

# Tài liệu : Định hướng Công nghệ - Evo-worker

Dự án: ITAPIA

Module: Evo-worker

Phiên bản: 1.0 (Sơ khai)

## 1. Mục tiêu và Triết lý Thiết kế

**Evo-worker** là bộ não học hỏi và tiến hóa của ITAPIA. Nó là một service chạy nền, bất đồng bộ, có nhiệm vụ tự động khám phá và tối ưu hóa các quy tắc ra quyết định, quản lý rủi ro, và tìm kiếm cơ hội.

Triết lý thiết kế:

- **Dựa trên Thuật toán Tiến hóa (Evolutionary Algorithm-based):** Sử dụng các nguyên tắc của quá trình tiến hóa tự nhiên để tìm kiếm các giải pháp tốt trong một không gian tìm kiếm rộng lớn.
- **Linh hoạt và Đa năng (Flexible & Multi-purpose):** Nền tảng phải đủ linh hoạt để áp dụng các loại thuật toán tiến hóa khác nhau cho các bài toán khác nhau.
- **Hướng dữ liệu (Data-Driven):** Mọi quy tắc được tiến hóa đều phải được đánh giá một cách khách quan dựa trên hiệu suất trên dữ liệu lịch sử (backtesting).
- **Thích ứng (Adaptive):** Kiến trúc phải cho phép tích hợp các yếu tố cá nhân hóa (hồ sơ rủi ro người dùng) vào quá trình tối ưu hóa.

## 2. Công nghệ Cốt lõi

- **Nền tảng Thuật toán: DEAP (Distributed Evolutionary Algorithms in Python).**
  - **Lý do chọn:** DEAP là một "hộp công cụ" cực kỳ linh hoạt, không áp đặt cấu trúc. Nó cho phép tùy chỉnh sâu về biểu diễn cá thể và các toán tử di truyền, đồng thời hỗ trợ nhiều loại thuật toán tiến hóa khác nhau (GA, GP, ES, NSGA-II).
- **Nền tảng Quy tắc:** Tái sử dụng hoàn toàn **Kiến trúc Quy tắc (Rule Architecture)** từ `sharedlib`. `Evo-worker` sẽ trực tiếp đọc, thao tác, và ghi các quy tắc dưới định dạng **JSON Cây Biểu thức**.

## 3. Các "Chiến dịch Tiến hóa" (Evolution Campaigns)

`Evo-worker` sẽ không phải là một module nguyên khối, mà sẽ chạy các "chiến dịch" riêng biệt để tối ưu hóa từng module Advisor.

### a. Tối ưu hóa `Decision Maker` :

- **Bài toán:** Khám phá cấu trúc quy tắc.

- **Đối tượng Tiến hóa:** Toàn bộ cấu trúc Cây Biểu thức.
- **Thuật toán Đề xuất:** Genetic Programming (GP).
- **Mục tiêu:** Tìm ra công thức kết hợp các tín hiệu ( Technical , News , Forecasting ) để tạo ra điểm số quyết định có hiệu suất backtest cao nhất.

#### b. Tối ưu hóa Risk Management :

- **Bài toán:** Tối ưu hóa tham số.
- **Đối tượng Tiến hóa:** Các giá trị hằng số (constants) trong một cấu trúc quy tắc đã được "đóng băng" từ trước.
- **Thuật toán Đề xuất:** CMA-ES (Covariance Matrix Adaptation Evolution Strategy) hoặc Genetic Algorithm (GA).
- **Mục tiêu:** Tìm ra các ngưỡng tối ưu (ví dụ: mức PROFIT\_TRIGGER và PROFIT\_LOCK\_RATIO cho trailing stop-loss) để tối đa hóa lợi nhuận đã điều chỉnh theo rủi ro.

#### c. Tối ưu hóa Opportunity Finding :

- **Bài toán:** Khám phá quy tắc lọc.
- **Đối tượng Tiến hóa:** Toàn bộ cấu trúc Cây Biểu thức (tương tự Decision Maker ).
- **Thuật toán Đề xuất:** Genetic Programming (GP).
- **Mục tiêu:** Tìm ra các bộ lọc tạo ra một danh sách các cổ phiếu có hiệu suất tốt nhất trong giai đoạn tiếp theo.

### 4. Tích hợp Cá nhân hóa (Adaptive Fitness)

- **Tối ưu hóa Đa mục tiêu:** Evo-worker sẽ sử dụng thuật toán NSGA-II (có sẵn trong DEAP) để tối ưu hóa đồng thời nhiều mục tiêu, ví dụ: **Lợi nhuận Tối đa (Maximize Return)** và **Sụt giảm Tối thiểu (Minimize Drawdown)**.
- **Kết quả (Pareto Front):** Thay vì một quy tắc "tốt nhất", kết quả của quá trình tiến hóa sẽ là một **tập hợp các quy tắc tối ưu**, mỗi quy tắc đại diện cho một sự đánh đổi khác nhau giữa rủi ro và lợi nhuận.
- **Sử dụng:** Decision Maker sau này có thể chọn một quy tắc từ tập hợp này dựa trên hồ sơ rủi ro ( risk\_appetite ) của người dùng.