



Article

# Bio-Inspired Machine Learning for Distributed Confidential Multi-Portfolio Selection Problem

Ameer Tamoor Khan <sup>1,†</sup>, Xinwei Cao <sup>2,\*</sup>, Bolin Liao <sup>3,†</sup> and Adam Francis <sup>4,†</sup><sup>1</sup> Department of Computing, The Hong Kong Polytechnic University, Hong Kong 999077, China<sup>2</sup> School of Business, Jiangnan University, Wuxi 213031, China<sup>3</sup> College of Information Science and Engineering, Jishou University, Jishou 409811, China<sup>4</sup> Faculty of Science and Engineering, Swansea University, Swansea SA1 8EN, UK

\* Correspondence: xcao.fudan@gmail.com

† These authors contributed equally to this work.

**Abstract:** The recently emerging multi-portfolio selection problem lacks a proper framework to ensure that client privacy and database secrecy remain intact. Since privacy is of major concern these days, in this paper, we propose a variant of Beetle Antennae Search (BAS) known as Distributed Beetle Antennae Search (DBAS) to optimize multi-portfolio selection problems without violating the privacy of individual portfolios. DBAS is a swarm-based optimization algorithm that solely shares the gradients of portfolios among the swarm without sharing private data or portfolio stock information. DBAS is a hybrid framework, and it inherits the swarm-like nature of the Particle Swarm Optimization (PSO) algorithm with the BAS updating criteria. It ensures a robust and fast optimization of the multi-portfolio selection problem whilst keeping the privacy and secrecy of each portfolio intact. Since multi-portfolio selection problems are a recent direction for the field, no work has been done concerning the privacy of the database nor the privacy of stock information of individual portfolios. To test the robustness of DBAS, simulations were conducted consisting of *four* categories of multi-portfolio problems, where in each category, *three* portfolios were selected. To achieve this, 200 days worth of real-world stock data were utilized from 25 NASDAQ stock companies. The simulation results prove that DBAS not only ensures portfolio privacy but is also efficient and robust in selecting optimal portfolios.

**Keywords:** multi-portfolio; optimization; swarm algorithm; beetle antennae search; stochastic algorithm; distributed beetle antennae search; investment; stocks



**Citation:** Khan, A.T.; Cao, X.; Liao, B.; Francis, A. Bio-Inspired Machine Learning for Distributed Confidential Multi-Portfolio Selection Problem. *Biomimetics* **2022**, *7*, 124.

<https://doi.org/10.3390/biomimetics7030124>

Academic Editors: Stanislav N. Gorb, Giuseppe Carbone, Thomas Speck and Andreas Taubert

Received: 31 July 2022

Accepted: 19 August 2022

Published: 29 August 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

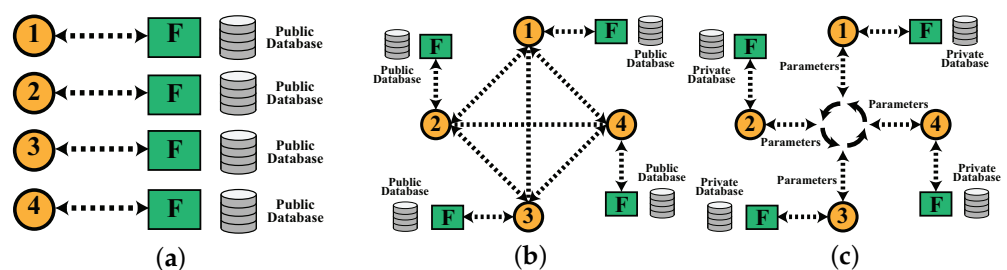
## 1. Introduction

Portfolio optimization is a hot research topic in academia, since it enables investors to make an optimal decision between profit and risk. This refers to the method of making the best investment in numerous stocks [1]. Apart from this, there are several other real-world constraints that researchers have taken care of over time, for instance, cardinality constraint, tax-aware constraint, lower and upper bounds, multiple portfolios, round-lot constraint, stock size, computational and time complexities.

There are several state-of-the-art algorithms that have been proposed over time to tackle these constraints. For instance, an unconstrained portfolio optimization problem can be easily solved using linear or quadratic programming, but constraints make it a complex optimization problem to solve. Ref. [2] showed that the simple addition of a cardinality constraint makes it an NP-complete problem. Therefore, there is no straightforward algorithm to compute the exact optimality of the problem. Techniques that use machine learning algorithms, heuristic algorithms, and black-box optimization are good alternatives to solve such highly complex, computationally and time-consuming problems. Ref. [3] proposed using multiple machine learning modules, i.e., coordinate descent, the proximal gradient, Dykstra, and alternating direction of multipliers. Their algorithm also accounts for

several constrained portfolios, i.e., equal risk contribution, diversified, and risk budgeting portfolios. Likewise, Ref. [4] combined several portfolio models, i.e., equal-weighted modeling (EQ), mean-variance model, and Monte Carlo simulation modeling. Furthermore, to improve the portfolio problem based on its time-series nature, they applied a long short-term memory (LSTM) model. However, the major drawback of these methods is the training of hundreds of hyper-parameters, which makes them computationally expensive and slow to compute.

In this paper, we are focused on the multi-portfolio selection problem while considering the investors' privacy. Since the multi-portfolio problem has only recently gained attention from academia and financial practitioners, no work has been conducted regarding the confidentiality and the secrecy of private databases and individual portfolios. The multi-portfolio selection problem enables the optimization of tens of hundreds of portfolios simultaneously, making it computationally economical and time efficient. There are several proposed methods for multi-portfolio optimization in the literature [5–17]. The authors of [18,19] were the pioneers of the multi-portfolio optimization domain. The objective was to maximize social welfare, which was the sum of the utility of the individual accounts. For instance, ref. [20] proposed a hybrid model that includes a heuristic and combinatorial framework and solved the multi-portfolio problem under a risk–budget constraint. Ref. [21] proposed a Cournot–Nash equilibrium framework to solve the multi-portfolio problem. The major drawback being that each portfolio was treated individually, assuming that the others gave the best response. Ref. [22] overcame this issue by proposing a joint optimization problem, where the model was able to optimize the portfolio as well as cost splitting among the portfolios. Ref. [23] employed an information pooling game mechanism for the multi-portfolio optimization. Ref. [24] added a risk measurement constraint along with the selection of multi-portfolios. Ref. [25] proposed fairness-aware multi-participant satisfying (FMS) criterion to model a target-oriented strategy which optimized client portfolios by maximizing the returns. The major drawback of these techniques is the lack of privacy for the investors. They also solve the multi-portfolio problem, collectively making asset data exposed, meaning data are not secure or confidential. In order to provide privacy to each investor, i.e., portfolios regarding their investment and portfolio selection, it is necessary to design a distributed framework that optimizes each portfolio locally and optimizes all portfolios collectively. The three types of optimizing models are shown in Figure 1. It shows that in local learning, all portfolios are optimized individually, which is computationally and time-wise inefficient. Likewise, typical swarm learning models do not account for the confidentiality of the data and portfolio, and the particles share critical information about the portfolios to reach the optimal solution. However, in the distributed system, each particle deals with a single portfolio locally, and the particles share the objective function value, i.e., gradients alone to each other. Thus, they efficiently optimize the multi-portfolio selection problem without violating privacy.



**Figure 1.** (a) shows the local learning where each BAS algorithm is assigned to one portfolio, and the objective is to optimize the respective portfolio. (b) shows the swarm architecture, where the swarm has access to a public database, where particles coordinate to obtain an optimal solution without considering the privacy of the database. (c) shows the distributed architecture, where particles only share the parameters, i.e., gradients, instead of the private information of the portfolios, i.e., private database, stock information, and client information.

In this paper, we propose a swarm variant of a known meta-heuristic algorithm known as Beetle Antennae Search (BAS). It mimics the food-collecting nature of the beetle in order to search for the optimal solution to a problem. BAS is a single particle searching algorithm, where the particle optimizes an objective function by searching the search space iteratively. The utility of BAS has expanded to several real-world problems [26–49], including the portfolio optimization. Ref. [50] employed BAS for the selection of the optimal portfolio under a non-convex cardinality constraint. Ref. [51] proposed a QBAS (Quantum Beetle Antennae Search) variant of BAS and solved the portfolio selection problem under cardinality constraint. Ref. [52] solved the time-varying portfolio selection problem under the transaction cost. Ref. [53] used the mean-variance model of portfolio optimization under two constraints: cardinality and transaction cost constraints. Ref. [54] proposed a hybrid framework of BAS-PSO and used it for portfolio optimization. From these research works, we can see the utility of BAS in portfolio optimization under real-world constraints. However, BAS has never been employed for the multi-portfolio optimization problem because of computational limitations. As mentioned earlier, BAS is a single particle searching algorithm, meaning it would be computationally challenging for BAS to optimize even a single portfolio with over 100 stock companies. Therefore, the efficiency of BAS will drop further if applied to the multi-portfolio selection problem.

We have proposed a swarm or distributed variant of BAS known as DBAS (Distributed Beetle Antennae Search). It is a hybrid variant with the swarm-like nature of Particle Swarm Optimization (PSO) and the BAS updating criteria. Each particle in DBAS will optimize a single portfolio, and collectively, the swarm will optimize all the portfolios without violating portfolio privacy. The DBAS will optimize the multi-portfolio in two stages by optimizing each portfolio locally and then optimizing all the portfolios globally without sharing any private data among portfolios. To the best of our knowledge, no researcher has considered the privacy issue while solving the multi-portfolio problem, since it is a newly emerging portfolio selection challenge. Through our proposed framework, we will optimize the multi-portfolio problem efficiently with low computation and time cost whilst ensuring the privacy of clients and their portfolios.

The rest of the paper is structured as follows: In Section 2, we will formulate the portfolio optimization problem. In Section 3, we will elaborate on the nature of BAS and will drive the DBAS variant. In Section 4, we will discuss the simulation results on a multi-portfolio selection problem with real-world stock data. In Section 5, we will conclude the paper with final remarks.

## 2. Problem Formulation

In this section, we will discuss the building blocks of the portfolio optimization problem. Later, we will elaborate on different portfolio models and select one for the DBAS algorithm. All the hyper-parameters are mentioned in Table 1.

**Table 1.** Hyper-Parameters.

Name	Notation	Values (Default)
Antennae Length	$d$	0.9
Step Size	$\tau$	0.9
Personal Antennae Length	$d_1$	0.9
Global Antennae Length	$d_2$	0.9
Step Multiplier	$\eta$	0.99

### 2.1. Building Blocks of Portfolio Selection

Here, we will discuss the building blocks of portfolio selection, i.e., expected return (profit), risk, total investment, total transaction cost, and cardinality constraint.

### 2.1.1. Expected Return

The primary factor in portfolio optimization is the maximization of expected return: in other words, the profit that investors are expecting from their investment in stocks. Let us say there are  $K$  total stocks and the normalized investment in all  $K$  stocks is given as  $E = [E_1, E_2, E_3, \dots, E_i, \dots, E_K] \in \mathbb{R}^K$ , where  $i$  is the  $i$ -th stock and  $E_i$  is the the normalized investment in the  $i$ -th stock. Now, let us say that the mean return of  $K$  stocks based on their past data is given as,  $\epsilon = [\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_i, \dots, \epsilon_K]^T \in \mathbb{R}^K$ . The objective is to find the optimal  $E$  such that it maximizes the expected return, which is given as

$$\max_E \zeta(E) = E\epsilon^T. \tag{1}$$

To further consolidate (1) and to ensure that the expected return is above a certain threshold  $k$ , the modification to (1) is given as,

$$\max_E \zeta(E) = E\epsilon^T - k \geq 0. \tag{2}$$

here,  $k$  is a scalar quantity and  $E\epsilon^T - k = 0$  means the break-even point (no profit-loss).

### 2.1.2. Risk

The maximization of profit (expected return) does not avoid the risk factor. The second objective is to minimize the risk. The risk minimization is given as,

$$\min_E v(E) = EQE^T, \tag{3}$$

where  $Q$  is a covariance matrix, and it is given as,

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & \dots & Q_{1j} \\ Q_{21} & Q_{22} & Q_{23} & \dots & Q_{2j} \\ Q_{31} & Q_{32} & Q_{33} & \dots & Q_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{i1} & Q_{i2} & Q_{i3} & \dots & Q_{ij} \end{bmatrix}, \tag{4}$$

where  $Q_{ij}$  is the variance of the  $i$ -th stock with  $j$ -th stock in the portfolio.

### 2.1.3. Total Investment

Now, we will discuss one of the primary constraints: total investment. It ensures the allocation of the total asset in the stocks, and it is given as,

$$\mathbf{1}E^T = 1, \tag{5}$$

where  $\mathbf{1}$  is a unit vector. As mentioned above,  $E^T$  is a normalized investment, so the sum of the amount invested in the portfolio should be 1 as shown in (5). To ensure this, we need an additional constraint which is given as,

$$\mathbf{0} \leq E \leq \mathbf{1}. \tag{6}$$

It will ensure that  $E$  remains positive and within  $[0, 1]$ .

### 2.1.4. Total Transaction Cost

The total transaction cost is another essential constraint in portfolio selection. The transaction cost is positive if the investor buys stock and is negative if the investor sells stock. Let us say that  $\Phi(E)$  represents the transaction cost of the portfolio; it is given as,

$$\Phi(E) = [\Phi_1(E_1), \Phi_2(E_2), \Phi_3(E_3), \dots, \Phi_i(E_i)], \tag{7}$$

where  $i \in \{1, 2, 3, \dots, K\}$  and  $\Phi_i(E_i)$  represent the transaction cost of the  $i$ -th stock. The total transaction cost refers to the sum of all the individual stocks, which is given as

$$\Phi_t(E) = \sum_{i=1}^K \Phi_i(E_i). \tag{8}$$

If  $\Phi_t(E) > 0$ , it means more stock is bought than sold. Likewise, if  $\Phi_t(E) < 0$ , it means that more stock is sold than bought. Generally, the transaction cost includes some additional charges, which we can formulate using a linear transaction model [55], which includes a linear function  $\alpha = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_i]$ , where  $i \in \{1, 2, 3, \dots, K\}$ . The transaction cost of the  $i$ -th stock in the portfolio is given as,

$$\Phi_i(E_i) = \alpha_i E_i. \tag{9}$$

We can rewrite (7) using (9) as,

$$\Phi(E) = \alpha E^T. \tag{10}$$

We can combine (5) and (10) into a single constraint, which is given as,

$$\chi(E) = (\mathbf{1} + \alpha)E^T = 1. \tag{11}$$

### 2.1.5. Cardinality Constraint

Another real-world constraint in portfolio selection is known as the cardinality constraint. So far, we have assumed that investment is made in all of the  $K$  stocks. However, in a real case scenario, it is observed that the investor may want to exclude some stocks  $n (< K)$  from the final portfolio selection, which is known as a cardinality constraint. We can make such a decision by introducing a binary decision vector, i.e.,  $\mathbf{b} = [b_1, b_2, b_3, \dots, b_i, \dots, b_n]$ , where  $b_i \in \{0, 1\}$ . According to the cardinality constraint,

$$\sum_{i=1}^K b_i = n. \tag{12}$$

We also need to reformulate (6) according to (12), which is given as,

$$\mathbf{0} \leq E \leq \mathbf{b}. \tag{13}$$

## 2.2. Portfolio Selection Models

Here, we will discuss the three known models for portfolio selection: mean variance model, efficient frontier model, and Sharpe ratio model.

### 2.2.1. Mean-Variance Model

The primary objective of the mean variance model is to minimize the risk involved in portfolio selection, and it treats the expected return (profit) as a constraint. We can formulate the mean variance model using (2), (3), and (11)–(13), which is given as,

Minimize:

$$\min_E v(E) \quad (14)$$

Subject to:

$$\zeta(E) = Ee^T - k \geq 0 \quad (15)$$

$$\chi(E) = (\mathbf{1} + \alpha)E^T = 1 \quad (16)$$

$$\sum_{i=1}^K b_i = n \quad (17)$$

$$\mathbf{0} \leq E \leq \mathbf{b}. \quad (18)$$

The mean variance model is not very flexible, since it restrains the growth in profits. To overcome it, we can make use of an efficient frontier model.

### 2.2.2. Efficient Frontier Model

In this model, we include expected return (profit) in the objective function. We can formulate the frontier model using (2), (3), and (11)–(13), which is given as,

Minimize:

$$\min_E \lambda v(E) - (1 - \lambda)\zeta(E) \quad (19)$$

Subject to:

$$\chi(E) = (\mathbf{1} + \alpha)E^T = 1 \quad (20)$$

$$\sum_{i=1}^K b_i = n \quad (21)$$

$$\mathbf{0} \leq E \leq \mathbf{b}. \quad (22)$$

here,  $\lambda \in [0, 1]$ . If  $\lambda = 0$ , then the model will maximize the expected-return (profit). However, if  $\lambda = 1$ , then the model will minimize the risk. The value of  $\lambda$  between  $[0, 1]$  will be a trade-off between profit and the risk. The only drawback of the model is that it involves the use of a hyper-parameter  $\lambda$ . To avoid this, we will look into a Sharpe model, which we will optimize using DBAS.

### 2.2.3. Sharpe Ratio Model

The Sharpe ratio model is another portfolio selection model which includes the ratio of profit  $\zeta(E)$  and risk  $v(E)$ . The objective is to maximize the ratio, which is given as,

$$SR = \frac{\zeta(E)}{v(E)}. \quad (23)$$

We will inverse the formulation (23) to form a minimization problem. The Sharpe ratio model will help to avoid the additional hyper-parameter  $\lambda$ . Now, the objective is to

minimize  $\frac{v(E)}{\xi(E)}$ , which will in turn maximize the profit and minimize the risk. Again, we can formulate the model using (2), (3), and (11)–(13), which is given as,

Minimization:

$$\min_E \frac{v(E)}{\xi(E)} \tag{24}$$

Subject to:

$$\chi(E) = (\mathbf{1} + \alpha)E^T = 1 \tag{25}$$

$$\sum_{i=1}^K b_i = n \tag{26}$$

$$\mathbf{0} \leq E \leq \mathbf{b}. \tag{27}$$

This model will be used for the portfolio optimization. In the next section, we will formulate the Distributed Beetle Antennae Search (DBAS), which we will be used to solve the portfolio problem given in (24)–(27).

### 3. Distributed Beetle Antennae Search (DBAS)

In this section, we will elaborate on the nature of BAS, its formulation, and the algorithm. Later, we will develop DBAS and will discuss its underlying algorithm.

#### 3.1. BAS Formulation and Algorithm

BAS has a biologically inspired meta-heuristic algorithm, which is inspired by the food-searching behavior of the beetle. It has two antennas, i.e., left  $X_l$  and right  $X_r$ . It registers the smell of the food on both antennae, and based on the intensity of the smell, it either moves in the left direction or right. The beetle repeats this procedure iteratively until it finally reaches the food source. Details on the behavior of beetles and BAS are presented in [56]. Here, we will look into updating the equations of the BAS algorithm, which are responsible for converging the searching particle toward the optimal solution.

Imagine a non-linear objective function  $F(\mathbf{X})$ , where  $\mathbf{X}$  is a vector of optimizing variables. We can represent an objective function as,

$$\min_{\mathbf{X}} F(\mathbf{X}). \tag{28}$$

Here, we are solely considering an objective function, while the real-world problem would include additional constraints, as shown in (24)–(27). The objective of the BAS is to find the optimal solution of (28) by tweaking the values of  $\mathbf{X}$  such that the maximum value of (28) is achieved. Here, it is worth mentioning that we can assume the maximization problem as well. The formulation will be a little different, which will be shown later during the derivation.

From this point forward, we will refer to the beetle as a particle. Let us say that the current position of the particle is  $\mathbf{X} \in \mathbb{R}^{1 \times K}$ . To compute the new position of the particle, we will move it slightly to the right  $\mathbf{X}_r$ , and slightly to the left  $\mathbf{X}_l$ , which is given as

$$\mathbf{X}_r = \mathbf{X} + d\mathbf{b} \tag{29}$$

$$\mathbf{X}_l = \mathbf{X} - d\mathbf{b}, \tag{30}$$

where  $\mathbf{b} \in \mathbb{R}^{1 \times K}$  is similar to  $\mathbf{X}$ . Here,  $d$  is an antenna length, which determines how big or small a step should be taken in each iteration. In the beginning,  $d$  will have a larger value allowing the particle to explore more of the search space, and over time when the particle approaches the optimal solution, the value of  $d$  becomes small.

The next step is to compute the objective function value at  $X_r$  and  $X_l$ , so that BAS can decide its course, i.e., either in the  $X_r$  direction or  $X_l$  direction. The computation of  $F(X)$  is given as,

$$F_r = F(X_r) \quad (31)$$

$$F_l = F(X_l), \quad (32)$$

where  $F_r$  and  $F_l$  are scalar values of the objective function at  $X_r$  and  $X_l$  respectively. Now, BAS will compute the next step based on the difference of the values of  $F_r$  and  $F_l$ , which is given as,

$$\Delta F = F_r - F_l \quad (33)$$

$$X = X - \tau \text{sgn}(\Delta F) \mathbf{b}, \quad (34)$$

where  $\text{sgn}$  is an activation function to translate the value of  $\Delta F$  between  $[-1, 1]$ . Likewise,  $\tau$  is another hyper-parameter, which is used to decide the step length of the BAS particle. This concludes the general framework of BAS. The algorithm is shown in Algorithm 1.

---

**Algorithm 1** BAS Algorithm.

---

**Objective Function:**

$$F(X, O)$$

**DBAS Initialization:**

$$\eta = 0.99$$

$$d = 0.9$$

$$\tau = 0.9$$

**Portfolio Initialization:**

$$T = \text{iter}$$

$$D = \text{dim}$$

$$X = \text{randn}(1, D)$$

$$O = [M_n, C_n] \quad M_n : \text{Mean} \quad C_n : \text{Covariance}$$

$$F_{best} = F(X, P)$$

$$X_{best} = X$$

**For**  $i = 1 : T$

$$\mathbf{b} = \text{randn}(1, D)$$

Compute  $X_r$  and  $X_l$  using (29) and (30)

Compute  $F_r$  and  $F_l$  using (31) and (32)

Compute  $X$  using (34)

$$F = F(X, O)$$

**If**  $F_{min} < F_{best}$  :

$$F_{best} = F_{max}$$

$$X_{best} = X$$

**Else:**

$$X = X_{best}$$

**End If**

$$d = \eta d + 0.01$$

$$\tau = \eta \tau$$

**End For**

---

This algorithm is designed for the portfolio selection problem. The objective function includes  $F(X, O)$ , where  $O$  includes all the parameters necessary for the portfolio problem, i.e.,  $M$  means of the stocks,  $C$  covariance matrix of the stocks, and other constraints. The single-particle BAS algorithm is limited to single portfolio optimization because of its computational limitations. In order to optimize multiple portfolios at once, we need to



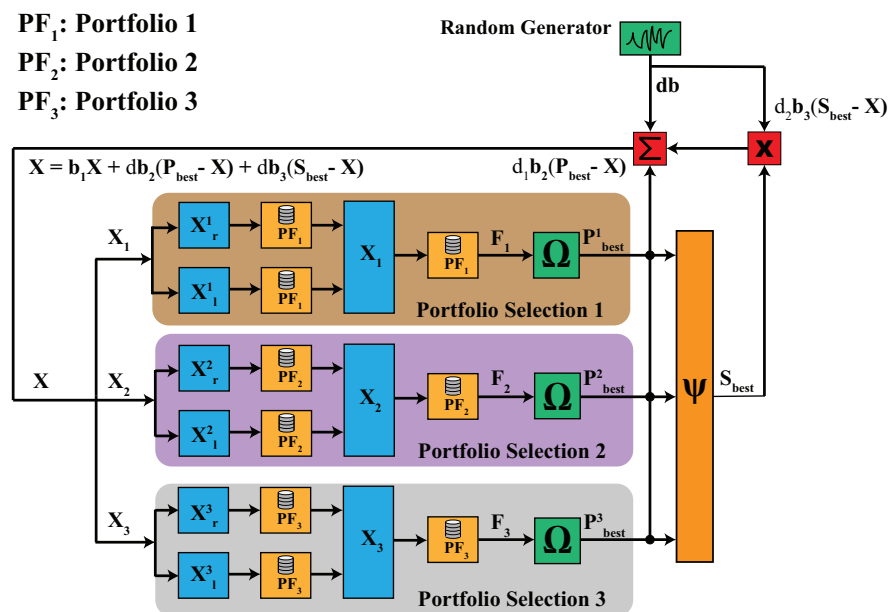
introduce a more robust nature in the already existing framework, which we will introduce in DBAS formulation.

**Remark 1.** The above formulation of BAS is for the minimization problem. For the maximization problem, we need to modify the (34), as shown below,

$$X = X + \tau \text{sgn}(\Delta F)b, \tag{35}$$

where the  $-$  sign is replaced with the  $+$  sign. It will help BAS to move up the valley in the search space.

In the next section, we will formulate the DBAS algorithm. Unlike BAS, the DBAS formulation will be better focused on the portfolio problem. However, it is worth mentioning that the algorithm is not problem-specific, as an objective function must be defined for any real-world problem. The schematic of the proposed algorithm is shown in Figure 2.



**Figure 2.** The schematic of the DBAS algorithm for the multi-portfolio selection problem. It consists of 3 portfolios and clearly shows that each particle  $[X_1, X_2, X_3]$  optimizes portfolio  $PF_1, PF_2,$  and  $PF_3,$  respectively. It can be seen that particles do not share any portfolio database or stock information among the swarm, which ensures privacy.

### 3.2. DBAS Formulation and Algorithm

DBAS is a hybrid variant of BAS and PSO (Particle Swarm Optimization) algorithms. It mimics the swarm-like nature of PSO, where each step of the particles is based on the particle's best position  $P_{best}$  and the swarm's best position  $S_{best}$ . However, it uses the BAS updating strategy given in (34).

#### 3.2.1. DBAS Updating Criteria

We will modify the updating criteria for  $X_r$  and  $X_l$  given in (29) and (30). The updating criteria depend on three factors: the particle's best position  $P_{best}$ , the swarm's best solution  $S_{best}$ , and the random direction  $db$ . The modified updating criteria is given as,

$$V = b_1 V + d_1 b_2 (P_{best} - X) + d_2 b_3 (S_{best} - X) \tag{36}$$

$$X_r = X + db + V \tag{37}$$

$$X_l = X - db - V, \tag{38}$$

where  $\mathbf{b} \in \mathbb{R}^{D \times K}$  and  $[b_1, b_2, b_3] \in \mathbb{R}$ . Here,  $\mathbf{X} \in \mathbb{R}^{D \times K}$  represents the particles positions,  $\mathbf{P}_{best} \in \mathbb{R}^{D \times K}$  shows the particles best positions, and  $\mathbf{S}_{best} \in \mathbb{R}^{D \times K}$  is the swarm's best position, which are stacked in a row to meet the dimensionality requirement. Likewise,  $[d, d_1, d_2]$  are scalars. Typical swarm algorithms are used to compute a single objective function, but our objective is to optimize multiple objective functions since each portfolio comprises different stocks. Therefore, it is worth mentioning that  $\mathbf{X} \in \mathbb{R}^{D \times K}$ , where  $D$  and  $K$  represent the number of portfolios and portfolio size, respectively. Likewise,  $\{\mathbf{b}_1, \mathbf{b}_2\} \in \mathbb{R}^{D \times K}$  are two random vectors and  $d$  represents the antenna size. Each particle in DBAS corresponds to a single portfolio. The objective is that each particle will optimize its corresponding portfolio locally, i.e.,  $\mathbf{P}_{best}^j \in \mathbb{R}^{1 \times K}$ , where  $j \in \{1, 2, 3, \dots, D\}$ , and the swarm will optimize all portfolios, i.e.,  $\mathbf{S}_{best}$ , collectively.

After computing  $\mathbf{X}_r$  and  $\mathbf{X}_l$ , the next step is to compute the objective function value for each portfolio, i.e., each particle. It is given as,

$$F_r = F(\mathbf{X}_r, \mathbf{O}) \tag{39}$$

$$F_l = F(\mathbf{X}_l, \mathbf{O}), \tag{40}$$

where  $\mathbf{O}$  includes all the parameters necessary for portfolio optimization as mentioned in the problem formulation section. Here,  $\{F_r, F_l\} \in \mathbb{R}^{D \times 1}$  and  $F(\cdot)$  is a vector of dimension  $D$ ; it includes the objective functions of all portfolios. Next, each position of particle  $\mathbf{X}$  is updated using,

$$\Delta F = F_r - F_l \tag{41}$$

$$\mathbf{X} = \mathbf{X} - \tau \text{sgn}(\Delta F) \mathbf{b}. \tag{42}$$

Then, the objective function value of each portfolio is computed, i.e., the particle at  $\mathbf{X}$ , which is given as,

$$F = F(\mathbf{X}, \mathbf{O}), \tag{43}$$

where  $F$  has similar dimensions as  $F_r$  and  $F_l$ . If the objective function value  $F_j(\cdot)$  of a  $j$ -th particle at  $\mathbf{X}_j \in \mathbb{R}^{1 \times K}$  is less than its personal best  $\mathbf{P}_{best}^j$ , then update  $\mathbf{P}_{best}^j$  with  $\mathbf{X}_j$ . Otherwise, retain the old best values. This updating rule for all the particles  $\mathbf{X}$  is given as,

$$\mathbf{P}_{best} = \Omega(\mathbf{X}) = \begin{cases} \mathbf{X}, & \text{if } F(\mathbf{X}) < \mathbf{P}_{best} \\ \mathbf{P}_{best}, & \text{otherwise.} \end{cases} \tag{44}$$

Now, the next step is to obtain the swarm's best position. This can be done by simply looking for the minimum value of the objective function in  $F(\cdot)$ , which is given as,

$$[F, j] = \min(F(\mathbf{X}, \mathbf{O})). \tag{45}$$

If the value of  $F$  is less than the swarm's best solution  $G_{best}$ , then update  $\mathbf{S}_{best}$ ; otherwise, retain the old best, which is given as,

$$\mathbf{S}_{best} = \psi(F) = \begin{cases} \mathbf{X}_j, & \text{if } F < G_{best} \\ \mathbf{S}_{best}, & \text{otherwise.} \end{cases} \tag{46}$$

To further randomize the process, we have included another condition to the update  $d$  and  $\tau$ . We have included a random variable  $r = \text{rand} \in [0, 1]$ ; if  $r < 0.5$ , then we will update the values of  $d$  and  $\tau$ . The pseudocode is shown in Algorithm 2.

**Algorithm 2** DBAS Algorithm.**Objective Function:**

$$F(\mathbf{X}, \mathbf{O})$$

**DBAS Initialization:**

$$\eta = 0.99$$

$$d = 0.9$$

$$d_1 = 0.9$$

$$d_2 = 0.9$$

$$\tau = 0.9$$

**Portfolio Initialization:**

$$T = iter \quad \% \text{Iterations}$$

$$D = pop \quad \% \text{Total Portfolios}$$

$$K = dim \quad \% \text{No. of Stocks In Each Portfolio}$$

$$\mathbf{X} = \text{randn}(D, K)$$

$$\mathbf{X}_{best} = \mathbf{X}$$

$$\mathbf{O} = [\mathbf{O}_1, \mathbf{O}_2, \mathbf{O}_3, \dots, \mathbf{O}_K]$$

$\% \mathbf{O}_K$  Includes Parameters For  $k$ -th Portfolio

$\% \mathbf{O}_K = [M_K, C_K]$   $M_K$  : Mean  $C_K$  : Covariance

$$P_{min} = F(\mathbf{X}, \mathbf{P})$$

$$[G_{best}, ind] = \min(F(\mathbf{X}, \mathbf{P}))$$

$$S_{best} = \mathbf{X}_{ind}$$

**For**  $i = 1 : T$

$$\mathbf{b} = \text{randn}(D, K)$$

$$b_1 = \text{randn}(1, 1)$$

$$b_2 = \text{randn}(1, 1)$$

$$b_3 = \text{randn}(1, 1)$$

Compute  $\mathbf{V}$  using (36)

Compute  $\mathbf{X}_r$  and  $\mathbf{X}_l$  using (37) and (38)

Compute  $F_r$  and  $F_l$  using (39) and (40)

Compute  $\mathbf{X}$  using (42)

**For**  $j = 1 : D$

$$F = F_j(\mathbf{X}_j, \mathbf{O}_j)$$

**If**  $F < P_{min}^j$  :

$$P_{min}^j = F$$

$$P_{best}^j = \mathbf{X}_j$$

**End If**

**End For**

$$[F, ind] = \min(F(\mathbf{X}, \mathbf{O}))$$

**If**  $F < G_{best}$  :

$$G_{best} = F$$

$$s_{best} = \mathbf{X}_{ind}$$

$$S_{best} = \text{repmat}(s_{best}, D, 1)$$

**End If**

$$r = \text{rand}$$

**If**  $r < 0.5$  :

$$d = \eta d + 0.01$$

$$d_1 = \eta d_1 + 0.01$$

$$d_2 = \eta d_2 + 0.01$$

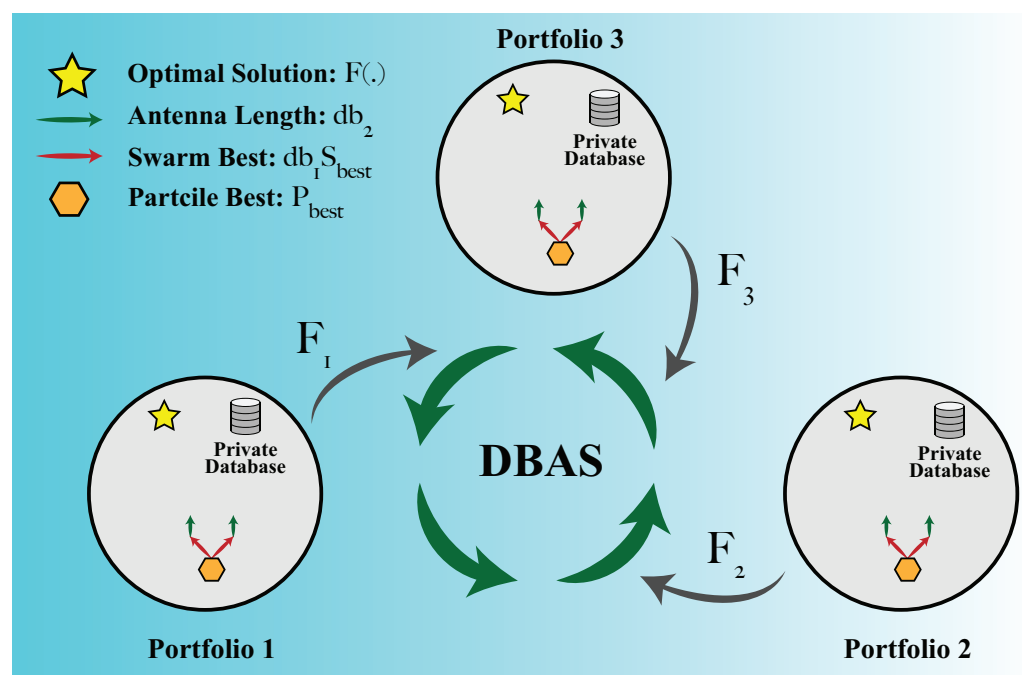
$$\tau = \eta \tau$$

**End If**

**End For**

### 3.2.2. DBAS Privacy Policy

Each particle in DBAS deals with a single portfolio. A particle alone has access to the private data of the assigned portfolio, including: stock names, investment amount, mean of the portfolio, and variance of the portfolio. The portfolio is like a black box. The particle  $X_j$  will enter the black box  $F_j(X_j, O)$ , will compute the objective function of the portfolio and will give out the objective function value  $F_j$  while keeping the privacy of the private database and portfolio information. The objective function value, i.e., gradient, will be available to all the other particles and based on all the gradients, i.e., objective function values of all particles, the DBAS algorithm will compute the swarm's best solution to decide the next step. It helps the DBAS to efficiently find the optimal selection of all portfolios without violating the confidentiality and privacy of portfolios. The concept of DBAS privacy is shown in Figure 3.



**Figure 3.** Each particle in DBAS is assigned to a single portfolio, and the particle alone has access to the private database of the portfolio. In DBAS, the particle in swarm only shares the objective function values  $F$ , i.e., gradients.

## 4. Simulation Results

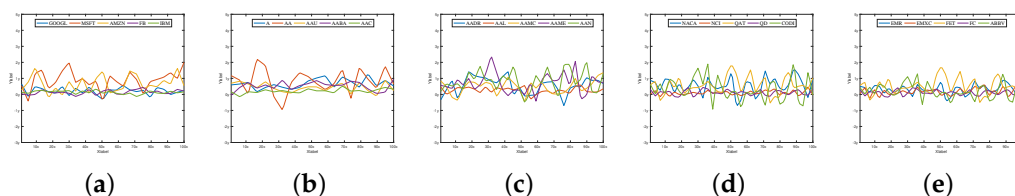
In this section, we will discuss the stock data used for the multiple portfolio selection problem. Furthermore, the DBAS algorithm will be employed to solve the optimization problem formulated in the problem formulation selection.

### 4.1. Stock Data

For the simulation, we collected real-world stock data from the NASDAQ stock market in 2017. The data comprise several hundred stock companies. A total of 25 known companies were selected for the simulation. Then, 200 days worth of stock closing prices were collected for each company to understand the data trend in more detail and to provide more robust portfolios. All the 25 companies are shown in Table 2. The volatility of the stock market is shown in Figure 4. It can be seen how abruptly the stock prices (return rate) change over time.

**Table 2.** NASDAQ Symbols of The Stock Companies In Portfolio Selection.

Companies 1–5	GOOGL	MSFT	AMZN	FB	IBM
Companies 6–10	A	AA	AAU	AABA	AAC
Companies 11–15	AADR	AAL	AAMC	AAME	AAN
Companies 16–20	NCA	NCI	QAT	QD	CODI
Companies 20–25	EMR	EMXC	FET	FC	ABBV



**Figure 4.** The return rate for all the 25 companies for 200 days of NASDAQ stock data in 2017. (a–e) represents the data of companies 1–5, companies 6–10, companies 11–16, companies 16–20, and companies 20–25, respectively.

In our simulation, there are four multi-portfolio optimization problems, i.e., 5 stock companies, 10 stock companies, 15 stock companies, and 20 stock companies. In each multi-portfolio selection problem, we will be optimizing *three* portfolios. For instance, in the *five* stock companies multi-portfolio selection problem, DBAS will optimize three portfolios, each looking for an optimal selection of stock companies. Considering the stochastic nature of the DBAS algorithm and how results can vary with each experiment, we performed simulation on each portfolio for 10 times, and the average or mean result is obtained. The computational resources used in the simulations are given below:

- Software: MATLAB;
- System: MacBook Pro;
- Processor: 2.2 GHz;
- Cores: 6–Core Intel Core i7;
- Memory: 6 GB 2400 MHz DDR4;
- Graphics: Radeon Pro 555X 4 GB.

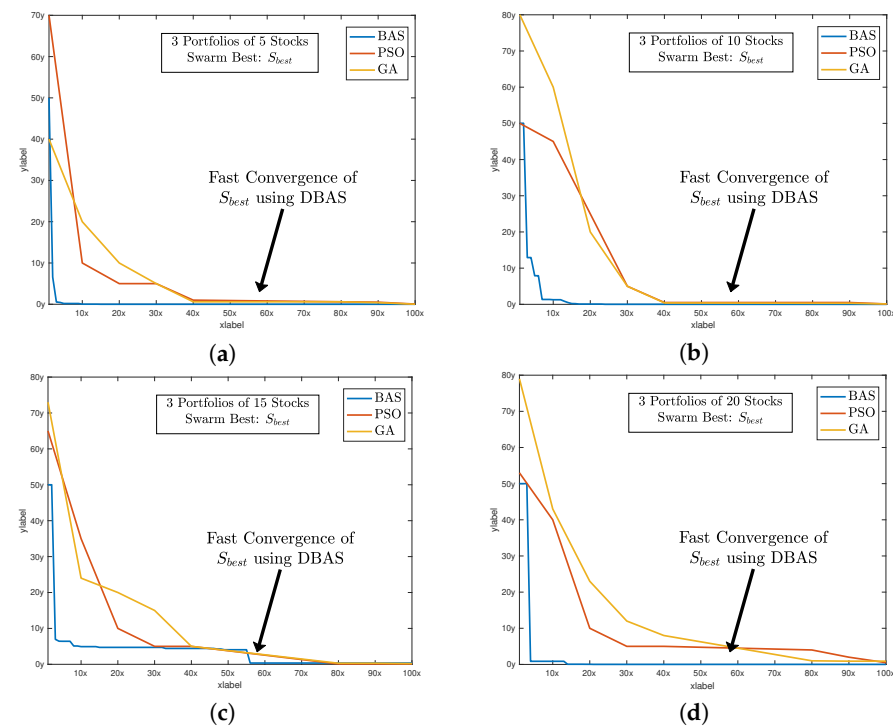
#### 4.2. Three Portfolios of Five Stock Companies

For the first simulation category, *three* portfolios were selected, each consisting of *five* companies. The objective is to optimize the three objective functions (24)–(27), i.e., portfolios, such that the raw data of each portfolio remain private and without sharing the private databases that DBAS optimizes. There are a few hyper-parameters in DBAS, i.e., antenna length  $d$  and particle step-size  $\tau$ . We selected the values of these hyper-parameters through trial and error, e.g.,  $\tau = 0.900$ ,  $d = 0.999$ , and  $\eta = 0.900$ , which are shown in Table 3. In addition, a few other parameters were selected for the simulation, i.e., total iterations  $T = 500$ , swarm population  $D = 3$ , dimension of each portfolio  $K = 5$ .

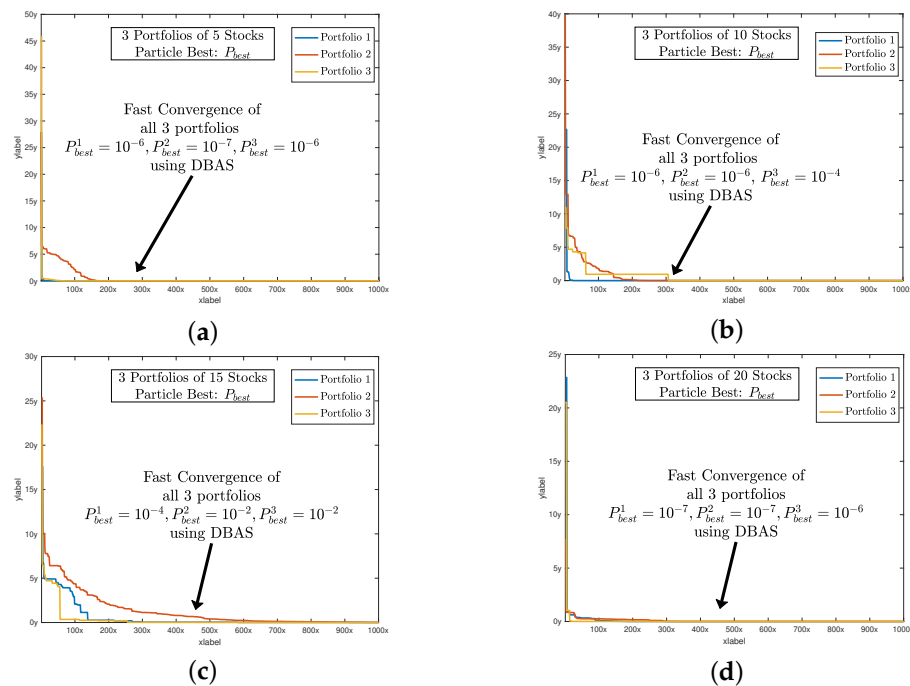
**Table 3.** DBAS Convergence Profile for Different Portfolios.

	Companies: 5			Companies: 10		
	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 1	Portfolio 2	Portfolio 3
$n$	5	5	5	7	8	9
$G_{best}$	$10^{-7}$	$10^{-7}$	$10^{-7}$	$10^{-6}$	$10^{-6}$	$10^{-6}$
$P_{best}$	$10^{-6}$	$10^{-7}$	$10^{-6}$	$10^{-6}$	$10^{-6}$	$10^{-4}$
$SR$	72.348	24.804	39.582	9.108	5.319	3.151
$(1 + \alpha)E = 1$	1.116	1.179	1.065	1.153	1.140	0.964
<b>DBAS:</b>						
$\eta$		0.900			0.900	
$d = d_1 = d_2$		0.999			0.900	
$\tau$		0.900			0.900	
$sgn$		$[-1, 1]$			$[-1, 1]$	
<b>Some Additional Parameters:</b>						
T		500			500	
D		3			3	
K		5			5	
$H(E, \sigma)$ Eval.		4500			4500	
	Companies: 15			Companies: 20		
	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 1	Portfolio 2	Portfolio 3
$n$	15	12	10	15	20	18
$G_{best}$	$10^{-4}$	$10^{-4}$	$10^{-4}$	$10^{-7}$	$10^{-7}$	$10^{-7}$
$P_{best}$	$10^{-4}$	0.020	0.001	$10^{-7}$	$10^{-7}$	$10^{-6}$
$SR$	7.088	3.667	4.667	58.71	21.56	191.6
$(1 + \alpha)E = 1$	0.930	1.088	1.103	0.9893	1.136	1.021
<b>DBAS:</b>						
$\eta$		0.850			0.950	
$d = d_1 = d_2$		0.900			0.900	
$\tau$		0.850			0.850	
$sgn$		$[-1, 1]$			$[-1, 1]$	
<b>Some Additional Parameters:</b>						
T		500			500	
D		3			3	
K		5			5	
$H(E, \sigma)$ Eval.		4500			4500	

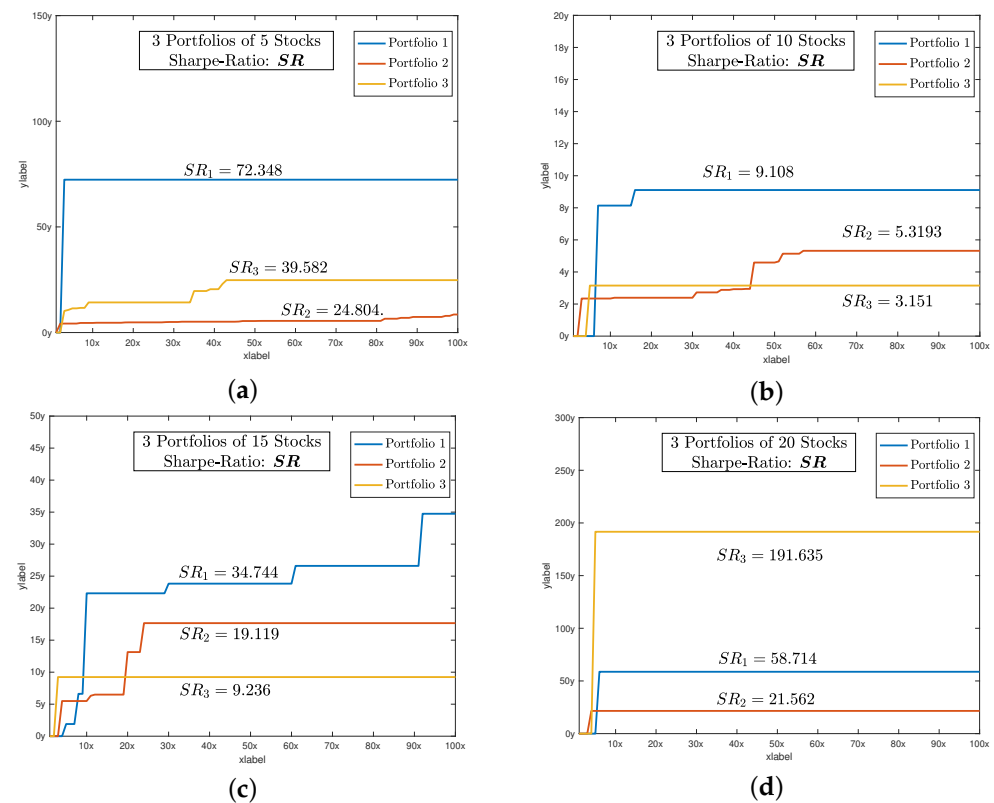
After setting all of the parameters. DBAS is ready to optimize the portfolio problem mentioned in (24)–(27). The simulation results are evaluated based on three parameters: swarm convergence  $G_{best}$ , particles convergence  $P_{best}$ , Sharpe ratio  $SR$ , and the total investment constraint  $(1 + \alpha)E = 1$ . The results are shown in Table 3 and the convergence profiles of  $G_{best}$ ,  $P_{best}$ ,  $SR$  and  $(1 + \alpha)E = 1$  are shown in Figures 5–8a. For three portfolios of five stock companies, the swarm best solution turns out to be  $10^{-7}$ , and the convergence profile in Figure 5a shows how efficiently and quick DBAS converges to the optimal solution. Likewise, the convergence of each particle, i.e., each portfolio, is also robust, as shown in Figure 6a. However, the convergence rates do not give us the relationship between the profit and risk. For that, the Sharpe ratio curve was computed. It can be seen in Figure 7a that each portfolio has achieved substantial profit with minimal risk involved. Finally, it is essential that the investment in stocks remains around  $\approx 1$ . We have shown the convergence profile of  $(1 + \alpha)E = 1$  in Figure 8a, and it can be seen that all three investments converge to around 1, which is required. The detailed results are also shown in Table 3.



**Figure 5.** The sharp and fast convergence of DBAS swarm’s  $G_{best}$  toward the minima while selecting optimal configuration of all 3 portfolios. It also shows the comparison with PSO and GA algorithms, and it can be seen that the convergence profile of DBAS is fast compared to others. (a) shows the global convergence profile of all algorithms. Likewise, (b) shows the convergence of BAS, PSO, and GA. And finally, (c,d) shows the convergence of 15 and 20 stocks of all three algorithms respectively.



**Figure 6.** The sharp and fast convergence of DBAS particles  $P_{best}$  toward their respective minimas while selecting an optimal configuration of their portfolios. Since we are dealing with three portfolios, we can see the three convergence profiles. (a) shows the global convergence profile of all algorithms. Likewise, (b) shows the convergence of BAS, PSO, and GA. And finally, (c,d) shows the convergence of 15 and 20 stocks of all three algorithms respectively.



**Figure 7.** The rising trend of Sharpe ratio  $SR$  as the DBAS moves toward the optimal solution of portfolio selection problem. The higher value of  $SR$  indicates that each portfolio has a higher profit  $\zeta(E)$  and lower risk  $v(E)$ , as shown in (23). (a) shows the convergence profile of sharpe-ratio of 5 stocks. Likewise, (b) shows the convergence of sharpe-ratio of 10 stocks. And finally, (c,d) shows the convergence of sharpe-ratio of 15 and 20 stocks respectively.

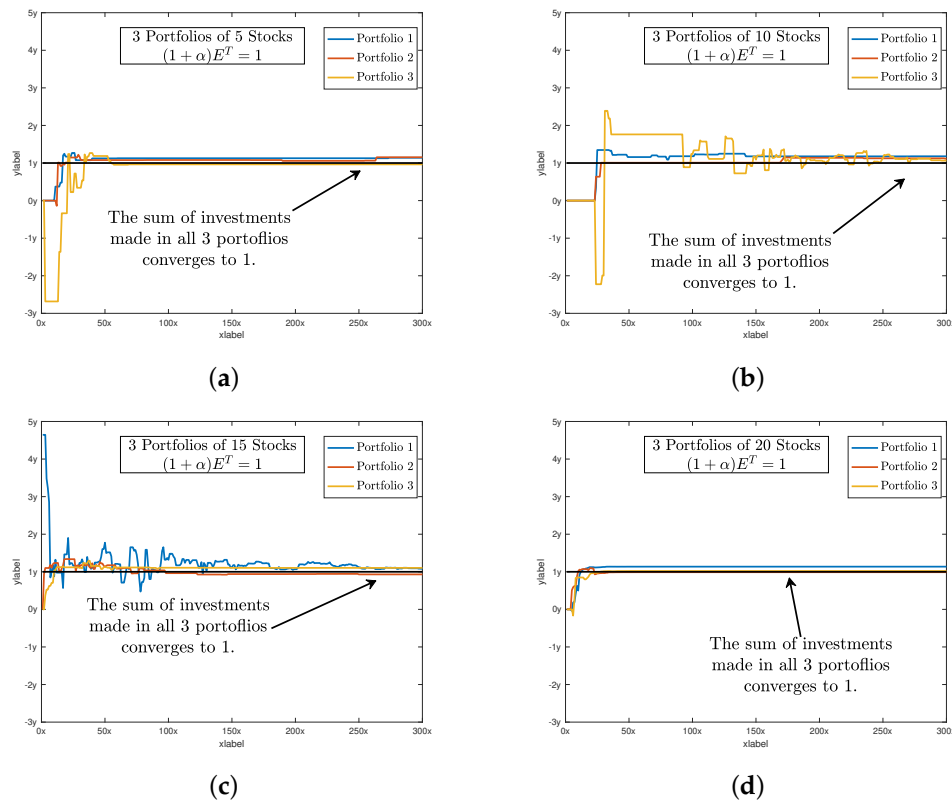
The main takeaway from the results is that DBAS can converge multi-portfolios and ensures the clients’ secrecy and privacy. In the simulated results, it can be seen that each particle in DBAS only shares the objective function value  $P_{best}^j$ , i.e., gradient with other particles, whereas the private information of the portfolio remains inaccessible to other portfolios.

### 4.3. Three Portfolios of 10 Stock Companies

For the second simulation scenario, we again selected *three* portfolios, each consisting of 10 companies. The values of hyper-parameters were once again chosen through trial and error, which are also shown in Table 3. In addition, the other parameters for the simulation include total Iterations  $T = 500$ , swarm population  $D = 3$ , dimension of each portfolio  $K = 10$ .

The simulation results are shown in Table 3, and the convergence profiles of  $G_{best}$ ,  $P_{best}$ ,  $SR$  and  $(1 + \alpha)E = 1$  are shown in Figures 5–8b. For the *three* portfolios of 10 stock companies, the swarm best solution turns out to be  $10^{-6}$ . Its convergence profile is shown in Figure 5b. Likewise, the convergence of each particle, i.e., each portfolio, is also robust and shown in Figure 6b. Their convergence values turn out to be optimal, i.e.,  $P_{best} = [10^{-6}, 10^{-6}, 10^{-4}]$ . The Sharpe ratio curve is shown in Figure 7b, and it can be seen that each portfolio has achieved substantial profit with minimal risk. We have also shown the convergence profile of  $(1 + \alpha)E = 1$  in Figure 8b, where it can be seen that all three investments converge to around 1, which is again required. The detailed results are also shown in Table 3.





**Figure 8.** It shows that as the portfolio selection problem converges toward the optimal solution, the sum of normalized investment  $(1 + \alpha)E \approx 1$ . There are some spikes in the beginning; however, all the particles of the respective portfolio converge to 1. (a) shows the convergence of normalized investment  $(1 + \alpha)E \approx 1$  of 5 stocks. Likewise, (b) shows the convergence of normalized investment of 10 stocks. And finally, (c,d) shows the convergence of normalized investment of 15 and 20 stocks respectively.

#### 4.4. Three Portfolios of 15 Stock Companies

For the third simulation scenario, we again selected *three* portfolios, each consisting of 15 companies. Once again, the values of the hyper-parameters were chosen through trial and error, which are also shown in Table 3.

The simulation results are shown in Table 3 and the convergence profiles of  $G_{best}$ ,  $P_{best}$ ,  $SR$  and  $(1 + \alpha)E = 1$  are shown in Figures 5–8c. For the *three* portfolios of 15 stock companies, the swarm’s best solution turns out to be  $10^{-4}$ . Its convergence profile is shown in Figure 5c. Likewise, the convergence of each particle, i.e., each portfolio, is also robust and shown in Figure 6c. Their convergence values turn out to be again optimal, i.e.,  $P_{best} = [10^{-4}, 0.020, 0.001]$ . The Sharpe ratio curve is shown in Figure 7b, and it can be seen that each portfolio has again achieved substantial profit with minimal risk. We have also shown the convergence profile of  $(1 + \alpha)E = 1$  in Figure 8c, and it can be seen that all three investments converge to around 1, which is required. The detailed results are also shown in Table 3.

#### 4.5. Three Portfolios of 20 Stock Companies

For the fourth and the last simulation scenario, we selected *three* portfolios, each consisting of 20 stock companies. Hyper-parameters are also shown in Table 3.

The simulation results are shown in Table 3 and the convergence profiles of  $G_{best}$ ,  $P_{best}$ ,  $SR$  and  $(1 + \alpha)E = 1$  are shown in Figures 5–8d. For *three* portfolios of 15 stock companies, the swarm’s best solution turns out to be  $10^{-7}$ . Its convergence profile is shown in Figure 5d. Likewise, the convergence of each particle, i.e., each portfolio, is also robust and shown in Figure 6d. Their convergence values turns out to be optimal, i.e.,

$P_{best} = [10^{-7}, 10^{-7}, 10^{-6}]$ . The Sharpe ratio curve is shown in Figure 7b, and it can be seen that the ratio is again substantially high. We have also shown the convergence profile of  $(1 + \alpha)E = 1$  in Figure 8d, and it can be seen that all three investments converge to around 1, which is required. The detailed results are also shown in Table 3.

The primary limitation of DBAS is the “Virtual particle,” which means that for each iteration, we have to compute the objective function value for three times for  $F_r$ ,  $F_l$ , and  $F$ . It could make it computationally expensive and consume time. In the future, we will work on eliminating the “virtual particle” limitation, and we will also include some other meta-heuristic algorithms such as GSK [57] to compare with. Likewise, we will work on some additional statistical analysis including non-parametric statistical tests such as Wilcoxon Signed Rank Test and Friedman.

## 5. Conclusions

In this paper, a framework to ensure the privacy of stock portfolios in the multi-portfolio selection problem was proposed. This involved the design of a distributed variant of the BAS algorithm known as Distributed Beetle Antennae Search (DBAS). DBAS combines the swarm-like behavior of PSO with BAS’s updating rule. Each particle in the swarm is assigned to optimize a single portfolio, and the particles only share the gradient of their respective portfolio with the swarm, avoiding the exposure of the private database and stock information of portfolios. The algorithm was then simulated on real-world stock data extracted from the NASDAQ stock market for 25 stock companies. The simulation is divided into *four* categories, and in each category, *three* portfolio models were optimized. The simulation results show that DBAS not only ensures the privacy of portfolios but is also robust, computationally economical, and time efficient.

**Author Contributions:** A.T.K.: methodology, software, validation; X.C.: Conceptualization, resources; B.L.: funding acquisition, formal analysis; A.F.: data curation, visualization, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Natural Science Foundation of China under grant 61966014.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Bienstock, D. Computational study of a family of mixed-integer quadratic programming problems. *Math. Program.* **1996**, *74*, 121–140. [\[CrossRef\]](#)
2. Ertenlice, O.; Kalayci, C.B. A survey of swarm intelligence for portfolio optimization: Algorithms and applications. *Swarm Evol. Comput.* **2018**, *39*, 36–52. [\[CrossRef\]](#)
3. Perrin, S.; Roncalli, T. Machine learning optimization algorithms & portfolio allocation. In *Machine Learning for Asset Management: New Developments and Financial Applications*; Wiley: Hoboken, NJ, USA, 2020; pp. 261–328.
4. Ta, V.D.; Liu, C.M.; Tadesse, D.A. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Appl. Sci.* **2020**, *10*, 437. [\[CrossRef\]](#)
5. Leung, M.F.; Wang, J. Minimax and Biobjective Portfolio Selection Based on Collaborative Neurodynamic Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2825–2836. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Kadry, S.; Smaili, K. A design and implementation of a wireless iris recognition attendance management system. *Inf. Technol. Control.* **2007**, *36*, 323–329.
7. Kadry, S. On the generalization of probabilistic transformation method. *Appl. Math. Comput.* **2007**, *190*, 1284–1289. [\[CrossRef\]](#)
8. Katsikis, V.N. Computational methods in portfolio insurance. *Appl. Math. Comput.* **2007**, *189*, 9–22. [\[CrossRef\]](#)
9. Zhang, Y.; Gong, H.; Yang, M.; Li, J.; Yang, X. Stepsize Range and Optimal Value for Taylor–Zhang Discretization Formula Applied to Zeroing Neurodynamics Illustrated via Future Equality-Constrained Quadratic Programming. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 959–966. [\[CrossRef\]](#)

10. Liao, B.; Zhang, Y. Different Complex ZFs Leading to Different Complex ZNN Models for Time-Varying Complex Generalized Inverse Matrices. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 1621–1631. [CrossRef]
11. Katsikis, V.N. *An Alternative Computational Method for Finding the Minimum-Premium Insurance Portfolio*, Proceedings of the AIP Conference Proceedings, Penang, Malaysia, 10–12 April 2016; AIP Publishing LLC: Melville, NY, USA, 2016; Volume 1738, p. 480020.
12. Stanimirović, P.S.; Katsikis, V.N.; Li, S. Integration enhanced and noise tolerant ZNN for computing various expressions involving outer inverses. *Neurocomputing* **2019**, *329*, 129–143. [CrossRef]
13. Lai, Z.R.; Dai, D.Q.; Ren, C.X.; Huang, K.K. A Peak Price Tracking-Based Learning System for Portfolio Selection. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2823–2832. [CrossRef] [PubMed]
14. Stanimirović, P.S.; Ćirić, M.; Katsikis, V.N.; Li, C.; Ma, H. Outer and (b, c) inverses of tensors. *Linear Multilinear Algebra* **2020**, *68*, 940–971. [CrossRef]
15. Brajević, I.; Ignjatović, J. An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems. *J. Intell. Manuf.* **2019**, *30*, 2545–2574. [CrossRef]
16. Brajević, I.; Stanimirović, P. An improved chaotic firefly algorithm for global numerical optimization. *Int. J. Comput. Intell. Syst.* **2018**, *12*, 131–148. [CrossRef]
17. Lai, Z.R.; Dai, D.Q.; Ren, C.X.; Huang, K.K. Radial Basis Functions With Adaptive Input and Composite Trend Representation for Portfolio Selection. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6214–6226. [CrossRef] [PubMed]
18. O’Cinneide, C.; Scherer, B.; Xu, X. Pooling Trades in a Quantitative Investment Process. *J. Portf. Manag.* **2006**, *32*, 33–43. [CrossRef]
19. Liao, B.; Hua, C.; Can, X.; Katsikis, V.N.; Li, S. Complex Noise-Resistant Zeroing Neural Network for Computing Complex Time-Dependent Lyapunov Equation. *Mathematics* **2022**, *10*, 2817. [CrossRef]
20. Ji, R.; Lejeune, M.A. Risk-budgeting multi-portfolio optimization with portfolio and marginal risk constraints. *Ann. Oper. Res.* **2018**, *262*, 547–578. [CrossRef]
21. Savelsbergh, M.W.P.; Stubbs, R.A.; Vandenburg, D. Multiportfolio Optimization: A Natural Next Step. In *Handbook of Portfolio Construction*; Guerard, J.B., Ed.; Springer US: Boston, MA, USA, 2010; pp. 565–581. [CrossRef]
22. Iancu, D.; Trichakis, N. Fairness and Efficiency in Multiportfolio Optimization. *Oper. Res.* **2014**, *62*, 1285–1301. [CrossRef]
23. Jing, F. Information pooling game in multi-portfolio optimization. *Contrib. Game Theory Manag.* **2017**, *10*, 27–41.
24. Zhang, G.; Zhang, Q. Multiportfolio optimization with CVaR risk measure. *J. Data Inf. Manag.* **2019**, *1*, 91–106. [CrossRef]
25. Yu, G.; Cai, X.; Long, D.Z. Multi-Portfolio Optimization: A Fairness-Aware Target-Oriented Model 2020. Available online: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3740629](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3740629) (accessed on 30 July 2022).
26. Khan, A.H.; Li, S.; Luo, X. Obstacle avoidance and tracking control of redundant robotic manipulator: An RNN-based metaheuristic approach. *IEEE Trans. Ind. Informatics* **2019**, *16*, 4670–4680. [CrossRef]
27. Khan, A.H.; Cao, X.; Li, S.; Katsikis, V.N.; Liao, L. BAS-ADAM: An ADAM based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 461–471. [CrossRef]
28. Khan, A.T.; Li, S.; Cao, X. Control framework for cooperative robots in smart home using bio-inspired neural network. *Measurement* **2021**, *167*, 108253. [CrossRef]
29. Khan, A.T.; Li, S. Human guided cooperative robotic agents in smart home using beetle antennae search. *Sci. China Inf. Sci.* **2021**, *65*, 122204. [CrossRef]
30. Khan, A.T.; Li, S.; Li, Z. Obstacle avoidance and model-free tracking control for home automation using bio-inspired approach. In *Advanced Control for Applications: Engineering and Industrial Systems*; Wiley: Hoboken, NJ, USA, 2021; p. e63.
31. Khan, A.T.; Li, S.; Zhou, X. Trajectory optimization of 5-link biped robot using beetle antennae search. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3276–3280. [CrossRef]
32. Khan, A.T.; Cao, X.; Li, Z.; Li, S. Enhanced Beetle Antennae Search with Zeroing Neural Network for online solution of constrained optimization. *Neurocomputing* **2021**, *447*, 294–306. [CrossRef]
33. Zivkovic, M.; Bacanin, N.; Venkatachalam, K.; Nayyar, A.; Djordjevic, A.; Strumberger, I.; Al-Turjman, F. COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach. *Sustain. Cities Soc.* **2021**, *66*, 102669. [CrossRef]
34. Wu, Q.; Shen, X.; Jin, Y.; Chen, Z.; Li, S.; Khan, A.H.; Chen, D. Intelligent beetle antennae search for UAV sensing and avoidance of obstacles. *Sensors* **2019**, *19*, 1758. [CrossRef]
35. Li, X.; Jiang, H.; Niu, M.; Wang, R. An enhanced selective ensemble deep learning method for rolling bearing fault diagnosis with beetle antennae search algorithm. *Mech. Syst. Signal Process.* **2020**, *142*, 106752. [CrossRef]
36. Huang, J.; Duan, T.; Zhang, Y.; Liu, J.; Zhang, J.; Lei, Y. Predicting the permeability of pervious concrete based on the beetle antennae search algorithm and random forest model. *Adv. Civ. Eng.* **2020**, *2020*, 8863181. [CrossRef]
37. Sun, Y.; Zhang, J.; Li, G.; Wang, Y.; Sun, J.; Jiang, C. Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes. *Int. J. Numer. Anal. Methods Geomech.* **2019**, *43*, 801–813. [CrossRef]
38. Khan, A.T.; Cao, X.; Li, S.; Katsikis, V.N.; Brajevic, I.; Stanimirovic, P.S. Fraud detection in publicly traded US firms using Beetle Antennae Search: A machine learning approach. *Expert Syst. Appl.* **2021**, *191*, 116148. [CrossRef]
39. Khan, A.T.; Cao, X.; Brajevic, I.; Stanimirovic, P.S.; Katsikis, V.N.; Li, S. Non-linear Activated Beetle Antennae Search: A Novel Technique for Non-Convex Tax-Aware Portfolio Optimization Problem. *Expert Syst. Appl.* **2022**, *197*, 116631. [CrossRef]
40. Khan, A.H.; Cao, X.; Xu, B.; Li, S. Beetle Antennae Search: Using Biomimetic Foraging Behaviour of Beetles to Fool a Well-Trained Neuro-Intelligent System. *Biomimetics* **2022**, *7*, 84. [CrossRef]

41. Liao, B.; Han, L.; He, Y.; Cao, X.; Li, J. Prescribed-Time Convergent Adaptive ZNN for Time-Varying Matrix Inversion under Harmonic Noise. *Electronics* **2022**, *11*, 1636. [[CrossRef](#)]
42. Liao, B.; Huang, Z.; Cao, X.; Li, J. Adopting Nonlinear Activated Beetle Antennae Search Algorithm for Fraud Detection of Public Trading Companies: A Computational Finance Approach. *Mathematics* **2022**, *10*, 2160. [[CrossRef](#)]
43. Khan, A.T.; Cao, X.; Li, Z.; Li, S. Evolutionary Computation Based Real-time Robot Arm Path-planning Using Beetle Antennae Search. *EAI Endorsed Trans. Robot.* **2022**, *1*, 1–10. [[CrossRef](#)]
44. Chen, Z.; Walters, J.; Xiao, G.; Li, S. An Enhanced GRU Model With Application to Manipulator Trajectory Tracking. *EAI Endorsed Trans. Robot.* **2022**, *1*, 1–11. [[CrossRef](#)]
45. Ijaz, M.U.; Khan, A.T.; Li, S. Bio-inspired BAS: Run-time Path-planning And The Control of Differential Mobile Robot. *EAI Endorsed Trans. Robot.* **2022**, *1*, 1–10. [[CrossRef](#)]
46. Hameed, S.W. Peaks Detector Algorithm after CFAR for Multiple Targets Detection. *EAI Endorsed Trans. Robot.* **2022**, *1*, 1–7. [[CrossRef](#)]
47. Khan, A.T.; Cao, X.; Li, S. Dual Beetle Antennae Search system for optimal planning and robust control of 5-link biped robots. *J. Comput. Sci.* **2022**, *60*, 101556. [[CrossRef](#)]
48. Khan, A.H.; Cao, X.; Li, S. Obstacle avoidance based decision making and management of articulated agents. In *Management and Intelligent Decision-Making in Complex Systems: An Optimization-Driven Approach*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–29.
49. Khan, A.H. Neural Network and Metaheuristic Based Learning and Control of Articulated Robotic Agents. Ph.D. Thesis, Hong Kong Polytechnic University, Hong Kong, China, 2021.
50. Khan, A.T.; Cao, X.; Li, S.; Hu, B.; Katsikis, V.N. Quantum beetle antennae search: A novel technique for the constrained portfolio optimization problem. *Sci. China Inf. Sci.* **2020**, *64*, 152204. [[CrossRef](#)]
51. Khan, A.H.; Cao, X.; Katsikis, V.N.; Stanimirović, P.; Brajević, I.; Li, S.; Kadry, S.; Nam, Y. Optimal portfolio management for engineering problems using nonconvex cardinality constraint: A computing perspective. *IEEE Access* **2020**, *8*, 57437–57450. [[CrossRef](#)]
52. Chen, T.; Zhu, Y.; Teng, J. Beetle swarm optimisation for solving investment portfolio problems. *J. Eng.* **2018**, *2018*, 1600–1605. [[CrossRef](#)]
53. Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Li, S.; Cao, X. *Time-Varying Mean-Variance Portfolio Selection under Transaction Costs and Cardinality Constraint Problem via Beetle Antennae Search Algorithm (BAS)*, *Proceedings of the Operations Research Forum, Online, 31 August–3 September*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 2, pp. 1–26.
54. Khan, A.R.; Khan, A.T.; Salik, M.; Bakhsh, S. An optimally configured HP-GRU model using hyperband for the control of wall following robot. *Int. J. Robot. Control Syst.* **2021**, *1*, 66–74. [[CrossRef](#)]
55. Zhu, H.; Wang, Y.; Wang, K.; Chen, Y. Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem. *Exp. Syst. Appl.* **2011**, *38*, 10161–10169. [[CrossRef](#)]
56. Jiang, X.; Li, S. BAS: Beetle Antennae Search Algorithm for Optimization Problems. *arXiv* **2017**, arXiv:1710.10724.
57. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1501–1529. [[CrossRef](#)]