

NGHIÊN CỨU XÂY DỰNG THUẬT TOÁN LẬP KẾ HOẠCH CHO CÔNG TÁC BẢO TRÌ, BẢO DƯỠNG

Đào Quang Khoa, Trần Ngọc Trung, Nguyễn Thành Trung, Trần Vũ Tùng, Hoàng Kỳ Sơn, Ngô Hữu Hải

Công ty Điều hành Dầu khí Biển Đông

Email: trungtn@biendongpoc.vn

<https://doi.org/10.47800/PVSI.2024.01-10>

Tóm tắt

Sử dụng dữ liệu thu thập từ ứng dụng quản lý bảo trì thiết bị, Công ty Điều hành Dầu khí Biển Đông (Bien Dong POC) đã nghiên cứu khả năng áp dụng thuật toán di truyền để hỗ trợ công tác lập kế hoạch cho công tác bảo trì bảo dưỡng, từ đó khai thác được năng lực tính toán nhanh của máy tính, rút ngắn thời gian lập kế hoạch, đảm bảo tối ưu hóa nguồn lực, cũng như đảm bảo sự thống nhất trong quá trình lập kế hoạch và tổ chức thực hiện. Kết quả nghiên cứu cho thấy việc áp dụng thuật toán di truyền giúp lập kế hoạch chính xác với độ khớp cao, xử lý được các yêu cầu về tiến độ, mức độ ưu tiên và nguồn lực cho từng mã công việc, trong khi thời gian tính toán được rút ngắn đáng kể so với cách làm truyền thống. Trên cơ sở nghiên cứu, có thể thiết lập các công cụ tự động áp dụng thuật toán này để hỗ trợ nghiệp vụ lập kế hoạch bảo dưỡng giúp nâng cao hiệu suất và độ chính xác của quy trình quản lý bảo trì, bảo dưỡng, đặc biệt là tại các cơ sở sản xuất quy mô lớn.

Từ khóa: Thuật toán di truyền, lập kế hoạch, bảo trì bảo dưỡng.

1. Giới thiệu

Công tác bảo trì, bảo dưỡng phòng ngừa (preventive maintenance) tại các cơ sở sản xuất được thực hiện định kỳ và có kế hoạch nhằm đảm bảo hệ thống công nghệ hoạt động an toàn, giảm nguy cơ hỏng hóc cũng như thời gian ngừng hoạt động ngoài kế hoạch. Điều kiện hoạt động sản xuất và vận hành thiết bị trên cụm giàn khai thác mỏ Hải Thạch - Mộc Tinh (cách đất liền 300 km) khiến cho việc điều động nguồn lực, gồm con người và thiết bị, gặp nhiều khó khăn. Do đó, Bien Dong POC đặt mục tiêu tận dụng tối đa nguồn lực tại chỗ cho công tác bảo trì, bảo dưỡng phòng ngừa cho một số lượng lớn thiết bị trên giàn khai thác. Với khả năng tập trung hóa dữ liệu và các tiến bộ về hệ thống truyền tải dữ liệu từ các giàn ngoài khơi, công cụ lập kế hoạch sẽ hỗ trợ cho việc tối ưu các hoạt động vận hành khai thác và bảo trì bảo dưỡng [1, 2].

Tại Bien Dong POC, công tác bảo trì bảo dưỡng được quản lý tập trung bằng phần mềm quản lý bảo trì thiết bị (Computerized Maintenance Management System -

CMMS). Việc tận dụng năng lực tính toán của máy tính và ứng dụng các thuật toán tiên tiến để tạo ra công cụ lập kế hoạch cho công tác bảo trì bảo dưỡng sẽ giúp rút ngắn thời gian triển khai, tối ưu hóa các nguồn lực, đảm bảo sự thống nhất trong việc lập kế hoạch và tổ chức thực hiện.

2. Một số thuật toán được sử dụng lập kế hoạch cho công việc

Bài toán lập kế hoạch cho công việc dựa trên những điều kiện ràng buộc cho trước về thời gian, nguồn lực, mức độ ưu tiên để xác định thứ tự công việc. Các đặc trưng cơ bản của bài toán lập thời gian biểu cho công việc gồm:

Tập công việc: Mô tả tính chất công việc, ví dụ như lập danh sách mã công việc, lập danh sách các đơn vị, lập bảng thời gian làm việc của nhân công;

Nguồn lực (tài chính, máy móc thiết bị, nhân sự) để thực hiện công việc;

Thời gian giới hạn: Mô tả các dạng thời gian, ví dụ như thời gian cho từng mã công việc, ca làm việc, ngày làm việc;

Ràng buộc: Khi phân công công việc, cần cụ thể số lượng nhân công thực hiện;



Ngày nhận bài: 16/1/2024. Ngày phản biện đánh giá và sửa chữa: 16/1 - 5/2/2024.

Ngày bài báo được duyệt đăng: 5/2/2024.

Mục tiêu: Hoàn thành công việc nhiều nhất có thể với tài nguyên hiện có. Khi các mục tiêu thỏa mãn thì các điều kiện cũng được thỏa mãn.

Ngoài ra, do nguồn lực hạn chế, việc đáp ứng các yêu cầu như phân bố nguồn lực hợp lý, sắp xếp công việc liên tiếp nhau hay thời lượng dành cho công việc,... sẽ phụ thuộc vào tổng số công việc và nguồn lực đang có.

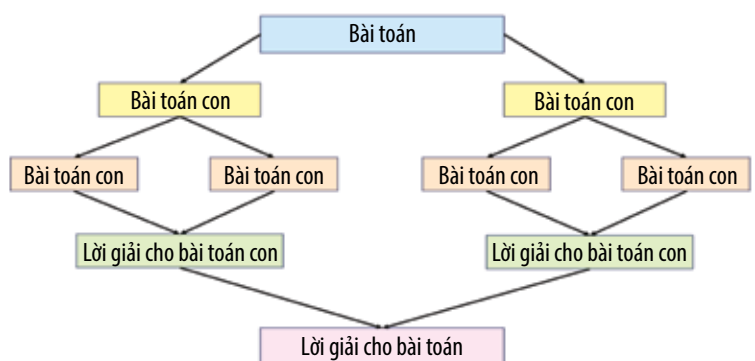
Trong các tình huống không quá phức tạp, khi các yếu tố ảnh hưởng đến thời gian biểu không thay đổi nhiều, việc thiết lập kế hoạch có thể được thực hiện thông qua thuật toán lập trình tuyến tính [3]. Tuy nhiên, khi số lượng công việc nhiều và phức tạp, nguồn lực hạn chế và điều kiện thường xuyên thay đổi, những thuật toán này thường gặp vấn đề khi tạo ra quá nhiều tổ hợp, dẫn đến khả năng đưa ra các giải pháp không thể thực hiện kịp thời theo sự thay đổi đã phát sinh. Một số nghiên cứu xây dựng các hướng dẫn logic, từ đó tạo ra một tập hợp giới hạn các giải pháp theo cách tiếp cận được xác định bởi thời gian cho phép, như lập luận từ phía sau (backward-chaining) [4] hoặc lập luận mờ (fuzzy reasoning) [5].

2.1. Thuật toán Brute force

Thuật toán Brute force giúp tìm ra phương án tối ưu bằng cách lựa chọn trong tập hợp tất cả các phương án của bài toán. Khi bài toán quá lớn, việc áp dụng thuật toán Brute force sẽ không đảm bảo về thời gian cũng như kỹ thuật. Để khắc phục nhược điểm này, trong quá trình duyệt sẽ luôn giữ lại “phương án mẫu” (sample statistics), có giá trị nhỏ nhất tại thời điểm được ghi nhận [6]. Một cách khác để sàng lọc phương án tối ưu là sử dụng phương pháp “đánh giá nhánh cận” (branch and bound). Bằng cách tính toán độ cải thiện từ việc xây dựng phương án theo hướng để ra so với “phương án mẫu”, tăng độ thực thi của thuật toán.

Khi lập kế hoạch, công việc sẽ được chia cho các trường hợp có thể xảy ra. Sau mỗi lần phân chia, các điều kiện ràng buộc sẽ được xem xét. Tập hợp công việc được phân phối nào thỏa mãn điều kiện thì sẽ được lấy.

- Ưu điểm: Với thuật toán này, đáp số cuối cùng của đầu ra (output) đảm bảo tối ưu nhất so với các thuật toán khác. Ngoài ra,



Hình 1. Lưu đồ của thuật toán chia để trị (divide and conquer) [8].

thuật toán này đòi hỏi ít bộ nhớ và cài đặt đơn giản [7].

- Nhược điểm: Thời gian thực thi lớn; việc thực hiện tương đương với phân chia công việc theo cách thủ công thông thường [7].

Về mặt nguyên tắc, thuật toán luôn tìm được nghiệm nếu bài toán có nghiệm bằng cách kiểm tra toàn bộ tập trường hợp xảy ra. Nhưng trên thực tế, các bài toán lập kế hoạch không áp dụng phương pháp này, vì số công việc và số lần lặp lại rất cao trong khi thời gian tính toán, năng lực và dung lượng bộ nhớ đều có hạn.

2.2. Thuật toán chia để trị (divide and conquer)

Thuật toán chia để trị sử dụng kỹ thuật từ trên xuống (top-down approach), chia bài toán lớn thành nhiều bài toán nhỏ và tương đương. Từng lớp bài toán nhỏ tiếp tục được chia đến khi gặp “bài toán cơ sở” (preliminary problem) có thể dễ dàng thu được kết quả. Sau khi giải các bài toán cơ sở này, kết quả được phân tích và tổng hợp ngược lại để có lời giải cho các bài toán nhỏ và cuối cùng có được kết quả cho bài toán lớn. Thuật toán này được chia thành 2 luồng xử lý chính: chia bài toán (từ trên xuống) và tổng hợp kết quả (từ dưới lên). Mục tiêu chính là chia công việc lớn thành các công việc nhỏ hơn, đến khi có thể phân chia công việc trực tiếp.

- Ưu điểm: Khi gặp việc khó xử lý, chia nhỏ thành các việc nhỏ dễ dàng hơn, để giảm thiểu độ phức tạp. Một nhiệm vụ có thể có nhiều tác vụ nhỏ thực hiện song song. Tiêu tốn ít tài nguyên bộ nhớ khi phân bố [9].

Nhược điểm: Việc sử dụng đệ quy để tổng hợp kết quả tiêu tốn nhiều thời gian. Khó hiện thực hóa thuật toán để có thể phân bố từng lớp bài toán, khả năng cao tồn tại vòng lặp bài toán trước đó đã phân chia [9].

Bài toán sắp xếp kế hoạch công việc theo công việc có độ phức tạp cao trên tập cơ sở dữ liệu lớn và là bài toán đa mục tiêu. Vì vậy, nếu xem xét chia nhỏ công việc để giảm bớt độ phức tạp, sau đó khi kết hợp kết quả có thể xảy ra xung đột thời gian công việc. Ví dụ, công việc có thể bị chia thành quá nhiều việc nhỏ không cần thiết hoặc lặp lại, làm phát sinh vấn đề tối ưu hóa để thỏa mãn đầu ra. Vì vậy, không nên

chỉ áp dụng đơn thuần thuật toán chia để trị mà nên kết hợp với các phương pháp khác để tối ưu nguồn lực.

2.2. Thuật toán Heuristic

Thuật toán Heuristic được thiết kế để giải quyết bài toán nhanh hơn và hiệu quả hơn so với các phương pháp truyền thống bằng cách đổi tính tối ưu, độ chính xác hoặc tính đầy đủ để có được tốc độ xử lý cao. Để làm điều này, thuật toán Heuristic tạo ra giải pháp riêng lẻ hoặc được sử dụng để cung cấp đường cơ sở tốt dựa trên phân tích yếu tố cụ thể gắn với bài toán và được bổ sung các thuật toán tối ưu hóa để xác định phương hướng tìm kiếm lời giải [10].

Những thuật toán cơ sở dành cho bài toán lập kế hoạch đã nêu trên đều có thể xây dựng thành thuật toán Heuristic dựa vào các nguyên lý sau:

- Nguyên lý vét cạn thông minh (intelligent brute-forcing): Trong bài toán tìm kiếm công việc tối ưu có điều kiện, khi không gian tìm kiếm lớn, không gian tìm kiếm được giới hạn lại hoặc thực hiện kiểu tìm kiếm đặc biệt dựa vào đặc thù của công việc hoặc thông tin thêm và cụ thể về điều kiện ràng buộc để nhanh chóng tìm ra mục tiêu;
- Nguyên lý tham lam (greedy): Tiêu chuẩn tối ưu của toàn bộ công việc sẽ được chọn để làm hành động cho phạm vi cục bộ của từng bước trong quá trình tìm kiếm công việc thỏa mãn điều kiện;
- Nguyên lý thứ tự (sorting): Các công việc sẽ thực hiện dựa trên cấu trúc thứ tự hợp lý của không gian đang khảo sát để nhanh chóng đạt được một tập phân lịch biểu tốt;
- Nguyên lý hướng đích (heuristic initiative): Khi xây dựng thuật toán Heuristic thường sử dụng các hàm đánh giá thô và kết quả phụ thuộc vào trạng thái hiện tại của bài toán tại mỗi bước giải. Nhờ giá trị này, có thể chọn được cách hành động tương đối hợp lý trong từng bước của thuật toán [12, 13].

- Ưu điểm: Tìm được lời giải tốt, dễ hiểu và dễ thực thi; chi phí giải bài toán sử dụng thuật toán Heuristic thấp hơn đáng kể do dễ dàng và nhanh chóng đưa ra kết quả so với thuật toán tối ưu; kết quả đạt được gần với cách suy nghĩ và hành động của con người [14].

- Nhược điểm: Thuật toán Heuristic chỉ dừng tại kết quả mà thuật toán cho là tốt, song có thể bỏ qua những kết quả khác còn tốt hơn, thậm chí tốt nhất; khó đáp ứng yêu cầu khi bài toán ngày càng lớn; khi thêm bớt điều kiện hoặc đầu vào công việc cần tinh chỉnh lại thuật toán cho phù hợp với cấu hình, gây mất thời gian [14].

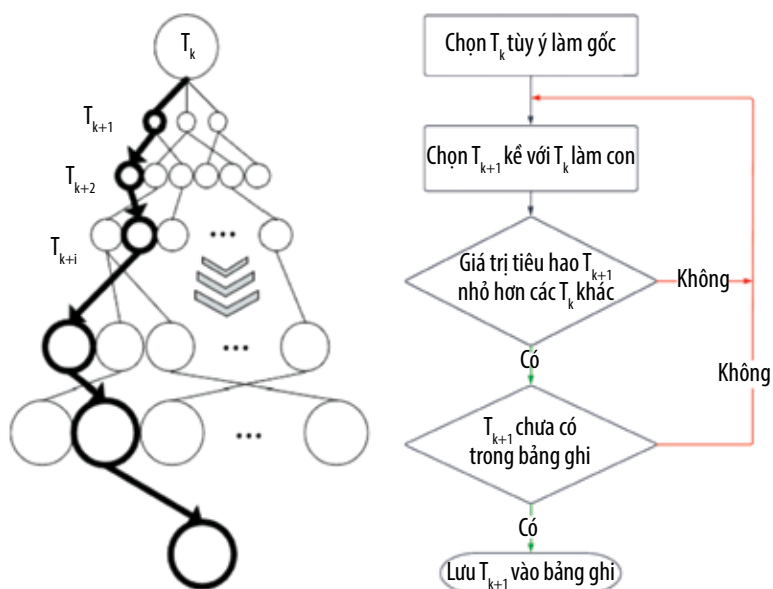
Khi lập kế hoạch, các dữ liệu công việc trước và sau đều có mối quan hệ chặt chẽ. Do đó, các công việc không hiệu quả hoặc gây khó khăn về sau trong tập hợp sẽ bị loại bỏ ngay từ đầu. Đây chính là nhiệm vụ phù hợp của thuật toán di truyền sử dụng thuật toán Heuristic nhằm tìm ra thế hệ thỏa mãn bài toán.

Do bản chất của các công việc không thể biết trước, không có giải pháp chính xác hoặc công thức chưa được biết, thuật toán Heuristic ưu tiên thời gian hoàn thành nhanh nhất cho toàn bộ công việc và hạn chế tối thiểu vòng lặp, đồng thời, là công cụ mạnh mẽ để tính toán tiệm cận các giải pháp theo tiệm cận nhánh (branch and bound), cho phép tìm độ rộng công việc.

Một số bài toán áp dụng thuật toán Heuristic vẫn chưa xây dựng được hướng đi ưu việt. Một số bài toán xây dựng được thuật toán song không thể áp dụng được do không đủ tài nguyên để cung cấp. Các vấn đề như vậy có thể giải quyết đơn giản bằng cách thu thập thêm dữ liệu mà không tốn tài nguyên xử lý hoặc phải xem xét các vấn đề nội bộ xảy ra khi áp dụng thuật toán.

2.3. Thuật toán di truyền

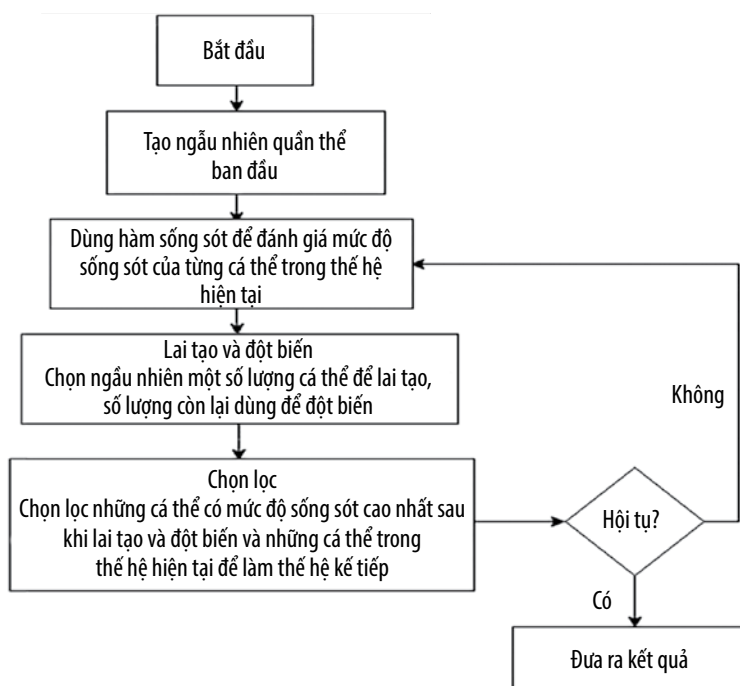
Thuật toán di truyền (genetic algorithm, GA) được phân lớp thuật toán tiến hóa và lấy cảm hứng từ quá trình chọn lọc tự nhiên. Trong bối cảnh thứ-và-sai ngẫu nhiên (random



Hình 2. Lưu đồ của thuật toán Heuristic giải quyết bài toán Traveling salesman đơn giản hóa [11].

Bảng 1. So sánh thuật ngữ tiến hóa và thuật ngữ giải quyết vấn đề dưới góc nhìn tiến hóa

Tiến hóa (Evolution)	Giải quyết vấn đề (Problem solving)
Môi trường (environment)	Vấn đề (problem)
Cá thể (individual)	Giải pháp (candidate solution)
Mức sống sót (fitness)	Chất lượng (quality)



Hình 3. Lưu đồ phương hướng lập thời gian biểu công việc bằng cách sử dụng thuật toán di truyền.

trial-and-error), hay còn được gọi là quy trình giải quyết vấn đề theo kiểu tạo và thử nghiệm, sẽ tồn tại tập hợp những cá thể có mức độ thích nghi và sống sót cao nhất. Chất lượng xác định cơ hội các cá thể sẽ được lưu giữ và sử dụng làm hạt giống để xây dựng các thế hệ tiếp theo.

2.3.1. Các thành phần trong thuật toán di truyền

Những thuật ngữ cơ bản trong thuật toán di truyền:

- **Quần thể (population):** Một quần thể ban đầu sẽ bao gồm các cá thể có các đặc tính khác nhau, những đặc tính này dùng để quy định khả năng sinh tồn của cá thể;
- **Lai tạo (crossover):** Phương pháp dùng 2 cá thể ngẫu nhiên để lai tạo nhằm tạo ra cá thể có độ thích nghi với môi trường tốt hơn thế hệ trước;
- **Đột biến (mutation):** Phương pháp thay đổi ngẫu nhiên một phần của gen trong một cá thể trong quần thể. Việc này giúp tạo ra sự đa dạng gen trong quần thể, làm tăng khả năng tìm kiếm cá thể có độ thích nghi tốt với môi trường;
- **Chọn lọc (selection):** Giữ lại các cá thể ưu việt thích nghi tốt với môi trường tự nhiên và loại bỏ các cá thể yếu kém;
- **Mức sống sót (fitness function):** Để đánh giá mức độ thích ứng của cá thể đối với môi trường.

2.3.2. Quá trình xử lý thuật toán di truyền

Thuật toán di truyền sẽ tạo ngẫu nhiên quần thể ban đầu (initial population) có số lượng cá thể (individual) cho trước. Trong quá trình thực thi, cá thể nào có mức độ thích nghi với môi trường cao hơn, phát triển hơn (có độ sống sót cao) thì sẽ được giữ lại, đồng thời loại bỏ các cá thể yếu kém. Thuật toán di truyền giải quyết bài toán theo thứ tự:

Lai tạo (crossover) → Đột biến (mutation) → Chọn lọc (selection)

Điều kiện tiên quyết để thực hiện thuật toán di truyền là phải xác định được quần thể ban đầu cũng như đề xuất ra được hàm mục tiêu thích hợp nhằm giải quyết bài toán tối ưu. Đối với bài toán lập thời gian biểu cho công việc hiện tại - sắp xếp kế hoạch cho từng nhóm với số lượng nhiệm vụ cho trước, lưu đồ cho hướng giải quyết như Hình 3.

3. Phân tích dữ liệu bảo trì, bảo dưỡng của Bien Dong POC

3.1. Bảng dữ liệu danh mục công việc

Tập dữ liệu danh mục công việc được cung cấp gồm 37 cột, trong đó có 8 cột mang thông tin quan trọng được sử dụng cho thuật toán di truyền bao gồm:

- **WONUM (work order number):** Mã số định danh công việc, ví dụ như H13779495;
- **PARENT:** Cờ định danh công việc bao hàm toàn bộ công việc khác (hay còn gọi là công việc lớn sẽ bao gồm nhiều công việc nhỏ và thời gian diễn ra công việc nhỏ phải được thực hiện xong trong thời gian của công việc lớn chứa nó);
- **PRIORITY:** Mức độ ưu tiên của công việc, bao gồm 4 mức độ chính: khẩn cấp (urgent), cao (high), trung bình (medium), thấp (low);
- **SITE:** Địa điểm làm việc, có 2 địa điểm chính: Hải Thạch (HT) và Mộc Tinh (MT);
- **ESTDUR:** Thời gian hoàn thành công việc tương ứng với WONUM cùng hàng. Chỉ số này được tính theo giờ;
- **TARGSTARTDATE:** Mốc thời gian bắt đầu công việc (được ghi theo định dạng tháng/ngày/năm giờ/phút/giây);

- TARGCOMPDATE: Mốc thời gian hoàn thành công việc (được ghi theo định dạng tháng/ngày/năm giờ/phút/giây);

- BDPOCDISCIPLINE: Nhóm sẽ thực hiện WONUM tương ứng cùng hàng, bao gồm 4 nhóm chính là: E&I, PROD, MECH và RES.

Một phần của bảng dữ liệu công việc trong khoảng thời gian 1 tháng được trích xuất cho mục đích nghiên cứu thuật toán. Theo đó, số lượng công việc cần phải được lập kế hoạch là 53 công việc. Một phần của bảng dữ liệu danh mục công việc được thể hiện tại Hình 4.

3.2. Bảng dữ liệu về nguồn lực

Tập dữ liệu được cung cấp thể hiện nhân công làm việc (resource) tại mỗi địa điểm Hải Thạch và Mộc Tinh, trong đó có 4 cột mang thông tin quan trọng sử dụng cho thuật toán di truyền bao gồm:

- DATE: Ngày làm việc (với định dạng tháng/ngày/năm);

- BDPOCDISCIPLINE: Nhóm sẽ nhận việc tương ứng với dữ liệu của cột DATE, bao gồm 4 nhóm đã liệt kê ở phần trên;

- MANDAY_HT: Số nhân công của nhóm tương ứng cùng hàng dữ liệu trên giàn khai thác Hải Thạch;

- MANDAY_MT: Số nhân công của nhóm tương ứng cùng hàng dữ liệu trên giàn khai thác Mộc Tinh.

Một phần của bảng dữ liệu về nguồn lực được thể hiện tại Hình 5.

Date	HT	MT	Bdpocdiscipline
1/3/2022	4	1	DECK
1/3/2022	4	1	MECH
1/3/2022	11	1	PROD
1/3/2022	3	0	RES
1/3/2022	6	2	E&I
2/3/2022	4	1	DECK
2/3/2022	4	1	MECH

Hình 5. Một phần bảng dữ liệu về nguồn lực.

Wonum	Description	Status	Worktype	Location	Estdur	Bdpocdiscipline	Targstartdate	Targcompdate
H13828313	36M - RT-HT1-90XX-EI-0009/10 - PAGA beacon loop 01/02B- Visual check, preservation	SCHED	PM	HT1-PAGB	24	E&I	3/1/2022	3/31/2022
H13828335	1M-PM FREE FALL LIFEBOAT, Free Fall Lifeboat, HT-SCF-8601B	SCHED	PM	HT-SCF-8601B	20	MECH	3/1/2022	3/31/2022
H13828407	6M-CFT & Calibration Fire and Gas Sys at PQP-Cellar Deck	SCHED	PM	HT-CD-FG	90	E&I	3/1/2022	3/31/2022
H13829771	6M-CFT & Calibration Fire and Gas Sys at PQP-Helli Deck & LQ Roof	SCHED	PM	HT-A01	36	E&I	3/1/2022	3/31/2022
H13829908	6M PM for Radio System MT1	SCHED	PM	MT1-RADIO	30	E&I	3/1/2022	3/31/2022
H13830695	12 months PM devices for system 86A- PQP Lifeboat	SCHED	PM	HT-SCF-8601B	20	E&I	3/1/2022	3/31/2022
H13830851	12 months PM devices for system 86B- PQP Lifeboat	SCHED	PM	HT-SCF-8601A	20	E&I	3/1/2022	3/31/2022
H13831013	12M-PM Shutoff/ Control Valve of Gas Turbine Generator 1	SCHED	PM	NULL	8	MECH	3/1/2022	3/31/2022
H13831015	3M PM LQ Food Container, LQ Freezer PQP, HT-LQ-CONT	SCHED	PM	HT-LQ-CONT	4	MECH	3/1/2022	3/31/2022
H13831410	PM and Calibration Portable Gas Detector	SCHED	PM	HT-GO-OFFICE	24	E&I	3/3/2022	3/31/2022
H13831566	3M-PM HVAC GO UNIT, HVAC Control Panel (General Office Building), HT-CH-7161A	WMATL	PM	HT-CH-7161A	40	MECH	3/4/2022	3/31/2022
H13831741	6M PM LQ Refrigeration Equipment, Galley equipment, HT-GL	WMATL	PM	HT-GL	22	MECH	3/8/2022	3/31/2022
H13831779	1M PM PQP Crane, PQP South Crane, HT-CR-7301	WMATL	PM	HT-CR-7301	40	MECH	3/11/2022	3/31/2022
H13831792	12M - Maintenance and EX inspection for electrical equipment of Air Compressor B - System 61	SCHED	PM	HT-K-6110B	24	E&I	3/13/2022	3/31/2022

Hình 4. Một phần bảng dữ liệu danh mục công việc.

4. Xây dựng thuật toán lập thời gian biểu công việc dựa trên bộ dữ liệu

4.1. Các bước xây dựng thuật toán

Dựa trên tập dữ liệu quan sát được, thuật toán cần thành lập thời gian biểu sao cho mỗi công việc phải được hoàn thành trong khoảng thời gian cho trước, số lượng nhân công yêu cầu cho 1 ngày phải nhỏ hơn số nhân công được cho sẵn và các công việc yêu cầu phải làm liên tiếp nhau. Từ mục tiêu để ra, bài toán lập thời gian biểu cho công việc bao gồm 3 bước chính:

- Bước 1 - Xử lý và mã hóa: Tiến hành chuyển các dữ liệu cần tìm thành dạng chuỗi bit;
- Bước 2 - Sử dụng thuật toán di truyền: Sử dụng các phương pháp lai tạo và đột biến nhằm hướng đến kết quả tốt nhất;
- Bước 3 - Đánh giá kết quả: Sau khi chạy xong thuật toán di truyền, so sánh mức sống sót cao nhất qua từng epoch, đồng thời thống kê tổng số lỗi vi phạm điều kiện (output number of errors).

4.2. Xử lý và mã hóa dữ liệu

a, Đối với cột TARGSTARTDATE và TARGCOMPDATE: Cắt bỏ thông tin giờ/giây/phút sau dấu tháng/ngày/năm bởi thông tin này có thể hiển thị hoặc không trên một số dữ liệu. Sau đó, tiến hành chuyển đổi định dạng thông tin lịch biểu từ tháng/ngày/năm sang thành ngày/tháng/năm. Cuối cùng, cách thể hiện năm được quy định năm hiện tại là 0001 và năm kế tiếp là 0002. Các bước thể hiện quá trình trên được biểu diễn ở Hình 6.

Tất cả dữ liệu thuộc vùng thời gian tương ứng của bảng dữ liệu về nguồn lực được khoanh vùng và lọc ra. Gọi dR là tập miêu tả vùng này, End of File - EOF là hàng cuối tập tin bảng tính, cần lọc ra thông tin các hàng r với bộ lọc f thỏa mãn điều kiện (1).



Hình 6. Biến đổi định dạng thời gian 2 cột TARGSTARTDATE và TARGCOMPDATE.

Bảng 2. Ánh xạ cấp độ bằng từ ngữ qua thể hiện bằng chữ số

Cấp độ miêu tả bằng chữ	Cấp độ miêu tả bằng số
Gấp (URGENT)	3
Cao (HIGH)	2
Trung bình (MEDIUM)	1
Thấp (LOW)	0

wonum	parent	priority	site	bdpocdiscipline	estdur	targstartdate	targcompdate	r_estdur	
0	H13779495	NaN	0	HT	MECH	20.0	01/03/0002	31/03/0002	2
1	H13824912	NaN	1	HT	MECH	20.0	01/03/0002	31/03/0002	2
2	H13824914	NaN	1	HT	PROD	1.0	01/03/0002	31/03/0002	1
3	H13824916	NaN	1	HT	E&I	12.0	01/03/0002	31/03/0002	2
4	H13824920	H13824916	1	HT	E&I	2.0	01/03/0002	31/03/0002	1

Hình 7. Một phần bảng dữ liệu sau xử lý.

Bảng 3. Các thành phần thuật toán di truyền và cách biểu diễn tương ứng

Thành phần	Biểu diễn
Đặc trưng (representation)	Chuỗi bit (Bit-strings)
Chọn cha mẹ (parent selection)	Phương pháp chọn ngẫu nhiên (Random selection)
Kết hợp lai (recombination)	Phương pháp lai tạo (crossover)
Đột biến (mutation)	Phương pháp lật bit (bit-flip)
Chọn kế thừa (survival selection)	Mức sống sót (fitness proportional)

$$dR = \{3/1/2022 \rightarrow 31/5/2022\}$$

$$\forall \{r \in (TARGSTARTDATE, TARGCOMPDATE), 0 < r \leq EOF\}, (1)$$

$$f: (r \in dR \rightarrow store) \wedge (r \notin dR \rightarrow remove)$$

b, Đối với cột EST_DUR: Thông tin EST_DUR được chuyển đổi số giờ hoàn thành công việc về số ngày hoàn thành công việc, với nguyên tắc quy đổi 1 ngày làm việc là 10 giờ (1 ngày = 10 giờ).

c, Đối với cột PRIORITY: Phép ánh xạ mức độ bằng các con số được thực hiện giúp cho việc sắp xếp độ quan trọng theo thứ tự nhất định để có thể dễ dàng phát hiện việc nào làm trước, việc nào làm sau. Cách ánh xạ phân loại được thể hiện qua Bảng 2.

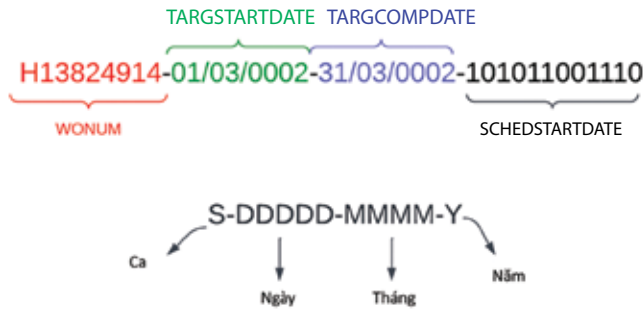
Sản phẩm của quá trình xử lý dữ liệu thô là bảng dữ liệu con cô đọng hơn, được sắp xếp tuần tự, thông tin được biểu diễn linh hoạt và logic. Một phần của bảng dữ liệu con này được thể hiện tại Hình 7.

4.3. Sử dụng thuật toán GA

Đối với bài toán lập thời gian biểu cho công việc hiện tại, từng phương pháp ứng với mỗi thành phần của thuật toán di truyền được thể hiện trong Bảng 3.

4.3.1. Đặc trưng (representation)

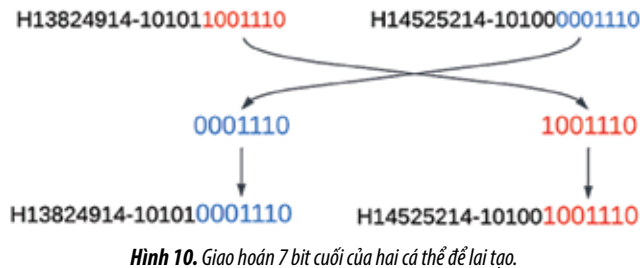
Dựa trên sự quan sát trên tập dữ liệu cho trước, mỗi cá thể (chromosome) sẽ bao gồm tập hợp các nhiệm vụ được phân phối cho một nhóm xác định. Hình 8 biểu diễn chuỗi cá thể đã được mã hóa với các thông số sau:



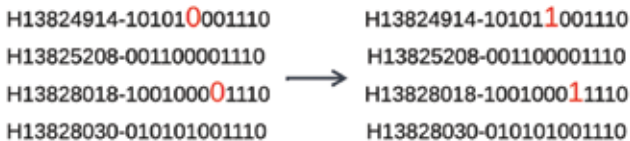
Hình 8. Các thành phần mã hóa của một cá thể dữ liệu được đặt ra.



Hình 9. Chọn ngẫu nhiên 2 trong 3 cá thể vào nguồn giao phối.



Hình 10. Giao hoán 7 bit cuối của hai cá thể để lai tạo.



Hình 11. Kỹ thuật lật bit chuyển bit 0 thành bit 1.

- WONUM: Dùng để phân biệt từng nhiệm vụ riêng biệt;
- TARGSTARTDATE và TARGCOMPDATE: Là thông số thời gian bắt đầu và kết thúc công việc để tính độ fitness;
- SCHEDSTARTDATE: Là thông số thời gian được lên kế hoạch để thực hiện công việc và được biểu diễn bằng chuỗi bit.

Cấu trúc chuỗi SCHEDSTARTDATE bao gồm 12 bit, với các cụm bit được định dạng như sau:

- 1 bit đầu tiên thể hiện thời gian ca làm việc của ngày. Trong đó, 0 là ca sáng, 1 là ca chiều;

- 5 bit tiếp theo để thể hiện số thứ tự ngày của tháng (thể hiện 30 hoặc 31 ngày).

- 4 bit tiếp theo thể hiện số thứ tự tháng hiện tại;
- 1 bit cuối cùng thể hiện năm của công việc đang làm (0 là năm hiện tại và 1 là năm kế tiếp). Quy định năm hiện tại là 0001 và năm kế tiếp là 0002;

Khi thực hiện lai tạo hay đột biến, từng chuỗi bit trong mỗi nhiệm vụ được thực thi nhằm tạo ra những cá thể mới (chuỗi bit). Chuỗi bit của mỗi công việc sẽ được giải mã (decode) ra ngày, tháng, năm thực tế sau khi chạy xong thuật toán.

4.3.2. Chọn cha mẹ (parent selection)

Ở thành phần này, các cá thể không trùng nhau được lựa chọn một cách ngẫu nhiên để làm nguồn giao phối (mating pool) cho bước kết hợp lai (recombination). Vì trong lúc lai tạo sẽ có xác suất tạo ra các cá thể có độ sống sót cao hơn so với thế hệ cũ nên không cần chọn nguồn nguyên liệu để lai tạo có mức sống sót cao. Thay vào đó, việc chọn ngẫu nhiên sẽ cho 1 nguồn gen có thể sinh sản ra thế hệ con có mức sống sót tối ưu hơn.

4.3.3. Kết hợp lai (recombination)

Có nhiều phương pháp khác nhau để thực hiện việc lai tạo, ví dụ như lai tạo một điểm (one-point crossover), lai tạo n-điểm (n-point crossover) hoặc lai tạo đồng nhất (uniform crossover).

Kỹ thuật lai tạo 1 điểm được thực hiện bằng cách chọn ngẫu nhiên 1 nhiệm vụ trong 2 cá thể được chọn để lai tạo. Sau đó, chọn ngẫu nhiên một vị trí của chuỗi bit trong cả 2 nhiệm vụ vừa được chọn và giao hoán đuôi của cả 2 chuỗi bit cho nhau tại vị trí đó (Hình 10).

4.3.4. Đột biến (mutation)

Đột biến là quá trình tạo ra thế hệ con bởi một cá thể thay vì cần đến 2 cá thể như phương pháp kết hợp lai. Trong quá trình đột biến, cá thể sẽ ngẫu nhiên thay đổi bất kỳ đoạn nào ở mã gen, ở đây là thay đổi chuỗi bit. Phương pháp lật bit (bit-flip) cho chuỗi nhị phân được sử dụng trong nghiên cứu này để làm giàu tính đa dạng cho thế hệ sau. Quá trình đột biến được minh họa ở Hình 11.

4.3.5. Chọn kế thừa (survival selection)

Ở bước này, thế hệ con mới có độ thích ứng với môi trường cao hơn so với thế hệ cũ được chọn ra. Việc chọn ra kết quả tối ưu bằng cách xác định độ sống sót cho từng cá thể đã lai tạo, đột biến và từ thế hệ cũ. Từ đó, chọn được thế hệ mới từ những cá thể có độ sống sót cao nhất.

Bảng 4. Bộ ngũ đầu vào, thời gian thực thi và độ sống sót tối đa ghi nhận được

(IPS, n _p , MR, E, T)	Thời gian thực thi (giờ)	Độ sống sót tối đa
(100,50,0,1,20,PROD)	477	0,0909
(200,100,0,1,20,PROD)	952	0,0909
(300,150,0,1,20,PROD)	1727	1,0000

WONUM	SCHEDSTARTDATE	SHIFT
H13830315	24/3/2022	1
H13830317	22/3/2022	1
H13830319	22/3/2022	1
H13830321	5/3/2022	0
H13830323	17/3/2022	1
H13830325	6/3/2022	0
H13831025	10/3/2022	0
H13831382	20/3/2022	1
H13831386	13/3/2022	0
H13831388	3/3/2022	0
H13831390	6/3/2022	0
H13831392	22/3/2022	1
H13831394	6/3/2022	0
H13831396	4/3/2022	0

Hình 12. Bảng dữ liệu kết quả đầu ra.

4.3.6. Mức sống sót (fitness function)

Mức sống sót là thang đo để tính mức độ thích ứng của từng chromosome với môi trường (ở đây là những điều kiện được đặt ra). Mức sống sót càng cao (hay thấp tùy vào hàm sống sót đã định nghĩa) thì thể hiện rằng cá thể đó thích nghi với môi trường càng tốt hoặc xấu. Trong hàm sống sót, các điều kiện được định nghĩa rõ và phân chia thành 2 loại như sau:

- Điều kiện cứng (hard constraint) là điều kiện mà cá thể không được phép vi phạm. Trong bài toán đang xét, thời gian SCHEDSTARTDATE (thời gian dự định bắt đầu thực hiện công việc) không được nằm ngoài khoảng thời gian cho trước (nằm ngoài TARGSTARTDATE và TARGCOMPDATE);
- Điều kiện mềm (soft constraints) là những điều kiện mà cá thể được phép vi phạm nhưng hạn chế ở mức độ thấp nhất để vẫn đảm bảo tối ưu kết quả. Ví dụ, khoảng thời gian giữa 2 nhiệm vụ liên tiếp không được cách nhau lớn hơn một khoảng cho trước.

Hàm fitness được định nghĩa như sau:

$$f(HC, SC) = \frac{1}{HC \times 10 + SC \times 3 + 1} \quad (2)$$

Trong đó:

HC: Điều kiện cứng;

SC: Điều kiện mềm.

Theo phương trình (2), độ sống sót càng cao thì kết quả càng tối ưu.

4.4. Đánh giá kết quả

Đối với việc xây dựng bài toán lập thời gian biểu cho công việc, những biến đầu vào cần điều chỉnh là:

- T (team): Những nhóm để lập thời gian biểu cho công việc như MECH, PROD, E&I;
- IPS (initial population size): Tổng số lượng cá thể ban đầu;
- n_p (number of parents): Số lượng cá thể dùng cho việc lai tạo;
- MR (mutation rate): Tỷ lệ đột biến trên từng cá thể;
- E (epoch): Số vòng lặp để chạy mô hình.

Dữ liệu trên sẽ tạo thành một bộ ngũ (5-tuple) có dạng (IPS, n_p, MR, E, T). Bảng 4 thể hiện kết quả của thuật toán cho các công việc của nhóm PROD với các bộ ngũ tương ứng. Có thể thấy, thuật toán có khả năng lập thời gian biểu cho công việc tốt với kết quả sống sót cao.

Kết quả đầu ra của thuật toán có thể được trích xuất thành bảng dữ liệu, với các cột mang thông tin gồm:

- WONUM: Mã số định danh công việc;
- SCHEDSTARTDATE: Thời gian mà thuật toán gợi ý bắt đầu thực hiện công việc tương ứng;
- SHIFT: Ca làm việc trong ngày mà thuật toán gợi ý có thể bắt đầu thực hiện công việc tương ứng, trong đó 0 là ca sáng và 1 là ca chiều.

Một phần của bảng dữ liệu đầu ra của thuật toán được thể hiện trong Hình 12. Dựa vào gợi ý của thuật toán, Bien Dong POC có thể nhanh chóng lập thời gian biểu cho công tác bảo dưỡng.

5. Kết luận

Nghiên cứu tập trung vào việc áp dụng thuật toán di truyền để lập kế hoạch cho công tác bảo trì bảo dưỡng, được thử nghiệm trên bộ dữ liệu bảo trì bảo dưỡng thu thập từ hệ thống quản lý trên các giàn khai thác của Bien Dong POC. Bộ dữ liệu được xử lý và mã hóa các điều kiện, gồm thời hạn phải thực thi, mức độ ưu tiên, nguồn lực, để chuyển đổi thành dạng chuỗi bit. Sau đó, thuật toán di truyền được áp dụng trên bộ dữ liệu đã được mã hóa, sử dụng phương pháp lai tạo và phương pháp đột biến để tạo ra kết quả tối ưu nhất.

Kết quả nghiên cứu cho thấy thuật toán có khả năng sắp xếp kế hoạch chính xác với độ khớp cao, có thể xử lý

nhiều điều kiện về thời gian, mức độ ưu tiên và nguồn lực cho từng công việc. Do đó, có thể triển khai các công cụ tự động sử dụng thuật toán này để hỗ trợ công tác lập kế hoạch bảo trì bảo dưỡng, từ đó giúp tối ưu nguồn lực, nâng cao hiệu suất và độ chính xác của quy trình quản lý bảo trì, bảo dưỡng.

Tài liệu tham khảo

[1] Tran Vu Tung, Tran Ngoc Trung, Ngo Huu Hai, and Nguyen Thanh Tinh, "Digital transformation in oil & gas company - A case study of Bien Dong POC", *Petrovietnam Journal*, Volume 10, pp. 67 - 78, 2020.

[2] Trần Ngọc Trung, Trần Vũ Tùng, Hoàng Kỳ Sơn, Ngô Hữu Hải, và Đào Quang Khoa, "Thực tiễn triển khai nền tảng số hóa tập trung tại mỏ Hải Thạch - Mộc Tinh", *Tạp chí Dầu khí*, Số 12, trang 47 - 56, 2020.

[3] M. Alardhi and A.W. Labi, "Preventive maintenance scheduling of multi-cogeneration plants using integer programming", *Journal of the Operational Research Society*, Volume 59, Issue 4, pp. 503 - 509, 2008. DOI: 10.1057/palgrave.jors.2602386.

[4] Jae Kyu Lee and Sunn Beom Kwon, "An expert systems development planner using a constraint and rule-based approach", *Expert Systems with Applications*, Volume 9, Issue 1, pp. 3 - 14, 1995. DOI: 10.1016/0957-4174(94)00043-U.

[5] R. Pereira, J.L. Sánchez, and J. Rives, "Knowledge-based maneuver and fire support planning", *Expert Systems with Applications*, Volume 17, Issue 2, pp. 77 - 87, 1999. DOI: 10.1016/S0957-4174(99)00025-1.

[6] Trần Đức Huyền, *Phương pháp giải các bài toán trong tin học*. Nhà xuất bản Giáo dục, 1999.

[7] Q. Larson, "Brute force algorithms explained", 6/1/2020. [Online]. Available: <https://www.freecodecamp.org/news/brute-force-algorithms-explained/>.

[8] Bhavya, "Divide and conquer algorithm in Python". [Online]. Available: <https://pythonwife.com/divide-and-conquer-algorithm-in-python/>.

[9] M. Manley, "Divide and conquer paradigm", *ACM Transactions on Computational Logic*, 2014.

[10] Gerd Gigerenzer, Ralph Hertwig, and Thorsten Pachur, *Heuristics: The foundations of adaptive behavior*. Oxford University Press, 2011. DOI: 10.1093/acprof:oso/9780199744282.001.0001.

[11] Yang Li, Junbin Gao, Mingyuan Bai, Chengjun Li, and Gang Liu, "A Heuristic algorithm based on tour rebuilding operator for the traveling salesman problem". DOI: 10.48550/arXiv.1911.01663.

[12] Gerd Gigerenzer and Wolfgang Gaissmaier, "Heuristic decision making", *Annual Review of Psychology*, Volume 62, Issue 1, pp. 451 - 482, 2011. DOI: 10.1146/annurev-psych-120709-145346.

[13] Stephen Dale, "Heuristics and biases: The science of decision-making", *Business Information Review*, Volume 2, Issue 2, pp. 93 - 99, 2015. DOI: 10.1177/0266382115592536.

[14] Zbigniew Michalewicz and David B. Fogel, *How to solve it: Modern heuristics*. Springer, 2013. DOI: 10.1007/978-3-662-04131-4

DEVELOPING A TIME SCHEDULING ALGORITHM FOR MAINTENANCE TASKS: A CASE STUDY AT HAI THACH - MOC TINH FIELDS

Dao Quang Khoa, Tran Ngoc Trung, Nguyen Thanh Trung, Tran Vu Tung, Hoang Ky Son, Ngo Huu Hai

Bien Dong Petroleum Operating Company (Bien Dong POC)

Email: trungtn@biendongpoc.vn

Summary

Using dataset sourced from the maintenance management application of Bien Dong Petroleum Operating Company (Bien Dong POC), the authors studied to employ genetic algorithms to schedule maintenance tasks, thereby shortening planning time, optimizing the involved resources, ensuring the consistency of the maintenance planning and implementation. Findings indicate that genetic algorithms effectively generate schedules with a high degree of accuracy, addressing multiple constraints related to time, priority, and resources for each task code, while significantly reducing computation time compared to manual scheduling methods. The study manifests the potential for implementing tools that automatically deploy such algorithms to support engineers, enhancing the efficiency and precision of maintenance planning and management process, especially at large-scale production facilities.

Key words: Genetic algorithm, maintenance scheduling.