|  |
| --- |
|  |
| **TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  **VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  **---------------------------------------** |
|  |
|  |
|  |
|  |
| **LUẬN VĂN THẠC SỸ** |
|  |
|  |
| |  |  | | --- | --- | | **Tên bài tập lớn:** | Phát triển và thử nghiệm một số thuật toán trên MOEA Framework cho bài toán đấu thầu nhiều vòng | | **Sinh viên:** | Đoàn Tuấn Anh – Lớp CNTT 2016B | | **Giảng viên:** | PGS.TS. Huỳnh Quyết Thắng | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Hà Nội – 2019 |

# LỜI CAM ĐOAN

Tôi Đoàn Tuấn Anh

Xin cam đoan đây là công trình nghiên cứu của cá nhân tôi dưới sự hướng dẫn của **PGS.TS Huỳnh Quyết Thắng**, Bộ môn Công nghệ phần mềm – Viện Công nghệ thông tin và Truyền thông – Trường Đại học Bách Khoa Hà Nội.

Các tài liệu tham khảo trong nghiên cứu được liệt kê tại phần Tài liệu tham khảo ở cuối báo cáo, các nội dung trích dẫn đã ghi rõ nguồn gốc.

Các số liệu, kết quả nêu trong báo cáo là trung thực, rõ ràng.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  **Sinh viên thực hiện nghiên cứu**  **(ký, ghi rõ họ tên)** |

# LỜI CẢM ƠN

Lời đầu tiên tôi xin gửi lời cám ơn chân thành và sâu sắc tới Thầy PGS.TS. Huỳnh Quyết Thắng, Bộ môn Công nghệ phần mềm – Viện Công nghệ thông tin và Truyền thông – Trường Đại học Bách Khoa Hà Nội.

Trong quá trình thực hiện nghiên cứu, Thầy đã hướng dẫn và chỉ bảo tôi ân cần, cung cấp cho tôi nhiều tài liệu quan trọng là nền tảng để tôi tìm tòi nghiên cứu sâu hơn về đề tài “Phát triển và thử nghiệm một số thuật toán trên MOEA Framework cho bài toán đấu thầu nhiều vòng”. Mỗi lời góp ý của Thầy giúp tôi định hướng tốt hơn, đúng đắn hơn cho đề tài mình đã chọn.

Tiếp theo tôi xin dành lời cảm ơn tới bạn bè, các thành viên trong lớp CNTT CH2016B và các bạn sinh viên Lớp KSTN CNTT K59 đã không ngần ngại việc chia sẻ kiến thức cho tôi trong quá trình thực hiện nghiên cứu này.

# MỤC LỤC

[LỜI CAM ĐOAN 2](#_Toc3529764)

[LỜI CẢM ƠN 3](#_Toc3529765)

[MỤC LỤC 4](#_Toc3529766)

[DANH MỤC CÁC BẢNG VÀ HÌNH VẼ 8](#_Toc3529767)

[DANH MỤC CÁC HÌNH VẼ 8](#_Toc3529768)

[DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT 9](#_Toc3529769)

[CHƯƠNG 1. TỔNG QUAN 10](#_Toc3529770)

[1. Đầu thầu và đấu thầu nhiều vòng 10](#_Toc3529771)

[a) Khái niệm đấu thầu 10](#_Toc3529772)

[b) Đấu thầu nhiều vòng 11](#_Toc3529773)

[2. Các vấn đề của bài toán đấu thầu nhiều vòng 13](#_Toc3529774)

[3. Công nghệ sử dụng: 14](#_Toc3529775)

[4. Nhiệm vụ của luận văn 14](#_Toc3529776)

[Chương 2. Cơ sở lý thuyết 15](#_Toc3529777)

[1. Lý thuyết trò chơi: 15](#_Toc3529778)

[a) Khái niệm 15](#_Toc3529779)

[b) Biểu diễn trò chơi: 15](#_Toc3529780)

[Dạng chuẩn tắc 16](#_Toc3529781)

[Dạng mở rộng 16](#_Toc3529782)

[c) Các loại trò chơi: 17](#_Toc3529783)

[Trò chơi hợp tác và trò chơi không hợp tác 17](#_Toc3529784)

[Trò chơi đối xứng và bất đối xứng 18](#_Toc3529785)

[Trò chơi tổng bằng không và tổng khác không 18](#_Toc3529786)

[Trò chơi đồng thời và trò chơi tuần tự 19](#_Toc3529787)

[Trò chơi thông tin hoàn hảo, thông tin không hoàn hảo 19](#_Toc3529788)

[Các trò chơi dài vô tận 20](#_Toc3529789)

[2. Cân bằng Nash: 20](#_Toc3529790)

[Định nghĩa cơ bản của cân bằng Nash 21](#_Toc3529791)

[Ý nghĩa của cân bằng Nash: 21](#_Toc3529792)

[3. Giải thuật di truyền: 22](#_Toc3529793)

[a) Khái niệm 22](#_Toc3529794)

[b) Quy trình thực hiện giải thuật di truyền 23](#_Toc3529795)

[Khởi tạo: 23](#_Toc3529796)

[Tính toán độ thích nghi 23](#_Toc3529797)

[Chọn lọc 23](#_Toc3529798)

[Quá trình sinh sản 24](#_Toc3529799)

[4. Vấn đề đa mục tiêu (MOP – Multiobjective Problem) 25](#_Toc3529800)

[a) Tối ưu hóa đa mục tiêu 25](#_Toc3529801)

[b) Tối ưu Pareto 26](#_Toc3529802)

[c) Các giải thuật tiến hóa đa mục tiêu (MOEA – Multiobjective Evolutionary Algorithms) 27](#_Toc3529803)

[Xếp hạng Pareto 28](#_Toc3529804)

[Chia sẻ độ thích nghi (fitness sharing) 28](#_Toc3529805)

[*Chia sẻ độ thích nghi dựa vào số đếm vùng lân cận*. 28](#_Toc3529806)

[Phương pháp dùng khoảng cách mật độ 29](#_Toc3529807)

[Đặc điểm của giải thuật VEGA: 29](#_Toc3529808)

[Đặc điểm của giải thuật MOGA: 30](#_Toc3529809)

[Đặc điểm của giải thuật NSGA: 30](#_Toc3529810)

[Đặc điểm của giải thuật NSGA-II: 30](#_Toc3529811)

[Đặc điểm của giải thuật SPEA: 30](#_Toc3529812)

[Đặc điểm của giải thuật SPEA-2 30](#_Toc3529813)

[CHƯƠNG 3. GIẢI QUYẾT BÀI TOÁN 31](#_Toc3529814)

[1. Mô hình hóa bài toán đấu thầu nhiều vòng theo lý thuyết trò chơi và cân bằng Nash 31](#_Toc3529815)

[a) Trò chơi đấu thầu nhiều vòng 31](#_Toc3529816)

[b) Đầu vào của bài toán: 31](#_Toc3529817)

[c) Đầu ra của bài toán: 32](#_Toc3529818)

[d) Mục tiêu của bài toán 32](#_Toc3529819)

[Lợi ích của chủ đầu tư 33](#_Toc3529820)

[Lợi ích của từng nhà thầu 33](#_Toc3529821)

[Cân bằng lợi ích-quan hệ giữa các nhà thầu 34](#_Toc3529822)

[Đảm bảo chất lượng cho dự án 35](#_Toc3529823)

[2. MOEA Framework: 35](#_Toc3529824)

[a) Giới thiệu 35](#_Toc3529825)

[b) Cấu hình sử dụng MOEA: 36](#_Toc3529826)

[Các yếu tố phụ thuộc 36](#_Toc3529827)

[Cấu hình MOEA Framework 36](#_Toc3529828)

[3. Triển khai bài toán đấu thầu nhiều vòng trên MOEA Framework 38](#_Toc3529829)

[ProjectBID 38](#_Toc3529830)

[Package 39](#_Toc3529831)

[Contractor 39](#_Toc3529832)

[ProductOfPackage 39](#_Toc3529833)

[ProductOfContractor 39](#_Toc3529834)

[Product 39](#_Toc3529835)

[Discount 39](#_Toc3529836)

[CHƯƠNG 4. Thử nghiệm và đánh giá 42](#_Toc3529837)

[1. Dữ liệu thử nghiệm: 42](#_Toc3529838)

[2. Các thuật toán thử nghiệm 43](#_Toc3529839)

[a) Thuật toán NSGA-III 43](#_Toc3529840)

[b) Thuật toán GDE3 44](#_Toc3529841)

[c) Thuật toán ε-MOEA 45](#_Toc3529842)

[Khái niệm ε-dominate 45](#_Toc3529843)

[Chi tiết thuật toán ε-MOEA 46](#_Toc3529844)

[3. Chương trình thử nghiệm 48](#_Toc3529845)

[4. Đánh giá các thuật toán 49](#_Toc3529846)

[a) Các tham số đánh giá 49](#_Toc3529847)

[Hypervolumn 49](#_Toc3529848)

[Generational Distance (GD): 50](#_Toc3529849)

[ε+-Indicator: 50](#_Toc3529850)

[Hình 16. Mô tả 51](#_Toc3529851)

[Spacing: 51](#_Toc3529852)

[Population Size 52](#_Toc3529853)

[b) Kết quả cụ thể: 52](#_Toc3529854)

[KẾT LUẬN 56](#_Toc3529855)

[DANH MỤC TÀI LIỆU THAM KHẢO 58](#_Toc3529856)

# DANH MỤC CÁC BẢNG VÀ HÌNH VẼ

[Bảng 1. Mô tả trò chơi chuẩn tắc 15](#_Toc3490981)

[Bảng 2. Mô tả trò chơi đối xứng 17](#_Toc3490982)

[Bảng 3. Mô tả trò chơi có tổng bằng không 18](#_Toc3490983)

[Bảng 4. So sánh trò chơi đồng thời và trò chơi tuần tự 18](#_Toc3490984)

[Bảng 5. Mô tả đại diện thông tin một sản phẩm 41](#_Toc3490985)

[Bảng 6. Mô tả đại diện thông tin một nhà thầu 41](#_Toc3490986)

[Bảng 7. Mô tả đại diện thông tin một gói thầu 42](#_Toc3490987)

[Bảng 8. Giá trị các tham số cho tập data1.json và runSeed = 1 53](#_Toc3490988)

[Bảng 9. Giá trị tham số so sánh cho tập data1.json với runSeed=5 53](#_Toc3490989)

[Bảng 10. Giá trị các tham số cho tập data2.json và runSeed = 1 53](#_Toc3490990)

[Bảng 11. Giá trị tham số so sánh cho tập data2.json với runSeed=5 54](#_Toc3490991)

# DANH MỤC CÁC HÌNH VẼ

[Hình 1. Ví dụ về trò chơi mở rộng 17](#_Toc3491026)

[Hình 2. Trò chơi thông tin đầy đủ 20](#_Toc3491027)

[Hình 3. Quá trình lai ghép 24](#_Toc3491028)

[Hình 4. Quá trình đột biến 24](#_Toc3491029)

[Hình 5. Sự đánh đổi giữa hai mục tiêu của một vấn đề đa mục tiêu tạo thành các tập dominated và non-dominated 26](#_Toc3491030)

[Hình 6. Tập Pareto và pareto front và sự ánh xạ giữa chúng. Tập Pareto chứa các biến, tập Pareto front chứa các mục tiêu; vùng xám trên hình là vùng không gian bị dominate bởi tập Pareto front 27](#_Toc3491031)

[Hình 7. Hướng dẫn sử dụng MOEA Framework trên Eclipse 1 37](#_Toc3491032)

[Hình 8. Hướng dẫn sử dụng MOEA Framework trên eclipse 2 37](#_Toc3491033)

[Hình 9. Hướng dẫn sử dụng MOEA Framework trên eclipse 3 38](#_Toc3491034)

[Hình 10. Biểu diễn các lời giải trên không gian mục tiêu 46](#_Toc3491035)

[Hình 11. Biểu diễn quá trình chọn lọc trong thuật toán ε-MOEA 47](#_Toc3491036)

[Hình 12. Giao diện chương trình xây dựng 48](#_Toc3491037)

[Hình 13. Giao diện chức năng lựa chọn thuật toán và các tham số 49](#_Toc3491038)

[Hình 14. Khoảng không gian Hypervolumn 50](#_Toc3491039)

[Hình 15. Mô tả Generation Distance 50](#_Toc3491040)

[Hình 16. Mô tả ε+-Indicator 51](#_Toc3491041)

[Hình 17. Mô tả Spacing 51](#_Toc3491042)

[Hình 18. Biểu đồ so sánh 03 thuật toán trên tập dữ liệu data1.json 53](#_Toc3491043)

# DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

# CHƯƠNG 1. TỔNG QUAN

|  |
| --- |
| Chương này đưa ra cái nhìn bao quát của toàn bộ luận văn, bao gồm những khái niệm cơ bản nhất về:   * Đấu thầu và đấu thầu nhiều vòng * Bài toán cần giải quyết trong đấu thầu nhiều vòng * Công nghệ sử dụng * Nhiệm vụ của luận văn |

## Đầu thầu và đấu thầu nhiều vòng

### Khái niệm đấu thầu

Theo quy định tại khoản 12 Điều 4 [Luật đấu thầu 2013:](https://luatduonggia.vn/luat-dau-thau-nam-2013/)

“Đấu thầu là quá trình lựa chọn nhà thầu để kí kết và thực hiện hợp đồng cung cấp dịch vụ tư vấn, dịch vụ phi tư vấn, mua sắm hàng hóa, xây lắp; lựa chọn nhà đầu tư để kí kết và thực hiện hợp đồng dự án đầu tư theo hình thức đối tác công tư, dự án đầu tư có sử dụng đất trên cơ sở bảo đảm cạnh tranh, công bằng, minh bạch và hiệu quả kinh tế.”

Theo [Wikipedia](https://vi.wikipedia.org/wiki/Đấu_thầu), đấu thầu là quá trình chủ đầu tư lựa chọn được một nhà thầu đáp ứng các yêu cầu của mình theo qui định của pháp luật.

Trong nền kinh tế thị trường, người mua (chủ đầu tư) tổ chức đấu thầu để người bán (các nhà thầu) cạnh tranh nhau. Mục tiêu của người mua là có được hàng hóa và dịch vụ thỏa mãn các yêu cầu của mình về kỹ thuật, chất lượng và chi phí thấp nhất. Mục đích của nhà thầu là giành được quyền cung cấp hàng hóa dịch vụ đó với giá đủ bù đắp các chi phí đầu vào và đảm bảo mức lợi nhuận cao nhất có thể. Như vậy, bản chất của đấu thầu đã được xã hội thừa nhận như 1 sự cạnh tranh lành mạnh để được thực hiện 1 việc nào đó, một yêu cầu nào đó.

Đặc điểm của đấu thầu

**Thứ nhất**: Đấu thầu là 1 hoạt động thương mại. Trong đó bên dự thầu là các thương nhân có đủ điều kiện và mục tiêu mà bên dự thầu hướng tới là lợi nhuận, còn bên mời thầu là xác lập được hợp đồng mua bán hàng hóa, sử dụng dịch vụ với các điều kiện tốt nhất cho họ.

**Thứ hai**: Đấu thầu là 1 giai đoạn tiền hợp đồng. Hoạt động đấu thầu luôn gắn với quan hệ mua bán hàng hóa, cung ứng dịch vụ. Trong nền kinh tế đấu thầu không diễn ra như một hoạt động độc lập, nó chỉ xuất hiện khi con người có nhu cầu mua sắm hàng hóa và sử dụng dịch vụ. Mục đích cuối cùng của đấu thầu là là giúp bên mời thầu tìm ra chủ thể có khả năng cung cấp hàng hóa, dịch vụ với chất lượng và giá cả tốt nhất. Sau khi quá trình đấu thầu hoàn tất, người trúng thầu sẽ cùng với người tổ chức đấu thầu đàm phán, để kí hợp đồng mua bán hàng hóa, cung ứng dịch vụ hay xây lắp công trình.

**Thứ ba:** Hình thức pháp lí của quan hệ đấu thầu hàng hóa, dịch vụ là hồ sơ mời thầu và hồ sơ dự thầu. Hồ sơ mời thầu là văn bản pháp lí do bên mời thầu lập, trong đó có đầy đủ những yêu cầu về kĩ thuật, tài chính và thương mại của hàng hóa cần mua sắm, dịch vụ cần sử dụng. Còn hồ sơ dự thầu thể hiện năng lực, mức độ đáp ứng của bên dự thầu trước các yêu cầu trong hồ sơ mời thầu.

**Thứ tư**: Giá của gói thầu: xét trên góc độ giá cả thì đấu thầu cần thiết phải có sự khống chế về giá, gọi là giá gói thầu hoạc dự toán \_ được đưa ra bởi bên mời thầu theo khẳ năng tài chính của bên mời thầu. Bên dự thầu đưa ra giá cao hơn khẳ năng tài chính của bên mời thầu thì dù có tốt đến mấy cũng khó có thể thắng thầu. bên dự thầu nào đáp ứng được các yêu cầu của bên mời thầu mà có giá càng thấp thì sẽ càng có cơ hội chiến thắng.

Trong thực tế, việc tổ chức thầu nhằm mục tiêu hiệu quả - cạnh tranh – công bằng – minh bạch tuy nhiên khi tham gia đấu thầu cả bên chủ đầu tư và các nhà thầu tham gia đều có thêm một mục tiêu chung đó là không làm mất lòng đối phương, giữ mỗi quan hệ làm ăn, hợp tác lâu dài.

Đấu thầu bao gồm hai giai đoạn:

Giai đoạn 1: giai đoạn sơ tuyển lựa chọn nhà thầu. Tùy theo quy mô, tính chất của gói thầu, chủ đầu tư thông báo mời thầu trên các phương tiện thông tin đại chúng hoặc gửi thư mời thầu. Chủ đầu tư có trách nhiệm cung cấp cho các nhà thầu tham dự hồ sơ mời dự thầu bao gồm các thông tin sơ bộ về gói thầu và các nội dung chính của hồ sơ mời dự thầu. Nhà thầu tham dự sơ tuyển phải nộp hồ sơ dự thầu kèm theo bảo lãnh dự thầu nhằm đảm bảo nhà thầu đã qua giai đoạn sơ tuyển phải tham gia dự thầu.

Giai đoạn 2: giai đoạn đấu thầu. Các nhà thầu tham gia đấu thầu đưa ra giải pháp, dịch vụ với các mực giá cả. Chủ đầu tư xem xét, lựa chọn nhà thầu. Nhà thầu được lựa chọn là nhà thầu có giá dự thầu hợp lý và mang lại hiệu quả cao nhất cho dự án.

### Đấu thầu nhiều vòng

Đối với các dự án lớn, thời gian thực hiện dự án thường kéo dài. Do đó, dự án thường được chia nhỏ thành nhiều gói nhỏ hơn để triển khai theo từng giai đoạn phù hợp. Thay vì đầu tư cho toàn bộ dự án tại một thời điểm duy nhất thì chủ đầu tư sẽ tổ chức thầu cho từng gói vào các thời điểm khác nhau. Mục đích vẫn là để đem lại lợi ích tối đa cho chủ đầu tư, đồng thời giảm thiểu rủi ro trong thời gian thực hiện dự án.

Trong thực tế, khi một gói thầu được đưa ra, chủ đầu tư và các nhà thầu đều mong muốn thu được lợi nhuận từ gói thầu này. Tuy nhiên, khi mức giá đấu thầu phụ thuộc vào từng nhà thầu thì quyền quyết định lựa chọn nhà thầu nào lại do chủ đầu tư quyết định. Quyết định của chủ đầu tư phần lớn là do quá trình đàm phán và tầm nhìn của chủ đầu tư mà không có một cơ sở lý thuyết khoa học nào chứng minh rằng quyết định đó là tối ưu. Do đó, dự án có gói thầu đó có nguy cơ gặp phải một số rủi ro không đáng có như: mức chi phí vượt hạn mức cho phép do mức giá chênh lệch quá cao, thời gian dự án kéo dài kèm theo lãi suất chiết khấu tăng cao; không duy trì được mối làm ăn, hợp tác lâu dài; nếu nhà thầu bị ép trong gói này thì phải tăng gói kia để bù lỗ, dẫn tới chủ đầu tư thiệt hại chỗ khác, nhiều nhà thầu đều thế thì chủ đầu tư khó lòng kiểm soát lợi nhuận. [2]

Mặc dù quyết định chọn nhà thầu cho một lần đấu thầu (thường chỉ dựa trên phân tích chủ quan của một người hay một nhóm người thuộc bên chủ thầu) đã là việc khó khăn và rủi ro cao thì các quyết định trong dự án đấu thầu nhiều vòng còn khó khăn và rủi ro hơn nhiều.

Khó khăn đầu tiên trong dự án đấu thầu nhiều vòng là lựa chọn các thời điểm đấu thầu. Việc lựa chọn thời điểm đấu thầu là vô cùng quan trọng vì thời điểm đấu thầu ảnh hưởng trực tiếp tới kinh phí của dự án. Mức giá của các loại nguyên vật liệu luôn luôn biến động theo thời gian, kèm theo giá trị lãi suất chiết khấu, nếu may mắn, chọn đúng thời điểm mức giá cả đạt mức nhỏ nhất thì số tiền phải bỏ ra sẽ ở mức thấp nhất. Nhưng ngược lại, nếu chọn không đúng thời điểm, số tiền thực tế phải bỏ ra so với dự kiến ban đầu sẽ lên tới mức vượt qua tầm kiểm soát khiến dự án gặp phải rủi ro vô cùng lớn.

Khó khăn thứ hai đi liền khó khăn đầu tiên, tại thời điểm lựa chọn, mua bao nhiêu là phù hợp? Lý do để chia một gói thầu to, một dự án ra làm nhiều gói thầu nhỏ tại nhiều thời điểm khác nhau là vì nhiều lý do chủ quan lẫn khách quan, do dự án quá lớn cần chia ra để dễ quản lý, giảm rủi ro cũng như tối ưu số tiền phải thanh toán nhờ chính sách giảm giá từ các nhà thầu cũng như sự biến động giá trị tiền qua thời gian . Như vậy, vấn đề không chỉ nằm ở việc chọn thời điểm đấu thầu mà còn nằm ở việc tại thời điểm đó, mua số lượng là bao nhiêu. Số lượng mua này có đủ để thi công cho tới thời điểm đấu thầu tiếp theo không và với lượng mua đó cùng mức giá tại thời điểm đó có đem lại lợi nhuận không hay có gây nguy cơ rủi ro cho dự án không.

Khó khăn thứ ba là khó khăn chung của tất cả các bài toán đấu thầu, đó là lựa chọn nhà thầu. Làm thế nào để lựa chọn được các nhà thầu thực hiện các gói thầu với chất lượng đảm bảo, giá thành hợp lý và có chính sách hậu mãi tốt?

Giải quyết được ba khó khăn trên, việc đấu thầu nhiều vòng sẽ mang lại lợi ích cho cả chủ đầu tư và các nhà thầu.

Một cách giải quyết các khó khăn trên cho bài toán đấu thầu nhiều vòng là tìm cách mô tả nó dưới dạng trò chơi không hợp tác với người chơi là chủ đầu tư và các nhà thầu, mỗi người chơi có các chiến lược của riêng mình; sau đó dựa trên lý thuyết trò chơi để tìm cân bằng Pareto Nash cho trò chơi này.

## Các vấn đề của bài toán đấu thầu nhiều vòng

Giải quyết 3 khó khăn được đặt ra tại mục 1.2 là quá trình đi tìm lời giải cho bài toán đấu thầu nhiều vòng. Về mặt toán học, đấu thầu nhiều vòng là bài toán tối ưu nhiều mục tiêu, nhiều ràng buộc; rất khó để một người hay một nhóm người có thể đưa ra được nghiệm tối ưu toàn cục một cách nhanh chóng và chính xác. Do đó, một yêu cầu bức thiết được đưa ra cho máy tính là trợ giúp ra quyết định đối với bài toán đấu thầu nhiều vòng dựa vào các thông tin sẵn có của dự án và thông tin từ các nhà thầu. Một hướng tiếp cận để giải quyết bài toán trên là mô tả nó dưới dạng trò chơi với người chơi là chủ đầu tư và các nhà thầu, mỗi người chơi có các chiến lược của riêng mình. Việc giải bài toán đấu thầu nhiều vòng lúc này tương ứng với việc đi tìm cân bằng Nash cho trò chơi được mô tả.

Cụ thể hơn, khi quy bài toán đàm phán đấu thầu nhiều vòng về dạng mô hình trò chơi, các bên sẽ đưa ra tập chiến lược của mình và chủ đầu tư sẽ cố gắng tìm ra một lời giải sao cho có thể đem lại lợi ích nhiều nhất cho bản thân nhưng lại không làm mất lòng đối phương. Đó chính là điểm cân bằng Nash trong trò chơi. Trong bài toán này:

* Dự án có một danh sách các sản phẩm cần mua và được phân chia thành các gói thầu;
* Mỗi gói thầu sẽ bao gồm các thông tin về khoảng thời gian thực thi, tên, số lượng từng sản phẩm mua trong gói, tổng kinh phí nhà thầu dự kiến chi cho gói đó;
* Chủ đầu tư có các chỉ số quan hệ nhà thầu-chủ đầu tư và chỉ số chất lượng (uy tín) của từng nhà thầu;
* Một danh sách các nhà thầu được cho trước, trong đó mỗi nhà thầu cung cấp số lượng và danh sách các gói thầu tham gia, mỗi gói thầu được đặc trưng bởi đơn giá dự thầu, tỷ lệ giảm giá (chiết khấu) sản phẩm trong khoảng thời gian xác định nếu gói thầu diễn ra tại thời điểm đó;

Một cách rõ ràng hơn, máy tính cần đưa ra một hoặc một tập các câu trả lời cho các câu hỏi: Tổ chức đấu thầu vào thời điểm nào? Tại thời điểm đó thì mua gì, mua của ai và mua bao nhiêu? Trả lời những câu hỏi đã nêu sẽ cho những điểm cân bằng Nash của “trò chơi đấu thầu nhiều vòng” được mô tả.

## Công nghệ sử dụng:

MOEA FRAMEWORK là công nghệ sẽ sử dụng để thực triển khai phương án giải quyết bài toán đấu thầu nhiều vòng. Trong chương hai sẽ đề cập rõ ràng hơn về việc mô hình hóa bài toán đấu thầu nhiều vòng nhưng khái quát bài toán đấu thầu nhiều vòng được tiếp cận theo hướng là một trò chơi. Sau khi áp dụng lý thuyết trò chơi để mô hình hóa bài toán thành bài toán tối ưu hóa đa mục tiêu, bài toán sẽ được giải quyết bằng các thuật toán tiến hóa đa mục tiêu đã được tích hợp sẵn trên MOEA Framework.

## Nhiệm vụ của luận văn

Nghiên cứu mô hình hóa bài toán đấu thầu nhiều vòng dựa vào lý thuyết trò chơi và cân bằng Nash

Tìm hiểu các thuật toán di truyền, thuật toán tiến hóa đa mục tiêu để giải quyết bài toán đấu thầu đã được mô hình hóa.

Xây dựng mô hình bài toán đấu thầu nhiều vòng, triển khai các thuật toán tiến hóa đa mục tiêu dựa vào MOEA Framework để giải bài toán đấu thầu nhiều vòng.

Đưa ra so sánh đánh giá các thuật toán tiến hóa đa mục tiêu đã sử dụng

# Chương 2. Cơ sở lý thuyết

|  |
| --- |
| Chương này của luận văn bao gồm các lý thuyết chính được sử dụng để mô hình hóa và giải quyết bài toán đấu thầu nhiều vòng, bao gồm:   * Lý thuyết trò chơi * Cân bằng Nash * Giải thuật di truyền * Vấn đề đa mục tiêu và các giải thuật tiến hóa đa mục tiêu |

## Lý thuyết trò chơi:

### Khái niệm

Lý thuyết trò chơi là một ngành chuyên nghiên cứu về việc đưa ra quyết định chiến lược. Lý thuyết trò chơi thường được mô tả như một lý thuyết trong toán học, nghiên cứu tình huống trong đó người chơi sẽ hành động theo những cách khác nhau để tối ưu hóa lợi ích của mình. Điều quan trọng nhất của lý thuyết này là phương pháp tiếp cận để đưa ra các quyết định để giải quyết một vấn đề nào đó. Hay nói cách khác là nó sẽ xác định xác suất thành công khi cho trước một không gian chiến lược.

Lý thuyết trò chơi được áp dụng sử dụng vào rất nhiều các ngành, lĩnh vực như Chính trị học, Đạo đức học, Kinh tế. . . và đặc biệt là Khoa học máy tính ứng dụng trong Trí tuệ nhân tạo và Điều khiển học. Lý thuyết trò chơi đã dần đóng vai trò ngày càng quan trọng trong logic và khoa học máy tính. Một số lý thuyết logic có cơ sở trong ngữ nghĩa trò chơi, mô phỏng các tính toán tương tác với nhau.

### Biểu diễn trò chơi:

Các trò chơi được nghiên cứu xem xét trong lý thuyết trò chơi là các đối tượng toán học được xác định rõ. Để xác định đầy đủ, một trò chơi phải xác định các yếu tố sau: các người chơi tham gia trò chơi; những thông tin và hành động có sẵn cho mỗi người chơi tại mỗi thời điểm quyết định(tập các chiến lược) và cơ chế thưởng phạt tương ứng với mỗi tổ hợp các chiến lược. Các yếu tố này thường được sử dụng cùng với một khái niệm giải pháp lựa chọn để suy ra một tập hợp các chiến lược cân bằng cho mỗi người chơi. Những chiến lược cân bằng xác định một trạng thái cân bằng của các trò chơi - một trạng thái ổn định, trong đó một trong hai kết quả xảy ra hoặc một loạt các kết quả xảy ra với xác suất đã biết.

#### Dạng chuẩn tắc

Một trò chơi dạng chuẩn tắc:

|  |  |  |
| --- | --- | --- |
|  | Người chơi 2 chọn cột trái | Người chơi 2 chọn cột phải |
| Người chơi 1 chọn hàng trên | 4, 3 | -1, -1 |
| Người chơi 1 chọn hàng trên | 0, 0 | 3, 4 |

###### Mô tả trò chơi chuẩn tắc

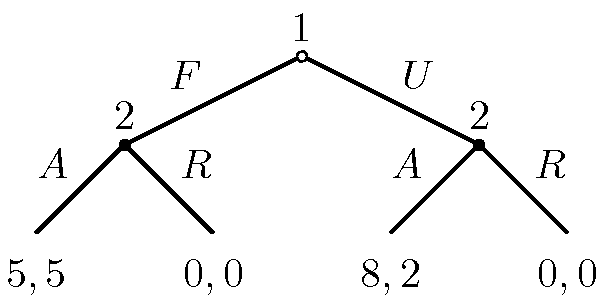
Trò chơi chuẩn tắc(hay trò chơi chiến lược) thường được biểu diễn bằng một ma trận cho biết thông tin về các người chơi, chiến lược và cơ chế thưởng phạt. Nói chung, nó có thể biểu diễn bằng bất kì hàm nào đó mà liên kết giữa cơ chế thưởng phạt với tổ hợp các chiến lược, hành động tương ứng. Lấy một ví dụ: có hai người chơi, một người chọn hàng, một người chọn cột. Mỗi người chơi có hai chiến lược, mỗi chiến lược được biểu diễn một ô được xác định bởi số hiệu hàng và số hiệu cột của nó. Mức độ thưởng phạt được ghi trong ô đó. Giá trị thứ nhất là mức thưởng phạt cho đấu người chơi theo hàng; giá trị thứ hai là mức thưởng phạt tương ứng với người chơi theo cột. Giả sử người chơi 1 chọn hàng trên và người chơi 2 chọn cột trái. Khi đó, người chơi 1 nhận được 4 điểm và người chơi 2 nhận được 3 điểm.

Khi một trò chơi được biểu diễn bằng dạng chuẩn tắc, nó được coi rằng mỗi người chơi hành động một cách đồng thời hoặc ít nhất không biết hành động của người kia. Nếu mỗi người chơi có thông tin về sự lựa chọn của người chơi khác thì trò chơi thường được biểu diễn theo kiểu mở rộng.

Mỗi dạng trò chơi mở rộng có một dạng trò chơi chuẩn tắc tương ứng. Tuy nhiên, việc chuyển đổi từ hình thức chuẩn tắc sang dạng mở rộng có thể dẫn đến việc biểu diễn và tính toán trong thực tế trở nên khó khăn hơn.

#### Dạng mở rộng

Các trò chơi dạng mở rộng có thể được sử dụng để hợp thức hóa các trò chơi với một trình tự thời gian của các lượt đi. Các trò chơi sẽ được biểu diễn bằng cây. Mỗi đỉnh (hoặc nút) biểu diễn một điểm mà người chơi có thể lựa chọn. Người chơi được xác định bằng một số ghi cạnh đỉnh. Các đoạn thẳng đi ra từ đỉnh đó biểu diễn các hành động có thể cho người chơi đó. Mức thưởng phạt được ghi rõ tại đáy cây. Các dạng mở rộng có thể được xem như là một sự tổng quát nhiều người chơi của một cây quyết định. Trong trò chơi trong hình gồm có hai người chơi. Người chơi 1 đi trước và chọn F hoặc U. Người chơi 2 nhìn thấy nước đi của người chơi 1 và chọn A hoặc R. Sau khi hai người chơi đã đưa ra sự lựa chọn của mình thì trò chơi coi như được hoàn thành và mỗi người chơi sẽ nhận được cơ chế thưởng phạt tương ứng. Giả sử người chơi 1 chọn U và sau đó người chơi 2 chọn A. Khi đó, người chơi 1 được 8 điểm và người chơi 2 được 2 điểm.



1. Ví dụ về trò chơi mở rộng

Các trò chơi mở rộng còn có thể mô tả các trò chơi đi đồng thời và các trò chơi với những thông tin không hoàn hảo. Để biểu diễn nó, sử dụng hoặc là một đường chấm chấm nối hai đỉnh khác nhau hoặc một đường tròn vẽ quanh hai đỉnh khác nhau để biểu diễn rằng chúng đều thuộc cùng một tập hợp thông tin.

### Các loại trò chơi:

#### Trò chơi hợp tác và trò chơi không hợp tác

Trò chơi hợp tác là trò chơi mà giữa các người chơi có hình thành các cam kết ràng buộc. Còn trong trò chơi không hợp tác thì điều này không xảy ra. Thông thường thì sự trao đổi, giao tiếp giữa các người chơi được cho phép trong trò chơi hợp tác. Trong hai loại trò chơi, trò chơi không hợp tác có thể mô hình hóa tình huống đến các chi tiết nhỏ nhất và cho các kết quả chính xác. Trò chơi hợp tác tập trung vào các trò chơi có độ phức tạp lớn. Một nỗ lực đáng kể đã được thực hiện nhằm liên kết giữa hai cách tiếp cận. Chương trình Nash đã xây dựng những giải pháp hợp tác là các điểm cân bằng không hợp tác. Trò chơi lai là trò chơi chứa các thành phần hợp tác và không hợp tác.

#### Trò chơi đối xứng và bất đối xứng

Một trò chơi đối xứng là một trò chơi mà cơ chế thưởng phạt ứng với một chiến lược cụ thể nào đó chỉ phụ thuộc vào chiến thuật được sử dụng, chứ không phụ thuộc vào người nào đang chơi. Nếu như tính danh của những người chơi có thể thay đổi mà không làm thay đổi phần lợi đối với chiến thuật chơi thì một một trò chơi là đối xứng. Các bài toán song đề tù nhân, săn hươu. . . đều là các trò chơi đối xứng.

|  |  |  |
| --- | --- | --- |
|  | E | F |
| E | 1, 2 | 0, 0 |
| F | 0, 0 | 1, 2 |

###### Mô tả trò chơi đối xứng

Đa số những trò chơi bất đối xứng được nghiên cứu là những trò chơi mà tập hợp chiến thuật khác nhau được sử dụng bởi hai người chơi. Ví dụ trò chơi tối hậu thư và trò chơi kẻ độc tài. . . là các trò chơi không đối xứng vì mỗi người chơi có các chiến lược khác nhau. Tuy nhiên, trong một vài trường hợp trò chơi vẫn có chiến lược giống nhau cho cả hau người chơi nhưng vẫn là trò chơi bất đối xứng.

#### Trò chơi tổng bằng không và tổng khác không

Trò chơi tổng bằng không là một trường hợp đặc biệt của trò chơi tổng không đổi, trong đó sự lựa chọn của người chơi không làm thay đổi các nguồn lực có sẵn. Trong trò chơi tổng bằng không, với mọi tổ hợp của các chiến lược chơi, tổng điểm của tất cả các người chơi trong ván chơi luôn bằng 0. Nói một cách không chính thức, đấu thủ này hưởng lợi từ thiệt hại của đối thủ khác. Các loại cờ cổ điển như cờ vây, cờ vua và cờ tướng cũng là các trò chơi tổng bằng 0

|  |  |  |
| --- | --- | --- |
|  | A | B |
| A | -1, 1 | 3, -3 |
| B | 0, 0 | -2, 2 |

###### Mô tả trò chơi có tổng bằng không

Nhiều trò chơi mà các nhà lý thuyết trò chơi nghiên cứu, trong đó có song đề tù nhân là các trò chơi tổng khác không, do có một số kết cục có tổng kết quả lớn hơn hoặc nhỏ hơn 0. Nói một cách không chính thức, trong các trò chơi tổng khác 0 thì một lợi ích của người chơi này không nhất thiết phải tương ứng với thiệt hại của người chơi kia. Có thể biến đổi một trò chơi bất kỳ thành một trò chơi tổng bằng 0 bằng cách bổ sung một người chơi bù nhìn sao cho thiệt hại của người chơi bù nhìn này bù lại tổng thu hoạch của các người chơi khác.

#### Trò chơi đồng thời và trò chơi tuần tự

Trò chơi đồng thời là trò chơi mà cả hai người chơi di chuyển đồng thời hoặc nếu nếu không di chuyển cùng lúc thì người chơi đi sau không thể xác định được những hành động của người chơi đã đi trước. Trò chơi tuần tự là trò chơi mà người chơi mà người chơi sau có một số thông tin về hành động trước đó. Thông tin này không cần phải là thông tin đầy đủ về mọi hành động của người chơi trước đó. Ví dụ, người chơi 1 có thể biết rằng người chơi 2 đã thực hiện những hành động cụ thể nào trước đó nhưng người chơi đó lại không biết được những hành động có sẵn khác của người chơi 2

So sánh trò chơi đồng thời và trò chơi tuần tự:

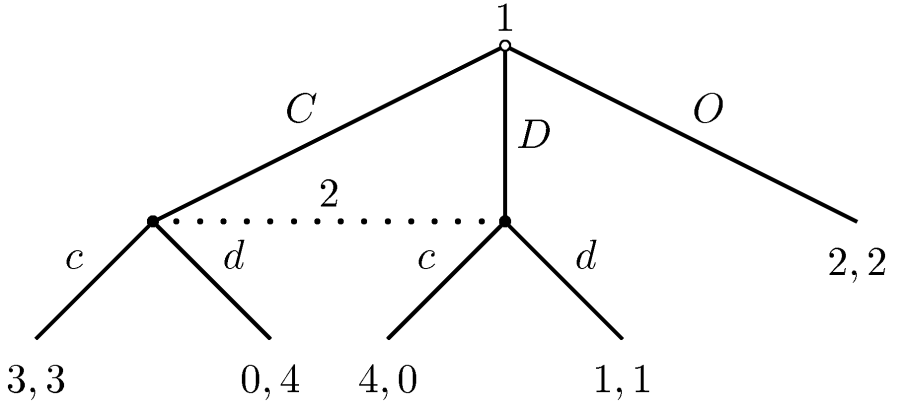
|  |  |  |
| --- | --- | --- |
|  | Trò chơi tuần tự | Trò chơi đồng thời |
| Biểu diễn bằng | Cây quyết định | Ma trận thưởng phạt |
| Trục thời gian | Có | Không |
| Còn được gọi là | Trò chơi mở rộng | Trò chơi chiến lược |

###### So sánh trò chơi đồng thời và trò chơi tuần tự

Biểu diễn dạng chuẩn tắc được dùng để biểu diễn các trò chơi đồng thời và biểu diễn dạng mở rộng được dùng cho các trò chơi tuần tự.

#### Trò chơi thông tin hoàn hảo, thông tin không hoàn hảo

Các trò chơi thông tin hoàn hảo lập thành một tập con quan trọng của các trò chơi tuần tự. Một trò chơi được gọi là có thông tin hoàn hảo nếu mọi người chơi biết tất cả các nước đi của người chơi khác đã thực hiện. Do vậy chỉ có các trò chơi tuần tự mới có thể là các trò chơi thông tin hoàn hảo. Hầu hết các trò chơi nghiên cứu trong lý thuyết trò chơi là các trò chơi có thông tin không hoàn hảo, tuy một số trò chơi hay như cờ vây, cờ vua lại là trò chơi thông tin hoàn hảo.



1. Trò chơi thông tin đầy đủ

Tính chất thông tin hoàn hảo thường bị nhầm lẫn với khái niệm về thông tin đầy đủ. Tính chất thông tin đầy đủ đòi hỏi rằng mỗi người chơi biết về các chiến lược và thành quả thu được của các người chơi khác, nhưng không nhất thiết về các hành động của họ.

#### Các trò chơi dài vô tận

Bởi các lí do hiển nhiên, các trò chơi được nghiên cứu bởi các nhà kinh tế gia và những người chơi trong thế giới thực nhìn chung là kết thúc trò chơi trong hữu hạn các bước đi. Các nhà toán học lý thuyết không bị cản trở bởi điều đó và lý thuyết về tập hợp đặc biệt nghiên cứu về các trò chơi kết thúc sau vô hạn các bước đi bởi người thắng là không biết được cho đến sau khi các bước đó hoàn thành.

Sự chú ý thường không phải là quá nhiều về cách nào tốt nhất để chơi trò chơi mà đơn giản là chỉ phụ thuộc vào người chơi hay người kia có hay không một chiến thuật chiến thắng.

## Cân bằng Nash:

Cân bằng Nash là một trong những khái niệm được sử dụng nhiều nhất trong các ứng dụng của lý thuyết trò chơi trong kinh tế học khi phải nghiên cứu những hành vi tương tác lẫn nhau của những tác nhân có những mục tiêu mâu thuẫn nhau.

#### Định nghĩa cơ bản của cân bằng Nash

Nếu tồn tại một tập hợp các chiến cho một trò chơi với đặc tính là không một đối thủ nào có thể hưởng lợi bằng cách thay đổi chiến lược hiện tại của mình khi các đối thủ khác không thay đổi, tập hợp các chiến lược đó và phần thu nhận tương ứng tạo nên cân bằng Nash. Nói cách khác, cân bằng Nash đạt được nếu như thay đổi một cách đơn phương của bất cứ ai trong số các đối thủ cũng sẽ làm cho chính người đó thu lợi ít hơn mức có được với chiến lược hiện tại. Khái niệm này áp dụng cho những trò chơi gồm từ hai đối thủ trở lên và Nash đã chỉ ra rằng tất cả các khái niệm khác nhau về giải pháp (solution) trong các trò chơi được đưa ra trước đó đều có cân bằng Nash.

Một ví dụ đơn giản: trong một trò chơi gồm hai đối thủ cùng chọn song song một số bất kỳ từ 0 đến 10, người nào chọn số lớn hơn sẽ thua và phải trả tiền cho người kia. Trò chơi này chỉ có 1 cân bằng Nash duy nhất: cả hai đối thủ đều chọn số 0. Bất kỳ sự lựa chọn nào khác (mà không biết sự lựa chọn của đối thử kia) cũng có thể làm đối thủ thua cuộc.

#### Ý nghĩa của cân bằng Nash:

Trong trò chơi gồm n người chơi, mỗi người chơi có sự lựa chọn các chiến lược để thực hiện, ứng với mỗi người chơi là một sự chi trả của người chơi cho tất cả các kết quả có thể xảy ra tương ứng với sự lựa chọn chiến lược của các người chơi. Mỗi người chơi có thể lựa chọn một chiến lược hỗn hợp và kết hợp các lựa chọn các chiến lược hỗn hợp của những người chơi khác xác định kết quả trung bình hoặc giá trị kỳ vọng cho mỗi người chơi. Định lý Nash nói rằng mỗi người chơi có một tập các chiến lược hỗn hợp tối ưu khi biết sự lựa chọn chiến lược hỗn hợp của các người chơi khác. Mỗi chiến lược hỗn hợp tối ưu đưa đến kết quả trong giá trị kỳ vọng lớn nhất có thể cho người chơi khi biết chiến lược hỗn hợp của các người chơi khác. Một cân bằng Nash là một sự lựa chọn của chiến lược hỗn hợp mà kết quả cho mỗi người chơi là các giá trị kỳ vọng lớn nhất có thể ứng với chiến lược hỗn hợp của các người chơi khác.

Một trò chơi có thể có nhiều hoặc không có cân bằng Nash. Nash cũng chứng minh rằng nếu cho phép các chiến lược hỗn hợp (mixed strategies) tức là các đối thủ chọn ngẫu nhiên các chiến lược dựa vào khả năng đã được ấn định trước, thì bất cứ một trò chơi với n đối thủ nào trong đó mỗi đối thủ có thể chọn trong giới hạn cho trước nhiều chiến lược sẽ có ít nhất 1 cân bằng Nash của các chiến lược hỗn hợp.

Cân bằng Nash giúp làm rõ sự phân biệt giữa các trò chơi hợp tác và không hợp tác. Trong các trò chơi không hợp tác không tồn tại cơ chế thỏa thuận, vì thế chỉ có thỏa thuận cân bằng mới được duy trì. Một hướng lí thuyết trò chơi mới đươc mở đường bằng cân bằng Nash xóa bỏ sự phân biệt này bằng cách xóa bỏ các cơ chế áp đặt có liên quan trong mô hình trò chơi, từ đó các trò chơi được mô hình hóa với tính chất không hợp tác.

## Giải thuật di truyền:

### Khái niệm

Các quy tắc di truyền trong chọn lọc tự nhiên cung cấp cho hướng tiến hóa sinh học của sinh vật để thích ứng và đặc điểm của môi trường các nhân tố tốt nhất được lựa chọn thông qua các đặc điểm thích nghi môi trường, từ đó chúng có khả năng tốt hơn trong việc tồn tại và tái sản xuất. Lựa chọn này phát triển từ thế hệ này sang thế hệ khác cho phép tăng sự thích nghi trung bình quần thể.

Genetic algorithms (thuật giải di truyền) là một giải thuật mô phỏng theo quá trình chọn lọc tự nhiên, là kỹ thuật chung giúp giải quyết vấn đề bài toán bằng cách mô phỏng sự tiến hóa của con người hay của sinh vật nói chung (dựa trên thuyết tiến hóa muôn loài của Darwin) trong điều kiện qui định sẵn của môi trường. Lấy ý tưởng từ quá trình tiến hoá tự nhiên, xuất phát từ một lớp các lời giải tiềm năng ban đầu, GA tiến hành tìm kiếm trên không gian lời giải bằng cách xây dựng lớp lời giải mới tốt hơn (tối ưu hơn) lời giải cũ. Quá trình xây dựng lớp lời giải mới được tiến hành dựa trên việc chọn lọc, lai ghép, đột biến từ lớp lời giải ban đầu. Quần thể lời giải trải qua quá trình tiến hoá: ở mỗi thế hệ lại tái sinh các lời giải tương đối tốt, trong khi các lời giải “xấu” thì chết đi.

Trong giải thuật di truyền, một tập các biến của bài toán đưa ra được mã hóa sang một chuỗi (hay một cấu trúc mã hóa khác) tương tự như một nhiễm sắc thể trong tự nhiên. Mỗi chuỗi bao gồm một giải pháp có thể của bài toán. Giải thuật di truyền sử dụng các toán tử được sinh ra bởi sự chọc lọc tự nhiên một quần thể các chuỗi nhị phân (hoặc các cấu trúc khác), mã hóa khoảng tham số trên mỗi thế hệ, khảo sát các phạm vi khác nhau của không gian tham số, và định hướng tìm kiếm đối với khoảng mà là xác suất cao để tìm kiếm sự thực hiện tốt hơn. Thuật toán di truyền gồm có bốn quy luật cơ bản là lai ghép, đột biến, sinh sản và chọn lọc tự nhiên. ([02] Coello Coello, 2007)

Giải thuật di truyền sử dụng một số thuật ngữ của ngành di truyền học như: Nhiễm sắc thể, Quần thể, Gen… Nhiễm sắc thể (NST) được tạo thành từ các Gen (được biểu diễn của một chuỗi tuyến tính). Mỗi gen mang một số đặc trưng và có vị trí nhất định trong NST. Mỗi NST sẽ biểu diễn một lời giải của bài toán.

### Quy trình thực hiện giải thuật di truyền

#### Khởi tạo:

Quần thể là một tập hợp các cá thể có cùng một số đặc điểm nào đấy. Trong giải thuật di truyền quan niệm quần thể là một tập các lời giải của một bài toán. Quần thể ban đầu ảnh hưởng khá nhiều đến hiệu quả giải thuật, tuy nhiên trong nhiều bài toán thì quần thể ban đầu thường được lựa chọn ngẫu nhiên. Thường phụ thuộc vào kích thước chuỗi mã hóa. VD: Nếu có NST 32 bits, thì kích thước quần thể nên cao hơn 16.

Kích thước quần thể cho biết có bao nhiêu cá thể trong một quần thể trong mỗi thế hệ. Các nghiên cứu và các thử nghiệm đã cho thấy kích thước quần thể không nên quá bé cũng như không quá lớn. Nếu có quá ít cá thể thì sẽ làm giảm không gian tìm kiếm của giải thuật và dễ rơi vào các cục bộ địa phương, như vậy sẽ dễ xảy ra trường hợp bỏ qua các lời giải tốt. Tuy nhiên nếu có quá nhiều cá thể cũng sẽ làm cho giải thuật chạy chậm đi, ảnh hưởng đến hiệu quả tính toán của giải thuật. Các nghiên cứu cũng đã chỉ ra không có lợi khi tăng kích thước quần thể lên quá một giới hạn cho phép.

#### Tính toán độ thích nghi

Sau khi hoàn thành quá trình lai ghép chéo tạo ra các thế hệ mới nhằm duy trì và tạo sự đa dạng trong quần thể thì cần phải tính lại độ thích nghi cho từng cá thể mới hình thành. Số lượng các cá thể trong quần thể tăng lên qua lai ghép và độ thích nghi giữa các cá thể không có sự chênh lệch đáng kể. Do đó, các cá thể có độ thích nghi cao chưa hẳn chiếm ưu thế trong thế hệ tiếp theo. Vì vậy, cần ấn định tỷ lệ đối với hàm thích nghi nhằm nâng cao khả năng cho các nhiễm sắc thể đạt độ thích nghi cao hay chính là đánh giá chất lượng lời giải cho bài toán.

#### Chọn lọc

Toán tử chọn lọc (hay tái sinh) thường là toán tử đầu tiền áp dụng trong một quần thể. Toán tử chọn lọc là hình thức chọn lọc cá thể tốt nhất và có dạng như một tổ hợp lai ghép. Ý tưởng ban đầu là chọn một cá thể có độ thích nghi trên trung bình rồi đưa vào tổ hợp lai ghép. Toán tử chọn lọc thường được sử dụng chính là để chọn lọc các cá thể có độ thích nghi phù hợp tương ứng với điều kiện đặt ra của bài toán.

#### Quá trình sinh sản

##### Lai ghép

Toán tử lai ghép là quá trình tạo mới được tiến hành tại bước tiếp theo sau khi chọn lọc cá thể thích hợp trong một quần thể bằng cách đưa vào tổ hợp lai ghép. Trong toán tử lai ghép có hai cá thể được chọn một cách ngẫu nhiên từ tổ hợp lai ghép. Toán tử lai ghép chính là quá trình tạo NST mới trên cơ sở các NST cha mẹ bằng cách ghép một đoạn trên NST cha mẹ với nhau được một cá thể mới để đưa vào quần thể.



1. Quá trình lai ghép

##### Đột biến

Toán tử đột biến là cá thể con mang một số đặc tính không có trong mã di truyền của cha mẹ hay một đoạn mã di truyền đã bị thay đổi tức là khác với cá thể cha mẹ. Toán tử di truyền được thực hiện bằng cách chọn ngẫu nhiên một đoạn mã di truyền trong quần thể rồi tạo ra một số k ngẫu nhiên trong khoảng từ 1 đến m với điều kiện 1 ≤ k ≤ m rồi thay đổi mã thứ k và đưa vào quần thể để tham gia quá trình tiến hóa ở thế hệ tiếp theo.



1. Quá trình đột biến

Như ví dụ trên có thể thấy đã có ba đoạn mã di truyền bị thay đổi so với bản mã di truyền gốc và tạo ra một mã di truyền mới tương ứng với một cá thế mới trong quần thể. Điều quan trọng đối với quần thể chính là nhu cầu đột biến để duy trì sự đa dạng trong quần thể.

## Vấn đề đa mục tiêu (MOP – Multiobjective Problem)

### Tối ưu hóa đa mục tiêu

Tối ưu hóa là là quá trình xác định giải pháp tốt nhất trong số các giải pháp thay thế (Miettinen, 1999). Trong khi tối ưu hóa một mục tiêu sử dụng một tiêu chí duy nhất để xác định giải pháp tốt nhất trong số các giải pháp thay thế thì tối ưu hóa đa mục tiêu sử dụng hai hoặc nhiều tiêu chí để xác định giải pháp tốt nhất. Các tiêu chí sử dụng để so sánh đánh giá các giải pháp được gọi là mục tiêu. Khi có nhiều mục tiêu, hai hay vài mục tiêu có thể xung đột nhau (có nghĩa là việc cải thiện mục tiêu này dẫn đến sự suy giảm của mục tiêu khác) do đó không có giải pháp tối ưu duy nhất nào cho các vấn đề đa mục tiêu hay nói một cách khác không có lời giải duy nhất cho bài toán tối ưu hóa đa mục tiêu.

Một cách toán học, bài toán tối ưu hoá đa mục tiêu là bài toán tối ưu hoá (cực đại hoặc cực tiểu) một tập các hàm mục tiêu *fi(x)* xác định trên tập Ω :

*F(x) = (f1(x), f2(x), …, fM(x)),*

đồng thời thoả mãn một số ràng buộc :

*ci(x) = 0,* ∀ *i ∈ E*

*cj(x) ≤ 0,* ∀ *j ∈ I*

Trong đó, *x* gọi là biến quyết định, là vector *L* chiều :

Như đã khẳng định ở trên, bài toán tối ưu hoá đa mục tiêu thường không có một lời giải đảm bảo tối ưu trên tất cả các mục tiêu mà sẽ tồn tại một tập các lời giải "**tốt**".

Để đánh giá lời giải nào tốt hơn lời giải nào, người ta đưa ra khái niệm **dominate.** Lời giải *x1* được gọi là dominate (trội hơn) lời giải *x2* khi :

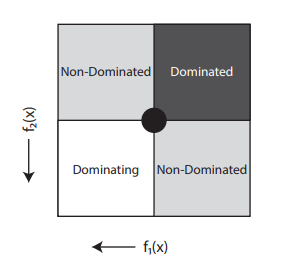
*fi(x1) ≤ fi(x2)* ∀ *i ∈ {1, 2, …, M}*

*j, fj(x1) < fj(x2)*

tức là *x1* không tồi hơn *x2* tại tất cả các mục tiêu và tốt hơn *x2* tại ít nhất một mục tiêu.

### Tối ưu Pareto

Với các bài toán tối ưu đa mục tiêu do không có lời giải tối ưu duy nhất nên khái niệm về sự tối ưu đa mục tiêu được Francis Ysidro Edgeworth và Vilfredo Pareto đưa ra vào năm 2007 và được gọi là tối ưu Pareto. Sự tối ưu của Pareto coi các giải pháp là vượt trội hoặc kém hơn so với giải pháp khác chỉ khi nó vượt trội trong tất cả các mục tiêu hoặc kém hơn trong tất cả các mục tiêu, tương ứng. Trong vấn đề đa mục tiêu (MOP) sẽ luôn bắt gặp sự đánh đổi giữa một trong các mục tiêu còn lại, tức là sẽ tồn tại các giải pháp vượt trội hơn giải pháp khác ở một số mục tiêu nhưng lại kém hơn ở những mục tiêu khác. Các giải pháp như vậy tạo thành tập non-dominated (không bị chi phối) được thấy rõ trong hình ……



1. Sự đánh đổi giữa hai mục tiêu của một vấn đề đa mục tiêu tạo thành các tập dominated và non-dominated

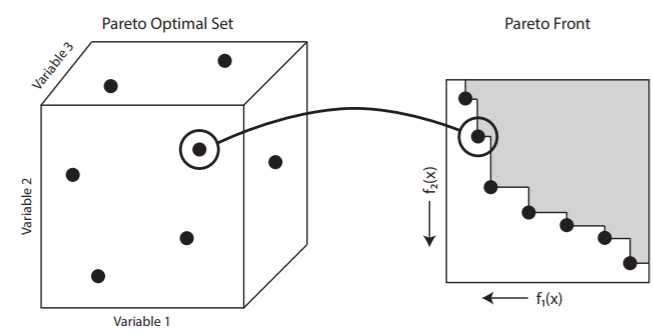
Khái niệm dominate. Giải pháp *x1* được gọi là dominate (trội hơn) giải pháp *x2* khi :

*fi(x1) ≤ fi(x2)* ∀ *i ∈ {1, 2, …, M}*

*j, fj(x1) < fj(x2)*

tức là *x1* không tồi hơn *x2* tại tất cả các mục tiêu và tốt hơn *x2* tại ít nhất một mục tiêu.

Hai giải pháp được gọi là non-dominated nếu chúng không dominate lẫn nhau. Tập tất cả các giải pháp non-dominated được gọi là tập Pareto, giá trị các hàm mục tiêu được tạo ra bởi các giải pháp non-dominated tạo thành Pareto front (có thể tạm gọi là mặt trước Pareto)



1. Tập Pareto và pareto front và sự ánh xạ giữa chúng. Tập Pareto chứa các biến, tập Pareto front chứa các mục tiêu; vùng xám trên hình là vùng không gian bị dominate bởi tập Pareto front

Một lời giải được gọi là Pareto optimal (tối ưu Pareto) nếu không tìm được lời giải nào dominate nó. Tập các lời giải như vậy gọi là một Pareto Front (có thể gọi là mặt trước Pareto). Trong luận văn này, mối quan hệ dominated pareto được áp dụng cho các mục tiêu chứ không phải các giá trị biến, do đó khi nhắc đến tối ưu Pareto là đang nhắc đến tập Pareto Front (PF). Các giải thuật cho bài toán tối ưu hoá đa mục tiêu thường hướng đến việc tìm ra một tập tối ưu Pareto của bài toán tức là tìm ra một tập Pareto front của bài toán.

### Các giải thuật tiến hóa đa mục tiêu (MOEA – Multiobjective Evolutionary Algorithms)

Như đã đề cập tại mục 3 chương 2, giải thuật di truyền hay còn được gọi là thuật toán tiến hóa là một lớp các thuật toán tìm kiếm và tối ưu hóa được lấy cảm hứng từ các quá trình tiến hóa tự nhiên. Giải thuật tiến hóa đặc biệt phù hợp để giải quyết các bài toán tối ưu đa mục tiêu. Các giải thuật tiến hóa truyền thống có thể được biến cải để tìm kiếm tập Pareto-được-biết-tốt-nhất trong bài toán tối ưu đa mục tiêu chỉ trong một lượt chạy. Và các giải thuật tiến hóa sau khi được cải tiến đó được gọi là các giải thuật tiến hóa đa mục tiêu.

Giải thuật MOEA đầu tiên được biết là Vector Evaluated Genetic Algorithm (VEGA) được đề nghị bởi Schaffer, 1985 [9]. Sau đó, nhiều MOEA khác đã được phát triển bao gồm Multi-objective Genetic Algorithm (MOGA) bởi Fonseca và Fleming, năm 1993 ([10] Fleming, 1993), Niched Pareto Genetic Algorithm (NPGA) bởi Horn và các cộng sự, năm 1994 ([11] J. Horn, 1994), Weight-Based Genetic Algorithm (WBGA) bởi Hajela và Lin, năm 1992 ([12] P. Hajela P, 1995), Random Weight Genetic Algorithm (RWGA) bởi Murata và Ishibuchi, năm 1995 ([13] T. Murata, 1995), Nondominated Sorting Genetic Algorithm (NSGA) bởi Srinivas và Deb, năm 1994 ([14] N. Srinivas, 1994), Strength Pareto Evolutionary Algorithm (SPEA) bởi Zitzler và Thiele năm 1999 ([15] E. Zitzler, 1999), SPEA cải tiến (SPEA2) bởi Zitzler và các cộng sự năm 2001 ([16] E. Zitzler, 2001), Pareto-Archived Evolution Strategy (PAES) bởi Knowles và Corne năm 2000 ([17] J. D. Knowles, 2000), Pareto Enveloped-based Selection Algorithm (PESA) bởi Corne và các cộng sự năm 2000 ([18] D. W. Corne, 2000), Region-based Selection in Evolutionary Multiobjective Optimization (PESA-II) bởi Corne và các cộng sự năm 2001 ([19] D. W. Corne, 2001), Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) bởi Deb và các cộng sự năm 2002 ([20] K. Deb, 2002), Rank-Density Based Genetic Algorithm (RDGA) bởi Lu và Yen năm 2003 ([21] H. Lu, 2003) và Dynamic Multi-Objective Evolutionary Algorithm (DMOEA) Yen và Lu năm 2003 ([22] G. G. Yen, 2003).

Điểm khác biệt giữa các giải thuật MOEA nằm ở cách gán độ thích nghi (fitness assignment), cách duy trì quần thể ưu tú (elitism) và các tiếp cận nhằm đa dạng hóa quần thể.

#### Xếp hạng Pareto

Đây là phương pháp dùng để gán độ thích nghi rất hay được sử dụng. Phương pháp này bao gồm việc gán thứ hạng 1 cho các cá thể không bị vượt trội trong quần thể và đưa chúng ra ngoài vòng xem xét; rồi tìm tập cá thể không bị vượt trội mới để gán thứ hạng 2 và tiếp tục như vậy.

#### Chia sẻ độ thích nghi (fitness sharing)

Đây là một trong những phương pháp dùng để đa dạng hóa quần thể. Phương pháp chia sẻ độ thích nghi khuyến khích tìm kiếm trên những vùng chưa được thăm dò của Pareto front bằng cách giảm bớt độ thích nghi của các lời giải ở những vùng cá thể mật độ cao. Kỹ thuật chia sẻ độ thích nghi với *số đếm vùng lân cận* (niche count) được mô tả như sau.

### Chia sẻ độ thích nghi dựa vào số đếm vùng lân cận.

Phương pháp này đòi hỏi phải giảm bớt độ thích nghi *fi* của một cá thể *i* bằng cách chia nó cho số đếm vùng lân cận *mi* được tính cho cá thể đó. Tức là độ thích nghi dùng chung được tính bằng *fi*/*mi*. Số đếm vùng lân cận *mi* là giá trị ước lượng vùng lân cận của cá thể *i* đông đúc như thế nào. Nó được tính cho từng cá thể trong quần thể hiện hành theo công thức:

*Sh*[*d*] là một hàm của *d*[*i*, *j*] sao cho: *Sh*[0] = 1 và *Sh*[*d* ≥ *σshare*] = 0.

Thông thường:

*σshare* là *bán kính vùng lân cận*, được người dùng xác định để ước lượng độ cách biệt tối thiểu mong muốn giữa hai lời giải cuối cùng. Các cá thể có khoảng cách trong phạm vi *σshare* bị giảm bớt độ thích nghi vì chúng ở trong cùng vùng lân cận.

#### Phương pháp dùng khoảng cách mật độ

Là một phương pháp đa dạng hóa quần thể không phải xác định thông số *σshare.* Phương pháp này đòi hỏi tính khoảng cách mật độ là giá trị ước lượng mật độ lời giải bao quanh một điểm được xét *i* trong quần thể. Đại lượng này là giá trị trung bình của hai điểm lấy hai bên của điểm được xét *i* dọc theo mỗi trục mục tiêu. Đại lượng này được dùng trong cơ chế chọn cha mẹ như sau: lấy ngẫu nhiên hai lời giải *x* và *y*; nếu chúng có cùng *mức không vượt trội* (non-domination rank) thì lời giải nào có khoảng cách mật độ cao hơn sẽ được chọn; ngược lại lời giải có mức không vượt trội thấp hơn sẽ được chọn.

Ngoài ra, việc duy trì quần thể ưu tú là một vấn đề quan trọng trong tối ưu hóa đa mục tiêu bằng giải thuật MOEA. Trong ngữ cảnh của giải thuật MOEA, tất cả những lời giải không bị vượt trội được phát hiện bởi MOEA được coi như là những *lời giải ưu tú*. Có hai chiến lược thường dùng để hiện thực việc duy trì quần thể ưu tú: (i) lưu trữ các lời giải ưu tú trong chính quần thể và (ii) lưu trữ các lời giải ưu tú trong một danh sách thứ cấp bên ngoài quần thể và đưa chúng trở lại quần thể.

#### Đặc điểm của giải thuật VEGA:

* Gán độ thích nghi: quần thể được phân thành K tiểu quần thể (K là số mục tiêu). Các cá thể trong mỗi tiểu quần thể được đánh giá theo một mục tiêu riêng.
* Cơ chế đa dạng hóa: không có.
* Cách duy trì quần thể ưu tú: không có.

#### Đặc điểm của giải thuật MOGA:

* Gán độ thích nghi: dùng cách xếp hạng Pareto (Pareto ranking).
* Cơ chế đa dạng hóa: chia sẻ độ thích nghi (fitness sharing) dùng số đếm vùng lân cận.
* Cách duy trì quần thể ưu tú: không có.

#### Đặc điểm của giải thuật NSGA:

* Gán độ thích nghi: xếp hạng dựa vào sắp thứ tự mức độ không vượt trội (non-domination sorting).
* Cơ chế đa dạng hóa: chia sẻ độ thích nghi (fitness sharing) dùng số đếm vùng lân cận.
* Cách duy trì quần thể ưu tú: không có

#### Đặc điểm của giải thuật NSGA-II:

* Gán độ thích nghi: xếp hạng dựa vào sắp thứ tự mức độ không vượt trội (non-domination sorting).
* Cơ chế đa dạng hóa: phương pháp dùng khoảng cách mật độ (crowding distance).
* Cách duy trì quần thể ưu tú: có.

#### Đặc điểm của giải thuật SPEA:

* Gán độ thích nghi: xếp hạng dựa vào kho lưu ngoài (external archive) của những lời giải không bị vượt trội.
* Cơ chế đa dạng hóa: gom cụm (clustering) để tỉa bớt quần thể ngoài (external population).
* Cách duy trì quần thể ưu tú: có.

#### Đặc điểm của giải thuật SPEA-2

* Gán độ thích nghi: dựa vào sức mạnh của các cá thể vượt trội (dominator).
* Cơ chế đa dạng hóa: dùng mật độ dựa vào láng giềng gần nhất thứ k.
* Cách duy trì quần thể ưu tú: có.

# CHƯƠNG 3. GIẢI QUYẾT BÀI TOÁN

|  |
| --- |
| Chương này là chương chính giải quyết bài toán đấu thầu nhiều vòng bao gồm các nội dung:   * Mô hình hóa bài toán đấu thầu nhiều vòng * Cách thức triển khai một vấn đề trên MOEA Framework * Triển khai bài toán đấu thầu nhiều vòng đã mô hình hóa |

## Mô hình hóa bài toán đấu thầu nhiều vòng theo lý thuyết trò chơi và cân bằng Nash

### Trò chơi đấu thầu nhiều vòng

Bài toán đàm phán đấu thầu nhiều vòng có thể được mô hình hóa thành một dạng trò chơi với thông tin đầy đủ được đại diện bởi các tập chiến lược như sau:

trong đó: So là tập các chiến lược của chủ đầu tư, Fo là lợi nhuận của chủ đầu tư, Sc là tập các chiến lược của nhà thầu, Fc là lợi nhuận của các nhà thầu.

Khi trò chơi bắt đầu tức là khi triển khai đấu thầu một dự án thì cả chủ thầu và các nhà thầu khi tham gia đều cố gắng thu lại lợi ích lớn nhất cho mình từ các gói thầu

Cân bằng Nash của trò chơi đấu thầu nhiều vòng đó là

* Chủ đầu tư không phải chi trả quá cao cho dự án mà vẫn nhận được các sản phẩm giá trị, các nhà thầu với mức giá hợp lý đều thu được lợi nhuận nhất định từ dự án.
* Các nhà thầu với mức giá hợp lý đều thu được lợi nhuận nhất định từ dự án và được đánh giá cao, là đối tác làm ăn lâu dài với các chủ dự án.

Các bên sẽ đưa ra tập chiến lược của mình và chủ đầu tư sẽ cố gắng tìm ra một lời giải đảm bảo đem lại lợi ích nhiều nhất cho bản thân nhưng lại không làm mất lòng đối phương.

### Đầu vào của bài toán:

* Dự án có một danh sách các sản phẩm cần mua và được phân chia thành các gói thầu theo kế hoạch của nhà đầu tư.
* Dự án sẽ diễn ra trong một khoảng thời gian nhất định. Trong thời gian đó dự án sẽ tổ chức một số giai đoạn đấu thầu, mỗi lần đấu thầu thì chủ đầu tư sẽ mua một gói thầu cần thiết cho dự án.
* Dự án có nhiều nhà thầu tham gia, trong đó có cả các nhà thầu tin cậy và nhà thầu mới tham gia.
* Mỗi nhà thầu có khả năng cung cấp một số mặt hàng theo khả năng của mình và có chiến lược kinh doanh riêng của mình ví dụ như chiến lược giảm giá.
* Không phải mỗi nhà thầu đều tham gia vào tất cả gói thầu. Chủ thầu sẽ có đánh giá cho từng nhà thầu về chất lượng nhà thầu cũng như quan hệ nhà thầu – chủ đầu tư.

### Đầu ra của bài toán:

Danh sách gói thầu với nhà thầu được lựa chọn, và thời điểm tổ chức gói thầu tương ứng

### Mục tiêu của bài toán

Chiến lược thuần túy của nhà thầu là phương án giảm giá đối với mỗi sản phẩm trong từng gói thầu. Phương án giảm giá của nhà thầu thay đổi theo từng giai đoạn, được các nhà thầu tính toán dựa trên giá gốc mà nhà thầu đã mua, giá bán niêm yết. Bằng cách đưa vào chiến lược giảm giá, các nhà thầu sẽ thu hút được nhà đầu tư hơn so với các đối thủ bởi giá niêm yết của các nhà thầu sẽ không chênh lệch nhau là mấy, tuy nhiên không thể giảm giá quá nhiều bởi có thể dẫn đến lỗ. Vì thế nhà thầu sẽ cân nhắc kĩ lưỡng việc giảm giá của từng gói thầu một cách hợp lý để vừa thu hút chủ đầu tư và vừa sinh được lời.

Đối với chủ đầu tư, việc chọn nhà thầu có giá bán, mức giảm giá hợp lý là một yếu tố, tuy nhiên sẽ là không nên chọn những nhà thầu kém chất lượng, không đảm bảo cho tiến độ dự án làm giảm uy tín của chủ đầu tư. Nhà đầu tư có thể dựa vào lịch sử của mỗi nhà thầu để đánh giá chất lượng của nhà thầu đó. Ngoài ra, nhà đầu tư cũng muốn lựa chọn các nhà thầu sao cho mức độ thân thiện với các nhà thầu là hợp lý để đảm bảo các mối làm ăn sau này.

Như vậy đối với mỗi lựa chọn của chủ đầu tư, sẽ có một công thức đánh giá lợi nhuận của chủ đầu tư dựa trên giá bán, mức giảm giá của các nhà thầu, đối với mỗi chủ đầu tư cũng sẽ có một công thức đánh giá lợi nhuận của nhà thầu khi tham gia vào dự án. Các lợi nhuận này sẽ được đánh giá max để thỏa mãn về mặt lợi nhuận của những người tham gia, để đảm bảo chất lượng dự án tổng chất lượng của các nhà thầu trúng thầu sẽ được đánh giá max, để đảm bảo mức độ thân thiện với các nhà thầu, mỗi nhà thầu sẽ có chỉ số đánh giá trước quan hệ với chủ đầu tư và công thức ở đây sẽ đánh giá theo kiểu “san sẻ” lợi nhuận và quan hệ với chủ đầu tư giữa các nhà thầu.

#### Lợi ích của chủ đầu tư

Giả sử chi phí dự kiến của dự án cho gói thầu thứ n là A, chi phí thanh toán thực tế cho gói thầu đó là B (thông thường A > B) thì phần chênh lệch A – B chính là lợi nhuận của chủ đầu tư trong gói thầu đó. Tính tổng lợi nhuận của chủ đầu tư trên tất cả những gói thầu ta thu được lợi nhuận của chủ đầu tư trong toàn dự án. Đối với một gói thầu bất kỳ, chi phí mà chủ đầu tư trả cho gói thầu hay chính là giá bán đề xuất của nhà thầu cho gói thầu được tính bởi công thức:

Trong đó, Pi (price) là giá bán của một đơn vị sản phẩm i, NUi (number of units) là số lượng đơn vị sản phẩm đó, di(t) là tỉ lệ chiết khấu của sản phẩm tại thời điểm t, Xn là số lượng sản phẩm của gói thầu thứ n. Bn được tính là tổng giá bán đề xuất cho tất cả các sản phẩm trong gói thầu thứ n. Chú ý rằng di(t) là một hàm thay đổi theo thời gian, một nhà thầu sẽ chiết khấu một loại sản phẩm với tỷ lệ nhất định tại các thời gian định trước, trong khi tại các thời điểm khác, mặt hàng sẽ không được chiết khấu. Đây là thông tin quan trọng ảnh hưởng tới chiến lược về thời điểm diễn ra gói thầu của chủ đầu tư.

Chủ đầu tư sẽ dựa vào chiến lược lựa chọn thời điểm diễn ra gói thầu và lựa chọn nhà thầu cho từng gói để tối thiểu hóa số tiền phải trả hay tối đa hóa lợi nhuận thu về. Dựa vào phân tích về số tiền phải trả cho một gói thầu trên, tổng lợi nhuận của chủ đầu tư sẽ được tính theo công thức:

Trong đó: N là số lượng gói thầu, là chi phí ước tính (Estimated Cost) của chủ đầu tư cho gói thầu n và là chi phí chủ đầu tư phải trả cho gói thầu thứ n, đã được phân tích trước đó, r là tỷ lệ lạm phát và S, t tương ứng là thời điểm bắt đầu dự án (coi như thời điểm tham chiếu giá trị tiền) và thời điểm gói thầu diễn ra. Thời điểm t là quan trọng vì ảnh hưởng trực tiếp tới số tiền chủ đầu tư phải trả cho gói thầu. Thành phần thể hiện việc quy chiếu giá trị của tiền về thời điểm ban đầu. Để có ước tính chính xác lợi nhuận, ta cần quan tâm tới giá trị của tiền vì nó thay đổi theo thời gian, 1 đồng tiền tại thời điểm hiện tại có giá trị hơn 1 đồng tiền ở thời điểm tương tai. Tại thời điểm t, giá trị của một đồng tiền chỉ bằng giá trị của nó tại thời điểm S nên Fo được tính bằng việc quy đổi lợi nhuận về thời điểm bắt đầu dự án.

#### Lợi ích của từng nhà thầu

Ta giả sử i là một nhà thầu bất kỳ tham gia vào đấu thầu một số gói của dự án. Nhà thầu này sẽ thực hiện các chiến lược nhất định để cực đại hóa lợi nhuận mà họ dự kiến thu được. Lợi nhuận dự kiến của nhà thầu này sẽ bằng tổng lợi nhuận thu được từ từng gói thầu mà họ tham gia. Với một gói thầu mà nhà thầu này tham gia, lợi nhuận sẽ bằng chênh lệch của giá bán đề xuất (hay chính là giá mà chủ đầu tư mua sản phẩm) với giá mà nhà thầu nhập các sản phẩm trong gói thầu đó:

Trong đó Mn là lợi nhuận nhà thầu thu được với gói thầu n, Pi là giá bán đề xuất của một đơn vị sản phẩm với chủ đầu tư, di(t) là tỷ lệ chiết khấu của một đơn vị sản phẩm đó theo thời gian t, Ri là giá nhập của đơn vị sản phẩm đó và NUi là số lượng đơn vị sản phẩm cần cung cấp trong gói thầu. Lợi nhuận dự kiến của một nhà thầu bất kỳ được tính bởi công thức:

Trong đó: N là số lượng gói thầu, thể hiện nhà thầu i có tham gia gói thầu n hay không, thể hiện lợi nhuận nhà thầu thu được với gói thầu này. Các thành phần r, t, S như trước tương ứng thể hiện tỷ lệ lạm phát (hoặc tỷ lệ mất giá đồng tiền), thời điểm gói thầu diễn ra và thời điểm bắt đầu dự án (coi như thời điểm tham chiếu giá trị đồng tiền).

#### Cân bằng lợi ích-quan hệ giữa các nhà thầu

Để thể hiện ảnh hưởng của mối quan hệ tới từng nhà thầu, sử dụng yếu tố cân bằng lợi ích-quan hệ giữa các nhà thầu bằng tổng hiệu số của tích lợi nhuận và chỉ số quan hệ giữa các nhà thầu và được tính theo công thức:

với P là số lượng nhà thầu, ai là hằng số thể hiện mối quan hệ giữa nhà thầu và chủ đầu tư. Khi C = 0 thì các nhà thầu đều thỏa mãn về lợi ích và yếu tố quan hệ khi tham gia dự án.

Cân bằng lợi ích-quan hệ là một yếu tố quan trọng trong tìm kiếm cân bằng Nash, bởi lẽ rõ ràng chiến lược về giá là chưa đủ để lựa chọn ra nhà thầu tốt. Ý tưởng của cân bằng lợi ích-quan hệ xuất phát từ ý tưởng của cân bằng Nash, tại thời mọi nhà thầu thỏa mãn với số tiền thu được và giữ được quan hệ nhất định. Trong thực tế, để có cân bằng này, nhà thầu sẽ đánh đổi một phần lợi nhuận mình có được để giữ quan hệ, nhưng vân giữ cho lợi ích-quan hệ không chênh nhau quá nhiều. Điểm cân bằng này cũng tương ứng với điểm mà C đạt giá trị nhỏ nhất.

#### Đảm bảo chất lượng cho dự án

Mặc dù ta đã xét đến yếu tố lợi nhuận chủ đầu tư, nhà thầu, và quan hệ nhà thầu-chủ đầu tư, sẽ là thiếu xót nếu không đề cập tới chất lượng sản phẩm cung cấp cho dự án. Chất lượng dự án là yếu tố quan trọng cuối cùng trong chiến lược của chủ dự án trong trò chơi đấu thầu nhiều vòng. Nếu bỏ qua yếu tố này khi đi tìm cân bằng Nash, chủ dự án có thể thua cuộc trong trò chơi: anh ta thu được nhiều lợi nhuận, nhưng chất lượng sản phẩm lại không đảm bảo.

Để giải quyết vấn đề này, mỗi nhà thầu được gán một chỉ số chất lượng q trong khoảng [0, 1], q là hằng số chuyên gia thể hiện cho uy tín của nhà thầu, gián tiếp thể hiện chất lượng sản phẩm họ cung cấp. Giá trị q càng lớn thể hiện nhà thầu chất lượng càng cao. Chủ đầu tư sẽ có chiến lược để tối đa hóa tổng giá trị chất lượng của các gói thầu, hay nói khác, tối đa hóa:

với N là số lượng gói thầu và là chất lượng của nhà thầu trúng gói thầu n, sẽ lấy giá trị trong tập chỉ số chất lượng của các nhà thầu tham gia vào thầu gói thầu n.

Như vậy, bài toán đấu thầu nhiều vòng là một trò chơi giữa chủ đầu tư và các nhà thầu, trong đó, chiến lược của chủ đầu tư là tập trung vào tối đa lợi nhuận của mình và tối đa chỉ số chất lượng cho dự án. Chiến lược của từng nhà thầu là tối đa lợi nhuận của chính mình nhưng vẫn đảm bảo tối thiểu cân bằng lợi ích-quan hệ giữa các nhà thầu. Cân bằng Nash là điểm mà chủ đầu tư và các nhà thầu đều cảm thấy chiến lược của mình được thỏa mãn.

## MOEA Framework:

### Giới thiệu

MOEA Framework là một thư viện Java mã nguồn mở, được sử dụng để phát triển và thử nghiệm các thuật toán tiến hóa đa mục tiêu (MOEAs) và các thuật toán tối ưu hóa đa mục tiêu. MOEA Framwork hỗ trợ các giải thuật di truyền, tiến hóa khác biệt, tối ưu hóa dòng hạt, lập trình di truyền và nhiều thuật toán tối ưu khác. MOEA Framework cũng cung cấp các công cụ cần thiết để nhanh chóng thiết kế, phát triển và thử nghiệm các thuật toán tối ưu hóa thống kê.

MOEA Framework là một giải pháp nhanh chóng đáng tin cậy cho người lập trình muốn triển khai các thuật toán tiến hóa đa mục tiêu như: NSGA-II, NSGA-III, εMOEA, εNSGA-II, PAES, PESA2, SPEA2, IBEA, SMS-EMOA, GDE3, SMPSO, OMOPSO, CMA-ES và MOEA/D. Các thuật toán này được tối hóa hiệu suất do đó có thể sử dụng cho các ứng dụng cần hiệu suất cao. Với việc hỗ trợ các thư viện JMET, PISA, MOEA Framework cung cấp quyền truy cập vào 30 thuật toán tối ưu hóa đa mục tiêu.

MOEA Framework cung cấp một bộ cơ sở các thuật toán, các vấn đề và toán tử tìm kiếm đồng thời cũng cung cấp khả năng mở rộng một cách dễ dàng. Người lập trình có thể sử dụng giao diện SPI để tích hợp các thuật toán, vấn đề mới vào trong MOEA Framewrok.

MOEA Framework có cấu trúc hướng đối tượng, cho phép kết hợp các thành phần hiện có để xây dựng các thuật toán tối ưu hóa mới. Với một chức năng cần thiết không được tìm thấy trên MOEA Framwork, người lập trình có thể mở rộng từ một lớp hiện có hoặc thêm một lớp mới.

MOEA Framwork là mã nguồn mở với nguồn tài liệu đầy đủ, được cập nhật thường xuyên. Có thể dễ dàng tìm kiếm và sử dụng thông qua mạng internet.

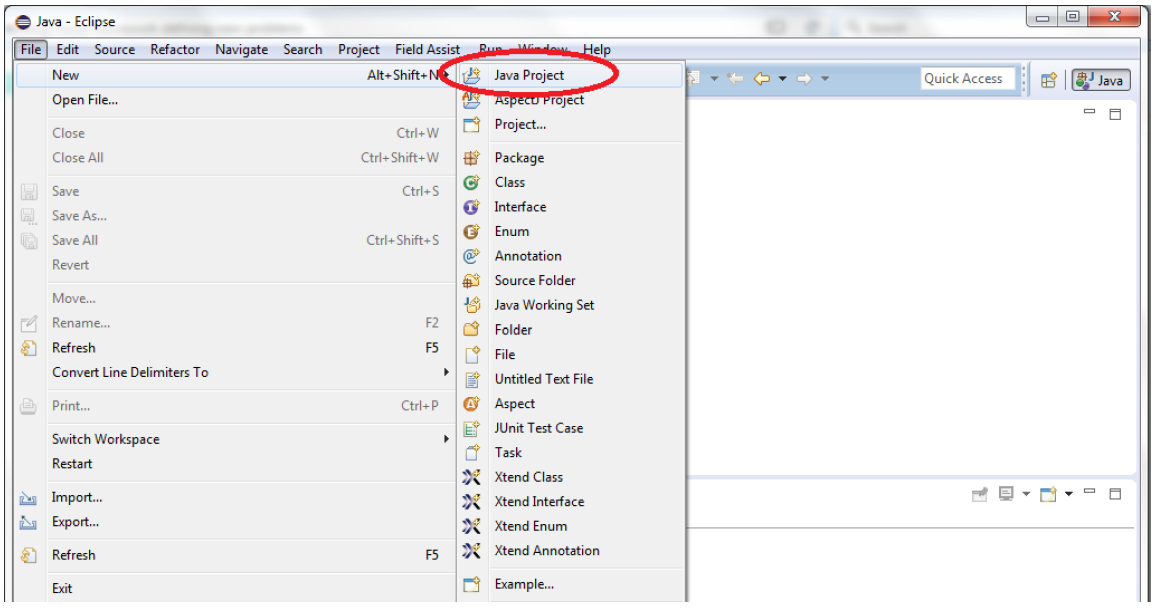
### Cấu hình sử dụng MOEA:

#### Các yếu tố phụ thuộc

MOEA Framework là mã nguồn mở chạy trên Java 6 trở đi. Ngoài việc cần java 6 trở đi, để có thể biên dịch MOEA Framework và phát triển các thuật toán máy tính cần cài đặt eclipse.

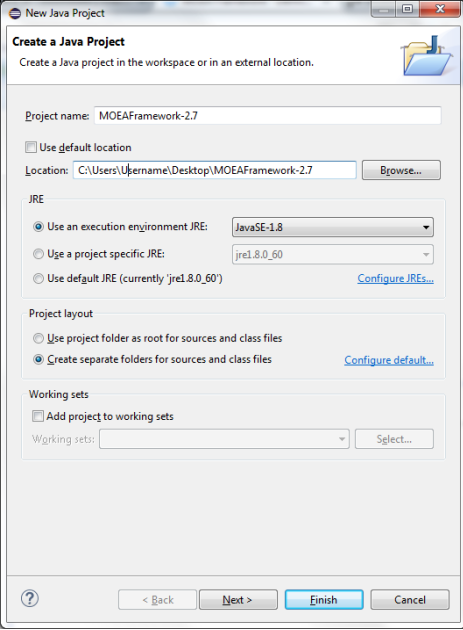
#### Cấu hình MOEA Framework

Tải xuống thư mục mã nguồn mở kèm với hướng dẫn của MOEA Framework theo link: <http://bit.ly/1N5sHjO>. Để sử dụng MOEA Framework có thể sử dụng trực tiếp mã nguồn mở hoặc biên dịch toàn bộ MOEA Framework thành một thư viện java (\*.jar). Cách thức sử dụng trực tiếp mã nguồn mở đó là tạo ra một Project mới trong Eclipse



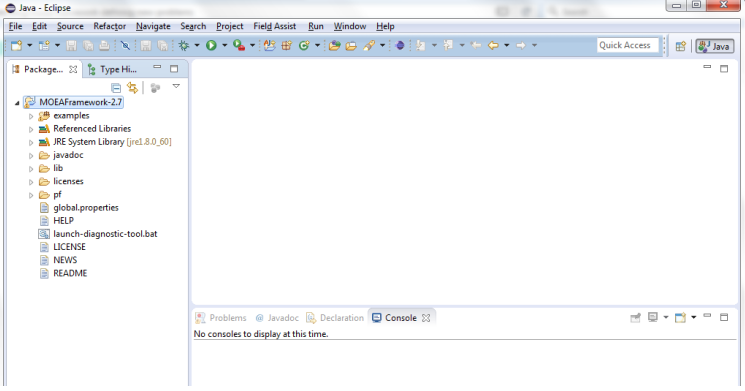
1. Hướng dẫn sử dụng MOEA Framework trên Eclipse 1

Trên cửa sổ “New Java Project” sẽ xuất hiện; bỏ chọn “Use default locatioin”; Chọn “Browse…”; sau đó chọn tới đường dẫn chính xác của thư mục MOEA Framework được giải nén từ tệp tin tải xuống ở bước trên.



1. Hướng dẫn sử dụng MOEA Framework trên eclipse 2

Đã tạo xong MOEA Framework chứa toàn bộ mã nguồn mở đã được phát triển. Hiện tại có thể biên dịch MOEA Framework thành thư viện hoặc tạo các lớp riêng biệt để triển khai các bài toán, tùy chỉnh các thuật toán đã có sẵn trong framework.



1. Hướng dẫn sử dụng MOEA Framework trên eclipse 3

## Triển khai bài toán đấu thầu nhiều vòng trên MOEA Framework

Mỗi một bài toán thực tế khi triển khai trên MOEA Framework sẽ tương ứng với một Problem, với bài toán đấu thầu nhiều vòng ta sẽ có BIDProblem; lớp BIDProblem kế thừa lớp AbstractProblem được thiết kế sẵn trong MOEA Framework

Như đã mô hình hóa ở trên đầu vào của bài toán cụ thể sẽ là:

* Các thông tin chung của dự án: tên, thời gian bắt đầu
* Thông tin của các gói thấu: tên, khoảng thời gian diễn ra, các nhà thầu có thể tham gia, các sản phẩm cần mua trong gói thầu, tổng số tiền dự kiến của gói thầu
* Thông tin của các nhà thầu: tên, các sản phẩm có thể cung cấp sẽ kèm theo các thông tin giá nhập của sản phẩm, giá bán của sản phẩm, thời gian và tỷ lệ giảm giá
* Ngoài ra sẽ có các thông tin liên quan: tỷ lệ lạm phát, chất lượng của từng nhà thầu, đánh giá mối quan hệ của nhà thầu với chủ đầu tư.

Tương ứng với đầu vào trên các đối tượng sẽ được triển khai bao gồm:

ProjectBID: Là lớp trừu tượng khái quát hóa các dự án của chủ đầu tư chứa các thông tin: ID, inflationRate, startDate, packages, contractors, products tương ứng mô tả ID, tỷ lệ lạm phát, danh sách các gói thầu, danh sách các nhà thầu tham gia dự án, danh sách tất cả các sản phẩm cần mua của dự án.

Package: Là lớp trừu tượng khái quát hóa các gói thầu chứa các thông tin: ID, from, to, joinedContractors, products, estimatedCost tương ứng mô tả ID, khoảng thời gian có thể diễn ra từ from đến to, các nhà thầu có thể tham gia, các sản phẩm có thông tin về số lượng cần mua (là một danh sách các đối tượng của lớp trừu tượng ProductOfPackage ), tổng chi phí dự tính.

Contractor: Là lớp trừu tượng khái quát hóa các nhà thầu tham gia trong dự án chứa các thông tin: ID, description, quality, relationship, products tương ứng mô tả ID, tên, hệ số nói lên chất lượng của nhà thầu, hệ số nói lên mối quan hệ của nhà thầu với chủ đầu tư, danh sách sản phẩm do nhà thầu cung cấp (là một danh sách các đối tượng của lớp trừu tượng ProductOfContractor).

ProductOfPackage: Là lớp trừu tượng khái quát hóa thông tin sản phẩm của một gói thầu: productID, product, quantity tương ứng cho biết ID của sản phẩm, sản phẩm tương ứng trong danh sách sản phẩm của dự án (là một đối tượng của lớp trừu tượng Product), số lượng cần mua trong gói thầu.

ProductOfContractor: Là lớp trừu tượng khái quát hóa thông tin cung cấp sản phẩm của một nhà thầu: productID, product, sellPrice, buyPrice, discount tương ứng cho biết ID của sản phẩm, sản phẩm tương ứng trong danh sách sản phẩm của dự án (là một đối tượng của lớp trừu tượng Product), giá bán, giá nhập, danh sách các khoảng thời gian giảm giá và hệ số giảm giá tương ứng (là một danh sách các đối tượng của lớp trừu tượng Discount)

Product: Là lớp trừu tượng khái quát hóa thông tin sản phẩm của dự án: ID, Name, Description, Unit tương ứng cho biết ID, tên, mô tả và đơn vị của sản phẩm trong dự án.

Discount: Là lớp trừu tượng khái quát hóa thông tin giảm giá sản phẩm của nhà thầu: from, to, rate cho biết khoảng thời gian giảm giá chỉ từ from đến to với hệ số giảm giá là rate.

Để có thể triển khai lớp BIDProblem cần cung cấp thông tin về số lượng biến tham gia vào quá trình tối ưu hóa, các mục tiêu của việc tối ưu hóa. Tương ứng việc cung cấp các thông tin này là việc cần viết lại hai hàm ghi đè trong lớp đó là newSolution() và evaluate().

Hàm newSolution phía dưới chỉ rõ số lượng biến của bái toán là bằng 2 lần số lượng gói thầu của dự án. Lý do xác định được là do bài toán phải đưa ra danh sách các nhà thầu được chọn và ấn định thời gian tổ chức thầu đối với từng gói thầu trong dự án. Với mỗi gói thầu sẽ có danh sách các nhà thầu có thể tham gia, hoàn toàn có thể xác định nhà thầu được chọn dựa theo thứ tự trong danh sách này. Do đó biến đầu tiên của một gói thầu là số nguyên thể hiển vị trí của nhà thầu được chọn trong danh sách các nhà thầu tham gia. Khoảng giá trị của biến là từ 0 đến độ dài danh sách nhà thầu có thể tham gia vào gói thầu. Biến thứ hai sẽ thể hiện gói thầu được thực hiện vào thời điểm nào; là biến số nguyên có giá trị từ 0 đến hiệu số của ngày cuối cùng và ngày đầu tiên của gói thầu. Khi có giá trị của biến thứ hai này ta xác định bằng cách lấy ngày đầu tiên cộng với giá trị biến sẽ ra ngày tổ chức thầu.

|  |
| --- |
| @Override  **public** Solution newSolution() {  // **TODO** Auto-generated method stub  Solution solution = **new** Solution(\_projectBID.Packages().size() \* 2, \_numberOfObj);  **for**(**int** i =0; i < \_projectBID.Packages().size(); i++) {  solution.setVariable(i \*2, EncodingUtils.*newInt*(0, \_projectBID.Packages().get(i).joinedContractors().size() -1));  solution.setVariable(i\*2+1, EncodingUtils.*newInt*(0, DateUtil.*diff*(\_projectBID.Packages().get(i).to(), \_projectBID.Packages().get(i).from())));  }  **return** solution;  } |

Hàm evaluate() chỉ rõ các hàm mục tiêu của BIDProblem. Đúng theo mô hình đã được xác định, BIDProblem có bốn mục tiêu là:

Lợi nhuận của chủ đầu tư tương ứng là hàm ProjectProfit

Lợi nhuận của tất cả các nhà thầu tương ứng là hàm ContractorProfit

Cân bằng lợi ích giữa các nhà thầu tương ứng là KeepContractorRelationship

Chất lượng của dự án tương ứng là hàm ProjectQuality

Mỗi hàm đều có được tính theo công thức đã nêu khi phân tích bài toán, xác định mô hình.

|  |
| --- |
| @Override  **public** **void** evaluate(Solution solution) {  // **TODO** Auto-generated method stub  //Danh sach nha thau duoc chon  **int**[] x = **new** **int**[\_projectBID.Packages().size()];  //Danh sach thoi gian duoc chon  **int**[] y = **new** **int**[\_projectBID.Packages().size()];      **for** (**int** i = 0; i < \_projectBID.Packages().size(); i++) {  x[i] = EncodingUtils.*getInt*(solution.getVariable(i\*2));  y[i] = EncodingUtils.*getInt*(solution.getVariable(i\*2 + 1));  }  // Danh sach tat cac cac objective co san  \_availableObjective[0] = -ProjectProfit(x,y);  \_availableObjective[1] = -ContractorProfit(x,y);  \_availableObjective[2] = KeepContractorRelationship(x,y);  \_availableObjective[3] = -ProjectQuality(x,y);  //se add toan bo cac object thanh ham vao day  **for** (**int** i = 0; i < numberOfObjective(); i++) {  solution.setObjective(i, \_availableObjective[i]);  }  } |

Sau khi tạo thành công BIDProblem, việc triển khai chạy các thuật toán sẽ sử dụng Executor. Executor sẽ được gán xử lý BIDProblem với hàm withProblemClass; việc thay đổi các thuật toán sẽ sử dụng hàm withAlgorithm. Hiện nay, Executor có thể thực hiện lời gọi đến 30 thuật toán chỉ bằng cách thay đổi tham số là tên gọi của thuật toán.

|  |
| --- |
| Executor executor = **new** Executor()  .withProblemClass(BIDProblem.**class**, projectBID, 4)  .withAlgorithm("NSGAIII")  .withMaxEvaluations(10000); |

Executor sau khi hoàn thành nhiệm vụ sẽ cho ra kết quả, có thể thực hiện so sánh kết quả nhận được từ các lần thực hiện một thuật toán hoặc các thuật toán khác nhau thông qua Analyzer được tích hợp sẵn trong MOEA Framework.

|  |
| --- |
| Analyzer analyzer = **new** Analyzer()  .withSameProblemAs(executor)  .includeHypervolume()  .showStatisticalSignificance();  analyzer.addAll("NSGAIII", executor.withAlgorithm("NSGAIII").runSeeds(1));  analyzer.printAnalysis();  analyzer.getAnalysis().print();  **new** Plot().add(analyzer).show(); |

# CHƯƠNG 4. Thử nghiệm và đánh giá

|  |
| --- |
| Chương này chứa những kết quả chính của luận văn mô tả các nội dung sau:   * Mô tả tập dữ liệu thử nghiệm * Mô tả các thuật toán thử nghiệm * Mô tả chương trình thử nghiệm đã xây dựng * Đưa ra kết quả thực thi các thuật toán |

## Dữ liệu thử nghiệm:

Dữ liệu thử nghiệm của luận án gồm hai tập được đặt trong thư mục data đính kèm với chương trình thử nghiệm là data1.json và data2.json. Hai tập có cấu trúc như nhau mô tả về hai dự án.

Dự án trong tập dữ liệu thử nghiệm thử nhất bắt đầu diễn ra từ ngày 1/1/2017, thực hiện mua sắm 43 loại sản phẩm, việc mua sắm được chia ra thành 6 gói thầu, quá trình tổ chức đấu thầu có 5 nhà thầu tham gia. Dự án trong tập dữ liệu thử nghiệm thứ hai bắt đầu diễn ra từ ngày 1/1/2017, thực hiện mua sắm 42 loại sản phẩm, việc mua sắm được chia ra thành 3 gói thầu, quá trình tổ chức đấu thầu có 4 nhà thầu tham gia. Thông tin về các sản phẩm, nhà thầu và gói thầu được mô tả đại diện theo các bảng dưới đây:

Thông tin của một sản phẩm cần mua của dự án bao gồm:

|  |  |  |
| --- | --- | --- |
| **ID** | **Mô tả** | **Đơn vị** |
| 0 | Máy trạm (PC) | Bộ |
| … | | |
| 42 | Đào tạo hướng dẫn sử dụng, vận chuyển và nhân công lắp đặt | Gói |

###### Mô tả đại diện thông tin một sản phẩm

Thông tin của một nhà thầu bao gồm:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Thông tin chung** | **Sản phẩm** | **Giá bán** | **Giá nhập** | **Giảm giá** | | |
| **Từ** | **Đến** | **Tỷ lệ** |
| 0 | Công ty Cổ phần tin học Hồng Phúc; Q=0.45 R=0.78 | 0 Máy trạm PC | 11388 | 9705 | 1/1/2017 | 8/1/2017 | 0.01 |
| 8/1/2017 | 8/3/2017 | 0.12 |
| 8/3/2017 | 16/3/2017 | 0.05 |
| 16/3/2017 | 5/7/2017 | 0.02 |
| … | | | | | |

###### Mô tả đại diện thông tin một nhà thầu

Thông tin của một gói thầu bao gồm:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Thời gian | Nhà thầu tham gia | Sản phẩm cần mua | | Giá dự kiến |
| ID | Số lượng |
| 0 | 7/1/2017 - 20/2/2017 | 1,0,4,2,3 | 16 | 2 | 3114320 |
| … | |

(Trong đó, nhà thầu tham gia là một danh sách mã các nhà thầu không theo thứ tự)

###### Mô tả đại diện thông tin một gói thầu

Ngoài ra, tập dữ liệu thử nghiệm còn cung cáp thông tin về việc trượt giá, lạm phát được thể hiện bằng tham số inflation\_rate= 0.01

## Các thuật toán thử nghiệm

### Thuật toán NSGA-III ([26] Guillermo Campos Ciro, 2016)

Thuật toán **NSGA-III** được phát triển trên cở sở của **NSGA** và **NSGA-II**.

**NSGA** là thuật toán di truyền sử dụng khái niệm dominate trong quá trình đánh giá hàm thích nghi của các cá thể trong quần thể. Đại khái ta hiểu NSGA dùng khái niệm vừa nêu trên để xếp hạng các cá thể, sau đó tìm ra tập Pareto Front. Ngoài tiêu chí ở trên thì thật ra NSGA còn dùng 1 tiêu chí nữa là độ bao phủ, lời giải nào có độ bao phủ càng lớn (càng cách xa những lời giải khác) thì ưu tiên càng cao. Điều này là tiêu chí phụ đảm bảo cho tập lời giải đa dạng chứ không bị co cụm.

Tổng quan các bước của **NSGA:**

* Sinh tập cá thể ngẫu nhiên.
* Lai ghép và đột biến.
* Xếp hạng rồi chọn lọc.
* Nếu thỏa mãn thì dừng lại, chưa thì lại bắt đầu quay lại bước 2.

**NSGA-II** thì cải thiện hơn NSGA ở bước phân lớp Pareto Front.

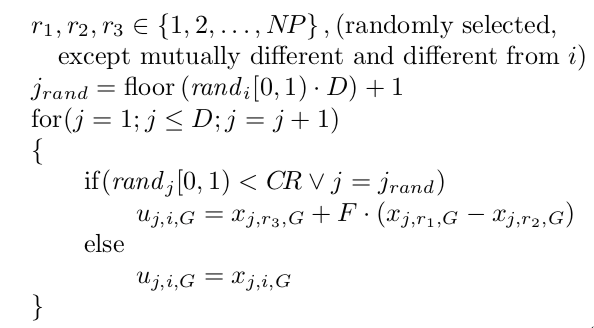
**NSGA-III** thì các bước từ đầu đến lai ghép và đột biến y hệt như NSGA-II tuy nhiên có khác ở bước chọn lọc:

* Đầu tiên xác định số **reference point** (hiểu là số cá thể trong quần thể).
* Sau khi lai ghép đột biến để sinh ra con thì tìm giá trị nhỏ nhất của từng **Objective Function (OF)**, sau đót trừ tất cả giá trị OF khác cho giá trị nhỏ nhất tìm được này (hiểu 1 cách hình học là kéo các trục tọa độ vào điểm sát nhất).
* Mỗi cá thể sẽ cho ra 1 kết quả tương ứng với từng hàm mục tiêu -> chuyển hóa thành 1 điểm trên mỗi trục tọa độ -> gộp các điểm này được 1 điểm trên cả hệ trục đại diện cho lời giải.
* Lập ra **hyper plan** với các reference point dựa vào các điểm vừa được kéo vào trục tọa độ. Các refernce point được chia đều ra trên mặt phẳng reference.
* Với tất cả điểm còn lại, xác định xem gần trục tọa độ nào nhất, sau khi đã xác định được, vẽ 1 đường vuông góc từ điểm này đến trục tọa độ đấy.
* Xác định điểm reference point nào gần với đường vuông góc đó nhất thì sẽ gắn điểm đó với điểm đang xét.
* Tiếp tục làm như vậy, nếu có nhiều hơn 1 điểm gắn với cùng 1 **reference point** thì sẽ lấy điểm nào gần hơn.
* Cuối cùng ta sẽ lấy ra được các điểm cho quá trình di truyền tiếp theo.

### Thuật toán GDE3 ([25] Saku Kukkonen - Kanpur Genetic Algorithms Laboratory, 2005)

Thuật toán **GDE3** cũng dựa trên giải thuật di truyền tuy nhiên lại không có lai ghép mà chỉ có đột biến và chọn lọc. Tư tưởng của **GDE** thuật toán là đột biến cá thể x thành cá thể x’ theo 1 quy tắc nhất định. Sau đó so sánh x với x’ xem cái nào tốt hơn thì lấy. **GDE2** và **GDE3** sau đó phát triển dựa trên nền tảng của **GDE**.

Công thức đột biến của **GDE** là:



Trong đó **F** và **CR** là 2 biến điều khiển (do ta chọn trước). Đoạn **j = jrand** là để đảm bảo sau 1 thế hệ thì có ít nhất 1 cá thể được đột biến.

Nhận thấy **GDE** quá nhạy cảm trong việc chọn biến điều khiển (**F** và **CR**) nên **GDE2** cải tiến bằng việc ra quyết định dựa trên **Crowdedness** khi mà x và x’ không **dominate** được nhau, nâng cao độ đa dạng lời giải nhưng lại làm chậm độ hội tụ của toàn quần thể vì nó ưu tiên các cá thể biên.

**GDE3** mở rộng phương thức đột biến với **M Objectives** và **K Constraints**.

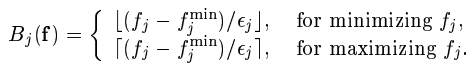
Quy tắc selection của **GDE3**:

* Cả 2 không khả thi: cái mới được chọn nếu violate constraints ít hơn cái cũ.
* 1 cái khả thi và 1 không khả thi thì cái khả thi được chọn.
* Cả 2 khả thi: cái mới được chọn nếu weakly dominate cái cũ ở hàm mục tiêu. Cái cũ được chọn nếu dominate cái cũ. Chọn cả hai nếu không cái nào dominate cái nào.
* Sau 1 thế hệ, số lượng cá thể có thể tăng lên. Nếu vào trường hợp này thì sẽ giảm size quần thể giống **NSGA-II** (dựa trên **non-dominance** & **crowdedness**).

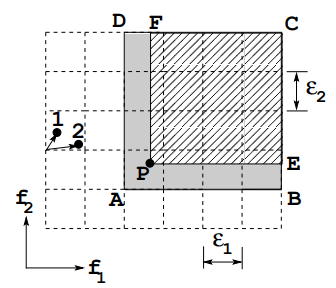
### Thuật toán ε-MOEA ([24] Kalyanmoy Deb, 2003)

#### Khái niệm ε-dominate

**ε-dominate** là một khái niệm tổng quát của **dominate**. Gán cho mỗi lời giải một vecto đặc trưng B = (B1, B2, …, Bm) với *m* là số mục tiêu cần tối ưu, trong đó:



Ở đây, có thể coi như chia khoảng giá trị của các hàm mục tiêu *fj* thành các đoạn liên tiếp có độ dài *εj*, kết quả ta thu được một hyper-boxes. Trong trường hợp *m=2*, ta thu được một lưới trên mặt phẳng toạ độ (hai chiều ứng với hai hàm mục tiêu) như hình dưới đây:



1. Biểu diễn các lời giải trên không gian mục tiêu

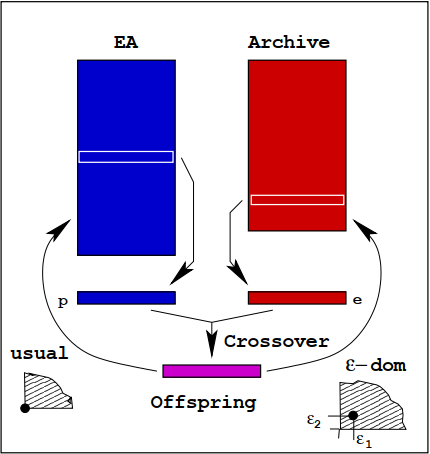
Các điểm A, B, C, D, E, F, P biểu diễn biến quyết định có giá trị hàm mục tiêu là toạ độ tương ứng. Trong trường hợp tối ưu hoá min cả *f1*và *f2*, theo định nghĩa **dominate** thông thường, dễ thấy, P dominate các điểm trong hình chữ nhật PECF. Tuy nhiên, trong khái niệm **ε-dominate**,điểm A sẽ được dùng để đánh giá ε-dominate cho điểm P, tức là với trường hợp này điểm P ε-dominate các điểm trong hình chữ nhật ABCD.

Với định nghĩa như trên, ta thấy, hai điểm có thể ε-dominate lẫn nhau khi cùng nằm trong một ô vuông của lưới (có cùng vecto đặc trưng). Đồng thời, khi các giá trị *εj* càng tiến đến 0 thì ε-dominate càng gần với dominate thông thường.

#### Chi tiết thuật toán ε-MOEA

Thuật toán **ε-MOEA** là một giải thuật di truyền, tức mỗi lời giải được coi như một cá thể, tập lời giải là quần thể, hai lời giải (cha, mẹ) có thể kết hợp (lai) với nhau để tạo thành lời giải mới (cá thể con). Thuật toán ε-MOEA gồm các bước:

* Bước 1: Khởi tạo ngẫu nhiên quần thể ban đầu P(0)
* Bước 2: Chọn các các thể tốt nhất trong P(0) đưa vào quần thể E(0) (quần thể lưu trữ/bảo tồn)
* Bước 3: Chọn ngẫu nhiên một cá thể trong mỗi quần thể P(0), E(0), lai ghép với nhau tạo thành cá thể con *c* (trong trường hợp tổng quát có thể tạo ra nhiều cá thể con *ci*)
* Bước 4 (chọn lọc): với mỗi cá thể con được tạo ra, quyết định giữ lại hay không dựa vào khái niệm **ε-dominate**.
* Bước 5: Dừng lại nếu thoả mãn điều kiện dừng (số vòng lặp tối đa, quần thể đủ tốt, không thay đổi, …) hoặc quay lại bước 3 nếu không thoả mãn.



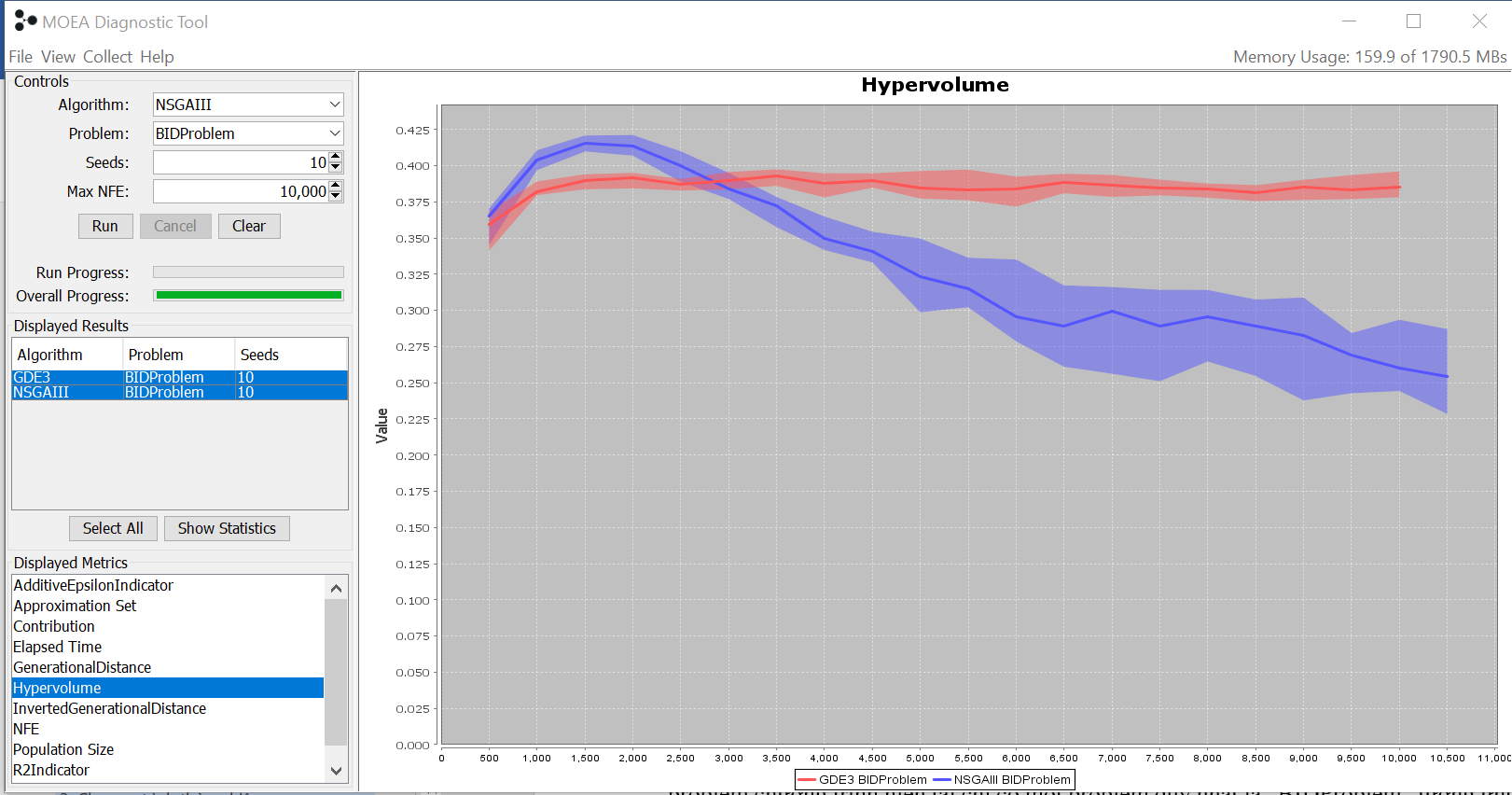
1. Biểu diễn quá trình chọn lọc trong thuật toán ε-MOEA

Việc quyết định giữ lại cá thể con *c* trong P (hoặc E) hay không dựa trên khái niệm **ε-dominate** đã trình bày ở trên theo quy tắc:

* Nếu tất cả cá thể trong quần thể đều ε-dominate *c* thì bỏ qua *c.*
* Nếu có ít nhất 1 cá thể bị dominate bởi *c* thì giữ lại *c* và loại bỏ 1 cá thể ngẫu nhiên trong số các cá thể đó.
* Nếu không xảy ra 1 trong 2 trường hợp trên:
  + Nếu 1 cá thể thuộc quần thể có cùng vecto đặc trưng với *c* thì chọn giữ lại cá thể nào có giá trị hàm mục tiêu gần giống vecto đặc trưng hơn (theo khoảng cách Euclide)
  + Ngược lại, giữ lại *c* và loại bỏ một cá thể ngẫu nhiên trong quần thể

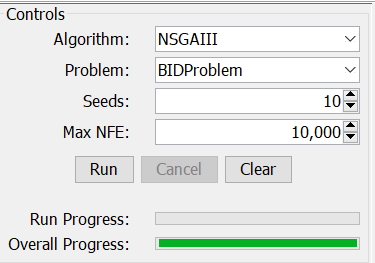
## Chương trình thử nghiệm

Chương trình được xây dựng bằng ngôn ngữ java dựa trên MOEA Framework. Giao diện chính của chương trình với 1/4 bên trái là các lựa chọn cho người sử dụng, 3/4 bên phải là biểu đồ so sánh các giá trị tham số đánh giá của các thuật toán.



1. Giao diện chương trình xây dựng

Chương trình cho phép lựa chọn các thuật toán thông qua combobox “Algorithm”; với problem chương trình hiện tại chỉ có một problem duy nhất là “BIDProblem” tương ứng là bài toán đấu thầu nhiều vòng đã được triển khai trên MOEA Framework; Seeds là số lần chạy cho một thuật toán; NFE là số lần đánh giá một hàm mục tiêu nhất định (là điều kiện dừng của thuật toán)



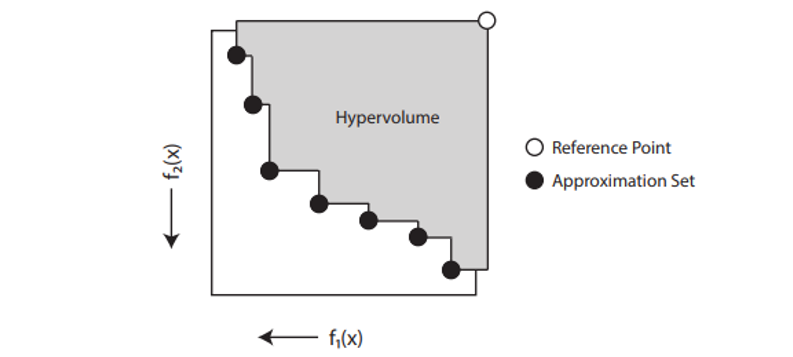
1. Giao diện chức năng lựa chọn thuật toán và các tham số

## Đánh giá các thuật toán

### Các tham số đánh giá

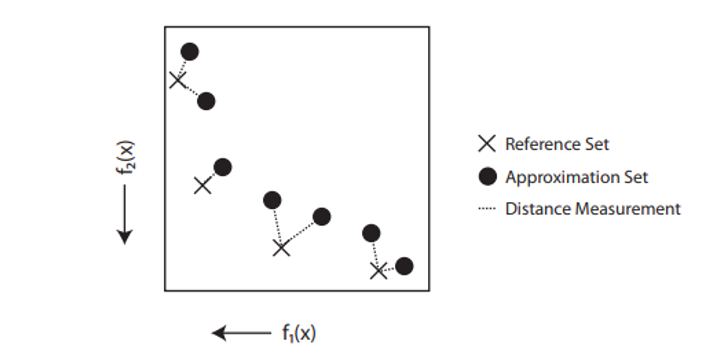
Quá trình các thuật toán MOEA thực hiện, sẽ đưa ra tập xấp xỉ của tập Pareto hoặc tập Pareto front lý do là vì việc tìm chính xác tập Pareto nhiều khi là cực kỳ tốn kém, nên điều kiện dừng chỉ thực hiện ở việc xem xét một hoặc một số mục tiêu nhất định tùy vào từng thuật toán. Việc đánh giá chất lượng các thuật toán được thực hiện thông quan các tập xấp xỉ tìm được của thuật toán đó. Do các MOEA khác nhau sẽ có xu hướng hoạt động tốt hơn trong các số liệu khác nhau ([04] Bosman, 2003), Deb và Jain ([05] Deb, 2002) đã để nghị sử dụng hai mục tiêu để so sánh các MOEA đó là proximity (sự gần gũi) và diversity (sự đa dạng). MOEA Framework cung cấp sẵn các tham số; các tham số đó được tính thông qua một tập tham chiếu. Tập tham chiếu được tạo ra từ trước khi thực hiện so sánh các thuật toán .

Hypervolumn: Là đại lượng được tính cho một tập Pareto front (với các giải pháp được tạo ra từ MOEA Framework từ nay gọi là tập xấp xỉ), cho biết thể tích của không gian bị chi phối (dominated) bởi tập xấp xỉ đang xem xét ([06] Knowles, 2002). Thể tích này càng lớn chứng tỏ tập giải pháp hiện tại vượt trội hơn càng nhiều các giải pháp khác. Xem xét hai thuật toán, thuật toán nào cho giá trị HyperVolumn cao hơn chứng to thuật toán đó tìm ra giải pháp tốt hơn.



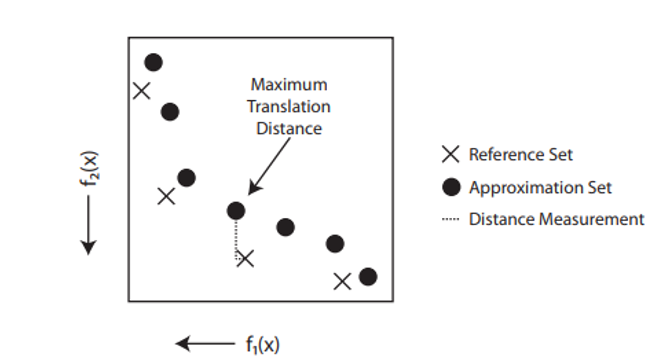
1. Khoảng không gian Hypervolumn

Generational Distance (GD): là khoảng cách trung bình từ mọi giải pháp trong tập gần đúng đến giải pháp gần nhất trong tập tham chiếu. Như vậy nó đo sự gần gũi của tập gần đúng tới tập tham chiếu. ([07] Hadka, 2012)



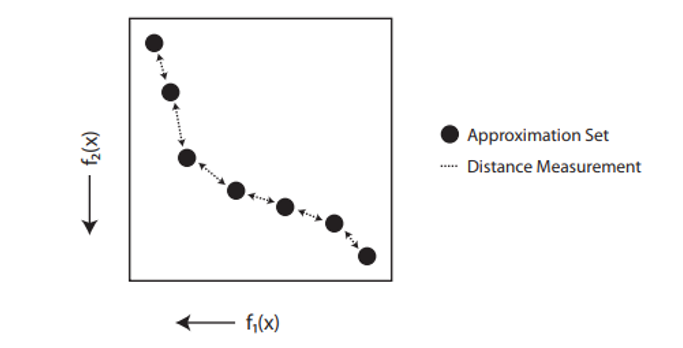
1. Mô tả Generation Distance

ε+-Indicator: Đo khoảng cách nhỏ nhất mà tập xấp xỉ phải được dịch để dominated hoàn toàn tập tham chiếu. Giá trị này thấp cho thấy sự gần gữi tốt ở tập xấp xỉ. ([02] Coello Coello, 2007)



1. *Mô tả* ε+-Indicator

Spacing: Đo lường sự đồng nhất của khoảng cách giữa các giải pháp trong một tập xấp xỉ. Một tập xấp xỉ tốt sẽ không chứa các cụm giải pháp dày đặc được phân tách bằng các khoảng trống lớn. Spacing không liên quan đến tập tham chiếu nên có những tập xấp xỉ sẽ cho spacing tốt nhưng tập đó không gần với tập tham chiếu. Do đó, để đánh giá một tập xấp xỉ cần sử dụng phối hợp nhiều tham số. ([02] Coello Coello, 2007)



1. Mô tả Spacing

Population Size: Cho biết số lượng giải pháp trong một tập xấp xỉ. Xem xét hai thuật toán, thuật toán nào cho PopulationSize lớn chứng tỏ khả năng duy trì dân số của thuật toán đó tốt hơn.

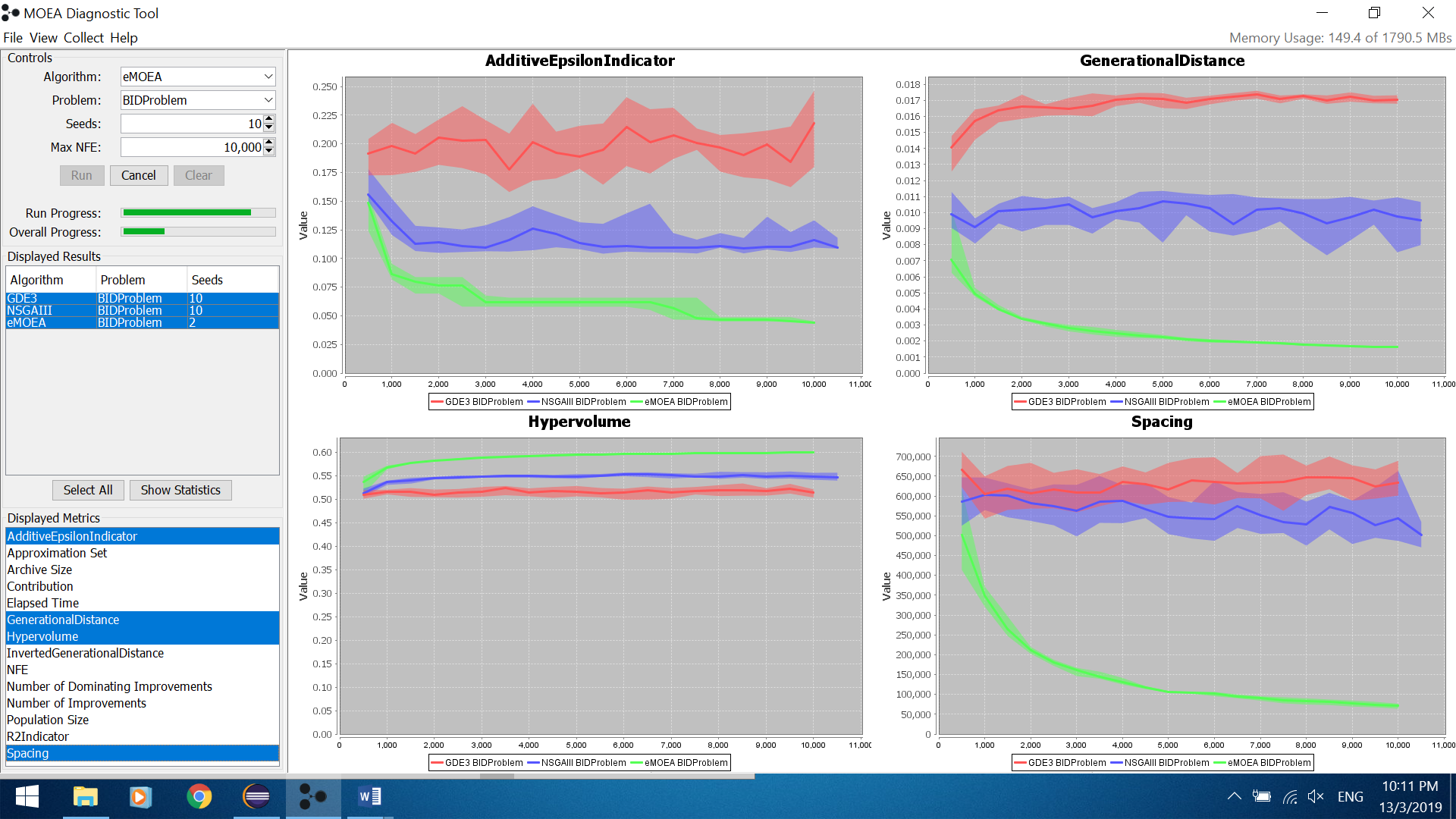
Trong MOEA Framework, để việc so sánh các thuật toán được nhất quán, các tham số Hypervolumn, ε+-Indicator, Generation Distance được chuẩn hóa thành các giá trị nằm trong khoàng [0, 1] với 1 đại diện cho giá trị tối ưu (tức là càng gần 1 càng biểu đạt tập xấp xỉ Paretor Front đang xét càng tốt). Để so sánh, đầu tiên Reference Set được chuẩn hóa theo giới hạn tối thiểu và tối đa của chính nó sao cho tất cả các điểm trong tập tham chiếu nằm trong [0, 1]N. Thứ hai, mỗi bộ xấp xỉ Pareto front được chuẩn hóa bằng các giới hạn khác nhau. Thứ ba, các tham số so sánh được tính bằng cách sử dụng các bộ dữ liệu đã chuẩn hóa. Cuối cùng, các tham số so sánh được chuyển đổi theo các phương trình khác nhau để đảm bảo giá trị 1 biểu thị giá trị tối ưu có thể đạt được theo số liệu. Trong đó,

HyperVolumn được chuyển đổi theo công thức: ([01] Hadka, 2016)

ε+-Indicator, Generation Distance được chuẩn hóa theo công thức: ([01] Hadka, 2016)

### Kết quả cụ thể:

Sử dụng chương trình phân tích, chạy tập dữ liệu data1.json các thuật toán NSGAIII, GDE3, eMOEA số lần chạy 10 lần cho mỗi thuật toán, được kết quả so sánh như hình:



1. Biểu đồ so sánh 03 thuật toán trên tập dữ liệu data1.json

Trên biểu đồ cho thấy Hypervolumn của ε-MOEA cao nhất vào khoảng 0.6, hai thuật toán NSGAIII, GDE3 lần lượt ở mức 0.55 và 0.52; tuy nhiên với hai tham số ε+-Indicator, Generation Distance ε-MOEA cho kết quả thấp nhất tiếp theo là NSGAIII và GDE3. Như vậy có thể thấy với tập dữ liệu data1.json ε-MOEA có sự đa dạng tốt (không gian bị dominated bởi tập xấp xỉ Pareto front rộng) nhưng chưa gần với tập xấp xỉ; ngược lại GDE3 gần với tập xấp xỉ nhất nhưng sự đa dạng lại không cao. Một vấn đề rất đáng lưu ý đó là ε-MOEA có thời gian thực thi cao hơn trong khi NGSAIII và GDE3 không nhiều tuy nhiên việc tính toán các tham số so sánh cho ε-MOEA lại tốn rất nhiều thời gian. Do vậy, khi thực hiện trong chương trình phân tích tổng thời gian từ khi bắt đầu đến khi hiển thị kết quả của ε-MOEA là chênh lệch vô cùng lớn so với NGSAIII và GDE3. Một số kết quả cụ thể được thể hiện như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tham số** | **NSGAIII** | **GDE3** | **eMOEA** |
| Thời gian chạy (T) | 37(s) | 34(s) | 39(s) |
| Hypervolume (HV) | 0.411 | 0.438 | 0.477 |
| GenerationalDistance (GD) | 0.002 | 0.001 | 0.000 |
| InvertedGenerationalDistance (IGD) | 0.107 | 0.058 | 0.000 |
| AdditiveEpsilonIndicator (AEI) | 0.162 | 0.105 | 0.034 |
| MaximumParetoFrontError (MPfR) | 0.077 | 0.108 | 0.036 |
| Spacing (S) | 476707.432 | 616934.564 | 77986.789 |
| Contribution (C) | 0.011 | 0.021 | 0.968 |
| R1Indicator (R1) | 0.185 | 0.185 | 0.185 |
| R2Indicator (R2) | 532206.855 | 725895.926 | 509244.064 |
| R3Indicator (R3) | 577971.466 | 786888.622 | 552230.901 |

###### Giá trị các tham số cho tập data1.json và runSeed = 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tham số** | **NSGAIII** | | | **GDE3** | | | **eMOEA** | | |
| **Min** | **Median** | **Max** | **Min** | **Median** | **Max** | **Min** | **Median** | **Max** |
| T | 163(s) | | | 168(s) | | | 197(s) | | |
| HV | 0.415 | 0.427 | 0.434 | 0.424 | 0.427 | 0.433 | 0.473 | 0.474 | 0.474 |
| GD | 0.000 | 0.001 | 0.001 | 0.001 | 0.001 | 0.002 | 0.000 | 0.000 | 0.000 |
| IGD | 0.028 | 0.036 | 0.057 | 0.035 | 0.037 | 0.038 | 0.002 | 0.002 | 0.003 |
| ASI | 0.080 | 0.107 | 0.116 | 0.091 | 0.115 | 0.161 | 0.025 | 0.041 | 0.052 |
| MPfE | 0.049 | 0.068 | 0.087 | 0.044 | 0.059 | 0.092 | 0.075 | 0.121 | 0.134 |
| S | 477192 | 486605 | 637265 | 553514 | 660760 | 786901 | 65049 | 72077 | 79176 |
| C | 0.195 | 0.199 | 0.208 | 0.187 | 0.203 | 0.207 | 0.181 | 0.202 | 0.212 |
| R1 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 |
| R2 | 659425 | 775570 | 813133 | 511595 | 520591 | 541902 | 502892 | 507505 | 511672 |
| R3 | 715510 | 841613 | 882291 | 556108 | 565913 | 589838 | 545416 | 550374 | 555019 |
| T2 | 0.5(s) | | | 0.5(s) | | | 1249(s) | | |

###### Giá trị tham số so sánh cho tập data1.json với runSeed=5

|  |  |  |  |
| --- | --- | --- | --- |
| **Tham số** | **NSGAIII** | **GDE3** | **eMOEA** |
| Thời gian chạy | 16 (s) | 18 (s) | 18(s) |
| Hypervolume | 0.268 | 0.282 | 0.283 |
| GenerationalDistance | 0.000 | 0.000 | 0.000 |
| InvertedGenerationalDistance | 0.000 | 0.000 | 0.000 |
| AdditiveEpsilonIndicator | 0.000 | 0.000 | 0.000 |
| MaximumParetoFrontError | 0.000 | 0.000 | 0.000 |
| Spacing | 114204660 | 48730186 | 3743492 |
| Contribution | 1.000 | 1.000 | 1.000 |
| R1Indicator | 0.185 | 0.185 | 0.185 |
| R2Indicator | 278303667 | 276302751 | 276308975 |
| R3Indicator | 313009752 | 309021845 | 309101722 |

###### Giá trị các tham số cho tập data2.json và runSeed = 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tham số** | **NSGAIII** | | | **GDE3** | | | **eMOEA** | | |
| **Min** | **Median** | **Max** | **Min** | **Median** | **Max** | **Min** | **Median** | **Max** |
| T | 80 (s) | | | 86(s) | | | 82(s) | | |
| HV | 0.247 | 0.272 | 0.277 | 0.281 | 0.282 | 0.282 | 0.284 | 0.284 | 0.285 |
| GD | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| IGD | 0.006 | 0.019 | 0.031 | 0.001 | 0.001 | 0.002 | 0.000 | 0.000 | 0.000 |
| ASI | 0.058 | 0.119 | 0.133 | 0.015 | 0.018 | 0.054 | 0.005 | 0.007 | 0.010 |
| MPfE | 0.001 | 0.004 | 0.084 | 0.001 | 0.003 | 0.007 | 0.003 | 0.004 | 0.006 |
| S | 36514245 | 62863451 | 135006369 | 47377334 | 55708642 | 65681077 | 1080155 | 1825133 | 1946851 |
| C | 0.180 | 0.196 | 0.226 | 0.215 | 0.223 | 0.236 | 0.189 | 0.193 | 0.225 |
| R1 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 | 0.185 |
| R2 | 277304817 | 277738955 | 287201429 | 276282632 | 276296824 | 276395527 | 276267235 | 276313552 | 276421653 |
| R3 | 310220995 | 310706665 | 321292338 | 308983338 | 308999210 | 309109595 | 308886079 | 308937864 | 309058729 |
| T2 | 0.5 (s) | | | 0.4 (s) | | | 330 (s) | | |

###### Giá trị tham số so sánh cho tập data2.json với runSeed=5

# KẾT LUẬN

Luận văn tốt nghiệp của sinh viên với đề tài: “Phát triển và thử nghiệm một số thuật toán trên MOEA Framework cho bài toán đấu thầu nhiều vòng” cơ bản đã hoàn thành. Luận văn đã giải quyết được các nhiệm vụ đặt ra ban đầu bao gồm:

1. Nghiên cứu mô hình hóa bài toán đấu thầu nhiều vòng dựa vào lý thuyết trò chơi và cân bằng Nash
2. Tìm hiểu các thuật toán di truyền, thuật toán tiến hóa đa mục tiêu để giải quyết bài toán đấu thầu đã được mô hình hóa.
3. Xây dựng mô hình bài toán đấu thầu nhiều vòng, dựa vào MOEA Framework triển khai các thuật toán tiến hóa đa mục tiêu NSGAIII, GDE3, εMOEA để giải bài toán đấu thầu nhiều vòng.
4. Đưa ra so sánh đánh giá các thuật toán tiến hóa đa mục tiêu đã sử dụng với các tham số Hypervolumn, ε+-Indicator, Generation Distance, Thời gian thực thi thuật toán

**Đóng góp khoa học của luận văn:**

Luận văn đã xây dựng vấn đề đấu thầu nhiều vòng thành một đối tượng mới riêng biệt trong MOEA Framework. Luận văn làm rõ 03 tham số so sánh cho các thuật toán tiến hóa đa mục tiêu Hypervolumn, ε+-Indicator, Generation Distance. Sử dụng 03 tham số này để so sánh 03 thuật toán tiến hóa đa mục tiêu: NSGAIII, GDE3, εMOEA.

Các kết quả của luận văn có thể trở thành nguồn tham khảo rất hữu dụng khi thực hiện cải tiến các thuật toán tiến hóa đa mục tiêu.

**Những khó khăn gặp phải trong quá trình thực hiện luận văn:**

1. Chưa tiếp cận với thực tế một cuộc đấu thầu nhiều vòng, việc mô tả bài toán còn phụ thuộc nhiều vào kiến thức trong tài liệu tham khảo và tra cứu internet. Không thấy rõ được sự xung đột trong quá trình thực tế. Lý do, không có điều kiện làm việc trong các đơn vị thực hiện tổ chức đấu thầu, thông tin chi tiết của các buổi đấu thầu thường không được công khai.
2. Khó khăn trong việc thu thập dữ liệu nên chỉ sử dụng 02 bộ dữ liệu được cung cấp bởi giáo viên hướng dẫn.

**Những hạn chế tồn tại:**

1. Chương trình xây dựng hiện tại chỉ sử dụng cho các bộ dữ liệu có cùng cấu trúc với bộ dữ liệu đang sử dụng.
2. Chương trình xây dựng rất hữu ích trong việc nghiên cứu phát triển các thuật toán nhưng lại không thể áp dụng ngay vào thực tế. Lý do, bài toán thực tế có nhiều yếu tố tác động cần xem xét để thêm vào quá trình mô hình hóa như việc dự án bị kéo dài, khối lượng công việc bị tăng, các vấn đề liên quan đến nhà thầu hoặc nguồn vốn của chủ đầu tư…

**Hướng phát triển**

Do điều kiện cá nhân còn nhiều hạn chế, nên vấn đề nghiên cứu về “Phát triển và thử nghiệm một số bài toán MOEA Framework cho bài toán đấu thầu nhiều vòng” trong khuôn khổ của luận văn này chỉ là những nghiên cứu ban đầu. Từ các kết quả của nghiên cứu này cho thấy bản thân tác giả phải tiếp tục nghiên cứu và triển khai theo các hướng sau:

* Tiếp tục thêm vào một số thuật toán tiến hóa đa mục tiêu khác.
* Nghiên cứu sâu hơn về các thuật toán có hướng đề xuất những cải tiến mới dựa vào kết quả so sánh các thuật toán hiện tại.
* Sử dụng kết quả của chương trình hiện tại, xây dựng một sản phẩm công nghệ thông tin để hỗ trợ ra quyết định trong các dự án cần đến đấu thầu và đấu thầu nhiều vòng.

# DANH MỤC TÀI LIỆU THAM KHẢO

[01] Hadka, D. (2016). *Beginner’s Guide to the MOEA Framework.* Copyright 2011-2016.

[02] Coello Coello, C. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems.* Springer Science+Business Media LLC, New York, NY.: 2007.

[03] Knowles, J. a. (2002). *. On metrics for comparing non-dominated sets. - pages 711–716.* Honolulu: In Congress on Evolutionary Computation.

[04] Bosman, P. A. (2003). *The balance between proximity and diversity in multiobjective evolutionary algorithms. .* IEEE Transactions on Evolutionary Computation. 7(2):174–188.

[05] Deb, K. a. (2002). *Running performance metrics for evolutionary multi-objective optimization. KanGAL.* Kanpur: KanGAL Report No. 2002004, Kanpur Genetic Algorithms Laboratory (KanGAL).

[06] Knowles, J. a. (2002). *On metrics for comparing non-dominated sets.* Honolulu, HI.: In Congress on Evolutionary Computation (CEC 2002), pages 711–716.

[07] Hadka, D. a. (2012). *Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization.* Evolutionary Computation, 20(3):423–452.

[08] A. Konak, D. W. (2006). *Multi-objective optimization using genetic algorithms: A tutorial.* J. Reliability Engineering and System Safety, No. 91, pp. 992-1007.

[09] J.D.Schaffer. (1985). *Multiple objective optimization with vector evaluated genetic algorithms.* Proceedings of the International Conference on Genetic Algorithm and their Applications.

[10] Fleming, C. M. (1993). *Multiobjective genetic algorithms.* London, UK: IEE Colloquium on Genetic Algorithms for Control Systems Engineering (Digest No. 1993/130), 28 May 1993.

[11] J. Horn, N. N. (1994). *A niched Pareto genetic algorithm for multiobjective optimization.* Orlando, RL, USA: Proceedings of the first IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, 27–29 June.

[12] P. Hajela P, C. Y. (1995). *Genetic search strategies in multicriterion optimal design.* Perth, WA, Australia: Struct Optimization , 4(2), 1992, pp. 99–107. on Evolutionary Computation, 29 November –1 December.

[13] T. Murata, H. I. (1995). *MOGA: multi-objective genetic algorithms.* Perth, WA, Australia: Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, 29 November –1 December.

[14] N. Srinivas, K. D. (1994). *Multiobjective optimization using nondominated sorting in genetic algorithms.* J. Evol Comput 2(3), pp. 221–48.

[15] E. Zitzler, L. T. (1999). *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach.* IEEE Trans Evol Comput, 3(4), pp. 257–71.

[16] E. Zitzler, M. L. (2001). *SPEA2: improving the strength Pareto evolutionary algorithm.* Zurich, Switzerland: Swiss Federal Institute Techonology.

[17] J. D. Knowles, D. W. (2000). *Approximating the nondominated front using the Pareto archived evolution strategy.* Evol Comput, 8(2), pp.149–72.

[18] D. W. Corne, J. D. (2000). *The Pareto envelope-based selection algorithm for multiobjective optimization.* Paris, France: Proceedings of sixth International Conference on Parallel Problem Solving from Nature, 18–20 September.

[19] D. W. Corne, N. R. (2001). *PESA-II: region-based selection in evolutionary multiobjective optimization.* San Francisco, CA: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001).

[20] K. Deb, A. P. (2002). *A fast and elitist multiobjective genetic algorithm: NSGA-II.* IEEE Trans Evol Comput, 6(2), pp.182–97.

[21] H. Lu, G. G. (2003). *Rank-density-based multiobjective genetic algorithm and benchmark test function study.* IEEE Trans Evol Comput, 7(4), pp.325–43.

[22] G. G. Yen, H. L. (2003). *Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation.* IEEE Trans Evol Comput, 7(3), pp.253–74.

[23] Thanh Hoàng Ngọc, D. T. (n.d.). *Ứng dụng giải thuật tiến hóa đa mục tiêu trong thiết kế tối ưu kiến trúc mạng viễn thông.*

[24] Kalyanmoy Deb, M. M. (2003). *A Fast Multi-objective evolutionary algorithm for finding well-spread pareto-optimal solution.* Kanpur, Indian: Indial Institute of technology Kanpur.

[25] Saku Kukkonen - Kanpur Genetic Algorithms Laboratory, J. L.-D. (2005). *(2005) "GDE3: The third Evolution Step of Generalized Differential Evolution".* Kanpur.

[26] Guillermo Campos Ciro, F. D. (2016). *A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resouce constraints.*

[27] SV: Trần Hải Linh, G. P.-T. (2016). *ĐATN Xây dựng phần mềm trợ giúp ra quyết định cho đấu thầu nhiều vòng dựa vào ứng dụng.* Hà Nội: Đại học Bách Khoa.