

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN



TỐI ƯU DANH MỤC ĐẦU TƯ MỜ VÀ ỨNG DỤNG

ĐỒ ÁN II

Chuyên ngành: Hệ Thống Thông Tin Quản Lý

Giảng viên hướng dẫn: TS. Trần Ngọc Thắng

Sinh viên thực hiện: Trần Thị Thanh Tâm

Lớp: HTTTQL 02 K67

Mã số sinh viên: 20227261

HÀ NỘI - 2026

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

1. Mục tiêu và nội dung của đề án

2. Kết quả đạt được

3. Ý thức làm việc của sinh viên

Hà Nội, ngày 5 tháng 1 năm 2026
Giảng viên hướng dẫn

Lời cảm ơn

Trong quá trình thực hiện đồ án, em đã nhận được sự hướng dẫn và hỗ trợ quý báu từ các thầy cô giáo. Em xin bày tỏ lòng biết ơn chân thành đến TS. Trần Ngọc Thăng - giảng viên khoa Toán - Tin, Trường Đại học Bách khoa Hà Nội - người đã tận tình chỉ bảo, định hướng và giúp đỡ em trong suốt thời gian thực hiện đề tài.

Em cũng xin chân thành cảm ơn các thầy cô trong Khoa Toán - Tin nói riêng và Đại học Bách khoa Hà Nội nói chung, những người đã giảng dạy, truyền đạt cho em những kiến thức quý báu, giúp em có nền tảng lý thuyết vững vàng để hoàn thành đồ án này.

Mặc dù đã cố gắng hoàn thiện, nhưng do giới hạn về thời gian và kinh nghiệm thực tế, đồ án không tránh khỏi những thiếu sót. Em mong nhận được sự góp ý từ các thầy cô để có thể tiếp tục hoàn thiện bản thân và nâng cao năng lực chuyên môn trong tương lai.

Hà Nội, ngày 5 tháng 1 năm 2026
Sinh viên thực hiện

Trần Thị Thanh Tâm

Tóm tắt nội dung đề án

Đề án “**Tối ưu danh mục đầu tư mờ và ứng dụng**” tập trung nghiên cứu và áp dụng lý thuyết tập mờ vào bài toán tối ưu hóa danh mục đầu tư tài chính. Việc lựa chọn đề tài xuất phát từ nhu cầu thực tiễn trong quản lý danh mục, khi thị trường tài chính luôn tồn tại tính không chắc chắn, biến động mạnh và nhiều rủi ro khó mô hình hóa chính xác bằng các phương pháp toán học truyền thống. Các mô hình cổ điển thường giả định dữ liệu chính xác và phân phối xác suất ổn định, trong khi thực tế thị trường chứa đựng nhiều yếu tố mơ hồ, nhiều và rủi ro cực đoan.

Lý thuyết tập mờ cho phép mô hình hóa hiệu quả tính không chắc chắn của dữ liệu tài chính, từ đó hỗ trợ xây dựng một cách tiếp cận quản trị rủi ro toàn diện hơn. Bên cạnh đó, việc kết hợp các phương pháp tối ưu hiện đại còn giúp cải thiện hiệu năng tính toán, giảm thời gian xử lý và nâng cao khả năng ứng dụng trong thực tế, đặc biệt phù hợp với điều kiện thị trường chứng khoán Việt Nam.

Nội dung chính của đề án bao gồm các phần sau:

- **Cơ sở lý thuyết về số mờ:** Trình bày tổng quan về logic mờ và các dạng số mờ phổ biến như số mờ tam giác, số mờ hình thang và số mờ Gaussian.
- **Xây dựng bài toán tối ưu đa mục tiêu:** Bài toán được xây dựng với mục tiêu tối ưu hóa đồng thời hai chỉ số tài chính quan trọng là tỉ số Sharpe (Sharpe Ratio – SR) và tỉ số VaR (Value at Risk Ratio – VR). Mục tiêu là xác định tập nghiệm Pareto tối ưu, qua đó đạt được sự cân bằng giữa lợi nhuận kỳ vọng, rủi ro biến động và rủi ro cực đoan trong danh mục đầu tư.
- **Phương pháp giải quyết:** Đề án tiến hành so sánh hiệu quả của các giải thuật tiến hóa đa mục tiêu phổ biến như NSGA-II, NSGA-III, MOEA/D, AGE-MOEA-II và SMS-EMOA. Đồng thời, tác giả đề xuất một hướng tiếp cận mới dựa trên gradient, cụ thể là phương pháp ϵ -constraint kết hợp với thuật toán SLSQP, nhằm nâng cao tốc độ hội tụ và hiệu quả tính toán.
- **Thực nghiệm và ứng dụng:** Đề án sử dụng dữ liệu giá đóng cửa của 60 mã cổ phiếu trên thị trường chứng khoán Việt Nam trong giai đoạn từ năm 2023 đến năm 2025 để kiểm chứng các phương pháp đề xuất. Kết quả thực nghiệm cho thấy phương pháp ϵ -constraint kết hợp SLSQP có ưu thế vượt trội về tốc độ xử lý (nhanh hơn nhiều lần so với các giải thuật heuristic), đồng thời vẫn đảm bảo chất lượng nghiệm và khả năng tìm kiếm tập nghiệm Pareto hiệu quả.

Danh mục viết tắt

Viết tắt	Giải thích
SR	Tỉ số Sharpe (Sharpe Ratio)
TFN	Số mờ tam giác (Triangular Fuzzy Number)
TrFN	Số mờ hình thang (Trapezoidal Fuzzy Number)
GFN	Số mờ Gaussian (Gaussian Fuzzy Number)
GrC	Tính toán hạt (Granular Computing)
TFIG	Lý thuyết tạo hạt thông tin mờ (Theories of Fuzzy Information Granulation)
FIG	Tạo hạt thông tin mờ (Fuzzy Information Granulation)
VaR	Rủi ro mất mát cực đoan trong điều kiện xấu (Value at Risk)
VR	Tỉ số lợi nhuận trên mỗi đơn vị rủi ro VaR (Value at Risk Ratio)
HV	Chỉ số đánh giá chất lượng tập nghiệm Pareto (Hypervolume)
NSGA	Thuật toán di truyền sắp xếp không trội (Non-dominated Sorting Genetic Algorithm)
NSGA-II, NSGA-III	Các phiên bản cải tiến của thuật toán NSGA
MOEA/D	Thuật toán tiến hóa đa mục tiêu dựa trên phân rã (Multiobjective Evolutionary Algorithm Based on Decomposition)
AGE-MOEA, AGE-MOEA-II	Thuật toán tiến hóa đa mục tiêu dựa trên hướng dẫn xấp xỉ (Approximation-Guided Evolutionary Multi-Objective Algorithm)
SMS-EMOA	Thuật toán tiến hóa đa mục tiêu dựa trên lựa chọn chỉ số S (Symmetric Selection Evolutionary Multi-Objective Algorithm)
SLSQP	Giải thuật quy hoạch bình phương tối thiểu tuần tự (Sequential Least Squares Programming)
QP	Quy hoạch bậc hai (Quadratic Programming)
EP	Quần thể lưu trữ bên ngoài (External Population)

Bảng 1. Danh mục viết tắt sử dụng trong đồ án

Danh sách bảng

1	Danh mục viết tắt sử dụng trong đồ án	4
1.1	Bảng điểm môn Toán theo số báo danh (Nguồn: Kaggle)	15
2.1	Lợi suất của 20 chứng khoán dưới dạng số mờ tam giác	26
2.2	Bảng các chỉ tiêu đánh giá (giá trị max)	27
2.3	Bảng tổng hợp kết quả so sánh các thuật toán	27
2.4	Kết quả thực nghiệm của mô hình	31
3.1	Dữ liệu ACB.VN	34
3.2	Dữ liệu lợi nhuận cổ phiếu (1/2023)	34
4.1	So sánh thời gian thực hiện các thuật toán tối ưu đa mục tiêu khi có cùng giá trị Hypervolume	47

Danh sách hình vẽ

1.1	Tập chính xác	10
1.2	Tập mờ	10
1.3	Số mờ Tam Giác	11
1.4	Số mờ Hình Thang	12
1.5	Số mờ Gaussian	13
2.1	Quy trình chọn lọc NSGA-II	21
2.2	15 điểm tham chiếu được hiển thị trên mặt phẳng tham chiếu chuẩn hóa cho bài toán ba mục tiêu với $p = 4$	22
2.3	So sánh pareto của 5 thuật toán	27
2.4	Lưu đồ thuật toán Phương pháp tối ưu đa mục tiêu theo ε -constraint kết hợp SLSQP	30
2.5	Pareto fronts Phương pháp tối ưu đa mục tiêu theo ε -constraint kết hợp SLSQP	31
3.1	Biểu diễn số mờ cho mã cổ phiếu ACB.VN	36
3.2	So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ tam giác	38
3.3	Đồ thị hội tụ Hypervolume theo thể hệ trong trường hợp số mờ tam giác	38
3.4	So sánh thời gian chạy của các thuật toán trong trường hợp số mờ tam giác	39
3.5	Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ tam giác	39
3.6	So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ hình thang	41
3.7	Đồ thị hội tụ Hypervolume theo thể hệ trong trường hợp số mờ hình thang	41
3.8	So sánh thời gian chạy của các thuật toán trong trường hợp số mờ hình thang	42
3.9	Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ hình thang	42
3.10	So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ Gaussian	44
3.11	Đồ thị hội tụ Hypervolume theo thể hệ trong trường hợp số mờ Gaussian	44
3.12	So sánh thời gian chạy của các thuật toán trong trường hợp số mờ Gaussian	45

3.13	Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ Gaussian	45
4.1	Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ tam giác	46
4.2	Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ hình thang	46
4.3	Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ Gaussian	46

Mục lục

Danh mục viết tắt	4
Danh mục bảng	5
Danh mục hình	6
1 Lý thuyết số mờ áp dụng trong lựa chọn danh mục đầu tư phù hợp	10
1.1 Tổng quan về hàm mờ	10
1.1.1 Định nghĩa	10
1.1.2 Các dạng số mờ phổ biến	10
1.2 Phương pháp mờ hóa	13
2 Bài toán tối ưu danh mục đầu tư theo 2 chỉ số SR và VR	17
2.1 Sharpe ratio và Value at Risk ratio trong bài toán tối ưu đa mục tiêu	17
2.1.1 Tổng quan	17
2.2 Các phương pháp giải thuật heuristic	19
2.2.1 Thuật toán NSGA-II, NSGA-III	20
2.2.2 Thuật toán dùng phương pháp phân rã MOEA/D	22
2.2.3 AGE-MOEA-II	24
2.2.4 SMS-EMOA	25
2.3 Phương pháp tối ưu đa mục tiêu theo ε -constraint	27
2.3.1 Nhận xét tính giả lõm ngặt của hàm mục tiêu	27
2.3.2 Phương pháp tối ưu đa mục tiêu theo ε -constraint kết hợp thuật toán SLSQP	28
3 Bài Toán Thực Tế	32
3.1 Thu Thập Dữ Liệu và xử lý	32
3.2 So sánh hiệu quả giữa các phương pháp	36
3.2.1 Hàm tam giác	36

3.2.2	Số mờ hình thang	40
3.2.3	Số mờ Gaussian	43
4	Tổng kết và so sánh	46

Chương 1: Lý thuyết số mờ áp dụng trong lựa chọn danh mục đầu tư phù hợp

1.1 Tổng quan về hàm mờ

1.1.1 Định nghĩa

Trong toán học cổ điển, khi xét trạng thái của a có thuộc vào tập hợp A hay không thì ta có hàm thuộc:

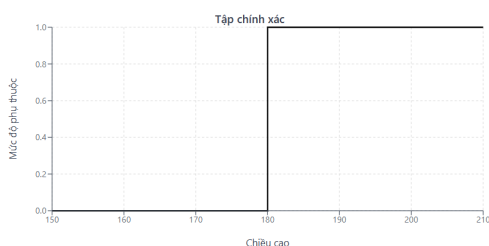
$$\mu_A(a) = \begin{cases} 0, & \text{nếu } a \notin A, \\ 1, & \text{nếu } a \in A. \end{cases}$$

Nghĩa là a chỉ có thể thuộc hoặc không thuộc vào tập A . Tuy nhiên, trong thực tế thì nhiều thông tin có thể được đánh giá nhưng ở một mức độ không rõ ràng (ví dụ: Anh ấy khá thông minh, tốc độ của xe máy rất nhanh,...). Logic mờ ra đời để biểu diễn các thông tin không rõ ràng này. Logic mờ sử dụng các công thức toán học cho phép biểu diễn tri thức dựa trên mức độ phụ thuộc:

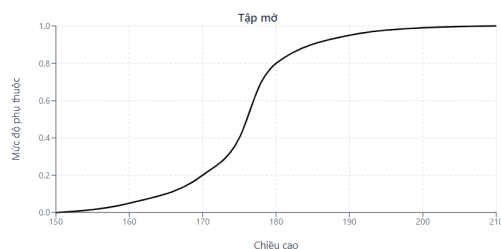
- Hoàn toàn đúng: 1 (True)
- Hoàn toàn sai: 0 (False)
- Đúng ở một mức độ nào đó: $x \in (0; 1)$

Như vậy, $\mu_A(a)$ của hàm mờ là một hàm liên tục, nhận giá trị trong khoảng $[0; 1]$

So sánh mức độ thuộc của tập chính xác và tập mờ



Hình 1.1. Tập chính xác



Hình 1.2. Tập mờ

1.1.2 Các dạng số mờ phổ biến

Số mờ tam giác (Triangular Fuzzy Number – TFN)

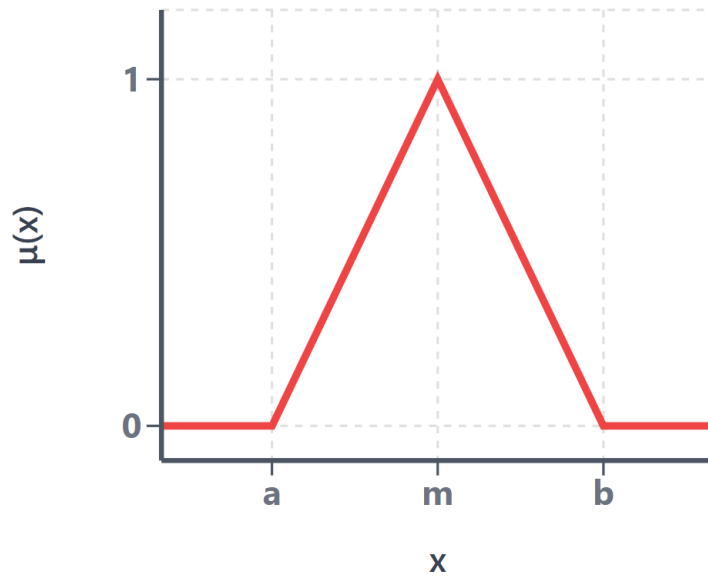
Một số mờ tam giác được ký hiệu:

$$\tilde{A} = (a, b, c), \quad a \leq b \leq c$$

Hàm thuộc:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x > c \end{cases}$$

- a : giá trị nhỏ nhất
- b : giá trị trung tâm (độ thuộc = 1)
- c : giá trị lớn nhất



Số Mờ Tam Giác

Hình 1.3. Số mờ Tam Giác

Số mờ hình thang (Trapezoidal Fuzzy Number – TrFN)

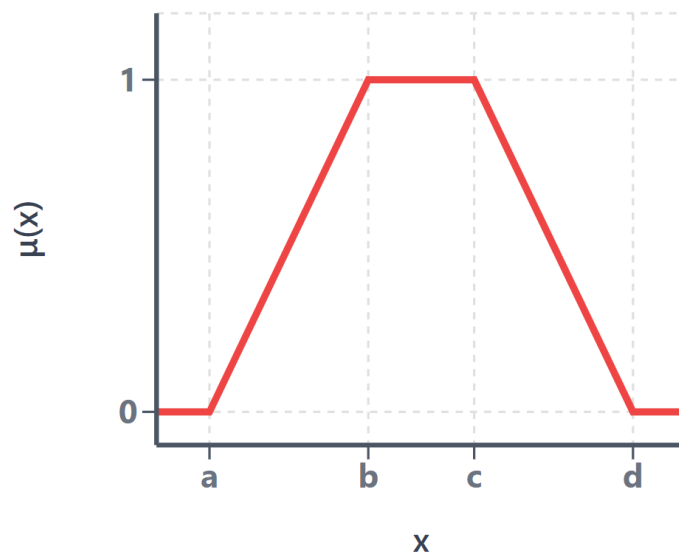
Một số mờ hình thang:

$$\tilde{A} = (a, b, c, d), \quad a \leq b \leq c \leq d$$

Hàm thuộc:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x > d \end{cases}$$

Khoảng $[b, c]$ là “đỉnh bằng” (độ thuộc = 1).



Số Mờ Hình Thang

Hình 1.4. Số mờ Hình Thang

Số mờ Gaussian (Gaussian Fuzzy Number – GFN)

Một số mờ Gaussian được biểu diễn:

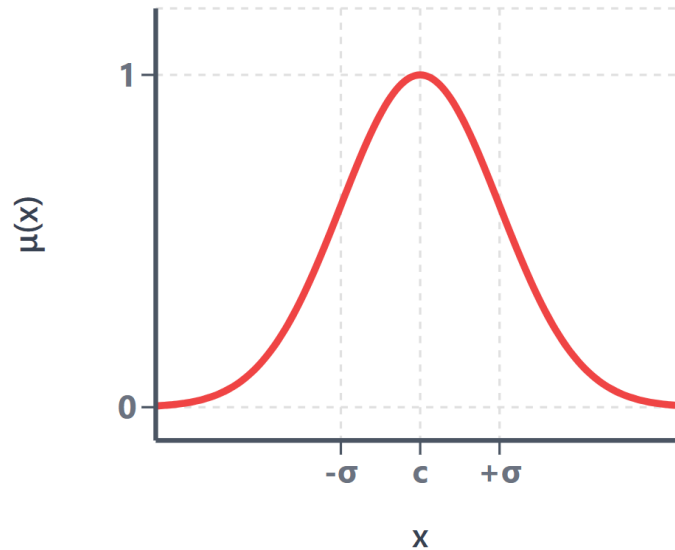
$$\tilde{A} = (m, \sigma)$$

- m : giá trị trung tâm
- σ : độ lệch chuẩn (xác định độ rộng)

Hàm thuộc:

$$\mu_{\tilde{A}}(x) = e^{-\frac{(x-m)^2}{2\sigma^2}}$$

Độ thuộc giảm dần theo khoảng cách từ trung tâm m .



Số Mờ Gaussian

Hình 1.5. Số mờ Gaussian

1.2 Phương pháp mờ hóa

Mờ hóa (fuzzification) là quá trình chuyển đổi các giá trị dữ liệu rõ (crisp values) thành các giá trị mờ thông qua hàm thuộc (membership function). Thay vì biểu diễn một đại lượng bằng một giá trị xác định, mờ hóa cho phép mô tả mức độ thuộc của đại lượng đó đối với một tập mờ, với giá trị thuộc nằm trong khoảng $[0,1]$. Cách tiếp cận này đặc biệt phù hợp trong các bài toán có tính không chắc chắn, nhiều hoặc khó xác định ranh giới chính xác giữa các trạng thái.

Trong nhiều nghiên cứu, các tham số của hàm thuộc thường được lựa chọn dựa trên kinh nghiệm hoặc đánh giá chủ quan của chuyên gia. Tuy nhiên, cách tiếp cận này có thể dẫn đến sai lệch và thiếu tính khách quan. Do đó, đề án này sẽ sử dụng phương pháp mờ hóa dựa trên thống kê dữ liệu, trong đó các tham số của hàm mờ được xác định trực tiếp từ các đặc trưng thống kê của dữ liệu thực nghiệm.

Cụ thể, các tham số của hàm thuộc được suy ra từ các đại lượng thống kê mô tả như giá trị nhỏ nhất, trung vị, tứ phân vị, giá trị lớn nhất, kỳ vọng và độ lệch chuẩn. Phương pháp này đảm bảo rằng các tập mờ phản ánh đúng phân bố thực tế của dữ liệu, đồng thời giảm thiểu sự can thiệp chủ quan trong quá trình xây dựng mô hình

mờ.

Hàm mờ tam giác (Triangular Membership Function)

Hàm mờ tam giác được xác định bởi ba tham số (a, b, c) , trong đó a và c lần lượt là biên trái và biên phải của tập mờ, còn b là giá trị trung tâm. Hàm thuộc có dạng:

$$\mu(x) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ \frac{c-x}{c-b}, & b < x < c, \\ 0, & x \geq c. \end{cases} \quad (1.1)$$

Trong phương pháp mờ hóa dựa trên thống kê dữ liệu, các tham số (a, b, c) được xác định lần lượt bởi giá trị nhỏ nhất, trung vị và giá trị lớn nhất của dữ liệu mẫu. Cách lựa chọn này giúp phản ánh vị trí điển hình của dữ liệu và phạm vi biến thiên tổng thể.

Hàm mờ hình thang (Trapezoidal Membership Function)

Hàm mờ hình thang được mô tả bởi bốn tham số (a, b, c, d) , trong đó khoảng $[b, c]$ biểu thị vùng có mức độ thuộc bằng 1. Hàm thuộc được xác định như sau:

$$\mu(x) = \begin{cases} 0, & x \leq a, \\ \frac{x-a}{b-a}, & a < x \leq b, \\ 1, & b < x \leq c, \\ \frac{d-x}{d-c}, & c < x < d, \\ 0, & x \geq d. \end{cases} \quad (1.2)$$

Trong nghiên cứu này, các tham số (a, b, c, d) được xác định từ các tứ phân vị của dữ liệu, cụ thể là giá trị nhỏ nhất, tứ phân vị thứ nhất Q_1 , tứ phân vị thứ ba Q_3 và giá trị lớn nhất. Cách tiếp cận này cho phép giảm ảnh hưởng của các giá trị ngoại lai và làm nổi bật vùng giá trị có độ tin cậy cao.

Hàm mờ Gaussian (Gaussian Membership Function)

Hàm mờ Gaussian được xây dựng dựa trên phân bố chuẩn, với hai tham số là kỳ vọng μ và độ lệch chuẩn σ . Hàm thuộc có dạng:

$$\mu(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \quad (1.3)$$

Trong phương pháp mờ hóa dựa trên thống kê dữ liệu, hai tham số μ và σ được ước

lượng lần lượt bằng giá trị trung bình và độ lệch chuẩn của dữ liệu mẫu. Hàm mờ Gaussian phù hợp với các tập dữ liệu có phân bố gần chuẩn và cho phép biểu diễn mức độ thuộc một cách trơn và liên tục.

Ví dụ: Ta xét điểm thi thử THPT quốc gia một lớp như sau:

Bảng 1.1. Bảng điểm môn Toán theo số báo danh (Nguồn: Kaggle)

SBD	Điểm	SBD	Điểm
1000001	8.4	1000026	7.2
1000002	8.6	1000027	8.2
1000003	8.2	1000028	8.8
1000004	4.8	1000029	9.0
1000005	8.6	1000030	8.0
1000006	5.8	1000031	6.8
1000007	6.0	1000032	8.2
1000008	8.2	1000033	8.6
1000009	8.2	1000034	7.8
1000010	7.4	1000035	8.0
1000011	7.2	1000036	6.4
1000012	8.2	1000037	7.6
1000013	5.2	1000038	8.2
1000014	7.6	1000039	7.2
1000015	7.6	1000040	7.0
1000016	6.0	1000041	6.2
1000017	7.0	1000042	7.0
1000018	8.2	1000043	8.0
1000019	8.2	1000044	9.0
1000020	7.6	1000045	8.8
1000021	8.4	1000046	7.8
1000022	7.6	1000047	4.8
1000023	8.8	1000048	6.8
1000024	8.4	1000049	7.6
1000025	7.8		

Tham số các hàm mờ được xác định từ dữ liệu:

- Hàm tam giác (Triangular MF)

Các giá trị đặc trưng: $\min = 4.8$, $\text{median} = 7.8$, $\max = 9$.

Tham số: $(a, b, c) = (4.8, 7.8, 9)$.

- Hàm hình thang (Trapezoidal MF)

Các tứ phân vị: $\min = 4.8$, $Q_1 = 7.0$, $Q_3 = 8.2$, $\max = 9$.

Tham số: $(a, b, c, d) = (4.8, 7.0, 8.2, 9)$.

- Hàm Gaussian (Gaussian MF)

Giá trị trung bình và độ lệch chuẩn: $\mu = 7.57$, $\sigma = 0.98$.

Hàm:

$$\mu(x) = \exp\left(-\frac{(x - 7.57)^2}{2(0.98)^2}\right).$$

Chương 2: Bài toán tối ưu danh mục đầu tư theo 2 chỉ số SR và VR

2.1 Sharpe ratio và Value at Risk ratio trong bài toán tối ưu đa mục tiêu

2.1.1 Tổng quan

Giả sử ta có một danh mục đầu tư p có n tài sản, mỗi tài sản có trọng số w_i ($\sum_{i=1}^n w_i = 1$). Lúc này, lợi suất kì vọng của danh mục đầu tư được tính theo công thức

$$E(R_p) = \sum X_i \cdot w_i$$

Với X_i là lợi suất của từng loại tài sản.

Phương sai của một danh mục ($V(X)$) là chỉ số đánh giá mức độ biên động của danh mục được tính theo công thức:

$$V(x) = \sum_{i=1}^n \sum_{j=1}^n \sigma_{ij} w_i w_j$$

với σ_{ij} là hiệp phương sai của 2 tài sản thứ i và j .

Tối ưu hóa danh mục đầu tư là quá trình xác định tỷ trọng các tài sản nhằm đạt được mức lợi nhuận kỳ vọng cao nhất tương ứng với một mức độ rủi ro nhất định. Giải pháp của vấn đề này chính là đa dạng hóa danh mục đầu tư chính là đa dạng hóa lĩnh vực đầu tư và điều chỉnh trọng số các tài sản

Tỉ số Sharpe (Sharpe Ratio) là một chỉ số tài chính dùng để đánh giá hiệu quả đầu tư so với mức độ rủi ro mà nhà đầu tư phải chấp nhận. Tỉ số Sharpe được định nghĩa như sau:

$$\text{Sharpe Ratio} = \frac{E(R_p) - R_f}{\sqrt{V(x)}}$$

Trong đó:

- $E(R_p)$: Lợi suất kỳ vọng của danh mục p ,
- R_f : Lãi suất phi rủi ro,
- $\sqrt{V(x)}$: phương sai của danh mục p .

Ta có thể xem biên động (volatility) là một thước đo rủi ro. Với mỗi đơn vị rủi

ro, tỉ số Sharpe thể hiện phần lợi suất vượt trội (*excess return*) mà nhà đầu tư nhận được so với mức sinh lời an toàn. Sharpe Ratio càng cao cho thấy danh mục đầu tư càng hiệu quả, vì mỗi đơn vị rủi ro mang lại mức lợi suất vượt trội lớn hơn.

Tiếp theo ta xét đến giá trị $Var(x)$ được định nghĩa là Rủi ro mất mát cực đoan trong điều kiện xấu:

$$VaR_{\beta}(x) = \inf\{x \in \mathbb{R} \mid P(X \leq x) \geq 1 - \beta\}.$$

$Var(x)$ tập trung vào đuôi phân phối – phần thể hiện các trường hợp xấu nhất cho biết: Trong điều kiện xấu, với xác suất $1 - \beta$, danh mục có thể lỗ tối đa bao nhiêu?

Mỗi chỉ số liên quan đến $Var(x)$ đó là VaR Ratio (Return-to-VaR Ratio) là một chỉ số đo lường “mức lợi nhuận nhận được trên mỗi đơn vị rủi ro VaR”.

$$VR = \frac{E(R_p) - R_f}{Var(x)}$$

Tối đa hóa đồng thời 2 tỷ số SR và VR mang lại một cách tiếp cận toàn diện hơn trong việc tối ưu hóa danh mục đầu tư, bằng cách nâng cao lợi nhuận đã điều chỉnh theo rủi ro, cải thiện sự ổn định của danh mục, hỗ trợ ra quyết định một cách sáng suốt hơn, và tăng cường mức độ tự tin của nhà đầu tư. Đề án này sẽ tập trung vào nghiên cứu bài toán tối ưu đồng thời 2 mục tiêu:

$$\begin{aligned} \max_{x \in X} \quad & (SR(x), VR(x)) \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1, \\ & 0 \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n. \end{aligned}$$

Ứng dụng trong trường hợp các số mờ

Trong trường hợp số mờ, ta sẽ cần tính toán lại các moment thống kê. Trong phạm vi của đề án, việc tính toán được thực hiện với loại số mờ tam giác, hình thang và Gaussian như sau:

Số mờ tam giác

$$E[A] = \frac{\alpha_1 + 4\alpha_2 + \alpha_3}{6},$$

$$V[A] = \frac{(\alpha_2 - \alpha_1)^2 + (\alpha_3 - \alpha_2)^2 + (\alpha_2 - \alpha_1)(\alpha_3 - \alpha_2)}{18},$$

$$\text{VaR}_\beta[A] = \begin{cases} 2\beta\alpha_2 + (1 - 2\beta)\alpha_1, & \beta \leq 0.5, \\ (2\beta - 1)\alpha_3 + (2 - 2\beta)\alpha_2, & \beta > 0.5. \end{cases}$$

Giả sử $A = (a, b, c, d)$ là một **số mờ hình thang (Trapezoidal Fuzzy Number)**

$$\mathbb{E}[A] = \frac{a + 2b + 2c + d}{6}. \quad (2.1)$$

$$\mathbb{V}[A] = \frac{(b + d - a - c)^2 + 6(b - c)^2 - 4(b - c)(b + d - a - c)}{24}. \quad (2.2)$$

$$\text{VaR}_\alpha(A) = \begin{cases} (1 - 2\alpha)a + 2\alpha b, & 0 < \alpha \leq 0.5, \\ (2 - 2\alpha)c + (2\alpha - 1)d, & 0.5 < \alpha < 1. \end{cases} \quad (2.3)$$

Nếu $A = \mathcal{N}(e, \sigma)$ là một **số mờ Gaussian**, thì:

$$\mathbb{E}[A] = e, \quad (2.4)$$

$$\mathbb{V}[A] = \sigma^2, \quad (2.5)$$

$$\text{VaR}_\beta[A] = e - (\ln(1 - \beta) - \ln \beta) \frac{\sqrt{6} \sigma}{\pi}. \quad (2.6)$$

2.2 Các phương pháp giải thuật heuristic

Giải thuật heuristic (heuristic algorithm hay thuật toán kinh nghiệm) là phương pháp giải quyết vấn đề dựa trên quy tắc kinh nghiệm, phán đoán hợp lý hoặc chiến lược rút gọn nhằm tìm ra lời giải “đủ tốt” trong thời gian ngắn, thay vì truy tìm lời giải tối ưu tuyệt đối. Heuristic đánh đổi giữa chất lượng nghiệm và tốc độ xử lý, cho phép tìm nghiệm gần đúng nhưng khả thi ngay cả với các bài toán có quy mô lớn. Phương pháp này đặc biệt hữu ích trong các bài toán tối ưu phức tạp như TSP, VRP, phân lô hàng hay lập lịch sản xuất — những bài toán có không gian nghiệm khổng lồ và thường thuộc nhóm NP-hard hoặc NP-complete. Với các bài toán như vậy, các phương pháp tối ưu chính xác thường không khả thi do thời gian tính toán tăng theo cấp số nhân, nên heuristic giúp thu hẹp phạm vi tìm kiếm và tiếp cận nhanh các vùng nghiệm tiềm năng.

2.2.1 Thuật toán NSGA-II, NSGA-III

NSGA (Non-dominated Sorting Genetic Algorithm) là một thuật toán di truyền (Genetic Algorithm) được thiết kế dành riêng cho bài toán tối ưu đa mục tiêu (multi-objective optimization). Thay vì chỉ tối ưu một hàm mục tiêu, NSGA tìm kiếm tập nghiệm Pareto — tức các nghiệm không bị trội (non-dominated) bởi nghiệm nào khác. Điều này giúp NSGA đồng thời tối ưu nhiều mục tiêu xung đột (ví dụ: tối thiểu chi phí nhưng tối đa chất lượng, tối thiểu rủi ro nhưng tối đa lợi nhuận,...).

Ý tưởng chính của NSGA là phân loại quần thể theo mức độ không trội (non-dominated sorting). Thuật toán chia quần thể thành nhiều “tầng” Pareto: tầng 1 gồm các nghiệm không bị trội bởi nghiệm nào; tầng 2 gồm các nghiệm chỉ bị trội bởi nghiệm tầng 1; và cứ thế tiếp tục. Các nghiệm ở tầng thấp (gần Pareto front) được ưu tiên chọn lọc hơn.

Thuật toán NSGA-II

NSGA-II là một phương pháp heuristic được phát triển vào đầu những năm 2000 nhằm giải quyết các vấn đề trước đó của NSGA như độ phức tạp tính toán, thiếu tính ưu tú và nhu cầu phải chỉ định tham số chia sẻ. Quy trình của NSGAII khá đơn giản và dễ thực hiện.

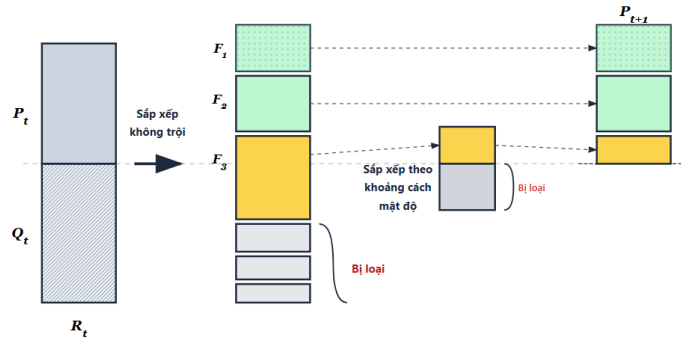
Ta xét một quần thể kết hợp

$$R_t = P_t \cup Q_t$$

Trong đó:

- P_t (Parent population): quần thể cha mẹ ở thế hệ t .
- Q_t (Offspring population): quần thể con được tạo ra từ P_t thông qua các toán tử di truyền như lai ghép (crossover) và đột biến (mutation).
- R_t là tất cả các cá thể trong quần thể cha mẹ và quần thể con tại thế hệ t .

Lúc này, R_t Là quần thể có kích thước $2N$ và vì các cá thể ưu tú của lần sinh trước và toàn thể cá thể của lần sinh này đều nằm trong tập R_t nên tính ưu tú của quần thể được đảm bảo. Tiếp theo ta sẽ tiến hành chia R_t thành các tập không trội theo thứ tự từ F_1 đến F_i (lấy F_i sao cho F_i là tập đầu tiên có tổng các nghiệm từ F_1 đến F_i lớn hơn N) rồi lấy N nghiệm theo thứ tự từ trên xuống, ở tập F_i để chọn chính xác N nghiệm thì ta cần sắp xếp các nghiệm trong tập toán tử so sánh dựa trên mật độ theo thứ tự giảm dần.



Hình 2.1. Quy trình chọn lọc NSGA-II

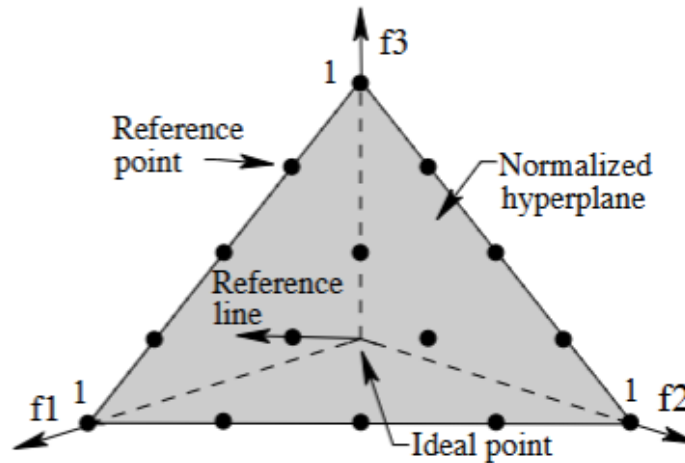
Phương pháp tiếp theo được phát triển từ phương pháp này đó là **NSGA-III** với phương pháp lựa chọn từ F_1 đến F_{i-1} giống như NSGA-II, với các giá trị còn lại trong tập F_i được xác định bằng cách sử dụng một tập hợp các điểm tham chiếu được xác định trước để đảm bảo tính đa dạng trong các giải pháp thu được. Các điểm tham chiếu được chọn có thể được xác định trước theo cách có cấu trúc hoặc được người dùng cung cấp theo sở thích. Trong trường hợp không có thông tin ưu tiên nào, bất kỳ vị trí có cấu trúc được xác định trước nào của các điểm tham chiếu đều có thể được áp dụng, phương pháp tiếp cận thường thấy nhất là phương pháp của Das và Dennis đặt các điểm trên một siêu mặt phẳng được chuẩn hóa – một hình đơn giản đơn vị $(M - 1)$ chiều – nghiêng đều với tất cả các trục mục tiêu và có điểm cắt là một trên mỗi trục. Nếu xem xét p phân chia dọc theo mỗi mục tiêu, tổng số điểm tham chiếu (H) trong một bài toán M mục tiêu được đưa ra:

$$H = \binom{M + p - 1}{p}.$$

Ví dụ: Với bài toán tối ưu 3 mục tiêu ($M=3$) và số lần chia $P=4$ ta có

$$H = \binom{3 + 4 - 1}{4} = 15.$$

vậy ta có 15 điểm tham chiếu trên siêu mặt phẳng



Hình 2.2. 15 điểm tham chiếu được hiển thị trên mặt phẳng tham chiếu chuẩn hóa cho bài toán ba mục tiêu với $p = 4$.

So với NSGA-II thì NSGA-III có sự đa dạng tốt hơn, phù hợp với bài toán tối ưu nhiều mục tiêu hơn và có đường Pareto đều hơn

2.2.2 Thuật toán dùng phương pháp phân rã MOEA/D

Bên cạnh các dòng thuật toán dựa trên sắp xếp không trội, một hướng tiếp cận khác cũng rất phổ biến là phương pháp phân rã, điển hình là thuật toán MOEA/D - phương pháp phân rã bài toán tối ưu đa mục tiêu thành các bài toán tối ưu hóa vô hướng con. Nó giải quyết bài toán này một cách đồng thời bằng cách tiến hóa một quần thể các giải pháp. Tại mỗi thế hệ, quần thể bao gồm các giải pháp tốt nhất từ trước đến nay cho mỗi bài toán con. Các mối quan hệ lân cận giữa các bài toán con này được xác định dựa trên khoảng cách giữa các véc-tơ hệ số tổng hợp (aggregation coefficient vectors) của chúng. Các giải pháp tối ưu cho hai bài toán con lân cận nên rất giống nhau. Mỗi bài toán con (tức là hàm tổng hợp vô hướng) được tối ưu hóa trong MOEA/D bằng cách chỉ sử dụng thông tin từ các bài toán con lân cận của nó. MOEA/D có các đặc điểm sau:

- Cung cấp một cách đơn giản nhưng hiệu quả để giới thiệu các phương pháp tiếp cận phân rã vào tính toán tiến hóa đa mục tiêu.
- Dễ dàng xử lý các vấn đề về gán độ thích nghi gán độ thích nghi (fitness assignment) và duy trì sự đa dạng (diversity maintenance) hơn so với các MOEA không dựa trên phân rã
- Có độ phức tạp tính toán tại mỗi thế hệ thấp hơn so với NSGA-II và MOGLS.

- Các kỹ thuật chuẩn hóa mục tiêu có thể được tích hợp vào MOEA/D để xử lý các mục tiêu có thang đo khác biệt (disparately scaled objectives).
- Việc sử dụng các phương pháp tối ưu hóa vô hướng trong MOEA/D là rất tự nhiên vì mỗi giải pháp được liên kết với một bài toán tối ưu hóa vô hướng.

Các bước của thuật toán:

Bước 1: Khởi tạo

- Bước 1.1: Đặt $EP = \emptyset$.
- Bước 1.2: Tính toán khoảng cách Euclid giữa hai vector trọng số bất kỳ và sau đó xác định T vector trọng số gần nhất với mỗi vector trọng số. Với mỗi $i = 1, \dots, N$, đặt

$$B(i) = \{i_1, \dots, i_T\},$$

trong đó $\lambda_{i_1}, \dots, \lambda_{i_T}$ là T vector trọng số gần nhất với λ_i .

- Bước 1.3: Tạo ra quần thể ban đầu x_1, \dots, x_N một cách ngẫu nhiên hoặc bằng một phương pháp cụ thể cho bài toán. Đặt

$$FV_i = F(x_i).$$

- Bước 1.4: Khởi tạo

$$z = (z_1, \dots, z_m)^T$$

bằng một phương pháp cụ thể cho bài toán.

Bước 2: Cập nhật

Với $i = 1, \dots, N$, thực hiện:

- Bước 2.1: Sinh sản: Chọn ngẫu nhiên hai chỉ số k, l từ $B(i)$, và sau đó tạo ra một nghiệm mới y từ x_k và x_l bằng cách sử dụng các toán tử di truyền.
- Bước 2.2: Cải tiến: Áp dụng một thủ thuật sửa chữa/cải tiến cụ thể cho bài toán lên y để tạo ra y' .
- Bước 2.3: Cập nhật z : Với mỗi $j = 1, \dots, m$, nếu

$$z_j < f_j(y'),$$

thì đặt

$$z_j = f_j(y').$$

- Bước 2.4: Cập nhật các láng giềng: Với mỗi chỉ số $j \in B(i)$, nếu

$$g_{tv}(y' \mid \lambda_j, z) \leq g_{tv}(x_j \mid \lambda_j, z),$$

thì đặt

$$x_j = y', \quad FV_j = F(y').$$

- Bước 2.5: Cập nhật EP :

Loại bỏ khỏi EP tất cả các vector bị chi phối bởi $F(y')$.

Thêm $F(y')$ vào EP nếu không có vector nào trong EP chi phối $F(y')$.

Bước 3: Tiêu chí dừng

Nếu tiêu chí dừng được thỏa mãn, thì dừng lại và xuất ra EP . Ngược lại, quay lại bước 2.

2.2.3 AGE-MOEA-II

AGE-MOEA (Approximation-Guided Evolutionary Multi-Objective Algorithm) là một thuật toán tiến hóa đa mục tiêu dựa trên chỉ số (indicator-based MOEA), được thiết kế đặc biệt cho bài toán many-objective (nhiều hơn 3 mục tiêu). Ý tưởng cốt lõi của AGE-MOEA không phải là tối ưu từng nghiệm riêng lẻ, mà là tối ưu chất lượng xấp xỉ của toàn bộ Pareto front. Thay vì so sánh nghiệm bằng quan hệ trội, AGE-MOEA sử dụng additive – indicator để đo lường mức độ mà tập nghiệm hiện tại xấp xỉ Pareto front lý tưởng. Trong quá trình chọn lọc, thuật toán áp dụng cơ chế loại nghiệm, trong đó mỗi nghiệm được đánh giá dựa trên mức suy giảm của – indicator nếu nghiệm đó bị loại khỏi quần thể. Nghiệm có ảnh hưởng nhỏ nhất đến chất lượng xấp xỉ Pareto front sẽ bị loại bỏ, trong khi các nghiệm đóng vai trò quan trọng trong việc duy trì độ bao phủ và đa dạng của Pareto front được giữ lại.

AGE-MOEA-II là phương pháp được phát triển từ AGE-MOEA dùng để mô hình hóa mặt trận không bị chi phối và để đo khoảng cách giữa các giải pháp không bị chi phối. Cụ thể, giải thuật này sử dụng phương pháp lặp Newton-Raphson để tìm nghiệm. Đây là phương pháp mạnh mẽ hơn nhiều và ít nhạy cảm hơn với việc lựa chọn điểm không bị chi phối được sử dụng làm tham chiếu. Sau đó, sửa dụng đường trắc địa để đo khoảng cách giữa 2 giải pháp không bị chi phối trên một đa tạp cong.

Kết quả thực nghiệm trong các nghiên cứu trước đây đã cho thấy AGE-MOEA-II đạt được giá trị Hypervolume (HV) cao hơn so với phiên bản tiền nhiệm AGE-MOEA và vượt trội hơn các MOEA tiên tiến khác, cụ thể là NSGA-III, MOEA/D, LMEA và VaEA.

Algorithm 1 AGE-MOEA-II: An improved front modeling-based MOEA

Require: M : Number of objectives

Require: N : Population size

Ensure: Final population P

```
1:  $P \leftarrow \text{RANDOM-POPULATION}(N)$ 
2: while not (stop_condition) do
3:    $Q \leftarrow \text{GENERATE-OFFSPRING}(P)$ 
4:    $\mathbb{F} \leftarrow \text{FAST-NONDOMINATED-SORT}(P \cup Q)$ 
5:    $\mathbb{F}_1 \leftarrow \text{NORMALIZE}(\mathbb{F}_1)$  {Eq. 9}
6:    $d \leftarrow 1$  {First non-dominated rank}
7:   while  $|P| + |\mathbb{F}_d| < N$  do
8:      $\mathbb{F}_\perp \leftarrow \text{MANIFOLD-PROJECTION}(\mathbb{F}_d, p)$ 
9:      $D \leftarrow \text{GEODESIC-DIV}(\mathbb{F}_\perp, p)$  {Eq. 15}
10:     $\text{SURVIVAL-SCORE}(D, \mathbb{F}, d, p)$ 
11:     $P \leftarrow P \cup \mathbb{F}_d$ 
12:     $d \leftarrow d + 1$ 
13:   end while
14:    $\text{SORT}(\mathbb{F}_d)$  {by survival scores}
15:    $P \leftarrow P \cup \mathbb{F}_d[1 : (N - |P|)]$ 
16: end while
17: return  $P$ 
```

2.2.4 SMS-EMOA

SMS-EMOA được thiết kế để đạt được Hypervolume tối đa với một số lượng điểm hữu hạn. Hơn nữa, để giảm thiểu vấn đề chọn điểm tham chiếu phù hợp và hướng tới một thuật toán đơn giản có thể được song song hóa theo cách không đồng bộ và mở rộng bằng các kỹ thuật tích hợp tri thức, chẳng hạn như các phép đánh giá hàm gần đúng. SMS-EMOA kết hợp các ý tưởng mượn từ các EMOA khác, như NSGA-II đã được thiết lập vững chắc và các chiến lược lưu trữ đã được trình bày bởi Knowles và cộng sự. Đây là một thuật toán ổn định dựa trên hai trụ cột: sắp xếp không bị chi phối (non-dominated sorting) và Hypervolume được áp dụng làm tiêu chí lựa chọn để loại bỏ cá thể đóng góp ít thể tích siêu lớn nhất vào mặt trận (front) xếp hạng tồi nhất.

Thuật toán cơ bản như sau:

Algorithm 2 SMS-EMOA: S-metric Selection Evolutionary Multi-Objective Algorithm

Require: μ : Population size

Ensure: Final population P

- 1: $P_0 \leftarrow \text{init}()$ {Initialize random population of μ individuals}
 - 2: $t \leftarrow 0$
 - 3: **while** not termination condition fulfilled **do**
 - 4: $q_{t+1} \leftarrow \text{generate}(P_t)$ {generate offspring by variation}
 - 5: $P_{t+1} \leftarrow \text{Reduce}(P_t \cup \{q_{t+1}\})$ {select μ best individuals}
 - 6: $t \leftarrow t + 1$
 - 7: **end while**
 - 8: **return** P_t
-

Bắt đầu với một quần thể khởi tạo gồm μ cá thể, một cá thể mới được tạo ra bằng cách sử dụng các toán tử biến đổi ngẫu nhiên. Cá thể mới sẽ trở thành thành viên của quần thể tiếp theo nếu việc thay thế một cá thể khác dẫn đến chất lượng quần thể cao hơn đối với chỉ số S .

Algorithm 3 Reduce(Q): Selection procedure for SMS-EMOA

Require: Q : Population with $\mu + 1$ individuals

Ensure: Population with μ individuals

- 1: $\{R_1, \dots, R_v\} \leftarrow \text{fast-nondominated-sort}(Q)$ {all v fronts of Q }
 - 2: $r \leftarrow \arg \min_{s \in R_v} [\Delta_S(s, R_v)]$ { $s \in R_v$ with lowest $\Delta_S(s, R_v)$ }
 - 3: **return** $Q \setminus \{r\}$ {eliminate detected element}
-

Reduce(Q) được dùng để chọn ra μ cá thể của quần thể tiếp theo sử dụng thuật toán fast-nondominated-sort đã được sử dụng trong NSGA-II được áp dụng để phân vùng quần thể thành v tập hợp theo phép sắp xếp không bị chi phối được định nghĩa bởi Goldberg. **Ví dụ:** Ta có dữ liệu các hàm mờ của 20 cổ phiếu

Bảng 2.1. Lợi suất của 20 chứng khoán dưới dạng số mờ tam giác

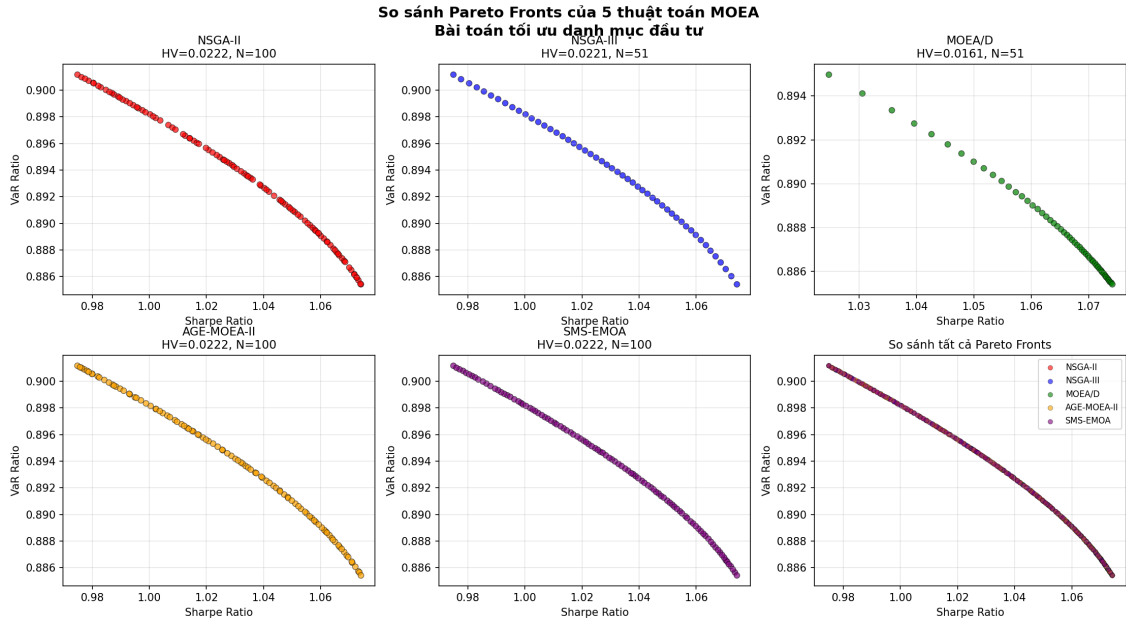
Mã	Lợi suất	Mã	Lợi suất
000001.SZ	(0.8803, 0.9923, 1.1083)	000016.SZ	(0.4476, 0.9854, 1.2594)
000002.SZ	(0.8796, 1.0018, 1.0838)	000017.SZ	(0.6494, 0.9999, 1.3699)
000004.SZ	(0.8408, 1.0150, 1.3210)	000018.SZ	(0.7589, 1.0089, 1.3729)
000005.SZ	(0.7130, 1.0324, 1.6114)	000019.SZ	(0.7176, 1.0150, 1.2800)
000006.SZ	(0.7548, 1.0110, 1.3230)	000020.SZ	(0.7585, 1.0268, 1.4698)
000009.SZ	(0.7637, 1.0053, 1.2073)	000021.SZ	(0.8993, 1.0100, 1.3110)
000010.SZ	(0.7686, 1.0025, 1.4505)	000022.SZ	(0.8251, 0.9878, 1.0248)
000011.SZ	(0.7894, 1.0141, 1.4221)	000023.SZ	(0.8114, 1.0044, 1.2644)
000012.SZ	(0.8446, 0.9888, 1.1218)	000024.SZ	(0.8529, 1.0335, 1.2825)
000014.SZ	(0.7490, 1.0127, 1.2777)	000025.SZ	(0.7177, 1.0868, 1.5928)

Ta có:

Chỉ tiêu	Giá trị lớn nhất
Sharpe ratio	537.035423
Var ratio	0.885423

Bảng 2.2. Bảng các chỉ tiêu đánh giá (giá trị max)

Có thể thấy giá trị Sharpe ratio lớn nhất gấp khoảng 500 lần so với Var ratio lớn nhất nên ta sẽ chia Sharpe ratio cho 500 để đảm bảo sự tương quan giữa hai tỉ số.



Hình 2.3. So sánh pareto của 5 thuật toán

Thuật toán	Hypervolume	Số nghiệm	Thời gian (s)
NSGA-II	0.022152	100	11.21
SMS-EMOA	0.022153	100	13.45
AGE-MOEA-II	0.022152	100	27.39
NSGA-III	0.022146	51	12.15
MOEA/D	0.016089	51	21.47

Bảng 2.3. Bảng tổng hợp kết quả so sánh các thuật toán

2.3 Phương pháp tối ưu đa mục tiêu theo ε -constraint

2.3.1 Nhận xét tính giả lồi ngặt của hàm mục tiêu

Xét miền khả thi $X \subset \mathbb{R}^n$ là một tập lồi, được xác định bởi các ràng buộc tuyến tính trên vector trọng số danh mục đầu tư. Hai hàm mục tiêu được xét trong bài toán là chỉ số Sharpe $SR(x)$ và Value-at-Risk ratio $VR(x)$, đều có dạng hàm phân số, trong đó tử số là hàm tuyến tính theo x và mẫu số là một thước đo rủi ro lồi theo x .

Cụ thể, chỉ số Sharpe được xác định bởi

$$SR(x) = \frac{\mathbb{E}[X] - r_f}{V(X)},$$

trong đó $\mathbb{E}[X]$ là hàm tuyến tính theo x , còn $V(X)$ là một hàm lồi và dương trên miền khả thi. Do đó, $SR(x)$ là một hàm phân số với tử số tuyến tính và mẫu số lồi dương, và được biết là một hàm giả lõm (pseudo-concave) trên miền khả thi X .

Tương tự, Value-at-Risk ratio được xác định bởi

$$VR(x) = \frac{\mathbb{E}[X] - r_f}{\text{VaR}(X)}.$$

Với giả thiết phân phối suất sinh lợi của danh mục là chuẩn (hoặc elip), $\text{VaR}(X)$ có dạng tuyến tính theo độ lệch chuẩn của danh mục, do đó là một hàm lồi và dương theo x . Khi đó, $VR(x)$ cũng là một hàm phân số với tử số tuyến tính và mẫu số lồi dương, và vì vậy là một hàm giả lõm trên miền khả thi X .

2.3.2 Phương pháp tối ưu đa mục tiêu theo ε -constraint kết hợp thuật toán SLSQP

Phương pháp ε -constraint được biết đến là phương pháp tối ưu một hàm mục tiêu, còn các hàm mục tiêu khác được sử dụng như các ràng buộc, và được đưa vào phần ràng buộc của mô hình như sau:

$$\max_x f_1(x) \quad \text{s.t.} \quad \begin{cases} f_2(x) \geq \varepsilon_2, \\ f_3(x) \geq \varepsilon_3, \\ \vdots \\ f_p(x) \geq \varepsilon_p, \\ x \in S. \end{cases} \quad (2.7)$$

Bằng cách thay đổi các tham số ở vế phải (ε_i), ta thu được tập các nghiệm hiệu quả (efficient solutions) của bài toán.

Trong đồ án này, phương pháp ε -constraint được áp dụng để giải bài toán tối ưu đa mục tiêu giữa chỉ số Sharpe mờ (SR) và Value-at-Risk ratio (VR). Tại mỗi lần lặp, một hàm mục tiêu được tối ưu hóa trong khi hàm còn lại được đưa vào dưới dạng ràng buộc ε . Bằng cách quét tham số ε trên miền giá trị khả thi, tập nghiệm Pareto được xây dựng.

Lúc này bài toán sẽ được đưa về:

$$\begin{aligned} \max_{x \in X} \quad & SR(x) \\ \text{vđk} \quad & VR(x) \geq \varepsilon, \\ & \sum_{i=1}^n x_i = 1, \\ & 0 \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n. \end{aligned}$$

và

$$\begin{aligned} \max_{x \in X} \quad & VR(x) \\ \text{vđk} \quad & SR(x) \geq \varepsilon, \\ & \sum_{i=1}^n x_i = 1, \\ & 0 \leq x_i \leq \bar{x}_i, \quad i = 1, \dots, n. \end{aligned}$$

Tính giả lỏm của các hàm mục tiêu $SR(x)$ và $VR(x)$ đã được nhận xét ở phần trước đã đảm bảo tính đúng đắn về mặt lý thuyết khi áp dụng các thuật toán tối ưu hóa dựa trên gradient, chẳng hạn như Sequential Least Squares Programming (SLSQP), để giải các bài toán con thu được từ phương pháp ε -constraint.

Phương pháp Sequential Least Squares Programming (SLSQP) giải bài toán tối thiểu một mục tiêu thông qua việc xây dựng và giải một chuỗi các bài toán con dạng quy hoạch bậc hai (Quadratic Programming – QP), thu được bằng cách xấp xỉ bậc hai hàm mục tiêu và tuyến tính hóa các ràng buộc tại mỗi vòng lặp.

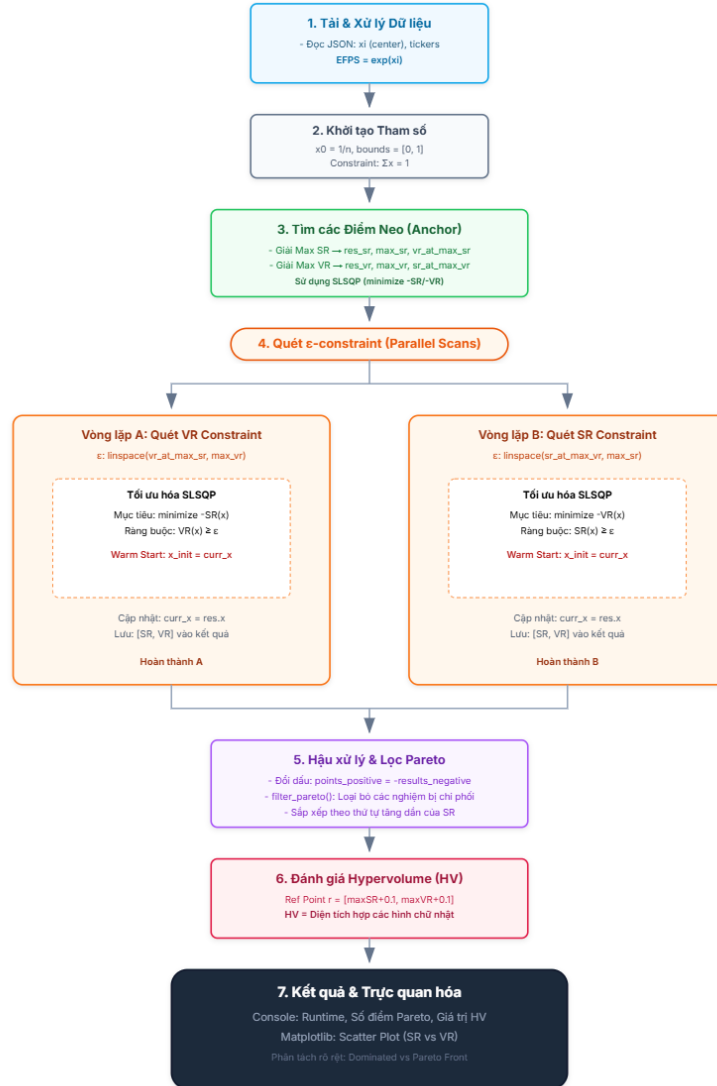
$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{vđk} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0}, \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}. \end{aligned} \tag{2.8}$$

Trong đó $f : \mathbb{R}^n \rightarrow \mathbb{R}$ là hàm mục tiêu, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{\text{eq}}}$ là các ràng buộc đẳng thức, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{\text{ineq}}}$ là các ràng buộc bất đẳng thức, và $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$ là các cận dưới và cận trên.

Tại điểm lặp \mathbf{x}_k , thuật toán xây dựng bài toán quy hoạch toàn phương:

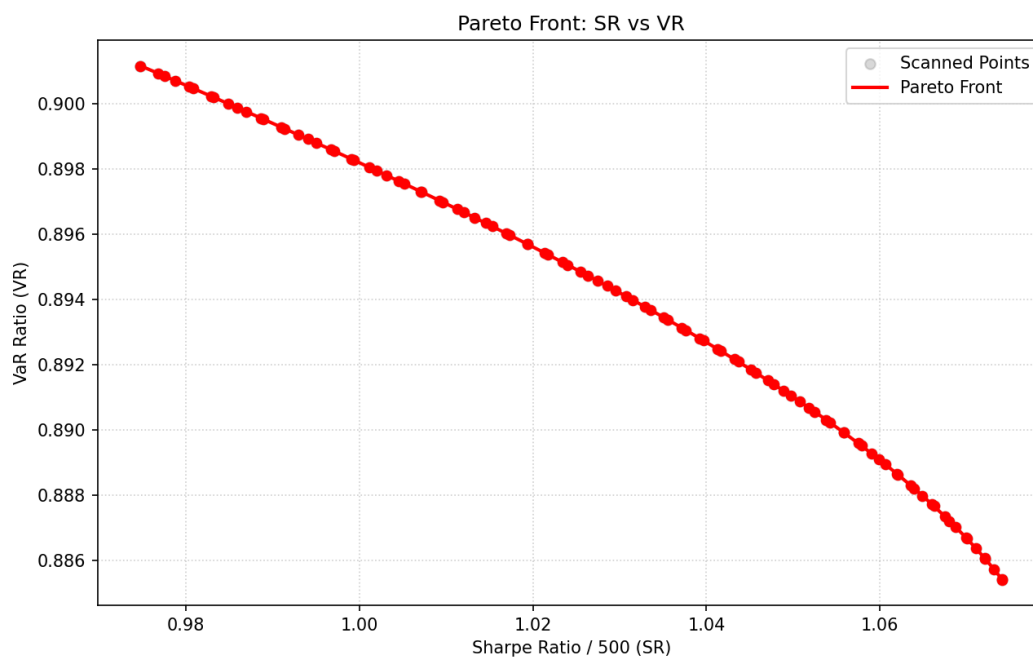
$$\begin{aligned}
\mathbf{d}_k &= \arg \min_{\mathbf{d} \in \mathbb{R}^n} \frac{1}{2} \mathbf{d}^T \mathbf{H}_k \mathbf{d} + \nabla f(\mathbf{x}_k)^T \mathbf{d} \\
\text{vđk} \quad &\mathbf{A}_k \mathbf{d} + \mathbf{c}_k = \mathbf{0}, \\
&\mathbf{l} \leq \mathbf{x}_k + \mathbf{d} \leq \mathbf{u}.
\end{aligned} \tag{2.9}$$

Từ đó ta xây dựng phương án giải cho bài toán này:



Hình 2.4. Lưu đồ thuật toán Phương pháp tối ưu đa mục tiêu theo ϵ -constraint kết hợp SLSQP

Áp dụng cho ví dụ ở trên ta có:



Hình 2.5. Pareto fronts Phương pháp tối ưu đa mục tiêu theo ε -constraint kết hợp SLSQP

Bảng 2.4. Kết quả thực nghiệm của mô hình

Chỉ tiêu	Giá trị
Thời gian chạy (giây)	0.36
Số điểm Pareto	99
Hypervolume	0.0221515650

Kết quả cho thấy rằng, với cùng một Hypervolume, thì phương pháp đề xuất có cải thiện về thời gian rất lớn

Chương 3: Bài Toán Thực Tế

Trong phạm vi của đề án này, dữ liệu giá cổ phiếu Việt Nam được thu thập trong khoảng thời gian 2 năm (từ 01/01/2023 đến 01/01/2025) với 60 mã cổ phiếu, được phân bổ ở nhiều nhóm ngành khác nhau như Ngân hàng, Bất động sản, Tiêu dùng, Năng lượng, Tài chính,... Sự đa dạng hóa này nhằm mục đích giảm thiểu mối tương quan giữa các cổ phiếu trong danh mục, từ đó phản ánh tốt hơn bài toán tối ưu hóa danh mục đầu tư trong thực tế.

3.1 Thu Thập Dữ Liệu và xử lý

Bước 1: Thu thập dữ liệu Dữ liệu được thu thập tự động bằng thư viện `yfinance` của Python, lấy giá đóng cửa (Close) hàng ngày của các mã cổ phiếu.

```
1 import yfinance as yf
2 import pandas as pd
3
4 tickers = [
5     # Banking
6     "ACB.VN", "BID.VN", "CTG.VN", "MBB.VN", "HDB.VN",
7     "STB.VN", "TCB.VN", "VCB.VN", "VIB.VN", "VPB.VN",
8
9     # Real Estate
10    "DIG.VN", "DXG.VN", "KDH.VN", "NLG.VN", "PDR.VN",
11
12    # Securities & Financial Services
13    "HCM.VN", "ORS.VN", "SSI.VN", "TVS.VN",
14    "VCI.VN", "VIX.VN", "VND.VN",
15
16    # Industrials & Materials
17    "BMP.VN", "DGC.VN", "DPG.VN", "DPM.VN",
18    "GEX.VN", "HPG.VN", "HSG.VN", "NKG.VN", "REE.VN",
19
20    # Consumer Goods, Agriculture & Fisheries
21    "ANV.VN", "DBC.VN", "FMC.VN", "HAG.VN",
22    "NAF.VN", "NSC.VN", "PAN.VN", "SBT.VN",
23    "SSC.VN", "VFG.VN", "VHC.VN",
24
25    # Energy & Oil Services
26    "PLX.VN", "PVT.VN",
27
28    # Technology & Telecommunications
29    "CMG.VN", "CTR.VN", "ELC.VN", "ICT.VN", "SAM.VN",
30
```

```

31     # Logistics, Transportation & Aviation
32     "GMD.VN", "HAH.VN", "SCS.VN", "TMS.VN",
33     "VNL.VN", "VOS.VN", "VTO.VN", "HVN.VN",
34
35     # Others
36     "IMP.VN", "OPC.VN", "PLX.VN", "VID.VN"
37 ]
38
39
40 data = yf.download(
41     tickers,
42     start="2023-01-01",
43     end="2025-01-01"
44 )["Close"]
45
46 print("Data_size:", data.shape)
47 print(data.tail())
48
49 data.to_csv("Gia_thuc_te_99_stocks.csv", index=True)

```

Kết quả đầu ra là một file CSV chứa giá đóng cửa của các loại cổ phiếu. Bảng dưới đây minh họa một phần nhỏ của dữ liệu thu thập được trong tháng 01/2023. Cần lưu ý rằng, đối với một số mã cổ phiếu, có thể tồn tại các giá trị thiếu (NaN) do nhiều nguyên nhân khác nhau như ngày nghỉ lễ, ngừng giao dịch hoặc vấn đề từ nguồn dữ liệu. Việc xử lý các giá trị thiếu này sẽ được trình bày chi tiết trong chương tiếp theo.

Bảng 3.1. Dữ liệu ACB.VN

Date	ACB.VN
1/3/2023	13612.64
1/4/2023	13582.66
1/5/2023	13702.59
1/6/2023	13882.5
1/9/2023	13852.51
1/10/2023	14062.4
1/11/2023	14512.15
1/12/2023	14692.06
1/13/2023	14482.17
1/16/2023	14991.89
1/17/2023	15171.8
1/18/2023	15291.73
1/19/2023	15591.57
1/27/2023	15801.46
1/30/2023	15231.77
1/31/2023	15621.55
2/1/2023	15111.83

Bước 2: Tính lợi nhuận theo ngày cho từng cổ phiếu

Bảng 3.2. Dữ liệu lợi nhuận cổ phiếu (1/2023)

Date	ACB.VN	ANV.VN	BID.VN
2023-01-03	-0.22027	3.734432	-0.97087
2023-01-04	0.883005	6.000003	-0.12256
2023-01-05	1.312915	4.528296	2.208592
2023-01-06	-0.21599	1.805058	-1.56063
2023-01-09	1.515153	-1.59574	0.731708
2023-01-10	3.198287	-0.90091	-0.12107
2023-01-11	1.239674	6.909099	0.484843
2023-01-12	-1.42857	-0.68028	0.844392
2023-01-13	3.51967	0.684942	6.93781
2023-01-16	1.199997	3.401359	0.559279
2023-01-17	0.790516	1.973678	0.333705
2023-01-18	1.960784	0.322585	1.884692
2023-01-19	1.346149	0	-3.04678
2023-01-27	-3.6053	-1.76849	-0.11223
2023-01-30	2.559054	4.582648	1.348317
2023-01-31	-3.26296	-6.88576	-5.09978

Bước 3: Tạo hàm mờ với lợi nhuận đã tính

Algorithm 4 Single Fuzzy Model Construction

Require: Time series X, ε, δ
Ensure: Gaussian, Triangular, Trapezoidal fuzzy sets

- 1: Remove missing values from X and sort increasingly
- 2: Initialize granules $G = \emptyset$, current granule $C = \{x_1\}$
- 3: **for** $x_i \in X, i = 2, \dots, n$ **do**
- 4: **if** $|x_i - \text{mean}(C)| \leq \varepsilon$ **then**
- 5: $C \leftarrow C \cup \{x_i\}$
- 6: **else**
- 7: $G \leftarrow G \cup \{C\}, C \leftarrow \{x_i\}$
- 8: **end if**
- 9: **end for**
- 10: $G \leftarrow G \cup \{C\}$
- 11: Compute $(c_g, \sigma_g, |g|)$ for each $g \in G$
- 12: Sort G by c_g and initialize merged set $M = \{g_1\}$
- 13: **for** $g_i \in G, i = 2, \dots, |G|$ **do**
- 14: **if** $|c_{g_i} - c_{M_{last}}| \leq \delta$ **then**
- 15: Merge g_i into M_{last} using weighted mean
- 16: **else**
- 17: $M \leftarrow M \cup \{g_i\}$
- 18: **end if**
- 19: **end for**
- 20: $N = \sum_{g \in M} |g|$
- 21: $c = \frac{1}{N} \sum_{g \in M} |g| c_g$
- 22: $\sigma = \sqrt{\frac{1}{N} \sum_{g \in M} |g| (\sigma_g^2 + c_g^2) - c^2}$
- 23: Gaussian: (c, σ)
- 24: Triangular: $(c - 2\sigma, c, c + 2\sigma)$
- 25: Trapezoidal: $(c - 3\sigma, c - 0.5\sigma, c + 0.5\sigma, c + 3\sigma)$
- 26: **return** Fuzzy membership parameters

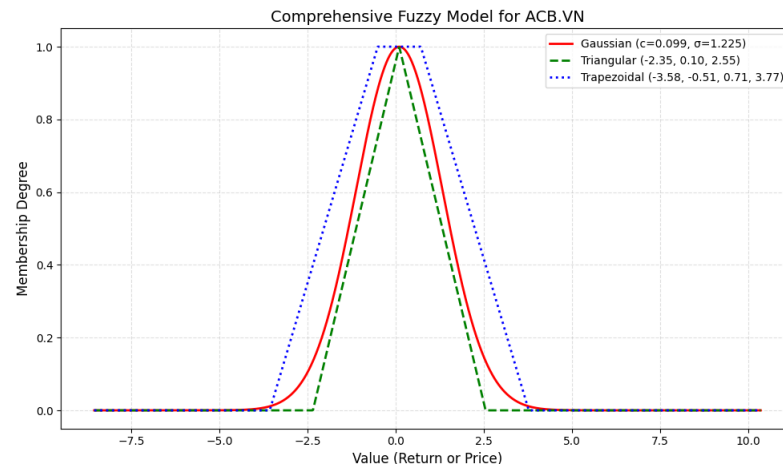
Kết quả ta sẽ có được 3 loại hàm mờ của mỗi mã cổ phiếu: Ví dụ:

```
1 {  
2   "ACB.VN": {  
3     "gaussian": {  
4       "center": 0.0986022517324121,  
5       "sigma": 1.2253996374381133  
6     },  
7     "triangular": {  
8       "a": -2.3521970231438143,  
9       "b": 0.0986022517324121,  
10      "c": 2.549401526608639  
11    },  
12    "trapezoidal": {  
13      "a": -3.5775966605819276,  
14      "b": -0.5140975669866445,  
15      "c": 0.7113020704514688,
```

```

16         "d": 3.7748011640467523
17     },
18     "size": 498
19 }
20 }

```



Hình 3.1. Biểu diễn số mờ cho mã cổ phiếu ACB.VN

3.2 So sánh hiệu quả giữa các phương pháp

Các tham số chung:

- Số cổ phiếu: 60
- Kích thước quần thể: 100
- Số thế hệ đối với các giải thuật heuristic: 500
- Seed: 42

3.2.1 Hàm tam giác

Các hàm tính toán:

```

1  # TRIANGULAR FUZZY NUMBER CALCULATION FUNCTIONS
2  def expected_tfn(fuzzy):
3      #Calculate expected value of triangular fuzzy number
4      return (fuzzy[0] + 4.0 * fuzzy[1] + fuzzy[2]) / 6.0
5
6  def variance_tfn(fuzzy):
7      #Calculate variance of triangular fuzzy number
8      return (np.power(fuzzy[1] - fuzzy[0], 2) +
9              np.power(fuzzy[2] - fuzzy[1], 2) +
10             (fuzzy[1] - fuzzy[0]) * (fuzzy[2] - fuzzy[1])) /
11             18.0

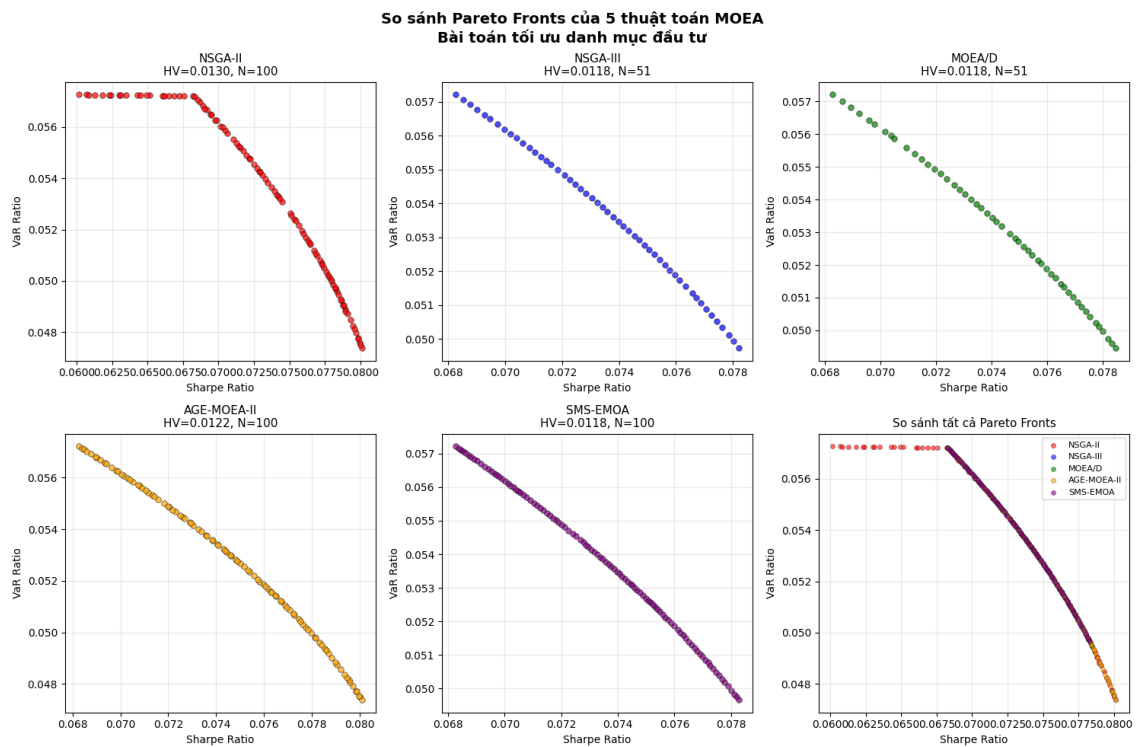
```

```

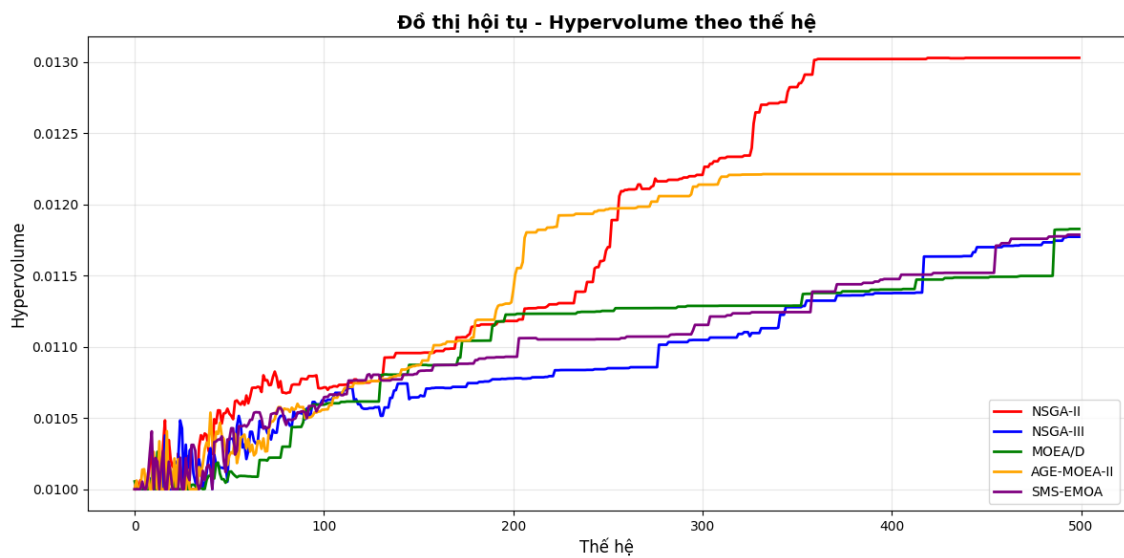
11
12 def value_at_risk_tfn(fuzzy, alpha=0.9):
13     #Calculate Value at Risk of triangular fuzzy number
14     if alpha <= 0.5:
15         return 2 * alpha * fuzzy[1] + (1 - 2 * alpha) * fuzzy[0]
16     else:
17         return (2 * alpha - 1) * fuzzy[2] + (2 - 2 * alpha) *
            fuzzy[1]
18
19 def sharpe_ratio_tfn(fuzzy, rf=RISK_FREE_RATE):
20     #Calculate Sharpe Ratio of triangular fuzzy number
21     var = variance_tfn(fuzzy)
22     if var <= 1e-10:
23         return 0.0
24     return (expected_tfn(fuzzy) - rf) / var
25
26 def var_ratio_tfn(fuzzy, rf=RISK_FREE_RATE):
27     #Calculate VaR Ratio of triangular fuzzy number
28     var = value_at_risk_tfn(fuzzy)
29     if abs(var) <= 1e-10:
30         return 0.0
31     return (expected_tfn(fuzzy) - rf) / var

```

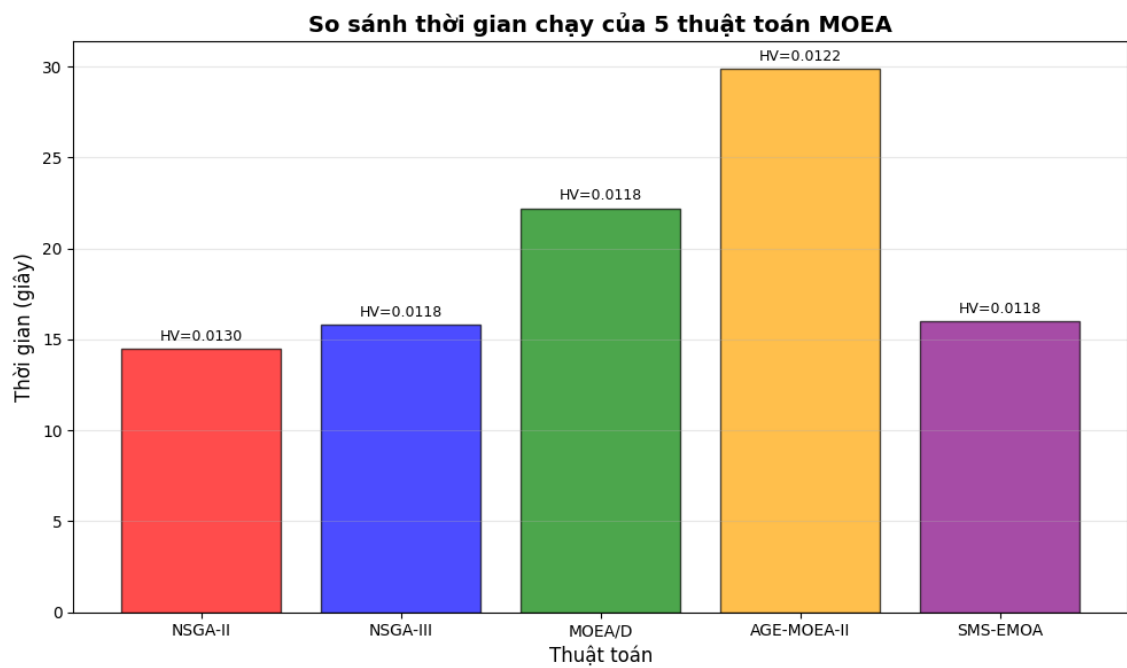
Phương pháp giải thuật heuristic



Hình 3.2. So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ tam giác

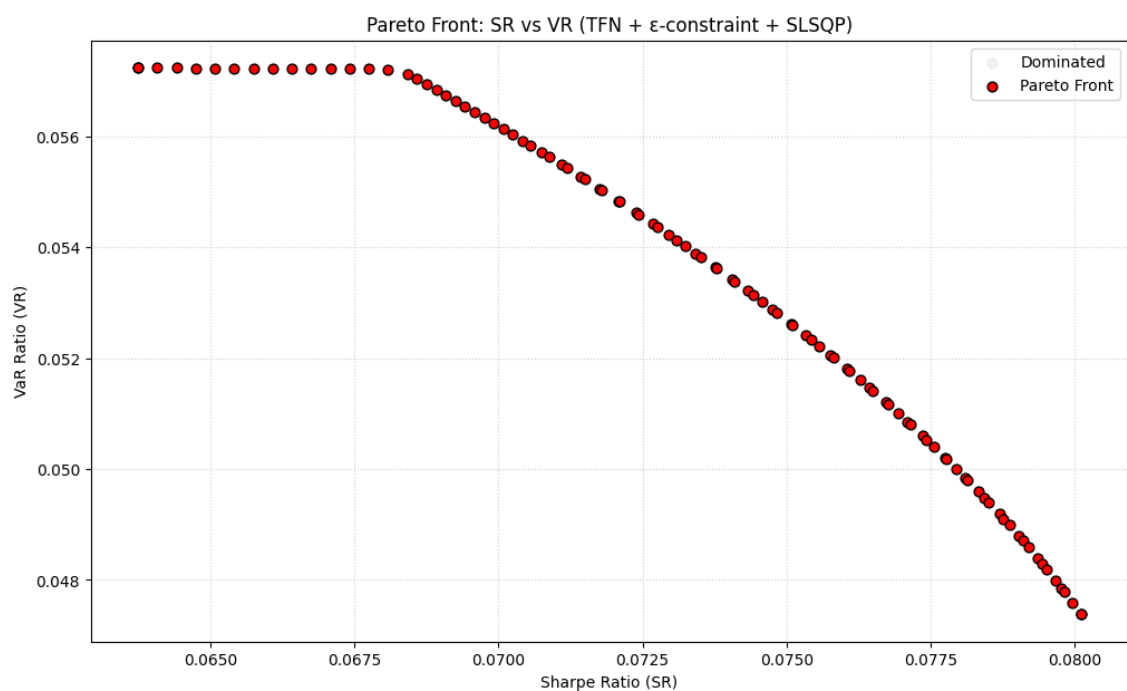


Hình 3.3. Đồ thị hội tụ Hypervolume theo thế hệ trong trường hợp số mờ tam giác



Hình 3.4. So sánh thời gian chạy của các thuật toán trong trường hợp số mờ tam giác

Phương pháp đề xuất



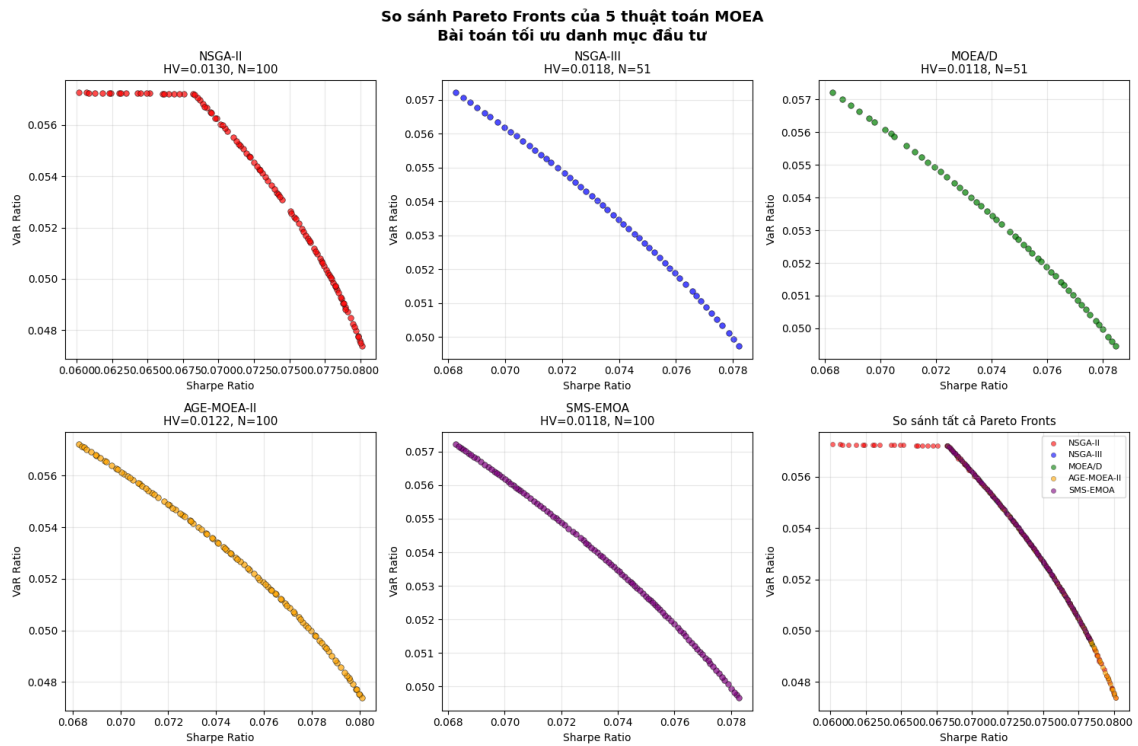
Hình 3.5. Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ tam giác

Runtime: 1.23 seconds
Pareto points: 100
Hypervolume: 0.0126725024

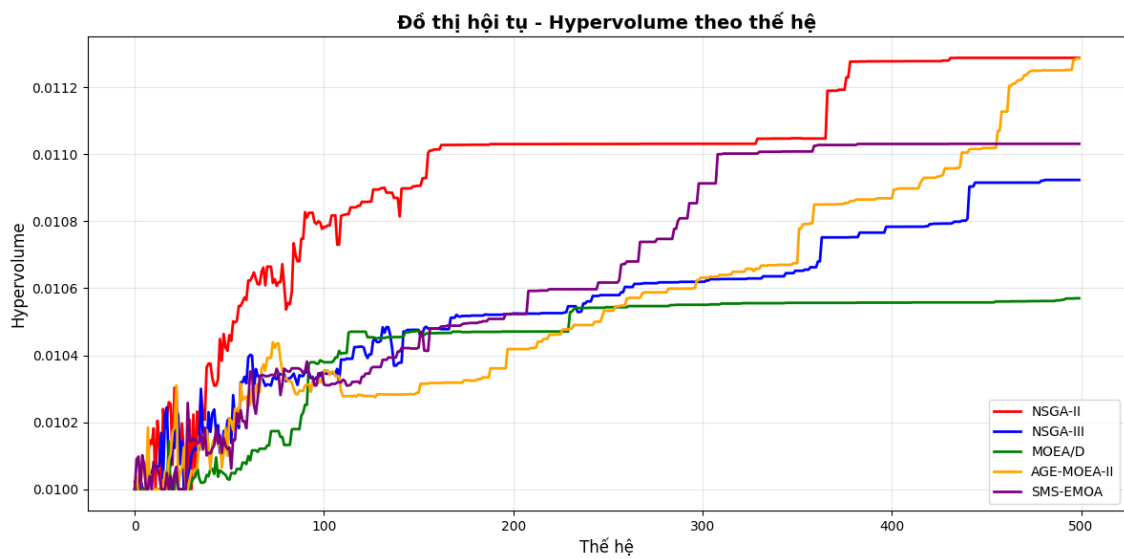
3.2.2 Số mờ hình thang

Phương pháp giải thuật heuristic

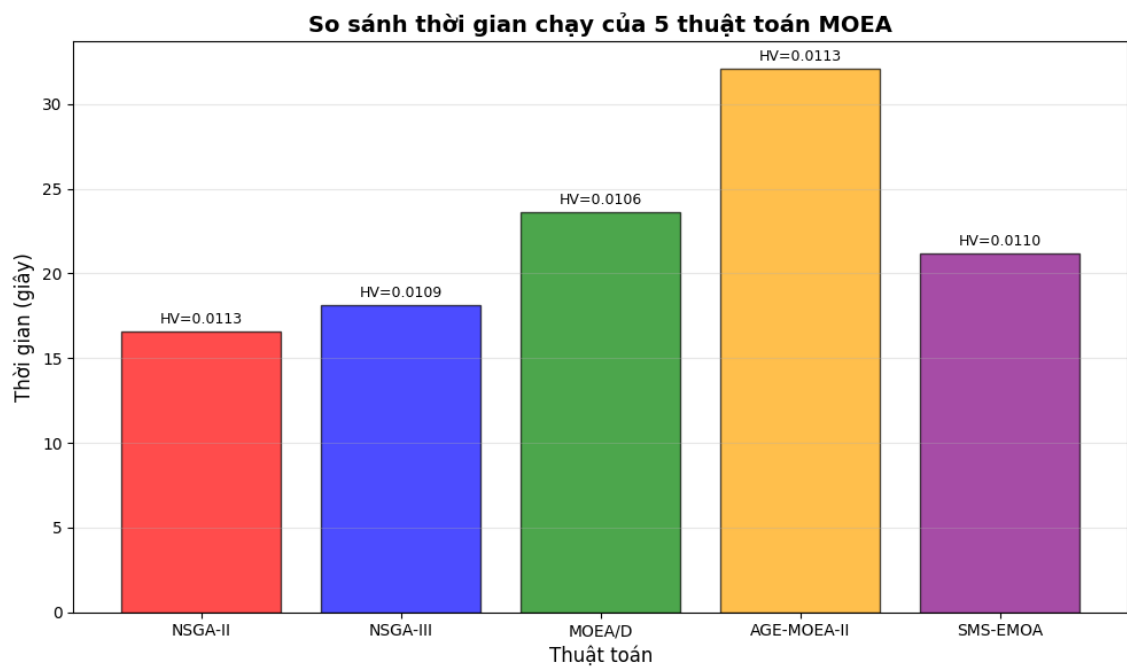
```
1 def expected_trfn(fuzzy):
2     return (fuzzy[0] + 2. * fuzzy[1] + 2. * fuzzy[2] + fuzzy[3])
3         / 6.
4
5 def variance_trfn(fuzzy):
6     return (np.power(fuzzy[1] + fuzzy[3] - fuzzy[0] - fuzzy[2],
7         2) + 6. * np.power(fuzzy[1] - fuzzy[2], 2) - 4. * (fuzzy
8         [1] - fuzzy[2]) * (fuzzy[1] + fuzzy[3] - fuzzy[0] - fuzzy
9         [2])) / 24
10
11 def value_at_risk_trfn(fuzzy):
12     alpha = 0.9
13     if alpha <= 0.5:
14         return 2 * alpha * fuzzy[1] + (1 - 2 * alpha) * fuzzy[0]
15     else:
16         return (2 * alpha - 1) * fuzzy[3] + (2 - 2 * alpha) *
17             fuzzy[2]
18
19 def VR_trfn(fuzzy):
20     return (expected_trfn(fuzzy) - 0.05) / value_at_risk_trfn(
21         fuzzy)
22
23 def SR_trfn(fuzzy):
24     return (expected_trfn(fuzzy) - 0.05) / variance_trfn(fuzzy)
25
26 def SR(X):
27     return SR_trfn((X.reshape(-1, 1).T @ TFPS).reshape(-1))
28
29 def VR(X):
30     return VR_trfn((X.reshape(-1, 1).T @ TFPS).reshape(-1))
```



Hình 3.6. So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ hình thang

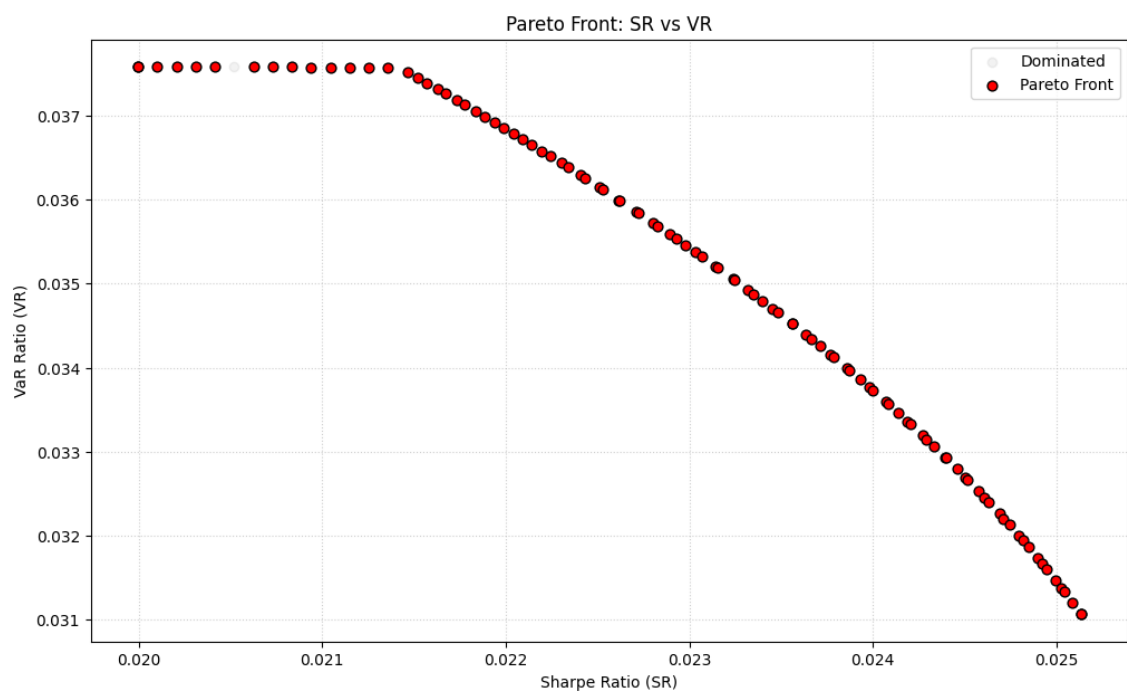


Hình 3.7. Đồ thị hội tụ Hypervolume theo thế hệ trong trường hợp số mờ hình thang



Hình 3.8. So sánh thời gian chạy của các thuật toán trong trường hợp số mờ hình thang

Phương án đề xuất



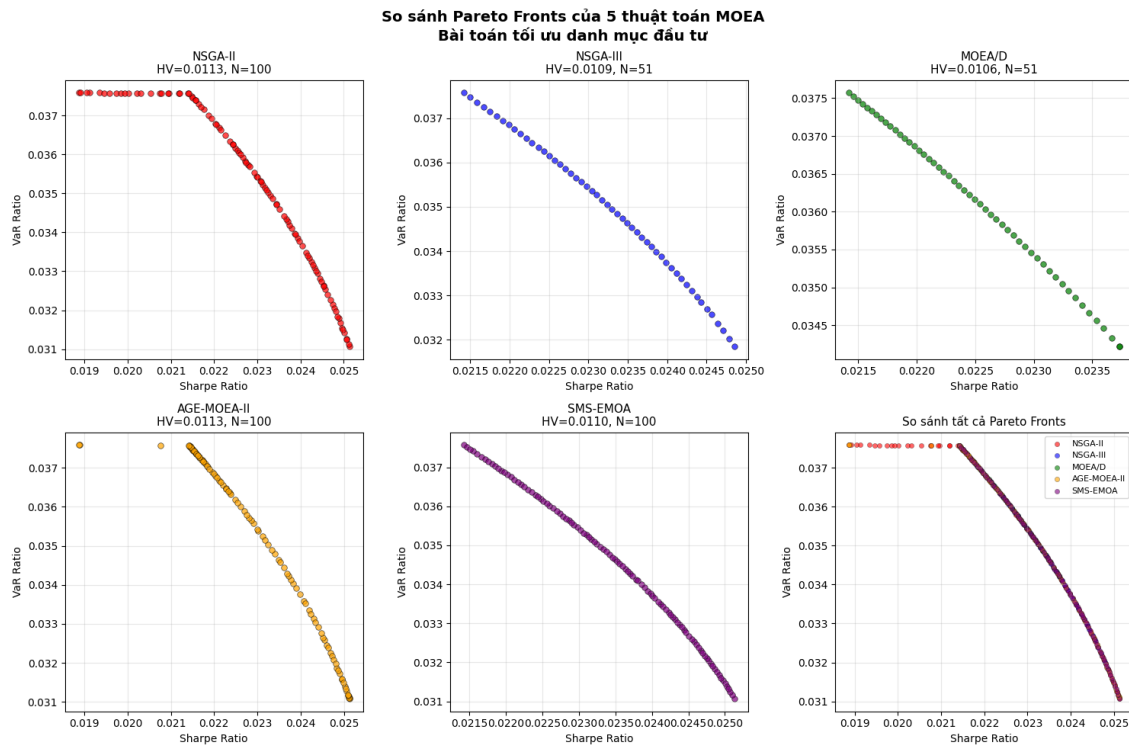
Hình 3.9. Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ hình thang

Runtime: 1.58 seconds
Pareto points: 99
Hypervolume: 0.0111747253

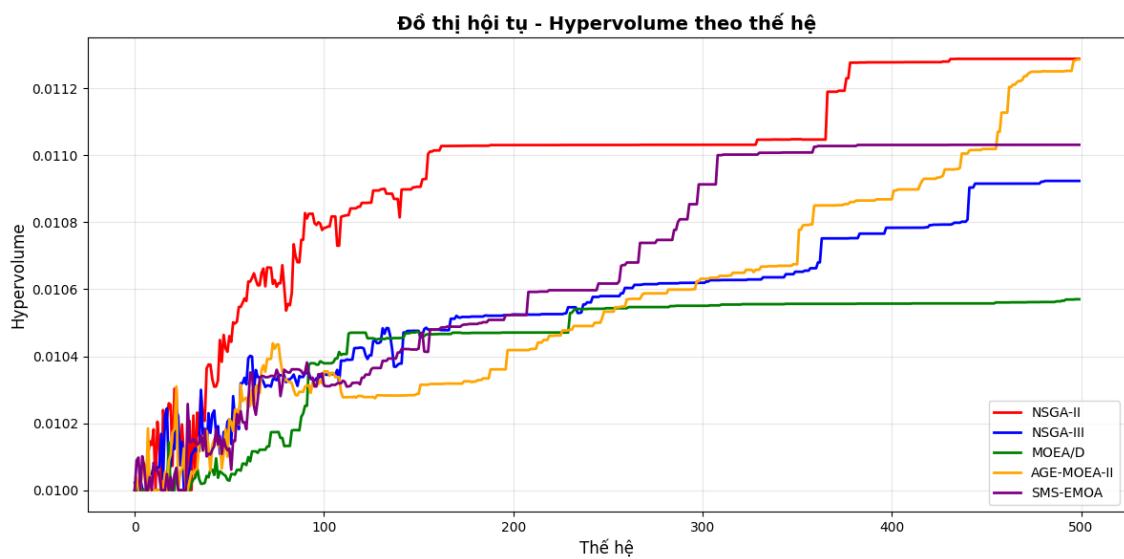
3.2.3 Số mờ Gaussian

Phương pháp giải thuật heuristic

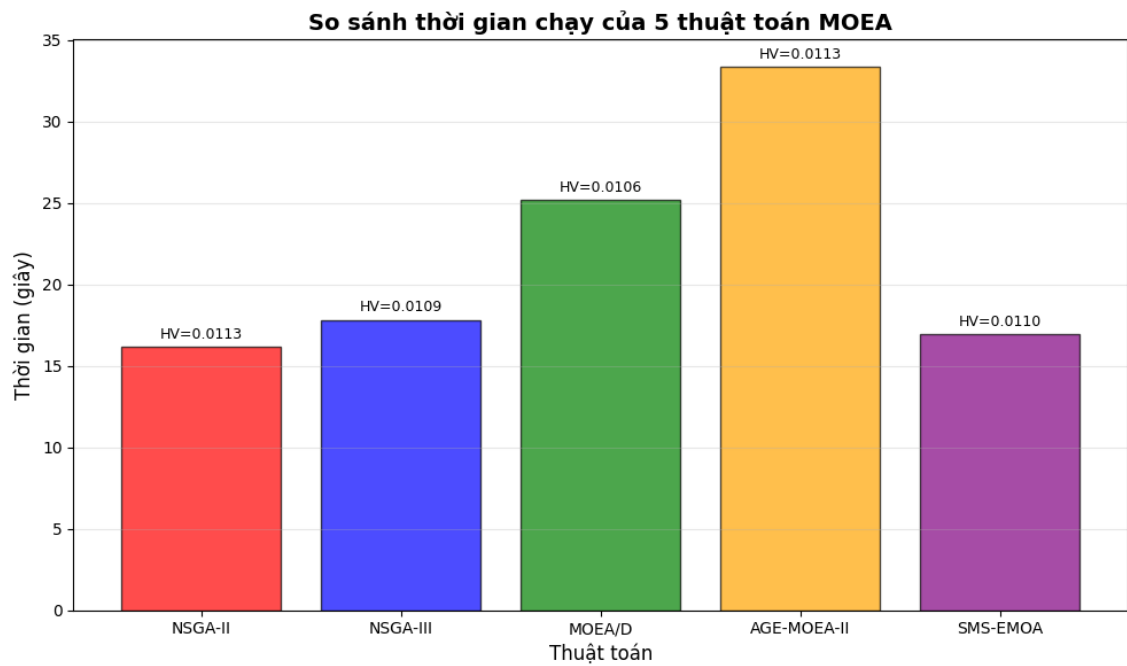
```
1 def expected_efn(fuzzy):
2     return (np.sqrt(6) * fuzzy * np.log(2)) / np.pi
3
4 def variance_efn(fuzzy):
5     return 2.0031 * np.power(fuzzy, 2)
6
7 def value_at_risk_efn(fuzzy):
8     alpha = 0.9
9     return np.max([(np.log(alpha) - np.log(1 - alpha)) * np.sqrt
10                    (6) * fuzzy / np.pi, np.zeros_like(fuzzy)], axis=0)
11
12 def VR_efn(fuzzy):
13     return (expected_efn(fuzzy) - 0.05) / value_at_risk_efn(
14         fuzzy).reshape(-1)
15
16 def SR_efn(fuzzy):
17     return (expected_efn(fuzzy) - 0.05) / variance_efn(fuzzy)
18
19 def SR(X):
20     return SR_efn(X.reshape(-1, 1).T @ EFPS).reshape(-1)
21
22 def VR(X):
23     return VR_efn(X.reshape(-1, 1).T @ EFPS).reshape(-1)
```



Hình 3.10. So sánh pareto fronts của 5 thuật toán MOEA trong trường hợp số mờ Gaussian

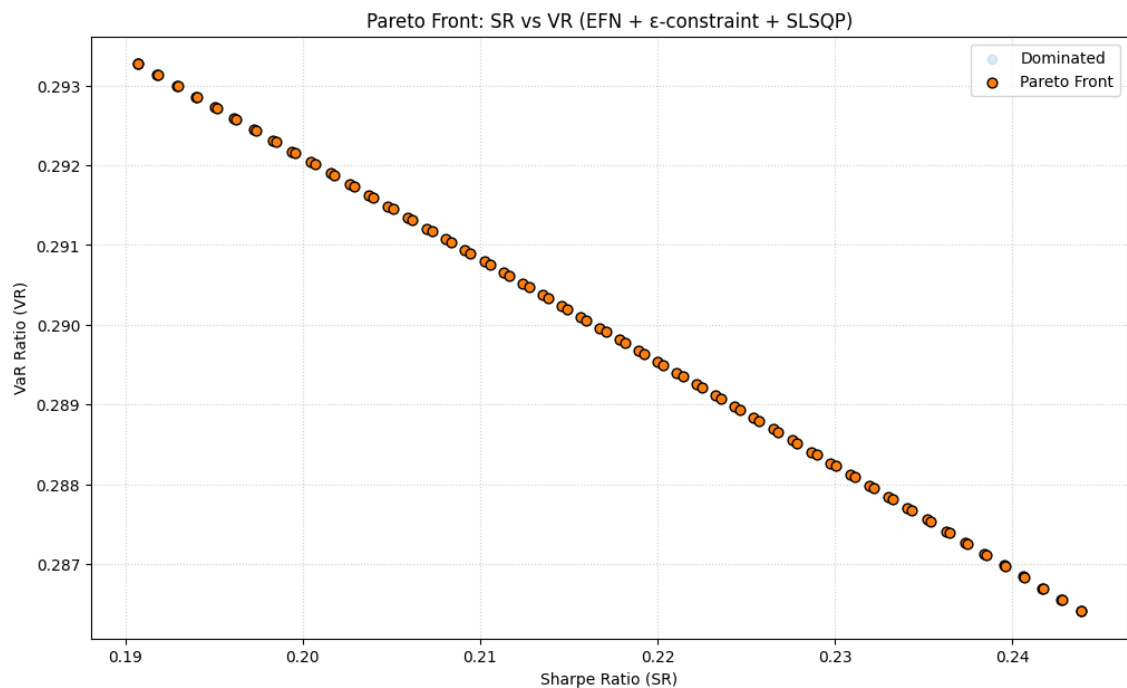


Hình 3.11. Đồ thị hội tụ Hypervolume theo thế hệ trong trường hợp số mờ Gaussian



Hình 3.12. So sánh thời gian chạy của các thuật toán trong trường hợp số mờ Gaussian

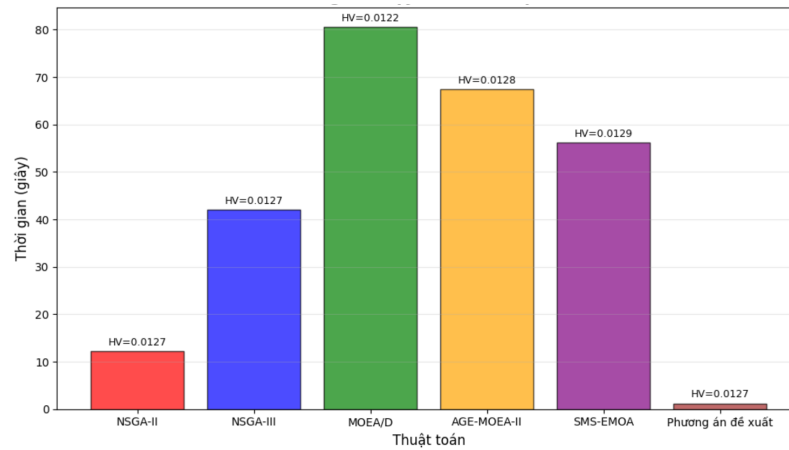
Phương án đề xuất



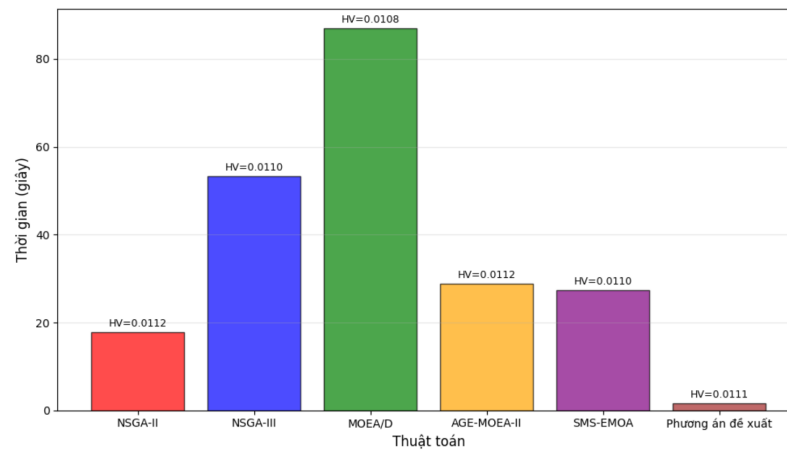
Hình 3.13. Kết quả pareto fronts phương án đề xuất trong trường hợp số mờ Gaussian

Runtime: 1.70 seconds
Pareto points: 100
Hypervolume: 0.0161806736

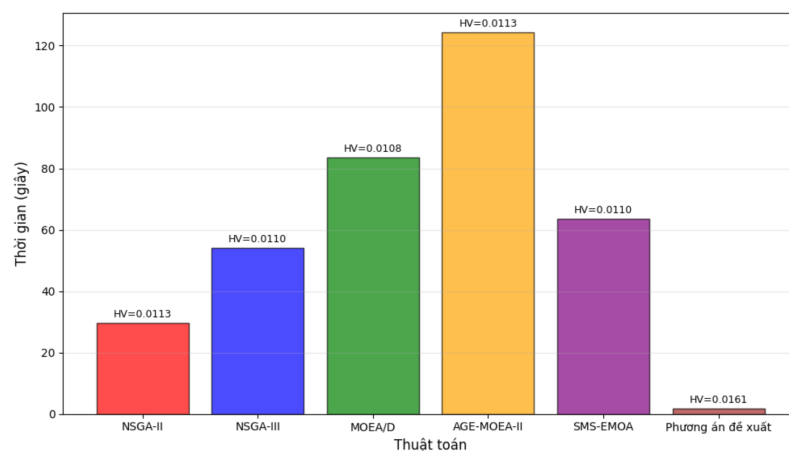
Chương 4: Tổng kết và so sánh



Hình 4.1. Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ tam giác



Hình 4.2. Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ hình thang



Hình 4.3. Biểu đồ so sánh thời gian chạy của các thuật toán với số mờ Gaussian

Bảng 4.1. So sánh thời gian thực hiện các thuật toán tối ưu đa mục tiêu khi có cùng giá trị Hypervolume

Dạng số mờ	Thuật toán	Hypervolume	Số nghiệm	Thời gian (s)	Vòng lặp
Tam giác	Phương án đề xuất	0.0127	100	1.23	50
	NSGA-II	0.0127	100	11.33	333
	NSGA-III	0.0126	51	24.91	805
	AGE-MOEA-II	0.0127	100	42.27	712
	SMS-EMOA	0.0129	100	28.12	852
	MOEA/D	0.0122	51	47.56	1001
Hình thang	Phương án đề xuất	0.0111	99	1.58	50
	NSGA-II	0.0112	100	16.59	210
	AGE-MOEA-II	0.0112	100	68.79	1000
	NSGA-III	0.0110	51	18.10	160
	SMS-EMOA	0.0110	100	43.15	1000
	MOEA/D	0.0108	51	50.14	1000
Gaussian	Phương án đề xuất	0.0161	100	1.70	50
	SMS-EMOA	0.0110	100	26.51	1000
	NSGA-II	0.0112	100	28.60	1000
	NSGA-III	0.0110	51	29.61	1000
	MOEA/D	0.0108	51	41.24	1000
	AGE-MOEA-II	0.0112	100	59.70	1000

Lưu ý: màu vàng là phương án tốt nhất, màu cam là những phương án đã chạy quá 1000 vòng lặp nhưng vẫn chưa đạt được giá trị Hypervolume mong muốn.

Trong cả ba dạng số mờ (tam giác, hình thang và Gaussian), ta có thể thấy sự vượt trội của phương án ε -constraint về thời gian và thậm chí là cả Hypervolume.

Trường hợp số mờ tam giác

- Phương pháp ε -constraint đạt Hypervolume có kết quả tương đương các thuật toán tiến hóa, nhưng thời gian chạy thấp hơn rất nhiều (1.23 s), giảm hơn 10 lần so với NSGA-II.
- MOEA/D và NSGA-III có số nghiệm Pareto ít hơn (51 nghiệm), phản ánh cách tiếp cận phân rã mục tiêu dẫn tới tập nghiệm thưa hơn, dù Hypervolume vẫn không giảm đáng kể.

Trường hợp số mờ hình thang

- Các thuật toán tiến hóa (NSGA-II, AGE-MOEA-III) cho Hypervolume gần như tương đương nhau .
- ε -constraint tiếp tục cho thời gian chạy thấp nhất (1.58 s), trong khi Hypervolume chỉ thấp hơn NSGA-II một lượng rất nhỏ.

- AGE-MOEA-II là thuật toán chậm nhất, cho thấy độ phức tạp tăng khi xử lý miền khả thi rộng hơn của số mờ hình thang.
- MOEA/D và NSGA-III có số nghiệm Pareto ít hơn (51 nghiệm), phản ánh cách tiếp cận phân rã mục tiêu dẫn tới tập nghiệm thưa hơn, dù Hypervolume vẫn không giảm đáng kể.

Trường hợp số mờ Gaussian

- ϵ -constraint đạt Hypervolume cao nhất (0.01618), vượt trội so với các thuật toán tiến hóa, đồng thời vẫn duy trì thời gian chạy thấp nhất (1.70 s).
- Các thuật toán khác dù đã chạy hơn 1000 vòng lặp vẫn chưa đạt được giá trị Hypervolume mong muốn.
- Điều này cho thấy khi hàm thuộc trơn (Gaussian), các phương pháp gradient-based tận dụng được tính trơn để hội tụ nhanh và hiệu quả hơn.

Như vậy, phương pháp được đề xuất trong đồ án là lựa chọn phù hợp cho các bài toán tối ưu đa mục tiêu với số mờ trơn, đặc biệt trong bối cảnh cần tốc độ và khả năng mở rộng.

Kết luận

Kết luận đồ án

Đồ án đã hoàn thành mục tiêu nghiên cứu, đạt được những kết quả có ý nghĩa quan trọng cả về mặt lý thuyết lẫn thực tiễn trong lĩnh vực quản lý danh mục đầu tư thông minh. Cụ thể:

- **Phát triển thành công mô hình tối ưu hóa mới dựa trên lý thuyết tập mờ:** Đồ án đã xây dựng và ứng dụng hiệu quả các số mờ (tam giác, hình thang, Gaussian) để mô hình hóa bản chất bất định và biến động của thị trường tài chính. Cách tiếp cận này cho phép biểu diễn các thông số lợi nhuận và rủi ro một cách linh hoạt và sát với thực tế hơn so với các mô hình xác suất cổ điển.
- **Đề xuất và kiểm chứng phương pháp tối ưu ưu việt:** Đồ án đã đề xuất một phương pháp lai ghép sáng tạo, kết hợp kỹ thuật ϵ -constraint dựa trên gradient với thuật toán SLSQP để giải bài toán đa mục tiêu. Kết quả thử nghiệm so sánh toàn diện với 5 thuật toán tiến hóa hàng đầu (NSGA-II, NSGA-III, MOEA/D, AGE-MOEA-II, SMS-EMOA) đã khẳng định ưu thế vượt trội của phương pháp đề xuất trên hai phương diện:
 - **Hiệu suất tính toán:** Phương pháp đề xuất đạt tốc độ xử lý nhanh hơn khoảng 10 lần trong nhiều trường hợp (ví dụ: 1.23 giây so với 14.48 giây của NSGA-II), cho thấy khả năng ứng dụng cho các bài toán quy mô lớn và yêu cầu ra quyết định nhanh.
 - **Chất lượng tập nghiệm:** Chỉ số Hypervolume – thước đo chất lượng tập Pareto – do phương pháp đề xuất đạt được là tương đương hoặc **vượt trội hơn**, đặc biệt trong trường hợp sử dụng số mờ Gaussian phức tạp, chứng tỏ khả năng tìm kiếm những giải pháp cân bằng tốt hơn giữa rủi ro và lợi nhuận.
- **Tính khả thi và giá trị ứng dụng thực tế cao:** Mô hình đã được kiểm chứng thành công trên bộ dữ liệu thực tế gồm 60 mã cổ phiếu niêm yết tại thị trường Việt Nam trong giai đoạn 2023–2025. Kết quả cho thấy mô hình không chỉ khả thi về mặt kỹ thuật mà còn là một công cụ tiềm năng, có khả năng hỗ trợ ra quyết định đầu tư tối ưu trong điều kiện thị trường đầy biến động.

Hướng phát triển đồ án trong tương lai

Để mở rộng và phát triển sâu hơn nữa những kết quả đã đạt được, đề tài này có thể được tiếp tục theo các hướng sau:

1. **Mở rộng khung đa mục tiêu:** Nghiên cứu tích hợp thêm các tiêu chí tối ưu hóa khác ngoài lợi nhuận và rủi ro, chẳng hạn như tính thanh khoản, chi phí giao dịch, hoặc các yếu tố Môi trường, Xã hội và Quản trị).
2. **Hoàn thiện quy trình xử lý thông tin bất định:** Nâng cao phương pháp “mờ hóa” dữ liệu bằng cách kết hợp với các kỹ thuật học máy và phân tích hạt thông tin (Granular Computing) tiên tiến. Mục tiêu là dự báo chính xác hơn các tham số của hàm thuộc từ dữ liệu lịch sử.
3. **Đánh giá khả năng mở rộng và tổng quát hóa:** Thử nghiệm mô hình trên quy mô dữ liệu lớn hơn hoặc với các lớp tài sản đa dạng hơn như trái phiếu, tiền điện tử, hay ngoại hối. Việc này nhằm kiểm chứng tính linh hoạt, khả năng mở rộng và hiệu quả tổng quát của thuật toán trong các bối cảnh đầu tư khác nhau.
4. **Phát triển nguyên mẫu hệ thống hỗ trợ quyết định (DSS):** Hướng tới việc tích hợp các thuật toán đã phát triển vào một nền tảng phần mềm có khả năng xử lý dữ liệu thời gian thực, tự động hóa quy trình tối ưu hóa và đề xuất tái cân bằng danh mục theo chu kỳ hoặc theo sự kiện thị trường.

Tài liệu tham khảo

- [1] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. IEEE Transactions on Evolutionary Computation, 6(2), 182–197.
- [2] Deb, K., & Jain, H. (2014). *An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints*. IEEE Transactions on Evolutionary Computation, 18(4), 577–601.
- [3] Zhang, Q., & Li, H. (2007). *MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition*. IEEE Transactions on Evolutionary Computation, 11(6), 712–731.
- [4] Panichella, A. (2018). *An Improved Pareto Front Modeling Algorithm for Large-scale Many-Objective Optimization*. IEEE Transactions on Evolutionary Computation, 22(5), 699–715.
- [5] Vuong, N. D., & Thang, T. N. (2015). *Optimizing Over Pareto Set of Semistrictly Quasiconcave Vector Maximization and Application to Stochastic Portfolio Selection*. Journal of Optimization Theory and Applications, 165(3), 1085–1108.
- [6] Haimes, Y. Y., Lasdon, L. S., & Wismer, D. A. (1971). *On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization*. IEEE Transactions on Systems, Man, and Cybernetics, 1(3), 296–297.
- [7] Nguyen, K. T., Nguyen, T. X. H., Thai, M. H., Cu, X. Q., Le, V. B., & Tran, N. T. (2023). *A Novel Pseudoconvex Programming Approach for Multi-criteria Fuzzy Portfolio Optimization with Uncertainty Ratio*. Applied Soft Computing, 146, 110682.