Nature-Inspired Portfolio Diversification Using Ant Brood Clustering

Ashish Lakhmani(✉), Ruppa K. Thulasiram, and Parimala Thulasiraman

Department of Computer Science, University of Manitoba, Winnipeg, Canada
lakhmana@myumanitoba.ca,
{tulsi.thulasiram,parimala.thulasiraman}@umanitoba.ca

Abstract. Portfolio diversification is a crucial strategy for mitigating risk and enhancing long-term returns. This paper introduces a unique approach to large-scale diversification using Ant Brood Sorting clustering, a nature-inspired algorithm, in conjunction with co-integration measure of time series. Traditional diversification strategies often struggle during uncertain market times. In contrast, the proposed method leverages Ant Brood Sorting to group similar stocks based on the co-integration of their closing prices. This approach allows for the creation of diversified portfolios from a wide range of stocks. The study presents promising results, with clusters of stocks showing both high correlation and cosine similarity, validating the effectiveness of the approach. Silhouette score, a measure of cluster quality, and inter-cluster analysis demonstrate support in validating the results of the study by displaying similarities between the stocks being clustered and distinctiveness with stocks in other clusters. The research contributes to the application of nature-inspired algorithms in large-scale portfolio diversification, offering potential benefits for investors seeking resilient and balanced portfolios.

Keywords: Ant Brood Sorting · Portfolio Diversification · Cointegration · Clustering

1 Introduction

Portfolio Diversification (PD) involves spreading investments across a variety and different types of assets to reduce the overall risk of the portfolio and enhance the potential for long-term returns. The idea behind PD is to avoid putting all the eggs into one basket, i.e. to avoid putting the bulk of the total portfolio budget in similar types of assets so that the risk exposure to any one kind of asset is limited. Different asset types perform differently under diverse market conditions. By diversifying the investments, investors mitigate the impact of individual assets or similar kinds of assets on the whole portfolio and safeguard the capital. While PD may not accurately predict the highest return on assets, it spreads the risk effectively. Diversification not only helps in mitigating risk but also provides the opportunity to reap benefits in different segments of assets, achieving a balanced and resilient portfolio that stands the test of time.

S. Smith et al. (Eds.): EvoApplications 2024, LNCS 14634, pp. 115–130, 2024.
https://doi.org/10.1007/978-3-031-56852-7_8

The process of building a portfolio involves the selection of assets and finding suitable weights to allocate to each asset. PD entails the selection of assets in order to spread the unsystematic risk of the portfolio across all the assets in the portfolio, whereas Portfolio Optimization entails deciding the appropriate weights of each asset in the portfolio to maximize the overall return of the portfolio while minimizing overall risk. With an increase in the number of assets under a portfolio, it becomes challenging when there is a vast number of assets to choose from [11]. In recent years, nature-inspired algorithms have been considered on a large scale in computational finance literature [4]. The benefits of using nature-inspired algorithms come from their ability to quickly explore the possible solutions to a problem and efficiently exploit the solutions to improve upon them.

The conventional way to attain diversification in a portfolio is to select stocks from different asset classes, different industry sectors, or different geographical regions [21]. Some of the common diversification strategies are based on concepts such as the law of large numbers, correlation, capital asset pricing model, and risk parity. These diversification strategies, however, have failed to work when diversification was needed the most for risk aversion [12]. In this study, we present a PD strategy to group similar types of stocks by applying a nature-inspired heuristics called Ant Brood Sorting clustering technique on the statistical property of stocks' closing prices.

The remainder of this paper is structured as follows: Sect. 2 presents the related works of using nature-inspired computing in financial time series and discusses the motivation behind this study. Section 3 presents the definitions of the methods used in this study. Section 4 presents the experiment setup in detail along with the implementation of the experiment. Section 5 shows the results obtained and Sect. 6 concludes this study.

2 Related Work and Motivation

The legacy portfolio creation used classical time series models in creating optimal portfolios. Many professionals still use these time series models in the stock selection process before forming a portfolio [19] despite the fact that time series models have been shown to be inferior to computational algorithmic models [8]. Stock selection has long been recognized as a difficult and crucial task. Choosing stocks for successful portfolio development is heavily reliant on trustworthy stock ranking. Recent breakthroughs in machine learning and data mining have created substantial opportunity to handle these difficulties more effectively [10]. Huang [10] used Support Vector Regression (SVR) and Genetic Algorithms (GAs) to create a stock selection model. Their model used SVR to forecast each stock's future return, while GA optimized model parameters and input data.

Portfolio selection using nature-inspired algorithms has shown advantage over traditional methods because of superior searching ability through the heuristics [1]. Oduntan et al. [21] used Ant Brood Sorting clustering method to gain intelligence from time series data and use that intelligence to form clusters of similar stocks to create diversified portfolios. Liu et al. [16] used a variation of Ant Brood Clustering (ABC) to cluster financial time series data and received a high-quality clustering result as depicted by the intra-cluster distance. ABC Sorting has also been found to have promising results when hybridized with other algorithms [20].

Meta-heuristic algorithms have been proven to find the best answers for a wide range of complex and unique portfolio models [11]. Durán et al. [5] explored using memetic algorithm for multiobjective investment portfolio optimization with cardinality restrictions in the context of the Markowitz model. Hasan et al. [9] used whale optimization algorithm, a nature-inspired approach that mimics the haunting process of the sea whale, for portfolio optimization on the data-set of DAX-100, the German stock exchange index consisting of 100 stocks. Oduntan et al. [20] tested using and brood sorting clustering algorithm based on grouping of broods amongst ants, to gather financial intelligence from time-series of 30 stocks for portfolio diversification. Meng et al. [15] used grey wolf optimizer, a meta-heuristic optimizing algorithm inspired by the hunting behavior of grey wolves, for stock selection out of 200 stocks. Mazumdar et al. [18] used swarm intelligence for portfolio optimization and construction from a pool of 100 stocks. Shahid et al. [25] presented a novel portfolio selection strategy using gradient-Based Optimizer on a data-set of 30 stocks and compared its performance with a particle swarm optimization approach. There are about 65,000 stocks listed in stock exchanges worldwide. To achieve an effective diversification, it is essential to consider a vast number of stocks. A broad selection of stocks across different sectors, industries, and market segments can help mitigate the impact of poor-performing stocks on the overall portfolio.

To best of our knowledge, there hasn't been any prior research that utilizes a nature-inspired algorithm for selecting stocks across a wide spectrum of stocks for portfolio diversification. Moreover, Ant Brood Clustering, one of the interesting nature-inspired algorithms, has not been used in prior studies for portfolio diversification with a large number of stocks, the focus of this study. Therefore, our study represents a distinctive contribution to the application of a nature-inspired algorithm for a large scale portfolio diversification.

3 Ant Brood Clustering

Deneubour et al. [6] proposed a computing model inspired by behavior of ant colonies that clean their nest by collecting and organizing corpses into piles. The core idea is that ants wander in the nest to pick corpses from isolated areas and drop corpses where more related and similar items are, as shown in Figs. 1 and 2, thus growing the clusters in the colony. The likelihood of ants picking and dropping corpses are calculated mathematically. One uniqueness of

this clustering algorithm is that we do not predefine the number of clusters to be formed. The algorithm is agnostic of number of clusters, it creates the clusters as it deems necessary as per the underlying mathematical formulas.

3.1 Measuring Object Similarity for Clustering

Lumer and Faieta [17] proposed a variation of the work by Deneubour et al. [6] by introducing a way to measure similarity between objects in the swarm when clustering. Given a 2-d grid (m x m) of spacial terrain where elements/objects are laid out randomly, ants, also referred as agents, perform a random walk on the grid. When an unladen ant gets to a point in the grid which has an element present in that grid, the probability of an ant to pick that element is given by:

$$P_p = \left(\frac{k_1}{k_1 + f}\right)^2 \tag{1}$$

whereas when a laden ant reaches to an empty point in the grid, the probability of that ant to drop the element is given by:

$$P_d = \left(\frac{f}{k_2 + f}\right)^2 \tag{2}$$

where k_1 and k_2 are constants. f is the similarity density measure and is calculated as:

$$f(o_i) = \frac{1}{s^2} \sum_{o_i \epsilon Neigh(s*s)(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha}\right] if \quad f > 0 \tag{3}$$

$$Otherwise f(o_i) = 0$$

where $d(o_i, o_j)$ is a measure of the similarity distance between the object o_i and another object o_j within its neighborhood, $s * s$ is the number of grids in the neighborhood of object o_i, and α is a parameter used to define the scale for dissimilarity, i.e. how close two items should be to be considered close. The similarity measure that we use in this study is the level of co-integration between the stocks and is described in detail in the following subsection.

Clustering algorithms are techniques used to group similar items together. We apply ant brood sorting clustering to identify group of similarly behaving assets for portfolio diversification, which will help in constructing diversified portfolios.

3.2 Co-integration

Co-integration refers to a long-term statistical relationship between two or more time series that move together in a stable way, though the time series individually may have short-term fluctuations or trends. Co-integration shows the equilibrium connection between different individual time series and helps explain

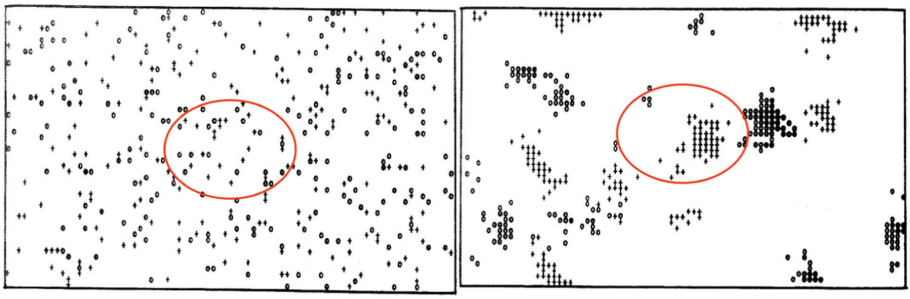


Fig. 1. Ant Brood Clustering Process (Adopted from [6])

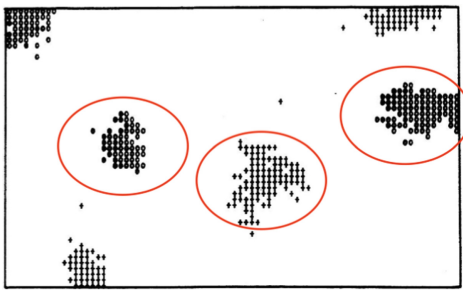


Fig. 2. Clusters of similar kind of items formed (Adopted from [6])

their behavior over time. The concept of co-integration was introduced by Engle and Granger [7] and is commonly known as Engle-Granger co-integration theory. This concept is heavily used in the finance industry, predominantly in a trading strategy called pair-trading [13,24,26], and [14].

Engle-Granger cointegration test performs the following two-step process that determines if there is co-integration between two time series [3]:

Step-1 Augmented Dickey-Fuller(ADF) Test: Conduct unit root test of both time series to determine if both time series have the same order of integration. The ADF test is applied using the model:

$$\Delta Y_t = \alpha + \beta t + \omega . Y_{t-1} + \sum_{i=1}^{k} \delta_i \Delta Y_{t-1} + \epsilon_t \tag{4}$$

The null hypothesis of the ADF test is that $\omega = 0$, which implies the presence of a unit root (non-stationarity), and evidence that $\omega < 0$ implies stationarity. Perform the ADF test for each time series and record the ADF test statistics using Eq. (5) and check if the null hypothesis can be rejected.

$$DF_\tau = \frac{\hat{\omega}}{SE(\hat{\omega})} \tag{5}$$

Step-2 Estimate the Co-integration Relationship Between Both Time Series: Use the standard Ordinary Least Square (OLS) regression and test for the stationarity in the residuals obtained by Eq. (6). OLS regression is a widely used statistical method for finding the best-fitting straight line as close as possible to the data-points in a linear regression model. It finds the estimated values of α and β by minimizing the error term ε_t. If the statistics value are lower than a critical value (usually 0.01 or 0.05) in the stationarity test (Eq. 7), we say that two time series are co-integrated.

$$Y_t = \alpha + \beta X_t + \varepsilon_t \rightarrow \varepsilon_t = Y_t - \alpha - \beta X_t \tag{6}$$

$$Check : \varepsilon_t \sim I(0) \tag{7}$$

For this study, we use an open-source python library [23] that runs the co-integration between two time series and provides t-statistic of unit-root test on residuals and P-value as results. We use the P-value, the measure of probability of cointegration between two time series as the similarity measure $d(o_i, o_j)$ in Eq. (3).

4 Dataset, Algorithm and Experiments

This study focuses on employing a nature-inspired algorithm to choose stocks from a broad range of options for the purpose of portfolio diversification.

4.1 Dataset

We use the individual stocks from the S&P 500 index for this study. The S&P 500 index is widely regarded as a benchmark for the overall performance of the United States of America (U.S) stock market. It consists of 500 large, established companies from various sectors, representing a good portion of the total market capitalization in the United States. It comprises companies from different sectors, including technology, finance, healthcare, consumer goods, etc., therefore, allowing us to examine the clustering on a broad scale of stocks. We use yfinance python library to download the daily adjusted closing prices of individual stocks of S&P 500 index for the past 8 years, from July 2015 to June 2023. We compute the P-value of all the pairs of stocks from these 500 stocks and save them in the cache memory to use in our experiment.

4.2 Implementation

We first implement the Ant Brood Sorting coupled with the co-integration test's P-value to test if this experiment can create cluster stocks of similar types of stocks. Algorithm 1 illustrates the steps that we did in this experiment to cluster similar types of stocks. We begin with initializing a 2-D grid (m x m) and place stocks and ants randomly on the grid. The value of m can be user-defined, we

use Boryczka's [2] recommendation of $m = \sqrt{10 * n}$, where n is the number of stocks to be clustered.

Next, until the iteration termination condition is met, we keep looping through all the ants in each iteration. We check for each Ant if it's unladen and if there's a stock present at Ant's current location. If both conditions are true, we calculate the probability of Ant picking up that stock by comparing the similarity of the stock at Ant's location with stocks in its neighborhood. Similarly, if Ant is laden and its current location is free of any stock, we calculate the probability of Ant's dropping the laden stock by comparing the similarity of the stock with stocks in the neighborhood. If the probability is greater than a pre-determined user value, the ants pick or drop the stock at their position, respectively.

After picking/dropping the stock, the ants randomly move to a new spot in the grid within a predefined neighborhood. If the ant is laden, the priority is given to an empty site, and if the ant is unladen, we give priority to a site occupied by a stock. If no desirable sites are available, ant moves to any random site in the neighborhood.

To handle a situation of overlapping of a site that already has a stock with a laden ant moving to this site, we keep the stock laden by the ant hidden so that no other ant can pick this already laden stock and we also restrict the laden ant from dropping the stock at that site so that the site doesn't have two or more stocks at a single site. The neighborhood that the ants explore for their next step is bigger than the neighborhood used for calculating the probability of pick-up/drop-off actions. This improves the ants' ability to navigate through the spatial terrain for better and more efficient exploration of sites to pick/drop stocks. Ants move along the grid and perform pick-up and drop-off of stocks during each iteration based on the availability of stocks, empty sites, probability, and similarity/dissimilarity of stock within the neighborhood. The iteration terminates if, for a user-defined number of consecutive times, there is no pick or drop performed by the ants and all the ants are unladen. This termination condition makes sure that the iterations terminate when global optima is obtained. At the end of the iterations, we observe the resulting clusters and check for the validity of clusters if they have similar kinds of stocks.

4.3 Parameter Tuning

k_1 and k_2 are the threshold constants for picking and dropping, respectively, in Ant Brood Clustering. The value of these constants will have to be set in a way that when f (similarity measure) is $<< k_1$ then probability of picking up an item is close to 1, whereas when $f << k_2$ then probability of dropping an item is close to 0. R_p and R_d are comparator probability of pick and drop actions of the ants. These values are user-defined between 0 and 1 and are used to accelerate or brake the pick/drop speed of ants in the grid. We set these parameters in a way that the picking and dropping are set in a controlled yet loose fashion. We didn't intend to keep the movement too tight or too loose as it may cause a bottleneck or wandering explosion. The main objective of this algorithm is to

see if it can cluster similar kinds of stocks, so we need a good flow of picking and dropping for forming clusters. For the process termination, we check for 1000 iterations for no pick or drop performed by the ants.

Algorithm 1. Ant Brood Clustering

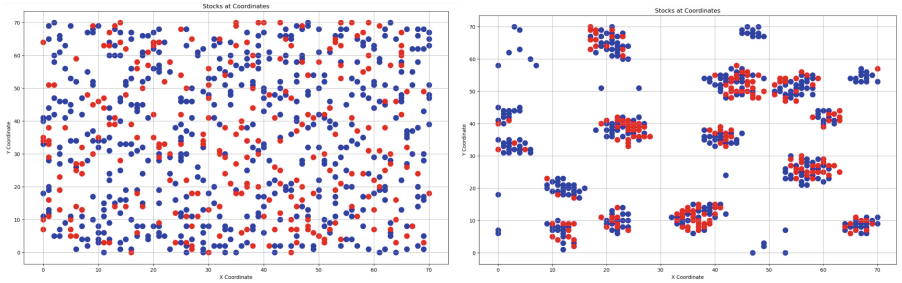
```
INITIALIZE: Stocks and Ants randomly on the 2-d grid.
while Iterations termination condition is False do
  for each ant i do
    if Ant is unladen and the site at current location of ant has a stock then
      Compute f(O_i) in the neighborhood using equation (3).
      Compute probability of picking up the item (P_p) using equation (1).
      Predetermine a pick-up probability comparator R_p between 0 and 1.
      if R_p < P_p then
        Ant picks up the stock.
      end if
    else if ant is laden and the site at current location of ant is empty then
      Compute f(O_i) in the neighborhood using equation (3).
      Compute probability of dropping the item (P_d) using equation (2).
      Predetermine a pick-up probability comparator R_d between 0 and 1.
      if R_d < P_d then
        Drop the item at ants current site.
      end if
    end if
    Move the ant to next random site in the exploration neighborhood as per (4.2)
  end for
  Check for iteration termination condition mentioned in 4.2.
end while
Plot the clusters formed by final locations of stocks.
```

5 Results and Discussions

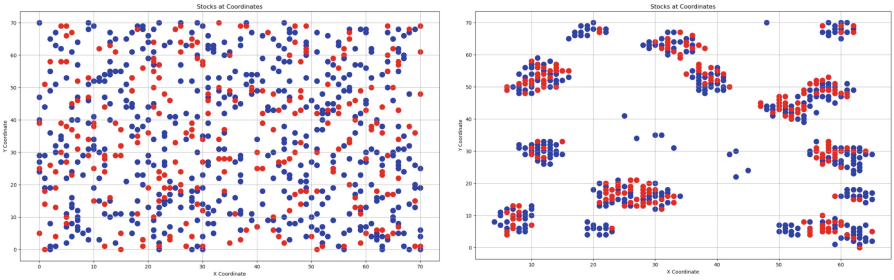
After parameter tuning and the successful termination of iterations, we capture the results of three random scenarios as shown in Fig. 3. The left sub-figures show the initial distribution of stocks and ants on the grid and the right sub-figures show the final results of the experiment. The blue scatters in the grid are stocks and the reds are ants. In Fig. 3, we provide results with 3 different random scenarios and it can be observed from the sub-figures that our experiments are successfully clustering a large number of stocks.

5.1 Heatmap and Cosine Similarity Results

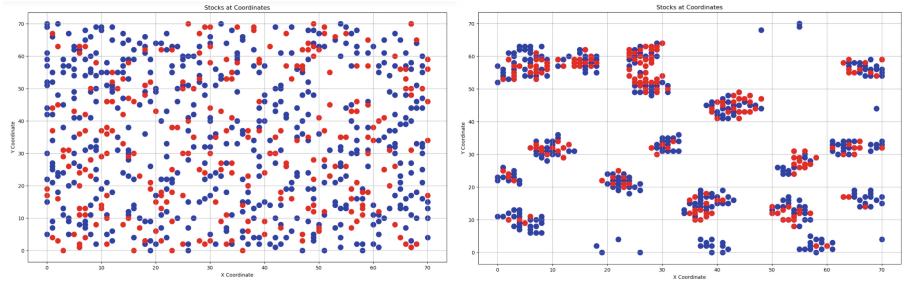
The next step is to analyze the clusters to validate the clustering results. We use the correlation of each pair of stocks within individual clusters for validation. Figure 4 shows the heatmap of clusters from the results. It can be observed that at least 85% of the pairs of stocks within each cluster have a positive correlation.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Fig. 3. Initial grid of ants and stocks (left) vs Final Clsuters obtained (right) in 3 random scenarios. Blue scatters represent stocks and red scatters represent ants. (Color figure online)

This explains that more than 85% of the stocks within each cluster have a similar magnitude. Since the clusters are made of time series, we use another validation measure known as cosine similarity. Cosine similarity is a measure of cosine of the angle between two vectors and is a measure of similarity in the directions of vectors [27]. Unlike Euclidian distance, cosine similarity is not highly sensitive to slight deformations such as seasonality in time series. The range of cosine similarity values typically falls between -1 and 1, where -1 suggests that the

two vectors are diametrically opposed whereas 1 indicates that the two vectors being compared are identical in direction.

The average cosine similarity of stock pairs within each cluster is more than 0.9. For the three scenarios in Figure 3(a)-(c) the actual values are: 0.9221, 0.9319 and 0.9319 respectively. This validates our experimental results that not only the magnitude but also the directions of stocks within each cluster are similar.

5.2 Silhouette Score

To conduct a comprehensive validation of the end results, we also computed the Silhouette score of clusters formed as an alternative method to validate our results from multiple vantage points. Silhouette score [22] is a measure of the quality of clusters that is based on the tightness and separation of clusters. Silhouette score is calculated for each data point (stock in our case) by calculating the similarity of the data point with other data points in the same cluster and the dissimilarity with data points in other clusters. One importance of using silhouette scores for cluster validation is that they rely solely on the actual arrangement of items in clusters and are not influenced by the clustering algorithm used. Equation (8) presents the calculation for the Silhouette score $s(i)$ for a stock i where $a(i)$ is the mean distance between i and all other stocks in the same cluster and $b(i)$ is the mean distance of i from all stocks in the nearest neighbor cluster. We used Euclidean distance between stocks' closing prices to calculate the distance metric. The score ranges from -1 to $+1$, where a score near to $+1$ signifies a strong match of an item with its own cluster and a poor match to the neighbor cluster whereas a score near -1 means that the item is a better match for the neighbor cluster. We calculated the silhouette score for each individual cluster by taking the average silhouette score of all the stocks in that cluster. Table 1 shows the silhouette score obtained for the clusters of all 3 random scenarios presented in this study. The average silhouette scores for all 3 scenarios obtained are 0.43, 0.30, and 0.36. In general, a silhouette score of 0.30 and above is considered relatively moderate-high and indicates that the stocks within the clusters are relatively similar and there is a decent separation between clusters. This shows a positive validation for clusters formed, in that the clustering is effective and that the clusters are distinct.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{8}$$

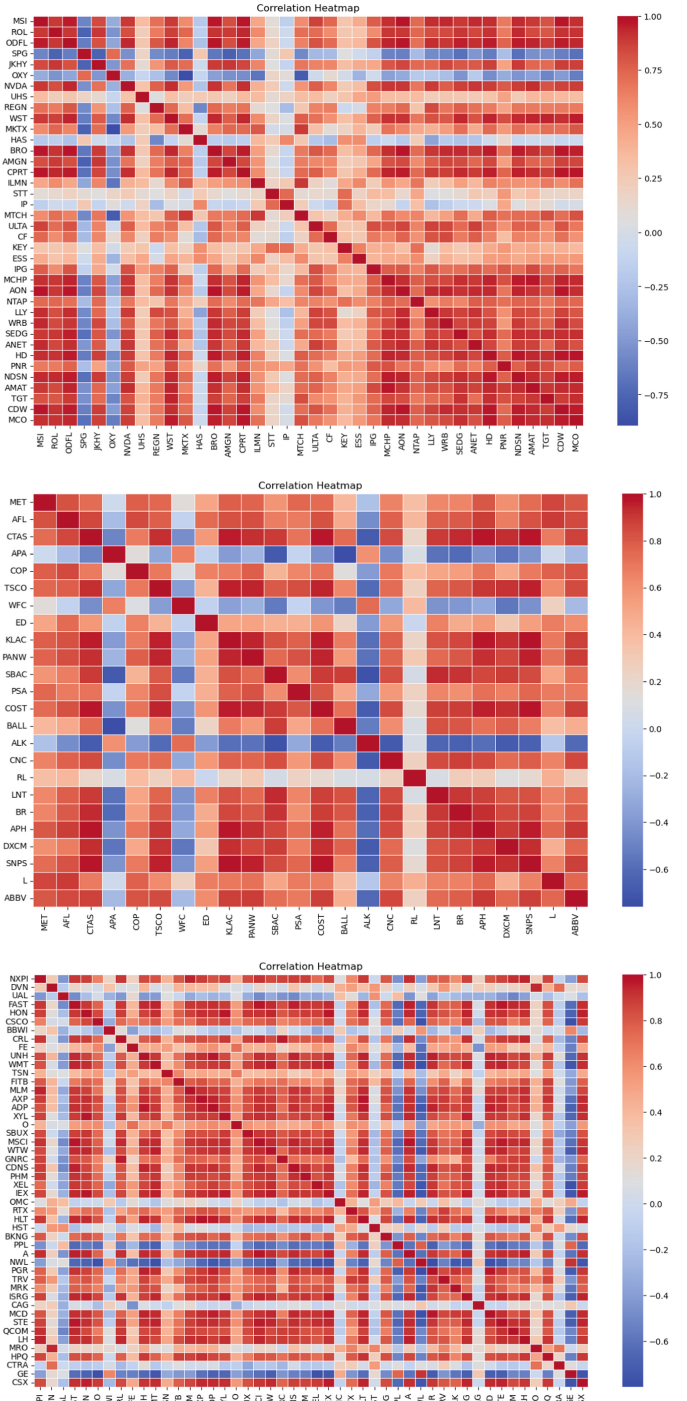


Fig. 4. Heatmap of Clusters

5.3 Inter-cluster Analysis

Table 2 presents an inter-cluster analysis of the results obtained in three (3) random scenarios. For all three (3) scenarios, we calculated the mean annualized returns of stocks within each cluster and the percentage of stock returns falling with one standard deviation of that mean. The analysis shows that the mean returns of different clusters in each scenario are significantly different from each other. For example, returns for clusters in scenario 1 range from 12% to 30%, from 10% to 30% in scenario 2, and from 9% to 23% in scenario 3. This validates that the clusters formed are different from each other in terms of annualized returns. It can also be observed that about 70%, on average, of the stocks' returns in each cluster fall within one standard deviation of the mean return of that cluster. In a normal distribution, about 68% of the data falls within one standard deviation of the mean, our result of 70% indicates that the standard deviation for stock returns in the clusters obtained are closely packed around the mean and show relatively little dispersion, thus asserting that the stocks within a cluster are close to each other.

5.4 Additional Discussion

To the best of our knowledge, the existing number of studies on the application of Ant brood sorting clustering for portfolio diversification for a direct comparison of results is limited, with no common metrics to compare this study. Oduntan et al. [21] clustered 30 stocks by running the experiment for 100,000 iterations. In their experiment, it was found that ants exhibited a tendency to allocate a significant portion of their time to random walks rather than effectively moving objects [16]. The number of actual iterations taken by our experiment prior to termination in all three (3) scenarios is less than 800, which explains that our experiment was efficient in clustering a large number of stocks.

Drawing a direct comparison of this heuristics-based study with deterministic clustering approaches such as K-Means, DBSCAN, etc., is not a sound approach due to their inherent differences. The data used by deterministic approaches generally contain multiple observations along with multiple features for each particular object, whereas we used just the time series of daily closing prices of stocks for this experiment so using the same data for deterministic methods will not be effective in generating and comparing results with deterministic methods. Deterministic methods aim to find accurate solutions, whereas Heuristics are typically used for complex problems where finding an optimal solution is computationally expensive or infeasible.

Table 1. Silhouette score for all 3 scenarios

	Scenario 1			Scenario 2			Scenario 3		
Cluster ID	(a)	(b)	**Score**	(a)	(b)	**Score**	(a)	(b)	**Score**
1	8470.66	5708.33	**−0.33**	1999.85	3376.69	**0.41**	2435.90	4079.50	**0.40**
2	3472.41	5708.33	**0.39**	9809.93	6521.73	**−0.34**	2300.42	3992.17	**0.42**
3	1908.44	5140.44	**0.63**	2399.20	3516.97	**0.32**	2127.45	3943.08	**0.46**
4	11195.34	7274.37	**−0.35**	1084.84	3074.51	**0.65**	2336.09	4082.30	**0.43**
5	2119.12	5164.97	**0.59**	1843.80	3348.83	**0.45**	2394.16	4092.61	**0.42**
6	1932.78	5103.87	**0.62**	2326.67	3472.49	**0.33**	1704.42	3814.09	**0.55**
7	2480.60	5374.99	**0.54**	2210.69	3449.42	**0.36**	5858.34	5329.69	**−0.09**
8	3184.92	5615.96	**0.43**	1593.67	3193.63	**0.50**	1681.57	3808.00	**0.56**
9	1872.99	5067.09	**0.63**	4726.10	3904.11	**−0.17**	2220.68	4020.99	**0.45**
10	1562.36	4960.98	**0.69**	2409.21	3511.90	**0.31**	5073.02	5329.69	**0.05**
11	3297.26	5597.41	**0.41**	2399.16	3510.56	**0.32**	2311.66	4022.25	**0.43**
12	1586.83	4955.55	**0.68**	3146.70	3904.11	**0.19**	3838.80	4700.80	**0.18**
13	2496.12	5374.92	**0.54**	1874.68	3293.18	**0.43**	974.01	3666.49	**0.73**
14	2374.82	5256.36	**0.55**	2292.58	3461.15	**0.34**	1438.64	3689.59	**0.61**
15	2807.78	5588.30	**0.50**	1727.58	3270.06	**0.47**	7620.32	6239.00	**−0.18**
Avg. Score	**0.43**			**0.30**			**0.36**		

Table 2. Inter-cluster analysis

	Scenario 1		Scenario 2		Scenario 3	
Cluster ID	Returns	% within 1 std	Returns	% within 1 std	Returns	% within 1 std
1	21.48	71.43	18.67	60.00	16.86	76.92
2	30.30	57.14	18.27	72.73	23.38	57.14
3	17.38	76.47	18.18	70.59	15.49	73.33
4	18.57	70.83	19.31	76.92	19.13	67.35
5	15.37	69.23	14.31	68.57	15.54	72.41
6	17.40	73.91	10.51	69.44	21.45	77.59
7	17.93	66.67	19.14	76.47	13.81	66.67
8	22.30	76.32	15.01	83.33	11.11	64.00
9	18.13	72.92	17.89	72.22	21.82	76.32
10	12.08	70.37	27.75	74.29	8.89	69.23
11	12.88	71.43	15.48	67.74	19.23	65.85
12	16.36	72.73	14.39	71.11	13.99	75.00
13	27.31	73.68	30.87	81.25	12.60	63.33
14	13.22	64.71	12.53	80.00	21.08	69.23
15	15.75	79.59	18.90	68.57	9.42	66.67

6 Conclusion

This study offers a unique contribution to the field of nature-inspired computation for large-scale portfolio diversification using Ant Brood Sorting clustering in conjunction with the co-integration of time series. The results demonstrate the algorithm's ability to effectively cluster stocks based on their similarity and the feasibility of using this method to create diversified portfolios from a large pool of stocks. This study represents a unique and valuable contribution to the field of portfolio diversification, offering a scalable approach to enhance risk management and potentially improve portfolio performance.

The correlation analysis demonstrates that over 85% of stock pairs within individual clusters exhibit positive correlations and an average cosine similarity of more than 0.9 further reinforces the consistency of stock behavior within clusters, thus validating the quality of the clusters formed and indicating that stocks within each cluster exhibited both similar magnitudes and directions. The Silhouette score analysis adds an additional layer of validation, affirming that the clusters are tightly packed and well-separated, with average scores exceeding 0.30. The inter-cluster analysis supports the validation of distinctiveness between each cluster by showcasing significant differences in mean annualized returns between clusters and more than 70% of stocks' returns in each cluster falling within one standard deviation of the cluster mean. This further confirms that the stocks within each cluster share similar financial performance characteristics and that the clusters are distinct from each other.

This approach holds promise for investors seeking to bring more robust and resilient diversification within their portfolios in diverse market conditions. For future studies, we intend to use the stocks from clusters formed in this study to create diversified portfolios and optimize the weights of the stocks using another nature-inspired algorithm called Particle Swarm Optimization. Further research in this study may also explore incorporating the seasonality factor of time series to update the clusters accordingly.

Acknowledgement. The first author acknowledges financial support from Professor Thulasiram and Graduate Enhancement of Tri-agency Stipends (GETS), University of Manitoba. The last two authors acknowledge the Discovery Grants from the Natural Sciences and Engineering Research Council (NSERC) Canada.

References

1. Arslan, H., Uğurlu, O., Eliiyi, D.T.: An overview of new generation bio-inspired algorithms for portfolio optimization, pp. 207–224. Springer Nature Singapore, Singapore (2022). https://doi.org/10.1007/978-981-16-8997-0_12
2. Boryczka, U.: Ant clustering algorithm. Intelligent Information Systems 1998 (01 2008)
3. Bui, Q., Ślepaczuk, R.: Applying hurst exponent in pair trading strategies on nasdaq 100 index. Phys. A: Stat. Mech. Appl. **592**, 126784 (2022). https://doi.org/10.1016/j.physa.2021.126784

4. Chen, Y., Zhao, X., Yuan, J.: Swarm intelligence algorithms for portfolio optimization problems: Overview and recent advances. Mobile Information Systems 2022 (07 2022). https://doi.org/10.1155/2022/4241049
5. Colomine Durán, F., Cotta, C., Fernández-Leiva, A.J.: Epoch-based application of problem-aware operators in a multiobjective memetic algorithm for portfolio optimization. In: Correia, J., Smith, S., Qaddoura, R. (eds.) Applications of Evolutionary Computation, pp. 210–222. Springer Nature Switzerland, Cham (2023)
6. Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chrétien, L.: The dynamics of collective sorting robot-like ants and ant-like robots. In: From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior, pp. 356–365 (1991)
7. Engle, R.F., Granger, C.W.J.: Co-integration and error correction: representation, estimation, and testing. Econometrica **55**(2), 251–276 (1987). http://www.jstor.org/stable/1913236
8. Freitas, F.D., De Souza, A.F., de Almeida, A.R.: Prediction-based portfolio optimization model using neural networks. Neurocomputing **72**(10), 2155–2170 (2009). https://doi.org/10.1016/j.neucom.2008.08.019 lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Designn (ISDA 2007)
9. Hasan, F., Ahmad, F., Shahid, M., Khan, A., Ahmad, G.: Solving portfolio selection problem using whale optimization algorithm. In: 2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM), pp. 1–5 (2022). https://doi.org/10.1109/ICCAKM54721.2022.9990079
10. Huang, C.F.: A hybrid stock selection model using genetic algorithms and support vector regression. Appl. Soft Comput. **12**(2), 807–818 (2012). https://doi.org/10.1016/j.asoc.2011.10.009
11. Kalayci, C.B., Ertenlice, O., Akbay, M.A.: A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. Expert Syst. Appl. **125**, 345–368 (2019). https://doi.org/10.1016/j.eswa.2019.02.011
12. Koumou, G.B.: Diversification and portfolio theory: a review. Fin. Markets. Portfolio Mgmt. **34**(3), 267–312 (2020)
13. Krauss, C.: Statistical arbitrage pairs trading strategies: review and outlook. J. Econ. Surv. **31**(2), 513–545 (2017). https://doi.org/10.1111/joes.12153
14. Liang, S., Lu, S., Lin, J., Wang, Z.: Hardware accelerator for engle-granger cointegration in pairs trading. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2020). https://doi.org/10.1109/ISCAS45731.2020.9180586
15. Liu, M., Luo, K., Zhang, J., Chen, S.: A stock selection algorithm hybridizing grey wolf optimizer and support vector regression. Expert Syst. Appl. **179**, 115078 (2021). https://doi.org/10.1016/j.eswa.2021.115078
16. Liu, Y.Y., Thulasiraman, P., Thulasiram, R.K.: Parallelizing active memory ants with mapreduce for clustering financial time series data. In: 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), pp. 137–144 (2016). https://doi.org/10.1109/BDCloud-SocialCom-SustainCom.2016.31
17. Lumer, E.D., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3: from animals to animats 3, pp. 501–508 (1994)

18. Mazumdar, K., Zhang, D., Guo, Y.: Portfolio selection and unsystematic risk optimisation using swarm intelligence. J. Bank. Financial Technol. 4 (01 2020). https://doi.org/10.1007/s42786-019-00013-x
19. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to time series analysis and forecasting. John Wiley & Sons (2015)
20. Oduntan, O.I., Thulasiraman, P.: Hybrid metaheuristic algorithm for clustering. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–9 (2018). https://doi.org/10.1109/SSCI.2018.8628863
21. Oduntan, O.I., Thulasiraman, P., Thulasiram, R.: Portfolio diversification using ant brood sorting clustering, pp. 256–261 (2014). https://doi.org/10.1109/NaBIC.2014.6921888
22. Rousseeuw, P.: Rousseeuw, p.j.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (11 1987). https://doi.org/10.1016/0377-0427(87)90125-7
23. Seabold, S., Perktold, J.: statsmodels: Econometric and statistical modeling with python. In: 9th Python in Science Conference (2010)
24. Sen, J.: Designing efficient pair-trading strategies using cointegration for the indian stock market. In: 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), pp. 1–9 (2022). https://doi.org/10.1109/ASIANCON55314.2022.9909455
25. Shahid, M., Ashraf, Z., Shamim, M., Ansari, M.S.: A novel portfolio selection strategy using gradient-based optimizer. In: Saraswat, M., Roy, S., Chowdhury, C., Gandomi, A.H. (eds.) Proceedings of International Conference on Data Science and Applications: ICDSA 2021, Volume 2, pp. 287–297. Springer Singapore, Singapore (2022). https://doi.org/10.1007/978-981-16-5348-3_23
26. Tingjin Yan, M.C.C., Wong, H.Y.: Pairs trading under delayed cointegration. Quant. Finance **22**(9), 1627–1648 (2022). https://doi.org/10.1080/14697688.2022.2064760
27. Xia, P., Zhang, L., Li, F.: Learning similarity with cosine similarity ensemble. Inform. Sci. **307**, 39–52 (2015). https://doi.org/10.1016/j.ins.2015.02.024