|  |
| --- |
|  |
| **TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  **VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  **---------------------------------------** |
|  |
|  |
|  |
|  |
| **BÁO CÁO BÀI TẬP LỚN MÔN HỌC**  **KINH TẾ CÔNG NGHỆ PHẦN MỀM** |
|  |
|  |
| |  |  | | --- | --- | | **Tên bài tập lớn:** | Áp dụng MOEA Framework cho bài toán Đấu thầu nhiều vòng | | **Nhóm sinh viên:** | 1. Nguyễn Thượng Khánh, Lớp KSTN CNTT K59  2. Nguyễn Xuân Tùng, Lớp KSTN CNTT K59  3. Nguyễn Thanh Hà, Lớp KSTN CNTT K59 | | **Giảng viên:** | PGS.TS. Huỳnh Quyết Thắng | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Hà Nội – 2018 |

# 

# LỜI CAM ĐOAN

Tập thể các sinh viên:

1. Nguyễn Thượng Khánh, Lớp KSTN CNTT K59

2. Nguyễn Xuân Tùng, Lớp KSTN CNTT K59

3. Nguyễn Thanh Hà, Lớp KSTN CNTT K59

Xin cam đoan đây là công trình nghiên cứu của tập thể 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, và Nghiên cứu sinh, Thạc sỹ Trịnh Bảo Ngọc.

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 26 tháng 11 năm 2018*  **Nhóm thực hiện nghiên cứu**  **(ký, ghi rõ họ tên)** |

# LỜI CẢM ƠN

Lời đầu tiên chúng 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 chúng tôi ân cần, cung cấp cho chúng tôi nhiều tài liệu quan trọng là nền tảng để tôi tìm tòi nghiên cứu sâu hơn về đề tài “*Áp dụng MOEA Framework vào bài toán đấu thầu nhiều vòng*”. Mỗi lời góp ý của Thầy giúp chúng tôi định hướng tốt hơn, đúng đắn hơn cho đề tài mình đã chọn.

Tôi cũng xin gửi lời cám ơn tới Thạc sỹ Trịnh Bảo Ngọc đã tận tình giúp đỡ, và cung cấp nhiều kiến thức và tài liệu bổ ích và cần thiết. Để từ đó, chúng tôi có đủ khả năng và hiểu biết để thực hiện và hoàn thành nghiên cứu này một cách tốt nhất.

Lời tiếp theo chúng tôi xin dành lời cảm ơn tới bạn bè, các thành viên trong lớp KSTN CNTT K59 đã quan tâm, chia sẻ, ủng hộ về vật chất và tinh thần lớn lao để chúng tôi có động lực học tập và hoàn thành nghiên cứu này.

# MỤC LỤC

[LỜI CAM ĐOAN i](#_Toc534474153)

[LỜI CẢM ƠN ii](#_Toc534474154)

[MỤC LỤC iii](#_Toc534474155)

[DANH MỤC CÁC BẢNG VÀ HÌNH VẼ v](#_Toc534474156)

[DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT vi](#_Toc534474157)

[MỞ ĐẦU 1](#_Toc534474158)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1](#_Toc534474159)

[1.1. Khái niệm về đấu thầu, đấu thầu nhiều vòng. 1](#_Toc534474160)

[1.2. Lý thuyết trò chơi và cân bằng Nash 3](#_Toc534474161)

[1.3. Bài toán tối ưu hoá đa mục tiêu 5](#_Toc534474162)

[1.4. Giải thuật di truyền 6](#_Toc534474163)

[CHƯƠNG 2: MÔ HÌNH BÀI TOÁN ĐẤU THẦU NHIỀU VÒNG 7](#_Toc534474164)

[2.1. Mô tả bài toán đấu thầu nhiều vòng 7](#_Toc534474165)

[2.2. Mô hình toán học 9](#_Toc534474166)

[2.3. MOEA Framework là gì, ứng dụng vào bài toán 11](#_Toc534474167)

[2.4. Các thuật toán sử dụng cho MOEA Framework 11](#_Toc534474168)

[CHƯƠNG 3: CÀI ĐẶT VÀ ĐÁNH GIÁ THỰC NGHIỆM 26](#_Toc534474169)

[3.1. Xây dựng chương trình 26](#_Toc534474170)

[3.2. Các tiêu chí đánh giá 27](#_Toc534474171)

[3.3. Kết quả thử nghiệm 28](#_Toc534474172)

[3.4. Nhận xét và đánh giá 29](#_Toc534474173)

[KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 29](#_Toc534474174)

[1. Kết luận 29](#_Toc534474175)

[2. Định hướng phát triển 29](#_Toc534474176)

[Trong thời gian tời, đề tài tiếp tục nghiên cứu, tìm hiểu các thuật toán đã được cải tiến, thực hiện hiệu chỉnh các tham số cần thiết để cải tiến thời gian chạy và tối ưu giá trị Payoff của các thuật toán. 29](#_Toc534474177)

[DANH MỤC TÀI LIỆU THAM KHẢO 30](#_Toc534474178)

# DANH MỤC CÁC BẢNG VÀ HÌNH VẼ

**Danh mục các hình vẽ**

Hình 1. Biểu diễn các lời giải trên không gian mục tiêu 14

Hình 2. Biểu diễn quá trình chọn lọc trong thuật toán ε-MOEA 15

Hình 3. Mô tả phương pháp chọn lọc trong các thuật toán MOEA hiện đại 18

Hình 4. Mô phỏng các vecto vị trí của một cá thể trong quần thể 22

Hình 5. Sơ đồ giải thuật PSO 23

**Danh mục các bảng**

Bảng 1. Kết quả thử nghiệm các thuật toán 28

# DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

# 

# MỞ ĐẦU

Bố cục của báo cáo bao gồm phần mở đầu, kết luận và ba chương:

***Chương 1:*** Trình bày ngắn gọn các cơ sở lý thuyết, bao gồm:

* Khái niệm về đấu thầu, đấu thầu nhiều vòng là gì.
* Lý thuyết trò chơi và cân bằng Nash.
* Bài toán tối ưu hoá đa mục tiêu.
* Giải thuật di truyền.

***Chương 2:*** Mô hình bài toán đầu thầu nhiều vòng

* Mô tả bài toán đấu thầu nhiều vòng
* Mô hình hóa toán học bài toán.
* MOEA Framework là gì, ứng dụng vào bài toán này như nào.
* Các thuật toán sử dụng cho MOEA Framework.

**Chương 3:** Cài đặt và đánh giá thực nghiệm các giải thuật với MOEA Framework

* Trình bày cài đặt bài toán với các thuật toán khác nhau
* Phân tích kết quả, so sánh thuật toán

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

|  |
| --- |
| Chương này trình bày các vấn đề lý thuyết sau:   * Khái niệm về đấu thầu, đấu thầu nhiều vòng. * Lý thuyết trò chơi và cân bằng Nash. * Giải thuật di truyền. |

## 1.1. Khái niệm về đấu thầu, đấu thầu nhiều vòng.

Theo Wikipedia, **Đấu thầu** là một quá trình chủ đầu tư lựa chọn được một nhà thầu đáp ứng các yêu cầu của mình theo quy định của luật pháp. Trong nền kinh tế thị trường, người mua 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.

Để cho dễ hiểu thì ta hình dung chủ thầu là một tổ chức có một dự án (ví dụ như xây 1 trường học). Để xây trường học thì chủ thầu cần rất nhiều vật tư thiết bị, nhân công lao động… mà họ không tự có sẵn. Vì vậy họ muốn mua vật tư, thuê nhân công của công ty khác. Tuy nhiên lại có rất nhiều công ty cung cấp dịch vụ này nên chủ thầu phải tổ chức ra một cuộc đấu thầu để chọn ra công ty (nhà thầu) nào đưa ra cho họ giá cả thấp nhất mà vẫn thỏa mãn nhu cầu. Đồng thời các nhà thầu tham gia cũng nhắm đến mục tiêu là được chọn cung cấp mặt hàng (bán được hàng) cho chủ thầu nhưng phải thu được lợi nhuận cao nhất có thể.

Khái niệm đầu thầu nhiều vòng chỉ hình thức mà dự án được chia nhỏ thành nhiều gói nhỏ, triển khai theo từng giai đoạn. Có thể do dự án kéo dài trong nhiều năm, có nhiều công việc cần làm nên thay vì đầu tư cho toàn bộ dự án tại một thời điểm duy nhất thì chủ đầu tư sẽ chia nhỏ ra. Việc chia ra này để cho dễ quản lý dự án hơn đồng thời giảm thiểu rủi ro trong thời gian thực hiện dự án.

Trong đấu thầu nhiều vòng, quyết định của chủ đầu tư phần lớn là do quá trình đàm phán và tầm nhìn của chủ đầu tư mà không dựa trên một cơ sở lý thuyết khoa học nào chứng minh rằng quyết định đó là tối ưu. Do đó, dự án có nguy cơ gặp phải một số rủi ro như: mức chi phí vượt hạn mức cho phép, 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.

Quyết định chọn nhà thầu cho một lần đấu thầu đã là việc khó khăn và rủi ro cao thì các quyết định trong dự án đấu thầ 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 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? Số lượng này phải đảm bảo đủ để thi công dự án tại thời điểm đó tuy nhiên cũng phải cân đối phù hợp với những thời điểm khác.

Một khó khăn nữa 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?

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.

## 1.2. Lý thuyết trò chơi và cân bằng Nash

Lý thuyết trò chơi là một nhánh của toán học ứng dụng được sử dụng để phân tích các tình huống cạnh tranh mà kết quả không phụ thuộc vào sự lựa chọn của một bên hay còn là cơ hội lựa chọn của các người chơi khác. Bởi vậy, kết quả sẽ phụ thuộc vào quyết định của tất cả người chơi, trong đó mỗi người chơi sẽ cố gắng dự đoán sự lựa chọn của những người chơi còn lại để có thể đưa ra lựa chọn tốt nhất cho mình.

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

Trong lý thuyết trò chơi lợi ích của các người chơi cạnh tranh nhau, kết quả không phụ thuộc vào sự lựa chọn của một bên hay còn là cơ hội lựa chọn của các người chơi khác. Bởi vậy, kết quả sẽ phụ thuộc vào quyết định của tất cả người chơi, trong đó mỗi người chơi sẽ cố gắng dự đoán sự lựa chọn của những người chơi còn lại để có thể đưa ra lựa chọn tốt nhất cho mình.

Trong lý thuyết trò chơi, một trò chơi được biểu diễn bởi:

*G = (N, Si, ui)*

Trong đó:

* *G* là trò chơi
* *N =* {1, 2, …, n} là tập gồm *n* người chơi
* *Si* là tập hành động của người chơi thứ *i* (*i ∈ N*)
* *S =*{*s* *|* *s = (si)i∈N,* *si∈Si*, ∀*i ∈ N* } là tổ hợp hành động của các người chơi (một cấu hình/chiến lược của trò chơi)
* *ui* là hàm payoff (thể hiện cơ chế thưởng phạt) của người chơi thứ *i*với một cấu hình S: *(s1, s2, …, sn)* 🡪 *ui(s1, s2, …, sn)*, có thể là lợi nhuận (cần tối đa hoá) hoặc chi phí (cần tối thiểu hoá).

Cân bằng Nash xác định một chiến lược tối ưu cho các trò chơi khi chưa có điều kiện tối ưu nào được xác định trước đó. Định nghĩa cơ bản của cân bằng Nash là: Nếu tồn tại một tập hợp các chiến lược cho một trò chơi với đặc tính là không có 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 cùng kết quả tương ứng nhận được 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 sẽ làm cho chính người đó thu lợi ích ít hơn mức có được với chiến lược hiện.

Một cấu hình *s\* = (s1, …, si, …, sn)* được gọi là cân bằng Nash khi:

*ui(si\*, s-i\*) ≥ ui(si, s-i\*),* ∀*si ∈ Si, ∀ i ∈ N*

## 1.3. Bài toán tối ưu hoá đa mục tiêu

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 :

Thông thường, bài toán tối ưu hoá đa mục tiêu thường không có một lời giải mà 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.

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 (tập tối ưu Pareto). 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.

## 1.4. Giải thuật di truyền

Giải thuật di truyền là một loại thuật toán ứng dụng nguyên tắc di truyền gen trong sinh học vào để giải các bài toán. Cụ thể là trong sinh học, các cá thể trong một quần thể đều được đặc trưng bởi các gen. Và những gen này sau các quá trình chọn lọc, lai ghép, đột biến thì đều tiến hóa để thích nghi hơn với môi trường xung quanh.

Tương tự trong toán học, các lời giải của bài toán được coi như các gen (ban đầu có thể chưa phải tốt nhất), sau khi ta chọn lọc, lai ghép và đột biến các lời giải theo quy tắc toán học nhất định ta sẽ được lời giải mới “tốt hơn” so với lời giải cũ, cứ như thế cho đến khi ta tìm được lời giải đủ tốt thỏa mãn 1 tiêu chí đặt trước.

Giải thuật di truyền rất phù hợp cho những bài toán đa mục tiêu và đấu thầu nhiều vòng là một trong số đó khi mà ta phải thỏa mãn rất nhiều tiêu chí: lợi ích chủ thầu, lợi ích từng nhà thầu, tổng lợi ích các nhà thầu, tổng lợi ích nhà thầu và chủ thầu.

# CHƯƠNG 2: MÔ HÌNH BÀI TOÁN ĐẤU THẦU NHIỀU VÒNG

|  |
| --- |
| Chương này trình bày các vấn đề sau:   * Mô tả bài toán đấu thầu nhiều vòng * Mô hình hóa toán học bài toán. * MOEA Framework là gì, ứng dụng vào bài toán này như nào. * Các thuật toán sử dụng cho MOEA Framework. |

## 2.1. Mô tả bài toán đấu thầu nhiều vòng

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 - làm sao để việc đàm phán đấu thầu nhiều vòng thành công? Làm sao để lựa chọn được nhà thầu hợp lý? Phân phối các gói thầu cho các nhà thầu phù hợp để hài hòa lợi ích giữa các bên?

**Trạng thái cân bằng lợi ích**

* 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.

Chúng ta quy bài toán đàm phán đấu thầu nhiều vòng ở đây về dạng mô hình trò chơi thông tin hoàn hảo với n người chơi. Các bên sẽ đưa ra các 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.

**Đầu vào 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 bài toán:** Danh sách gói thầu với nhà thầu được lựa chọn, và thời điểm tổ chức gói thầu tương ứng

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.

## 2.2. Mô hình toán học

Gọi:

* N là số lượng gói thầu
* M là số nhà thầu tham gia
* T là thời điểm bắt đầu dự án
* ECi là chi phí ước tính (Estimated Cost) của gói thầu i.
* r là tỉ lệ lạm phát của đồng tiền
* mi là số lượng hàng hóa trong gói thầu i
* ci là nhà thầu đã trúng gói thầu i
* ti là thời điểm tố chức gói thầu i
* SPjci là giá bán (Sell price) sản phẩm j của nhà thầu ci ở gói thầu i
* BPjci là giá mua vào (Buy price) sản phẩm j của nhà thầu ci ở gói thầu i
* Mij là số lượng sản phẩm j cần mua ở gói thầu i
* Dijtci là chiết khấu (discount) mà nhà thầu ci đã áp dụng cho sản phẩm thứ j của gói thầu i tại thời điểm ti
* ai là tham số đánh giá quan hệ nhà thầu i với chủ đầu tư
* qi là tham số đánh giá chất lượng của nhà thầu i

**Lợi ích của chủ đầu tư:**

**Lợi ích của nhà thầu thứ k:**

Trong đó:

* i xác định đối với gói thầu i nều nhà thầu k mua gói thầu i, ci = k

**Cân bằng lợi ích giữa các nhà thầu:**

**Tổng lợi ích của tất cả các nhà thầu:**

**Hàm đảm bảo chất lượng dự án:**

Ngoài mục tiêu lợi ích giữa các nhà thầu đánh giá cực tiểu, các mục tiêu còn lại sẽ đánh giá cực đại. Dự án có M nhà thầu tham gia thì số mục tiêu của bài toán là:

## 2.3. MOEA Framework là gì, ứng dụng vào bài toán

Để giải quyết bài toán này thì ta có thể tự xây dựng thuật toán từ đầu, tuy nhiên cũng có 1 cách tiếp cận khác nhanh hơn là sử dụng Framework (bao gồm các thuật toán đã được xây dựng sẵn) để chạy ra kết quả, vì mục đích của đề tài là so sánh các thuật toán với nhau chứ không phải là tối ưu thuật toán.

Một trong các Framework đó là MOEA Framework. MOEA Framework là một thư viện mã nguồn mở viết bằng Java dành cho việc phát triển và thí nghiệm những thuật toán đa mục tiêu. MOEA Framework hỗ trợ giải thuật di truyền, tiến hóa khác biệt, phương pháp tối ưu bầy đàn, lập trình di truyền, tiến hóa ngữ pháp và nhiều cái khác. Rất nhiều thuật toán được Framework cung cấp sẵn ví dụ như NSAG-II, NSGA-III, ε-MOEA, GDE3, PAES, PESA2, SPEA2, IBEA, SMS-EMOA, SMPSO, OMOPSO, CMA-ES, and MOEA/D. Người dùng chỉ cần cung cấp đầu vào bài toán, chọn thuật toán, sau đó Framework sẽ xử lý và đưa ra kết quả. Ngoài ra, MOEA Framework còn cung cấp những công cụ cần thiết cho việc thiết kế, phát triển, triển khai và kiểm thử thống kê tối ưu thuật toán.

## 2.4. Các thuật toán sử dụng cho MOEA Framework

Trong đề tài này, dưới sự hướng dẫn của anh NCS, nhóm chúng em sử dụng **6 thuật toán** là: **NSGA-III, ε-MOEA, GDE3, PESA2, IBEA, SMPSO**. Lý do là vì 6 thuật toán này có cách tiếp cận khác nhau với bài toán, đảm bảo độ đa dạng để có thể so sánh được.

**2.4.1. Thuật toán NSGA-III**

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:**

1. Sinh tập cá thể ngẫu nhiên.
2. Lai ghép và đột biến.
3. Xếp hạng rồi chọn lọc.
4. 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.

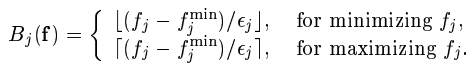
**NSGA-III** thì các bước từ đầu đến lai ghép và đột biến y hệt như NSGA-II tuy nhiên có khác ở bước chọn lọc:

* Đầu tiên xác định số **reference point** (hiểu là số cá thể trong quần thể, sẽ giải thích sau).
* Sau khi lai ghép đột biến để sinh ra con thì tính min với từng **Objective Function (OF)**, sau đót trừ tất cả f khác cho min 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.

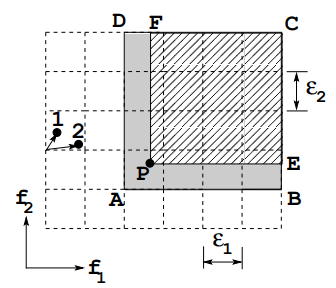
**2.4.2. Thuật toán** **ε-MOEA**

1. **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:



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

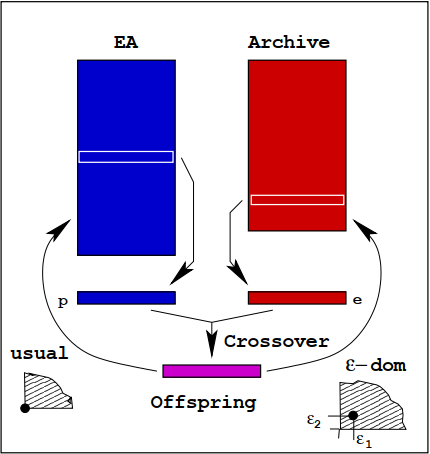
Các điểm A, B, C, D, E, F, P biểu diễn biến quyết định có giá trị hàm mục tiêu là toạ độ tương ứng. Trong trường hợp tối ưu hoá min cả *f1*và *f2*, theo định nghĩa **dominate** thông thường, dễ thấy, P dominate các điểm trong hình chữ nhật PECF. Tuy nhiên, trong khái niệm **ε-dominate**,điểm A sẽ được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

1. **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.



Hình 2. Biểu diễn quá trình chọn lọc trong thuật toán ε-MOEA

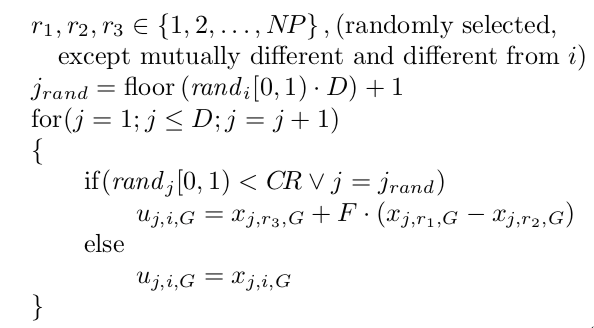
Việc quyết định giữ lại cá thể con *c* trong P (hoặc E) hay không dựa trên khái niệm **ε-dominate** đã trình bày ở trên theo quy tắc:

* 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

**2.4.3. Thuật toán GDE3**

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**.

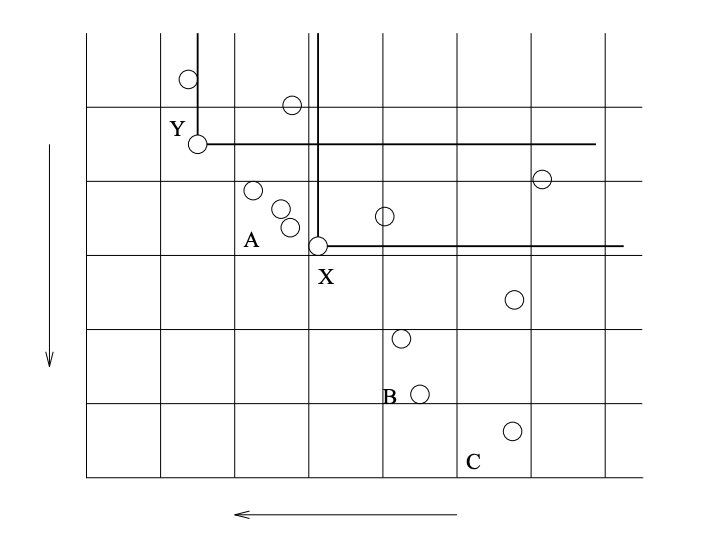
Quy tắc selection của **GDE3**:

* 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**).

**2.4.4. Thuật toán PESA2**

1. **Chọn lọc dựa trên cá thể (Individual-based selection)**

***Hình 1*** biểu diễn chiến lược chọn lọc chính được sử dụng trong các thuật toán tiến hoá đa mục tiêu hiện tại. Trong hình, một số điểm được vẽ trên không gian mục tiêu cho một vấn đề 2 mục tiêu, và mục đích của ta là tối thiểu hoá theo cả 2 chiều. Không gian mục tiêu được chia thành các hyperbox.



Hình 3. Mô tả phương pháp chọn lọc trong các thuật toán MOEA hiện đại

Trong các thuật toán như PAES hay PESA, việc một hyperbox bị chiếm bởi các các thể được sử dụng để chọn lọc. Một tập lưu trữ được duy trì chỉ chứa các solution không bị trội. Việc chọn lọc chỉ được thực hiện từ tập này. Hàm fitness của một cá thể được tính bằng số lượng solution khác cùng hyperbox với cá thể đó.

1. **Chọn lọc dựa trên vùng (Region-based selection) - Thuật toán PESA2**

Thuật toán PESA2 dựa trên nguyên lý chuẩn của một thuật toán tiến hoá, duy trì 2 quần thể: một quần thể nội bộ (internal population) kích thước cố định và một quần thể ngoài (external population) (ví dụ như một tập lưu trữ) không cố định số cá thể nhưng giới hạn kích thước quần thể.

Quần thể nội bộ lưu trữ các solution được tạo ra từ tập lưu trữ bằng các phép biến đổi khác nhau, và tập lưu trữ chỉ chứa các solution vượt trội được phát hiện trong suốt quá trình tìm kiếm.

Một lưới quyết định được tạo ra trên không gian mục tiêu để duy trì đa dạng quần thể. Lưới quyết định được chia thành nhiếu khối nhỏ (hyperbox). Số lượng giải pháp trong một hyperbox được gọi là mật độ của hyperbox và được sử dụng để phân biệt các giải pháp trong 2 quá trình quan trọng của một giải thuật tiến hoá đa mục tiêu (EMO): chọn lọc dựa vào giao phối và chọn lọc từ môi trường.

Không giống như những thuật toán EMO khác, việc giao phối trong PESA-II được thực hiện theo khu vực hơn là theo cá nhân. Một hyperbox được chọn đầu tiên và sau đó cá thể được chọn để tiến hoá sẽ được chọn ngẫu nhiên từ hyperbox đã chọn, do đó các hyperbox đông hơn sẽ không đóng góp nhiều cá thể hơn so với các hyperbox thưa hơn.

Trong quá trình chọn lọc từ môi trường, các cá thể ứng cử viên trong quần thể nội bộ lần lượt được thêm vào tập lưu trữ (archive set) nếu nó không bị trội bởi bất kỳ cá thể nào trong quần thể, và không bị trội bởi bất cứ cá thể nào trong archive set. Khi một cá thể được thêm vào archive set, sẽ có sự thay đổi của archive set và lưới môi trường. Đầu tiên, các cá thể trong tập lưu trữ bị trội bởi cá thể ứng cử viên sẽ bị loại bỏ để chắc chắn rằng chỉ có các cá thể không bị trội mới có trong archive set. Sau đó, lưới được kiểm tra xem biên của nó có bị thay đổi bởi việc thêm và bớt các cá thể. Cuối cùng, nếu việc thêm phần tử làm tập lưu trữ bị đầy, một phần từ ngẫu nhiên trong hyperbox đông nhất sẽ bị loại bỏ.

**2.4.5. Thuật toán IBEA**

Ý tưởng chính của thuật toán IBEA (IBEAHD) là sử dụng một chỉ số hypervolume nhị phân trong quá trình chọn lọc, khi xác định solution nào sẽ được giữ lại trong thế hệ tiếp theo. Chỉ số nhị phân hypervolumn này gán một giá trị thực cho 2 tập giải pháp liên quan đến cùng một điểm trong không gian mục tiêu.

được định nghĩa là miền không gian bị trội bởi B nhưng không bị trội bởi A

Với biểu thị hypervolume được hình thành bởi bộ giải pháp A, tương tự, là hypervolume của sự kết hợp giữa 2 bộ giải pháp A và B. âm nếu tất cả các solution trong B đều bị trội bởi solution trong A.

Mã giả của thuật toán IBEA được trình bày ở dưới (IBEA algorithm).

Trước hết, IBEA tạo ngẫu nhiên quần thể khởi tạo ở bước 1, sau đó các bước sau được lặp cho đến khi điều kiện dừng được thoả mãn. Giá trị của mục tiêu được co dãn và giá trị hàm fitness được gán cho mỗi cá thể ở bước 2 và bước 3. Bước 4 thực hiện việc chọn lọc từ môi trường, duyệt qua các cá thể kém nhất trong P dựa vào chỉ số I cho đến khi còn lại cá thể. Giá trị chỉ số của các cá thể còn lại trong P sẽ được cập nhật. Bước 6 và bước 7 thực hiện việc tạo các cá thể mới và thêm chúng vào quần thể P.

**IBEA algorithm**

**Input**

: kích thước quần thể

N: tổng số lần đánh giá các giải pháp

: nhân tố tăng giảm giá trị mục tiêu

: giá trị chỉ số (sai khác hypervolume)

**Output:**

A - tập xấp xỉ Pareto

**Bước 1. Khởi tạo**

Khởi tạo ngẫu nhiên quần thể P và đặt giá trị đếm m.

**Bước 2. Mở rộng giá trị mục tiêu**

1. Tìm cận dưới và cận trên của mỗi mục tiêu *i.*

và

2. Tăng cận trên:

3. Chuyển đổi mỗi mục tiêu vào miền [0, 1]

**Bước 3. Gán giá trị hàm fitness**

1. Tính chỉ số sử dụng các giá trị thay vì và xác định chỉ số có trị tuyệt đối lớn nhất

2. Tính giá trị hàm fitness cho tất cả giá trị

**Bước 4. Lựa chọn từ môi trường**

Duyệt qua 3 bước sau cho đến khi kích thước quần thể P

1. Chọn một cá thể có giá trị hàm fitness nhỏ nhất:

2. Xoá khỏi P

3. Cập nhật giá trị hàm fitness cho toàn bộ cá thể còn lại

**Bước 5. Điều kiện dừng**

Nếu trả về tập A là các giải pháp không bị trội trong P, nếu không, tiếp tục.

**Bước 6. Chọn lọc từ giao phối**

Áp dụng các toán tử chọn lọc để chọn cặp bố mẹ từ P.

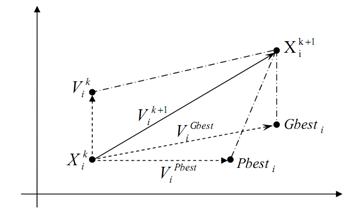
**Bước 7. Biến đổi**

Thực hiện lai ghép trên cặp bố mẹ để tạo ra 2 cá thể con, và dùng các toán tử đột biến trên các cá thể con này. Sau đó thêm vào P. Tăng m lên 1 đơn vị. Quay lại bước 2.

**2.4.6. Thuật toán SMPSO**

1. **Thuật toán PSO**

PSO được khởi tạo bằng một nhóm cá thể(nghiệm) ngẫu nhiên và sau đó tìm nghiệm tối ưu bằng cách cập nhật các thế hệ. Trong mỗi thế hệ, mỗi cá thể được cập nhật theo hai giá trị tốt nhất. Gía trị thứ nhất là nghiệm tốt nhất đạt được cho tới thời điểm hiện tại, gọi là Pbest. Một nghiệm tối ưu khác mà cá thể này bám theo là nghiệm tối ưu toàn cục Gbest, đó là nghiệm tốt nhất mà cá thể lân cận cá thể này đạt được cho tới thời điểm hiện tại. Nói cách khác, mỗi cá thể trong quần thể cập nhật vị trí của nó theo vị trí tốt nhất của nó và của cá thể trong quần thể tính tới thời điểm hiện tại.



Hình 4. Mô phỏng các vecto vị trí của một cá thể trong quần thể

Trong đó :

 : Vị trí cá thể thứ i tại thế hệ thứ k

 : Vận tốc cá thể i tại thế hệ thứ k

Pbesti : Vị trí tốt nhất của cá thể thứ i

Gbesti : Vị trí tốt nhất của cá thể trong quần thể

**Vận tốc và vị trí của mỗi cá thể được tính như sau**:

vik+1 = w.vik + c1.r1().(pbesti – xik) + c2.r2().(gbest – xik)

xik+1 = xit + vik+1

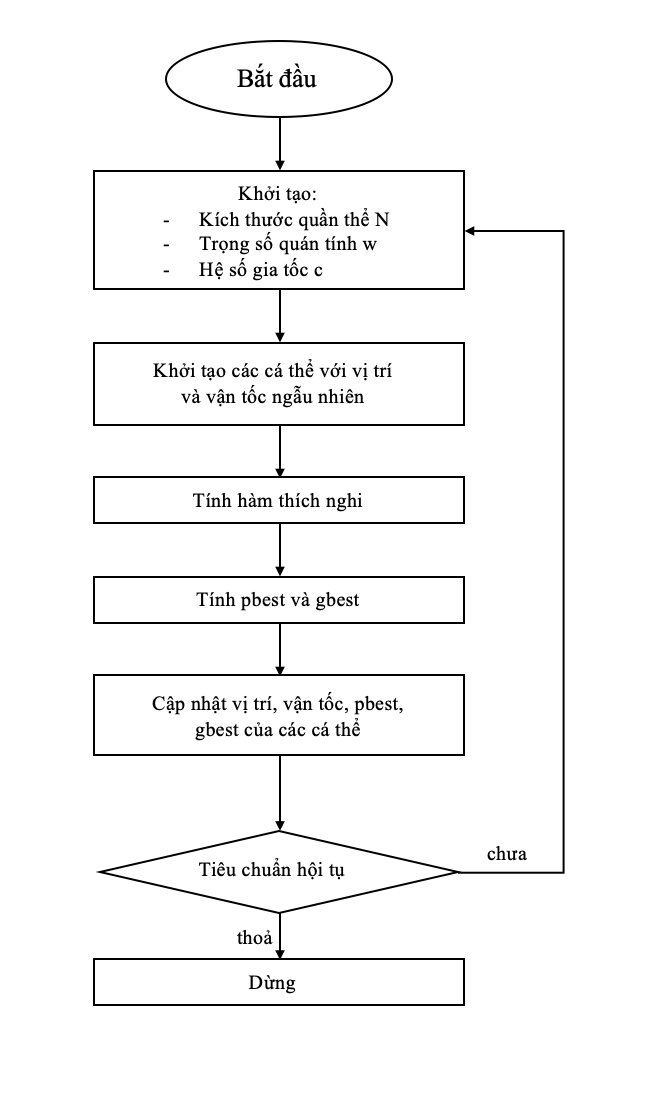
Trong đó

w : trọng số quán tính

c1, c2 : các hệ số gia tốc

r1, r2 :  số ngẫu nhiên giữa 0 và 1

**Lưu đồ thuật giải PSO**:



Hình 5. Sơ đồ giải thuật PSO

1. **Thuật toán SMPSO**

Trong giải thuật PSO thông thường, r1 và r2 được chọn ngẫu nhiên trong đoạn [0, 1] và c1, c2 là các tham số điều khiển ảnh hưởng của các cá thể đến cá thể toàn cục đầu đàn.

Để điều khiên vận tốc của cá thể, thay vì sử dụng cận trên và cận dưới của hệ số vận tốc, ta sử dụng một hệ số tương quan:

với

**SMPSO algorithm**

1: initializeSwarm()

2: evaluation()

3: initializeLeadersArchive()

4: generation = 0

5: while generation < maxGenerations do

6: computeSpeed()

7: updatePosition()

8: mutation()

9: evaluation()

10: updateLeadersArchive()

11: updateParticlesMemory()

12: generation++

13: end while

14: returnLeadersArchive()

# CHƯƠNG 3: CÀI ĐẶT VÀ ĐÁNH GIÁ THỰC NGHIỆM

|  |
| --- |
| Chương này trình bày các vấn đề sau:   * Xây dựng chương trình * Các tiêu chí đánh giá * Kết quả thử nghiệm * Nhận xét và đánh giá |

## 3.1. Xây dựng chương trình

Chương trình thử nghiệm được viết bằng ngôn ngữ Java, sử dựa trên các thư viện có sẵn của MOEA Framework.

**3.1.1. Các lớp models**

Các lớp models là các lớp (trong lập trình hướng đối tượng) để biểu diễn các đối tượng trong bài toán đấu thầu nhiều vòng, bao gồm:

* Lớp **Package:** biểu diễn gói thầu
* Lớp **Constractor:** biểu diễn nhà thầu
* Lớp **Product:** biểu diễn sản phẩm
* **…**

**3.1.2. Định nghĩa bài toán trong MOEA Framework**

**MOEA Framework** đã cài đặt sẵn một số giải thuật (bao gồm 6 giải thuật trình bày ở trên), cần định nghĩa bài toán, các biến quyết định, hàm mục tiêu và ràng buộc để MOEA Framework giải quyết.

Một bài toán được định nghĩa bằng cách *implement* lớp trừu tượng AbstractProblem. MOEA Framework chấp nhận các biến quyết định thuộc một số kiểu dữ liệu: số nguyên, số thực, chuỗi bit, … Vì vậy, để đơn giản, ta quy một lời giải của bài toán đấu thầu nhiều vòng về một vector số nguyên (có số chiều bằng số biến quyết định). Do có *N* gói thầu, mỗi gói thầu cần xác định 2 yếu tố là thời gian tổ chức và nhà thầu trúng gói thầu đó, mỗi lời giải sẽ được quy về một vector *2\*N* chiều, với thời gian tổ chức xác định bởi số ngày tính từ thời điểm bắt đầu dự án, nhà thầu được xác định bởi mã số (số thứ tự) nhà thầu.

Như trình bày ở trên, ta có 4 hàm mục tiêu cần tối ưu:

* Lợi nhuận của chủ đầu tư
* Tổng lợi nhuận của các nhà thầu
* Cân bằng lợi ích giữa các nhà thầu (tốt hơn khi lợi nhuận các nhà thầu ít chênh lệch hơn)
* Chất lượng dự án (dựa trên uy tín sẵn có của nhà thầu)

Đồng thời, theo thực tế, các ràng buộc sau cần được thoả mãn:

* Lợi nhuận của chủ đầu tư không âm
* Lợi nhuận của mỗi nhà thầu không âm

Bài toán đấu thầu nhiều vòng được cài đặt trong lớp **MultiRoundBidProblem.**

## 3.2. Các tiêu chí đánh giá

Như đã trình bày, bài toán tối ưu hoá đa mục tiêu thường không tìm được chính xác một lời giải tốt nhất mà kỳ vọng đạt được là tìm được một tập các lời giải tốt (tập Pareto). Khi đã tìm được một tập Pareto, để đánh giá lời giải nào là tốt hơn, cần đưa ra một công thức (một hàm của các hàm mục tiêu) để tính ra được một giá trị đặc trưng cho độ "**tốt**" của lời giải đó, giá trị này gọi là **payoff**. Ngoài ra, để đánh giá một thuật toán là tốt hay không, ngoài việc dựa vào giá trị payoff của các lời giải tìm được, phân phối lời giải cũng là một yếu tố thường được quan tâm đến. Tuy nhiên, trong khuôn khổ bài tập lớn này, chúng em chỉ dùng giá trị payoff của lời giải để so sánh, đánh giá thuật toán (kèm theo thời gian chạy).

Công thức tính payoff :

* Với mỗi mục tiêu, tìm giá trị *fmin, fmax* như đối với bài toán tối ưu hoá đơn mục tiêu.
* Với mỗi lời giải, “**độ** **tốt**” tại từng mục tiêu *f* được tính theo công thức:
* với tối ưu hoá min
* với tối ưu hoá max
* Pay-off của lời giải được tính bằng tổng “độ tốt” tại từng mục tiêu nhân trọng số tương ứng. Như vậy, giá trị payoff nằm trong khoảng [0; 1] và càng gần 0 thì lời giải càng tốt.

Lời giải tốt nhất mà thuật toán tìm được (sau một số lần chạy) sẽ dùng để so sánh với các thuật toán khác.

## 3.3. Kết quả thử nghiệm

Chương trình được thử nghiệm trên 2 bộ dữ liệu đầu vào (gồm 1 bộ dữ liệu được cung cấp và 1 bộ dữ liệu tự tạo). Với một bộ dữ liệu, mỗi thuật toán được chạy 20 lần để tìm ra lời giải tốt nhất trong tất cả các lần, thời gian chạy của thuật toán được tính bằng thời gian trung bình của 20 lần chạy. Quần thể được khởi tạo ngẫu nhiên gồm 100 cá thể, số cá thể tối đa được tạo ra trong một lần chạy là 10000 (MOEA Framework yêu cầu đặt một giới hạn cho số cá thể tối đa được tạo ra).

Môi trường thử nghiệm :

* Windows 10
* Intel Core i5 6200U (Skylake), 2.7GHz, RAM 8GB.
* Java 11

Kết quả thử nghiệm cụ thể được lưu trong file đính kèm. Dưới đây là bảng so sánh thời gian tính (giây) và giá trị payoff của lời giải tốt nhất mà các thuật toán tìm được:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | NSGA-III  *O(mN2)* | **ε**-MOEA  *O(mN2)* | GDE3 | PESA2  *O(Np)* | IBEA  *O(pN2)* | SMPSO  *O(pnlog(n))* |
| Bộ dữ liệu 1 | | | | | | |
| Thời gian | 5.255 | 7.285 | 6.169 | 4.881 | 6.048 | 4.758 |
| Payoff | 0.3159 | 0.3036 | 0.3157 | 0.3040 | 0.3035 | 0.3097 |
| Bộ dữ liệu 2 | | | | | | |
| Thời gian | 6.604 | 8.951 | 7.998 | 6.119 | 7.099 | 6.424 |
| Payoff | 0.2280 | 0.2349 | 0.2144 | 0.2253 | 0.2077 | 0.2077 |

Bảng 1. Kết quả thử nghiệm các thuật toán

## 3.4. Nhận xét và đánh giá

Từ kết quả thử nghiệm trên, IBEA là thuật toán cho ra kết quả tốt nhất đổi với 2 bộ dữ liệu, mặt khác PESA2 và SMPSO là các thuật toán có thời gian chạy nhanh nhất. Ngoài ra, ta có một só nhận xét, đánh giá như sau:

* Thứ tự thời gian tính toán khá tương đồng nhau với 2 bộ dữ liệu.
* ε-MOEA và NSGA-III tuy có cùng độ phức tạp, tuy nhiên ε-MOEA việc tính toán ε-dominate tốn thời gian hơn nhiều so với dominate thông thường nên có sự chênh lệch về thời gian tính.
* Trong cả 2 bộ dữu liệu, IBEA, NSGA-III đều lần lượt là các thuật toán có kết quả tốt nhất và tồi nhất. NSGA-III dựa trên cơ sở đơn giản (non-dominate sorting) tuy nhiên rất dễ bị hội tụ vào cực trị địa phương (dù có một số cải tiến so với NSGA).
* ε-MOEA và PESA2 đều có điểm chung là chia không gian mục tiêu thành các hyperbox, hiệu quả thuật toán phụ thuộc vào độ rộng của không gian mục tiêu nên có sự khác nhau về thứ tự kết quả so với các thuật toán khác trên 2 bộ dữ liệu.
* Tương tự, SMPSO và GDE3 có điểm chung là chọn lọc không qua giao phối, mà dựa trên việc biến đổi từng cá thể đơn lẻ (gây đột biến, hoặc biến đổi có quy tắc) nên thứ tự kết quả có sự tương đồng trên 2 bộ dữ liệu.

KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

## 1. Kết luận

Qua quá trình thử nghiệm, thuật toán IBEA cho giá trị tốt nhất đối với bộ dữ liệu mà nhóm được cung cấp, đồng thời đã rút ra một số nhận xét về kết quả, đánh giá những đặc trưng riêng của từng thuật toán có thể làm ảnh hưởng đến kết quả thu được.

## 2. Định hướng phát triển

## Trong thời gian tời, đề tài tiếp tục nghiên cứu, tìm hiểu các thuật toán đã được cải tiến, thực hiện hiệu chỉnh các tham số cần thiết để cải tiến thời gian chạy và tối ưu giá trị Payoff của các thuật toán.

Tiếp tục tìm hiểu và thử nghiệm các thuật toán mới, đồng thời xây dựng và thu thập thêm các bộ dữ liệu kiểm thử mới và thực hiện thêm thử nghiệm để xác định các thuật toán phù hợp với từng bộ dữ liệu khác nhau.

Nhóm đề xuất một hướng tiếp cận khác đối với bài toán đó là quy bài toán tối ưu hoá đa mục tiêu về bài toán tối ưu hoá đơn mục tiêu (hàm mục tiêu chính là hàm tính payoff), như vậy, ta có thể tận dụng được những lợi thế:

* Giảm thời gian cho việc so sánh để xác định quan hệ dominate.
* Giảm số mục tiêu xuống còn 1, trực tiếp giảm độ phức tạp của một số thuật toán.
* Có thể áp dụng các thuật toán, kỹ thuật trong bài toán tối ưu hoá đơn mục tiêu để giải.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1] Phạm Minh Chuẩn, *Hệ thống gợi ý sử dụng thuật toán tối ưu bầy đàn,* Kỷ yếu Hội Nghị Quốc gia lần thứ VIII về nghiên cứu cơ bản và ứng dụng CNTT, 2015, p.122-134.

**Tiếng Anh**

[1] ZITZLER, Eckart; KÜNZLI, Simon. Indicator-based selection in multiobjective search. In: *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, Heidelberg, 2004. p. 832-842.

[2] NEBRO, Antonio J., et al. Smpso: A new pso-based metaheuristic for multi-objective optimization. In: *Computational intelligence in multi-criteria decision-making, 2009. mcdm'09. ieee symposium on*. IEEE, 2009. p. 66-73.

[3] DEB, Kalyanmoy, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 2002, 6.2: 182-197.

[4] CORNE, David W.; KNOWLES, Joshua D.; OATES, Martin J. The Pareto envelope-based selection algorithm for multiobjective optimization. In: *International conference on parallel problem solving from nature*. Springer, Berlin, Heidelberg, 2000. p. 839-848.

[5] COELLO, CA Coello. Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine*, 2006, 1.1: 28-36.

[6] DEB, Kalyanmoy; MOHAN, Manikanth; MISHRA, Shikhar. Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary computation*, 2005, 13.4: 501-525.

[7] HADKA, David. Moea framework-a free and open source java framework for multiobjective optimization. version 2.11. *URL http://www. moeaframework. org* [access on November 1st], 2015.

[8] Li, W., Özcan, E., John, R., Drake, J.H., Neumann, A. and Wagner, M., 2017, June. A modified indicator-based evolutionary algorithm (mIBEA). In *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 1047-1054). IEEE.

[9] Atashpendar, A., Dorronsoro, B., Danoy, G., & Bouvry, P. (2016). A parallel cooperative coevolutionary SMPSO algorithm for multi-objective optimization. In *IEEE International Conference on High Performance Computing Simulation (HPCS)* (pp. 713-720).