

**BỘ XÂY DỰNG
TRƯỜNG ĐẠI HỌC XÂY DỰNG MIỀN TRUNG**



Nguyễn Xuân Thành

**NGHIÊN CỨU ỨNG DỤNG
THUẬT TOÁN TỐI ƯU HÓA ĐA MỤC TIÊU TRONG
BỐ TRÍ TỔNG MẶT BẰNG THI CÔNG**

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Ngành Kỹ thuật Xây Dựng

Mã số: 8580201

**Cán bộ hướng dẫn chính: TS. Đặng Bảo Lợi
Cán bộ hướng dẫn phụ: ThS.NCS. Phạm Duy Hiếu**

Đắk Lắk - 2025

LỜI CẢM ƠN

Tác giả xin trân trọng cảm ơn các thầy cô giáo, các cán bộ phòng Quản lý Đào tạo, khoa Xây Dựng, đã tạo mọi điều kiện trong suốt quá trình học tập và nghiên cứu.

Tác giả xin gửi lời cảm ơn tới bạn bè và đồng nghiệp đã góp ý, giúp đỡ trong quá trình thực hiện và hoàn thành đề án.

Tác giả xin bày tỏ lòng biết ơn sâu sắc tới gia đình và người thân đã hỗ trợ và tin tưởng tôi trong suốt quá trình thực hiện đề án.

Đặc biệt, tác giả xin trân trọng cảm ơn TS. Đặng Bảo Lợi và ThS.NCS. Phạm Duy Hiếu đã nhiệt tình giúp đỡ, hướng dẫn, tạo mọi điều kiện thuận lợi nhất trong quá trình thực hiện và hoàn thành đề án.

TÁC GIẢ ĐỀ ÁN



Nguyễn Xuân Thành

LỜI CAM ĐOAN

Tôi xin cam đoan Đề án thạc sĩ này là công trình nghiên cứu khoa học độc lập của tôi. Các số liệu khoa học, kết quả nghiên cứu của Đề án là trung thực và có nguồn gốc rõ ràng.

TÁC GIẢ ĐỀ ÁN



Nguyễn Xuân Thành

MỤC LỤC

DANH MỤC CÁC BẢNG.....	I
DANH MỤC CÁC HÌNH VẼ.....	II
MỞ ĐẦU.....	1
1. Lý do chọn đề tài	1
2. Mục đích nghiên cứu	1
3. Đối tượng nghiên cứu và phạm vi nghiên cứu	1
4. Phương pháp nghiên cứu	2
5. Cơ sở khoa học, thực tiễn của đề tài.....	2
6. Kết quả đạt được và vấn đề còn tồn tại.....	2
Chương 1 - TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU.....	3
1.1. Giới thiệu đề tài nghiên cứu	3
1.2. Tổng quan các nghiên cứu trước đây.....	4
1.3. Cơ sở lý thuyết các thuật toán tối ưu hóa	6
1.3.1. Thuật toán Genetic Algorithms (GA).....	6
1.3.2. Thuật toán Non-Dominated Sorting Genetic Algorithm II (NSGA-II).....	8
1.3.3. Thuật toán Particle Swarm Optimization (PSO)	10
1.3.4. Thuật toán Multi-Objective Particle Swarm Optimization (MOPSO)	13
1.3.5. Thuật toán NSGA-II lai MOPSO	14
1.4. Phương pháp nghiên cứu	16
1.5. Tổng kết chương 1	17
Chương 2 - XÂY DỰNG MÔ HÌNH TỐI ƯU HÓA MẶT BẰNG THI CÔNG.....	19
2.1. Khái quát các hạng mục phụ trợ	19
2.2. Các giả định của mô hình	20

2.3. Các ràng buộc của mô hình	21
2.3.1. Ràng buộc về phạm vi khu đất	21
2.3.2. Ràng buộc về phạm vi hoạt động cần trục tháp	22
2.3.3. Ràng buộc trùng lặp và khoảng cách tối thiểu giữa các hạng mục	23
2.4. Các hàm mục tiêu của mô hình.....	23
2.4.1. Hàm mục tiêu tối ưu hóa khoảng cách di chuyển	23
2.4.2. Hàm mục tiêu tối ưu hóa an toàn lao động.....	24
2.4.2.1. An toàn từ hoạt động cần trục tháp <i>CSC</i>	25
2.4.2.2. An toàn từ hạng mục nguy hiểm <i>NHCC</i>	27
2.4.2.3. An toàn tại giao điểm tuyến đường đông đúc <i>IPC</i>	28
2.5. Khái niệm tối ưu Pareto	29
2.6. Mô hình hóa bài toán bằng phần mềm MATLAB.....	30
2.7. Tổng kết chương 2	30
Chương 3 - ỨNG DỤNG MÔ HÌNH TỐI ƯU HÓA MẶT BẰNG THI CÔNG.....	32
3.1. Áp dụng mô hình cho công trình 1	32
3.2. Áp dụng mô hình cho công trình 2	41
3.3. Áp dụng mô hình cho công trình 3	50
3.4. Tổng kết chương 3	59
KẾT LUẬN VÀ KIẾN NGHỊ.....	61
TÀI LIỆU THAM KHẢO.....	62
PHỤ LỤC 01	64
PHỤ LỤC 02	85

DANH MỤC CÁC BẢNG

Bảng 1.1. So sách thuật toán GA và NSGA-II.....	10
Bảng 2.1. Trọng số về mức độ gần gũi w_{ij}	24
Bảng 2.2. Trọng số kiểm soát rủi ro HCW_{ij}	28
Bảng 3.1. Thông tin các hạng mục của mô hình thi công 1.....	33
Bảng 3.2. Trọng số gần gũi giữa các hạng mục mô hình thi công 1.....	34
Bảng 3.3. Trọng số kiểm soát rủi ro mô hình thi công 1.....	35
Bảng 3.4. Hệ số mật độ di chuyển mô hình thi công 1.....	36
Bảng 3.5. Tham số đầu vào cho mô hình thi công 1.....	37
Bảng 3.6. Thông tin các hạng mục của mô hình thi công 2.....	42
Bảng 3.7. Trọng số gần gũi giữa các hạng mục mô hình thi công 2.....	43
Bảng 3.8. Trọng số kiểm soát rủi ro mô hình thi công 2.....	44
Bảng 3.9. Hệ số mật độ di chuyển mô hình thi công 2.....	45
Bảng 3.10. Tham số đầu vào cho mô hình thi công 2.....	46
Bảng 3.11. Thông tin các hạng mục của mô hình thi công 3.....	52
Bảng 3.12. Trọng số gần gũi giữa các hạng mục mô hình thi công 3.....	53
Bảng 3.13. Trọng số kiểm soát rủi ro mô hình thi công 3.....	54
Bảng 3.14. Hệ số mật độ di chuyển mô hình thi công 3.....	55
Bảng 3.15. Tham số đầu vào cho mô hình thi công 3.....	56

DANH MỤC CÁC HÌNH VẼ

Hình 1.1. Sơ đồ thuật toán GA.....	7
Hình 1.2. Sơ đồ thuật toán NSGA-II.....	9
Hình 1.3. Cảnh đàn chim tìm kiếm môi	11
Hình 1.4. Sơ đồ thuật toán PSO	11
Hình 1.5. Mô tả quá trình cập nhật vị trí và vận tốc [10].....	13
Hình 1.6. Sơ đồ thuật toán MOPSO.....	14
Hình 1.7. Sơ đồ thuật toán NSGA-II lai MOPSO	15
Hình 1.8. Sơ đồ phương pháp nghiên cứu	16
Hình 2.1. Ràng buộc khu đất.....	21
Hình 2.2. Ràng buộc phạm vi cần trục tháp.....	22
Hình 2.3. Ràng buộc trùng lặp hạng mục và khoảng cách tối thiểu	23
Hình 2.4. An toàn liên quan đến cần trục tháp.....	26
Hình 2.5. Các điểm giao nhau giữa các tuyến đường có lưu lượng di chuyển cao ..	29
Hình 2.6. Mô tả tối ưu Pareto.....	30
Hình 3.1. Mặt bằng mô hình thi công 1	32
Hình 3.2. Biểu đồ các giải pháp Pareto mô hình thi công 1	37
Hình 3.3. Mặt bằng mô hình 1, giải pháp số 1, thuật toán NSGA-II lai MOPSO	39
Hình 3.4. Mặt bằng mô hình 1, giải pháp số 19, thuật toán NSGA-II lai MOPSO ..	40
Hình 3.5. Mặt bằng mô hình 1, giải pháp số 9, thuật toán NSGA-II lai MOPSO	40
Hình 3.6. Mặt bằng mô hình thi công 2	41
Hình 3.7. Biểu đồ các giải pháp Pareto mô hình thi công 2	46
Hình 3.8. Mặt bằng mô hình 2, giải pháp số 1, thuật toán NSGA-II lai MOPSO	48
Hình 3.9. Mặt bằng mô hình 2, giải pháp số 22, thuật toán NSGA-II lai MOPSO ..	48
Hình 3.10. Mặt bằng mô hình 2, giải pháp số 10, thuật toán NSGA-II lai MOPSO ..	49
Hình 3.11 Mặt bằng thực tế bố trí thi công dự án Simona Heights	50
Hình 3.12 Mặt bằng mô phỏng bố trí thi công dự án Simona Heights	51
Hình 3.13. Biểu đồ các giải pháp Pareto mô hình thi công 3	56
Hình 3.14. Mặt bằng mô hình 3, giải pháp số 1, thuật toán NSGA-II lai MOPSO ..	58

III

Hình 3.15. Mặt bằng mô hình 3, giải pháp số 4, thuật toán NSGA- II lai MOPSO .58

Hình 3.16. Mặt bằng mô hình 3, giải pháp số 9, thuật toán NSGA-II lai MOPSO ..59

MỞ ĐẦU

1. Lý do chọn đề tài

- Trong xây dựng, thông thường bố trí tổng mặt bằng thi công mang tính định tính, dựa trên kinh nghiệm thi công, mối liên hệ giữa các đối tượng nhưng không đưa ra phân tích đánh giá cụ thể. Sự chính xác và hiệu quả của việc bố trí tổng mặt bằng thi công có thể ảnh hưởng đáng kể đến an toàn lao động, chi phí vận chuyển, tiến độ thi công.
- Khi công nghệ ngày càng phát triển, việc sử dụng phần mềm kết hợp với thuật toán để tối ưu hóa tổng mặt bằng thi công không chỉ giúp tăng hiệu quả mà còn phản ánh sự hiện đại và linh hoạt trong công trình xây dựng.
- Từ những lý do trên, học viên đã lựa chọn đề tài “**Nghiên cứu ứng dụng thuật toán tối ưu hóa đa mục tiêu trong bố trí tổng mặt bằng thi công**” để ứng dụng thuật toán Non-Dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Particle Swarm Optimization (MOPSO) cũng như thuật toán lai kết hợp trong việc bố trí tổng mặt bằng thi công, tối ưu hóa giải pháp, giúp nâng cao hiệu quả bố trí tổng mặt bằng, từ đó đẩy nhanh tiến độ thi công và an toàn cho công trình xây dựng nói chung.

2. Mục đích nghiên cứu

- Nghiên cứu ứng dụng của các thuật toán tối ưu hóa.
- So sánh mức độ hiệu quả, ứng dụng của thuật toán NSGA-II và NSGA-II lai MOPSO để tối ưu hóa việc bố trí tổng mặt bằng công trường với 2 mục tiêu là khoảng cách di chuyển và an toàn lao động.
- Từ cơ sở lý thuyết của thuật toán NSGA-II, MOPSO có thể áp dụng vào nhiều vấn đề tối ưu hóa khác trong tương lai.

3. Đối tượng nghiên cứu và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Tổng mặt bằng thi công công trình xây dựng, bao gồm: các hạng mục theo bản vẽ thiết kế, các hạng mục phụ trợ như cần trục tháp, bãi tập kết vật liệu, thiết bị, lán trại và mối quan hệ giữa các hạng mục này về khoảng cách di chuyển và an toàn lao động.

- Phạm vi nghiên cứu: Sử dụng thuật toán NSGA-II, NSGA-II lai MOPSO để tối ưu hóa việc bố trí tổng mặt bằng thi công bằng phần mềm MATLAB, đối với dạng công trình xây dựng dân dụng hình chữ nhật sử dụng phương pháp thi công truyền thống đổ tại chỗ và thi công lắp ghép, trong giai đoạn thi công phần thân.

4. Phương pháp nghiên cứu

- Phương pháp thu thập, nghiên cứu tài liệu về cách thức tối ưu của thuật toán NSGA-II, MOPSO và cách lập trình bằng phần mềm MATLAB.
- Phương pháp phân tích, so sánh, tổng hợp các kết quả nghiên cứu của các bài báo tối ưu tổng mặt bằng thi công.

5. Cơ sở khoa học, thực tiễn của đề tài

- Cơ sở khoa học: Thuật toán NSGA-II được phát triển bởi Kalyanmoy Deb và cộng sự [1], MOPSO được phát triển bởi Carlos A. Coello Coello, Maximino Salazar Lechuga [2], cả 2 thuật toán này đều được trình bày tại tổ chức công nghệ toàn cầu Institute of Electrical and Electronics Engineers (IEEE) vào năm 2002.
- Tính thực tiễn: Bằng cách áp dụng thuật toán tối ưu trong bố trí tổng mặt bằng thi công, có thể tối ưu hóa thời gian vận chuyển vật liệu xây dựng, thiết bị và an toàn lao động. Đặc biệt là trong các dự án xây dựng lớn, nơi mà quản lý tài nguyên và tối ưu hóa không gian là rất quan trọng. Ngoài ra, tổng mặt bằng thi công cũng có thể được điều chỉnh nhanh chóng để đáp ứng các thay đổi từ thiết kế trong quá trình thi công. Kết quả của nghiên cứu có thể được phát triển và áp dụng cho nhiều loại dự án xây dựng khác nhau, từ các dự án dân dụng đến các dự án công nghiệp và cơ sở hạ tầng.

6. Kết quả đạt được và vấn đề còn tồn tại

- Kết quả đạt được: Nghiên cứu đã thành công trong việc bố trí tổng mặt bằng thi công bằng cách áp dụng thuật toán tối ưu hóa đa mục tiêu. Kết quả cho thấy thuật toán NSGA-II lai MOPSO mang lại các giải pháp Pareto tốt hơn so với NSGA-II.
- Vấn đề còn tồn tại: Phạm vi áp dụng còn giới hạn cho các công trình có hình dạng chữ nhật, chỉ áp dụng mô phỏng tương tự cho các công trình có hình dạng phức tạp như đa giác không đều, đường cong.

Chương 1 - TỔNG QUAN ĐỀ TÀI NGHIÊN CỨU

1.1. Giới thiệu đề tài nghiên cứu

- Bố trí tổng mặt bằng thi công là một trong những yếu tố quan trọng quyết định hiệu quả của quá trình xây dựng. Một mặt bằng được sắp xếp khoa học không chỉ giúp tối ưu hóa không gian sử dụng mà còn góp phần giảm chi phí, nâng cao năng suất lao động và đảm bảo an toàn trên công trường. Tuy nhiên, do sự đa dạng về quy mô, địa hình và yêu cầu kỹ thuật của từng dự án, việc bố trí hợp lý các hạng mục phụ trợ như nhà kho, bãi vật liệu, bãi gia công hay cần trục vẫn là một bài toán phức tạp.
- Bố trí tổng mặt bằng thi công không chỉ đơn thuần là việc sắp đặt các hạng mục trên một khu vực cố định mà còn phải đáp ứng nhiều ràng buộc khác nhau, bao gồm khoảng cách di chuyển, khả năng tiếp cận, an toàn lao động và tiến độ thi công. Nếu mặt bằng không được bố trí hợp lý, các hệ lụy có thể xảy ra như lãng phí diện tích, phát sinh chi phí không cần thiết, giảm hiệu suất làm việc và tăng nguy cơ tai nạn lao động.
- Trong những năm gần đây, các phương pháp tối ưu hóa đã được áp dụng để giải quyết bài toán bố trí mặt bằng thi công. Các thuật toán tối ưu hóa đơn mục tiêu như Genetic Algorithm (GA) hay Particle Swarm Optimization (PSO) đã được sử dụng để tìm kiếm phương án bố trí hiệu quả dựa trên các tiêu chí nhất định. Tuy nhiên, do đặc thù của bài toán này liên quan đến nhiều yếu tố có thể mâu thuẫn với nhau, các thuật toán tối ưu đa mục tiêu như Non-Dominated Sorting Genetic Algorithm II (NSGA-II) hay Multi-Objective Particle Swarm Optimization (MOPSO) đang dần trở thành xu hướng nghiên cứu phổ biến.
- Việc ứng dụng các thuật toán tối ưu đa mục tiêu không chỉ giúp tìm ra các phương án bố trí hợp lý mà còn hỗ trợ việc đưa ra quyết định hiệu quả hơn. Những mô hình tối ưu này có khả năng xem xét đồng thời nhiều yếu tố như khoảng cách di chuyển giữa các khu vực, mức độ an toàn trong thi công. Đặc biệt, đối với các công trình có quy mô lớn và yêu cầu khắt khe về tiến độ, sự phối hợp chặt chẽ giữa các hạng mục trên công trường có thể ảnh hưởng trực tiếp đến tổng thể dự án.

- Bên cạnh đó, sự phát triển mạnh mẽ của công nghệ tính toán và mô phỏng đã tạo ra nhiều cơ hội để nâng cao hiệu quả tối ưu hóa bố trí mặt bằng thi công. Các công cụ phần mềm như MATLAB và nhiều phần mềm mô phỏng chuyên dụng khác đã giúp việc thử nghiệm, đánh giá và lựa chọn phương án bố trí tối ưu trở nên dễ dàng hơn, tiết kiệm thời gian và chi phí hơn so với các phương pháp truyền thống.

1.2. Tổng quan các nghiên cứu trước đây

- Việc tối ưu hóa tổng mặt bằng thi công đã được nghiên cứu trước đây. Các nghiên cứu này tập trung vào việc tìm ra các phương pháp bố trí tối ưu nhằm giảm chi phí, tăng hiệu suất làm việc và đảm bảo an toàn lao động trên công trường.

- Tối ưu hóa bố trí tổng mặt bằng thi công đơn mục tiêu như nghiên cứu của Vacharapoom Benjaoran và Vachara Peansupap (2019) [3] đã áp dụng Particle Swarm Optimization (PSO) để tính toán khoảng cách di chuyển giữa các hạng mục trên công trường, so sánh với thuật toán Genetic Algorithm (GA) và cho thấy PSO đạt hiệu quả cao hơn. Tuy nhiên, nghiên cứu sử dụng Excel 2013 để mô phỏng, gây khó khăn trong việc mô phỏng các mặt bằng thi công, các nghiên cứu trong tương lai cần tập trung vào việc nâng cấp mô phỏng bằng ngôn ngữ lập trình khác như MATLAB để nâng cao tính linh hoạt và độ chính xác của mô hình. Nghiên cứu của Dung-Nhan Nguyen Bui và Nang Duc Bui (2023) [4] cũng đã so sánh các thuật toán PSO, GA, Ant colony optimization (ACO) và chỉ ra rằng PSO cho kết quả tốt hơn. Tuy nhiên, bài toán gặp vấn đề về tốc độ hội tụ của PSO do các tham số như độ lớn của trọng số quán tính w và các yếu tố học tập c_1, c_2 chưa được tối ưu hóa, cần thêm các nghiên cứu bổ sung để xác định các hệ số tối ưu, cải thiện khả năng hội tụ của thuật toán.

- Tối ưu hóa bố trí tổng mặt bằng thi công đa mục tiêu như nghiên cứu của Mustafa Oral và các cộng sự (2018) [5] giới thiệu một phương pháp sử dụng MOPSO đa mục tiêu để tối ưu hóa vị trí các hạng mục tạm thời trên công trường, dựa trên việc giảm thiểu rủi ro an toàn và khoảng cách di chuyển giữa các nguồn lực. Tuy nhiên, nghiên cứu này chỉ giới hạn ở công trường đơn giản với một cầu tháp, chưa phù hợp với các công trường có hình dạng và quy mô khác. Do đó, hướng nghiên cứu tiếp

theo cần mở rộng mô hình này cho các công trường với nhiều cầu tháp, đồng thời tích hợp các yếu tố an ninh và môi trường. Tối ưu hóa cho vấn đề trao đổi giữa thời gian và chi phí xây dựng như nghiên cứu của Hong Zhang và Heng Li (2010) [6] phát triển một phương pháp đa mục tiêu MOPSO để cân bằng thời gian và chi phí thi công dựa trên tối ưu hóa Pareto. Tuy nhiên, nghiên cứu còn gặp khó khăn trong việc xác định tham số đầu vào cho thuật toán. Các nghiên cứu sẽ tập trung vào việc xây dựng nền tảng dữ liệu nhằm cải thiện khả năng xử lý dữ liệu thu thập và xem xét thêm các yếu tố không chắc chắn như thời gian và chi phí xây dựng cũng như chất lượng công trình. Sự đánh đổi giữa an toàn và chi phí trong bố trí công trường xây dựng cũng đã được trình bày bởi Khaled El-Rayes (2005) [7]. Mặc dù an toàn trong xây dựng là một trong những mục tiêu ưu tiên hàng đầu nhưng thực tế vẫn gặp phải các tai nạn nghiêm trọng. Nghiên cứu này đã phát triển một mô hình tối ưu hóa bố trí công trường không chỉ tối ưu hóa chi phí di chuyển của nguồn lực mà còn đảm bảo an toàn cho công nhân và thiết bị. Bài báo đã trình bày các khái niệm và tiêu chí hiệu suất để đo lường sự an toàn và chi phí di chuyển. Tuy nhiên bài báo chưa đưa ra được cơ sở lựa chọn các hệ số quan trọng của tiêu chí an toàn cần trực, tiêu chí kiểm soát nguy hiểm, tiêu chí giao điểm.

- Trong bố trí mặt bằng thi công, việc lựa chọn giải pháp tối ưu đòi hỏi đồng thời xem xét hai yếu tố khoảng cách di chuyển và an toàn lao động. Quá trình này là tìm ra cách sắp xếp các hạng mục tạm thời theo nhiều mục tiêu có thể mâu thuẫn với nhau, đồng thời phải tuân thủ các ràng buộc về tính hợp lý, không chồng chéo lên nhau hoặc vượt ra khỏi khu vực thi công. Việc lập kế hoạch bố trí mặt bằng công trường dưới dạng tối ưu hóa dẫn đến một bài toán phi tuyến với nhiều mục tiêu xung đột cần đạt được. Các nghiên cứu chỉ ra rằng các thuật toán tối ưu tỏ ra rất hiệu quả trong việc giải quyết các bài toán tối ưu hóa phức tạp dạng này như nghiên cứu của Vacharapoom Benjaoran và Vachara Peansupap (2019) [3], Hong Zhang và Heng Li (2010) [6], Dung-Nhan Nguyen Bui và Nang Duc Bui (2023) [4].

- Thuật toán NSGA-II và MOPSO là thuật toán tối ưu đa mục tiêu được sử dụng phổ biến trong việc giải quyết bài toán phân bổ nghiệm trên biên Pareto. Mỗi thuật

toán đều có mặt ưu điểm riêng, đã có nhiều nghiên cứu so sánh về 2 thuật toán này. Nghiên cứu của Rachmat Parayoga và Anna Maria Sri Asih (2021) [8] đã chỉ ra rằng NSGA-II có hiệu suất tốt hơn MOPSO ở số lượng giải pháp Pareto, thời gian tính toán. Tuy nhiên, đối với chỉ số đa dạng, MOPSO vẫn tốt hơn. Nghiên cứu của Mohamad Izdin Hlal và các cộng sự (2019) [9], phương pháp NSGA-II có hiệu suất mạnh mẽ hơn, trong khi phương pháp MOPSO nhanh hơn trong việc thực hiện tối ưu hóa.

1.3. Cơ sở lý thuyết các thuật toán tối ưu hóa

1.3.1. Thuật toán Genetic Algorithms (GA)

- Thuật toán di truyền GA là một thuật toán tối ưu dựa trên cơ chế tiến hóa tự nhiên, sử dụng các nguyên lý chọn lọc tự nhiên, lai ghép và đột biến để tìm kiếm các giải pháp tối ưu cho các bài toán tối ưu hóa.

- Thuật toán GA thường được sử dụng cho các bài toán tối ưu không có cấu trúc rõ ràng, các bài toán có không gian tìm kiếm lớn hoặc không thể giải quyết bằng các phương pháp toán học cổ điển. Sau đây là một thuyết minh chi tiết về cách thức hoạt động của thuật toán GA.

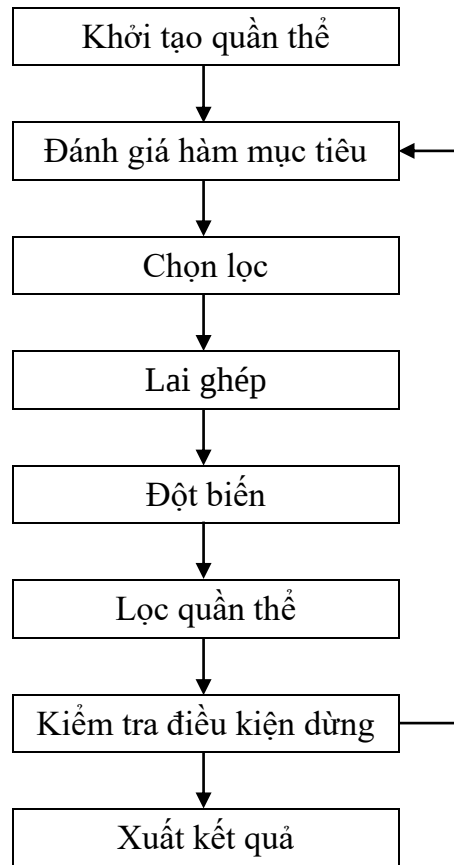
- Các thành phần cơ bản của thuật toán GA:

+ Quần thể: là tập hợp các cá thể, mỗi cá thể đại diện cho một giải pháp tiềm năng của bài toán tối ưu. Quần thể này được khởi tạo ngẫu nhiên và mỗi cá thể trong quần thể có một chuỗi mã hóa (dưới dạng nhị phân hoặc mã hóa số khác) để biểu diễn các tham số của bài toán.

+ Nhiễm sắc thể: Một cá thể trong GA được biểu diễn dưới dạng một chuỗi nhiễm sắc thể. Mỗi nhiễm sắc thể có thể được mã hóa dưới dạng chuỗi nhị phân, số nguyên, hoặc các cấu trúc khác, phụ thuộc vào loại bài toán cần giải quyết.

+ Hàm mục tiêu: là hàm đánh giá mức độ tốt của một cá thể trong quần thể, tức là nó đánh giá khả năng của cá thể đó trong việc giải quyết bài toán. Mục tiêu của thuật toán GA là tối đa hóa (hoặc tối thiểu hóa, tùy vào bài toán) giá trị của hàm thích nghi cho các cá thể.

- Thuật toán GA bao gồm một chuỗi các bước lặp đi lặp lại, qua đó tìm ra các giải pháp tối ưu cho bài toán. Quy trình thuật toán GA được mô tả trong hình 1.1.



Hình 1.1. Sơ đồ thuật toán GA

+ Khởi tạo quần thể: Một quần thể ban đầu được tạo ra ngẫu nhiên với một số lượng cá thể nhất định. Các cá thể trong quần thể này sẽ được mã hóa dưới dạng các chuỗi nhiễm sắc thể.

+ Đánh giá hàm mục tiêu: Mỗi cá thể trong quần thể được đánh giá bằng hàm thích nghi, để xác định mức độ tốt của nó đối với bài toán cần tối ưu. Hàm thích nghi có thể đo lường sai số hoặc hiệu suất của cá thể dựa trên các tham số đã chọn.

+ Chọn lọc: là quá trình lựa chọn các cá thể tốt để nhân giống và tạo ra thế hệ tiếp theo. Phương pháp chọn lọc phổ biến là "chọn lọc theo tỷ lệ thích nghi". Các cá thể có giá trị hàm thích nghi cao có xác suất được chọn lọc cao hơn, nhưng không phải là chắc chắn. Điều này giúp giữ lại sự đa dạng trong quần thể.

+ Lai ghép: Sau khi chọn lọc, các cặp cá thể sẽ được kết hợp với nhau qua một quá trình gọi là lai ghép. Lai ghép tạo ra các cá thể con bằng cách trao đổi một phần thông tin giữa các cá thể cha mẹ. Thông thường, lai ghép sẽ xảy ra ở các điểm ngẫu nhiên trong chuỗi nhiễm sắc thể của các cá thể cha mẹ. Mục tiêu của lai ghép là tạo ra những cá thể con có khả năng tốt hơn hoặc tương đương các cá thể cha mẹ trong quá trình tối ưu.

+ Đột biến: Đột biến là quá trình thay đổi ngẫu nhiên một phần của nhiễm sắc thể của cá thể con sau khi lai ghép. Điều này giúp duy trì sự đa dạng di truyền trong quần thể, tránh việc các cá thể trở nên đồng nhất và rơi vào cực trị địa phương.

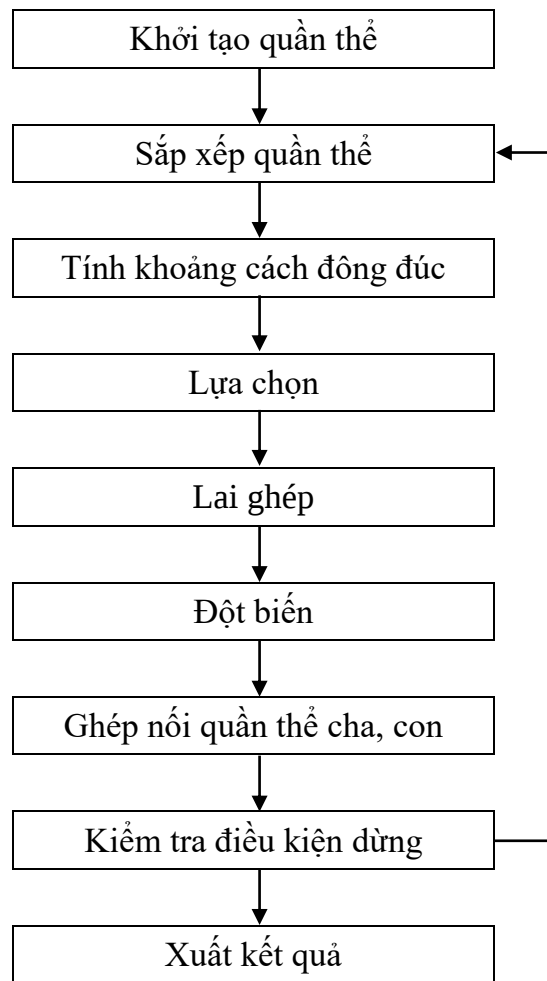
+ Lọc quần thể: Sau khi tạo ra các cá thể con qua lai ghép và đột biến, quần thể sẽ được cập nhật. Có thể giữ lại toàn bộ hoặc một phần các cá thể từ quần thể hiện tại và kết hợp với các cá thể con để tạo ra quần thể tiếp theo. Phương pháp giữ lại cá thể tốt nhất giúp đảm bảo rằng chất lượng quần thể không bị suy giảm trong quá trình tiến hóa.

+ Dừng thuật toán: Thuật toán sẽ dừng lại khi một trong các điều kiện dừng sau được thỏa mãn: Số vòng lặp đạt đến giới hạn, hàm thích nghi của cá thể tốt nhất đạt một giá trị thỏa mãn, các cá thể trong quần thể không thay đổi nhiều qua các thế hệ.

1.3.2. Thuật toán Non-Dominated Sorting Genetic Algorithm II (NSGA-II)

- NSGA-II dựa trên cơ sở thuật toán GA được phát triển bởi Kalyanmoy Deb và các cộng sự (2002) [1], sử dụng để giải quyết các bài toán tối ưu đa mục tiêu.

- NSGA-II được thiết kế để tìm ra tập các giải pháp Pareto tối ưu trong không gian nhiều mục tiêu. Các giải pháp Pareto tối ưu là những giải pháp mà không có giải pháp nào trong tập tìm được cải thiện một mục tiêu mà không làm giảm một mục tiêu khác. Sơ đồ của NSGA-II được trình bày ở hình 1.2.



Hình 1.2. Sơ đồ thuật toán NSGA-II

- Các bước của NSGA-II:

- + Khởi tạo quần thể: Tạo một quần thể ban đầu ngẫu nhiên.
- + Sắp xếp không trội: Chia quần thể thành các bậc dựa trên quan hệ không trội. Nghiệm nào không bị trội bởi nghiệm khác sẽ thuộc bậc 1. Các nghiệm tiếp theo được xếp vào các bậc 2, 3,...
- + Tính khoảng cách đồng đúc: Tính khoảng cách giữa các nghiệm trong cùng một bậc để đảm bảo đa dạng nghiệm trên biên Pareto.
- + Lựa chọn: Chọn các cá thể từ quần thể hiện tại dựa trên bậc không trội và khoảng cách đồng đúc.
- + Lai ghép và đột biến: Tạo ra thế hệ con bằng phép lai và đột biến.
- + Kết hợp quần thể cha-con: Kết hợp sau đó chọn ra quần thể tốt nhất dựa trên sắp xếp không trội và khoảng cách đồng đúc.

- NSGA-II có những đặc điểm sau:

- + Đảm bảo không có sự ưu tiên về mục tiêu nào (tức là không có trọng số giữa các mục tiêu).
- + Duy trì sự đa dạng trong quần thể, giúp tránh việc hội tụ quá sớm vào một vùng cụ thể của không gian giải pháp.

- Sự khác nhau giữa GA và NSGA-II được trình bày ở bảng 1.1.

Bảng 1.1. So sánh thuật toán GA và NSGA-II

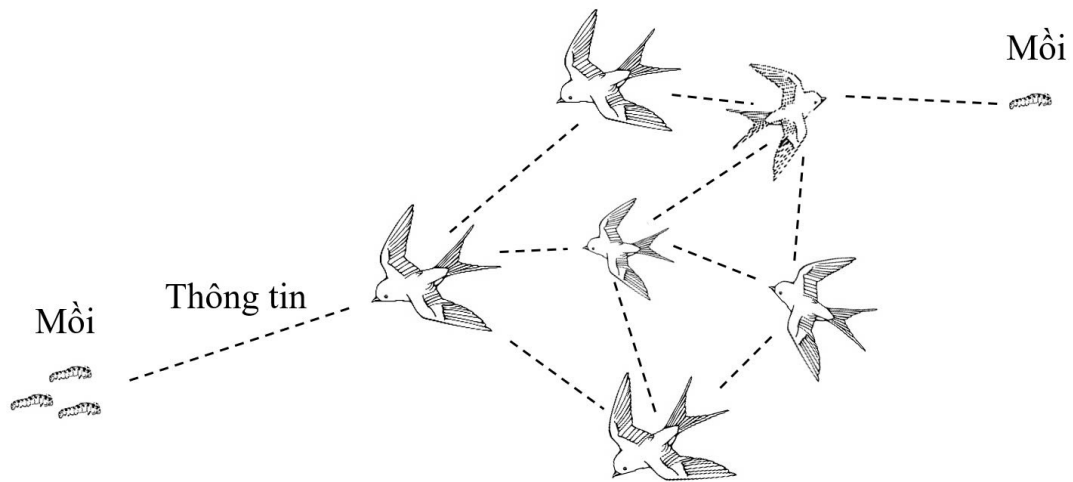
	GA	NSGA-II
Mục tiêu	Tối ưu đơn mục tiêu	Tối ưu đa mục tiêu
Phân loại nghiệm	Không phân loại bậc nghiệm	Sắp xếp không trội
Đa dạng nghiệm	Không có cơ chế duy trì đa dạng	Sử dụng khoảng cách đồng đúc
Lựa chọn cá thể	Dựa trên hàm 1 mục tiêu	Dựa trên bậc sắp xếp không trội và khoảng cách đồng đúc
Phân bố nghiệm	Hội tụ vào 1 nghiệm tốt nhất	Tạo ra biên Pareto đồng đều
Độ phức tạp	Thấp	Cao

1.3.3. Thuật toán Particle Swarm Optimization (PSO)

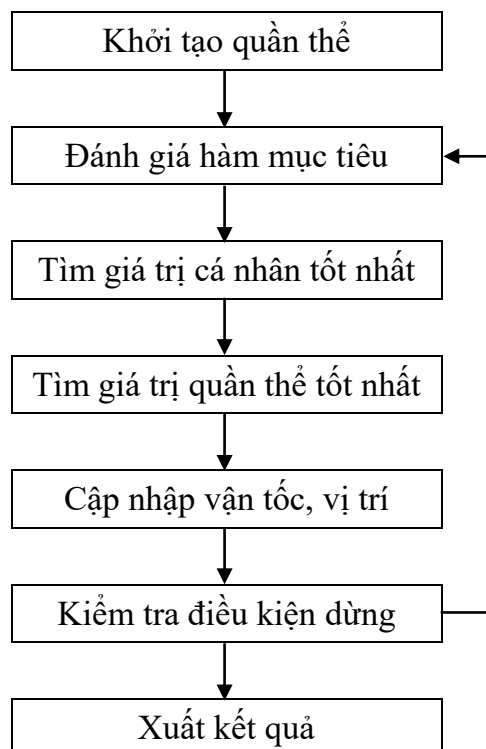
- PSO là thuật toán ưu hoá ngẫu nhiên dựa trên một quần thể gồm nhiều cá thể để tìm ra nghiệm tối ưu bằng cách cập nhật các vị trí tốt nhất. Mô phỏng hành vi tìm kiếm thức ăn của đàn cá hoặc bầy chim. Mỗi cá thể của bầy đàn được đặc trưng bởi hai tham số là vị trí và vận tốc. Một cá thể luôn tìm kiếm trong không gian tìm kiếm của chính nó để thay thế vị trí cũ bằng vị trí mới tốt nhất.

- Mô phỏng quá trình tìm kiếm thức ăn của bầy đàn được diễn tả ở hình 1.3, trong không gian tìm kiếm thức ăn lúc này là toàn bộ không gian ba chiều. Tại thời điểm bắt đầu tìm kiếm cả đàn bay theo một hướng nào đó, có thể là ngẫu nhiên. Tuy nhiên sau một thời gian tìm kiếm một số cá thể trong đàn bắt đầu tìm ra được nơi có chứa thức ăn. Tùy theo số lượng thức ăn vừa tìm kiếm, mà cá thể gửi tín hiệu đến các cá thể khác đang tìm kiếm ở vùng lân cận. Tín hiệu này lan truyền trên toàn

quần thể. Dựa vào thông tin nhận được mỗi cá thể sẽ điều chỉnh hướng bay và vận tốc theo hướng về nơi có nhiều thức ăn nhất. Cơ chế truyền tin như vậy thường được xem như là một kiểu hình của trí tuệ bầy đàn. Cơ chế này giúp cả đàn chim tìm ra nơi có nhiều thức ăn nhất trên không gian tìm kiếm vô cùng rộng lớn. Sơ đồ thuật toán được thể hiện ở hình 1.4.



Hình 1.3. Cảnh đàn chim tìm kiếm mồi



Hình 1.4. Sơ đồ thuật toán PSO

- Các bước chính của PSO:

+ Khởi tạo số lượng các cá thể ngẫu nhiên trong không gian tìm kiếm, số lượng cá thể phụ thuộc vào quy mô không gian tìm kiếm. Gán vị trí x và vận tốc v cho từng cá thể.

+ Mô tả quá trình di chuyển của cá thể được thể hiện ở hình 1.5. Mỗi cá thể thứ i sẽ di chuyển theo 2 yếu tố:

- Vị trí tốt nhất của cá thể i từng đi qua P_i :

$$P_i = (P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{in})$$

- Vị trí tốt nhất của đàn G :

$$f(G) = \min[f(P_i)] \text{ hoặc } f(G) = \max[f(P_i)]$$

+ Vị trí và tốc độ của mỗi cá thể được cập nhật như sau:

- Vận tốc mới của cá thể:

$$v_i(t+1) = w \times v_i(t) + c_1 \times r_1 \times (P_i(t) - x_i(t)) + c_2 \times r_2 \times (G(t) - x_i(t))$$

- Vị trí mới của cá thể:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Với: w : Hệ số điều chỉnh mức độ giữ lại vận tốc ban đầu

c_1 : Hệ số gia tốc cá thể

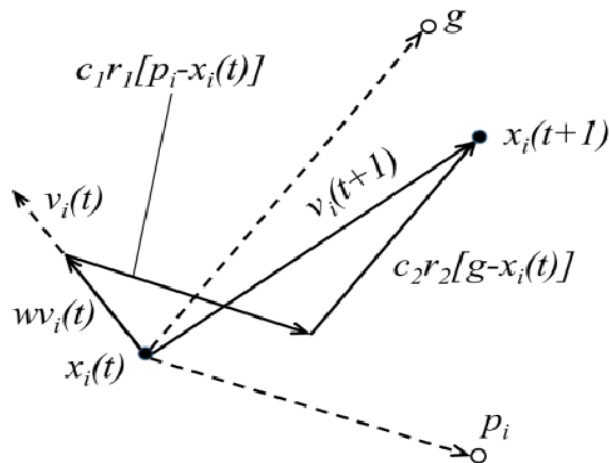
c_2 : Hệ số gia tốc quần thể

r_1, r_2 : Biến số ngẫu nhiên từ 0 đến 1

$w \times v_i(t)$: Thể hiện quán tính, tỷ lệ với vận tốc

$c_1 \times r_1 \times (P_i(t) - x_i(t))$: Nhận thức, tỷ lệ với sai khác giữa vị trí tốt nhất của mỗi P_i và vị trí tức thời x_i của nó.

$c_2 \times r_2 \times (G(t) - x_i(t))$: Xã hội, tỷ lệ với sai khác giữa vị trí tốt nhất của đàn G và vị trí tức thời x_i của con thứ i .



Hình 1.5. Mô tả quá trình cập nhật vị trí và vận tốc [10]

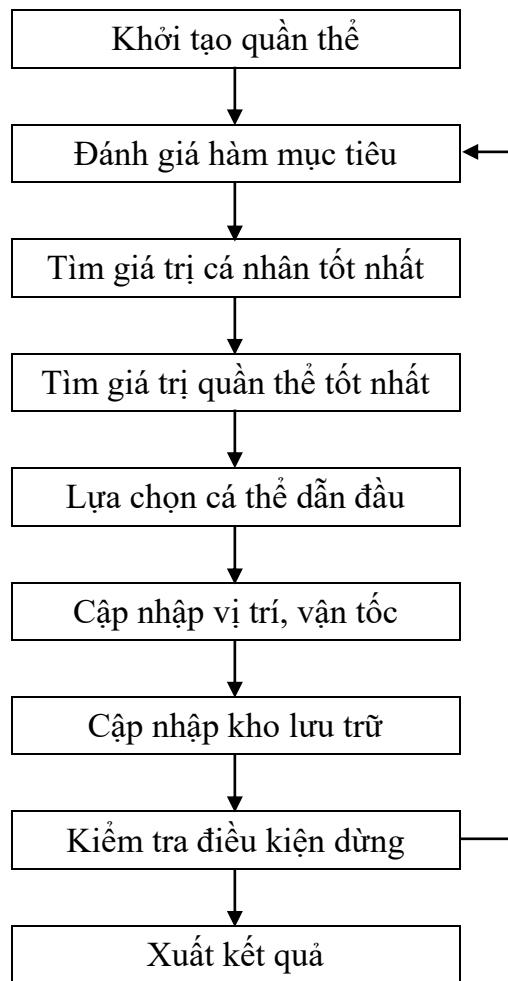
- Trong quá trình tìm kiếm, dựa vào thông tin nhận được về vị trí của cả đàn, các cá thể luôn cập nhật vận tốc di chuyển và vị trí của mình. Quá trình tính toán sẽ được lặp lại cho đến khi đạt được điều kiện dừng nhất định như số lần lặp tối đa, giá trị hàm mục tiêu đủ nhỏ hoặc khi các cá thể không thay đổi nữa.

1.3.4. Thuật toán Multi-Objective Particle Swarm Optimization (MOPSO)

- MOPSO là một biến thể của PSO như hình 1.6, được thiết kế để giải quyết các bài toán tối ưu nhiều mục tiêu. Khi có nhiều hơn một mục tiêu, thay vì tìm một nghiệm tối ưu duy nhất, thuật toán tìm tập hợp các nghiệm Pareto tối ưu.

- Các bước khởi tạo quần thể, đánh giá hàm mục tiêu, tìm giá trị cá nhân tốt nhất, tìm giá trị quần thể tốt nhất tương tự thuật toán PSO.

- Việc chọn cá thể dẫn đầu rất quan trọng, vì nó quyết định hướng di chuyển của các hạt và ảnh hưởng trực tiếp đến sự hội tụ cũng như tính đa dạng của tập nghiệm Pareto tối ưu. Hạt dẫn đầu giúp hướng dẫn từng hạt di chuyển về phía các vùng tiềm năng trên biên Pareto. Có thể lựa chọn hạt dẫn đầu dựa trên cơ chế ô lưới hoặc cơ chế khoảng cách đồng đúc. Các cá thể mới được thêm vào kho lưu trữ nếu chúng không bị chiếm ưu thế.

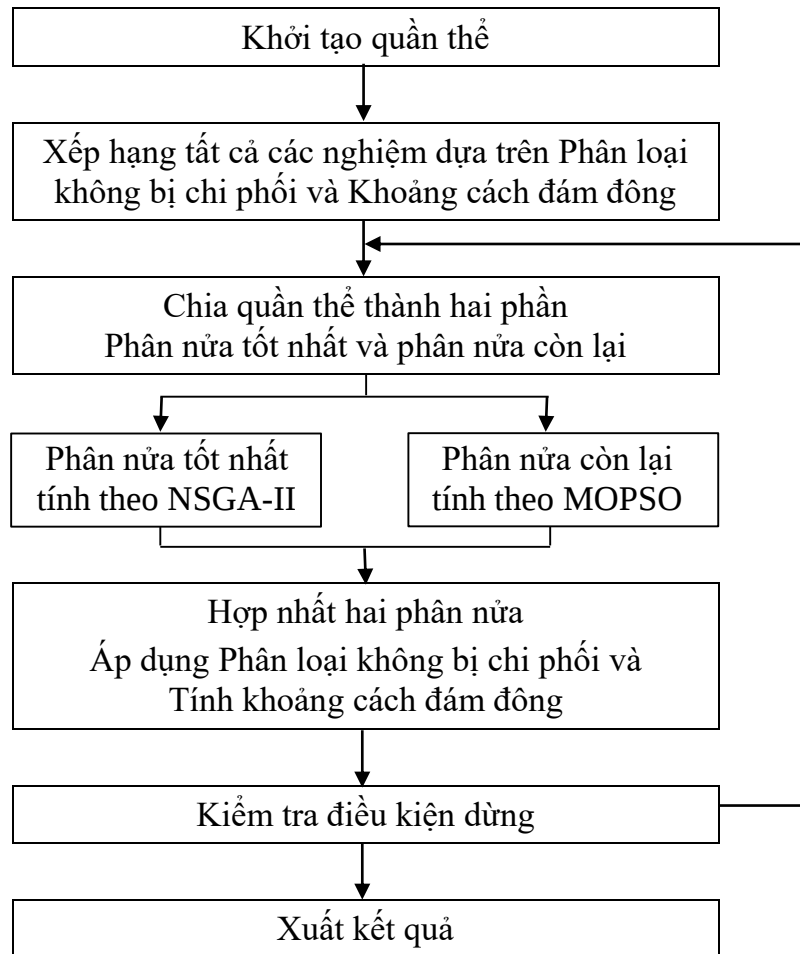


Hình 1.6. Sơ đồ thuật toán MOPSO

1.3.5. Thuật toán NSGA-II lai MOPSO

- Để giải quyết hiệu quả bài toán tối ưu hóa đa mục tiêu, tác giả đề xuất thuật toán lai kết hợp giữa NSGA-II và MOPSO. Mục tiêu của phương pháp đề xuất là tìm được tập nghiệm tiệm cận biên Pareto với độ hội tụ cao và đồng thời duy trì được sự đa dạng trong không gian nghiệm. Cụ thể, NSGA-II đảm nhiệm vai trò duy trì tính đa dạng thông qua cơ chế phân loại không bị chi phối và tính toán khoảng cách đám đông, trong khi đó MOPSO đảm nhận chức năng khai thác cục bộ để tăng cường khả năng hội tụ tại các vùng lời giải tiềm năng. Việc kết hợp hai thuật toán này giúp tạo ra một quy trình tìm kiếm cân bằng giữa thăm dò và khai thác, từ đó nâng cao hiệu quả của thuật toán tối ưu.

- Hình 1.7 minh họa quy trình thực hiện thuật toán NSGA-II lai MOPSO. Một quần thể các nghiệm được khởi tạo ngẫu nhiên và đánh giá thông qua các hàm mục tiêu. Các cá thể trong quần thể sau đó được xếp hạng dựa trên cơ chế phân loại không bị chi phối kết hợp với tính toán khoảng cách đám đông nhằm đảm bảo sự đa dạng của quần thể.



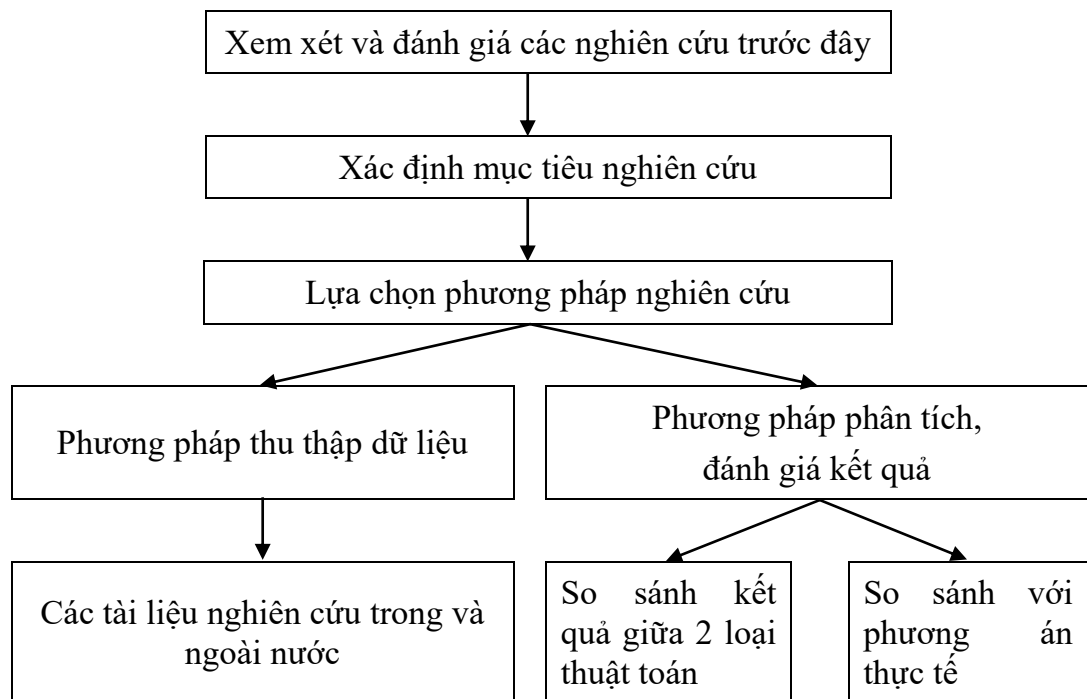
Hình 1.7. Sơ đồ thuật toán NSGA-II lai MOPSO

- Sau bước đánh giá, quần thể được chia thành hai nhóm con bằng nhau: một nửa tốt nhất bao gồm các nghiệm có chất lượng cao hơn, và một nửa còn lại gồm các cá thể ít ưu việt hơn. Mỗi nhóm được tiến hóa theo một chiến lược riêng biệt. Nhóm tốt nhất được tính toán theo NSGA-II, bao gồm lai ghép nhị phân mô phỏng và đột biến đa thức, nhằm khám phá thêm không gian nghiệm. Trong khi đó, nhóm ít ưu việt hơn được tính toán theo MOPSO, các cá thể được xem như các hạt di chuyển

theo công thức cập nhật vận tốc và vị trí dựa trên nghiệm tốt nhất cá nhân và nghiệm tốt nhất toàn cục.

- Sau khi hai nhóm được tiến hóa độc lập, chúng được hợp nhất để tái tạo lại quần thể mới. Toàn bộ quần thể sau khi hợp nhất tiếp tục được đánh giá lại theo cơ chế phân loại không bị chi phối và tính khoảng cách đám đông. Quá trình chia nhóm, tiến hóa và hợp nhất được lặp lại liên tục qua các thế hệ cho đến khi đạt đến tiêu chí dừng là số vòng lặp tối đa được xác định trước. Kết thúc quá trình, tập nghiệm không bị chi phối trong quần thể cuối cùng được trích xuất và xem như là biên Pareto tiệm cận của bài toán tối ưu.

1.4. Phương pháp nghiên cứu



Hình 1.8. Sơ đồ phương pháp nghiên cứu

- Sơ đồ phương pháp nghiên cứu được thể hiện ở hình 1.8. Quá trình nghiên cứu bắt đầu bằng việc tổng hợp và phân tích các tài liệu khoa học liên quan đến tối ưu hóa bố trí tổng mặt bằng thi công và các thuật toán tối ưu hóa đa mục tiêu. Các nghiên cứu trước đây được xem xét theo phương pháp truyền thống dựa trên kinh nghiệm, mô hình thuật toán tối ưu như GA, PSO, ACO và các phương pháp tối ưu hóa hiện đại khác trong bố trí mặt bằng. Những hạn chế của các nghiên cứu trước đây như

phần lớn tập trung vào bài toán đơn mục tiêu thay vì đa mục tiêu, chưa tích hợp đầy đủ các yếu tố thực tế như an toàn lao động, chi phí vận hành, môi trường..

- Dựa vào phân tích các nghiên cứu trước đây, nghiên cứu này đặt ra các mục tiêu:
 - + Phát triển mô hình tối ưu hóa bố trí tổng mặt bằng thi công đa mục tiêu giữa khoảng cách di chuyển và an toàn lao động.
 - + Đề xuất và cải tiến thuật toán tối ưu hóa, cụ thể là so sánh NSGA-II và NSGA-II lai MOPSO, nhằm nâng cao khả năng tìm kiếm nghiệm Pareto tối ưu hơn.
 - + Áp dụng mô hình vào bài toán thực tế, sử dụng dữ liệu từ các dự án xây dựng để kiểm chứng tính khả thi của phương pháp đề xuất.
- Dữ liệu phục vụ nghiên cứu được thu thập từ các dự án xây dựng như:
 - + Thông tin về mặt bằng xây dựng, điều kiện địa hình.
 - + Danh sách các hạng mục tạm thời: văn phòng điều hành, kho chứa vật liệu, trạm bê tông, cần trục tháp, bãi tập kết vật liệu...
 - + Quãng đường di chuyển giữa các hạng mục.
- Sau khi xây dựng mô hình và chạy thuật toán tối ưu, kết quả sẽ được phân tích, đánh giá theo hai hướng:
 - + Đánh giá các giải pháp tối ưu tìm được giữa hai thuật toán NSGA-II và NSGA-II lai MOPSO.
 - + So sánh, phân tích với dữ liệu thực tế để phản ánh mức độ hiệu quả của bài toán tối ưu hóa bố trí mặt bằng thi công.

1.5. Tổng kết chương 1

- Chương 1 đã trình bày tổng quan về đề tài nghiên cứu, nhấn mạnh vai trò quan trọng của việc tối ưu hóa bố trí tổng mặt bằng thi công trong xây dựng. Việc sắp xếp hợp lý các hạng mục tạm thời không chỉ ảnh hưởng đến chi phí và tiến độ mà còn tác động đến hiệu quả vận hành và an toàn lao động.
- Các thuật toán tối ưu đơn mục tiêu vẫn còn nhiều hạn chế, chưa phản ánh đầy đủ các yếu tố thực tiễn. Do đó, hướng tiếp cận đa mục tiêu, đặc biệt với các tiêu chí như khoảng cách di chuyển và an toàn lao động, ngày càng được quan tâm. Chương

đã giới thiệu các thuật toán tối ưu như GA, PSO, NSGA-II, MOPSO và NSGA-II lai MOPSO. NSGA-II có ưu điểm trong việc duy trì tính đa dạng nghiệm, còn MOPSO có khả năng hội tụ nhanh và khai thác tốt vùng lân cận nghiệm tốt. Việc kết hợp hai thuật toán này có thể mang lại hiệu quả tối ưu hơn.

Chương 2 - XÂY DỰNG MÔ HÌNH TỐI ƯU HÓA MẶT BẰNG THI CÔNG

2.1. Khái quát các hạng mục phụ trợ

- Cần trục tháp là thiết bị chính để nâng hạ vật liệu xây dựng như bê tông, thép, ván khuôn và các cấu kiện nặng. Vị trí lắp đặt cần đảm bảo tầm với bao quát các khu vực quan trọng như bãi tập kết cấu kiện đúc sẵn, bãi trộn bê tông, bãi tập kết thép và các hạng mục theo thiết kế. Phải được lắp đặt ở vị trí an toàn, có nền móng chắc chắn và không bị cản trở bởi các công trình lân cận. Cần trục tháp cũng phải đảm bảo khả năng xoay chuyển linh hoạt và tầm với tối ưu để giảm thiểu thời gian di chuyển vật liệu.
- Bãi tập kết cấu kiện đúc sẵn là nơi lưu trữ các cấu kiện bê tông đúc sẵn như dầm, cột, tấm sàn. Cần được bố trí gần khu vực thi công và trong tầm hoạt động của cần trục tháp để thuận tiện cho việc nâng hạ. Bãi tập kết phải đảm bảo mặt bằng rộng rãi, thông thoáng. Phải có biện pháp bảo vệ các cấu kiện khỏi tác động của thời tiết và hư hỏng cơ học.
- Bãi tập kết cát, đá, xi măng là nơi lưu trữ các vật liệu xây dựng cơ bản phục vụ cho trộn bê tông và vữa. Cần gần bãi trộn bê tông để giảm thiểu chi phí và thời gian vận chuyển. Cát, đá phải được phân loại rõ ràng và bảo quản đúng quy cách để tránh tình trạng nhiễm bẩn. Xi măng phải được bảo quản trong kho kín để tránh ẩm ướt.
- Bãi trộn bê tông, vữa là nơi thực hiện trộn bê tông và vữa phục vụ công trình. Cần gần bãi tập kết cát, đá, xi măng và nằm trong tầm hoạt động của cần trục tháp. Bãi trộn cần có hệ thống cấp nước ổn định và hệ thống thoát nước đảm bảo.
- Bãi tập kết thép là nơi lưu trữ thép nguyên liệu phục vụ gia công. Cần gần bãi gia công thép và thuận tiện cho việc di chuyển bằng xe nâng hoặc cần trục. Thép cần được phân loại và sắp xếp gọn gàng để dễ dàng kiểm soát số lượng và chất lượng.
- Bãi gia công thép là nơi cắt, uốn, gia công thép trước khi lắp đặt. Nên bố trí gần bãi tập kết thép để đảm bảo tiến độ gia công.
- Bãi tập kết giàn giáo ván khuôn là nơi lưu trữ giàn giáo và ván khuôn phục vụ công tác cốp pha. Nằm gần khu vực thi công để dễ dàng vận chuyển và lắp đặt.

- Kho thiết bị là nơi bảo quản các thiết bị xây dựng, cần bảo đảm an toàn và gần khu vực thi công để kịp thời cung ứng thiết bị khi cần.
- Kho chứa nhiên liệu là nơi lưu trữ xăng, dầu phục vụ máy móc thiết bị. Phải đặt xa các khu vực dễ cháy và có biện pháp an toàn phòng cháy chữa cháy.
- Máy phát điện dự phòng để đảm bảo nguồn điện liên tục cho công trường khi xảy ra sự cố mất điện. Đặt gần khu vực cần cung cấp điện ổn định như bãi trộn bê tông, cần trục tháp. Phải thường xuyên bảo trì để đảm bảo hoạt động ổn định.
- Nhà bảo vệ là nơi kiểm soát an ninh cho công trường. Đặt gần cổng ra vào công trình để kiểm soát người và phương tiện.
- Nhà làm việc cán bộ kỹ thuật là nơi làm việc, quản lý của đội ngũ kỹ thuật. Nên đặt ở vị trí trung tâm để dễ dàng giám sát công trình.
- Nhà xe công nhân là nơi để xe cho công nhân và cán bộ kỹ thuật. Cần gần cổng ra vào và các hạng mục thi công để thuận tiện cho việc di chuyển.
- Lán trại nghỉ ngơi là nơi nghỉ ngơi của công nhân. Nên đặt xa khu vực thi công để giảm tiếng ồn.
- Nhà vệ sinh phục vụ nhu cầu sinh hoạt cơ bản của công nhân, cần bố trí ở khu vực phù hợp để đảm bảo vệ sinh và tiện lợi.
- Các hạng mục phụ trợ trên công trường có mối quan hệ mật thiết với nhau, đòi hỏi sự sắp xếp khoa học và hợp lý để đảm bảo tiến độ, an toàn và hiệu quả trong quá trình thi công. Việc quản lý, giám sát và bảo trì các hạng mục này là yếu tố then chốt góp phần vào thành công của dự án.

2.2. Các giả định của mô hình

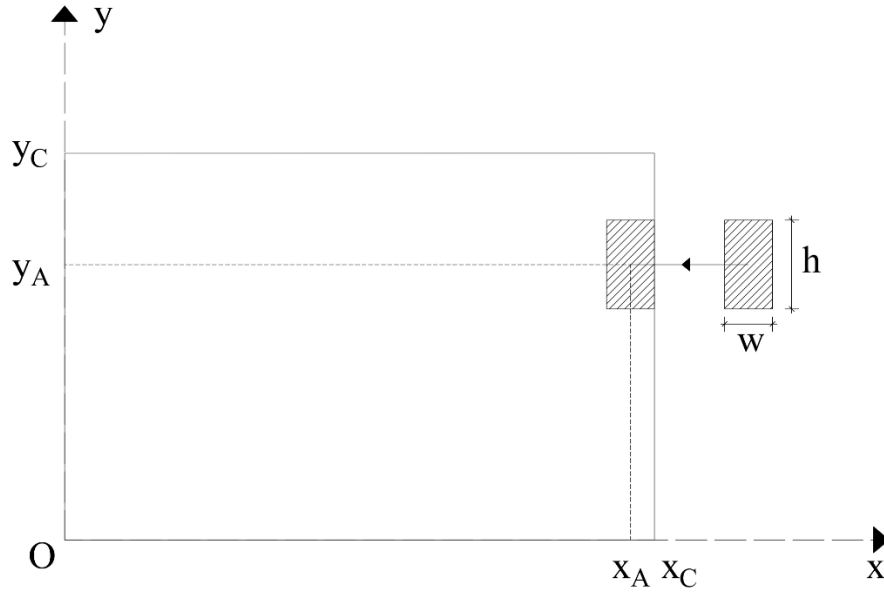
- Kích thước mặt bằng thi công hình chữ nhật được tích hợp với tọa độ Oxy, điểm O nằm ở góc dưới cùng bên trái. Mỗi hạng mục phụ trợ mang 3 ẩn số cần tối ưu gồm tọa độ theo trục x , tọa độ theo trục y và hệ số góc xoay r , không xoay với $r = 0$ và xoay 90° với $r = 1$.
- Để hạn chế mức độ phức tạp trong quá trình tính toán nhưng vẫn đảm bảo tạo ra các giải pháp tối ưu, quãng đường di chuyển giữa 2 hạng mục d_{ij} là đường thẳng

ngắn nhất từ tâm hạng mục thứ i đến tâm hạng mục thứ j , được tính như công thức (2.1).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2.1)$$

2.3. Các ràng buộc của mô hình

2.3.1. Ràng buộc về phạm vi khu đất



Hình 2.1. Ràng buộc khu đất

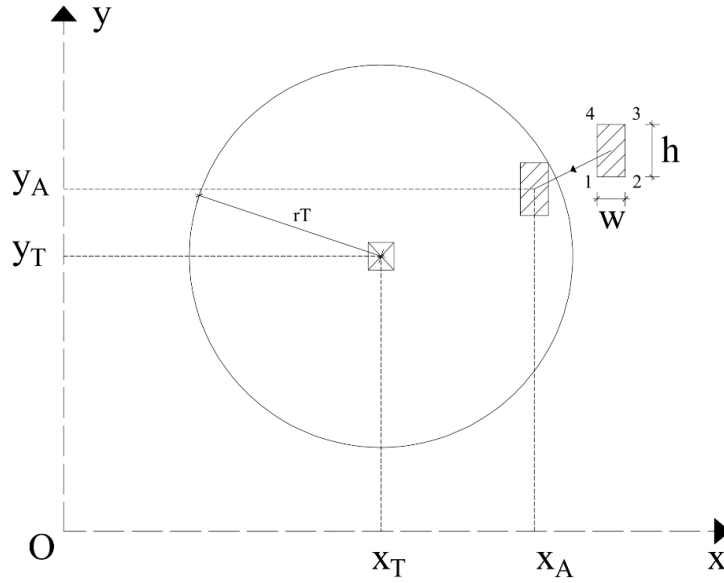
- Trong quá trình tối ưu, các hạng mục phụ trợ có thể di chuyển ra ngoài phạm vi khu đất do việc khởi tạo ngẫu nhiên vị trí ban đầu hoặc trong quá trình tìm kiếm nghiệm như hình 2.1. Để đảm bảo các hạng mục phụ trợ không vượt ra ngoài ranh giới khu đất, ràng buộc về tọa độ được sử dụng để đảm bảo rằng hạng mục thứ i khi di chuyển sẽ luôn nằm trong phạm vi khu đất thi công. Ràng buộc này được thể hiện ở công thức (2.2).

$$\begin{aligned} x_A(i) &\in \left[\min(x_C) + \frac{w(i)}{2}; \max(x_C) - \frac{w(i)}{2} \right] \\ y_A(i) &\in \left[\min(y_C) + \frac{h(i)}{2}; \max(y_C) - \frac{h(i)}{2} \right] \end{aligned} \quad (2.2)$$

Trong đó $x_A(i)$, $y_A(i)$ là tọa độ trọng tâm theo phương x và y của hạng mục

thứ i . x_C, y_C là tọa độ biên của khu đất thi công. $w(i), h(i)$ là chiều rộng và chiều dài của hạng mục thứ i theo phương x và phương y .

2.3.2. Ràng buộc về phạm vi hoạt động cần trục tháp



Hình 2.2. Ràng buộc phạm vi cần trục tháp

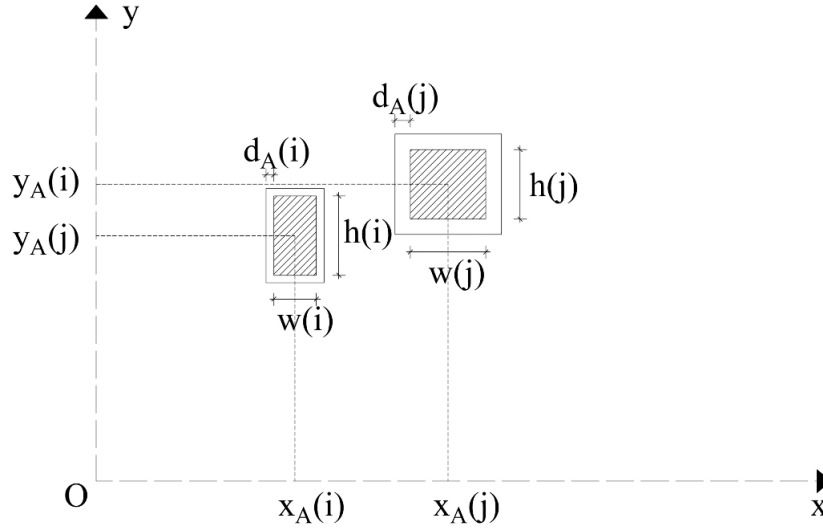
- Cần trục tháp là một trong những thiết bị quan trọng và phổ biến nhất tại các công trình xây dựng cao tầng và quy mô lớn. Khả năng vận chuyển, nâng hạ vật liệu và thiết bị của cần trục tháp phụ thuộc vào bán kính hoạt động của nó. Vì vậy, các hạng mục phụ trợ yêu cầu nâng hạ bằng cần trục tháp cần được bố trí sao cho đảm bảo nằm trong phạm vi bán kính này như hình 2.2. Ràng buộc này được thể hiện ở công thức (2.3).

$$\text{Max}(dd1, dd2, dd3, dd4) \leq r_T \quad (2.3)$$

Trong đó $dd1, dd2, dd3, dd4$ là khoảng cách từ cần trục tháp đến 4 đỉnh của hình chữ nhật. $dd(i)$ được tính bằng công thức (2.4); r_T là bán kính cần trục.

$$dd(i) = \sqrt{((x_A(i) \pm w(i)) - x_T)^2 + ((y_A(i) \pm h(i)) - y_T)^2} \quad (2.4)$$

2.3.3. Ràng buộc trùng lặp và khoảng cách tối thiểu giữa các hạng mục



Hình 2.3. Ràng buộc trùng lặp hạng mục và khoảng cách tối thiểu

- Trong quá trình di chuyển để tối ưu vị trí, các hạng mục phụ trợ phải được sắp xếp sao cho không được trùng lặp vị trí nhau và đảm bảo khoảng cách tối thiểu giữa các hạng mục như hình 2.3. Để đảm bảo ràng buộc này thì $\min(XV_{ij}, YV_{ij}) = 0$, XV_{ij} và YV_{ij} được trình bày theo công thức (2.5), (2.6).

$$XV_{ij} = \max\left(0, 1 - \frac{|x_A(i) - x_A(j)|}{\frac{w(i) + w(j)}{2} + \max(d_A(i), d_A(j))}\right) \quad (2.5)$$

$$YV_{ij} = \max\left(0, 1 - \frac{|y_A(i) - y_A(j)|}{\frac{h(i) + h(j)}{2} + \max(d_A(i), d_A(j))}\right) \quad (2.6)$$

Trong đó $d_A(i)$, $d_A(j)$ là khoảng cách tối thiểu của hạng mục thứ i , j để đảm bảo lối di chuyển tối thiểu.

2.4. Các hàm mục tiêu của mô hình

2.4.1. Hàm mục tiêu tối ưu hóa khoảng cách di chuyển

- Việc tối ưu khoảng cách di chuyển giữa các hạng mục giúp nâng cao hiệu quả hoạt động trên công trường xây dựng. Mục đích của hàm mục tiêu này là giá trị tổng quãng đường di chuyển của vật liệu, thiết bị, nhân lực phải nhỏ nhất, từ đó tiết kiệm

thời gian, chi phí và tăng năng suất lao động. Vacharapoom Benjaoran và Vachara Peansupap (2019) [3], Mustafa ORAL và các cộng sự (2018) [5], Chakrey Duong (2023) [11] đã nghiên cứu về cách tính hàm mục tiêu khoảng cách di chuyển như công thức (2.7).

$$C = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (d_{ij} \times w_{ij}) \quad (2.7)$$

Trong đó n là tổng số lượng các hạng mục trên công trình bao gồm cả hạng mục theo thiết kế và hạng mục phụ trợ. d_{ij} là khoảng cách giữa hạng mục tính từ tâm hạng mục thứ i đến tâm hạng mục thứ j theo công thức (2.1). Trọng số w_{ij} thể hiện mức độ gần gũi giữa hai hạng mục i và j được thể hiện ở bảng 2.1.

- Tiêu chí lựa chọn trọng số w_{ij} dựa trên tần suất vận chuyển giữa hai hạng mục. Trọng số này phản ánh mức độ di chuyển của vật liệu, thiết bị hoặc nhân lực. Những cặp hạng mục có tần suất vận chuyển càng cao thì nhu cầu bố trí gần nhau càng lớn, nhằm rút ngắn quãng đường di chuyển, giảm thời gian vận hành và hạn chế phát sinh chi phí không cần thiết. Ngược lại, những cặp hạng mục có tần suất vận chuyển ít có thể được bố trí xa hơn, do tác động của khoảng cách đến hiệu quả thi công là không đáng kể.

Bảng 2.1. Trọng số về mức độ gần gũi w_{ij}

STT	Mức độ gần gũi giữa các cơ sở	w_{ij}
1	Không mong muốn	6^0
2	Không quan trọng	6^1
3	Sự gần gũi bình thường	6^2
4	Quan trọng	6^3
5	Đặc biệt quan trọng	6^4
6	Cực kỳ cần thiết	6^5

2.4.2. Hàm mục tiêu tối ưu hóa an toàn lao động

- Bên cạnh việc giảm thiểu khoảng cách di chuyển, mục tiêu an toàn lao động cũng phải được đảm bảo. Khi khoảng cách di chuyển càng giảm thì mức độ an toàn cũng giảm và ít an toàn hơn. Vấn đề an toàn khi bố trí mặt bằng thi công cũng đã được

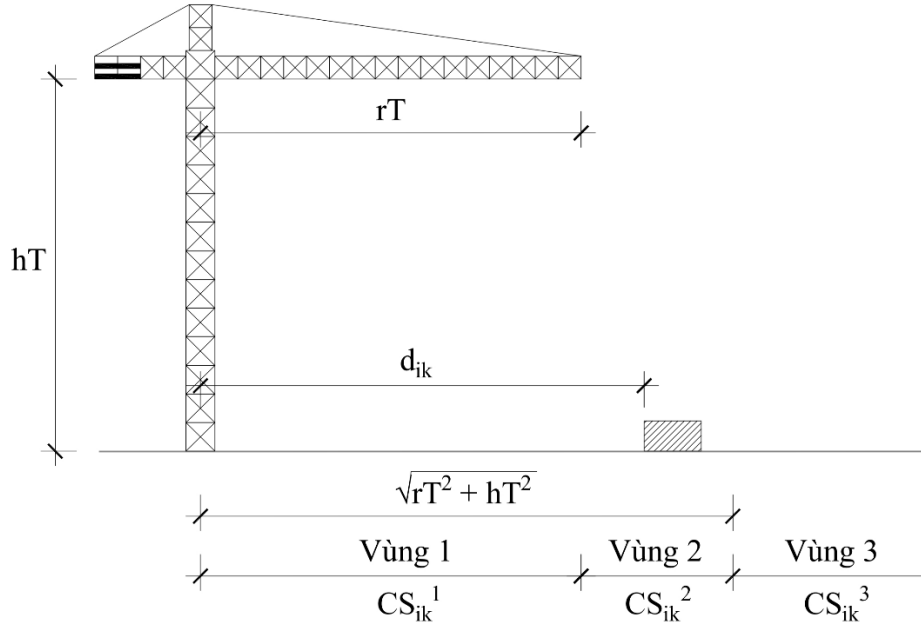
nghiên cứu bởi El-Rayes K, Asce M, Khalafallah A (2005) [7], Xin Ning, Jingyan Qi, Chunlin Wu, Wenjuan Wang (2018) [12]. Trong đề án này, hàm mục tiêu an toàn lao động *MSC* theo công thức (2.8) cần đạt được giá trị lớn nhất gồm 3 yếu tố là an toàn phạm vi hoạt động cần trục tháp *CSC*, an toàn hạng mục nguy hiểm *NHCC*, an toàn giao điểm tuyến đường đông đúc *IPC*.

$$MCS = w_1 \times CSC + w_2 \times NHCC + w_3 \times IPC \quad (2.8)$$

Trong đó w_1, w_2, w_3 là các hệ số tương ứng, phản ánh mức độ ưu tiên của từng yếu tố an toàn. An toàn từ hoạt động của cần trục tháp *CSC* thường được gán hệ số lớn nhất vì phạm vi hoạt động của cần trục tháp có ảnh hưởng trực tiếp đến rủi ro tai nạn nghiêm trọng, bao gồm va chạm, rơi thiết bị, vật liệu. Yếu tố an toàn của các hạng mục nguy hiểm *NHCC* và an toàn tại giao điểm tuyến đường đông đúc *IPC* giữ vai trò quan trọng thấp hơn. El-Rayes K, Asce M, Khalafallah A (2005) [7] đã đề xuất hệ số này là $w_1 = 0,6$; $w_2 = 0,2$; $w_3 = 0,2$.

2.4.2.1. An toàn từ hoạt động cần trục tháp *CSC*

- Cần trục tháp là thiết bị phổ biến để nâng và vận chuyển vật liệu, thiết bị nặng. Tuy nhiên, an toàn lao động liên quan đến phạm vi bán kính hoạt động của cần trục là một vấn đề cần được chú trọng để đảm bảo sự an toàn cho công nhân và các phương tiện làm việc trong khu vực thi công. Các tai nạn liên quan đến cần trục tháp bao gồm rơi vật liệu, đứt cáp, sập đổ cần trục. Các vùng nguy hiểm của cần trục được thể hiện trong hình 2.4. Vùng 1 là khu vực nằm trong phạm vi hoạt động của cần trục, có nguy cơ cao nhất do với vật rơi từ cần trục trong quá trình hoạt động. Vùng 2 nằm giữa vùng 1 và vùng 3 đại diện cho mức độ rủi ro do các tai nạn cần trục có xác suất thấp như nghiêng hoặc sập cần trục. Vùng 3 nằm ngoài các khu vực nguy hiểm của cần trục, do đó mức độ an toàn của nó không bị ảnh hưởng bởi hoạt động của cần trục. Giá trị an toàn cần trục tháp *CSC* được tính theo công thức (2.9).



Hình 2.4. An toàn liên quan đến cần trục tháp

$$CSC = \frac{\sum_{k=1}^K \frac{\sum_{i=1}^I (CS_{ik}^z)}{I}}{K} \quad (2.9)$$

Trong đó K là tổng số lượng cần trục tháp. I là tổng số lượng hạng mục phụ trợ. CS_{ik}^z là hiệu suất an toàn cần trục thứ k cho hạng mục phụ trợ thứ i được tính theo công thức (2.10).

$$CS_{ik}^z = \begin{cases} CS_{ik}^1 = \begin{cases} 0\% \\ 25\% \\ 50\% \end{cases} \\ CS_{ik}^2 = (100 - CS_{ik}^1) \times \left(\frac{d_{ik} - rT}{\sqrt{rT^2 + hT^2} - rT} \right) + CS_{ik}^1 \\ CS_{ik}^3 = 100\% \end{cases} \quad (2.10)$$

Với CS_{ik}^1 là mức độ nhạy cảm trong vùng hoạt động của cần trục tháp, độ nhạy cảm cao là 0%, độ nhạy cảm trung bình là 25%, độ nhạy cảm thấp là 50%.

CS_{ik}^2 là mức độ nhạy cảm trong vùng có rủi ro do các tai nạn cần trực có xác suất thấp như nghiêng hoặc sập cần trực. d_{ik} là khoảng cách từ cần trực thứ k đến điểm góc gần nhất của hạng mục phụ trợ thứ i . rT là bán kính cần trực. hT là chiều cao cần trực. CS_{ik}^3 là mức độ nhạy cảm trong vùng nằm ngoài các khu vực nguy hiểm của cần trực tháp.

2.4.2.2. An toàn từ hạng mục nguy hiểm NHCC

- Những hạng mục nguy hiểm thường gặp trên công trình bao gồm vật liệu dễ cháy và các chất độc hại. Vật liệu dễ cháy như nhiên liệu cho máy móc, thiết bị cần được đặt cách xa các nguồn điện, thiết bị tạo tia lửa hoặc nhiệt độ cao để giảm thiểu nguy cơ cháy nổ. Các chất độc hại thì cần được cách xa khu vực đông người để tối ưu an toàn về sức khỏe. Giá trị an toàn hạng mục nguy hiểm NHCC được tính theo công thức (2.11).

$$NHCC = \frac{HCC - HCC_{\min}}{HCC_{\max} - HCC_{\min}} \quad (2.11)$$

$$HCC = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (HCW_{ij} \times d_{ij}) \quad (2.12)$$

Trong đó HCC là tiêu chí kiểm soát rủi ro được tính theo công thức (2.12). $HCC_{\min}$ là giá trị tối thiểu thu được từ tất cả các bố trí công trường đã tạo. $HCC_{\max}$ là giá trị tối đa thu được từ tất cả các bố trí công trường đã tạo. d_{ij} là khoảng cách giữa hạng mục tính từ tâm hạng mục thứ i đến tâm hạng mục thứ j theo công thức (2.1). HCW_{ij} là trọng số kiểm soát rủi ro được thể hiện trong bảng 2.2, biểu thị mức độ nguy cơ tai nạn có thể xảy ra trên công trường nếu các cơ sở i và j không được tách biệt đầy đủ. Đối với những hạng mục có nguy cơ cháy nổ cao như kho nhiên liệu cho máy thi công khi bố trí gần các hạng mục có khả năng phát sinh tia lửa hoặc nhiệt độ cao, giá trị HCW_{ij} sẽ được chọn ở mức rất cao 100% nhằm nhấn mạnh yêu cầu bố trí tách biệt. Trong khi đó, các mối quan hệ tiềm ẩn nguy cơ trung bình được gán giá trị trọng số trung bình 50%, phản ánh mức độ yêu cầu tách biệt vừa phải. Những cặp hạng mục hầu như không tồn tại nguy cơ tai nạn như nhà điều

hành và bãi tập kết vật liệu không nguy hại, sẽ được gán trọng số thấp hoặc bằng 0% để không ảnh hưởng đến hàm mục tiêu.

Bảng 2.2. Trọng số kiểm soát rủi ro HCW_{ij}

STT	Mức độ mối nguy	HCW_{ij}
1	Rất cao	100%
2	Cao	75%
3	Trung bình	50%
4	Thấp	25%
5	Không có mối nguy	0%

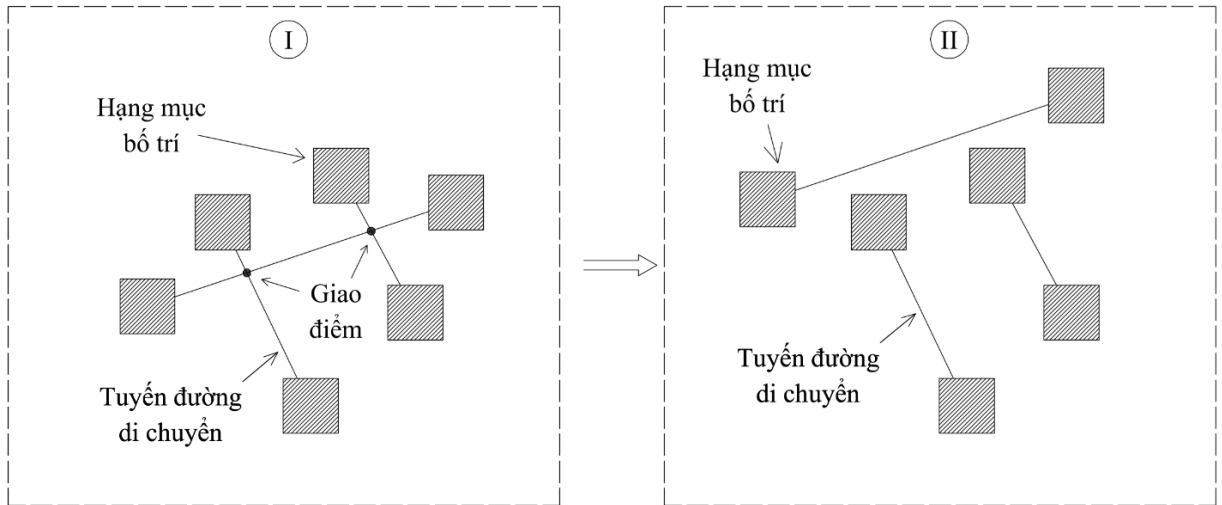
2.4.2.3. An toàn tại giao điểm tuyến đường đông đúc IPC

- Trong quá trình thi công, sự di chuyển thường xuyên của các nguồn lực như lao động, vật liệu và thiết bị có thể làm tăng nguy cơ xảy ra tai nạn khi các tuyến đường di chuyển giao nhau. Những điểm giao nhau giữa các tuyến đường có lưu lượng cao thường tiềm ẩn rủi ro lớn do khả năng va chạm và sự cố tăng cao tại các vị trí này. Vì vậy, việc tối ưu hóa bố trí mặt bằng công trường nhằm giảm thiểu số lượng điểm giao nhau giữa các tuyến đường có lưu lượng cao là một giải pháp quan trọng để cải thiện an toàn. Ví dụ, trong hình 2.5, bố trí II mang lại mức độ an toàn cao hơn so với bố trí I vì nó loại bỏ các điểm giao cắt không cần thiết. Tiêu chí điểm giao nhau IPC được tính như công thức (2.13).

$$IPC = \left(1 - \frac{IP}{IP_{\max}}\right) \times 100\% \quad (2.13)$$

$$IP_{\max} = \frac{NR \times (NR - 1)}{2} \quad (2.14)$$

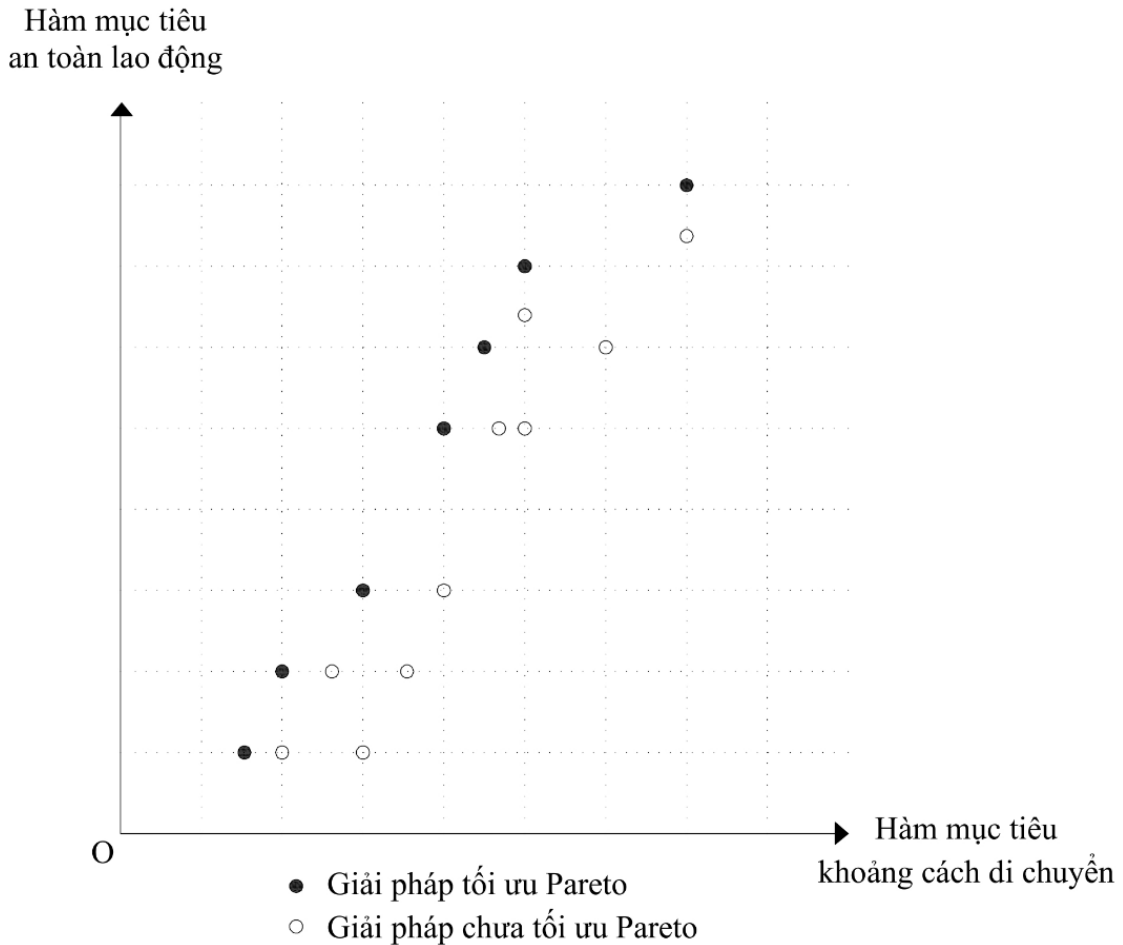
Trong đó IP là tổng số điểm giao nhau trên công trình được tạo ra giữa tất cả các tuyến đường có lưu lượng di chuyển cao. $IP_{\max}$ là số điểm giao nhau tối đa có thể có giữa tất cả các tuyến đường có lưu lượng cao, được tính theo công thức (2.14). NR là tổng số tuyến đường có lưu lượng cao trên công trường.



Hình 2.5. Các điểm giao nhau giữa các tuyến đường có lưu lượng di chuyển cao

2.5. Khái niệm tối ưu Pareto

- Trong quá trình bố trí tối ưu tổng mặt bằng thi công, việc sắp xếp các hạng mục như kho vật liệu, máy móc, khu vực làm việc ảnh hưởng trực tiếp đến hiệu quả thi công và an toàn lao động. Hai mục tiêu chính mâu thuẫn với nhau là hàm mục tiêu khoảng cách di chuyển và hàm mục tiêu an toàn lao động. Khoảng cách giữa các hạng mục càng ngắn thì việc vận chuyển vật liệu và thiết bị càng thuận tiện, giảm chi phí và thời gian thi công nhưng không tối ưu về an toàn. Khoảng cách giữa các hạng mục càng xa thì nguy cơ xảy ra tai nạn lao động càng giảm. Vì vậy cần tìm các giải pháp Pareto tối ưu là tập hợp các giải pháp bố trí mà không thể cải thiện khoảng cách di chuyển mà không làm giảm an toàn lao động và ngược lại. Các giải pháp được gọi là tối ưu Pareto được trình bày trong hình 2.6.



Hình 2.6. Mô tả tối ưu Pareto

2.6. Mô hình hóa bài toán bằng phần mềm MATLAB

- MATLAB là một phần mềm mạnh mẽ được sử dụng rộng rãi trong các lĩnh vực kỹ thuật, khoa học và toán học, cung cấp môi trường lập trình lý tưởng để tính toán số học, phân tích dữ liệu, phát triển thuật toán, mô phỏng hệ thống. Trong bài toán tối ưu tổng mặt bằng thi công này, phần mềm MATLAB được sử dụng để giải quyết các vấn đề tối ưu hóa có nhiều mục tiêu, đặc biệt là khi áp dụng các thuật toán tối ưu như NSGA-II, MOPSO. Code lập trình thuật toán tối ưu tổng mặt bằng thi công NSGA-II và NSGA-II lai MOPSO được trình bày ở phụ lục 01.

2.7. Tổng kết chương 2

- Chương 2 đã trình bày quá trình xây dựng mô hình tối ưu hóa bố trí tổng mặt bằng thi công, trong đó làm rõ đặc điểm và vai trò của các hạng mục phụ trợ như cần trục tháp, bãi tập kết vật liệu, nhà điều hành, lán trại nghỉ ngơi và các khu vực chức năng

khác. Các hạng mục phụ trợ này có mối quan hệ chặt chẽ và ảnh hưởng trực tiếp đến hiệu quả tổ chức thi công, chi phí vận hành và mức độ an toàn lao động.

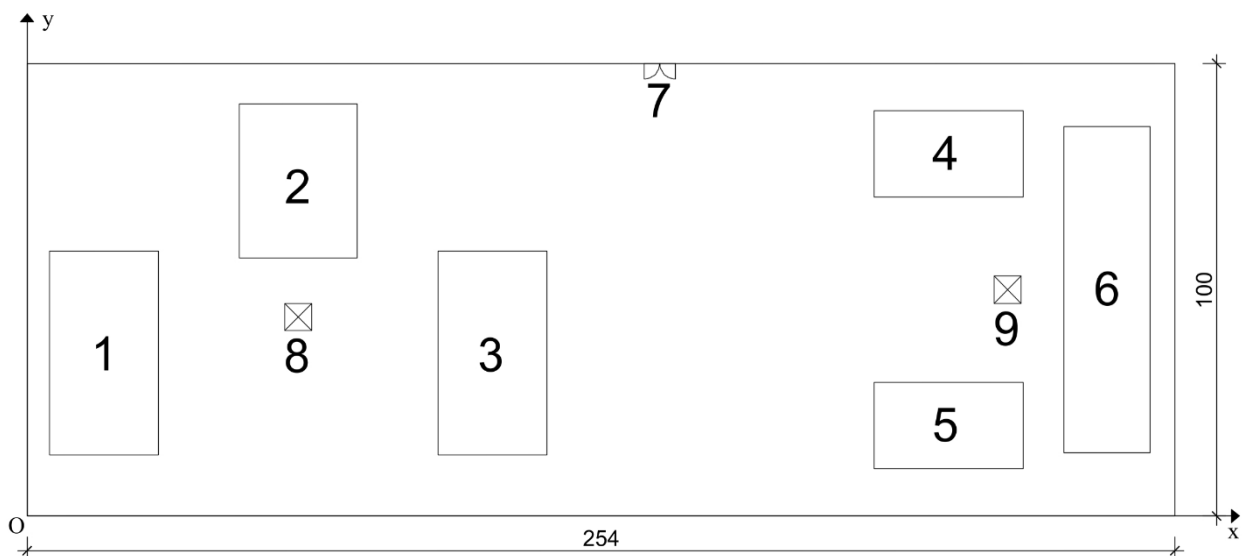
- Chương đã xác định các giả định, đồng thời thiết lập hệ thống ràng buộc không gian, bao gồm giới hạn diện tích mặt bằng, phạm vi hoạt động của cần trục tháp và khoảng cách tối thiểu nhằm tránh chồng lặp giữa các hạng mục. Hai hàm mục tiêu chính của mô hình bao gồm tối thiểu hóa tổng khoảng cách di chuyển và tối đa hóa mức độ an toàn lao động. Trong đó, hàm mục tiêu về an toàn được tính thông qua ba tiêu chí rủi ro từ hoạt động của cần trục tháp, từ các hạng mục nguy hiểm và tại các giao điểm tuyến đường đông đúc. Mô hình hướng tới tối ưu hóa đa mục tiêu theo khái niệm Pareto, nhằm thu được tập nghiệm không bị chi phối lẫn nhau.

- Cuối cùng, bài toán được mô hình hóa và lập trình trên phần mềm MATLAB, làm cơ sở cho việc ứng dụng và đánh giá hiệu quả mô hình trên các trường hợp trong chương tiếp theo.

Chương 3 - ỨNG DỤNG MÔ HÌNH TỐI ƯU HÓA MẶT BẰNG THI CÔNG

3.1. Áp dụng mô hình cho công trình 1

- Mô hình thi công 1 mô phỏng sơ đồ tổng thể Green Park CT15 Việt Hưng như hình 3.1. Tối ưu tổng mặt bằng thi công trong giai đoạn phân thân với các thông tin ở bảng 3.1. Các tòa nhà từ số 1 đến số 6 được bố trí theo thiết kế, do đó vị trí của chúng là cố định. Cần trực tiếp được lựa chọn vị trí và số lượng cụ thể, đảm bảo đáp ứng yêu cầu về tầm với và khả năng nâng tải. Công ra vào công trình cũng đã được xác định về kích thước và vị trí. Đối với các hạng mục phụ trợ còn lại, việc bố trí sẽ được tiến hành theo hướng tối ưu nhằm nâng cao hiệu quả sử dụng mặt bằng và phục vụ thi công thuận lợi nhất. Các bãi tập kết cấu kiện đúc sẵn được dùng cho tòa nhà số 1, 2, 3, đối với tòa nhà số 4, 5, 6 sử dụng phương pháp thi công đổ tại chỗ. Bảng 3.2 là trọng số gần gũi giữa các hạng mục để tính hàm mục tiêu khoảng cách di chuyển. Bảng 3.3 là trọng số kiểm soát rủi ro giữa các hạng mục, được tính bằng %, dùng để tính khoảng cách an toàn giữa các hạng mục nguy hiểm. Bảng 3.4 thể hiện mật độ di chuyển giữa các hạng mục, 0 là mật độ di chuyển thấp, 1 là mật độ di chuyển cao, để tính an toàn giao điểm giữa các hạng mục có mật độ di chuyển cao.



Hình 3.1. Mặt bằng mô hình thi công 1

Bảng 3.1. Thông tin các hạng mục của mô hình thi công 1

T T	Tên hạng mục	Tọa độ trọng tâm		Kích thước		Khoảng cách tối thiểu (m)	Hệ số phụ thuộc cần trục	CS _{ik} ¹ (%)
		x (m)	y (m)	w (m)	h (m)			
1	Tòa nhà 1	17	36	24	45	3	-	-
2	Tòa nhà 2	60	74	25	34	3	-	-
3	Tòa nhà 3	103	36	24	45	3	-	-
4	Tòa nhà 4	204	80	33	19	3	-	-
5	Tòa nhà 5	204	20	33	19	3	-	-
6	Tòa nhà 6	239	50	19	72	3	-	-
7	Cổng	140	100	7	80	1	-	-
8	Cần trục tháp 1	60	44	6	6	4	-	-
9	Cần trục tháp 2	217	50	6	6	4	-	-
10	Nhà bảo vệ	-	-	4	4	1	-	25
11	Bãi cầu kiện đúc sẵn 1	-	-	5	5	3	1	25
12	Bãi cầu kiện đúc sẵn 2	-	-	5	5	3	1	25
13	Bãi cầu kiện đúc sẵn 3	-	-	5	5	3	1	25
14	Kho thiết bị	-	-	6	6	2	-	50
15	Kho chứa nhiên liệu	-	-	5	5	2	-	50
16	Máy phát điện dự phòng	-	-	3	5	2	-	50
17	Bãi tập kết gạch	-	-	5	5	2	2	0
18	Bãi tập kết cát	-	-	5	5	2	-	0
19	Bãi tập kết đá	-	-	5	5	2	-	0
20	Bãi tập kết xi măng	-	-	5	5	2	-	0
21	Bãi tập kết cốt thép	-	-	12	5	2	2	0
22	Bãi gia công cốt thép	-	-	12	12	2	2	0
23	Bãi dàn giáo, ván khuôn	-	-	5	5	2	2	25
24	Bãi trộn bê tông, vữa	-	-	5	5	2	2	0
25	Nhà vệ sinh	-	-	2	2	2	-	25
26	Nhà để xe	-	-	6	12	2	-	25
27	Lán trại nghỉ ngơi	-	-	5	10	2	-	0
28	Nhà làm việc cán bộ	-	-	6	3	2	-	25

Bảng 3.2. Trọng số gần gũi giữa các hạng mục mô hình thi công 1

TT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	0	0	0	0	0	0	0	0	0	1	6⁄4	6⁄4	6⁄4	6⁄2	1	6⁄3	1	1	1	1	1	1	1	1	6⁄2	6⁄2	6⁄2	6⁄2	
2	0	0	0	0	0	0	0	0	0	1	6⁄4	6⁄4	6⁄4	6⁄2	1	6⁄3	1	1	1	1	1	1	1	1	6⁄2	6⁄2	6⁄2	6⁄2	
3	0	0	0	0	0	0	0	0	0	1	6⁄4	6⁄4	6⁄4	6⁄2	1	6⁄3	1	1	1	1	1	1	1	1	6⁄2	6⁄2	6⁄2	6⁄2	
4	0	0	0	0	0	0	0	0	0	1	1	1	1	6⁄2	1	6⁄3	6⁄3	1	1	1	6⁄3	6⁄3	6⁄3	6⁄3	6⁄2	6⁄2	6⁄2	6⁄2	
5	0	0	0	0	0	0	0	0	0	1	1	1	1	6⁄2	1	6⁄3	6⁄3	1	1	1	6⁄3	6⁄3	6⁄3	6⁄3	6⁄2	6⁄2	6⁄2	6⁄2	
6	0	0	0	0	0	0	0	0	0	1	1	1	1	6⁄2	1	6⁄3	6⁄3	1	1	1	6⁄3	6⁄3	6⁄3	6⁄3	6⁄2	6⁄2	6⁄2	6⁄2	
7	0	0	0	0	0	0	0	0	0	6⁄4	6⁄1	6⁄1	6⁄1	6⁄1	6⁄1	6⁄1	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	1	6⁄2	1	1	6⁄1	1	6⁄3
8	0	0	0	0	0	0	0	0	0	1	6⁄2	6⁄2	6⁄2	1	1	6⁄4	1	1	1	1	1	1	1	1	1	1	1	1	
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	6⁄4	6⁄2	1	1	1	6⁄2	6⁄2	6⁄2	6⁄2	1	1	1	1	
10	1	1	1	1	1	1	6⁄4	1	1	0	1	1	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	6⁄2	1	1	
11	6⁄4	6⁄4	6⁄4	1	1	1	6⁄1	6⁄2	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
12	6⁄4	6⁄4	6⁄4	1	1	1	6⁄1	6⁄2	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
13	6⁄4	6⁄4	6⁄4	1	1	1	6⁄1	6⁄2	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
14	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄1	1	1	6⁄2	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
15	1	1	1	1	1	1	6⁄1	1	1	1	1	1	1	1	0	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	
16	6⁄3	6⁄3	6⁄3	6⁄3	6⁄3	6⁄3	6⁄1	6⁄4	6⁄4	1	1	1	1	1	6⁄2	0	1	1	1	1	1	6⁄2	1	6⁄2	1	1	1	1	
17	1	1	1	6⁄3	6⁄3	6⁄3	6⁄2	1	6⁄2	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	
18	1	1	1	1	1	1	6⁄2	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	6⁄3	1	1	1	1	
19	1	1	1	1	1	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	6⁄3	1	1	1	1	
20	1	1	1	1	1	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	6⁄3	1	1	1	1	
21	1	1	1	6⁄3	6⁄3	6⁄3	6⁄2	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	0	6⁄4	1	1	1	1	1	
22	1	1	1	6⁄3	6⁄3	6⁄3	1	1	6⁄2	1	1	1	1	1	1	6⁄2	1	1	1	1	1	6⁄4	0	1	1	1	1	1	
23	1	1	1	6⁄3	6⁄3	6⁄3	6⁄2	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	
24	1	1	1	6⁄3	6⁄3	6⁄3	1	1	6⁄2	1	1	1	1	1	1	6⁄2	1	6⁄3	6⁄3	6⁄3	1	1	1	0	1	1	1	1	
25	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	6⁄2	1	
26	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄1	1	1	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	
27	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6⁄2	1	0	
28	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄2	6⁄3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	

Bảng 3.3. Trọng số kiểm soát rủi ro mô hình thi công 1

[illegible]

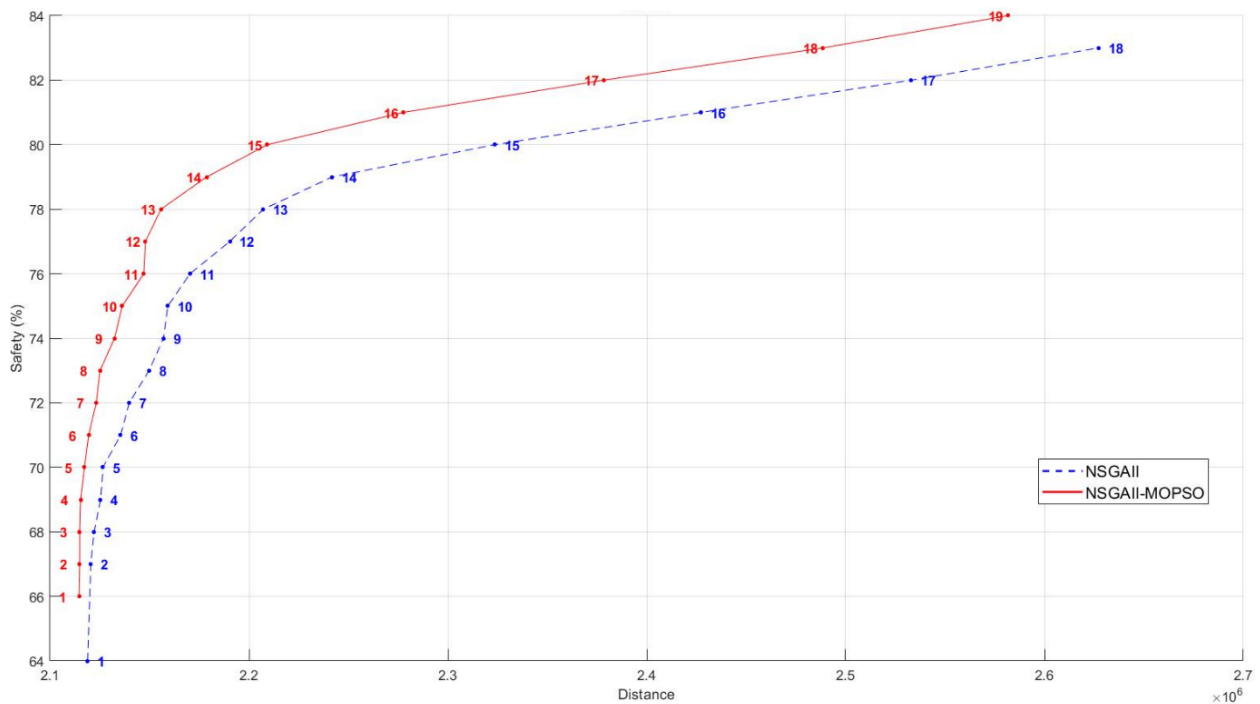
Bảng 3.4. Hệ số mật độ di chuyển mô hình thi công 1

[illegible]

- Tiến hành chạy bài toán tối ưu bằng phần mềm MATLAB với các tham số đầu vào như bảng 3.5.

Bảng 3.5. Tham số đầu vào cho mô hình thi công 1

TT	Tham số đầu vào	Thuật toán	
		NSGA-II	NSGA-II lai MOPSO
1	Số vòng lặp tối đa	300	300
2	Kích thước quần thể	200	200
3	Tỷ lệ lai ghép NSGAI	0,7	0,7
4	Tỷ lệ đột biến NSGAI	0,4	0,4
5	Hệ số giữ lại vận tốc ban đầu w	-	0,9
6	Hệ số gia tốc cá thể c_1	-	10
7	Hệ số gia tốc quần thể c_2	-	1
8	Tỷ lệ đột biến MOPSO	-	0,2



Hình 3.2. Biểu đồ các giải pháp Pareto mô hình thi công 1

- Sau khi chạy thuật toán tối ưu hóa bằng phần mềm MATLAB được biểu đồ các giải pháp như hình 3.2. Biểu đồ thể hiện mối quan hệ giữa khoảng cách di chuyển Distance và hệ số an toàn Safety của hai thuật toán tối ưu hóa: NSGA-II và NSGA-

II lai MOPSO. Các điểm trên đường cong đại diện cho các nghiệm không bị chi phối Pareto tương ứng với từng thuật toán.

- Kết quả cho thấy thuật toán NSGA-II lai MOPSO (đường nét liền) cho tập nghiệm Pareto tối ưu hơn so với NSGA-II (đường nét đứt). Cụ thể:

+ Với cùng mức độ khoảng cách, các nghiệm của NSGA-II lai MOPSO có hệ số an toàn cao hơn so với NSGA-II.

+ Với cùng giá trị an toàn, NSGA-II lai MOPSO đạt được tổng quãng đường di chuyển nhỏ hơn.

+ Đường cong của NSGA-II lai MOPSO phân bố đều và có xu hướng tiệm cận biên trên của đồ thị, phản ánh khả năng duy trì tính đa dạng và khai thác nghiệm tốt hơn.

+ Ngoài ra, thuật toán lai cũng cho số lượng nghiệm Pareto nhiều hơn (19 nghiệm so với 18), cho thấy khả năng tìm kiếm giải pháp tốt hơn.

- Đối với thuật toán NSGA-II, từ giải pháp số 1 đến giải pháp số 10 giá trị khoảng cách di chuyển tăng ít nhưng giá trị an toàn tăng đáng kể, từ giải pháp số 11 đến giải pháp số 18 giá trị khoảng cách di chuyển tăng mạnh nhưng giá trị an toàn tăng ít, có thể chọn giải pháp số 10 để bố trí tổng mặt bằng.

- Đối với thuật toán NSGA-II lai MOPSO:

+ Giải pháp số 1 có giá trị khoảng cách tối thiểu được thể hiện ở hình 3.3. Bãi tập kết cát, đá, xi măng được đặt gần bãi trộn bê tông, vữa. Bãi tập kết cốt thép và gia công cốt thép được đặt cạnh nhau. Các bãi tập kết gạch, dàn giáo ván khuôn, bãi gia công cốt thép và bãi trộn bê tông đều được đặt sát vào cần trục tháp 2 và cách đều tòa nhà 4, 5, 6 để tối ưu vận chuyển. Bãi tập kết cấu kiện đúc sẵn số 1, 2, 3 được đặt sát vào cần trục tháp 1 và tòa nhà 1, 2, 3.

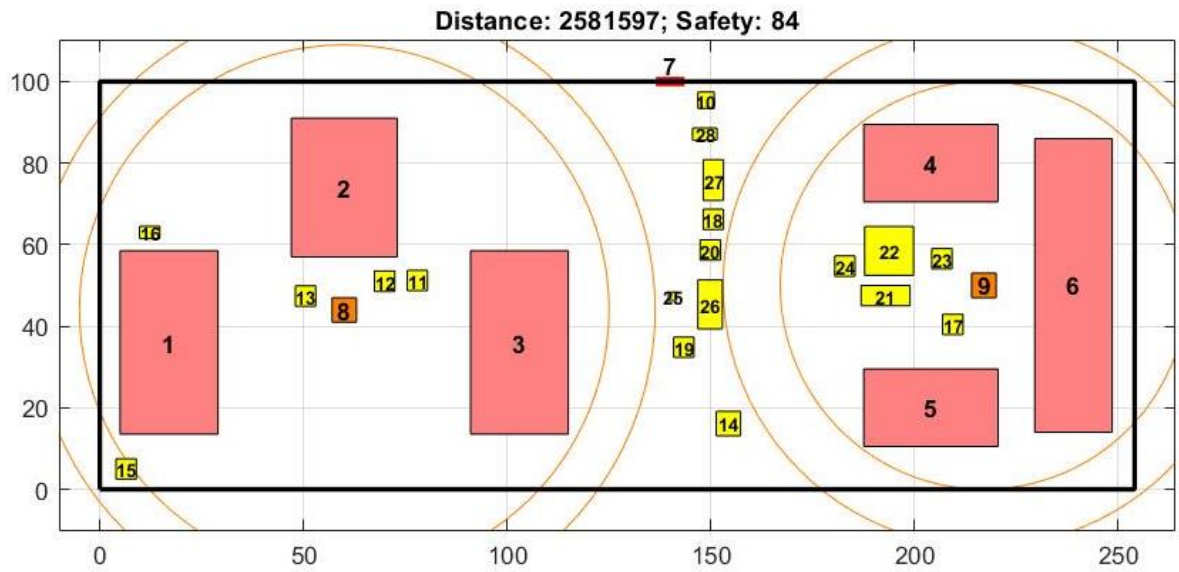
+ Giải pháp số 19 có giá trị an toàn tối đa được thể hiện ở hình 3.4. Các hạng mục phụ trợ không cần vận chuyển bằng cần trục như cát, đá, xi măng được đặt hoàn toàn bên ngoài phạm vi ảnh hưởng của cần trục để đảm bảo an toàn. Các hạng mục phụ trợ cần sự vận chuyển của cần trục như bãi trộn bê tông,

vữa, bãi tập kết và gia công thép, dàn giáo, ván khuôn, gạch, bãi tập kết cấu kiện đúc sẵn vẫn được đặt gần cần trục để đảm bảo tối ưu vận chuyển. Hạng mục nguy hiểm như kho chứa nhiên liệu và máy phát điện dự phòng đã được đặt cách xa nhau và các hạng mục còn lại.

+ Có thể chọn giải pháp số 9 để bố trí khi giá trị khoảng cách di chuyển tăng ít nhưng vẫn đảm bảo giá trị an toàn không quá thấp như hình 3.5. Các hạng mục phụ trợ như cát, đá, xi măng đã được đặt bên ngoài phạm vi nguy hiểm nhất của cần trục và nằm trong vùng nguy hiểm thứ 2 để đảm bảo khoảng cách đến bãi trộn bê tông, vữa không quá xa.



Hình 3.3. Mặt bằng mô hình 1, giải pháp số 1, thuật toán NSGA-II lai MOPSO



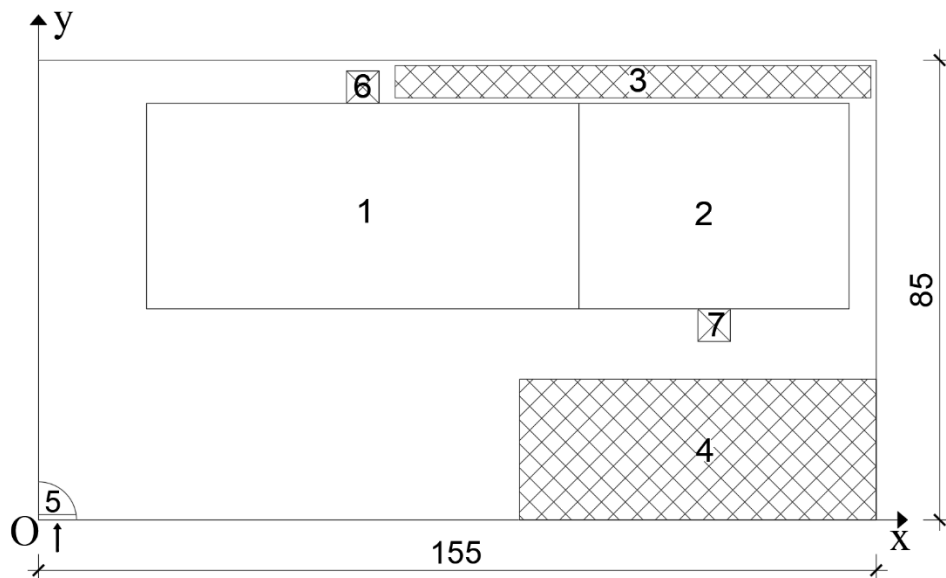
Hình 3.4. Mặt bằng mô hình 1, giải pháp số 19, thuật toán NSGA-II lai MOPSO



Hình 3.5. Mặt bằng mô hình 1, giải pháp số 9, thuật toán NSGA-II lai MOPSO

3.2. Áp dụng mô hình cho công trình 2

- Mô hình thi công 2 mô phỏng một phần mặt bằng của tổ hợp chung cư Eco Green Sài Gòn như hình 3.6. Tối ưu tổng mặt bằng thi công trong giai đoạn thi công phần thân với các thông tin được trình bày ở bảng 3.6. Tòa nhà số 1 và số 2 được giữ nguyên vị trí theo thiết kế. Số lượng và vị trí của hai cần trục tháp cũng đã được lựa chọn cố định, đảm bảo yêu cầu về tầm với và khả năng nâng tải. Cổng ra vào công trình được xác định về kích thước và vị trí. Các hạng mục phụ trợ còn lại sẽ được bố trí theo hướng tối ưu nhằm nâng cao hiệu quả sử dụng không gian và phục vụ tốt cho quá trình thi công. Vị trí số 3, số 4 là khu vực hạn chế diện tích thi công do vướng công trình hiện trạng và diện tích đất thi công. Bảng 3.7 là trọng số gần gũi giữa các hạng mục để tính hàm mục tiêu khoảng cách di chuyển. Bảng 3.8 là trọng số kiểm soát rủi ro giữa các hạng mục, được tính bằng %, dùng để tính khoảng cách an toàn giữa các hạng mục nguy hiểm. Bảng 3.9 thể hiện mật độ di chuyển giữa các hạng mục, 0 là mật độ di chuyển thấp, 1 là mật độ di chuyển cao, để tính an toàn giao điểm giữa các hạng mục có mật độ di chuyển cao.



Hình 3.6. Mặt bằng mô hình thi công 2

Bảng 3.6. Thông tin các hạng mục của mô hình thi công 2

T T	Tên hạng mục	Tọa độ trọng tâm		Kích thước		Khoảng cách tối thiểu (m)	Hệ số phụ thuộc cần trục	CS _{ik} (%)
		x (m)	y (m)	w (m)	h (m)			
1	Tòa nhà 1	60	58	80	38	-	-	-
2	Tòa nhà 2	125	58	50	38	-	-	-
3	Khu vực hạn chế thi công	110	81	88	6	-	-	-
4	Khu vực hạn chế thi công	122	13	66	26	-	-	-
5	Cổng	3	0	6	40	-	-	-
6	Cần trục tháp 1	60	80	6	6	-	-	-
7	Cần trục tháp 2	125	36	6	6	-	-	-
8	Nhà bảo vệ	-	-	3	3	1	-	25
9	Kho thiết bị	-	-	3	6	1	-	50
10	Kho chứa nhiên liệu	-	-	4	4	1	-	50
11	Máy phát điện dự phòng	-	-	2	3	1	-	50
12	Bãi tập kết cát, đá	-	-	7	4	2	-	25
13	Bãi tập kết xi măng	-	-	4	4	2	-	25
14	Bãi tập kết, gia công thép 1	-	-	6	12	2	1	0
15	Bãi tập kết, gia công thép 2	-	-	6	12	2	2	0
16	Bãi trộn bê tông, vữa 1	-	-	3	3	2	1	0
17	Bãi trộn bê tông, vữa 2	-	-	3	3	2	2	0
18	Nhà vệ sinh	-	-	1	2	1	-	25
19	Nhà để xe	-	-	3	5	1	-	25
20	Lán trại nghỉ ngơi	-	-	4	6	1	-	0
21	Nhà làm việc cán bộ	-	-	2	5	1	-	0

Bảng 3.7. Trọng số gần gũi giữa các hạng mục mô hình thi công 2

TT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0	0	0	0	0	0	0	1	6 ³	1	6 ³	1	1	6 ⁴	1	6 ⁴	1	6 ²	6 ³	6 ³	6 ³
2	0	0	0	0	0	0	0	1	6 ³	1	6 ³	1	1	1	6 ⁴	1	6 ⁴	6 ²	6 ³	6 ³	6 ³
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	6 ⁴	1	1	1	6 ³	6 ³	6 ²	6 ²	1	1	1	6 ²	1	6 ³
6	0	0	0	0	0	0	0	1	1	1	6 ³	1	1	6 ⁴	1	6 ⁴	1	1	1	1	1
7	0	0	0	0	0	0	0	1	1	1	6 ³	1	1	1	6 ⁴	1	6 ⁴	1	1	1	1
8	1	1	1	1	6 ⁴	1	1	0	6 ³	1	6 ¹	1	1	1	1	1	1	6 ¹	6 ³	1	6 ³
9	6 ³	6 ³	1	1	1	1	1	6 ³	0	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	0	6 ³	1	1	1	1	1	1	1	1	1	1
11	6 ³	6 ³	1	1	1	6 ³	6 ³	6 ¹	1	6 ³	0	1	1	6 ²	6 ²	6 ²	6 ²	1	1	1	1
12	1	1	1	1	6 ³	1	1	1	1	1	1	0	1	1	1	6 ⁴	6 ⁴	1	1	1	1
13	1	1	1	1	6 ³	1	1	1	1	1	1	1	0	1	1	6 ⁴	6 ⁴	1	1	1	1
14	6 ⁴	1	1	1	6 ²	6 ⁴	1	1	1	1	6 ²	1	1	0	1	1	1	6 ¹	1	6 ³	1
15	1	6 ⁴	1	1	6 ²	1	6 ⁴	1	1	1	6 ²	1	1	1	0	1	1	6 ¹	1	6 ³	1
16	6 ⁴	1	1	1	1	6 ⁴	1	1	1	1	6 ²	6 ⁴	6 ⁴	1	1	0	1	6 ¹	1	6 ³	1
17	1	6 ⁴	1	1	1	1	6 ⁴	1	1	1	6 ²	6 ⁴	6 ⁴	1	1	1	0	6 ¹	1	6 ³	1
18	6 ²	6 ²	1	1	1	1	1	6 ¹	1	1	1	1	1	6 ¹	6 ¹	6 ¹	6 ¹	0	1	6 ³	6 ²
19	6 ³	6 ³	1	1	6 ²	1	1	6 ³	1	1	1	1	1	1	1	1	1	1	0	1	1
20	6 ³	6 ³	1	1	1	1	1	1	1	1	1	1	1	6 ³	6 ³	6 ³	6 ³	6 ³	1	0	1
21	6 ³	6 ³	1	1	6 ³	1	1	6 ³	1	1	1	1	1	1	1	1	1	6 ²	1	1	0

Bảng 3.8. Trọng số kiểm soát rủi ro mô hình thi công 2

[illegible]

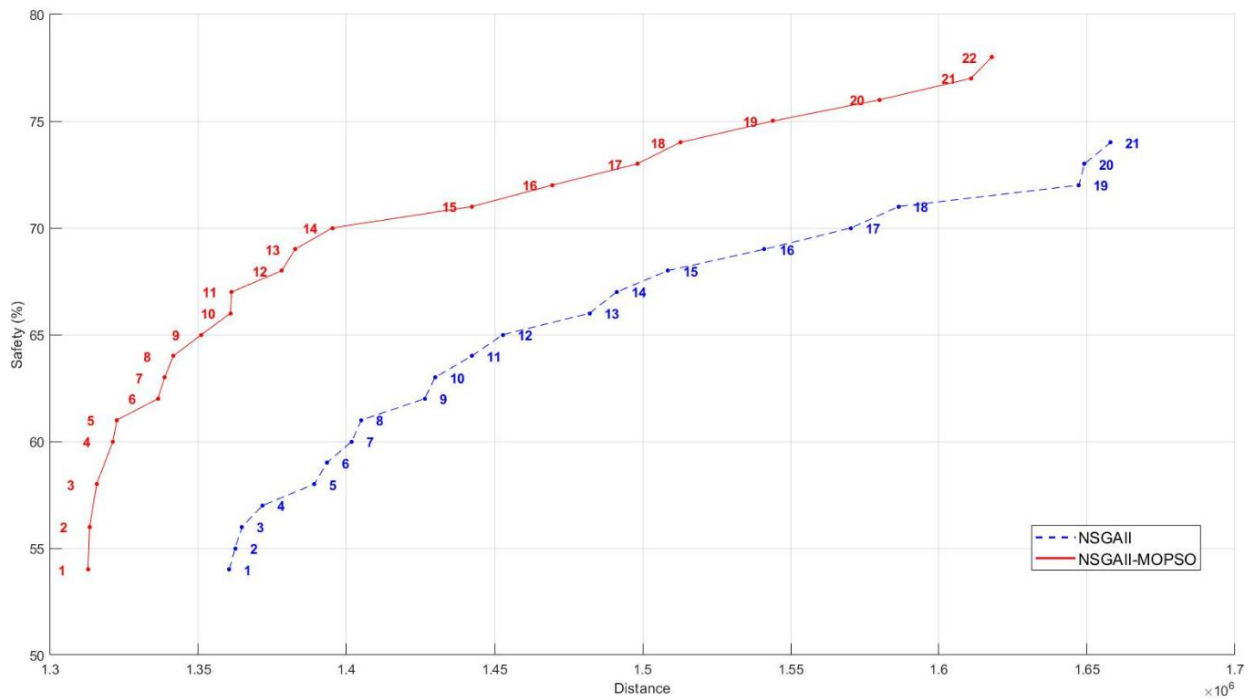
Bảng 3.9. Hệ số mật độ di chuyển mô hình thi công 2

[illegible]

- Tiến hành chạy bài toán tối ưu bằng phần mềm MATLAB với các tham số đầu vào như bảng 3.10.

Bảng 3.10. Tham số đầu vào cho mô hình thi công 2

TT	Tham số đầu vào	Thuật toán	
		NSGA-II	NSGA-II lai MOPSO
1	Số vòng lặp tối đa	300	300
2	Kích thước quần thể	200	200
3	Tỷ lệ lai ghép NSGAI	0,7	0,7
4	Tỷ lệ đột biến NSGAI	0,4	0,4
5	Hệ số giữ lại vận tốc ban đầu w	-	0,9
6	Hệ số gia tốc cá thể c_1	-	10
7	Hệ số gia tốc quần thể c_2	-	1
8	Tỷ lệ đột biến MOPSO	-	0,2



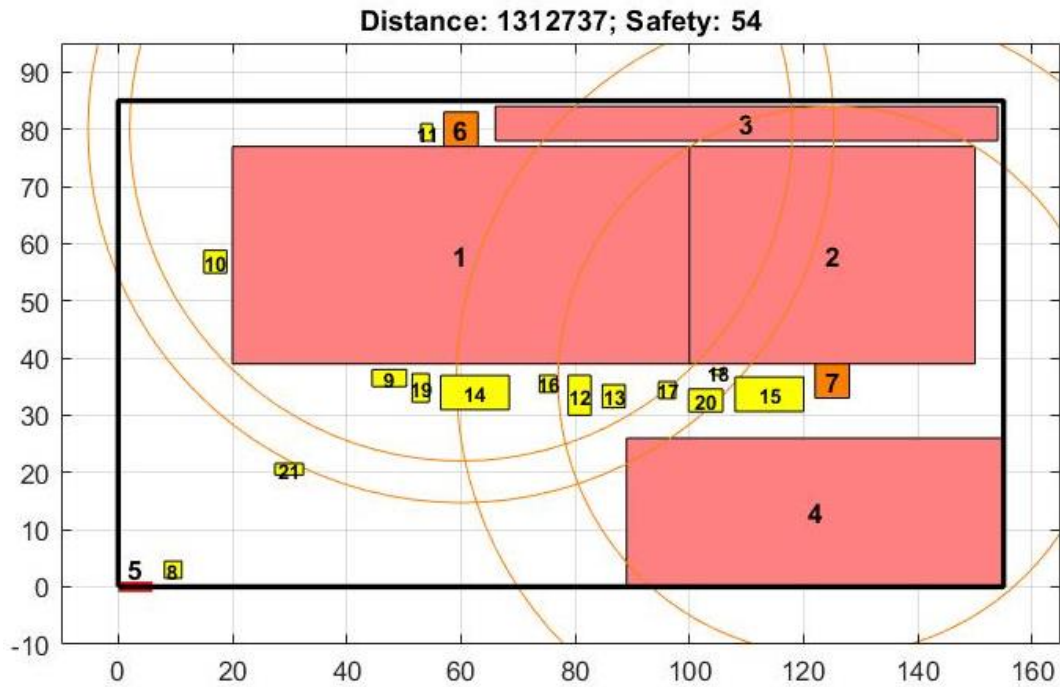
Hình 3.7. Biểu đồ các giải pháp Pareto mô hình thi công 2

- Sau khi chạy thuật toán tối ưu hóa bằng phần mềm MATLAB được biểu đồ các giải pháp như hình 3.7. Biểu đồ thể hiện mối quan hệ giữa khoảng cách di chuyển

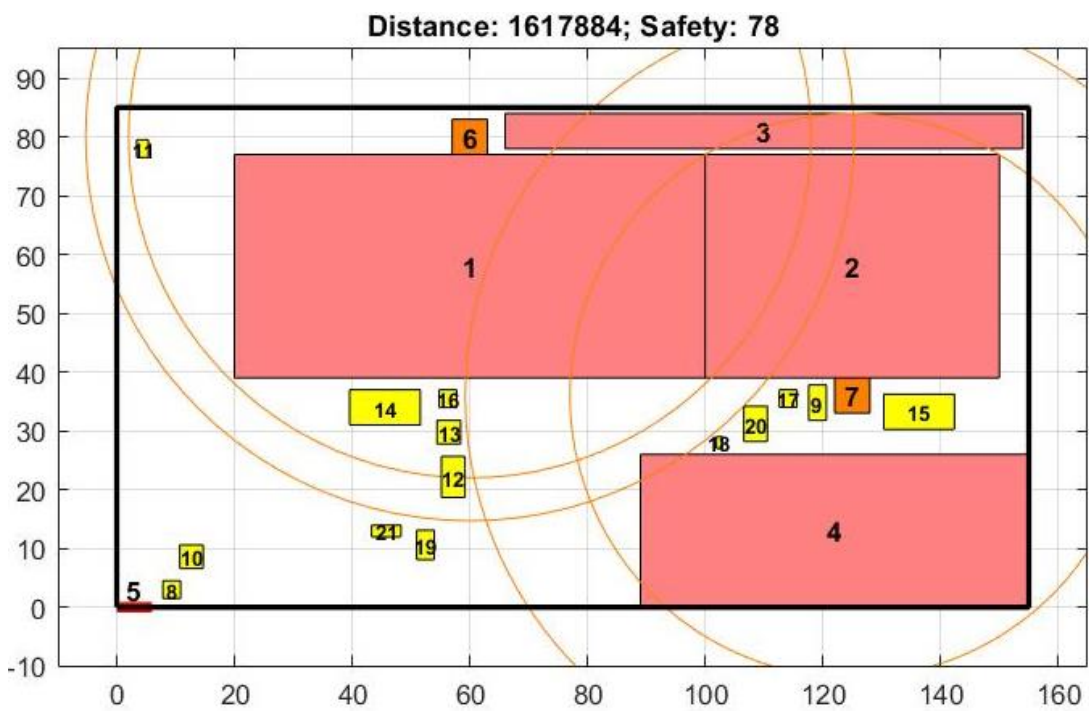
Distance và mức độ an toàn Safety của hai thuật toán được so sánh là NSGA-II (nét đứt) và NSGA-II lai MOPSO (nét liền). Quan sát biểu đồ hình 3.7 cho thấy:

- + Đường bao Pareto của NSGA-II lai MOPSO vượt trội hơn so với NSGA-II tại mọi vùng giá trị. Ở mỗi mức khoảng cách, NSGA-II lai MOPSO luôn đạt được hệ số an toàn cao hơn, với cùng mức an toàn, khoảng cách mà NSGA-II lai MOPSO đạt được luôn thấp hơn.
 - + Tập nghiệm của NSGA-II lai MOPSO rộng hơn (22 nghiệm so với 21 của NSGA-II), cho thấy khả năng tìm kiếm nghiệm đa dạng và toàn diện hơn của thuật toán lai.
- Đối với thuật toán NSGA-II, các giải pháp tìm được trên biểu đồ thay đổi đều cả về giá trị khoảng cách và giá trị an toàn, tùy vào lựa chọn của người bố trí tổng mặt bằng có thể lựa chọn 1 trong 21 giải pháp tối ưu Pareto.
- Đối với thuật toán NSGA-II lai MOPSO:
- + Giải pháp số 1 có giá trị khoảng cách tối thiểu được thể hiện ở hình 3.8. Bãi tập kết cát, đá, xi măng được đặt ở giữa 2 bãi trộn bê tông, vữa. Bãi tập kết, gia công cốt thép 1 được đặt sát tòa nhà số 1 nhưng không nằm sát cản trực tháp 1 vì diện tích xung quanh cản trực 1 không đủ. Bãi tập kết, gia công cốt thép 2 được đặt gần cản trực tháp 2 và tòa nhà 2.
 - + Giải pháp có giá trị an toàn tối đa được thể hiện ở hình 3.9. Các hạng mục phụ trợ như bãi trộn bê tông, bãi tập kết và gia công cốt thép cho mỗi tòa nhà được đặt sát vào mỗi cản trực, không đặt trong phạm vi chồng diện tích ảnh hưởng của 2 cản trực để đảm bảo an toàn. Hạng mục nguy hiểm như kho chứa nhiên liệu và máy phát điện dự phòng đã được đặt cách xa nhau và các hạng mục còn lại.
 - + Có thể chọn giải pháp số 9 để bố trí khi giá trị khoảng cách di chuyển tăng ít nhưng vẫn đảm bảo giá trị an toàn không quá thấp như hình 3.10. Láng trại nghỉ ngơi nơi tập trung nhiều người được đặt ra khỏi phạm vi ảnh hưởng chung của 2 cản trực. Nhà để xe đã được đặt ra khỏi phạm vi ảnh hưởng của

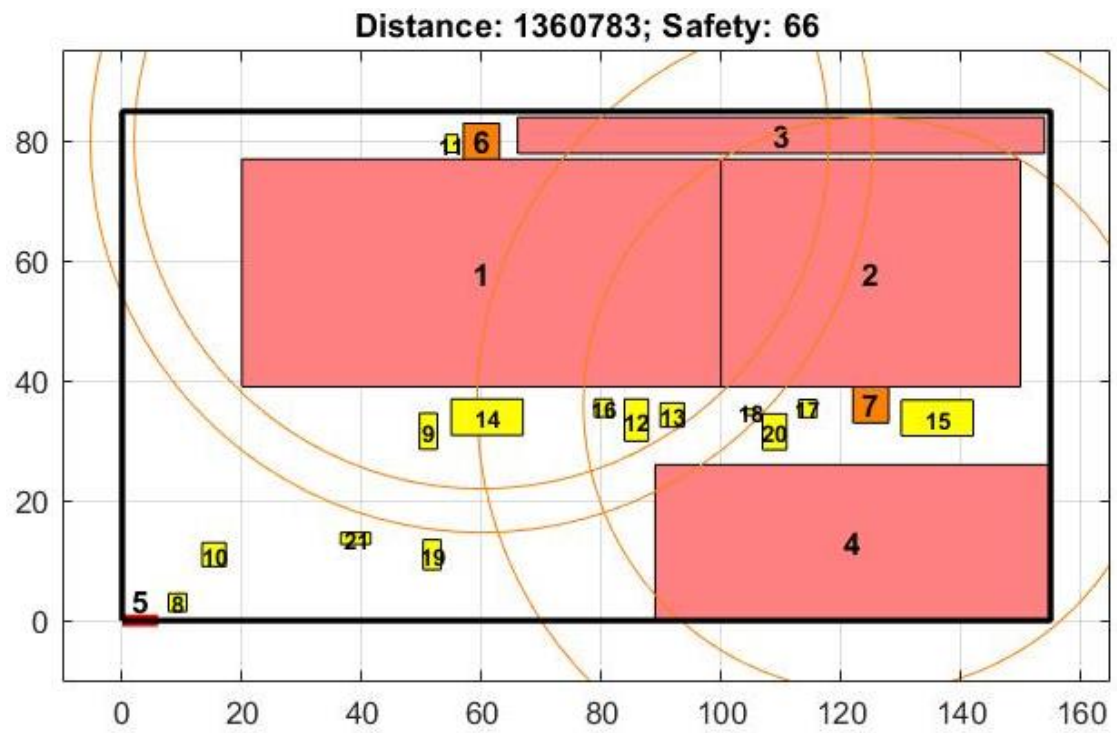
cần trực. Hạng mục nguy hiểm như kho chứa nhiên liệu cũng được đặt cách xa các hạng mục khác hơn giải pháp số 1.



Hình 3.8. Mặt bằng mô hình 2, giải pháp số 1, thuật toán NSGA-II lai MOPSO



Hình 3.9. Mặt bằng mô hình 2, giải pháp số 22, thuật toán NSGA-II lai MOPSO



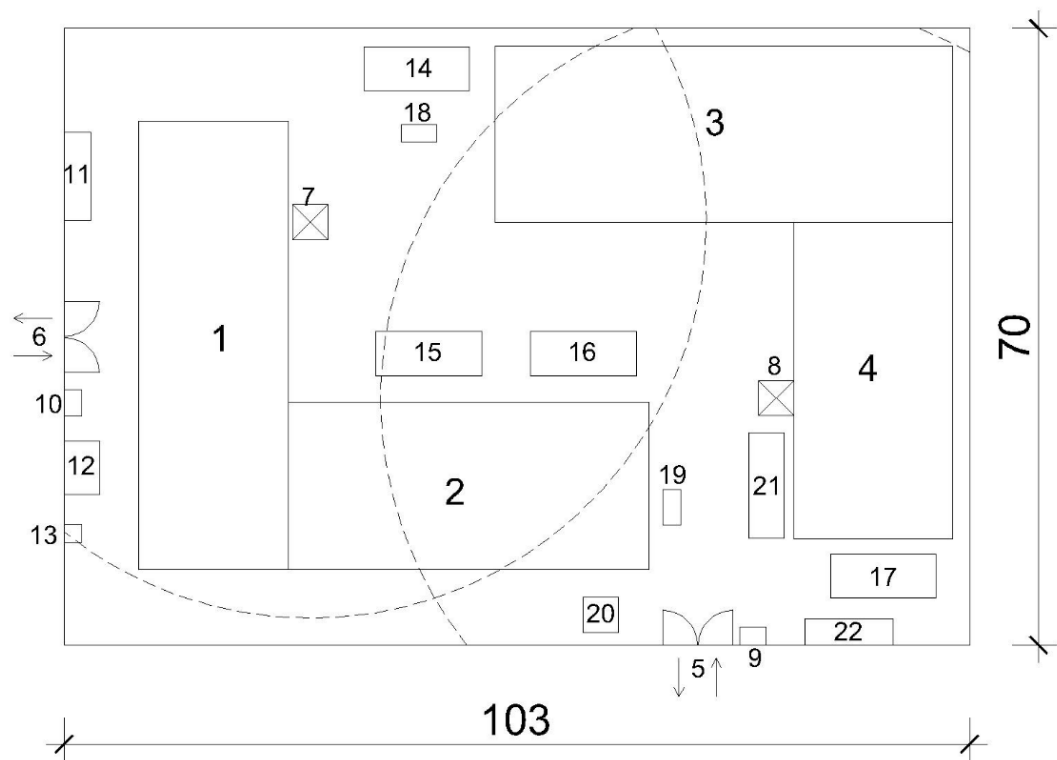
Hình 3.10. Mặt bằng mô hình 2, giải pháp số 10, thuật toán NSGA-II lai MOPSO

3.3. Áp dụng mô hình cho công trình 3

- Mô hình thi công 3 mô phỏng tổng mặt bằng bố trí công trình của dự án Simona Heights ở Quy Nhơn. Hình ảnh thực tế tại công trình theo hình 3.11 được mô phỏng lại như hình 3.12. Tiến hành bố trí tổng mặt bằng thi công trong giai đoạn thi công phần thân với thông tin các hạng mục được trình bày ở bảng 3.11.
- Các hạng mục cố định bao gồm xây dựng 4 tòa nhà, 2 căn trực tháp, 2 cổng ra vào được cố định vị trí theo mặt bằng hiện trạng. Các hạng mục phụ trợ còn lại như nhà bảo vệ, phòng ban chỉ huy, nhà kho, nhà vệ sinh, khu vực tập kết, gia công thép, khu vực tập kết ván khuôn, xà bần, bãi để xe, lán trại công nhân sẽ được bố trí theo hướng tối ưu nhằm nâng cao hiệu quả khoảng cách di chuyển và an toàn lao động. Các bảng 3.12, bảng 3.13, bảng 3.14 là bảng giá trị khảo sát ý kiến của 1 chỉ huy trưởng công trình, 1 cán bộ kỹ thuật xây dựng theo phụ lục 02.



Hình 3.11 Mặt bằng thực tế bố trí thi công dự án Simona Heights



Hình 3.12 Mặt bằng mô phỏng bố trí thi công dự án Simona Heights

Bảng 3.11. Thông tin các hạng mục của mô hình thi công 3

T T	Tên hạng mục	Tọa độ trọng tâm		Kích thước		Khoảng cách tối thiểu (m)	Hệ số phụ thuộc cần trục	CS _{ik} (%)
		x (m)	y (m)	w (m)	h (m)			
1	Tòa nhà 1	17	34	17	51	-	-	-
2	Tòa nhà 2	46	18	41	19	-	-	-
3	Tòa nhà 3	75	58	52	20	-	-	-
4	Tòa nhà 4	92	30	18	36	-	-	-
5	Cổng ra vào 1	72	0	8	54	-	-	-
6	Cổng ra vào 2	0	35	16	8	-	-	-
7	Cần trục tháp 1	28	48	4	4	-	-	-
8	Cần trục tháp 2	81	28	4	4	-	-	-
9	Nhà bảo vệ 1	-	-	3	2	1	-	0
10	Nhà bảo vệ 2	-	-	3	2	1	-	0
11	Phòng ban chỉ huy	-	-	3	10	2	-	0
12	Nhà kho	-	-	4	6	1	-	50
13	Nhà vệ sinh	-	-	2	2	1	-	25
14	Tập kết, gia công thép 1	-	-	5	12	2	1	0
15	Tập kết, gia công thép 2	-	-	5	12	2	1	0
16	Tập kết, gia công thép 3	-	-	5	12	2	2	0
17	Tập kết, gia công thép 4	-	-	5	12	2	2	0
18	Tập kết ván khuôn 1	-	-	4	2	2	1	50
19	Tập kết ván khuôn 2	-	-	4	2	2	2	50
20	Tập kết xà bần	-	-	4	4	2	-	50
21	Bãi để xe	-	-	4	12	2	-	0
22	Lán trại công nhân	-	-	3	10	2	-	0

Bảng 3.12. Trọng số gần gũi giữa các hạng mục mô hình thi công 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	1	1	1	1	6/2	6/5	1	1	1	6/3	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	6/2	6/2	1	6/5	1	1	1	6/3	1	1	1
3	0	0	0	0	0	0	0	0	1	1	1	6/2	6/2	1	1	6/5	1	6/3	1	1	1	1
4	0	0	0	0	0	0	0	0	1	1	1	6/2	6/2	1	1	1	6/5	1	6/3	1	1	1
5	0	0	0	0	0	0	0	0	6/5	1	6/5	1	1	6/2	6/2	6/2	6/2	6/2	6/2	6/2	1	1
6	0	0	0	0	0	0	0	0	1	6/5	1	1	1	6/2	6/2	6/2	6/2	6/2	6/2	6/3	6/3	6/3
7	0	0	0	0	0	0	0	0	1	1	1	1	1	6/3	6/3	6/3	6/3	6/3	6/3	6/3	1	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	6/3	6/3	6/3	6/3	6/3	6/3	6/3	1	1
9	1	1	1	1	6/5	1	1	1	0	1	1	6/3	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	6/5	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	6/3
11	1	1	1	1	6/5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
13	1	6/2	6/2	6/2	1	1	1	1	6/3	1	1	0	1	1	1	1	1	1	1	1	1	1
14	6/2	6/2	6/2	6/2	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	6/3
15	6/5	1	1	1	6/2	6/2	6/3	6/3	1	1	1	1	1	0	1	1	1	1	1	1	1	1
16	1	6/5	1	1	6/2	6/2	6/3	6/3	1	1	1	1	1	1	0	1	1	1	1	1	1	1
17	1	1	6/5	1	6/2	6/2	6/3	6/3	1	1	1	1	1	1	1	0	1	1	1	1	1	1
18	1	1	1	6/5	6/2	6/2	6/3	6/3	1	1	1	1	1	1	1	1	0	1	1	1	1	1
19	6/3	1	6/3	1	6/2	6/2	6/3	6/3	1	1	1	1	1	1	1	1	1	0	1	1	1	1
20	1	6/3	1	6/3	6/2	6/2	6/3	6/3	1	1	1	1	1	1	1	1	1	1	0	1	1	1
21	1	1	1	1	1	6/3	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
22	1	1	1	1	1	6/3	1	1	1	6/3	1	1	1	1	1	1	1	1	1	1	0	6/2
23	1	1	1	1	1	6/3	1	1	1	1	1	1	6/3	1	1	1	1	1	1	1	6/2	0

Bảng 3.13. Trọng số kiểm soát rủi ro mô hình thi công 3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	25	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	25	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	25	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	25	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	25	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	75	75	75	75	0	0	0	0	0	0	25	0	0	0	50	50	50	50	0	0	0	50
15	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	25	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	25	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	25	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	25	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	25	25	25	25	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	50
22	0	0	0	0	0	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	50	0	0	0

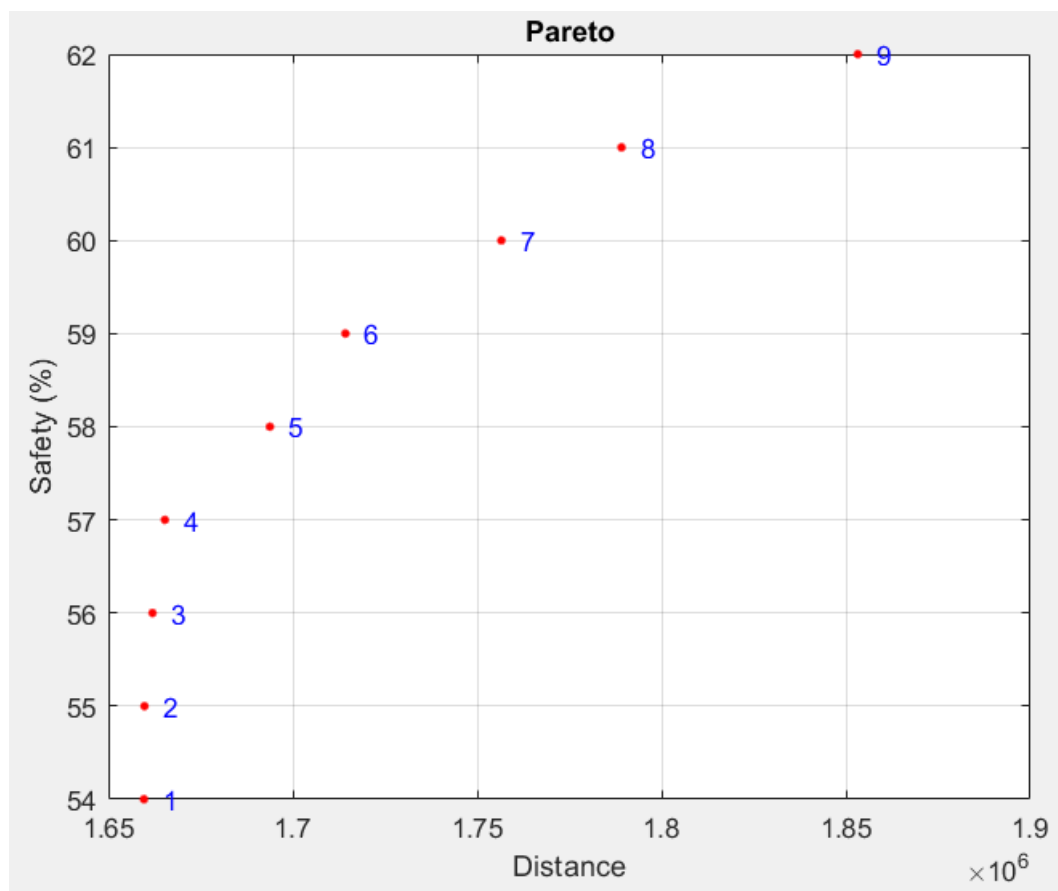
Bảng 3.14. Hệ số mật độ di chuyển mô hình thi công 3

[illegible]

- Tiến hành chạy bài toán tối ưu bằng phần mềm MATLAB với các tham số đầu vào như bảng 3.15.

Bảng 3.15. Tham số đầu vào cho mô hình thi công 3

TT	Tham số đầu vào	Thuật toán NSGA-II lai MOPSO
1	Số vòng lặp tối đa	400
2	Kích thước quần thể	400
3	Tỷ lệ lai ghép NSGAI	0,7
4	Tỷ lệ đột biến NSGAI	0,4
5	Hệ số giữ lại vận tốc ban đầu w	0,9
6	Hệ số gia tốc cá thể c_1	10
7	Hệ số gia tốc quần thể c_2	1
8	Tỷ lệ đột biến MOPSO	0,2



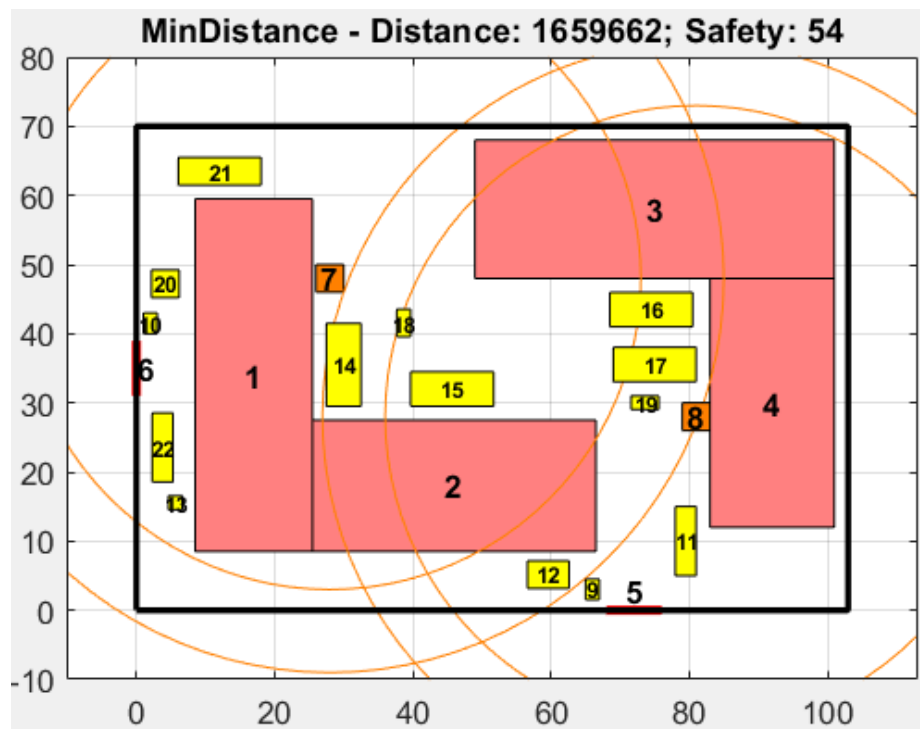
Hình 3.13. Biểu đồ các giải pháp Pareto mô hình thi công 3

- Hình 3.13 trình bày tập nghiệm Pareto thu được khi áp dụng thuật toán NSGA-II lai MOPSO. Tập nghiệm gồm 9 phương án Pareto phân bố đều thể hiện mối quan hệ đánh đổi giữa hai mục tiêu. Cụ thể, khi khoảng cách di chuyển tăng dần từ khoảng $1,66 \times 10^6$ đến $1,853 \times 10^6$, hệ số an toàn cũng tăng từ 54% lên đến 62%. Điều này cho thấy để đạt mức độ an toàn cao hơn, cần chấp nhận phương án có tổng quãng đường di chuyển lớn hơn. Ngược lại, nếu ưu tiên rút ngắn khoảng cách, mức độ an toàn sẽ giảm tương ứng.

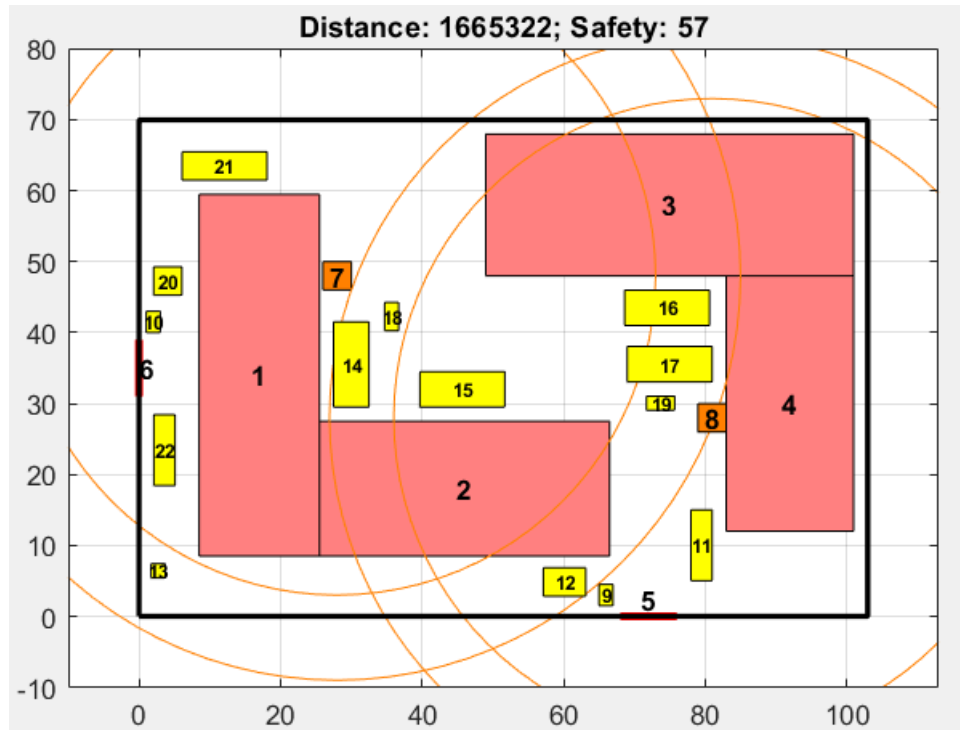
- Giải pháp số 1 như hình 3.14 có khoảng cách di chuyển ít nhất và mức độ an toàn thấp nhất. Giải pháp số 9 như hình 3.16 có khoảng cách di chuyển nhiều nhất và mức độ an toàn cao nhất. Giải pháp số 4 như hình 3.15 được cải thiện mức độ an toàn nhiều nhưng khoảng cách di chuyển tăng ít.

- Cả 3 giải pháp số 1, số 4, số 9 đều bố trí nhà bảo vệ, tập kết xà bần, lán trại ở gần cổng ra vào giống như bố trí mặt bằng thực tế, văn phòng ban chỉ huy được ở gần cổng ra vào số 5 giúp quan sát công trình được bao quát hơn, nhà kho của giải pháp tối ưu và bố trí thực tế đều được bố trí gần nhà trực bảo vệ để đảm bảo an toàn và gần cổng ra vào để thuận tiện cho việc vận chuyển. Bãi để xe được đặt gần cổng ra vào và nhà bảo vệ để đảm bảo an toàn.

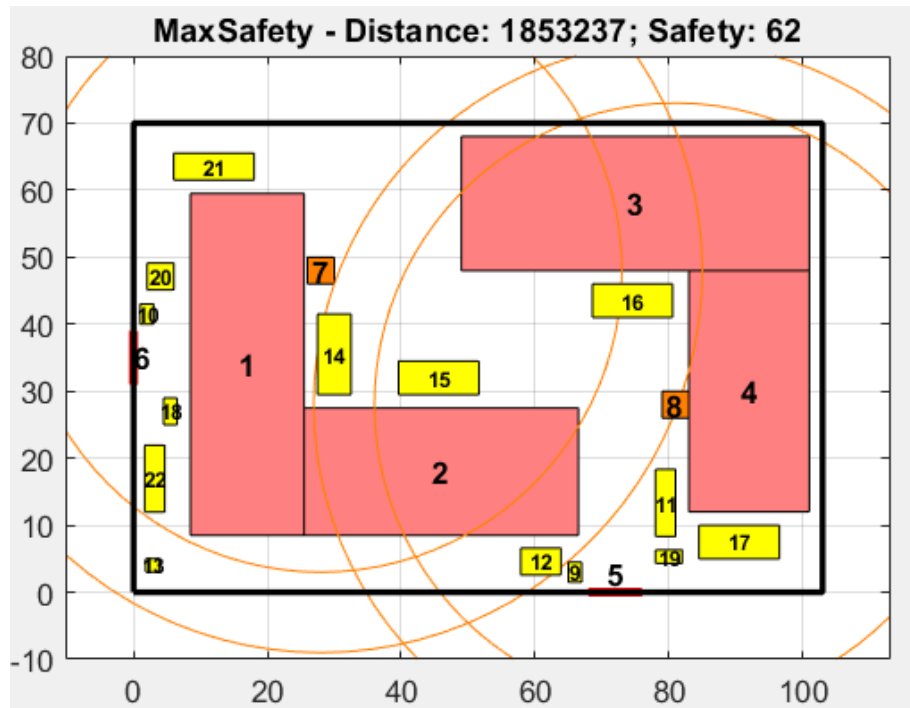
- Khu vực tập kết, gia công thép ở giải pháp số 1, số 4 đã được đặt sát vào từng tòa nhà thi công. Đối với giải pháp số 9, khu vực tập kết, gia công thép 4 đã được tối ưu hơn. Khu vực tập kết ván khuôn ở giải pháp số 1 gần vị trí thi công nhưng ít đảm bảo an toàn hơn so với giải pháp số 4, số 9. Nhà vệ sinh ở giải pháp số 1 và bố trí thực tế đều được bố trí trong khu vực nguy hiểm của cần trục tháp 1, ở giải pháp số 4, số 9 vị trí đã được tối ưu ra khỏi phạm vi ảnh hưởng nguy hiểm nhất của cần trục tháp 1.



Hình 3.14. Mặt bằng mô hình 3, giải pháp số 1, thuật toán NSGA-II lai MOPSO



Hình 3.15. Mặt bằng mô hình 3, giải pháp số 4, thuật toán NSGA- II lai MOPSO



Hình 3.16. Mặt bằng mô hình 3, giải pháp số 9, thuật toán NSGA-II lai MOPSO

3.4. Tổng kết chương 3

- Trong quá trình nghiên cứu và ứng dụng, ba mô hình mặt bằng thi công đã được xây dựng để đánh giá hiệu quả của thuật toán. Các kết quả thu được cho thấy ở mô hình công trình 1 thuật toán NSGA-II lai MOPSO cho giá trị tổng khoảng cách di chuyển thấp hơn từ 1% đến 6% so với NSGA-II khi cùng mức giá trị an toàn, ở mô hình công trình 2 mức cải thiện này lên đến 11%, chứng minh khả năng khai thác không gian giải pháp của thuật toán NSGA-II lai MOPSO tốt hơn NSGA-II thuần túy.

- Đối với mô hình công trình 3, chỉ áp dụng thuật toán NSGA-II lai MOPSO để tạo ra 9 phương án so sánh với phương án bố trí mặt bằng thực tế. Về mặt không gian bố trí cả ba giải pháp 1, 4, 9 đều có các bố trí tương đồng và bố trí tối ưu hơn so với phương án bố trí thực tế như các hạng mục như nhà bảo vệ, khu tập kết xà bần, lán trại đều được đặt gần cổng ra vào tương tự phương án thực tế, tạo điều kiện thuận lợi cho quản lý và kiểm soát công trình, kho vật tư và bãi để xe được đặt gần cổng và khu bảo vệ, thuận tiện cho vận chuyển và tăng cường yếu tố an toàn. Các khu

vực như gia công thép, tập kết ván khuôn, nhà vệ sinh đều được tối ưu an toàn nằm ngoài tầm hoạt động của cần trục tháp trong một số phương án.

- Như vậy, việc ứng dụng thuật toán NSGA-II lai MOPSO bằng phần mềm MATLAB đã chứng minh được sự hiệu quả trong bố trí mặt bằng thi công, không chỉ rút ngắn quãng đường vận chuyển, tăng mức độ an toàn, mà còn giúp tạo ra nhiều phương án bố trí linh hoạt, từ đó hỗ trợ kỹ sư lựa chọn giải pháp tối ưu phù hợp điều kiện công trường.

KẾT LUẬN VÀ KIẾN NGHỊ

Kết luận

Đề án đã xây dựng thành công mô hình tối ưu hóa bố trí tổng mặt bằng thi công dựa trên hai mục tiêu tối thiểu hóa khoảng cách di chuyển và tối đa hóa mức độ an toàn. Hai thuật toán NSGA-II và NSGA-II lai MOPSO được áp dụng để tối ưu hóa, từ đó so sánh hiệu quả. Kết quả cho thấy thuật toán NSGA-II lai MOPSO cho tập nghiệm Pareto chất lượng hơn, đa dạng hơn so với NSGA-II. Việc áp dụng mô hình tối ưu này không chỉ cải thiện hiệu quả sử dụng không gian mà còn hỗ trợ ban điều hành công trình có thêm nhiều phương án hợp lý để lựa chọn bố trí mặt bằng.

Kiến nghị

Mô hình đề xuất đã chứng minh hiệu quả trong việc tối ưu mặt bằng cho các công trình có dạng hình học đơn giản, nhưng khi áp dụng cho các công trình có hình dạng phức tạp như hình tam giác, hình tròn thì độ chính xác của thuật toán giảm do chưa mô phỏng đúng hình dáng thực tế. Trong nghiên cứu tiếp theo, cần phát triển thêm mô hình biểu diễn mặt bằng linh hoạt hơn, tích hợp thêm yếu tố thay đổi mặt bằng theo từng giai đoạn thi công nhằm phù hợp với thực tiễn triển khai.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[10] Trần Văn Khiêm (2018), “TỐI ƯU HOÁ CHẾ ĐỘ CẮT PHỤC VỤ ĐIỀU KHIỂN THÍCH NGHI QUÁ TRÌNH GIA CÔNG,” Tạp chí Khoa học và Công nghệ. Available: jst.tnu.edu.vn/jst/article/viewFile/503/pdf.

Tiếng Anh

- [1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan (2002), “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, doi: 10.1109/4235.996017.
- [2] Carlos A. Coello Coello and Maximino Salazar Lechuga (2002), “MOPSO : A Proposal for Multiple Objective Particle Swarm Optimization,” *IEEE*, doi: 10.1109/CEC.2002.1004388.
- [3] V. Benjaoran and V. Peansupap (2019), “Grid-based construction site layout planning with Particle Swarm Optimisation and Travel Path Distance,” *Construction Management and Economics*, vol. 38, no. 8, pp. 673–688, doi: 10.1080/01446193.2019.1600708.
- [4] D.-N. Nguyen Bui and N. Duc Bui (2023), “Optimize location tower crane and supply facilities on construction site by discrete PSO algorithm,” *Journal of Science and Transport Technology University of Transport Technology JSTT*, no. 2, pp. 1–10, 2023, doi: 10.58845/jstt.utt.2.
- [5] M. Oral, S. Bazaati, S. Aydinli, and E. Oral (2018), “Construction site layout planning: Application of multi-objective particle swarm optimization,” *Teknik Dergi/Technical Journal of Turkish Chamber of Civil Engineers*, vol. 29, no. 6, pp. 8691–8713, doi: 10.18400/tekderg.389638.
- [6] H. Zhang and H. Li (2010), “Multi-objective particle swarm optimization for construction time-cost tradeoff problems,” *Construction Management and Economics*, vol. 28, no. 1, pp. 75–88, doi: 10.1080/01446190903406170.

- [7] K. El-Rayes, M. Asce, and A. Khalafallah (2005), “Trade-off between Safety and Cost in Planning Construction Site Layouts,” *J Constr Eng Manag*, doi: 10.1061/ASCE0733-93642005131:111186.
- [8] Rachmat Parayoga and Anna Maria Sri Asih (2021), “Empirical study of MOPSO and NSGA II comparison in multi-objective location routing problem incorporating the service level of delivery,” *IEOM Society*. Available: ieomsociety.org/proceedings/2021indonesia/282.pdf
- [9] A. Mohamad Izdin Hlal, V. K. Ramachandaramurthya, B. Sanjeevikumar Padmanaban, C. Hamid Reza Kaboli, A. Aref Pouryekta, and D. Tuan Ab Rashid Bin Tuan Abdullah (2019), “NSGA-II and MOPSO based optimization for sizing of hybrid PV / wind / battery energy storage system,” *International Journal of Power Electronics and Drive Systems*, vol. 10, no. 1, pp. 463–478, doi: 10.11591/ijpeds.v10.i1.pp463-478.
- [11] C. Duong and V. Peansupap (2023), “A Development of Optimization Model for Construction Site Layout Planning Using Genetic Algorithm,” *The 28th National Convention on Civil Engineering*. Available: conference.thaince.org/index.php/ncce28/article/view/2459/1335
- [12] X. Ning, J. Qi, C. Wu, and W. Wang (2018), “A tri-objective ant colony optimization-based model for planning safe construction site layout,” *Autom Constr*, vol. 89, pp. 1–12, doi: 10.1016/j.autcon.2018.01.007.

PHỤ LỤC 01

1. Phần chương trình chính thuật toán NSGA-II

```

clc;
clear;
close all;
global lb ub
global nD nG nT nA
%% Problem Definition
CreateModel;
CostFunction=@(x)tf1(x); % Cost Function
nObj=2;% Number of Objective Functions
%% NSGA-II Parameters
MaxIt=300; % Maximum Number of Iterations
nPop=200; % Population Size
pCrossover=0.7; % Crossover Percentage
nCrossover=2*round(pCrossover*nPop/2); % Number of Parnets (Offsprings)
nCrossover=round(nCrossover/2);
pMutation=0.4; % Mutation Percentage
nMutation=round(pMutation*nPop); % Number of Mutants
nMutation=round(nMutation/2);
mug=0.01; % Mutation Rate
sigma=0.1*(ub-lb); % Mutation Step Size
%% Initialization NSGA-II
empty_individual.Position=[];
empty_individual.Cost=[];
empty_individual.Rank=[];
empty_individual.DominationSet=[];
empty_individual.DominatedCount=[];
empty_individual.CrowdingDistance=[];
pop=repmat(empty_individual,nPop,1);
%%
for i=1:nPop
    pop(i).Position=inittf1;
    pop(i).Position=lchecktf1(pop(i).Position);
    pop(i).Cost=CostFunction(pop(i).Position);
end
[pop, F]=NonDominatedSorting(pop);% Non-Dominated Sorting
pop=CalcCrowdingDistance(pop,F);% Calculate Crowding Distance
[pop, F]=SortPopulation(pop);% Sort Population
%%
for it=1:MaxIt
    % Crossover
    popc=repmat(empty_individual,round(nCrossover/2),2);
    for k=1:round(nCrossover/2)
        i1=randi([1 nPop/2]);
        p1=pop(i1);
        i2=randi([1 nPop/2]);
        p2=pop(i2);
        [popc(k,1).Position, popc(k,2).Position]=Crossover(p1.Position,p2.Position);
        popc(k,1).Position=lchecktf1(popc(k,1).Position);
        popc(k,2).Position=lchecktf1(popc(k,2).Position);
        popc(k,1).Cost=CostFunction(popc(k,1).Position);
        popc(k,2).Cost=CostFunction(popc(k,2).Position);
    end
    popc=popc(:);

```

```

% Mutation
popm= repmat(empty_individual,nMutation,1);
for k=1:nMutation
    i=randi([1 nPop/2]);
    p=pop(i);
    popm(k).Position=Mutate(p.Position,mug,sigma);
    popm(k).Position=lchecktf1(popm(k).Position);
    popm(k).Cost=CostFunction(popm(k).Position);
end
% Merge
pop=[pop
    popc
    popm];
% Non-Dominated Sorting
[pop, F]=NonDominatedSorting(pop);
% Calculate Crowding Distance
pop=CalcCrowdingDistance(pop,F);
% Sort Population
pop=SortPopulation(pop);
% Truncate
pop=pop(1:nPop/2);
% Non-Dominated Sorting
[pop, F]=NonDominatedSorting(pop);
% Calculate Crowding Distance
pop=CalcCrowdingDistance(pop,F);
% Sort Population
[pop, F]=SortPopulation(pop);
% Store F1
F1=pop(F{1});
%Rearrange in the final loop
if it==MaxIt
    uniqueCosts=zeros(0, 2);
    uniqueF1=[];
    for i=1: numel(F1)
        cost=F1(i).Cost;
        if ~ismember(cost, uniqueCosts, 'rows')
            uniqueCosts=[uniqueCosts; cost];
            uniqueF1=[uniqueF1, F1(i)];
        end
    end
    F1 = uniqueF1;
    % Sort F1 in ascending order of f2
    bValues = arrayfun(@(x) x.Cost(2), F1);
    [~, sortedIdx] = sort(bValues);
    F1 = F1(sortedIdx);
end
disp(['Iteration ' num2str(it) ': Number of F1 Members = ' num2str(numel(F1))]);
% Plot F1 Costs
figure(1)
PlotCosts(F1,it,MaxIt);
title("Pareto")
xlabel("Distance");
ylabel("Safety (%)");
PlotSolution(F1);
pause(0.01);
end
data1={};

```

```

data1{1, 1}='F1';
data1{2, 1}='Distance';
data1{3, 1}='Safety(%)';
data2={};
data2{1, 1}='F1';
startNum=nD + nG + nT + 1;
endNum=nD + nG + nT + nA;
data2(1, 3:(3 + (endNum - startNum))) = num2cell(startNum:endNum);
row=2;
for i=1: numel(F1)
    data1{1,1+i}=i;
    data1{2,1+i}=F1(i).Cost(1);
    data1{3,1+i}=F1(i).Cost(2);
    data2{row,1}=i;
    data2{row,2}='x';
    data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(1:nA), 2));
    row=row+1;
    data2{row,2}='y';
    data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(nA+1:nA*2), 2));
    row=row+1;
    data2{row,2}='r';
    data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(nA*2+1:nA*3), 0));
    row=row+1;
end
filename = 'output.xlsx';
if isfile(filename)
    delete(filename);
end
writecell(data1, filename, 'Sheet', 'Sheet1');
writecell(data2, filename, 'Sheet', 'Sheet2');
fprintf('Data has been exported to the file:', filename);

```

2. Phần chương trình chính thuật toán NSGA-II lai MOPSO

```

clc;
clear;
close all;
global lb ub
global nD nG nT nA
%% Problem Definition
CreateModel;
CostFunction=@(x)tf1(x);    % Cost Function
nObj=2;% Number of Objective Functions
%% NSGA-II Parameters
MaxIt=300;    % Maximum Number of Iterations
nPop=200;    % Population Size
pCrossover=0.7;    % Crossover Percentage
nCrossover=2*round(pCrossover*nPop/2); % Number of Parnets (Offsprings)
nCrossover=round(nCrossover/2);
pMutation=0.4;    % Mutation Percentage
nMutation=round(pMutation*nPop);    % Number of Mutants
nMutation=round(nMutation/2);
mug=0.01;    % Mutation Rate
sigma=0.1*(ub-lb); % Mutation Step Size
%% MOPSO Parameters
nRep=100;    % Repository Size
w=0.9;    % Inertia Weight
wdamp=0.99;    % Intertia Weight Damping Rate\
c1=10;    % Personal Learning Coefficient

```

```

c2=1;          % Global Learning Coefficient
nGrid=10;
alpha=0.2;     % Inflation Rate
beta=0.5;
gamma=2;       % Deletion Selection Pressure
mu=0.2;        % Mutation Rate
%% Initialization NSGA-II
empty_individual.Position=[];
empty_individual.Cost=[];
empty_individual.Rank=[];
empty_individual.DominationSet=[];
empty_individual.DominatedCount=[];
empty_individual.CrowdingDistance=[];
pop=repmat(empty_individual,nPop,1);
%% Initialization MOPSO
empty_particle.Position=[];
empty_particle.Velocity=[];
empty_particle.Cost=[];
empty_particle.Best.Position=[];
empty_particle.Best.Cost=[];
empty_particle.IsDominated=[];
empty_particle.GridIndex=[];
empty_particle.GridSubIndex=[];
popul=repmat(empty_particle,nPop/2,1);
%%
for i=1:nPop
    pop(i).Position=inittf1;
    pop(i).Position=lchecktf1(pop(i).Position);
    pop(i).Cost=CostFunction(pop(i).Position);
end
[pop, F]=NonDominatedSorting(pop);% Non-Dominated Sorting
pop=CalcCrowdingDistance(pop,F);% Calculate Crowding Distance
[pop, F]=SortPopulation(pop);% Sort Population
% Truncate and divide into two halves
popul1=pop((nPop/2)+1:nPop);
pop=pop(1:nPop/2);
%%
for it=1:MaxIt
    % Crossover
    popc=repmat(empty_individual,round(nCrossover/2),2);
    for k=1:round(nCrossover/2)
        i1=randi([1 nPop/2]);
        p1=pop(i1);
        i2=randi([1 nPop/2]);
        p2=pop(i2);
        [popc(k,1).Position, popc(k,2).Position]=Crossover(p1.Position,p2.Position);
        popc(k,1).Position=lchecktf1(popc(k,1).Position);
        popc(k,2).Position=lchecktf1(popc(k,2).Position);
        popc(k,1).Cost=CostFunction(popc(k,1).Position);
        popc(k,2).Cost=CostFunction(popc(k,2).Position);
    end
    popc=popc(:);
    % Mutation
    popm=repmat(empty_individual,nMutation,1);
    for k=1:nMutation
        i=randi([1 nPop/2]);
        p=pop(i);

```

```

    popm(k).Position=Mutate(p.Position,mug,sigma);
    popm(k).Position=lchecktf1(popm(k).Position);
    popm(k).Cost=CostFunction(popm(k).Position);
end
% Merge
pop=[pop
    popc
    popm];
% Non-Dominated Sorting
[pop, F]=NonDominatedSorting(pop);
% Calculate Crowding Distance
pop=CalcCrowdingDistance(pop,F);
% Sort Population
pop=SortPopulation(pop);
% Truncate
pop=pop(1:nPop/2);
% Non-Dominated Sorting
[pop, F]=NonDominatedSorting(pop);
% Calculate Crowding Distance
pop=CalcCrowdingDistance(pop,F);
% Sort Population
[pop, F]=SortPopulation(pop);
% Store F1
F1=pop(F{1});
%% MOPSO
for i=1:nPop/2
    popul(i).Position=popul1(i).Position;
    popul(i).Velocity=zeros([1 nA*3]);
    popul(i).Cost=popul1(i).Cost;
    % Update Personal Best
    popul(i).Best.Position=popul(i).Position;
    popul(i).Best.Cost=popul(i).Cost;
end
% Determine Domination
popul=DetermineDomination(popul);
rep=popul(~[popul.IsDominated]);
Grid=CreateGrid(rep,nGrid,alpha);
for i=1: numel(rep)
    rep(i)=FindGridIndex(rep(i),Grid);
end
%% MOPSO Main Loop
for i=1:nPop/2
    leader=SelectLeader(rep,beta);
    popul(i).Velocity = w*popul(i).Velocity ...
        +c1*rand([1 nA*3]).*(popul(i).Best.Position-popul(i).Position) ...
        +c2*rand([1 nA*3]).*(leader.Position-popul(i).Position);
    for j=1:nA*3
        popul(i).Position(j) = popul(i).Position(j) + popul(i).Velocity(j);
    end
    popul(i).Position=lchecktf1(popul(i).Position);
    popul(i).Cost = CostFunction(popul(i).Position);
    % Apply Mutation
    pm=(1-(it-1)/(MaxIt-1))^(1/mu);
    if rand<pm
        NewSol.Position=Mutatef1(popul(i).Position,pm,lb,ub);
        NewSol.Position=lchecktf1(NewSol.Position);
        NewSol.Cost=CostFunction(NewSol.Position);
    end
end

```

```

    if Dominates(NewSol,popul(i))
        popul(i).Position=NewSol.Position;
        popul(i).Cost=NewSol.Cost;
    elseif Dominates(popul(i),NewSol)
        % Do Nothing
    else
        if rand<0.5
            popul(i).Position=NewSol.Position;
            popul(i).Cost=NewSol.Cost;
        end
    end
end
%Assign the best value to the personal coefficient
if Dominates(popul(i),popul(i).Best)
    popul(i).Best.Position=popul(i).Position;
    popul(i).Best.Cost=popul(i).Cost;
elseif Dominates(popul(i).Best,popul(i))
    % Do Nothing
else
    if rand<0.5
        popul(i).Best.Position=popul(i).Position;
        popul(i).Best.Cost=popul(i).Cost;
    end
end
end
% Add Non-Dominated Particles to REPOSITORY
rep=[rep
popul(~[popul.IsDominated])];
% Determine Domination of New Respository Members
rep=DetermineDomination(rep);
% Keep only Non-Dminated Memebtrs in the Repository
rep=rep(~[rep.IsDominated]);
% Update Grid Indices
for i=1: numel(rep)
    rep(i)=FindGridIndex(rep(i),Grid);
end
% Check if Repository is Full
%   numel(rep)
if numel(rep)>nRep
    Extra=numel(rep)-nRep;
    for e=1:Extra
        rep=DeleteOneRepMemebr(rep,gamma);
    end
end
% Damping Inertia Weight
w=w*wdamp;
pop1=repmat(empty_individual,nPop/2,1);
for i=1:nPop/2
    if i<=length(rep)
        pop1(i).Position=rep(i).Position;
        pop1(i).Cost=rep(i).Cost;
    else
        pop1(i).Position=popul(i-length(rep)).Position;
        pop1(i).Cost=popul(i-length(rep)).Cost;
    end
end
end
pop2=[pop

```

```

    pop1];
% Non-Dominated Sorting
[pop2, F]=NonDominatedSorting(pop2);
% Calculate Crowding Distance
pop2=CalcCrowdingDistance(pop2,F);
% Sort Population
[pop2, F]=SortPopulation(pop2);
% Truncate and divide into two halves
pop=pop2(1:nPop/2);
%Rearrange in the final loop
if it==MaxIt
    uniqueCosts=zeros(0, 2);
    uniqueF1=[];
    for i=1: numel(F1)
        cost=F1(i).Cost';
        if ~ismember(cost, uniqueCosts, 'rows')
            uniqueCosts=[uniqueCosts; cost];
            uniqueF1=[uniqueF1, F1(i)];
        end
    end
    F1 = uniqueF1;
    % Sort F1 in ascending order of f2
    bValues = arrayfun(@(x) x.Cost(2), F1);
    [~, sortedIdx] = sort(bValues);
    F1 = F1(sortedIdx);
end
disp(['Iteration ' num2str(it) ': Number of F1 Members = ' num2str(numel(F1))]);
% Plot F1 Costs
figure(1)
PlotCosts(F1,it,MaxIt);
title("Pareto")
xlabel("Distance");
ylabel("Safety (%)");
PlotSolution(F1);
pause(0.01);
end
data1={};
data1{1, 1}='F1';
data1{2, 1}='Distance';
data1{3, 1}='Safety(%)';
data2={};
data2{1, 1}='F1';
startNum=nD + nG + nT + 1;
endNum=nD + nG + nT + nA;
data2(1, 3:(3 + (endNum - startNum))) = num2cell(startNum:endNum);
row=2;
for i=1: numel(F1)
    data1{1,1+i}=i;
    data1{2,1+i}=F1(i).Cost(1);
    data1{3,1+i}=F1(i).Cost(2);
    data2{row,1}=i;
    data2{row,2}='x';
    data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(1:nA), 2));
    row=row+1;
    data2{row,2}='y';
    data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(nA+1:nA*2), 2));
    row=row+1;
end

```

```

data2{row,2}='r';
data2(row, 3:(2+nA)) = num2cell(round(F1(i).Position(nA*2+1:nA*3), 0));
row=row+1;
end
filename = 'output.xlsx';
if isfile(filename)
    delete(filename);
end
writecell(data1, filename, 'Sheet', 'Sheet1');
writecell(data2, filename, 'Sheet', 'Sheet2');
fprintf('Data has been exported to the file:', filename);

```

3. Các hàm chương trình con function

3.1. Hàm tính khoảng cách đồng đúc

```

function pop=CalcCrowdingDistance(pop,F)
nF=numel(F);
for k=1:nF
    Costs=[pop(F{k}).Cost];
    nObj=size(Costs,1);
    n=numel(F{k});
    d=zeros(n,nObj);
    for j=1:nObj
        [cj, so]=sort(Costs(j,:));
        d(so(1),j)=inf;
        for i=2:n-1
            d(so(i),j)=abs(cj(i+1)-cj(i-1))/abs(cj(1)-cj(end));
        end
        d(so(end),j)=inf;
    end
    for i=1:n
        pop(F{k}(i)).CrowdingDistance=sum(d(i,:));
    end
end
end

```

3.2. Hàm tạo lưới phân chia không gian mục tiêu

```

function Grid=CreateGrid(pop,nGrid,alpha)
c=[pop.Cost];
cmin=min(c,[],2);
cmax=max(c,[],2);
dc=cmax-cmin;
cmin=cmin-alpha*dc;
cmax=cmax+alpha*dc;
nObj=size(c,1);
empty_grid.LB=[];
empty_grid.UB=[];
Grid= repmat(empty_grid,nObj,1);
for j=1:nObj
    cj=linspace(cmin(j),cmax(j),nGrid+1);
    Grid(j).LB=[-inf cj];
    Grid(j).UB=[cj +inf];
end
end

```

3.3. Hàm nhập dữ liệu mô hình bài toán

```

function CreateModel()
global xC yC
global xD yD wD hD dD nD
global xG yG wG hG dG nG

```

```

global xT yT wT hT dT rT zT nT
global wA hA dA nA sen Tn
global a hcw edges
global lb ub
%% Initialize the initial fixed items
% Coordinates of the site for the construction master plan
xC=[0 254 254 0];
yC=[0 0 100 100];
% Items according to the design
xD=[17 60 103 204 204 239];
yD=[36 74 36 80 20 50];
wD=[24 26 24 33 33 19];
hD=[45 34 45 19 19 72];
dD=[3 3 3 3 3 3];
% Gate
xG=[140];
yG=[100];
wG=[7];
hG=[40*2];
dG=[1];
% Tower crane
xT=[60 217];
yT=[44 50];
wT=[6 6];
hT=[6 6];
dT=[4 4];
rT=[65 50];
zT=[40 40];
% Mobile item
wA=[4 5 5 5 6 6 5 3 5 5 5 5 12 12 5 5 2 6 5 6];
hA=[4 5 5 5 6 5 5 5 5 5 5 5 12 5 5 2 12 10 3];
dA=[1 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2];
sen=[25 25 25 25 50 50 50 0 0 0 0 0 0 25 0 25 25 0 25];
Tn=[0 1 1 1 0 0 0 2 0 0 0 2 2 2 2 0 0 0 0];
%Proximity weight
a=[
0 0 0 0 0 0 0 0 0 0 1 6^4 6^4 6^4 6^2 1 6^3 1 1 1 1 1 1 1 1 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 1 6^4 6^4 6^4 6^2 1 6^3 1 1 1 1 1 1 1 1 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 1 6^4 6^4 6^4 6^2 1 6^3 1 1 1 1 1 1 1 1 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 1 1 1 1 6^2 1 6^3 6^3 1 1 1 6^3 6^3 6^3 6^3 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 1 1 1 1 6^2 1 6^3 6^3 1 1 1 6^3 6^3 6^3 6^3 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 1 1 1 1 6^2 1 6^3 6^3 1 1 1 6^3 6^3 6^3 6^3 6^2 6^2 6^2 6^2
0 0 0 0 0 0 0 0 0 0 6^4 6^1 6^1 6^1 6^1 6^1 6^1 6^2 6^2 6^2 6^2 6^2 1 6^2 1 1 6^1 1 6^3
0 0 0 0 0 0 0 0 0 0 1 6^2 6^2 6^2 1 1 6^4 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 6^4 6^2 1 1 1 6^2 6^2 6^2 6^2 1 1 1 1
1 1 1 1 1 1 6^4 1 1 0 1 1 1 6^2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6^2 1 1
6^4 6^4 6^4 1 1 1 6^1 6^2 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
6^4 6^4 6^4 1 1 1 6^1 6^2 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
6^4 6^4 6^4 1 1 1 6^1 6^2 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
6^2 6^2 6^2 6^2 6^2 6^2 6^2 6^1 1 1 6^2 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 6^1 1 1 1 1 1 1 1 0 6^2 1 1 1 1 1 1 1 1 1 1 1 1 1
6^3 6^3 6^3 6^3 6^3 6^3 6^1 6^4 6^4 1 1 1 1 1 6^2 0 1 1 1 1 1 6^2 1 6^2 1 1 1 1
1 1 1 6^3 6^3 6^3 6^2 1 6^2 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 6^2 1 1 1 1 1 1 1 1 0 1 1 1 1 1 6^3 1 1 1 1
1 1 1 1 1 1 6^2 1 1 1 1 1 1 1 1 1 0 1 1 1 1 6^3 1 1 1 1
1 1 1 1 1 1 6^2 1 1 1 1 1 1 1 1 1 1 0 1 1 1 6^3 1 1 1 1
1 1 1 6^3 6^3 6^3 6^2 1 6^2 1 1 1 1 1 1 1 1 1 1 0 6^4 1 1 1 1 1 1

```

[illegible]

```

0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
];
%%
nD=numel(wD);
nG=numel(wG);
nT=numel(wT);
nA=numel(wA);
%% Calculate the number of high-density routes
edges = [];
for i = 1:nA
    for j = i+1:nA
        if htr(i, j) == 1
            edges = [edges; i, j];
        end
    end
end
end
%% Preliminary limit
lb=[min(xC)*ones(1,nA) min(yC)*ones(1,nA) 0*ones(1,nA)];
ub=[max(xC)*ones(1,nA) max(yC)*ones(1,nA) 1*ones(1,nA)];
end

```

3.4. Hàm lai ghép

```

function [y1, y2]=Crossover(x1,x2)
    alpha=rand(size(x1));
    y1=alpha.*x1+(1-alpha).*x2;
    y2=alpha.*x2+(1-alpha).*x1;
end

```

3.5. Hàm xóa 1 phần tử khỏi kho lưu trữ

```

function rep=DeleteOneRepMemebr(rep,gamma)
    % Grid Index of All Repository Members
    GI=[rep.GridIndex];
    % Occupied Cells
    OC=unique(GI);
    % Number of Particles in Occupied Cells
    N=zeros(size(OC));
    for k=1:numel(OC)
        N(k)=numel(find(GI==OC(k)));
    end
    % Selection Probabilities
    P=exp(gamma*N);
    P=P/sum(P);
    % Selected Cell Index
    sci=RouletteWheelSelection(P);
    % Selected Cell
    sc=OC(sci);
    % Selected Cell Members
    SCM=find(GI==sc);

```

```

% Selected Member Index
smi=randi([1 numel(SCM)]);
% Selected Member
sm=SCM(smi);
% Delete Selected Member
rep(sm)=[];
end

```

3.6. Hàm xác định cá thể bị trội

```

function pop=DetermineDomination(pop)
nPop=numel(pop);
for i=1:nPop
    pop(i).IsDominated=false;
end
for i=1:nPop-1
    for j=i+1:nPop
        if Dominates(pop(i),pop(j))
            pop(j).IsDominated=true;
        end
        if Dominates(pop(j),pop(i))
            pop(i).IsDominated=true;
        end
    end
end
end
end

```

3.7. Hàm so sánh trội

```

function b=Dominates(x,y)
if isstruct(x)
    x=x.Cost;
end
if isstruct(y)
    y=y.Cost;
end
b = (x(1) <= y(1) && x(2) >= y(2)) && (x(1) < y(1) || x(2) > y(2)) && x(2)>0 && y(2)>0;
end

```

3.8. Hàm xác định chỉ số ô lưới

```

function particle=FindGridIndex(particle,Grid)
nObj=numel(particle.Cost);
nGrid=numel(Grid(1).LB);
particle.GridSubIndex=zeros(1,nObj);
for j=1:nObj
    particle.GridSubIndex(j)=...
        find(particle.Cost(j)<Grid(j).UB,1,'first');
end
particle.GridIndex=particle.GridSubIndex(1);
for j=2:nObj
    particle.GridIndex=particle.GridIndex-1;
    particle.GridIndex=nGrid*particle.GridIndex;
    particle.GridIndex=particle.GridIndex+particle.GridSubIndex(j);
end
end

```

3.9. Hàm khởi tạo vị trí ngẫu nhiên ban đầu

```

function [x]=inittf1
global xC yC
global wA hA nA
%% Random position
xA=zeros(1, nA);

```

```

yA=zeros(1, nA);
rA=rand(1, nA);
for i=1:nA
    xA(i) = min(xC) + wA(i)/2 + (max(xC) - min(xC)-wA(i)/2) * rand();
    yA(i) = min(yC) + hA(i)/2 + (max(yC) - min(yC)-hA(i)/2) * rand();
end
x=[xA yA rA];
end

```

3.10. Hàm kiểm tra và ràng buộc cá thể

```

function [x]=lchecktf1(x)
    global xC yC
    global xD yD wD hD dD nD
    global xG yG wG hG dG nG
    global xT yT wT hT dT nT rT
    global wA hA dA nA Tn
    xA=x(1:nA);
    yA=x(nA+1:nA*2);
    rA=x(nA*2+1:nA*3);
    theta=round(rA)*pi/2;
    w=cos(theta).*wA+sin(theta).*hA;
    h=cos(theta).*hA+sin(theta).*wA;
    ns=nD+nG+nT+nA;
    xs=[xD xG xT xA];
    ys=[yD yG yT yA];
    ws=[wD wG wT w];
    hs=[hD hG hT h];
    ds=[dD dG dT dA];
    d=zeros(ns,ns);
    %Non-overlapping constraint
    for i=nD+nG+nT+1:ns
        for j=1:ns
            if i~=j
                DELTA=max(ds(i),ds(j));
                XVij=max(0,1-abs(xs(i)-xs(j))/((ws(i)+ws(j))/2+DELTA));
                YVij=max(0,1-abs(ys(i)-ys(j))/((hs(i)+hs(j))/2+DELTA));
                if XVij>0 && YVij>0
                    if xs(i)>=xs(j) && ys(i)>=ys(j)
                        dx=xs(j)+(ws(i)+ws(j))/2+DELTA-xs(i);
                        dy=ys(j)+(hs(i)+hs(j))/2+DELTA-ys(i);
                        if dx>=dy
                            ys(i)=ys(i)+dy;
                        elseif dx<dy
                            xs(i)=xs(i)+dx;
                        end
                    elseif xs(i)>=xs(j) && ys(i)<ys(j)
                        dx=xs(j)+(ws(i)+ws(j))/2+DELTA-xs(i);
                        dy=ys(i)-(ys(j)-(hs(i)+hs(j))/2-DELTA);
                        if dx>=dy
                            ys(i)=ys(i)-dy;
                        elseif dx<dy
                            xs(i)=xs(i)+dx;
                        end
                    elseif xs(i)<xs(j) && ys(i)>=ys(j)
                        dx=xs(i)-(xs(j)-(ws(i)+ws(j))/2-DELTA);
                        dy=ys(j)+(hs(i)+hs(j))/2+DELTA-ys(i);
                        if dx>=dy
                            ys(i)=ys(i)+dy;

```

```

elseif dx<dy
    xs(i)=xs(i)-dx;
end
elseif xs(i)<xs(j) && ys(i)<ys(j)
    dx=xs(i)-(xs(j)-(ws(i)+ws(j))/2-DELTA);
    dy=ys(i)-(ys(j)-(hs(i)+hs(j))/2-DELTA);
    if dx>=dy
        ys(i)=ys(i)-dy;
    elseif dx<dy
        xs(i)=xs(i)-dx;
    end
end
end
end
end
end
xA=xs(nD+nG+nT+1:end);
yA=ys(nD+nG+nT+1:end);
%% Constraint within the tower crane radius
for i=1:nT
    for j=1:nA
        if Tn(j)==i
            if xA(j)<xT(i) && yA(j)<yT(i)
                dd=norm([xT(i)-(xA(j)-w(j)/2) yT(i)-(yA(j)-h(j)/2)]);
                if dd>rT(i)
                    dx=(xT(i)-(xA(j)-w(j)/2))*(dd-rT(i))/dd;
                    dy=(yT(i)-(yA(j)-h(j)/2))*(dd-rT(i))/dd;
                    xA(j)=xA(j)+dx;
                    yA(j)=yA(j)+dy;
                end
            elseif xA(j)>=xT(i) && yA(j)<yT(i)
                dd=norm([xT(i)-(xA(j)+w(j)/2) yT(i)-(yA(j)-h(j)/2)]);
                if dd>rT(i)
                    dx=(-xT(i)+(xA(j)+w(j)/2))*(dd-rT(i))/dd;
                    dy=(yT(i)-(yA(j)-h(j)/2))*(dd-rT(i))/dd;
                    xA(j)=xA(j)-dx;
                    yA(j)=yA(j)+dy;
                end
            elseif xA(j)<xT(i) && yA(j)>=yT(i)
                dd=norm([xT(i)-(xA(j)-w(j)/2) yT(i)-(yA(j)+h(j)/2)]);
                if dd>rT(i)
                    dx=(xT(i)-(xA(j)-w(j)/2))*(dd-rT(i))/dd;
                    dy=(-yT(i)+(yA(j)+h(j)/2))*(dd-rT(i))/dd;
                    xA(j)=xA(j)+dx;
                    yA(j)=yA(j)-dy;
                end
            elseif xA(j)>=xT(i) && yA(j)>=yT(i)
                dd=norm([xT(i)-(xA(j)+w(j)/2) yT(i)-(yA(j)+h(j)/2)]);
                if dd>rT(i)
                    dx=(-xT(i)+(xA(j)+w(j)/2))*(dd-rT(i))/dd;
                    dy=(-yT(i)+(yA(j)+h(j)/2))*(dd-rT(i))/dd;
                    xA(j)=xA(j)-dx;
                    yA(j)=yA(j)-dy;
                end
            end
        end
    end
end
end
end

```

```

end
%% Constraint within the site area
for i=1:nA
    xA(i)=min(max(xC)-w(i),xA(i));
    xA(i)=max(min(xC)+w(i),xA(i));
    yA(i)=min(max(yC)-h(i),yA(i));
    yA(i)=max(min(yC)+h(i),yA(i));
end
x=[xA yA rA];
end

```

3.11. Hàm đột biến nhiều giá trị

```

function y=Mutate(x,mu,sigma)
nVar=numel(x);
nMu=ceil(mu*nVar);
j=randsample(nVar,nMu);
j=randsample(nVar,1);
if numel(sigma)>1
    sigma = sigma(j);
end
y=x;
y(j)=x(j)+sigma.*randn(size(j));
end

```

3.12. Hàm đột biến một giá trị

```

function xnew=Mutatef1(x,pm,VarMin,VarMax)
nVar=numel(x);
j=randi([1 nVar]);
dx=pm*(VarMax-VarMin);
lb=x(j)-dx;
if lb<VarMin
    lb=VarMin;
end
ub=x(j)+dx;
if ub>VarMax
    ub=VarMax;
end
xnew=x;
xnew(j)=lb(j)-((lb(j)-ub(j))*rand);
end

```

3.13. Hàm sắp xếp không trội

```

function [pop, F]=NonDominatedSorting(pop)
nPop=numel(pop);
for i=1:nPop
    pop(i).DominationSet=[];
    pop(i).DominatedCount=0;
end
F{1}=[];
for i=1:nPop
    for j=i+1:nPop
        p=pop(i);
        q=pop(j);
        if Dominates(p,q)
            p.DominationSet=[p.DominationSet j];
            q.DominatedCount=q.DominatedCount+1;
        end
        if Dominates(q,p,q.Cost)
            q.DominationSet=[q.DominationSet i];
        end
    end
end

```

```

        p.DominatedCount=p.DominatedCount+1;
    end
    pop(i)=p;
    pop(j)=q;
end
if pop(i).DominatedCount==0
    F{1}=[F{1} i];
    pop(i).Rank=1;
end
end
k=1;
while true
    Q=[];
    for i=F{k}
        p=pop(i);
        for j=p.DominationSet
            q=pop(j);
            q.DominatedCount=q.DominatedCount-1;
            if q.DominatedCount==0
                Q=[Q j];
                q.Rank=k+1;
            end
            pop(j)=q;
        end
    end
    if isempty(Q)
        break;
    end
    F{k+1}=Q;
    k=k+1;
end
end

```

3.14. Hàm vẽ các điểm Pareto

```

function PlotCosts(pop,it,MaxIt)
    Costs=[pop.Cost];
    plot(Costs(1,:),Costs(2,:), 'r.', 'MarkerSize',8);
    if it==MaxIt
        numLabels = arrayfun(@num2str, 1:numel(pop), 'UniformOutput', false);
        text(Costs(1,.)+5000, Costs(2,:), numLabels, 'FontSize', 10, 'Color', 'b');
    end
    grid on;
end

```

3.15. Hàm vẽ bố trí mặt bằng thi công

```

function PlotSolution(F1)
    global xC yC
    global xD yD wD hD nD
    global xG yG wG nG
    global xT yT wT hT rT zT nT
    global wA hA nA
    if numel(F1)>1
        figure(5);
        idx_min_a = 1;
        idx_max_b = 1;
        min_a = F1(1).Cost(1);
        max_b = F1(1).Cost(2);
        for i = 2:numel(F1)

```

```

    if F1(i).Cost(1) < min_a
        min_a = F1(i).Cost(1);
        idx_min_a = i;
    end
    if F1(i).Cost(2) > max_b
        max_b = F1(i).Cost(2);
        idx_max_b = i;
    end
end
F1s(1) = F1(idx_min_a);
F1s(2) = F1(idx_max_b);
% Draw the item according to the design
for a=1:2
    subplot(2,1,a)
    Colorc=[1 0.5 0.5];
    for i=1:nD
        XD=[xD(i)-wD(i)/2 xD(i)+wD(i)/2 xD(i)+wD(i)/2 xD(i)-wD(i)/2];
        YD=[yD(i)-hD(i)/2 yD(i)-hD(i)/2 yD(i)+hD(i)/2 yD(i)+hD(i)/2];
        fill(XD,YD,Colorc);
        hold on;
        text(xD(i),yD(i),num2str(i),...
            'FontSize',10,...
            'FontWeight','bold',...
            'HorizontalAlignment','center',...
            'VerticalAlignment','middle');
    end
    % Draw gate
    for i=1:nG
        plot([xG(i)-(wG(i)/2) xG(i)+(wG(i)/2)], [yG(i) yG(i)], 'r-', 'LineWidth', 4);
        hold on;
        text(xG(i),yG(i),num2str(nD+i),...
            'FontSize',10,...
            'FontWeight','bold',...
            'HorizontalAlignment','center',...
            'VerticalAlignment','bottom');
    end
    % Draw the tower crane
    Colort=[1 0.5 0];
    for i=1:nT
        XT=[xT(i)-wT(i)/2 xT(i)+wT(i)/2 xT(i)+wT(i)/2 xT(i)-wT(i)/2];
        YT=[yT(i)-hT(i)/2 yT(i)-hT(i)/2 yT(i)+hT(i)/2 yT(i)+hT(i)/2];
        fill(XT,YT,Colort);
        hold on;
        text(xT(i),yT(i),num2str(nD+nG+i),...
            'FontSize',10,...
            'FontWeight','bold',...
            'HorizontalAlignment','center',...
            'VerticalAlignment','middle');
        rectangle('Position', [xT(i) - rT(i), yT(i) - rT(i), 2*rT(i), 2*rT(i)], ...
            'Curvature', [1, 1], ...
            'EdgeColor', Colort, ...
            'LineWidth', 0.6);
        rectangle('Position', [xT(i) - sqrt(rT(i)^2+zT(i)^2), yT(i) - sqrt(rT(i)^2+zT(i)^2), 2*sqrt(rT(i)^2+zT(i)^2),
            2*sqrt(rT(i)^2+zT(i)^2)], ...
            'Curvature', [1, 1], ...
            'EdgeColor', Colort, ...
            'LineWidth', 0.3);
    end
end

```

```

end
% Draw the mobile item
ColorA=[1 1 0];
xA=zeros(1, nA);
yA=zeros(1, nA);
rA=zeros(1, nA);
for i=1:nA
    xA=F1s(a).Position(1:nA);
    yA=F1s(a).Position(nA+1:nA*2);
    rA=F1s(a).Position(nA*2+1:nA*3);
    theta=round(rA)*pi/2;
    w=cos(theta).*wA+sin(theta).*hA;
    h=cos(theta).*hA+sin(theta).*wA;
    XA=[xA(i)-w(i)/2 xA(i)+w(i)/2 xA(i)+w(i)/2 xA(i)-w(i)/2];
    YA=[yA(i)-h(i)/2 yA(i)-h(i)/2 yA(i)+h(i)/2 yA(i)+h(i)/2];
    fill(XA,YA,ColorA);
    hold on;
    text(xA(i),yA(i),num2str(nD+nG+nT+i),...
        'FontSize',7,...
        'FontWeight','bold',...
        'HorizontalAlignment','center',...
        'VerticalAlignment','middle');
end
axis equal;
grid on;
xlim([-10 max(xC)+10]);
ylim([-10 max(yC)+10]);
plot([xC, xC(1)], [yC, yC(1)], 'k-', 'LineWidth', 2);
if a==1
    title(sprintf('MinDistance - Distance: %d; Safety: %d', F1s(a).Cost(1), F1s(a).Cost(2)));
elseif a==2
    title(sprintf('MaxSafety - Distance: %d; Safety: %d', F1s(a).Cost(1), F1s(a).Cost(2)));
end
hold off;
end
end
end

```

3.16. Hàm chọn lọc bánh xe roulette

```

function i=RouletteWheelSelection(P)
    r=rand;
    C=cumsum(P);
    i=find(r<=C,1,'first');
end

```

3.17. Hàm lựa chọn cá thể lãnh đạo

```

function leader=SelectLeader(rep,beta)
    % Grid Index of All Repository Members
    GI=[rep.GridIndex];
    % Occupied Cells
    OC=unique(GI);
    % Number of Particles in Occupied Cells
    N=zeros(size(OC));
    for k=1:numel(OC)
        N(k)=numel(find(GI==OC(k)));
    end
    % Selection Probabilities
    P=exp(-beta*N);

```

```

P=P/sum(P);
% Selected Cell Index
sci=RouletteWheelSelection(P);
% Selected Cell
sc=OC(sci);
% Selected Cell Members
SCM=find(GI==sc);
% Selected Member Index
smi=randi([1 numel(SCM)]);
% Selected Member
sm=SCM(smi);
% Leader
leader=rep(sm);
end

```

3.18. Hàm sắp xếp quần thể dựa vào khoảng cách và cấp bậc

```

function [pop, F]=SortPopulation(pop)
% Sort Based on Crowding Distance
[~, CDSO]=sort([pop.CrowdingDistance], 'descend');
pop=pop(CDSO);
% Sort Based on Rank
 [~, RSO]=sort([pop.Rank]);
pop=pop(RSO);
% Update Fronts
Ranks=[pop.Rank];
MaxRank=max(Ranks);
F=cell(MaxRank,1);
for r=1:MaxRank
    F{r}=find(Ranks==r);
end
end

```

3.19. Hàm tính giá trị hàm mục tiêu

```

function z=tf1(x)
global xD yD wD hD dD nD
global xG yG wG hG dG nG
global xT yT wT hT dT rT zT nT Tn
global wA hA dA nA sen
global a hcw edges
xA=x(1:nA);
yA=x(nA+1:nA*2);
rA=x(nA*2+1:nA*3);
theta=round(rA)*pi/2;
w=cos(theta).*wA+sin(theta).*hA;
h=cos(theta).*hA+sin(theta).*wA;
%% Distance objective function
ns=nD+nG+nT+nA;
xs=[xD xG xT xA];
ys=[yD yG yT yA];
ws=[wD wG wT w];
hs=[hD hG hT h];
ds=[dD dG dT dA];
d=zeros(ns,ns);
for i=1:ns-1
    for j=i+1:ns
        d(i,j)=norm([xs(i)-xs(j) ys(i)-ys(j)]);
        d(j,i)=d(i,j);
        DELTA=max(ds(i),ds(j));
    end
end

```

```

XVij=max(0,1-abs(xs(i)-xs(j))/((ws(i)+ws(j))/2+DELTA));
YVij=max(0,1-abs(ys(i)-ys(j))/((hs(i)+hs(j))/2+DELTA));
V(i,j)=min(XVij,YVij);
V(j,i)=V(i,j);
end
end
v=mean(V(:));
alpha=1e13; %av=alpha*v;
ad=a.*d;
SumAD=sum(ad(:));
f1=round(SumAD+alpha*v);
%% Safety objective function
%Tower crane safety
cs=zeros(nT,nA);
for i=1:nT
    for j=1:nA
        if Tn(j)==i
            cs(i,j)=0;
        else
            RT=norm([xT(i)-(xA(j)+w(j))/2 yT(i)-(yA(j)+h(j)/2)]);
            RB=norm([xT(i)-(xA(j)+w(j))/2 yT(i)-(yA(j)-h(j)/2)]);
            LT=norm([xT(i)-(xA(j)-w(j)/2 yT(i)-(yA(j)+h(j)/2)]);
            LB=norm([xT(i)-(xA(j)-w(j)/2 yT(i)-(yA(j)-h(j)/2)]);
            dd = min([RT RB LT LB]);
            if dd<=rT(i)
                cs(i,j)=sen(j);
            elseif dd>rT(i) && dd<=hypot(rT(i),zT(i))
                cs(i,j)=(100-sen(j))*(d(nD+nG+i,nD+nG+nT+j)-rT(i))/zT(i)+sen(j);
            else
                cs(i,j)=100;
            end
        end
    end
end
end
rScs=sum(cs, 2)/nA;
CSC=mean(rScs);
%Safety from hazardous items
hcc=sum(sum(hcw.*d));
persistent hccmax hccmin;
if isempty(hccmax)
    hccmax = hcc;
else
    hccmax = max(hccmax, hcc);
end
if isempty(hccmin)
    hccmin = hcc;
else
    hccmin = min(hccmin, hcc);
end
if hccmax==hccmin
    nhcc=0;
else
    nhcc=((hcc-hccmin)/(hccmax-hccmin))*100;
end
%Safety at pathway intersections
IP = 0;
nedges = size(edges, 1);

```

```

for i = 1:nedges
    for j = i+1:nedges
        x1 = xA(edges(i, 1));
        y1 = yA(edges(i, 1));
        x2 = xA(edges(i, 2));
        y2 = yA(edges(i, 2));
        x3 = xA(edges(j, 1));
        y3 = yA(edges(j, 1));
        x4 = xA(edges(j, 2));
        y4 = yA(edges(j, 2));
        d1 = (x3 - x1) * (y2 - y1) - (y3 - y1) * (x2 - x1);
        d2 = (x4 - x1) * (y2 - y1) - (y4 - y1) * (x2 - x1);
        d3 = (x1 - x3) * (y4 - y3) - (y1 - y3) * (x4 - x3);
        d4 = (x2 - x3) * (y4 - y3) - (y2 - y3) * (x4 - x3);
        if (d1 * d2 < 0) && (d3 * d4 < 0)
            IP = IP + 1;
        end
    end
end
IPmax=nedges*(nedges-1)/2;
IPC=(1-IP/IPmax)*100;
if v==0
    f2=round(0.6*CSC+0.2*nhcc+0.2*IPC);
else
    f2=v*(0.6*CSC+0.2*nhcc+0.2*IPC);
end
z=[f1; f2];
end

```

PHỤ LỤC 02

1. Đánh giá của chỉ huy trưởng Lưu An Long, làm việc tại Công ty TNHH Xây dựng và Đầu tư

Bảng 1. Đánh giá trọng số gần gũi của Lưu An Long

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	1	1	1	1	$\frac{6}{2}$	$\frac{6}{5}$	1	1	1	$\frac{6}{3}$	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{3}$	$\frac{6}{2}$	1	$\frac{6}{5}$	1	1	1	$\frac{6}{3}$	1	1	1
3	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{3}$	$\frac{6}{2}$	1	1	$\frac{6}{5}$	1	$\frac{6}{3}$	1	1	1	1
4	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{3}$	$\frac{6}{2}$	1	1	1	$\frac{6}{5}$	1	$\frac{6}{3}$	1	1	1
5	0	0	0	0	0	0	0	0	$\frac{6}{5}$	1	$\frac{6}{5}$	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{2}$	$\frac{6}{2}$	1	1	1
6	0	0	0	0	0	0	0	0	1	$\frac{6}{5}$	1	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$
7	0	0	0	0	0	0	0	0	1	1	1	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1
9	1	1	1	1	$\frac{6}{5}$	1	1	1	0	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	$\frac{6}{5}$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	$\frac{6}{3}$	1
11	1	1	1	1	$\frac{6}{5}$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
13	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	$\frac{6}{3}$	1	1	0	1	1	1	1	1	1	1	1	1	1
14	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{2}$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	$\frac{6}{3}$
15	$\frac{6}{5}$	1	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	0	1	1	1	1	1	1	1	1
16	1	$\frac{6}{5}$	1	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	1	0	1	1	1	1	1	1	1
17	1	1	$\frac{6}{5}$	1	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	1	1	0	1	1	1	1	1	1
18	1	1	1	$\frac{6}{5}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	1	1	1	0	1	1	1	1	1
19	$\frac{6}{3}$	1	$\frac{6}{3}$	1	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	0	1	1	1	1
20	1	$\frac{6}{3}$	1	$\frac{6}{3}$	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{3}$	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1	0	1	1	1
21	1	1	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
22	1	1	1	1	1	$\frac{6}{3}$	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1	0	$\frac{6}{2}$
23	1	1	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	$\frac{6}{2}$	0

Bảng 2. Đánh giá trọng số kiểm soát rủi ro của Lưu An Long

[illegible]

Bảng 3. Đánh giá hệ số mật độ di chuyển của Lưu An Long

[illegible]

2. Đánh giá của cán bộ kỹ thuật xây dựng Võ Văn Tài, làm việc tại Công ty TNHH Xây dựng và Đầu tư

Bảng 1. Đánh giá trọng số gần gũi của Võ Văn Tài

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{1}$	$\frac{6}{2}$	$\frac{6}{4}$	1	1	1	$\frac{6}{4}$	1	1	1	1
2	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{1}$	$\frac{6}{2}$	1	$\frac{6}{4}$	1	1	1	$\frac{6}{4}$	1	1	1
3	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{1}$	$\frac{6}{2}$	1	1	$\frac{6}{4}$	1	$\frac{6}{4}$	1	1	1	1
4	0	0	0	0	0	0	0	0	1	1	1	$\frac{6}{1}$	$\frac{6}{2}$	1	1	1	$\frac{6}{4}$	1	$\frac{6}{4}$	1	1	1
5	0	0	0	0	0	0	0	0	$\frac{6}{5}$	1	$\frac{6}{5}$	1	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	1	1	1
6	0	0	0	0	0	0	0	0	1	$\frac{6}{5}$	1	1	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$
7	0	0	0	0	0	0	0	0	1	1	1	1	1	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1
8	0	0	0	0	0	0	0	0	1	1	1	1	1	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1
9	1	1	1	1	$\frac{6}{5}$	1	1	1	0	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	$\frac{6}{5}$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	$\frac{6}{3}$	1
11	1	1	1	1	$\frac{6}{5}$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
13	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{1}$	1	1	1	1	$\frac{6}{3}$	1	1	0	1	1	1	1	1	1	1	1	1	1
14	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{2}$	$\frac{6}{2}$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	$\frac{6}{3}$
15	$\frac{6}{4}$	1	1	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	0	1	1	1	1	1	1	1	1
16	1	$\frac{6}{4}$	1	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	1	0	1	1	1	1	1	1	1
17	1	1	$\frac{6}{4}$	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	1	1	0	1	1	1	1	1	1
18	1	1	1	$\frac{6}{4}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	1	1	1	0	1	1	1	1	1
19	$\frac{6}{4}$	1	$\frac{6}{4}$	1	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	1	1	1	1	0	1	1	1	1
20	1	$\frac{6}{4}$	1	$\frac{6}{4}$	$\frac{6}{1}$	$\frac{6}{1}$	$\frac{6}{4}$	$\frac{6}{4}$	1	1	1	1	1	1	1	1	1	1	0	1	1	1
21	1	1	1	1	1	$\frac{6}{4}$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
22	1	1	1	1	1	$\frac{6}{4}$	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	1	1	1	0	$\frac{6}{2}$
23	1	1	1	1	1	$\frac{6}{4}$	1	1	1	1	1	1	$\frac{6}{3}$	1	1	1	1	1	1	1	$\frac{6}{2}$	0

Bảng 2. Đánh giá trọng số kiểm soát rủi ro của Võ Văn Tài

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	50	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	50	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	50	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	50	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	100	100	100	100	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	50
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	50	50	50	50	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	50
22	0	0	0	0	0	0	0	0	0	0	0	0	0	25	25	25	25	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	50	0	0

Bảng 3. Đánh giá hệ số mật độ di chuyển của Võ Văn Tài

[illegible]