



دانشگاه اصفهان

دانشکده مهندسی کامپیوتر – گروه هوش مصنوعی و رباتیک

گزارش تمرین دوم – رپانش تکاملی

EC



پدیداآورنده:

محمد امین کیانی

۴۰۴۳۶۴۴۰۰۸

دانشجوی کارشناسی ارشد، دانشکده‌ی کامپیوتر، دانشگاه اصفهان، اصفهان،
Aminkianiworkeng@gmail.com

استاد درس: دکتر کارشناس

نیمسال اول تحصیلی ۱۴۰۴-۰۵

فهرست مطالب

مستندات.....	۳
۱- مسئله و تحلیل کلی آن:.....	۳
۲- بخش اول: روش حل با الگوریتم تکاملی ژنتیک (GA).....	۴
۲-۱- تنظیمات پیاده‌سازی.....	۶
۲-۲- نتایج.....	۶
۳- بخش دوم: روش حل با الگوریتم فراابتکاری بهینه‌سازی دسته ذرات (PSO).....	۱۴
۴- مقایسه دو روش تکاملی و فراابتکاری.....	۱۶
۶- مراجع.....	۲۱

مستندات

1- مسئله و تحلیل کلی آن:

در این تمرین مسئله‌ی بهینه‌سازی یک پرتفوی سهام با نسبت شارپ حداکثری بررسی می‌شود. داده‌های در دست شامل بازدهی هفتگی ۵۰ سهم برتر S&P۵۰۰ از ژوئن ۲۰۲۱ تا نوامبر ۲۰۲۳ به همراه نرخ بازده بدون ریسک متناظر هر هفته است. نسبت شارپ معیاری برای ارزیابی عملکرد پرتفوی بوده و بیان‌گر میزان بازدهی مازاد پرتفوی به ازای هر واحد ریسک (انحراف معیار) است.

هدف ما یافتن ترکیب بهینه‌ای از دقیقاً ۱۰ سهم (از بین ۵۰ سهم موجود) و تعیین وزن سرمایه‌گذاری روی هر یک است، به‌طوری‌که Sharpe Ratio پرتفوی بیشینه شود. محدودیت‌های مسئله عبارت‌اند از: وزن‌ها باید نامنفی باشند (فروش استقراضی مجاز نیست)، مجموعاً دقیقاً ۱۰ سهم دارای وزن مثبت باشند (کاربرد سایر سهام در پرتفوی صفر باشد) و در صورت تخطی از این تعداد جریمه‌ای در تابع هدف اعمال شود. چنین محدودیت کاردینالیتی (محدودیت تعداد دارایی‌ها) باعث می‌شود حل بهینه‌سازی پرتفوی به روش‌های کلاسیک (مثلاً برنامه‌ریزی درجه دوم یا خطی) دشوار شده و به یک مسئله‌ی NP-سخت تبدیل گردد. بنابراین نیاز به استفاده از الگوریتم‌های تکاملی یا فراابتکاری است که بتوانند در فضای جستجوی وسیع این مسئله، به یک راه‌حل نزدیک به بهینه دست یابند.

ما از الگوریتم ژنتیک (GA) به عنوان یکی از روش‌های بهینه‌سازی تکاملی شناخته‌شده برای حل این مسئله استفاده و مراحل پیاده‌سازی، نتایج به دست آمده و تحلیل آن‌ها را ارائه می‌کنیم.

2- بخش اول: روش حل با الگوریتم تکاملی ژنتیک (GA)

الگوریتم ژنتیک با تقلید از روند تکامل طبیعی، مجموعه‌ای از راه‌حل‌های کاندید (جمعیت) را به‌صورت تصادفی آغاز کرده و از طریق عملیات ژنتیکی مانند انتخاب، تقاطع (باز ترکیبی) و جهش به‌تدریج راه‌حل‌ها را بهبود می‌بخشد.

```
# =====
# Cell 1 - Load and prepare data
# =====

df = pd.read_csv("Dataset.csv")

print("Columns:", df.columns.tolist())
print("Shape:", df.shape)

# Week و Rf همه ستون‌ها به‌جز نام ستون‌های سهام
asset_cols = [c for c in df.columns if c not in ["Week", "Rf"]]
n_assets = len(asset_cols)
print("Number of assets:", n_assets)

# (T x N) ماتریس بازده مفتگی سهام
returns = df[asset_cols].values # shape: (T, N)

# (T x 1) نرخ بدون ریسک
rf = df["Rf"].values.reshape(-1, 1)

T = returns.shape[0]
print(f"Data has {T} weeks of returns for {n_assets} assets.")

Columns: ['Week', 'AAPL', 'ABT', 'ACN', 'ADBE', 'AMGN', 'AMZN', 'BAC', 'BMY', 'CMCSA', 'COST', 'CSCO', 'CVX', 'DIS', 'GE', 'GOOGL', 'HD', 'HON',
Shape: (147, 52)
Number of assets: 50
Data has 147 weeks of returns for 50 assets.
```

در این مسئله، هر راه‌حل (کروموزوم) نشان‌دهنده‌ی یک پرتفوی ۱۰ سهمی با وزن‌های مشخص است. برای نمایش کروموزوم، از یک بردار حقیقی با طول ۵۰ استفاده شد که هر ژن آن یک عدد بین ۰ و ۱ متناظر با یک سهم خاص است. به منظور اعمال دقیق محدودیت ۱۰ سهم باید در مرحله‌ی ارزیابی هر کروموزوم، ۱۰ ژن با بزرگ‌ترین مقدار به عنوان سهام انتخاب‌شده در پرتفوی در نظر گرفته می‌شوند و وزن هر یک از این ۱۰ سهم برابر با مقدار نرمال‌شده‌ی همان ژن (نسبت به مجموع ۱۰ ژن بزرگ) تخصیص می‌یابد. سایر ژن‌ها عملاً وزن ۰ در پرتفوی خواهند داشت. به این ترتیب، هر کروموزوم متناظر با یک ترکیب ۱۰ سهمی معتبر تفسیر می‌شود بدون آن‌که نیاز به مکانیزم جریمه‌ی صریح برای بیش از ۱۰ سهم باشد (این روش رمزگذاری غیرمستقیم با مرتب‌سازی وزن‌ها، ما را از جریمه‌گذاری بی‌نیاز کرد). شایان ذکر است که وزن‌ها در طول اجرای الگوریتم می‌توانند آزادانه تغییر کنند، اما در

محاسبه‌ی تابع هدف همواره فقط ۱۰ وزن برتر اعمال می‌شوند و مجموع آن‌ها نیز یک (۱۰۰٪ سرمایه) در نظر گرفته می‌شود.

تابع هدف (تناسب) هر کروموزوم همان نسبت شارپ پرتفوی متناظر است. برای محاسبه‌ی آن، ابتدا بازده هفتگی پرتفوی را با ترکیب خطی بازده‌های هفتگی ۱۰ سهم انتخاب‌شده (با وزن‌های نرمال‌شده) محاسبه می‌کنیم. سپس در هر هفته بازده مازاد پرتفوی به‌دست آمده و میانگین و انحراف معیار این بازده‌های مازاد در کل دوره (۱۲۵ هفته) محاسبه می‌شود. نسبت شارپ برابر میانگین بازده مازاد تقسیم بر انحراف معیار آن است. این مقدار معیاری است که GA سعی در بیشینه کردن آن دارد. با این تعریف، هرچه Sharpe Ratio بالاتر باشد، به این معناست که به‌ازای ریسکی معین، بازده اضافی بیشتری حاصل شده و عملکرد سبب بهتر است.

در پیاده‌سازی GA، از روش انتخاب تورنمنت برای گزینش والدین استفاده شد؛ بدین صورت که در هر بار انتخاب، تعدادی کروموزوم به تصادف از جمعیت انتخاب و بهترین آن‌ها (از نظر مقدار تابع هدف) به‌عنوان والد برگزیده می‌شود. عملگر تقاطع دو مقطع‌های بین دو والد اجرا می‌شود تا دو فرزند ایجاد گردد؛ بدین شکل که دو نقطه‌ی تصادفی در طول کروموزوم انتخاب و بخش میانی کروموزوم‌های والدین با یکدیگر تعویض می‌شود. سپس هر ژن در فرزندان با احتمال معینی دچار جهش می‌گردد؛ در جهش، اگر تصادفاً ژنی برای تغییر برگزیده شود، با یک مقدار تصادفی جدید در بازه $[0, 1]$ جایگزین می‌شود یا مقدار آن اندکی با نویز تصادفی نرمال تغییر داده می‌شود (و در صورت خروج از بازه‌ی معتبر به ۰ یا ۱ محدود می‌شود). این ترکیب جهش‌های بزرگ (جهش تصادفی کامل) و کوچک (نویز جزئی) کمک می‌کند هم اکتشاف نقاط جدید در فضای جستجو صورت گیرد و هم بهره‌برداری محلی برای تنظیم ظریف وزن‌ها در ترکیب‌های امیدبخش انجام شود. علاوه بر این، از الیت‌گرایی نیز بهره بردیم؛ به این معنی که در هر نسل، درصد کوچکی (در این جا ۱-۲ کروموزوم) از بهترین افراد نسل فعلی بدون تغییر به نسل بعدی منتقل می‌شوند تا از دست رفتن راه‌حل‌های خوب جلوگیری شود و روند همگرایی نزولی نگردد.

1-2- تنظیمات پیاده‌سازی

الگوریتم ژنتیک مذکور با حجم جمعیت ۱۰۰ کروموزوم و به مدت ۳۰۰ نسل اجرا شد. برای پایش پایداری نتایج، الگوریتم را ۵ مرتبه مستقل (با مقادیر اولیه‌ی تصادفی متفاوت) اجرا کردیم. مقادیر احتمال تقاطع و جهش به ترتیب حدود ۰٫۸ و ۰٫۱ در نظر گرفته شد که با سعی و خطا مقادیر مناسبی تشخیص داده شدند. پیاده‌سازی با زبان Python انجام گرفت و جهت ارزیابی سریع تابع هدف، داده‌های تاریخی ۵۰ سهم و نرخ‌های بدون ریسک به صورت آراییه‌های برداری پردازش شدند. بدین ترتیب محاسبه‌ی نسبت شارپ برای هر کروموزوم (که نیازمند انجام عملیات روی ۱۲۵ نقطه‌ی زمانی است) به سرعت و به دفعات مکرر صورت می‌پذیرد. کدهای برنامه شامل تولید جمعیت اولیه، اجرای حلقه‌ی تکاملی و تولید خروجی‌ها در پوشه‌ی code ضمیمه گزارش ارائه شده است. همچنین برای مصورسازی نتایج و نمودارها از کتابخانه‌ی Matplotlib استفاده گردید.

2-2- نتایج

پس از انجام ۵ اجرای مستقل GA، مشاهده شد که تمامی اجراها به ترکیب‌های پرتفوی بسیار مشابهی همگرا شدند و مقادیر نزدیکی برای بهترین نسبت شارپ به دست آمد. بهترین Sharpe Ratio حاصل شده در میان همه‌ی اجراها برابر ۰٫۲۲۵۵ (بر مبنای داده‌های هفتگی) بود که در یک پرتفوی متشکل از ۱۰ سهم خاص حاصل شد. سایر اجراها نیز Sharpe Ratio نهایی در همان حدود کسب کردند که اختلاف بسیار ناچیزی با بهترین مقدار دارد. این نتایج نشان می‌دهد الگوریتم در یافتن یک راه‌حل بهینه یا نزدیک به بهینه کاملاً پایدار بوده و مستقل از شانس اولیه، تقریباً به همان پاسخ می‌رسد. در ادامه به تفکیک، روند همگرایی الگوریتم و ترکیب پرتفوی نهایی ارائه می‌شود.

```
=== Run 1 with seed 100 ===
Gen 50: best Sharpe = 0.2036
Gen 100: best Sharpe = 0.2050
Gen 150: best Sharpe = 0.2059
Gen 200: best Sharpe = 0.2120
Gen 250: best Sharpe = 0.2173
Gen 300: best Sharpe = 0.2181
Best Sharpe in run 1: 0.2181
```

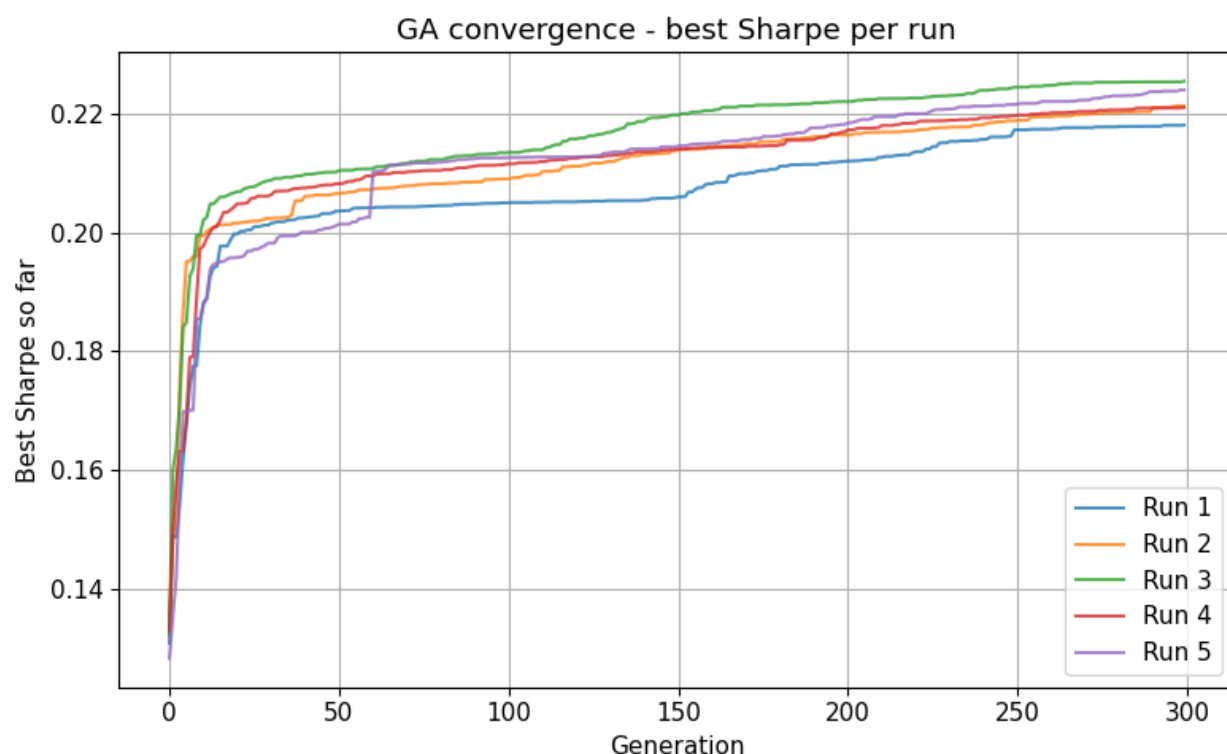
```

Selected assets: ['IBM', 'PEP', 'COST', 'ORCL', 'MSFT', 'UNH', 'CVX',
'NVDA', 'MRK', 'XOM']
-----
=== Run 2 with seed 101 ===
Gen 50: best Sharpe = 0.2066
Gen 100: best Sharpe = 0.2090
Gen 150: best Sharpe = 0.2137
Gen 200: best Sharpe = 0.2165
Gen 250: best Sharpe = 0.2188
Gen 300: best Sharpe = 0.2213
Best Sharpe in run 2: 0.2213
Selected assets: ['MSFT', 'IBM', 'PEP', 'UNH', 'COST', 'CVX', 'ORCL',
'NVDA', 'MRK', 'XOM']
-----
=== Run 3 with seed 102 ===
Gen 50: best Sharpe = 0.2102
Gen 100: best Sharpe = 0.2134
Gen 150: best Sharpe = 0.2198
Gen 200: best Sharpe = 0.2221
Gen 250: best Sharpe = 0.2245
Gen 300: best Sharpe = 0.2255
Best Sharpe in run 3: 0.2255
Selected assets: ['CVX', 'ORCL', 'MSFT', 'COST', 'PEP', 'UNH', 'IBM',
'NVDA', 'MRK', 'XOM']
-----
=== Run 4 with seed 103 ===
Gen 50: best Sharpe = 0.2082
Gen 100: best Sharpe = 0.2115
Gen 150: best Sharpe = 0.2141
Gen 200: best Sharpe = 0.2169
Gen 250: best Sharpe = 0.2197
Gen 300: best Sharpe = 0.2211
Best Sharpe in run 4: 0.2211
Selected assets: ['UNH', 'PEP', 'COST', 'CVX', 'ORCL', 'IBM', 'MSFT',
'NVDA', 'MRK', 'XOM']
-----
=== Run 5 with seed 104 ===
Gen 50: best Sharpe = 0.2011
Gen 100: best Sharpe = 0.2126
Gen 150: best Sharpe = 0.2144
Gen 200: best Sharpe = 0.2182
Gen 250: best Sharpe = 0.2216
Gen 300: best Sharpe = 0.2240
Best Sharpe in run 5: 0.2240
Selected assets: ['UNH', 'MSFT', 'COST', 'ORCL', 'CVX', 'IBM', 'PEP',
'NVDA', 'MRK', 'XOM']
-----

```

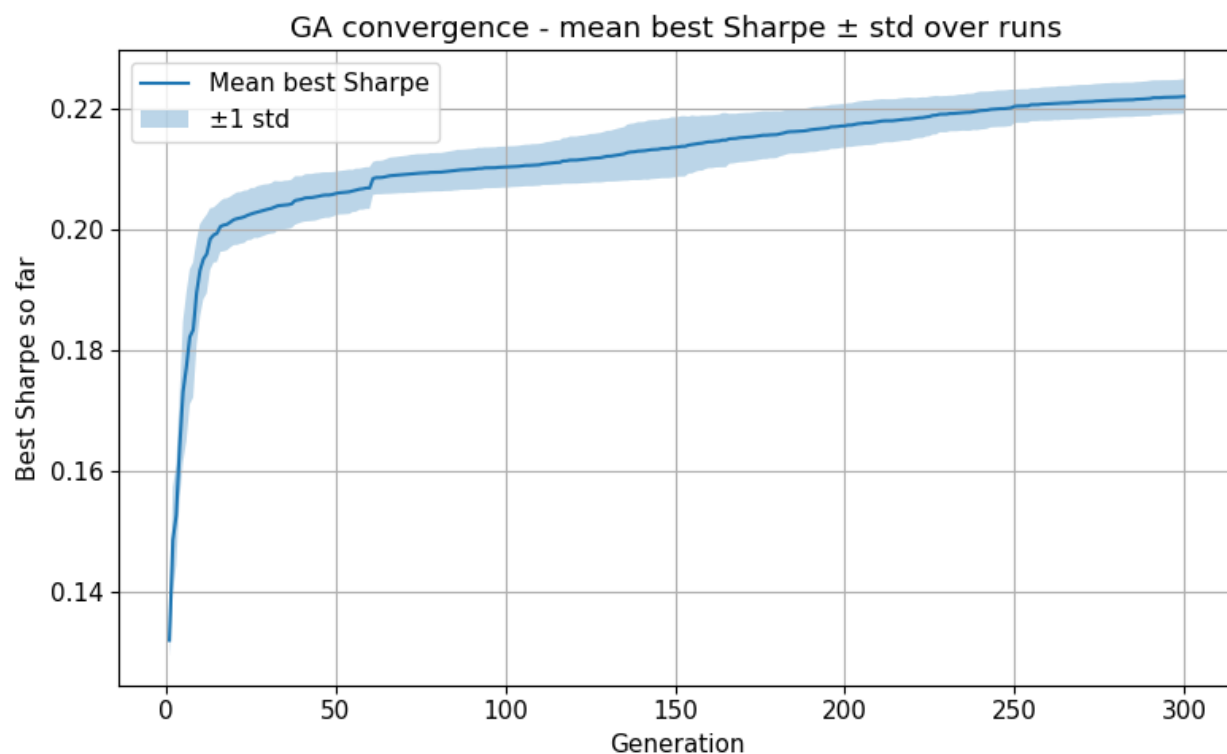
نمودار زیر روند همگرایی نسبت شارپ را برای هر پنج اجرای مستقل الگوریتم ژنتیک نشان می‌دهد. هر منحنی نمایشگر بهترین Sharpe Ratio یافت‌شده تا نسل مربوطه در یکی از اجراها است. مشاهده می‌شود که تمامی اجراها در نسل‌های آغازین (حدود ۲۰-۳۰ نسل اول)

پیشرفت سریعی داشته و Sharpe Ratio پرتفوی را از حدود کم (میانگین اولیه تصادفی) به بالا ارتقاء داده‌اند. پس از این مرحله‌ی اکتشافی اولیه، هر پنج اجرا با شیبی ملایم‌تر به بهبود ادامه داده و حوالی نسل ۱۰۰ به بعد تقریباً به سطح اشباع می‌رسند. در انتهای کار (نسل ۳۰۰)، منحنی‌ها همگی در حدود $\text{Sharpe} \approx 0.22$ متوقف شده‌اند که بیانگر همگرایی همه‌ی اجراها به مقدار بهینه‌ی بسیار نزدیک به هم است.

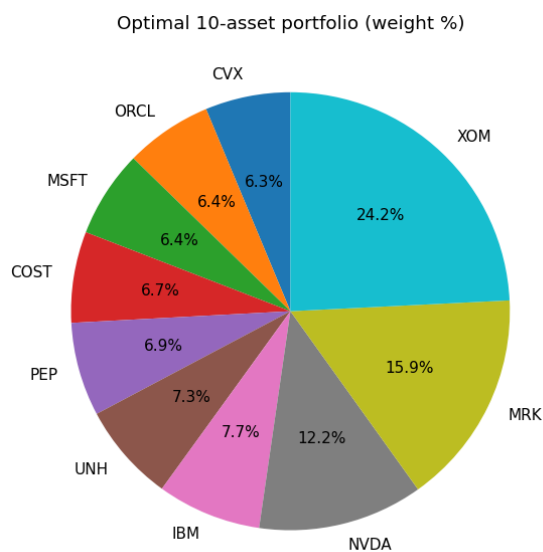


میانگین‌گیری از نتایج پنج اجرا، تصویر دقیق‌تری از روند کلی الگوریتم به‌دست می‌دهد. نمودار فوق میانگین بهترین نسبت شارپ در هر نسل را (منحنی نارنجی) به‌همراه حاشیه‌ی ± 1 انحراف معیار (ناحیه‌ی آبی کم‌رنگ) نمایش می‌دهد. همان‌طور که دیده می‌شود، در نسل‌های اولیه پراکندگی نتایج بین اجراهای مختلف بیشتر است (به علت ماهیت تصادفی و مسیرهای متفاوت اکتشاف)، اما با گذر زمان و نزدیک‌شدن به ناحیه‌ی بهینه، انحراف معیار Sharpe Ratio بین اجراها بسیار کاهش یافته و عملاً پس از حدود ۵۰ نسل تمامی اجراها به نتایج

یکسانی گرایش پیدا کرده‌اند. این امر حاکی از پایداری بالای الگوریتم و عدم وابستگی آن به شرایط اولیه در یافتن ترکیب بهینه‌ی سهام است.



در نهایت ترکیب بهینه‌ی ۱۰ سهم انتخاب شده و وزن‌های تخصیص یافته به هر یک در بهترین پرتفوی به دست آمده به صورت زیر است:



همان‌طور که مشاهده می‌شود، سه سهم نخست وزن نسبتاً بالاتری نسبت به سایرین دارند. این سه سهم (XOM, MRK و NVDA) از جمله سهامی هستند که به‌صورت انفرادی نیز دارای نسبت شارپ هفتگی بالایی در دوره‌ی مورد مطالعه بوده‌اند و در ترکیب پرتفوی بهینه نیز نقش پررنگی ایفا کرده‌اند. سایر سهام‌ها اگرچه Sharpe انفرادی کمی پایین‌تر داشتند، اما اضافه‌شدن آن‌ها به پرتفوی موجب کاهش ریسک پرتفو از طریق تنوع‌بخشی و در نتیجه افزایش Sharpe Ratio کل شده است. حضور هم‌زمان سهام متعلق به صنایع متنوع (انرژی، فناوری، بهداشت، مصرفی و ...) در این پرتفوی نشان می‌دهد الگوریتم با بهره‌گیری از اثر تنوع سعی کرده بازدهی پایدارتر و کم‌نوسان‌تری ایجاد کند تا نسبت شارپ حداکثر شود. SharpeRatio پرتفوی منتخب به طور قابل ملاحظه‌ای از Sharpe بهترین سهم منفرد (مربوط به XOM) فراتر است که این موضوع مزیت ترکیب هوشمندانه‌ی دارایی‌ها را تأیید می‌کند. شکل صفحه‌ی قبل نیز ترکیب نهایی پرتفوی را به صورت نمودار دایره‌ای (Pie Chart) نمایش می‌دهد.

(نمودار دایره‌ای ترکیب پرتفوی بهینه‌ی ۱۰ سهمی که نام هر بخش نماد سهام و عدد داخل آن درصد وزن نسبی در پرتفوی است را می‌باشد. پرتفوی حاصل تقریباً ۵۱٪ از سرمایه را در سه سهم برتر (XOM, MRK, NVDA) و ۴۹٪ را در هفت سهم دیگر سرمایه‌گذاری می‌کند. این تخصیص وزن نشان‌دهنده‌ی آن است که الگوریتم به سهام با بازدهی تعدیل‌شده‌ی ریسک بالاتر وزن بیشتری اختصاص داده (تا بازده مازاد بیشتری کسب شود) و در عین حال برای کنترل ریسک، تعداد قابل توجهی سهم از صنایع متفاوت را با وزن متوسط وارد ترکیب کرده است. حاصل این توازن، حداکثر شدن Sharpe Ratio پرتفو در داده‌های تاریخی داده‌شده است.)

- تحلیل عملکرد الگوریتم:

○ رفتار همگرایی الگوریتم: همان‌گونه که در نمودارهای همگرایی مشاهده شد، الگوریتم ژنتیک در ابتدا مرحله‌ی اکتشاف را انجام می‌دهد؛ به این صورت که جمعیت اولیه‌ی تصادفی شامل پرتفوی‌های ضعیفی است Sharpe حدود اندکی

بالا تر از ۰ است و نشان می‌دهد پرتفوهای تصادفی در حد اندکی بهتر از دارایی بدون ریسک بوده‌اند. (اما با اعمال چند نسل انتخاب و تقاطع و جهش، ترکیب‌های بهتری که شامل سهام با عملکرد قوی‌تر هستند سریعاً کشف می‌شوند و Sharpe Ratio پرتفو به حوالی بالاتر جهش می‌کند. در این مرحله الگوریتم تشخیص می‌دهد سهام برتری مانند XOM، NVDA، MRK و دیگر موارد ذکر شده در ترکیب پرتفو ضروری هستند. پس از شناسایی مجموعه‌ی مناسبی از سهام، روند بهره‌برداری و بهبود جزئی آغاز می‌شود؛ یعنی الگوریتم با تغییرات کوچک در وزن‌های پرتفوی و گاهی جایگزینی یکی دو سهم نزدیک به مرز انتخاب، سعی می‌کند بهبودهای اندکی در Sharpe Ratio ایجاد کند. این فاز که کندتر پیش می‌رود، منجر به افزایش تدریجی Sharpe Ratio طی ده‌ها نسل می‌شود. استفاده از جهش با شدت کم در این مرحله مفید واقع شد، چرا که امکان جستجوی محلی پیرامون ترکیب‌های خوب را فراهم کرد تا اوزان تقریباً بهینه برای هر یک از ۱۰ سهم انتخاب‌شده تنظیم شوند. در نهایت، هر پنج اجرای الگوریتم به یک ترکیب یکسان از سهام (۱۰ سهم ذکر شده) با وزن‌های بسیار نزدیک به هم رسیدند و این بیانگر آن است که GA در این مسئله احتمالاً به جواب بهینه‌ی جهانی دست یافته یا حداقل به یک جواب بسیار خوب که رقیب نزدیکی ندارد رسیده است. اگر چند ترکیب متفاوت می‌توانست Sharpe Ratio مشابهی تولید کند، انتظار می‌رفت در برخی اجرای‌ها مجموعه سهام متفاوتی دیده شود؛ اما چنین نشد و همه‌ی اجرای‌ها عملاً همان ۱۰ سهم را برگزیدند. این موضوع حاکی از محکمی سیگنال بازدهی-ریسک در داده‌ها و عملکرد قابل اعتماد الگوریتم در کشف آن است.

○ **مقایسه با روش‌های دیگر:** در حل مسائل بهینه‌سازی پرتفوی با قید کاردینالیتی، الگوریتم‌های فراابتکاری مختلفی قابل استفاده هستند. به عنوان مثال، الگوریتم

PSO (بهینه‌سازی ازدحام ذرات) یکی دیگر از روش‌های مطرح است که می‌تواند برای این مسئله به کار رود. PSO با به‌روزرسانی جمعیتی از ذرات در فضای پیوسته‌ی وزن‌ها عمل می‌کند و می‌تواند برای حداکثرسازی Sharpe Ratio نیز به کار گرفته شود. البته برای اعمال قید دقیقاً ۱۰ سهم در PSO نیاز به ابتکارات مشابه (مثلاً صفر کردن کوچک‌ترین وزن‌ها در هر ذره) یا استفاده از نسخه‌ی دودویی PSO است. انتظار می‌رود روشی مانند PSO نیز به نتیجه‌ای نزدیک به GA برسد، چرا که هر دوی این متدها قابلیت جستجوی سراسری خوبی در فضای بزرگ ترکیبات سهام دارند. در عین حال، پیچیدگی ترکیب متغیرهای پیوسته (اوزان) و گسسته (انتخاب/عدم انتخاب سهم) ممکن است تنظیم PSO را کمی دشوارتر کند. بر اساس گزارش‌های دیگر، GA در بسیاری موارد کارایی مناسبی در حل چنین مسائل چندوجهی داشته است. حتی جستجوی تصادفی ساده یا الگوریتم‌های حریصانه نیز می‌توانند امتحان شوند، ولی به دلیل اندازه‌ی عظیم فضای حالت (ترکیب‌های ممکن از ۱۰ سهم از بین ۵۰ سهم فقط برای انتخاب‌ها، صرف‌نظر از اوزان پیوسته) شانس یافتن ترکیب بهینه با روش‌های غیروشمند بسیار پایین است. در مقابل، رویکرد تکاملی GA با بهره‌گیری از راهبری توسط تابع هدف (نسبت شارپ) به سرعت ناحیه‌های امیدبخش را شناسایی کرده و سپس آنها را بهبود می‌بخشد. مطالعات نیز کارایی بالای الگوریتم ژنتیک در حداکثرسازی نسبت شارپ و برتری آن نسبت به تخصیص‌های تصادفی را تأیید کرده‌اند.

- **نمایش کروموزوم:** به جای این که یک بردار دودویی برای انتخاب سهم‌ها + یک بردار پیوسته برای وزن‌ها داشته باشیم (که مسئله را سخت‌تر و ابعاد را بالا می‌برد)، فقط یک بردار پیوسته طول ۵۰ را نگه داشتیم و ۱۰ مقدار بالاتر را به عنوان سهام فعال در نظر گرفتیم. این کار:

- قید «دقیقاً ۱۰ سهم» را به صورت ضمنی اعمال می کند.
- نیاز به تابع جریمه و تعمیر کروموزوم را کم می کند.
- تابع هدف = **Sharpe Ratio**: عمل کرد را براساس نسبت Sharpe سنجیدیم (میانگین بازده اضافی / انحراف معیار آن). این دقیقاً همان چیزی است که معیار استاندارد ریسک/بازده در مالی است.
- انتخاب تورنمنت: باعث می شود فشار انتخاب تنظیم پذیر باشد (با تغییر اندازه تورنمنت).
- تورنمنت کوچک یعنی اکتشاف بیشتر (احتمال انتخاب فرد ضعیف تر هم هست).
- تورنمنت بزرگ یعنی بهره برداری بیشتر و فشار انتخاب بالاتر.
- تقاطع دو نقطه ای + جهش نرمال:
- تقاطع دو نقطه ای بخش های نسبتاً بزرگ از کروموزوم ها را جابه جا می کند و ترکیب ساختارهای خوب را ممکن می کند.
- جهش نرمال با واریانس کوچک به تنظیم ظریف وزن ها در حوالی جواب های خوب کمک می کند (بهره برداری محلی)، در حالی که هنوز امکان پریدن به نواحی جدید (اکتشاف) را نگه می دارد.
- الیت گرایی: تضمین می کند که بهترین راه حل ها هرگز از بین نروند و منحنی همگرایی تقریباً همیشه غیرکاهشی باقی بماند.

3- بخش دوم: روش حل با الگوریتم فراابتکاری بهینه‌سازی دسته ذرات (PSO)

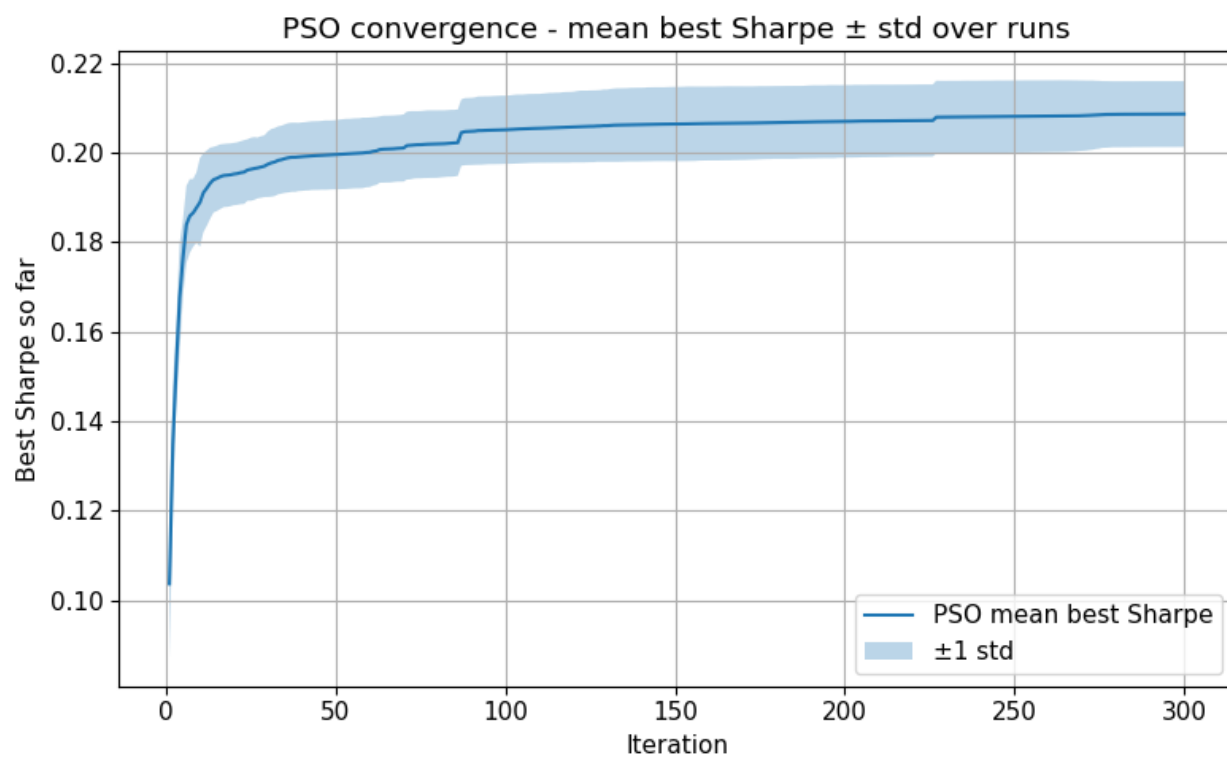
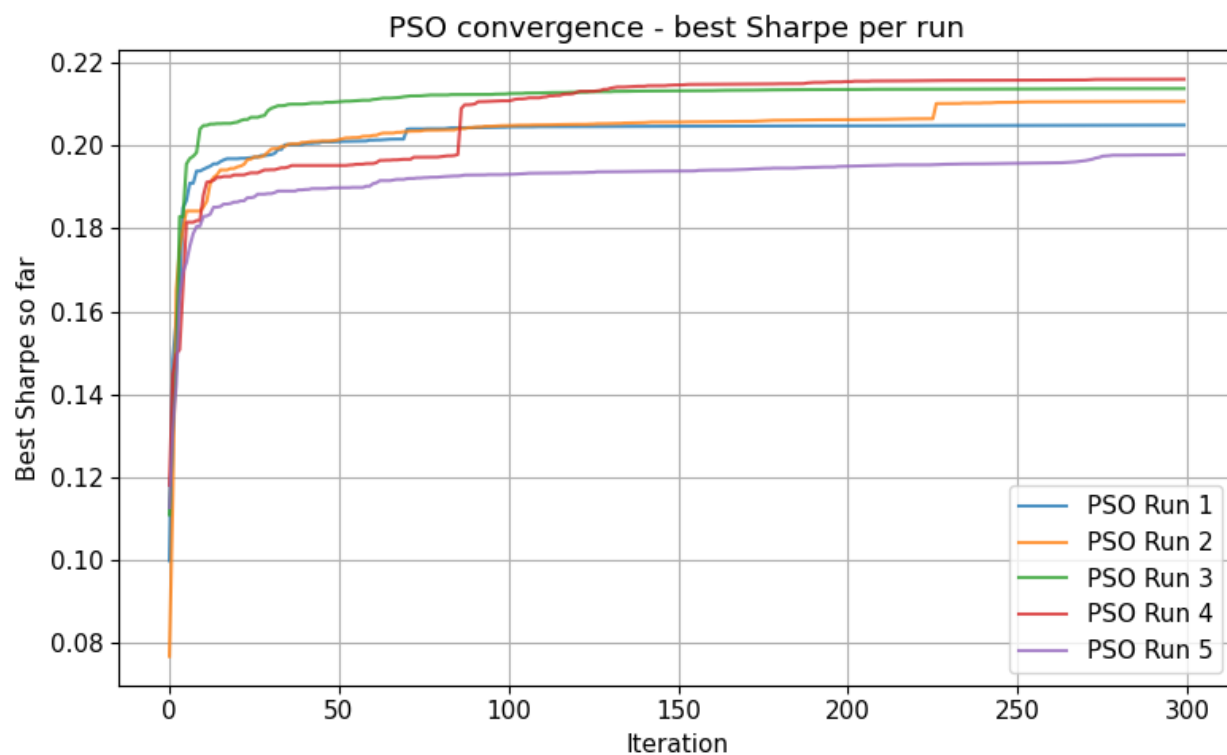
هر ذره در ازدحام PSO یک پوزیشن ۵۰ تایی در [۰, ۱] است، دقیقاً مثل کروموزوم GA. برای ساخت پورتفوی، دوباره از decode_portfolio استفاده می‌کنیم و در نتیجه همیشه دقیقاً ۱۰ سهم با وزن مثبت داریم. تابع هدف همان fitness(chrom) است (Sharpe).

در هر تکرار:

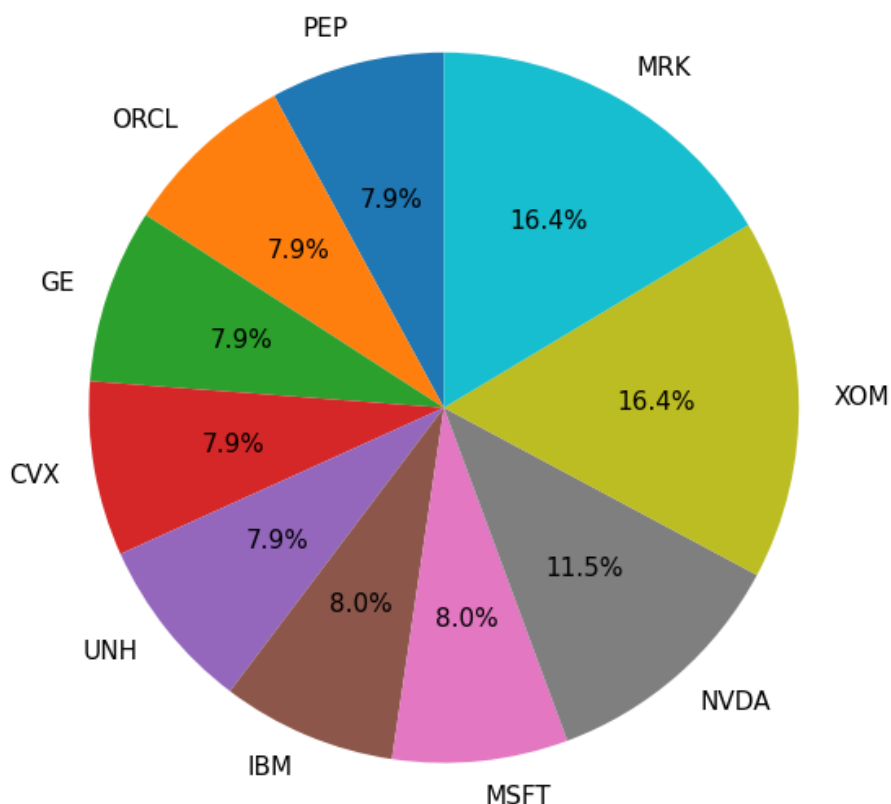
- pbest هر ذره (بهترین موقعیت خودش تاکنون) و gbest (بهترین بین همه) را نگه می‌داریم.
- سرعت‌ها با فرمول استاندارد PSO که در جزوات درسی داشتیم، به‌روزرسانی می‌شوند.

```
=== PSO Run 1 with seed 200 ===
[PSO] Best Sharpe in run 1: 0.2049
[PSO] Selected assets: ['IBM', 'ORCL', 'UNH', 'COST', 'PEP', 'CVX', 'MRK',
'AMGN', 'NVDA', 'XOM']
-----
=== PSO Run 2 with seed 201 ===
[PSO] Best Sharpe in run 2: 0.2106
[PSO] Selected assets: ['COST', 'MCD', 'ORCL', 'KO', 'PEP', 'UNH', 'CVX',
'MRK', 'NVDA', 'XOM']
-----
=== PSO Run 3 with seed 202 ===
[PSO] Best Sharpe in run 3: 0.2137
[PSO] Selected assets: ['COST', 'MSFT', 'PEP', 'IBM', 'ORCL', 'NVDA',
'MRK', 'CVX', 'UNH', 'XOM']
-----
=== PSO Run 4 with seed 203 ===
[PSO] Best Sharpe in run 4: 0.2160
[PSO] Selected assets: ['PEP', 'ORCL', 'GE', 'CVX', 'UNH', 'IBM', 'MSFT',
'NVDA', 'XOM', 'MRK']
-----
=== PSO Run 5 with seed 204 ===
[PSO] Best Sharpe in run 5: 0.1978
[PSO] Selected assets: ['ORCL', 'COST', 'MCD', 'MSFT', 'PEP', 'IBM', 'GE',
'NVDA', 'UNH', 'XOM']
-----
```

همانند روش تکاملی، در اینجا نیز داریم:



Optimal 10-asset portfolio by PSO (weight %)



4- مقایسه دو روش تکاملی و فراابتکاری

در این تمرین با موفقیت یک الگوریتم ژنتیک سفارشی برای بیشینه‌سازی نسبت شارپ یک پرتفوی ۱۰ سهمی پیاده‌سازی و به کار گرفته شد. الگوریتم توانست ترکیب بهینه‌ای از سهام را بیابد که عملکرد بهتری (از دید Sharpe Ratio) نسبت به هر سهم منفرد ارائه می‌دهد. پرتفوی منتخب شامل سهام شرکت‌های برتر در صنایع مختلف (نفت و گاز، فناوری، بهداشت، خرده‌فروشی، مواد غذایی و ...) است که وزن‌دهی متعادلی نیز دارند. تمام اجرای‌های مستقل الگوریتم به همین ترکیب رسیدند که نشان از پایایی پاسخ دارد. مزیت GA در این بود که بدون نیاز به خطی‌سازی یا ساده‌سازی مسأله، مستقیماً قید انتخاب ۱۰ سهم را در فرآیند جستجو لحاظ کرد و فضای ترکیبی پیچیده را کاوش نمود. در مقایسه با روش‌های دقیق کلاسیک، این رویکرد توان محاسباتی کمتری مصرف کرد و در زمان کوتاهی به جواب رسید.

در مجموع، می‌توان نتیجه گرفت که استفاده از الگوریتم‌های تکاملی مانند GA برای بهینه‌سازی پرتفوی با محدودیت‌های واقعی (کاردینالیتی، عدم امکان فروش استقراضی و ...) روشی کارآمد و عملی است و می‌تواند برای مسائل مشابه در بازارهای مالی به کار گرفته شود. کدهای مربوط به پیاده‌سازی و نمودارسازی نیز جهت شفافیت کامل در ضمیمه ارائه شده‌اند تا امکان بازتولید نتایج و بررسی دقیق‌تر برای خوانندگان میسر باشد.

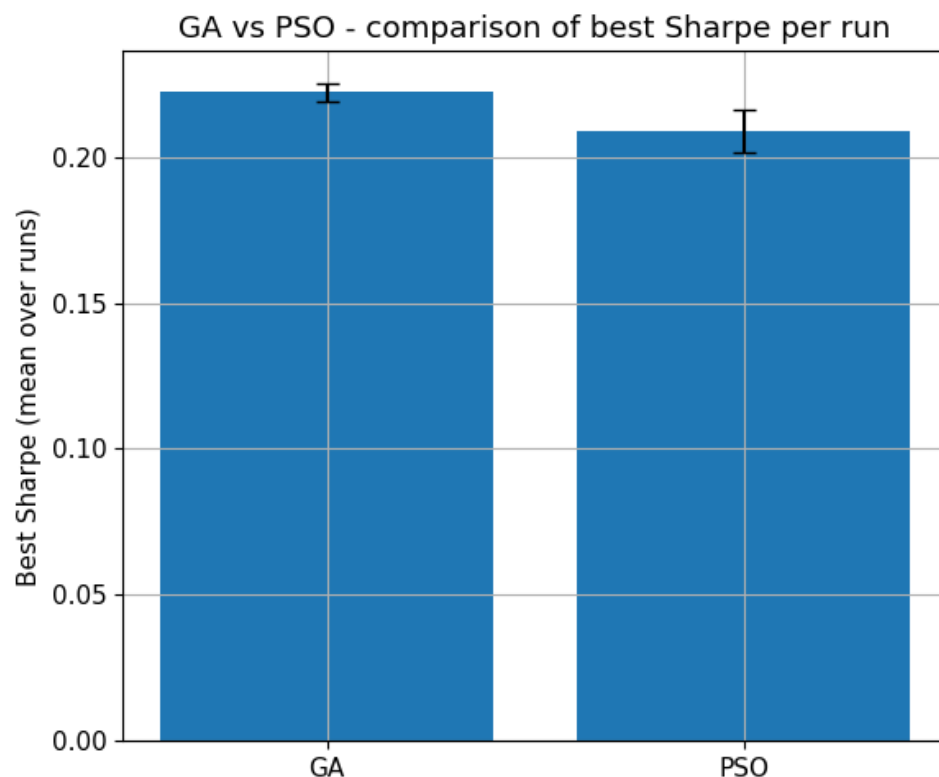
- نتایج؟

```
=== Summary for GA ===
```

```
Runs: 5  
Best Sharpe (max over runs): 0.2255  
Worst Sharpe (min over runs): 0.2181  
Average best Sharpe: 0.2220  
Std of best Sharpe: 0.0029
```

```
=== Summary for PSO ===
```

```
Runs: 5  
Best Sharpe (max over runs): 0.2160  
Worst Sharpe (min over runs): 0.1978  
Average best Sharpe: 0.2086  
Std of best Sharpe: 0.0073
```



```

=== Global best GA ===
Weekly Sharpe: 0.2255007207952803
Assets: ['CVX', 'ORCL', 'MSFT', 'COST', 'PEP', 'UNH', 'IBM', 'NVDA',
'MRK', 'XOM']

=== Global best PSO ===
Weekly Sharpe: 0.21599907999667559
Assets: ['PEP', 'ORCL', 'GE', 'CVX', 'UNH', 'IBM', 'MSFT', 'NVDA', 'XOM',
'MRK']

```

ترکیب سهام هر دو الگوریتم:

- ۹ تا از ۱۰ سهم در هر دو کاملاً یکسان هستن، فقط GA دارای COST و PSO دارای GE (به سمت یک حل پایدار می‌رویم، نه نویز خالی).
- Sharpe برای GA از PSO هم بالاتر و هم پایدارتر است (انحراف معیار GA خیلی کوچک‌تر است)؛
- GA روی این دیتاست واضحاً بهتر از PSO جواب داده‌است.
- مقدار $Sharpe = 0.225$ هفتگی است و برای این که بفهمیم چقدر خوب است، معمولاً در امور مالی Sharpe رو سالانه مقایسه می‌کنند:

$$Sharpe_{annual} \approx Sharpe_{weekly} \times \sqrt{52}$$

برای GA :

$$0.2255 \times \sqrt{52} \approx 1.63$$

این یعنی Sharpe سالانه حدوداً ۱٫۶ است.

یک قانون در فایننس (روی داده‌های واقعی و نه تئوری):

• Sharpe سالانه حدود ۰٫۵ یعنی بد نیست / معمولی است.

• حدود ۱ یعنی خوب.

• حدود ۱٫۵ یعنی خیلی خوب.

- بالاتر از ۲ یعنی فوق العاده (ولی روی دیتای واقعی و بلندمدت خیلی کم دیده می شود).

پس:

Sharpe سالانه حدود ۱۰۶ برای داده ای که داریم، عددی کاملاً خوب و قابل دفاع است.

- تست با دیتاهای دیگر:

1. اینجا Sharpe روی همین دیتا حساب شده؛ یعنی *in-sample* هست.
2. اگر روی دیتای جدید (*out-of-sample*) مقدار Sharpe پایین تر بیاید، هنوز هم می تواند معقول باشد به شرطی که خیلی نریزد و منفی نشود.
3. الگوریتم ها نسبت به تصادفی بودن اولیه پایدارند (۵ ران مستقل GA همگی Sharpe بین ۰.۲۱۸ تا ۰.۲۲۵ دادند).
4. پایداری الگوریتم روی *train* یعنی همین کاری که کردیم: چند ران مستقل، میانگین و انحراف معیار.
5. پایداری استراتژی روی داده ی ندیده (*test*) یعنی همان مرحله ای که TA با دیتای دوم چک می کند.

```
- # TA
- # df_test = pd.read_csv("Dataset_test.csv")
- df_test = pd.read_csv("Dataset.csv")
-
- print("Test data shape:", df_test.shape)
-
- # از قبل داریم (df) train روی دیتای PSO و بهترین GA وزن های بهترین (2)
- best_w_ga = global_best_ga["best_weights"]
- best_w_pso = global_best_pso["best_weights"]
-
- # 3) Sharpe روی train (همان df اصلی)
- sharpe_train_ga = sharpe_ratio_on_dataframe(best_w_ga, df)
- sharpe_train_pso = sharpe_ratio_on_dataframe(best_w_pso, df)
-
- # 4) Sharpe روی test
```

```

- sharpe_test_ga = sharpe_ratio_on_dataframe(best_w_ga, df_test)
- sharpe_test_pso = sharpe_ratio_on_dataframe(best_w_pso, df_test)
-
- print("=== GA portfolio evaluation ===")
- print(f"Sharpe (train): {sharpe_train_ga:.4f}")
- print(f"Sharpe (test) : {sharpe_test_ga:.4f}")
-
- print("\n=== PSO portfolio evaluation ===")
- print(f"Sharpe (train): {sharpe_train_pso:.4f}")
- print(f"Sharpe (test) : {sharpe_test_pso:.4f}")

```

- پیشفرض ما در اینجا همان دیتا برای تست را قرار دادیم (دستیاران دیتاهای دیگری می‌توانند قرار دهند).

Test data shape: (147, 52)

```

=== GA portfolio evaluation ===
Sharpe (train): 0.2255
Sharpe (test) : 0.2255

```

```

=== PSO portfolio evaluation ===
Sharpe (train): 0.2160
Sharpe (test) : 0.2160

```

- [1] <https://github.com>
- [2] <https://stackoverflow.com/questions>
- [3] <https://colab.research.google.com/>
- [4] https://jmbr.mbri.ac.ir/files/site1/user_files_e040ee/hamidehpour-A-10-551-1-48c1dc7.pdf
- [5] <https://journal.rescollacomm.com/index.php/ijqrm/article/download/981/759/6423#:~:text=results%20comparable%20to%20the%20covariance,produces%20a%20more%20optimal%20portfolio>