loại skill của r trùng với loại kỹ năng của L_i mà

tác vụ i yêu cầu. $h_{S_q} \geq h_{r,i}$: mức kỹ năng của tài nguyên thực hiện cao hơn hoặc bằng mức kỹ năng yêu cầu.

- Ràng buộc (6): tại mỗi thời điểm (q), mỗi tài nguyên chỉ được thực hiện tối đa một tác vụ. Nếu $\sum_{i=1}^n A_{i,k}^q = 0$ thì tài nguyên k không được gán cho tác vụ nào, nếu $\sum_{i=1}^n A_{i,k}^q = 1$ thì tài nguyên k được gán cho một tác vụ duy nhất.
- Ràng buộc (7): mỗi tác vụ chỉ được gán cho một tài nguyên duy nhất và chỉ được thực hiện bởi một tài nguyên duy nhất.

Trong bài toán MS-RCPSP, mỗi tác vụ có thêm các yêu cầu về kỹ năng (skill) của tài nguyên cần để thực hiện (như trong Bảng 3), mỗi tài nguyên có thể có nhiều kỹ năng khác nhau và mỗi kỹ năng có một mức kỹ năng cụ thể (như trong Bảng 4). Để thực hiện một tác vụ, tài nguyên cần có kỹ năng giống với kỹ năng mà tác vụ yêu cầu, và mức kỹ năng tương ứng phải lớn hơn hoặc bằng mức kỹ năng yêu cầu. Đối chiếu với ràng buộc này, từ danh sách tác vụ trong Bảng 3 và danh sách tài nguyên trong Bảng 4, ta có thể xác định được danh sách tài nguyên có thể thực hiện từng tác vụ như trong Bảng 5 dưới đây.

Bảng 3. Danh sách tác vụ và yêu cầu tài nguyên

Thứ tự	Tác vụ	Yêu cầu kỹ năng của tài nguyên thực hiện
1	W_1	$S_{2,2}$
2	W_2	$S_{1,1}$
3	W_3	$S_{2,1}$
4	W_4	$S_{1,1}$
5	W_5	$S_{1,2}$
6	W_6	$S_{2,1}$
7	W_7	$S_{2,2}$
8	W_8	$S_{1,2}$
9	W_9	$S_{2,1}$
10	W_{10}	$S_{1,1}$

Bảng 4. Danh sách tài nguyên và năng lực tài nguyên

Thứ tự	Tài nguyên	Các kỹ năng và mức kỹ năng của tài nguyên
1	L_1	$S_{1,2}, S_{2,1}$
2	L_2	$S_{2,1}, S_{2,2}$
3	L_3	$S_{1,2}, S_{2,1}$
4	L_4	$S_{1,1}, S_{2,2}$

Bảng 5. Tài nguyên có khả năng thực hiện tác vụ

Thứ tự	Tác vụ	Các tài nguyên có khả năng thực hiện tác vụ
1	W_1	L_2, L_4
2	W_2	L_1, L_3, L_4
3	W_3	L_1, L_2, L_3
4	W_4	L_1, L_3, L_4
5	W_5	L_1, L_3
6	W_6	L_1, L_2, L_3
7	W_7	L_2, L_4
8	W_8	L_1, L_3
9	W_9	L_1, L_3, L_4
10	W_{10}	L_1, L_2, L_3

Trong bảng 3, bảng 4 và bảng 5, ký hiệu: $S_{i,j}$ nghĩa là kỹ năng S_i và mức kỹ năng j . Tác vụ yêu cầu tài nguyên thực hiện phải đáp ứng loại kỹ năng và mức kỹ năng cụ thể. Tài nguyên thực hiện cần có cùng loại kỹ năng và mức kỹ năng

lớn hơn hoặc bằng mức kỹ năng yêu cầu.

Cụ thể:

Trong bảng 3, tác vụ W_1 , yêu cầu tài nguyên có loại kỹ năng S_2 và mức kỹ năng lớn hơn hoặc bằng 2. Đối chiếu với năng lực của tài nguyên trong bảng 3, ta thấy tài nguyên L_2 và L_4 có loại kỹ năng S_2 và mức kỹ năng của S_2 là 2, do vậy tài nguyên L_2 và L_4 có thể thực hiện tác vụ W_1 .

4. THUẬT TOÁN ĐỀ XUẤT

4.1. Phương pháp biểu diễn cá thể

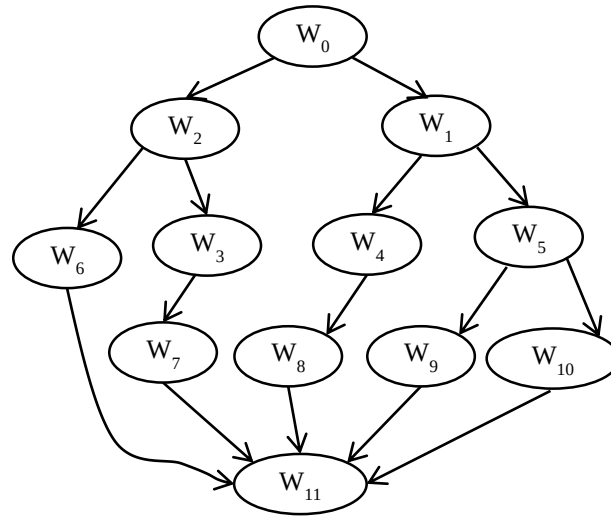
Trong giải thuật lập lịch, một lịch biểu (hay một cá thể) được thể hiện như một vector có chiều dài bằng số tác vụ cần thực hiện, mỗi phần tử của vector sẽ lưu giá trị là chỉ số của tài nguyên thực hiện tác vụ đó.

Với bài toán MS-RCPSP, một cá thể hay một lịch biểu là một vector có số phần tử bằng số tác vụ của dữ liệu đầu vào, giá trị mỗi phần tử của vector lịch biểu là chỉ số của tài nguyên sử dụng để thực hiện tác vụ đó. Tài nguyên này cần đáp ứng các yêu cầu ràng buộc theo điều kiện của bài toán, tức là có loại kỹ năng giống với kỹ năng yêu cầu và mức kỹ năng (level) phải lớn hơn hoặc bằng mức kỹ năng yêu cầu.

Ví dụ 1: Xét một dự án gồm 10 tác vụ, danh sách tác vụ thể hiện trong bảng 3, dự án có 4 tài nguyên được thể hiện trong bảng 4, như vậy:

- Tập hợp tác vụ $W = \{W_1, W_2, W_3, W_4, W_5, W_6, W_7, W_8, W_9, W_{10}\}$
- Tập hợp tài nguyên $L = \{L_1, L_2, L_3, L_4\}$.

Thứ tự thực hiện các tác vụ được biểu diễn bởi đồ thị ưu tiên như trong hình 1 dưới đây.

**Hình 1.** Đồ thị ưu tiên các tác vụ trong dự án

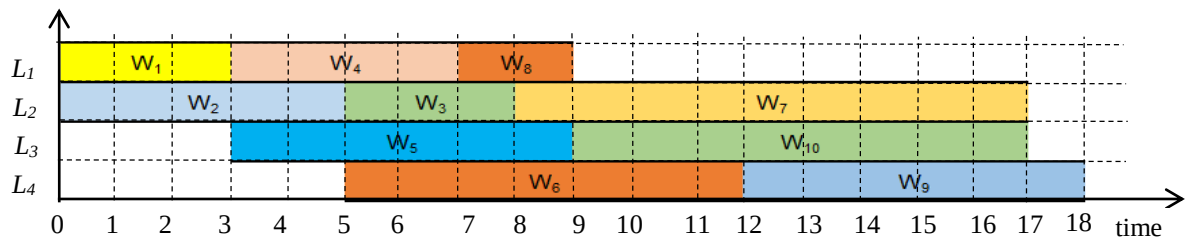
Tác vụ W_0 và W_{11} là hai tác vụ giả, để thể hiện điểm bắt đầu và điểm kết thúc của dự án. Thời gian thực hiện từng tác vụ (tính theo giờ) được thể hiện trong Bảng 6.

Bảng 6. Thời gian thực hiện các tác vụ

Tác vụ	W_1	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}
Thời gian	0	3	5	3	4	6	7	9	2	6	8	0

Hình 2 biểu diễn việc bố trí tài nguyên thực hiện các dự án theo thời gian. Việc bố trí tài nguyên thực hiện tác vụ cần đáp ứng được các

ràng buộc của bài toán về thứ tự ưu tiên thực hiện các tác vụ và khả năng thực hiện tác vụ của tài nguyên.

**Hình 2.** Tài nguyên thực hiện tác vụ theo thời gian

Với phương án lịch biểu được thể hiện trong Hình 2, ta có thể biểu diễn lịch biểu này dưới dạng một vector như sau:

Tác vụ	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}
Tài nguyên	1	2	2	1	3	4	2	1	4	3

Như vậy, tài nguyên L_1 sẽ thực hiện các tác vụ: W_1, W_4, W_8 ; tài nguyên L_2 thực hiện các tác vụ: W_2, W_3, W_7 ; tài nguyên L_3 thực hiện các tác vụ: W_5, W_{10} ; tài nguyên L_4 thực hiện các tác vụ: W_6

và W_9 .

Với các bài toán thuộc lớp NP-Hard, việc tìm lời giải tối ưu rất khó khăn do không gian nghiệm lớn, nên thường sử dụng các phương pháp tiến

hóa, cần tối ưu để tìm ra nghiệm gần đúng, đủ tốt trong thời gian phù hợp. Bài báo này đề xuất thuật toán MEM-RES để tìm lời giải cho bài toán MS-RCPSP, đây là thuật toán lai giữa giải thuật gần đúng Memetic và phương pháp Reassignment.

4.2. Phương pháp Reassignment

Trong bài toán MS-RCPSP, một lịch biểu là một cá thể biểu diễn tài nguyên thực hiện tác vụ thỏa mãn các ràng buộc ban đầu. Sau mỗi thế hệ, thuật toán sẽ tìm ra được lịch biểu tốt nhất (C_{best}), dựa trên lịch biểu này, thực hiện tiếp thay đổi các tài nguyên thực hiện từng tác vụ sao cho vẫn đảm bảo được ràng buộc của bài toán. Việc thay đổi tài nguyên thực hiện tác vụ được gọi là phương pháp tái thiết lập (Reassignment). Phương pháp này giúp mở rộng được không gian tìm kiếm và có thể tránh được cực trị địa phương.

Phương pháp Reassignment được thực hiện qua các bước như sau:

- **Bước 1:** Tìm tài nguyên sử dụng nhiều nhất trong C_{best} , gọi là L_b . Đây là tài nguyên kết thúc công việc sau cùng trong các tài nguyên sử dụng để thực hiện dự án.
- **Bước 2:** Duyệt lần lượt từng tác vụ được thực hiện bởi tài nguyên L_b theo thứ tự ưu tiên thực hiện của các tác vụ trên lịch biểu.

W^R_b : tập tác vụ đang được thực hiện bởi L_b

- **Bước 3:** Với mỗi tác vụ $W_i \in W^R_b$ ($i=1-sizeof(W^R_b)$), thực hiện:

- Tìm tất cả các tài nguyên có khả năng thực hiện được tác vụ W_i khác với tài nguyên hiện tại (L_b), ký hiệu là L^W ;
- Duyệt lần lượt từng tài nguyên có thể thực hiện tác vụ W_i , với mỗi tài nguyên $L_j^W \in L^W$, thực hiện:

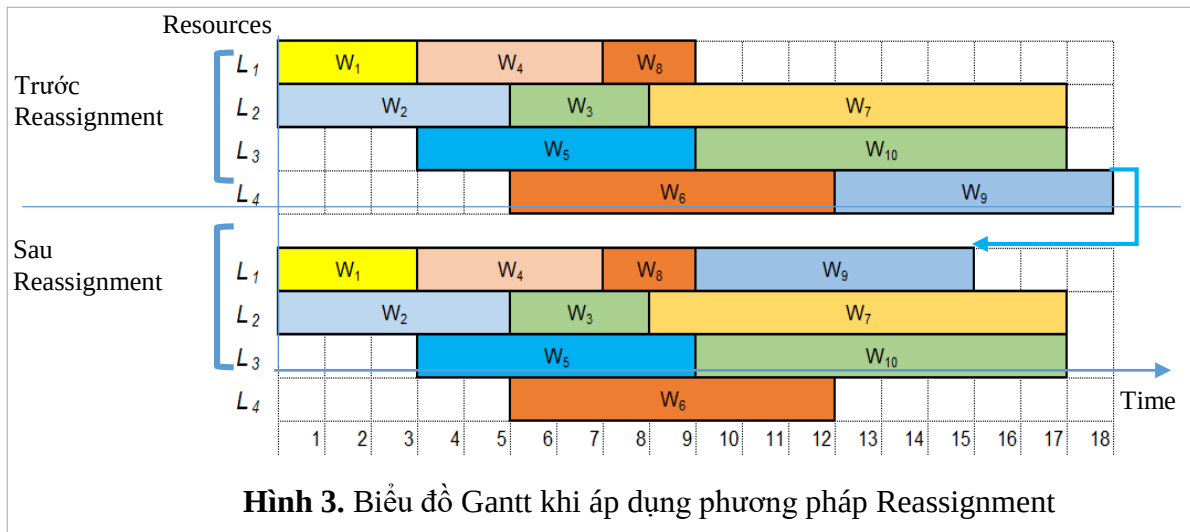
- (1) Thiết lập tài nguyên thực hiện W_i là R_j^W ;
- (2) Xóa W_i khỏi danh sách tác vụ mà là L_b đang thực hiện.

Sau (1) và (2) ta sẽ nhận được lịch biểu mới, tính makespan của lịch biểu mới $newbest$, nếu tốt hơn $makespan$ của C_{best} , thay thế C_{best} bằng lịch $newbest$ và dừng lại, nếu không, tiếp tục thực hiện vòng lặp để kiểm tra.

$$C_{best} \begin{cases} newbest & \text{if } f(newbest) < f(C_{best}) \\ C_{best} & \text{if } f(newbest) \geq f(C_{best}) \end{cases}$$

Quá trình tính toán này giúp thuật toán mở rộng không gian tìm kiếm bằng việc tạo ra các cá thể mới dựa trên cá thể tốt nhất. Do vậy, sau khi áp dụng Reassignment, chúng ta có thể nhận được các cá thể tốt hơn.

Hình 3 là ví dụ về việc áp dụng phương pháp Reassignment để tìm được phương án lịch biểu tốt hơn cho dự án trong **Ví dụ 1**. Trước khi áp dụng Reassignment, thời gian thực hiện tác vụ là 18 giờ, sau khi áp dụng phương pháp



Hình 3. Biểu đồ Gantt khi áp dụng phương pháp Reassignment

Reassignment, thời gian thực hiện dự án giảm xuống còn 17 giờ

Phương pháp Reassignment được trình bày chi tiết trong Algorithm 1 sau đây.

Algorithm 1. AlgReassignment

Input: *best* (các thể tốt nhất sau mỗi thế hệ)

Output: lịch biểu sau khi được hiệu chỉnh

```

1. makespan = f(best)
2. snew = best;
3. Lb = getResourceLast(snew) //tài nguyên hoàn thành sau cùng
4. Wb = {tập tác vụ thực hiện bởi Lb}
5. For i=1 to size(Wb)
6.   Wi = Wb[i];
7.   Li = {tập tài nguyên có thể sử dụng để chạy Wi # Lb}
8.   For j = 1 to size(Li)
9.     Li[j] = Li[j] + {Wi}
10.    Lb = Lb - {Wi}
11.    newduration= f(snew)
12.    If newduration< duration
13.      duration= newduration
14.      Return snew ;
15.    End if
16.    snew = best ;
17.  End for
18. End for
19. Return snew ;

```

End Function

Với:

- *f*: hàm tính thời gian thực hiện của một cá thể
 - size: trả lại kích thước của một tập/mảng
 - getResourceLast: trả về tài nguyên kết thúc thực hiện cuối cùng
-

4.3. Thuật toán MEM-RES

Algorithm 2 trình bày thuật toán MEM-RES.

Algorithm 2. MEM-RES.

Input: dataset, *gmax*: number of generation,

mutationProbability, crossoverProbability, localSearchProbability

Output: best and duration

```

1. Begin
2. {
3.   Read dataset.
4.   g = 0
5.   Pall = {Khởi tạo quần thể}
6.   {makespan, best, fitness} = {Tính toán các giá trị của quần thể ban đầu}
7.   while(g<gmax)
8.   {
9.     predecessors = Selection(Pall);           // Lựa chọn các cá thể
10.    offspring = Crossover(Pall, crossoverProbability); // Lai ghép
11.    Mutate(offspring, mutationProbability);       // Đột biến
12.    LocalSearch(offspring, localSearchProbability); // Tìm kiếm cục bộ
13.    Pall = CombinePopulation(Pall, offspring); // Kết hợp cá thể
14.    mới để tạo ra quần thể ở thế hệ hiện tại
15.    {new_makespan, new_best, fitness} = {Tính toán các giá trị của quần thể }
16.    new_bestR = AlgReassignment(new_best);
17.    new_makespanR = f(new_bestR) ;
18.    if new_makespanR < new_makespan then
19.    {
20.      new_best = new_bestR;

```

```

20.         new_makespan = new_makespanR ;
21.     }
22.     if (makespan < new_makespan)
23.         makespan = new_makespan;
24.         best = new_best;
25.     end if
26.     g = g+1
27. } // end while
28. return {best, makespan }
29. }

```

Với:

- f: hàm tính thời gian thực hiện của một cá thể
-

Trong Algorithm 2, các dòng 15-21 thể hiện việc áp dụng phương pháp tái thiết lập tài nguyên vào kết quả của MA để tăng khả năng nhận được các cá thể tốt hơn, nếu có, cá thể mới sẽ được thay thế cho cá thể cũ.

5. KẾT QUẢ THỰC NGHIỆM

5.1. Dữ liệu thực nghiệm

5.1.1 Phương pháp số hóa dữ liệu chuyền may công nghiệp

Trong lĩnh vực sản xuất may công nghiệp, để hoàn thành một hợp đồng với nhiều sản phẩm cùng loại, công ty may sẽ tổ chức các chuyền may để thực hiện những công đoạn của sản phẩm. Mỗi chuyền may có nhiều công nhân với các mức tay nghề khác nhau; các công đoạn may trong một chuyền may được bố trí theo thứ tự ưu tiên của quy trình sản xuất thành phẩm. Dữ liệu chuyền may gồm các đặc trưng sau:

- Để hoàn thành một sản phẩm, cần thực hiện qua nhiều công đoạn.
- Các công đoạn có thời gian thực hiện khác nhau và có yêu cầu công nhân thực hiện cần có trình độ (bậc thợ) nhất định để thực hiện.
- Công nhân bậc cao có thể thực hiện được các công đoạn yêu cầu bậc thợ thấp hơn.
- Công nhân có trình độ cao hơn trình độ yêu cầu, có thể hoàn thành công việc sớm hơn.
- Các công đoạn có quan hệ ưu tiên chặt chẽ với nhau, không hoàn thành công đoạn trước sẽ không thực hiện được công đoạn sau, do vậy số lượng ràng buộc ưu tiên là rất nhiều.
- Các công nhân thực hiện trong một hợp đồng

được tổ chức thành một tổ may

Các đặc trưng của dữ liệu chuyền may công nghiệp rất phù hợp với dữ liệu đầu vào của bài toán MS-RCPSP. Tuy nhiên, để có thể áp dụng dữ liệu chuyền may, ta cần chuyển đổi các thông tin này về định dạng mà Real-RCPSP có thể đọc và sử dụng được. Để số hóa dữ liệu chuyền may, ta thực hiện các quy tắc như sau:

- Mỗi hợp đồng may mặc là một dự án và có thể tạo thành nhiều file dữ liệu dựa vào cách bố trí tài nguyên thực hiện.
- Mỗi công đoạn trong chuyền may là một tác vụ
- Thời gian thực hiện một công đoạn là thời gian thực hiện một tác vụ
- Công nhân may có nhiều mức tay nghề hay bậc thợ khác nhau (từ 1 đến 7), bậc thợ sẽ tương ứng với mức kỹ năng trong mô hình bài toán MS-RCPSP.
- Mỗi công nhân là một tài nguyên, tài nguyên sẽ có mức kỹ năng (skill level) cụ thể
- Thứ tự thực hiện các công đoạn may thành phẩm chính là thứ tự ưu tiên thực hiện các tác vụ.
- Kết quả của việc số hóa dữ liệu chuyền may công nghiệp sẽ đưa ra các bộ dữ liệu phù hợp với yêu cầu đầu vào của bài toán MS-RCPSP.

5.1.2. Bộ dữ liệu chuyền may TNG

Để phục vụ cho thực nghiệm các thuật toán, dữ liệu chuyền may công nghiệp của công ty TNG (Company, n.d.) được nghiên cứu và số hóa thành bộ dữ liệu chuẩn phù hợp với bài toán MS-RCPSP. Việc số hóa được thực hiện với hai hợp

đồng may mặc như trong bảng 7.

Với mỗi hợp đồng may, công ty sẽ sử dụng 04 tổ may với số lượng công nhân khác nhau, lần lượt là 37,39,47,41.

Áp dụng quy tắc số hóa dữ liệu chuyển may với dữ liệu hợp đồng trong bảng 8, ta được bảng dữ liệu chuẩn, phù hợp với đầu vào của bài toán MS-RCPSP như trong bảng 8 dưới đây.

Bảng 7. Các hợp đồng may công nghiệp

STT	Mã hợp đồng	Loại sản phẩm	Số sản phẩm	Số công đoạn
1	WE1190/1698402 Liner Buy Mar 14-F19	Áo ni	33,693	71
2	FM4013/ 1536181 buy Nov 08- F19	Quần bơi	83,340	137

Bảng 8. Dữ liệu chuyển may của TNG

Bộ dữ liệu	Số tác vụ	Số công nhân	Số ràng buộc ưu tiên	Số mức kỹ năng	Thời gian thực hiện (PT)
TNG1	71	37	1026	6	409
TNG2	71	39	1026	6	325
TNG3	71	41	1026	6	296
TNG4	71	45	1026	6	392
TNG5	137	37	1894	6	1174
TNG6	137	39	1894	6	1052
TNG7	137	41	1894	6	871
TNG8	137	45	1894	6	996

Trong đó:

- **Số tác vụ:** số lượng công việc (trong chuyển may là số công đoạn) cần hoàn thành của dự án.
- **Số công nhân:** số lượng nhân công trong các tổ may, một người có thể thực hiện một công đoạn khác sau khi hoàn thành một công việc trước đó.
- **Số ràng buộc ưu tiên:** số lượng các công việc cần hoàn thành trước theo trình tự thực hiện để hoàn thành một sản phẩm.
- **Thời gian thực hiện (PT):** tổng thời gian thực hiện hợp đồng thực tế tương ứng với từng tổ may, tính theo giờ

5.2. Cấu hình dữ liệu đầu vào để thực nghiệm

Thuật toán MEM-RES có dữ liệu cài đặt ban đầu như sau:

- Về dữ liệu: sử dụng bộ dữ liệu TNG.
- Số cá thể khởi tạo ban đầu của quần thể: 100.
- Thuật toán thực hiện tiến hóa qua 70.000 thế hệ.
- Lặp lại kiểm thử trên mỗi thành phần của bộ

dữ liệu 30 lần.

- Thuật toán được cài đặt trên Matlab 2014
- Cấu hình máy chạy thuật toán: CPU Core i5, 2.2 GHz, RAM 6GB, Windows 10.

5.3. Kết quả thực nghiệm

Kết quả thực nghiệm với bộ dữ liệu TNG được trình bày trong bảng 9.

Dữ liệu thực nghiệm được tổng hợp, so sánh và đánh giá dựa trên ba yếu tố: BEST - giá trị makespan tốt nhất – đây là thời gian tối thiểu để thực hiện dự án), AVG – thời gian thực hiện trung bình của 30 lần thực nghiệm với mỗi bộ dữ liệu đầu vào và giá trị STD – giá trị độ lệch chuẩn giữa các lần thực nghiệm trên cùng bộ dữ liệu.

Các kết quả thực nghiệm của thuật toán đề xuất được so sánh với thuật toán GA-M, thuật toán này được cài đặt và cung cấp trong bộ công cụ GARunner (Myszkowski P. B., n.d.). Dữ liệu thực nghiệm là bộ dữ liệu TNG đã được số hóa từ dữ liệu chuyển may theo đúng chuẩn đầu vào của GARunner. Ngoài ra, bài báo cũng tiến hành thực nghiệm bộ dữ liệu TNG với thuật toán PSO gốc,

dữ liệu này được thể hiện trong bảng 9 thông qua các giá trị tương ứng BEST, AVG, STD.

Bảng 9. Dữ liệu chạy thực nghiệm thuật toán với bộ dữ liệu TNG

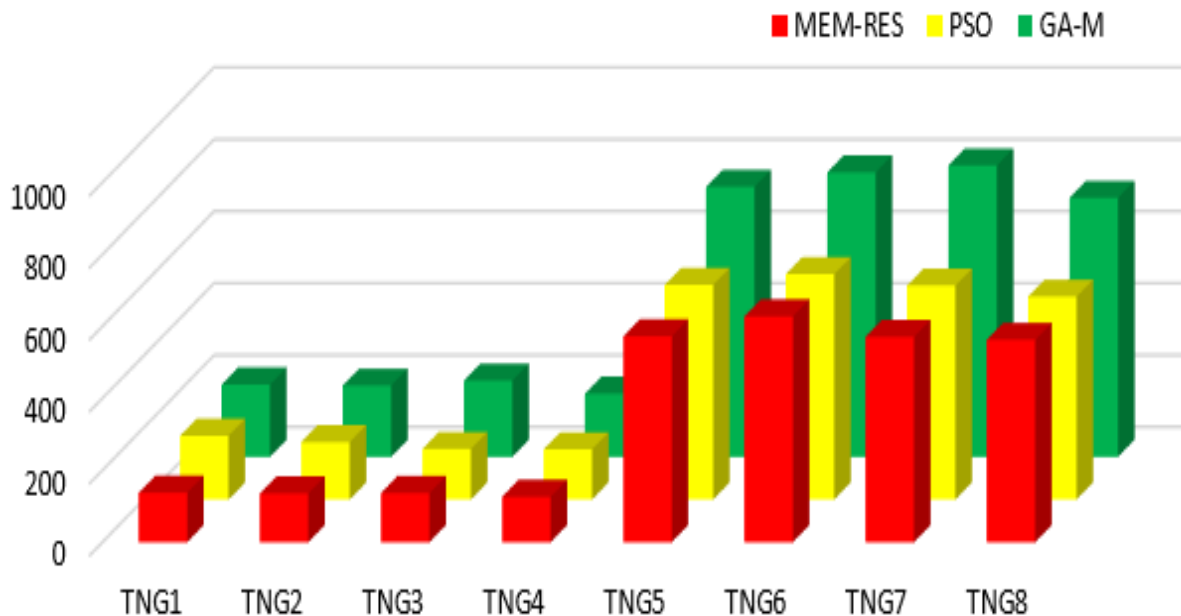
Dataset	GA-M			PSO			MEM-RES		
	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
TNG1	201	203	3.5	178	183	4.8	138	139	3.4
TNG2	198	205	8.3	160	166	6.2	136	142	6.7
TNG3	212	218	7.1	141	155	9.1	137	142	5.4
TNG4	176	187	11.3	140	147	8.3	127	135	8.1
TNG5	751	757	6.2	597	605	9.6	573	578	7.8
TNG6	791	796	5.5	627	639	8.5	628	636	5.5
TNG7	810	820	10.7	595	605	8.7	571	581	9.2
TNG8	720	728	9.4	564	567	2.3	563	569	4.2
Tổng			62.0			57.4			50.3

Kết quả thực nghiệm chỉ ra thuật toán MEM-RES hiệu quả hơn so với thuật toán GA-M khi so sánh các giá trị:

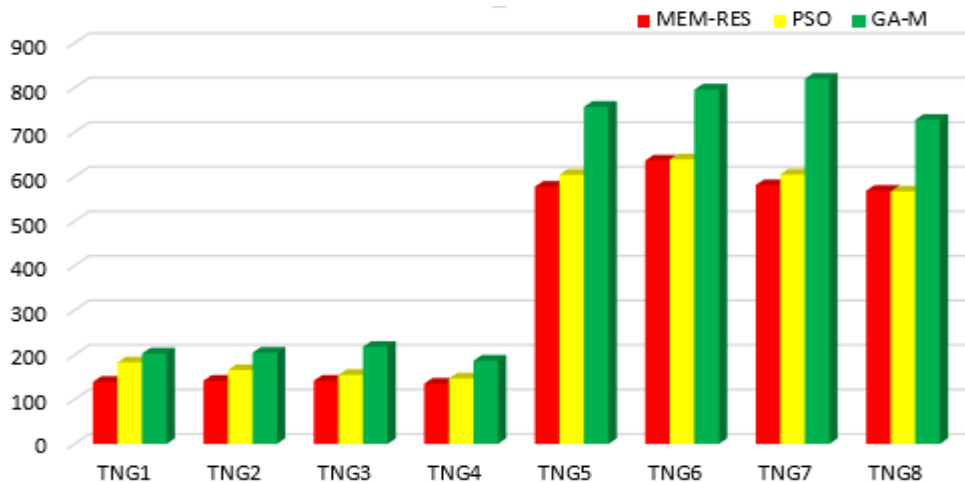
- Với giá trị BEST: MEM-RES tốt hơn GA-M và PSO 7/8 bộ dữ liệu, các giá trị tốt hơn từ 21,6% đến 35,6% so với GA-M và tốt hơn từ 0% đến 22,5% với PSO. Trong đó, có một bộ dữ liệu kết quả của PSO tốt hơn so với MEM-RES. Chi tiết so sánh giá trị BEST được thể hiện trong hình 4.
- AVG: MEM-RES tốt hơn từ 20,10% đến 34,8% so với GA-M, tốt hơn từ 0% đến 23,8% so với PSO. So sánh chi tiết giá trị

AVG được thể hiện trong hình 5.

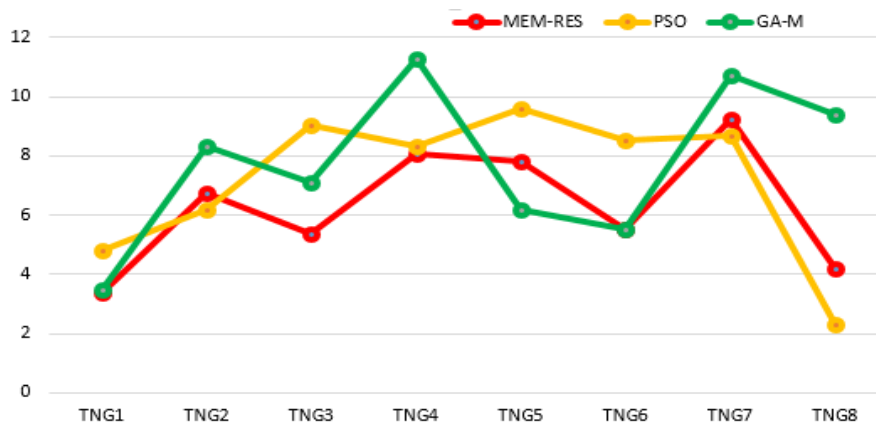
- STD (độ lệch chuẩn): GA-M có tổng độ lệch chuẩn trên 8 bộ dữ liệu của TNG là 62,0, với thuật toán PSO là 57,4 và của MEM-RES có tổng là 50,3. Kết quả này thể hiện thuật toán đề xuất MEM-RES có biên độ sai lệch kết quả giữa các lần thực nghiệm ít hơn so với thuật toán GA-M và PSO. Tuy nhiên, ba bộ dữ liệu của MEM-RES độ lệch chuẩn kém hơn các thuật toán khác, cụ thể TNG1 và TNG7 kém hơn PSO, TNG5 kém hơn GA-M. Chi tiết so sánh độ lệch chuẩn được thể hiện trong hình 6.



Hình 4. Giá trị tốt nhất (BEST) của MEM-RES so với PSO và GA-M



Hình 5. Giá trị bình quân (AVG) giữa các lần thực nghiệm của MEM-RES so với PSO và GA-M



Hình 6. Biểu đồ độ lệch chuẩn (STD) của MEM-RES so với PSO và GA-M

6. KẾT LUẬN

Việc lập kế hoạch điều phối sản xuất thông minh dựa trên việc tìm phương án sắp xếp các tài nguyên (công nhân) thực hiện các tác vụ (công việc) trong một dự án (hợp đồng) có thể thực hiện bằng việc áp dụng bài toán MS-RCPSP. Trong bài báo này, tác giả đã tìm hiểu bài toán MS-RCPSP, hệ thống hóa các quy ước và phát biểu bài toán thông qua các công thức toán học một cách rõ ràng. Để tìm lịch biểu cận tối ưu của bài toán, bài báo đã đề xuất một thuật toán tiến hóa dựa trên thuật toán Memetic lai với phương pháp Reassignment. Thuật toán mới thực hiện việc tìm kiếm nghiệm tốt hơn sau mỗi thế hệ tiến hóa của

Memetic bằng cách mở rộng sang các vùng nghiệm lân cận thông qua tái thiết lập tài nguyên thực hiện các tác vụ trên lịch biểu tốt nhất. Kết quả kiểm chứng cho thấy, thuật toán đề xuất MEM-RES phù hợp với bài toán MS-RCPSP và mang lại hiệu quả cao so với thuật toán đối sánh. Việc thực nghiệm được tiến hành dựa trên bộ dữ liệu thực tế của chuyên may công nghiệp TNG, dữ liệu này được số hóa từ các hợp đồng may mặc thành bộ dữ liệu chuẩn của bài toán MS-RCPSP. Các bước thực nghiệm được chạy trên quần thể khởi tạo ban đầu với 100 cá thể, tiến hóa qua 70.000 thế hệ và được kiểm chứng thông qua 30 lần chạy cho mỗi bộ dữ liệu. Kết quả được thu

thập, tổng hợp, đánh giá cho thấy: MEM-RES mang lại hiệu quả tốt hơn GA-M (thuật toán đối sánh), cụ thể: giá trị BEST tốt hơn GA-M từ 20,6% đến 35,6% và tốt hơn từ 0% đến 22,5% với PSO; giá trị AVG tốt hơn đến 20,1% đến 34,8% tốt hơn từ 0% đến 23,8% so với PSO. Lời giải của bài toán MS-RCPSP tìm được thông qua việc áp dụng thuật toán MEM-RES là một lịch biểu thể hiện cách sắp xếp tài nguyên thực hiện từng tác vụ của dự án (với bộ dữ liệu TNG, đây là cách bố trí công nhân thực hiện các công đoạn may thành phẩm của một hợp đồng). Lịch biểu này có thể áp dụng trong việc lập kế hoạch sản xuất tự động thay vì phương pháp lập lịch truyền thống (đa số dựa theo kinh nghiệm) hiện nay đồng thời cũng mở ra khả năng ứng dụng của bài toán MS-RCPSP trong thực tế lập kế hoạch sản xuất, đặc biệt với mô hình sản xuất theo dây chuyền. Tiếp theo, tác giả sẽ tiếp tục nghiên cứu thêm các ràng buộc của bài toán lập kế hoạch trong thực tế, bổ sung thêm các yếu tố tác động đến quá trình sản xuất để nâng cao hiệu quả của việc lập kế hoạch sản xuất tự động, mở rộng bộ dữ liệu thực nghiệm và nghiên cứu thêm các phương pháp giải.

TÀI LIỆU THAM KHẢO

- Afsar, S. e. (2022). Multi-objective enhanced memetic algorithm for green job shop scheduling with uncertain times. *Swarm and Evolutionary Computation*, 68 (2022): 101016.
- Ansari, M. A. (2021). A hybrid PSO-GA algorithm for multi-skill resource-constrained project scheduling problem. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 6207-6217.
- Asadujjaman, M. H. (2021). An Immune Genetic Algorithm for Solving NPV-Based Resource Constrained Project Scheduling Problem. *IEEE Access* 9, 26177-26195.
- Bhat, A. R. (2021). A modified differential evolution algorithm for multi-skill resource-constrained project scheduling problem. *Journal of Intelligent Manufacturing*, 32, 153-167.
- Castro-Gutierrez, J. &.R. (2021). Differential evolution algorithm with gradient descent mutation for the multi-skill resource-constrained project scheduling problem. *International Journal of Production Research*, 59(1), 175-192.
- Curi, R. D. (2020). Multi-objective genetic algorithm with multi-heuristics for the multi-skill resource-constrained project scheduling problem. *Expert Systems with Applications*, 156, 113399.
- Dakhli, M. G. (2020). A multi-objective genetic algorithm for the multi-skill resource-constrained project scheduling problem. *Applied Soft Computing*, 87, 105944.
- Emary, E. Z. (2020). A novel hybrid differential evolution and particle swarm optimization algorithm for multi-skill resource-constrained project scheduling problem. *Soft Computing*, 24(23), 17967-17980.
- Gholami, M. &. (2020). A novel multi-objective PSO algorithm for multi-skill resource-constrained project scheduling problem. *Engineering Applications of Artificial Intelligence*, 103, 104227.
- Huu Dang Quoc, L. N. (2020). New Effective Differential Evolution Algorithm for the Multi-Skill Resource Constrained Project Scheduling Problem. 2020 2nd International Conference on Computer Communication and the Internet (ICCCI 2020), June 26-29. Nagoya, Japan: IEEE-<http://www.iccci.org>
<https://ieeexplore.ieee.org/document/9145982>, DOI: 10.1109/ICCCI49374.2020.9145982.

- Huu Dang Quoc, L. N. (2022). An Effective Hybrid Algorithm Based on Particle Swarm Optimization with Migration Method for Solving the Multiskill Resource-Constrained Project Scheduling Problem. *Applied Computational Intelligence and Soft Computing (Q2)*, Article ID 6230145, 12 pages, DOI: 10.1155/2022/6230145.
- J. Cui, D. W. (2020). A hybrid quantum-behaved particle swarm optimization algorithm for the resource-constrained project scheduling problem. *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, (pp. 797-802).
- J. Lin, L. Z. (2020). A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Systems with Applications*, 140, 112915, 2020.
- J. Wu, Y. L. (2018). A hybrid differential evolution algorithm for the resource-constrained project scheduling problem. *IEEE Access*, vol. 6, pp. 34083-34093.
- Jedrzejowicz, P. a.-R. (2022). A-Team Solving Multi-Skill Resource-Constrained Project Scheduling Problem. *Procedia Computer Science*, 207 (2022): 3300-3309.
- Klein, R. (2012). *Scheduling of Resource-Constrained Projects*. Springer Science & Business Media.
- Laszczyk, M. &. (2019). Improved selection in evolutionary multi-objective optimization of multi-skill resource-constrained project scheduling problem. *Information Sciences*, 481, 412-431.
- Maghsoudlou, H. A.-N. (2017). Multi-skilled Project scheduling with level-dependent rework risk; three multi-objective mechanisms based on cuckoo search. *Applied Soft Computing*, 54, 46-61.
- Myszkowski, P. B. (2022). Investigation of benchmark dataset for many-objective Multi-Skill Resource Constrained Project Scheduling Problem. *Applied Soft Computing*, 127: 109253.
- Myszkowski, P. B. (n.d.). GARunner tool. Retrieved from http://imopse.ii.pwr.wroc.pl/rcpsp_spsp_library.html
- P.B. Myszkowski, M. L. (2019). iMOPSE: a library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem. *Soft Computing Journal*, 23: 32397.
- Pang, X. Z. (2021). A multi-objective genetic algorithm based on adaptive elitist archive for multi-skill resource-constrained project scheduling problem. *IEEE Access*, 9, 115258-115268.
- R. Kolisch, A. S. (1997). PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. *European journal of operational research*, 96(1), pp.205-216.
- Rathore, R. S. (2020). Multi-skill resource-constrained project scheduling problem: A particle swarm optimization-based approach. *International Journal of Applied Management Science*, 12(1), 30-52.
- Rostami, M. &. (2020). A hybrid differential evolution and bee algorithm for multi-skill resource-constrained project scheduling problem. *Expert Systems with Applications*.
- S. Sanaei, M. P. (2020). A new hybrid evolutionary algorithm for the multi-skill resource-constrained project scheduling problem.

Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 11, pp. 5071-5085.

Saad, H. M. (2021). Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling. IEEE Access, 9 (2021): 38488-38502.

Saad, H. M. (2021). Quantum-Inspired Genetic Algorithm for Resource-Constrained Project-Scheduling. IEEE Access, 9: 38488-38502.

Seo, W. e. (2022). Effective memetic algorithm for multilabel feature selection using hybridization-based communication. Expert Systems with Applications, 201 (2022): 117064.

Stiti, C. a. (2019). A new approach for the multi-site resource-constrained project scheduling problem. Procedia Computer Science, 164: 478-484.

Wilson, A. J. (2022). A review on memetic algorithms and its developments. Electrical and Automation Engineering, 1.1 (2022): 7-12.

X.S., Y. (2010). Nature-Inspired Metaheuristic Algorithms. Luniver Press, ISBN-13: 978-1-905986-28-6, 2010.

Xu, J. H. (2020). A hybrid genetic algorithm with resource re-allocation for multi-skill resource-constrained project scheduling problem. Applied Sciences, 10(21), 7674.

Y. Li, H. S. (2020). An improved differential evolution algorithm for resource-constrained project scheduling problem. Journal of Intelligent Manufacturing, , vol. 31, no. 2, pp. 461-472.

Yang X.S, D. S. (2009). Cuckoo search via Lévy flights. Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), (pp. 210-214). USA.

Zhou, T. e. (2022). Multi-objective stochastic project scheduling with alternative execution methods: An improved quantum-behaved particle swarm optimization approach. Expert Systems with Applications, 203: 117029.