

Article

# Diversification of Time-Varying Tangency Portfolio under Nonlinear Constraints through Semi-Integer Beetle Antennae Search Algorithm

Vasilios N. Katsikis \*  and Spyridon D. Mourtas 

Department of Economics, Division of Mathematics and Informatics, National and Kapodistrian University of Athens, Sofokleous 1 Street, 10559 Athens, Greece; spirosmourtas@gmail.com

\* Correspondence: vaskatsikis@econ.uoa.gr

**Abstract:** In finance, the most efficient portfolio is the tangency portfolio, which is formed by the intersection point of the efficient frontier and the capital market line. This paper defines and explores a time-varying tangency portfolio under nonlinear constraints (TV-TPNC) problem as a nonlinear programming (NLP) problem. Because meta-heuristics are commonly used to solve NLP problems, a semi-integer beetle antennae search (SIBAS) algorithm is proposed for solving cardinality constrained NLP problems and, hence, to solve the TV-TPNC problem. The main results of numerical applications in real-world datasets demonstrate that our method is a splendid substitute for other evolutionary methods.

**Keywords:** beetle antennae search algorithm; nonlinear programming; portfolio selection; tangency portfolio; Sharpe ratio



**Citation:** Katsikis, V.N.; Mourtas, S.D. Diversification of Time-Varying Tangency Portfolio under Nonlinear Constraints through Semi-Integer Beetle Antennae Search Algorithm. *AppliedMath* **2021**, *1*, 63–73. <https://doi.org/10.3390/appliedmath1010005>

Academic Editor: Tommi Sottinen

Received: 9 November 2021

Accepted: 16 December 2021

Published: 20 December 2021

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



**Copyright:** © 2021 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 models aid in the selection of financial assets by investors. As a result, when making financial decisions, portfolio management is critical. Nowadays, using modern optimization techniques, popular fields in finance such as securities trading, option replication, investment banking, risk management, and so on may be effectively handled. Nature inspired algorithms [1–4], conic programming [5], branch and bound technique [6], non-differential optimization and cutting planes techniques [7], Riesz-space theory [8,9] are some of these techniques. More precisely, the problem of finding a Markowitz based portfolio is tackled by nature inspired algorithms in [1,3,4] and by conic programming in [5]. To solve a portfolio insurance problem, the Riesz space theory is used in [8,9] and meta-heuristics are used in [2], whereas in [7], non-differential optimization and cutting planes techniques are employed to solve a conditional value at risk portfolio problem. This paper defines and explores a time-varying tangency portfolio under nonlinear constraints (TV-TPNC) problem as a nonlinear programming (NLP) problem. It is worth noting that the most efficient portfolio in finance is the tangency portfolio, which is formed by the intersection point of the the efficient frontier and capital market line (CML). Because meta-heuristics are commonly used to solve NLP problems, a semi-integer beetle antennae search (SIBAS) algorithm is proposed to solve the TV-TPNC problem.

SIBAS is a hybrid algorithm that combines the beetle antennae search (BAS) algorithm from [10] with the binary BAS (BBAS) algorithm from [11]. Broadly, nature inspired optimization algorithms have been widely used in a variety of scientific domains in recent years, such as finance, computer science and engineering. BAS was chosen in this paper from a vast range of nature inspired meta-heuristics because of its minimal time consumption. BAS has been used to address problems in engineering portfolio optimization [12], asset distribution [13], assets' insurance selection [14], pattern classification [15], machine learning [16], mathematical programming [11], electro-hydraulic position systems [17], integrated circuits [18], tomography diagnosis [19].

This work's main points can be summarized as follows:

- We define and explore the TV-TPNC problem as a NLP problem.
- To tackle NLP problems with cardinality constraints, a hybrid algorithm, called SIBAS, is proposed.
- We present the SIBAS efficiency against particle swarm optimization (PSO), differential evolution (DE) and slime mould algorithm (SMA) on a financial NLP problem.

The paper is constituted as follows. Section 2 defines and analyses the TV-TPNC problem. In Section 3, the SIBAS is provided for dealing with cardinality constrained NLP problems as well as the TV-TPNC problem. Section 4 offers two applications that employ real-world data to compare SIBAS performance to the PSO, SMA, and DE in various and somewhat large portfolio setups. The MATLAB repository that has been made available on GitHub is also mentioned in Section 4. This repository supports the readability and computational utility of this work by implementing all of the algorithms described in Section 3. Lastly, the final remarks are offered in Section 5.

## 2. Tangency Portfolio Optimization

The mean-variance optimization theory of Markowitz provides a mechanism for selecting assets (or securities) portfolios that trades off expected returns and risk of prospective portfolios. For a given level of risk, investors that utilize mean-variance resolution to maximize their expected return always prefer portfolios that are on the CML. If a feasible portfolio has the highest expected return among all portfolios with the same variance, or if it has the lowest variance among all portfolios with at least a specific expected return, it is said to be efficient. The efficient frontier of the portfolio universe is made up of a collection of efficient portfolios. The most efficient portfolio, dubbed the tangency portfolio, is found at the point where the CML intersects with the efficient frontier.

Any portfolio  $p$  with one or more risky assets and one risk-free asset may have a linear connection between its expected return  $r_p$  and its risk  $\sigma_p$ , according to Sharpe Ratio (SR) [20]. Mathematically, this can be stated in the following way:

$$r_p = r_f + S_p \sigma_p, \quad (1)$$

where  $r_f$  denotes the risk-free asset's return and  $S_p$  denotes the portfolio's SR, which is the risk premium per risk unit.

The tangency portfolio optimization given in [21] is the foundation of our approach to the TV-TPNC problem. A rationalistic risk averse investor's endowment will be divided, with a proportion  $\gamma$  invested in a risk-free asset and the rest  $(1 - \gamma)$  in a time-varying portfolio of risky assets  $p(t)$ ,  $t \subseteq \mathbb{N}$ , whereas  $S_p(t)$  is determined by the composition of  $p(t)$ , which is based on the common capital market hypothesis of one risk-free and many risky assets. Consider the market space  $X(t) = [x_1(t), x_2(t), \dots, x_n(t)] \in \mathbb{R}^n$  that contains  $n$  assets prices, the investor would choose the weights  $p_i(t)$ , for the assets  $i = 1, 2, \dots, n$ , included in the portfolio  $p(t) = [p_1(t), p_2(t), \dots, p_n(t)] \in \mathbb{R}^n$  to optimize  $S_p(t)$ . It is worth emphasizing that  $\gamma$  reflects the investor's risk aversion, and that all investor's  $p_i(t)$  must be the alike. As a result, the time-varying tangency portfolio  $p(t)$  can be computed without considering the risk aversion or utility function of the investor.

Moreover, investors prefer portfolios with a lower number of different assets since handling portfolios with a big number of various assets may be time intensive [22]. A key consideration during the portfolio selection process is that most of a portfolio's risk diversification may be achieved with a small but well-selected collection of assets [21]. Mathematically, a cardinality constraint (CC) can be used to any portfolio optimization problem to achieve this. Thus, the fixed number  $K$  denotes the exact amount of assets an investor can own, avoiding over-diversification, while CC is expressed as the binary vector  $D(t) = [D_1(t), D_2(t), \dots, D_n(t)] \in \mathbb{R}^n$ , which signify the assets in the portfolio and can have a value of 1 or 0, where  $D_i(t) = 1$  signifies that the investor owns the asset  $i$  and

$D_i(t) = 0$  signifies the opposite. Thus, the time-varying CC function can be formulated as follows:

$$D_i(t) = \begin{cases} 1, & p_i(t) > 0 \\ 0, & p_i(t) = 0 \end{cases} \tag{2}$$

MPT frequently considers an ideal market in which short sales are prohibited but shares are infinitely separable and hence may be sold in any non-negative partition, free of taxes and transaction costs (TC). TC can refer to a variety of expenses like fund loads, taxes, bid-ask spreads, brokerage charges, and so on. Inline with [23], we will consider  $\theta^-$ ,  $\theta^+$  the fixed charges prices generated from the sell and buy of an asset  $i$ , and  $\zeta^-$ ,  $\zeta^+$  the cost charges generated from the sell and buy of an asset  $i$ . Thus, TC generated from the sell and buy of an asset are separate, and the time-varying TC function can be formulated as follows:

$$G_i(t) = \begin{cases} 0, & p_i(t) = p_i(t-1) \\ \theta^+ + \zeta^+(p_i(t) - p_i(t-1))x_i(t), & p_i(t) > p_i(t-1) \\ \theta^- + \zeta^-(p_i(t-1) - p_i(t))x_i(t), & p_i(t) < p_i(t-1) \end{cases} \tag{3}$$

and  $G(t) = [G_1(t), G_2(t), \dots, G_n(t)] \in \mathbb{R}^n$ . Apart from the case of zero costs, (3) is nonconvex.

According to the aforementioned, if a market  $X(t)$  of  $n$  assets exists, in which only  $K$  of them have to be included in  $p(t)$ , the TV-TPNC problem can be formulated as follows:

$$\max_p \quad S_p(t) - \sum_{i=1}^n G_i(t) \tag{4}$$

$$\text{subject to} \quad S_p(t) = \frac{r_p(t) - r_f(t)}{\sigma_p(t)} \tag{5}$$

$$r_p(t) = \sum_{i=1}^n p_i(t)r_i(t) \tag{6}$$

$$\sigma_p(t) = \sqrt{\sum_{i=1}^n \sum_{j=1}^n p_i(t)p_j(t)\sigma_{ij}(t)} \tag{7}$$

$$\sum_{i=1}^n D_i(t) = K, \forall i, \tag{8}$$

where  $r_i(t)$  signifies the expected return of asset  $i$  at time  $t$ , and  $\sigma_{ij}(t)$  signifies the covariance among the expected returns of assets  $i$  and  $j$  at time  $t$ .

The following improvements are made to transform the TV-TPNC to an NLP problem and make it more realistic. We use past values (or delays) to construct the variance (risk), covariance matrix and expected return of the market  $X(t)$ . Representing the delays with the constant number  $\beta \in \mathbb{N}$ , we consider  $r(t) = [r_1(t), r_2(t), \dots, r_n(t)] \in \mathbb{R}^n$  the expected return of  $X(t)$ , where  $r_i(t) = \sum_{z=0}^{\beta-1} (x_i(t-z))/\beta \in \mathbb{R}$  signifies the asset's  $i$ ,  $i = 1, 2, \dots, n$ , expected return at time  $t$ , and  $C(t) \in \mathbb{R}^{n \times n}$  the covariance matrix of  $X(t)$  based on  $\beta$  in number delays. In this way, we can set  $r_p(t) = p^T(t)r(t)$  and  $\sigma_p(t) = \sqrt{p^T(t)C(t)p(t)}$ . It is worth noting that  $X(t)$  contains both risk-free and risky assets. However, when it comes to investing, there is no such thing as an asset that is risk-free because nothing can be guaranteed 100 percent. As a result, in our model, risk-free assets are defined as market assets with a variance below a small fixed value  $\alpha$ . Thus, setting  $H(t) = [h_1(t), h_2(t), \dots, h_n(t)]$ , where  $h_i(t) = 1$ , if  $\text{Var}[h_i(t)] < \alpha$  with  $\text{Var}[J]$  signifying the variance of  $i$ , and  $h_i(t) = 0$ , otherwise, we have that  $r_f(t) = p^T(t)(H(t) \odot r(t))$  with  $\odot$  signifying the Hadamard (or element-wise) product. It is also worth noting that the price of each asset  $x_i$  is normalized inline with its  $\beta$  in number delays.

The TV-TPNC problem may be expressed in the following NLP formation based on the aforementioned analysis:

$$\min_p \quad G^T(t)\mathbf{1} - \frac{p^T(t)(r(t) - H(t) \odot r(t))}{\sqrt{p^T(t)C(t)p(t)}} \quad (9)$$

$$\text{subject to} \quad p^T(t)\mathbf{1} = 1 \quad (10)$$

$$D^T(t)\mathbf{1} = K \quad (11)$$

$$\mathbf{0} \leq p(t) \leq \mathbf{1} \quad (12)$$

where  $\mathbf{0}, \mathbf{1} \in \mathbb{R}^n$  signify a zero vector and a vector of ones, respectively.

### 3. The Semi-Integer Beetle Antennae Search Model

The computational procedures utilized to handle the given financial NLP problem in a brief period of time with great accuracy are the main emphasis of this paper. As a result, a hybrid algorithm called SIBAS is developed, which is based on a nature inspired algorithm called BAS, whose primary advantage is its low time consumption. SIBAS combines BAS and BBAS to better handle cardinality constrained NLP problems.

#### 3.1. The SIBAS Algorithm

BAS is a nature inspired algorithm that finds the best solution to an optimization problem by mimicking the behavior of a beetle [10], while a binary type of BAS named BBAS was presented in [11]. In these algorithms, the way the beetle’s two antennae detect the intensity of a smell and use it to track food is related to finding the minimum of an objective function. Due to the fact that these algorithms are only applicable to optimization without constrains, a complementary procedure have to be used to keep solutions inside the acceptable range. The penalty method [24] is chosen as the supplementary procedure for manipulating nonconvex or convex constraints more effectively in this study.

Penalty methods work in a succession of steps, each time altering a set of penalty parameters and initiating a new one using the previous. Throughout the building of any sequence, the penalty function that follows is minimized:

$$F(w) = f(w) + U(R, q(w)), \quad (13)$$

where  $f(w)$  signifies the objective function. Furthermore,  $U(R, q(w))$  signifies the penalty term, where  $q(w)$  denotes the inequality/equality constraint and  $R$  denotes a set of penalty parameters. Commonly, this procedure has the benefit of being able to indulge any non-convex or convex constraint. Inhere, the penalty method incorporates the bracket operator  $\langle \cdot \rangle$  to manage  $U(R, q(w))$ . If the input value is positive, the bracket operator returns 0, else it returns the input value. Thus, penalty term can be formulated as follows:

$$U(R, q(w)) = R\langle q_j(w) \rangle^2, \quad \forall j, \quad (14)$$

where  $q_j(w)$  signifies the  $j$ -th inequality/equality constraint.

The SIBAS may be described as follows. At  $i$ -th time moment, consider the position of the beetle as a vector  $x_i, i = 1, 2, \dots$ . Then, the gathering of odour is the objective functions  $F_1(x)$  and  $F_2(x)$  at position  $x$ . As a result, the minimum value of  $F_1(x)$  and  $F_2(x)$  is linked to the odour’s source spot. Note the  $F_1(x)$  is (13) with only the CC of the NLP problem, while  $F_2(x)$  is (13) with all the rest inequality/equality constraints of the NLP problem. The model of seeking behavior is defined as follows by a random searching path of the beetle:

$$A = \text{round}(\text{rnd}(n, 1)), \quad (15)$$

where  $\text{rnd}(\cdot)$  and  $\text{round}(\cdot)$  signify a random and a round function, respectively, while  $n$  signifies the position's dimensions. The right ( $x_R$ ) and left ( $x_L$ ) antennae are composed as bellow to replicate the seeking behaviors of the beetle's antennae:

$$x_R = \begin{cases} 1, & x_{i-1} + A > 1 \\ 0, & x_{i-1} + A < 0 \end{cases} \quad (16)$$

$$x_L = \begin{cases} 1, & x_{i-1} - A > 1 \\ 0, & x_{i-1} - A < 0 \end{cases} \quad (17)$$

Moreover, assuming the candidate optimal solution as bellow:

$$x_C = \begin{cases} x_R, & F_1(x_R) < F_1(x_L) \\ x_L, & F_1(x_R) > F_1(x_L) \end{cases} \quad (18)$$

the behavior of detecting may be formulated as bellow:

$$x_i = \begin{cases} x_C, & F_1(x_C) < F_1(x_{i-1}) \\ x_{i-1}, & F_1(x_C) > F_1(x_{i-1}) \end{cases} \quad (19)$$

Note that  $i$  signifies the iteration number. Given that  $y$  is the optimal solution of  $F_1(\cdot)$ , a new random seeking path is created for optimizing  $F_2(\cdot)$ . Hence, setting  $g = \text{rnd}(n, 1)$  at position  $x_{i-1}$ , the random path is as bellow:

$$B = \frac{g}{2^{-52} + \|g\|} \quad (20)$$

Imitating the antennae motions, we have:

$$x_L = x_{i-1} - dB, \quad x_R = x_{i-1} + dB, \quad (21)$$

where the detecting diameter of the antennae is denoted by  $d$ , which is related to the ability to exploit. In addition, considering the candidate optimal solution:

$$x_C = \|x_{i-1} + \delta B \text{sign}(F_2(x_R) - F_2(x_L))\| \odot y, \quad (22)$$

where the term  $\delta$  refers to a size step that corresponds to the pace of convergence following an increase in  $i$  during the search. In this way, the optimal solution of  $F_1(\cdot)$  is merged with the solution of  $F_2(\cdot)$ , while only specific elements of  $x_C$  are allowed to be modified. Hence, the behavior of detecting may be formulated as bellow:

$$x_i = \begin{cases} x_C, & F_2(x_C) < F_2(x_{i-1}) \\ x_{i-1}, & F_2(x_C) > F_2(x_{i-1}) \end{cases} \quad (23)$$

Finally, the  $d$  and  $\delta$  update rules are as follows:

$$\delta = 0.991\delta, \quad d = 0.991d + 0.001. \quad (24)$$

### 3.2. SIBAS Approach on the TV-TPNC Problem and the Complete Process

Given the market dataset  $M$ , which comprises of assets prices time-series, the market space  $X(t)$  along with the span of time-period  $t$  are determined based on the delays number  $\beta$ . In addition,  $C(t)$  and  $R(t) = r(t) - H(t) \odot r(t)$  can be constructed based on the analysis presented in Section 2. Setting the initial position of the beetle as the initial portfolio of the TV-TPNC problem as well as the penalty functions according to the analysis presented in Section 3.1, the TV-TPNC problem of (9)–(12) can be solved with the SIBAS algorithm.

More precisely, the penalty functions for the TV-TPNC problem of (9)–(12) can be written in MATLAB routines as follows:

$$F_1(p) = f(p) + R^2 * (\text{sum}(p > 0) \sim K), \tag{25}$$

$$F_2(p) = f(p) + R * \text{sum}((\text{sum}(p) - 1 \sim 0) * (\text{sum}(p) - 1).^2 + (p < 0) * p.^2 + (p - 1 > 0) * (p - 1).^2), \tag{26}$$

where  $\text{sum}(\cdot)$  signifies the MATLAB routine for summing the elements of an input array and  $f(p)$  is (9).

The complete process to solve the TV-TPNC problem of (9)–(12) using the SIBAS approach is presented in Algorithm 1, where the  $\text{zeros}(\cdot)$ ,  $\text{mean}(\cdot)$ ,  $\text{var}(\cdot)$  and  $\text{cov}(\cdot)$  signify regular MATLAB routines.

**Algorithm 1** The complete process to solve the TV-TPNC problem of (9)–(12) using SIBAS.

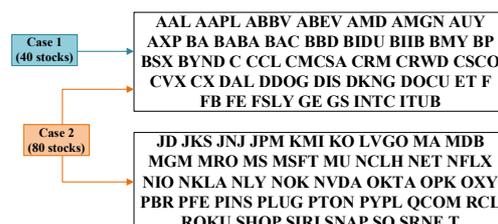
**Require:** The market dataset  $M$ ; the delays number  $\beta$ ; the initial portfolio  $p_{\text{in}}$  and the value of parameter  $\alpha$ .

- 1: Set  $[m, n] = \text{size}(M)$ ,  $t_{\text{end}} = m - \beta$ ,  $r = \text{zeros}(t_{\text{end}}, n)$ ,  $X = \text{zeros}(t_{\text{end}}, n)$  and  $C\{t_{\text{end}}, 1\} = \{ \}$
- 2: **for**  $t = 1 : t_{\text{end}}$  **do**
- 3:   Set  $s = M(t : \beta + t - 1, :)$ ,  $s = s ./ \max(s)$  and  $X(t, :) = s(\beta, :)$
- 4:   Set  $C\{t, 1\} = \text{cov}(s)$  and  $r(t, :) = \text{mean}(s) - (\text{var}(s) < \alpha) * \text{mean}(s)$
- 5: **end for**
- 6: Set  $p_{\text{opt}} = \text{zeros}(n, t_{\text{end}})$
- 7: Set  $p_{\text{opt}}(t)$  the optimal solution of SIBAS algorithm based on the initial portfolio  $p_{\text{in}}$
- 8: **for**  $t = 2 : t_{\text{end}}$  **do**
- 9:   Set  $p_{\text{opt}}(t)$  the optimal solution of SIBAS algorithm based on the previous portfolio  $p_{\text{opt}}(t - 1)$
- 10: **end for**
- 11: **return**  $p_{\text{opt}}(t)$  for  $t \in [1, t_{\text{end}}] \subseteq \mathbb{N}$

**Ensure:** The optimal solution  $p_{\text{opt}}(t)$  of the TV-TPNC problem of (9)–(12).

#### 4. Applications

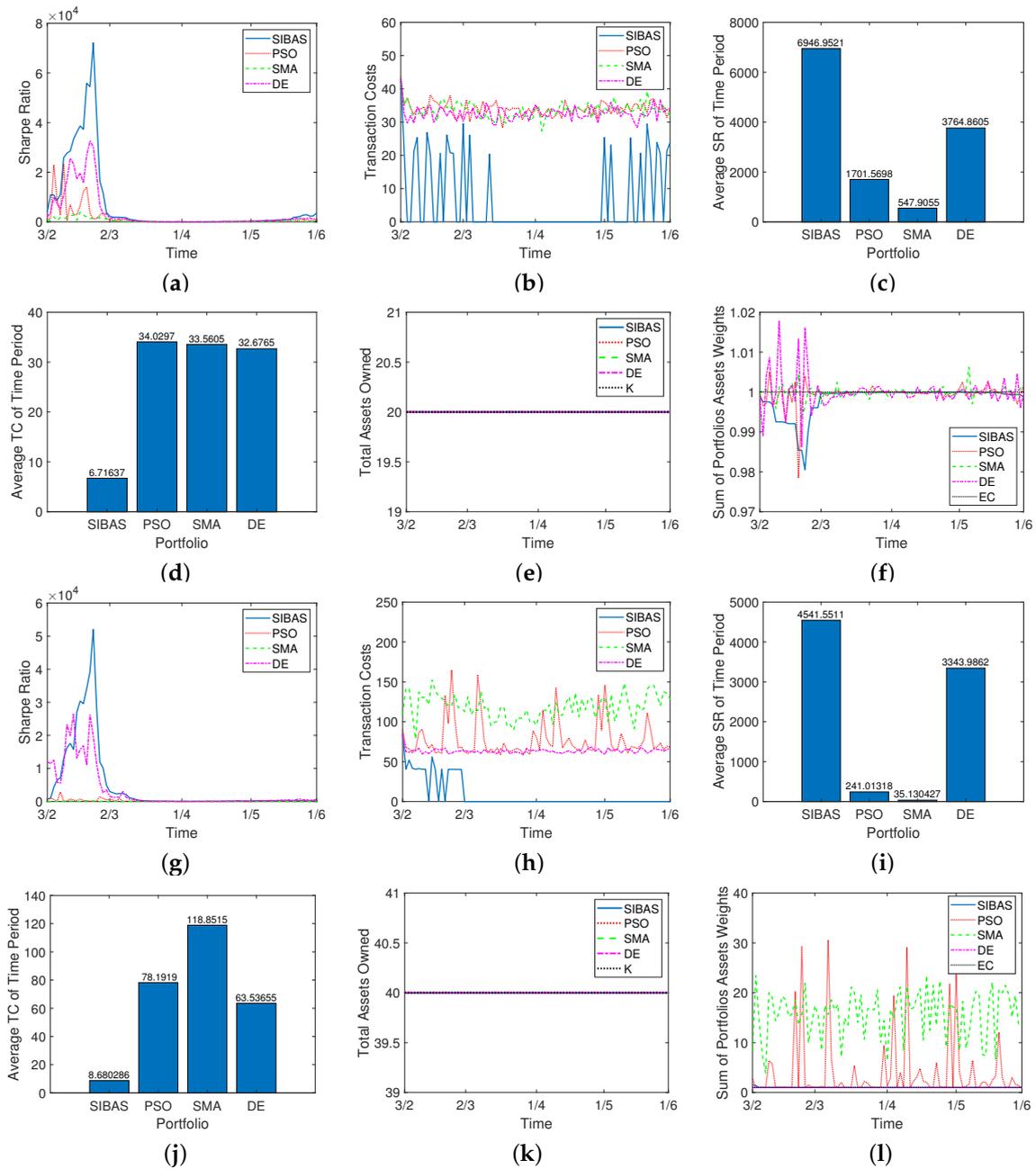
This section compares and contrasts SIBAS’ performance with those of state-of-the-art meta-heuristics algorithms such as PSO of MATLAB, DE of [25] and SMA of [26] in solving the TV-TPNC problem of (9)–(12). The daily close prices of the stocks shown in Figure 1 are the real-world data employed. This figure contains stocks’ ticker symbols divided into two portfolio’s cases. This section also contains information regarding the data and code availability. Moreover, in all experiments along with all the nature inspired algorithms used inhere, the penalty parameter has been set to  $R = 10^5$ , and the maximum iterations to  $10^3$ . The SIBAS parameters have been set to  $d = 0.2$  and  $\delta = 0.5$ , the PSO used with its default settings and the population size of SMA and DE have been set to 30 and 50, respectively. The variance (risk) number has been set to  $\alpha = 10^{-3}$ , and the delays number has been set to  $\beta = 40$ , while the parameters in (3) have been set to  $\zeta^- = 2$ ,  $\zeta^+ = 4$  and  $\theta^- = \theta^+ = 1$ .



**Figure 1.** The stocks that are employed in each portfolio case.

4.1. Real-World Data Portfolio Cases

In the  $s$ -th portfolio case,  $s = 1, 2$ , we assume the market dataset to be  $M \in \mathbb{R}^{123 \times w}$ . Note that  $s = 1$  has  $w = 40$  and  $K = 20$ , while  $s = 2$  has  $w = 80$  and  $K = 40$ . Based on this and the number of delays, we construct the market  $X(t) = [x_1, x_2, \dots, x_w] \in \mathbb{R}^w$  for  $t = 1$  to 83. That is,  $X(t)$  contains 83 daily prices of  $w$  in number stocks that correspond to the time-period 3/2/2020-1/6/2020. Because of the cardinality number, the optimal portfolio  $p_{opt}(t)$  holds exactly  $K$  in number stocks, at least one of which is risk-free. The findings for solving the TV-TPNC problem with initial portfolio  $p_{in} = \mathbf{1}/w \in \mathbb{R}^w$  are presented in Figure 2a-f for the portfolio case 1, and in Figure 2g-l for the portfolio case 2.



**Figure 2.** The SR and TC, the average SR and TC of time-period, the total assets owned and the equality constraint of the two portfolio cases. (a) SR in portfolio case 1; (b) TC in portfolio case 1; (c) Average SR in portfolio case 1; (d) Average TC in portfolio case 1; (e) Total assets owned in portfolio case 1; (f) Equality constraint in portfolio case 1; (g) SR in portfolio case 2; (h) TC in portfolio case 2; (i) Average SR in portfolio case 2; (j) Average TC in portfolio case 2; (k) Total assets owned in portfolio case 2; (l) Equality constraint in portfolio case 2.

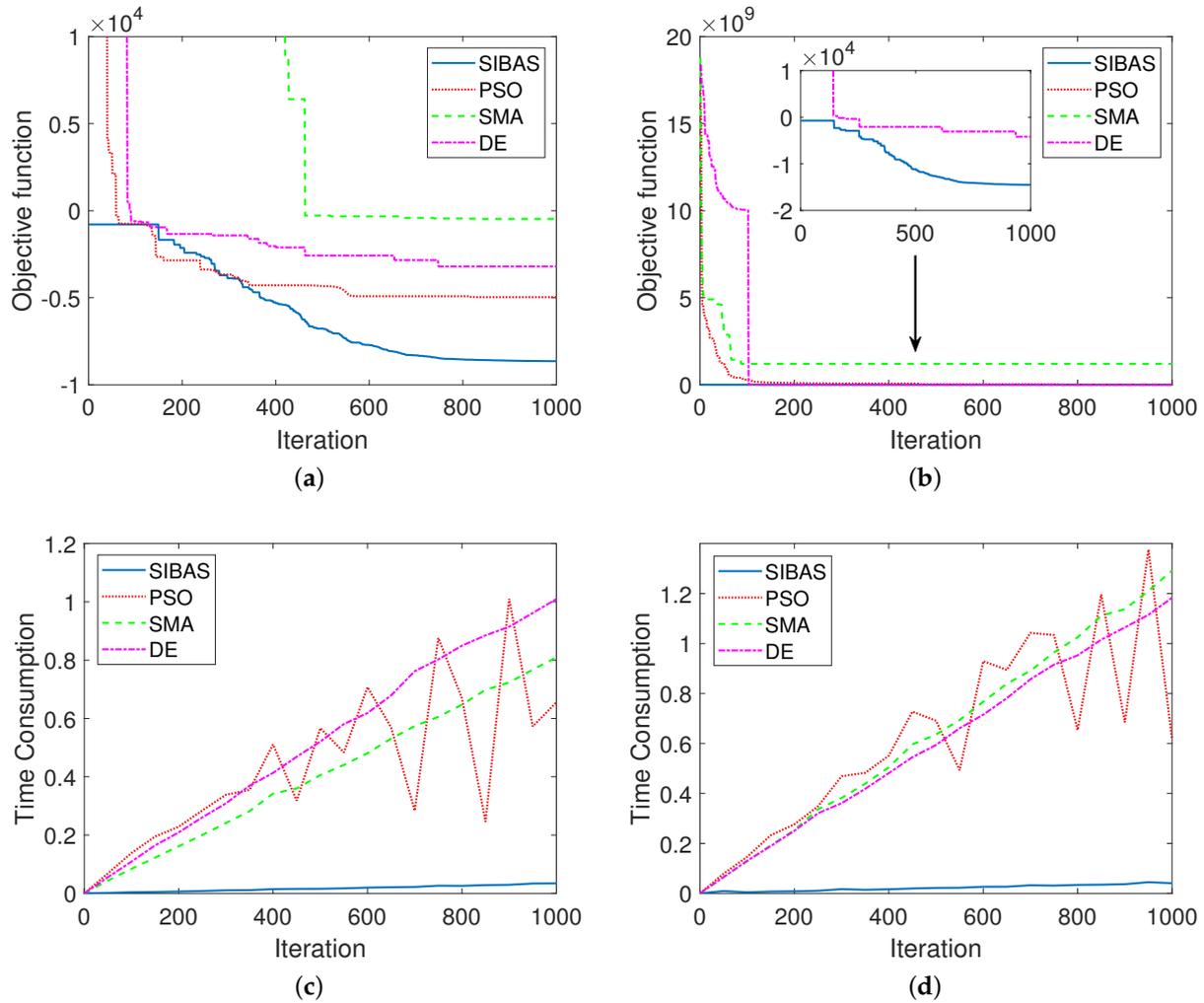
On the one hand, Figure 2a,g depict the SR of the portfolios under a market containing 40 and 80 stocks, respectively. Therein, it can be observed that the SR produced by the optimal portfolio of SIBAS is always higher than the optimal portfolios produced by PSO, SMA, and DE. Figure 2b,h show the TC, where it is observable that SIBAS optimal portfolios have always lower TC compared to PSO, SMA, and DE. The TC of the PSO, SMA, and DE optimal portfolios are similar in portfolio case 1, however they are not in portfolio case 2. Figure 2c,i show the average SR during the time period for the portfolio cases 1 and 2, respectively. According to these figures, SIBAS optimal portfolios produce the highest SR during the specific time period in both portfolio cases, while DE optimal portfolios produce the second highest SR and SMA optimal portfolios produce the lowest SR. Figure 2d,j show the average TC during the time period for the portfolio cases 1 and 2, respectively. Based on these figures, SIBAS optimal portfolios produce the lowest TC during the specific time period in both portfolio cases, while DE optimal portfolios produce the second lowest TC and SMA optimal portfolios produce the highest TC. Figure 2e,k show the total assets owned from the optimal portfolios produced by SIBAS, PSO, SMA, and DE during the time period along with the cardinality number  $K$  for the portfolio cases 1 and 2, respectively. Therein, it is observable that all portfolios always owns  $K$  in number assets and, hence, the CC is satisfied in both portfolio cases. Figure 2f,l show the sum of the optimal portfolios assets weights, which is the left part of (10), produced by SIBAS, PSO, SMA, and DE during the time period for the portfolio cases 1 and 2, respectively, along with the equality constraint (EC) number of (10), which is equal to 1. Therein, it is observable that the outcome of the SIBAS optimal portfolios always have the least noise and are closest to 1 in both portfolio cases. That is, SIBAS produces the best outcome in both portfolio cases, while SMA produces the second best outcome and DE the worst in portfolio case 1, and DE produces the second best outcome and SMA the worst in portfolio case 2.

Figure 3a,b present the SIBAS, PSO, SMA, and DE convergence in the portfolio cases 1 and 2, respectively, for  $t = 1$ , while the corresponding time consumption of SIBAS, PSO, SMA, and D at each iteration is presented in Figure 3c,d, respectively. That is, the value of (9) at each iteration of the SIBAS, PSO, SMA, and DE when the time-period for solving the TV-TPNC problem is 3 February 2020 is depicted in Figure 3a,b. In Figure 3a,b, we observe that SIBAS has the best convergence in both portfolio cases, whereas SMA has the worst, with PSO having the second best convergence in portfolio case 1 and DE having the second best convergence in portfolio case 2. In Figure 3c,d, we observe that SIBAS has the lowest time consumption in both portfolio cases, whereas DE has the highest time consumption in portfolio case 1 and SMA has the highest time consumption in portfolio case 2. Furthermore, the time consumption of PSO is non-linear and quite noisy in both portfolio cases. However, PSO has the second lowest time consumption in both portfolio cases at iteration 1000. SIBAS has the lowest computational complexity, as shown in Figure 3a,b, since it converges faster to the optimum solution in less iterations than PSO, SMA, and DE. Furthermore, SIBAS has the lowest time complexity, as shown in Figure 3c,d, since it takes less time to complete an iteration than PSO, SMA, and DE. As a result, SIBAS outperforms PSO, SMA, and DE when it comes to solving the TV-TPNC problem.

The average time consumption demanded from SIBAS, PSO, SMA, and DE to generate the optimal solutions for the TV-TPNC problem in both portfolio cases, on the other hand, is contained in Table 1. It is evident from this that SIBAS is always the quickest algorithm. More particularly, SIBAS is about 5 times faster in portfolio case 1 than the second fastest PSO, and more than 10 times faster in portfolio case 2. Moreover, SIBAS is about 28 times faster than the third faster DE in both portfolio cases, while it is more than 40 times faster in portfolio case 1 than the slowest SMA, and about 60 times faster in portfolio case 2.

The above analysis leads to the conclusion that SIBAS performed admirably and effectively in resolving the TV-TPNC problem. According to Figures 2 and 3 and Table 1 results, the SIBA produces more effective optimal portfolios than the PSO, SMA, and DE, whereas SMA produces the least effective ones. When contrasted to PSO, SMA, and DE,

the average time consumption of SIBAS is the shortest, whereas as the market dimension grows, its accuracy falls less than that of PSO, SMA, and DE. This implies that market size has a significant impact on SIBAS, PSO, SMA, and DE performance.



**Figure 3.** The SIBAS, PSO, SMA, and DE convergence and time consumption in the two portfolio cases for  $t = 1$ . (a) Algorithms’ convergence in portfolio case 1; (b) Algorithms’ convergence in portfolio case 2; (c) Algorithms’ time consumption in portfolio case 1; (d) Algorithms’ time consumption in portfolio case 2.

**Table 1.** Applications average time consumption.

Portfolio	SIBAS	PSO	SMA	DE
Case 1 (40 Stocks)	4.2 s	19.8 s	173.8 s	118.9 s
Case 2 (80 Stocks)	5.3 s	56.4 s	320.5 s	144.9 s

#### 4.2. MATLAB Repository

The whole design and implementation of the computational approaches suggested in this paper may be seen on GitHub: <https://github.com/SDMourtas/TV-TPNC> (accessed on 5 January 2021).

There, we created a MATLAB repository for solving the TV-TPNC problem inline with Algorithm 1. The MATLAB repository includes thorough installation instructions along with a meticulous implementation of the real-world data applications mentioned in Section 4. Furthermore, anyone may draw conclusions from their own findings by providing the repository’s main MATLAB function with their own data and adjusting the parameter values. Notice that the MATLAB repository’s data comes from Yahoo Finance

(<https://finance.yahoo.com/>) (accessed on 9 November 2021) and contains some of the market's most active stocks daily close prices.

## 5. Conclusions

The TV-TPNC problem is introduced in this paper as a NLP financial problem. The SIBAS algorithm for solving cardinality constrained NLP problems is introduced and then it is employed to solve the TV-TPNC problem. SIBAS effectiveness and accuracy have been demonstrated in two applications in different and somewhat large portfolio setups. In addition, SIBAS was compared to PSO, SMA, and DE, which are all popular meta-heuristics procedures. Based on our applications, we concluded that the SIBAS approach gives such a solution to the TV-TPNC problem, making it a very competitive option to PSO, SMA, and DE. The applications' findings reveal that the proposed procedure is accurate in two market configurations based on real-world data.

Some potential research areas can be identified.

1. The SIBAS could be compared to other popular meta-heuristics approaches in larger portfolios and other financial portfolio optimization problems.
2. The use of SIBAS in constraint optimization problems in different scientific domains.

**Author Contributions:** V.N.K., Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing—Original Draft; S.D.M., Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Writing—Original Draft. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used in the paper entitled "Diversification of Time-Varying Tangency Portfolio under Nonlinear Constraints through Semi-Integer Beetle Antennae Search Algorithm", are obtained from the Yahoo Finance (<https://finance.yahoo.com/>) (accessed on 9 November 2021). The complete development and implementation of the computational methods proposed can be obtained from GitHub in the following link: <https://github.com/SDMourtas/TV-TPNC> (accessed on 5 January 2021).

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

## References

1. Simos, T.E.; Mourtas, S.D.; Katsikis, V.N. Time-varying Black–Litterman portfolio optimization using a bio-inspired approach and neuronets. *Appl. Soft Comput.* **2021**, *112*, 107767. [[CrossRef](#)]
2. Katsikis, V.N.; Mourtas, S.D. *Computational Management; Modeling and Optimization in Science and Technologies*, Chapter Portfolio Insurance and Intelligent Algorithms; Springer: Cham, Switzerland, 2021; Volume 18, pp. 305–323. [[CrossRef](#)]
3. 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). *Oper. Res. Forum* **2021**, *2*, 18. [[CrossRef](#)]
4. Mourtas, S.D.; Katsikis, V.N. V-Shaped BAS: Applications on Large Portfolios Selection Problem. *Comput. Econ.* **2021**. [[CrossRef](#)]
5. Ye, K.; Parpas, P.; Rustem, B. Robust portfolio optimization: a conic programming approach. *Comp. Opt. Appl.* **2012**, *52*, 463–481. [[CrossRef](#)]
6. Konno, H.; Akishino, K.; Yamamoto, R. Optimization of a Long-Short Portfolio under Nonconvex Transaction Cost. *Comp. Opt. Appl.* **2005**, *32*, 115–132. [[CrossRef](#)]
7. Ogryczak, W.; Sliwinski, T. On solving the dual for portfolio selection by optimizing Conditional Value at Risk. *Comp. Opt. Appl.* **2011**, *50*, 591–595. [[CrossRef](#)]
8. Katsikis, V.N.; Mourtas, S.D. A heuristic process on the existence of positive bases with applications to minimum-cost portfolio insurance in  $C[a, b]$ . *Appl. Math. Comput.* **2019**, *349*, 221–244. [[CrossRef](#)]
9. Katsikis, V.N.; Mourtas, S.D. ORPIT: A Matlab Toolbox for Option Replication and Portfolio Insurance in Incomplete Markets. *Comput. Econ.* **2019**, *56*, 711–721. [[CrossRef](#)]
10. Jiang, X.; Li, S. BAS: Beetle Antennae Search Algorithm for Optimization Problems. *arXiv* **2017**, arXiv:1710.10724.
11. Medvedeva, M.A.; Katsikis, V.N.; Mourtas, S.D.; Simos, T.E. Randomized time-varying knapsack problems via binary beetle antennae search algorithm: Emphasis on applications in portfolio insurance. *Math. Methods Appl. Sci.* **2020**, *44*, 2002–2012. doi:10.1002/mma.6904. [[CrossRef](#)]
12. Khan, A.H.; Cao, X.; Katsikis, V.N.; Stanimirovic, P.; Brajevic, 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](#)]

13. Katsikis, V.N.; Mourtas, S.D.; Stanimirović, P.S.; Li, S.; Cao, X. Time-varying minimum-cost portfolio insurance under transaction costs problem via Beetle Antennae Search Algorithm (BAS). *Appl. Math. Comput.* **2020**, *385*, 125453. [[CrossRef](#)]
14. 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.* **2021** *64*, 152204. [[CrossRef](#)]
15. Wu, Q.; Ma, Z.; Xu, G.; Li, S.; Chen, D. A Novel Neural Network Classifier Using Beetle Antennae Search Algorithm for Pattern Classification. *IEEE Access* **2019**, *7*, 64686–64696. [[CrossRef](#)]
16. 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](#)]
17. Fan, Y.; Shao, J.; Sun, G. Optimized PID controller based on beetle antennae search algorithm for electro-hydraulic position servo control system. *Sensors* **2019**, *19*, 2727. [[CrossRef](#)]
18. Yue, Z.; Li, G.; Jiang, X.; Li, S.; Cheng, J.; Ren, P. A Hardware Descriptive Approach to Beetle Antennae Search. *IEEE Access* **2020**, *8*, 89059–89070. [[CrossRef](#)]
19. Chen, D.; Li, X.; Li, S. A Novel Convolutional Neural Network Model Based on Beetle Antennae Search Optimization Algorithm for Computerized Tomography Diagnosis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–12. [[CrossRef](#)]
20. Tobin, J. Liquidity Preference as Behavior Towards Risk. *Rev. Econ. Stud.* **1958**, *25*, 65–86. [[CrossRef](#)]
21. Maringer, D.G. *Portfolio Management with Heuristic Optimization*, 1st ed.; Advances in Computational Management Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 8. [[CrossRef](#)]
22. Jansen, R.; van Dijk, R. Optimal Benchmark Tracking with Small Portfolios. *J. Portf. Manag.* **2002**, *28*, 33–39. [[CrossRef](#)]
23. Lobo, M.S.; Fazel, M.; Boyd, S. Portfolio optimization with linear and fixed transaction costs. *Ann. Oper. Res.* **2007**, *152*, 341–365. [[CrossRef](#)]
24. Deb, K. *Optimization for Engineering Design: Algorithms and Examples*, 2nd ed.; PHI : Delhi, New Delhi, India, 2013.
25. Yang, X.S. *Nature-Inspired Optimization Algorithms*, 1st ed.; Elsevier Insights, Elsevier: Amsterdam, The Netherlands, 2014.
26. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]