

Portfolio Optimization

آراد وزیرپناه - سید علی حسینی

۲۵ تیر ۱۴۰۳

خلاصه:

در این گزارش قصد داریم مسئله Portfolio Optimization یا همان مسئله پرتفوی را با الگوریتم‌های مختلف مانند الگوریتم ژنتیک، تبرید شبیه‌سازی شده (Simulated annealing)، بهینه سازی ازدحام ذرات (PSO) و همچنین الگوریتم تبرید شبیه‌سازی شده تعمیم یافته و ترکیبی از این الگوریتم‌ها را با الگوریتم ژنتیک، بررسی کنیم. و سپس نتایج آن را با دنیای واقعی مقایسه کنیم تا ببینیم که آیا عملکرد خوبی داشته یا خیر.

مقدمه:

مسئله پرتفوی یک مسئله بهینه‌سازی در زمینه سرمایه‌گذاری است که هدف آن تعیین ترکیب بهینه دارایی‌ها در یک پرتفو (مجموعه از سرمایه‌گذاری‌ها) به منظور دستیابی به حداکثر بازده یا کمترین ریسک است. در این مسئله، سرمایه‌گذار باید تصمیم بگیرد که سرمایه خود را بین دارایی‌های مختلف تقسیم کند تا نسبت به حداکثر کردن بازده مورد انتظار یا کاهش دادن ریسک کلی سرمایه‌گذاری خود عمل کند.

در این مسئله، عواملی مانند بازده مورد انتظار، واریانس، همبستگی بین دارایی‌ها و محدودیت‌هایی مانند سقف سرمایه، محدودیت‌های مالی و سایر محدودیت‌ها باید در نظر گرفته شوند. با استفاده از تکنیک‌های بهینه‌سازی، مانند برنامه‌ریزی خطی یا برنامه‌ریزی غیرخطی، ترکیب بهینه دارایی‌ها به منظور دستیابی به هدف مورد نظر محاسبه می‌شود.

مسئله پرتفوی در زمینه مدیریت سرمایه و سرمایه‌گذاری بسیار مهم است و به کمک روش‌های بهینه‌سازی و تحلیل مالی، سرمایه‌گذاران می‌توانند ترکیبی از دارایی‌های مختلف را انتخاب کنند که هم تحت محدودیت‌های مختلف قرار داشته باشند و هم بهینه‌سازی شده باشند.

همچنین در این مسئله ما یک دو داده ورودی داریم و انتظار داریم با استفاده از این دو داده، سرمایه‌گذاری را به گونه‌ای بهینه کنیم که بیشترین سود و کمترین ضرر را داشته باشیم. حال داده‌ها به این صورت هستند:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
-0.004	0.009	0.019	0.009	0.003	-0.004	0.008	-0.003	0.011	0.015

جدول ۱: سود به ازای یک واحد خرید از هر ارز

همچنین ماتریس کوواریانس که بیان می‌کند وابستگی بین هر ارز با دیگری، چه مقدار است.

	BTC	ETHER	BNB	XRP	cardano	Dogecoin	Solana	Litecoin	TRON	polygon
BTC	0.006767	0.006637	0.004294	0.002996	0.006355	0.006678	0.011359	0.005337	0.003300	0.006084
ETHER	0.006637	0.008962	0.005332	0.003020	0.007932	0.009988	0.012955	0.006030	0.003335	0.008208
BNB	0.004294	0.005332	0.006423	0.002526	0.005652	0.007042	0.010523	0.006043	0.002859	0.006623
XRP	0.002996	0.003020	0.002526	0.006748	0.004102	0.004248	0.008098	0.003302	0.001575	0.003212
cardano	0.006355	0.007932	0.005652	0.004102	0.010016	0.010820	0.014545	0.006953	0.003611	0.008979
Dogecoin	0.006678	0.009988	0.007042	0.004248	0.010820	0.028296	0.015195	0.007066	0.003976	0.008687
Solana	0.011359	0.012955	0.010523	0.008098	0.014545	0.015195	0.033157	0.010946	0.007342	0.013879
Litecoin	0.005337	0.006030	0.006043	0.003302	0.006953	0.007066	0.010946	0.010162	0.003307	0.007670
TRON	0.003300	0.003335	0.002859	0.001575	0.003611	0.003976	0.007342	0.003307	0.003068	0.003940
polygon	0.006084	0.008208	0.006623	0.003212	0.008979	0.008687	0.013879	0.007670	0.003940	0.013594

انواع مدل:

سه مدل بهینه‌سازی می‌توان برای این مسئله ارائه داد. در همه این مدل‌ها، ما سود و ریسک داریم. که سود به این صورت تعریف می‌شود:

$$\mu^T x$$

که μ همان آرایه سود یا برگشت توقع شده، است. و همچنین x ، آرایه سرمایه‌گذاری ماست. همچنین ریسک بدین صورت تعریف می‌شود:

$$x^T C x$$

که C همان ماتریس کوواریانس است. در همه مدل‌ها، این شرط که جمع اعضای آرایه سرمایه‌گذاری برابر ۱ باشد، برقرار است.

مدل اول:

این مدل، سود را ماکزیمم می‌کند در حالی که ریسک از مقداری کم تر باشد.

$$\max \mu^T x$$

$$s.t. \ x^T C x \leq r_0$$

$$\sum_{j=1}^k x_j = 1 \text{ and } x_j \geq 0, \ j = 0, 1, 2, \dots$$

مدل دوم:

در این مدل، ریسک کمینه می‌شود در حالی که سود از حدی بیشتر باشد.

$$\min x^T C x$$

$$s.t. \ \mu^T x \geq l_0$$

$$\sum_{j=1}^k x_j = 1 \text{ and } x_j \geq 0, \ j = 0, 1, 2, \dots$$

مدل سوم:

در این مدل تابعی دیگر بهینه می‌شود که در آن، β ، یک ضریب است با توجه به اینکه کدام یک از ریسک یا سود بیشتر برای ما اهمیت دارد.

$$\min x^T C x - \beta \mu^T x$$

$$\sum_{j=1}^k x_j = 1 \text{ and } x_j \geq 0, \ j = 0, 1, 2, \dots$$

که ما مدل سوم را برای حل با الگوریتم‌های خود، انتخاب می‌کنیم.

توضیح درباره "مقدار ریسک پارامتریک" (VaR) و کاربرد آن در مقاله:

مفهوم "مقدار ریسک پارامتریک" (Parametric Value at Risk یا VaR) به عنوان یک سنج ریسک در حوزه مالی استفاده می شود تا برآوردی از خسارت هایی که یک سرمایه گذاری یا پرتفوی ممکن است در بازه زمانی مشخص با سطح اطمینان خاصی روبرو شود، ارائه دهد. این سنج برآوردی از حداکثر خسارتی است که یک پرتفوی می تواند با احتمال خاصی تجربه کند.

در این مقاله، نویسندگان از مقدار ریسک پارامتریک برای ارزیابی ریسک مرتبط با پرتفوی های متشکل از ارزش های رمزنگاری شده و طلا استفاده کرده اند. مقدار ریسک پارامتریک بر پایه فرضیات آماری استوار است و معمولاً فرض می شود که بازدهی دارایی ها از یک توزیع خاص، مانند توزیع نرمال، پیروی می کنند.

برای استفاده از مقدار ریسک پارامتریک، ابتدا پرتفوی مورد نظر را از نظر ارزش اولیه و وزن های مختلف مرتب کنید. سپس با استفاده از میانگین (بازدهی مورد انتظار) و انحراف معیار (پراکندگی) بازدهی هر دارایی در پرتفو، می توانید مقدار ریسک پارامتریک را محاسبه کنید. به عنوان مثال، با استفاده از مقدار ریسک پارامتریک با سطح اطمینان 95٪، می توانید برآوردی از حداکثر خسارتی که پرتفو شما ممکن است با احتمال 5٪ در بازه زمانی مشخص روبرو شود، بدست آورید.

در این مقاله، نویسندگان برای بهینه سازی پرتفوی های متشکل از ارزش های رمزنگاری شده و طلا، از مقدار ریسک پارامتریک استفاده کرده اند. این ابزار به آن ها کمک کرده است تا خطرات مرتبط با سرمایه گذاری در این دو کلاس دارایی را ارزیابی و پرتفوی بهینه ای را بر اساس آن تشکیل دهند.

معادله ۱ نشان دهنده محاسبه مقدار ریسک پارامتریک است:

$$\text{VaR} = \mu - z \cdot \sigma \quad (1)$$

در معادله ۱:

- μ نشان دهنده میانگین بازدهی مورد انتظار است.
- σ نشان دهنده انحراف معیار است که می توان آن را از ماتریس کوواریانس بدست آورد (در واقع قطر این ماتریس، همان واریانس ها هستند).
- z معیاری است که براساس سطح اطمینان تعیین می شود.

تبرید شبیه‌سازی شده با تابع هدف ارزش در معرض خطر VaR طبق مقاله:

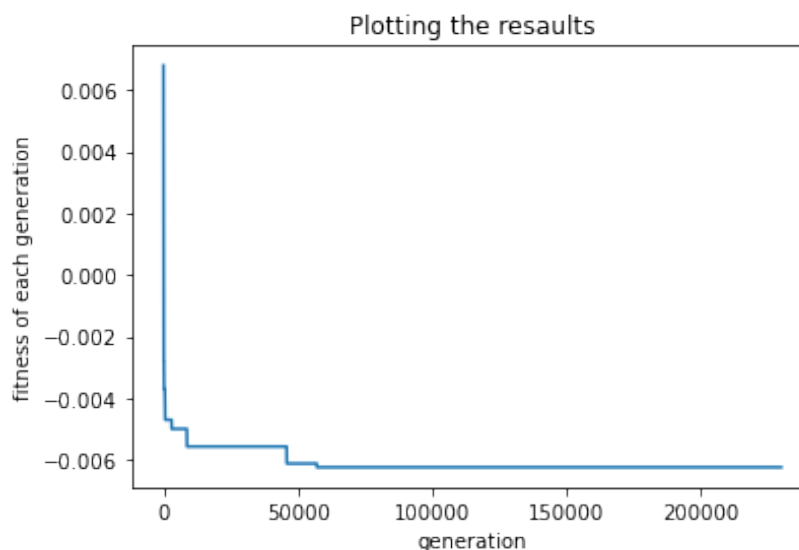
در این الگوریتم ما یک همسایه مورد نظر را انتخاب می‌کنیم و اگر نتیجه بهتری داشت که آن را به عنوان نقطه جدید در نظر می‌گیریم ولی اگر نتیجه بهتری نداشت یک احتمال رندوم را در نظر می‌گیریم که نقطه جدید را انتخاب کنیم که این کار باعث افزایش پویایی و فرار از نقطه بهینه محلی است. در این الگوریتم برای بهبود عملکرد آن دو تا روش برای پیدا کردن همسایه در نظر می‌گیریم که به صورت رندوم وزن دار که وزن آن را خودمان تعیین می‌کنیم یکی را انتخاب می‌کنیم. در روش اول فقط دو اندیس را باهم جابه‌جا می‌کنیم و در روش دوم به صورت رندوم وزن دار چند نقطه را انتخاب می‌کنیم و آن را با یک عدد رندوم جایگذاری می‌کنیم و در آخر هر عنصر را بر مجموع لیست تقسیم می‌کنیم تا مجموع یک باقی بماند.

در این الگوریتم ما تابع هدف را طبق مقاله Value at risk (ارزش در معرض خطر) قرار می‌دهیم. که خود انواع مختلفی دارد ولی مانند مقاله تابع هدف را parametric Var (مقدار پارامتریک در معرض خطر) قرار می‌دهیم.

الگوریتم به طور کامل همراه با نحوه محاسبه Var در نوت بوک پیاده سازی شده است. پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می‌دهد:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.001	0.348	0.285	0.010	0.009	0.004	0.015	0	0.016	0.308

بنابراین، var ما برابر 0.00623819- خواهد بود.
نمودار تغییرات تابع برای این پاسخ، به این صورت است:



تبرید شبیه سازی شده:

این الگوریتم نیز مانند الگوریتم قبلی است. تنها تفاوت آن در این است که تابع هدف آن همان

$$x^T C x - \beta \mu^T x$$

است. الگوریتم به طور کامل در نوک بوک پیاده سازی شده.

Algorithm 1 Simulated Annealing (SA) Algorithm

```

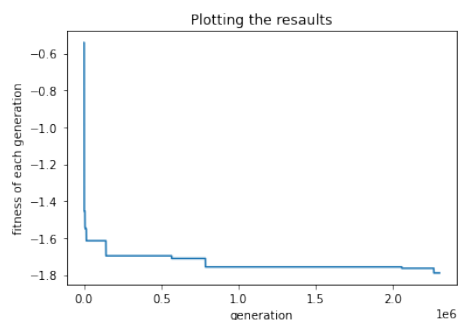
1: Initialize the current state:  $s \leftarrow s_{\text{initial}}$ 
2: Initialize the temperature:  $T \leftarrow T_{\text{initial}}$ 
3: while stopping criterion not met do
4:   Generate a candidate state:  $s' \leftarrow \text{generateCandidate}(s)$ 
5:   Calculate the energy difference:  $\Delta E \leftarrow \text{energy}(s') - \text{energy}(s)$ 
6:   if  $\Delta E < 0$  then
7:     Accept the candidate state:  $s \leftarrow s'$ 
8:   else
9:     Calculate the acceptance probability:  $p \leftarrow \exp\left(\frac{-\Delta E}{T}\right)$ 
10:    Generate a random number:  $r \leftarrow \text{random}(0, 1)$ 
11:    if  $r < p$  then
12:      Accept the candidate state:  $s \leftarrow s'$ 
13:    end if
14:  end if
15:  Update the temperature:  $T \leftarrow \text{coolingSchedule}(T)$ 
16: end while
17: return the best state found

```

پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می دهد ($\beta = 100$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0.010	0.787	0	0.009	0	0.011	0	0.059	0.121

بنابراین، سود ما برابر 0.0179 و ریسک برابر 0.0084 خواهد بود.
همچنین نمودار تغییرات تابع برای این پاسخ، به این صورت است:



الگوریتم بهینه سازی ازدحام ذرات:

در این الگوریتم، نقاط با توجه به موقعیتی که دارند، نقاط دیگر را جذب خود می کنند. بنابراین این الگوریتم برای این مسئله، که مسئله ای پیوسته است، احتمالا خیلی خوب عمل خواهد کرد. در زیر، گام های الگوریتم آمده است. الگوریتم به طور کامل در نوک بوک پیاده سازی شده.

Algorithm 2 Particle Swarm Optimization (PSO) Algorithm

```

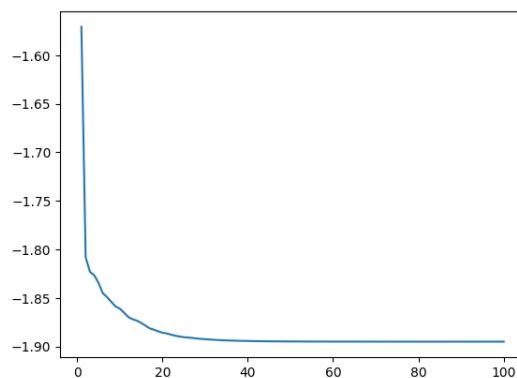
1: Initialize particle's positions and velocities
2: Initialize global best position and fitness
3: while stopping criterion not met do
4:   for all particles do
5:     Update particle's velocity using Eq. (1):
6:      $v_{ij} = w \cdot v_{ij} + c_1 \cdot r_1 \cdot (pbest_{ij} - x_{ij}) + c_2 \cdot r_2 \cdot (gbest_j - x_{ij})$ 
7:     Update particle's position using Eq. (2):
8:      $x_{ij} = x_{ij} + v_{ij}$ 
9:     Update particle's fitness value
10:    Update particle's personal best position and fitness
11:    if particle's fitness is better than global best fitness then
12:      Update global best position and fitness
13:    end if
14:  end for
15: end while
16: return global best position and fitness

```

پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می دهد ($\beta = 100$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.9486	0	0	0	0	0	0	0.0513

بنابراین، سود ما برابر 0.0190 و ریسک برابر 0.0096 خواهد بود. همچنین نمودار تغییرات تابع برای این پاسخ، به این صورت است:



الگوریتم جاذبه:

این الگوریتم بسیار شبیه به الگوریتم بهینه سازی ازدحام ذرات است. در این الگوریتم اجسام با وزن بیشتر (هزینه کم تر)، به یکدیگر جذب می شوند تا در نهایت به یک پاسخ مناسب برسیم. الگوریتم به طور کامل در نوک بوک پیاده سازی شده.

Algorithm 3 Gravity Algorithm

```

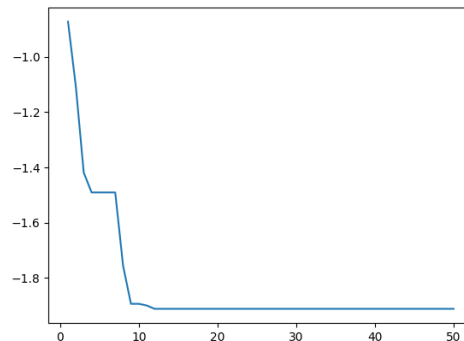
1: Initialize population
2: Evaluate the fitness of each solution
3: while stopping criterion not met do
4:   Calculate the total mass of the population:  $M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)}$ 
5:   for all particles do
6:     Calculate the gravitational constant:  $G(t) = G_0 e^{\frac{-\alpha t}{T}}$ 
7:     Calculate the Force:  $F_{ij} = G(t) \times \frac{Mass_i(t) \times Mass_j(t)}{R_{ij}(t) + \epsilon} (x_j(t) - x_i(t))$ 
8:     Calculate the acceleration:  $a_i = \frac{F_i}{M_i}$ 
9:     Update the velocity:  $v_i = v_i + a_i \cdot \Delta t$ 
10:    Update the position:  $x_i = x_i + v_i \cdot \Delta t$ 
11:   end for
12:   Evaluate the fitness of each solution
13:   Sort the population based on fitness
14:   Select the elite solutions
15:   Apply elitism strategy to update the population
16: end while
17: return the best solution found

```

پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می دهد ($\beta = 100$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.8921	0.1078	0	0	0	0	0	0

بنابراین، سود ما برابر 0.0181 و ریسک برابر 0.0105 خواهد بود. نمودار تغییرات:



الگوریتم ژنتیک:

در این الگوریتم ابتدا یک جمعیت اولیه رندوم تولید میکنیم سپس در هر تکرار ابتدا با انتخاب والدین مناسب (در الگوریتم پیاده سازی شده با tournament selection)، فرزندان را از بازترکیب والدین ساخته و سپس روی آن‌ها جهش می‌دهیم. که برای جهش دو روش را انتخاب کردیم که هربار به صورت رندوم وزن دار یکی را انجام می‌دهیم. یکی اینکه تعداد تصادفی از ژنوم‌های هر کروموزوم (پاسخ) را یک مقدار تصادفی بین ۰ و ۱ می‌دهیم و سپس آن را به فضای مورد نظر، انتقال می‌دهیم و دیگری اینکه به صورت رندوم مقدار دو تا از اندیس‌های کروموزوم را تغییر می‌دهیم. الگوریتم به طور کامل در نوک بوک پیاده سازی شده. همچنین الگوریتم کلی ژنتیک در زیر آمده است:

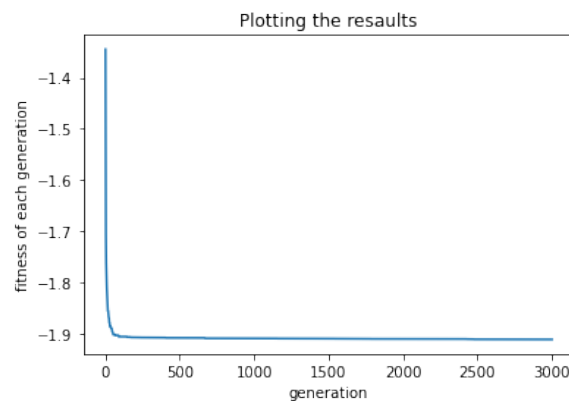
Algorithm 4 Genetic Algorithm (GA)

- 1: Initialize population using method M
 - 2: Evaluate the fitness of each individual using fitness function f
 - 3: **while** stopping criterion not met **do**
 - 4: Select parents for reproduction using selection method S
 - 5: Perform crossover using crossover operator C with probability P_C
 - 6: Perform mutation using mutation operator M with probability P_M
 - 7: Evaluate the fitness of each offspring using fitness function f
 - 8: Select individuals for the next generation using selection method S
 - 9: **end while**
 - 10: **return** the best individual found based on fitness function f
-

پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می‌دهد ($\beta = 100$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.000	0.0001	0.9991	0.0001	0	0	0.0001	0	0.0002	0.0001

بنابراین، سود ما برابر 0.0192 و ریسک برابر 0.0101 خواهد بود. نمودار تغییرات:



ترکیب ژنتیک و تبرید شبیه‌سازی شده:

در این الگوریتم ترکیبی می‌خواهیم به گونه‌ای از الگوریتم SA استفاده کنیم که در ابتدا (دمای زیاد) پویش ما بیشتر باشد و سپس با گذشت زمان (کاهش دما) پویش کم شده و انتفاع افزایش یابد. ایده به این صورت است که از $\frac{1}{iteration}$ برای دما در هر مرحله از همان الگوریتم SA که قبلاً استفاده کردیم، استفاده کنیم.

بنابراین در هر نسل، الگوریتم ژنتیک را انجام می‌دهیم. یعنی ابتدا با انتخاب والدین مناسب (در الگوریتم پیاده‌سازی شده با tournament selection)، ابتدا فرزندان را از بازترکیب والدین ساخته و سپس روی آن‌ها جهش می‌دهیم. این قسمت، همان قسمت ترکیب شده با SA است. در واقع در اینجا TL بار، جهش انجام می‌دهیم. اگر جهش بهتر بود، آن را جایگزین می‌کنیم، اگر نه، با همان احتمال در الگوریتم تبرید، این دو را جابه‌جا می‌کنیم. و سپس ادامه کار مانند الگوریتم ژنتیک خواهد بود.

در این الگوریتم برای جهش، تعداد تصادفی از ژنوم‌های هر کروموزوم (پاسخ) را یک مقدار تصادفی بین ۰ و ۱ می‌دهیم و سپس آن را به فضای مورد نظر، انتقال می‌دهیم.

Algorithm 5 GA-SA: Genetic Algorithm with Simulated Annealing Mutation

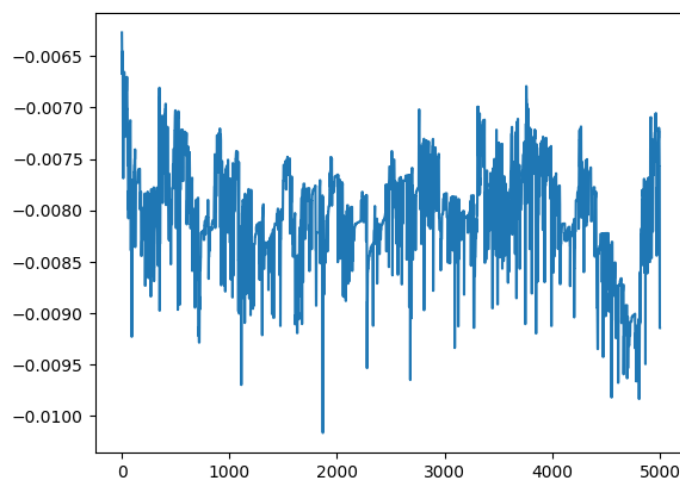
- 1: Initialize population for GA
 - 2: Evaluate the fitness of each individual
 - 3: **while** stopping criterion not met **do**
 - 4: Select parents for reproduction
 - 5: Perform crossover to create offspring
 - 6: **if** $\text{random}(0, 1) < p_m$ **then** ▷ Perform mutation
 - 7: Apply Simulated Annealing mutation to the offspring
 - 8: **end if**
 - 9: Evaluate the fitness of each offspring
 - 10: Select individuals for the next generation
 - 11: **end while**
 - 12: **return** the best individual found
-

پس از اجرا، الگوریتم به ما آرایه سرمایه‌گذاری زیر را می‌دهد ($\beta = 1$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.0047	0.0723	0.4599	0.0067	0.0047	0.0011	0.0289	0.0022	0.0001	0.419

بنابراین، سود ما برابر 0.0165 و ریسک برابر 0.0063 خواهد بود. نمودار تغییرات برای این پاسخ، در صفحه‌ی بعد آمده است. همچنین یک حالت دیگر با پارامترهای مختلف بررسی شده است.

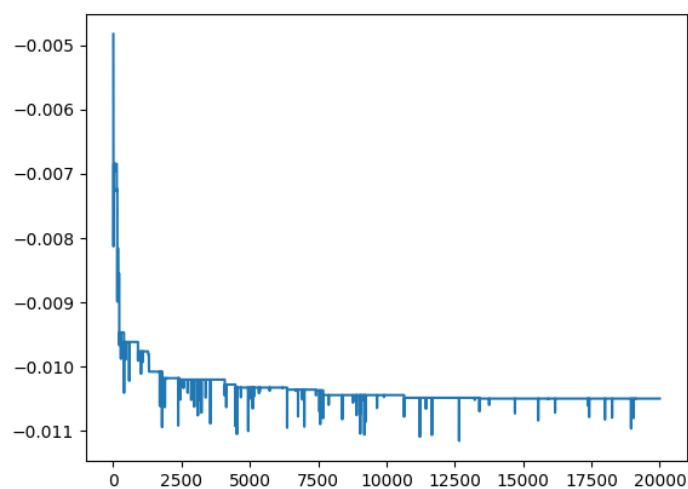
در این حالت، احتمال باز ترکیب برابر 0.9، احتمال جهش برابر 0.9 و درصد جایگزینی فرزندان و والدین در جمعیت، برابر 0.5 است.



در این حالت، احتمال باز ترکیب برابر 0.8، احتمال جهش برابر 0.5 و درصد جایگزینی فرزندان و والدین در جمعیت، برابر 0.4 است. می بینیم که این نمودار، رفتار بهتری نسبت به حالت قبل دارد، همچنین آرایه سرمایه گذاری آن در زیر آمده است.

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.0014	0.0313	0.5377	0.0007	0.0006	0.0009	0.0041	0.0015	0.0000	0.4213

بنابراین، سود ما برابر 0.0173 و ریسک برابر 0.0068 خواهد بود.



ترکیب ژنتیک و بهینه سازی ازدحام ذرات:

در این الگوریتم می‌خواهیم به گونه‌ای از الگوریتم PSO استفاده کنیم تا پاسخ‌های سریع‌تر و بهتری بدست آوریم (نسبت به الگوریتم ژنتیک تنها). برای اینکار، در هر نسل ابتدا یک بار الگوریتم PSO را اجرا می‌کنیم تا اعضای جمعیت کمی بهتر شوند و هر کدام مقداری به پاسخ‌های بهتر، نزدیک شوند. بنابراین همان الگوریتم ژنتیک را خواهیم داشت که در هر نسل، ابتدا روی اعضای جمعیت الگوریتم PSO اجرا می‌کنیم و سپس مابقی کارها در این نسل، همان الگوریتم ژنتیک خواهد بود (بازترکیب و جهش و ...).

در این پیاده سازی، از tournament selection برای انتخاب والدین، و از Whole Arithmetic برای بازترکیب استفاده شده. همچنین برای جهش، تعداد تصادفی از ژنوم‌های هر کروموزوم (پاسخ) را یک مقدار تصادفی بین ۰ و ۱ می‌دهیم و سپس آن را به فضای مورد نظر، انتقال می‌دهیم.

Algorithm 6 GA-PSO: Genetic Algorithm with Particle Swarm Optimization

- 1: Initialize population for GA
 - 2: Evaluate the fitness of each individual
 - 3: **while** stopping criterion not met **do**
 - 4: Apply PSO to improve the population
 - 5: Evaluate the fitness of each individual
 - 6: Select parents for reproduction
 - 7: Perform crossover to create offspring
 - 8: Perform mutation on the offspring
 - 9: Evaluate the fitness of each offspring
 - 10: Select individuals for the next generation
 - 11: **end while**
 - 12: **return** the best individual found
-

در این الگوریتم برای اجرای PSO هدف این است که به صورتی عمل کنیم که در ابتدا، پویا زیاد باشد و سپس با گذشت زمان، پویا کاهش یافته و کروموزوم‌ها کم‌تر از قبل جابه‌جا شوند و پاسخ‌ها ثابت شوند.

برای این کار به این صورت عمل می‌کنیم که، پارامترهای w, c_1, c_2 را با توجه به نسلی که در آن قرار داریم، تغییر می‌دهیم. برای مثال داریم:

$$c_1 = \frac{10 \times c_1}{iteration} \quad c_2 = \frac{10 \times c_2}{iteration} \quad w = \frac{10 \times w}{iteration}$$

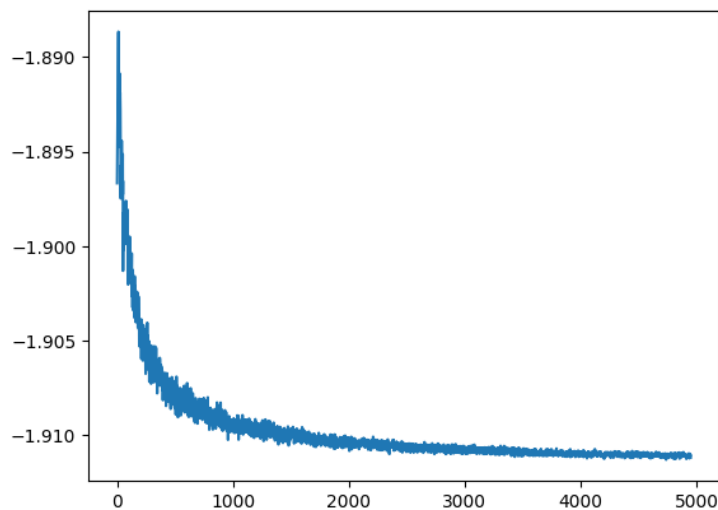
در این صورت، این پارامترها با گذشت نسل، کاهش میابند و تاثیر PSO با گذشت زمان کم‌تر می‌شود و جمعیت کم‌تر با این الگوریتم تغییر می‌کند.

در نمودار صفحه بعد، این نوسانات به طور کامل مشخص است که در ابتدا به واسطه الگوریتم PSO بیشتر است و سپس با گذشت نسل، این نوسانات کم‌تر می‌شود.

پس از اجرا، الگوریتم به ما آرایه سرمایه گذاری زیر را می‌دهد ($\beta = 100$):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.9992	0	0	0	0.0001	0	0	0.0004

بنابراین، سود ما برابر 0.0192 و ریسک برابر 0.0101 خواهد بود.



نتایج در مقابل واقعیت:

در این بخش می‌خواهیم بررسی کنیم که اگر با استفاده از نتایج بدست آمده از هر کدام از این الگوریتم‌ها، خرید کنیم، در کدام حالت سود بیشتری خواهیم کرد. ابتدا قیمت‌ها در روز 2 july را می‌بینیم:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.667	0.0757	110.22	19.241	0.0670	0.2903	0.4802	244.7	1912.3	30491.6

حال قیمت‌ها در روز 9 july (هفت روز بعد):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.685	0.08	97.01	21.622	0.0657	0.2867	0.4703	235	1868.27	30307.1

بنابراین جدول تغییرات به این صورت خواهد بود (به این معنی که به ازای یک واحد خرید از هر کدام، چه مقدار ضرر می‌کنیم):

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.026	0.056	-0.119	0.123	-0.019	-0.012	-0.02	-0.039	-0.023	-0.006

با توجه به بازار خوبی (___) که پشت سر گذاشتیم توی این هفته، تقریباً هر چی خریداری بشه، ضرر خواهد بود. و با تعجب بیشتر، ارزی که بیشترین ریسک رو برای خریداری داره، بیشترین سود رو به ما خواهد داد. |||:

نتیجه اخلاقی: بعضی وقتا تو زندگی باید ریسک کرد...

حالا با توجه دوباره به این بازار خوب، بریم ببینیم کدام یک از این الگوریتم‌ها، سود بیشتری به ما می‌رسونن. (متأسفانه تو این هفته، میشه ضرر کم‌تر. ولی خب ما حالت کلی و منطقی رو در نظر میگیریم)

تبرید شبیه‌سازی شده با تابع هدف ارزش در معرض خطر VaR طبق مقاله:

با توجه به این آرایه سرمایه‌گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.001	0.348	0.285	0.010	0.009	0.004	0.015	0	0.016	0.308

سود از 0.0140 تبدیل به -0.0159- میشود.

تبرید شبیه‌سازی شده:

با توجه به این آرایه سرمایه‌گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0.010	0.787	0	0.009	0	0.011	0	0.059	0.121

سود از 0.0179 تبدیل به -0.095- میشود.

الگوریتم بهینه سازی ازدحام ذرات:

با توجه به این آرایه سرمایه‌گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.9486	0	0	0	0	0	0	0.0513

سود از 0.0190 تبدیل به -0.1131- میشود.

الگوریتم جاذبه:

با توجه به این آرایه سرمایه‌گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.8921	0.1078	0	0	0	0	0	0

سود از 0.0181 تبدیل به -0.0929- میشود.

الگوریتم ژنتیک:

با توجه به این آرایه سرمایه گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.000	0.0001	0.9991	0.0001	0	0	0.0001	0	0.0002	0.0001

سود از 0.0192 تبدیل به -0.1188- میشود.

ترکیب ژنتیک و تبرید شبیه سازی شده:

با توجه به این آرایه سرمایه گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0.0014	0.0313	0.5377	0.0007	0.0006	0.0009	0.0041	0.0015	0.0000	0.4213

سود از 0.0173 تبدیل به -0.0648- میشود.

ترکیب ژنتیک و بهینه سازی ازدحام ذرات:

با توجه به این آرایه سرمایه گذاری:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
0	0	0.9992	0	0	0	0.0001	0	0	0.0004

سود از 0.0192 تبدیل به -0.1189- میشود.

مقایسه و نتیجه:

با توجه به جدولی که انتظار داشتیم به ازای یک واحد، مقداری سود برساند:

poly	TRON	Lite	solana	Doge	card	XRP	BNB	ETH	BTC
-0.004	0.009	0.019	0.009	0.003	-0.004	0.008	-0.003	0.011	0.015

و همچنین با توجه به ماتریس کوواریانس، خرید ارز Litecoin به طور قطع بهترین انتخاب برای ما است. که نتایج و پاسخ های نهایی الگوریتم ها هم نشان دهنده همین موضوع است. در واقع

در همه ی الگوریتم ها به درستی به دنبال خرید Litecoin رفته اند و در اکثر آن ها مقدار خریداری شده از این ارز تقریباً به میزان همان ۱ واحد بوده و همچنین مینیمم سراسری تابع معرفی شده هم در همین حالت اتفاق می افتد.

با بررسی نمودار تغییرات قیمت Litecoin و solana، دلیل این اتفاق (ضرر کردن به جای سود کردن) را متوجه می شویم.



Solana (ب)



Litecoin (ا)

بنابراین، با این نمودارها، ما انتظار داشتیم که Litecoin همچنان به افزایش قیمت ادامه دهد ولی چنین نشد و برعکس، برای Solana انتظار داشتیم که کمی تغییر قیمت داشته باشد ولی با وجود ریسک بالا (طبق جدول کوواریانس) سرمایه‌گذاری روی آن خیلی عاقلانه نباشد، در صورتی که افزایش قیمت چشم‌گیری داشته و تغییرات زیادی به خودش دیده.

حال فرض کنیم همه چیز منطقی و با توجه به تجربه گذشته و داده‌هایی که در اختیار داریم پیش برود. در این صورت ترتیب عملکرد الگوریتم‌ها از لحاظ پیدا کردن پاسخ بهتر (در نظر گرفتن سود در عین زیان) به صورت زیر خواهد بود:

1. Genetic combined with PSO
2. Genetic
3. PSO
4. Gravity
5. SA
6. Genetic combined with SA
7. SA using VaR as a cost function

بنابراین همه الگوریتم‌های پیاده‌سازی شده، از الگوریتم مقاله که با استفاده از تابع VaR پیاده‌سازی شده بود، بهتر عمل کردند.