axioms

MDPI

Article

SAIPO-TAIPO and Genetic Algorithms for Investment Portfolios

Juan Frausto Solis *, José L. Purata Aldaz, Manuel González del Angel, Javier González Barbosa and Guadalupe Castilla Valdez

Graduate Program Division, Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero, Ciudad Madero 89440, Mexico; luis.pa@cdmadero.tecnm.mx (J.L.P.A.); manuel.delangel1139@gmail.com (M.G.d.A.); jjgonzalezbarbosa@hotmail.com (J.G.B.); gpe_cas@yahoo.com.mx (G.C.V.)

* Correspondence: juan.frausto@itcm.edu.mx

Abstract: The classic model of Markowitz for designing investment portfolios is an optimization problem with two objectives: maximize returns and minimize risk. Various alternatives and improvements have been proposed by different authors, who have contributed to the theory of portfolio selection. One of the most important contributions is the Sharpe Ratio, which allows comparison of the expected return of portfolios. Another important concept for investors is diversification, measured through the average correlation. In this measure, a high correlation indicates a low level of diversification, while a low correlation represents a high degree of diversification. In this work, three algorithms developed to solve the portfolio problem are presented. These algorithms used the Sharpe Ratio as the main metric to solve the problem of the aforementioned two objectives into only one objective: maximization of the Sharpe Ratio. The first, GENPO, used a Genetic Algorithm (GA). In contrast, the second and third algorithms, SAIPO and TAIPO used Simulated Annealing and Threshold Accepting algorithms, respectively. We tested these algorithms using datasets taken from the Mexican Stock Exchange. The findings were compared with other mathematical models of related works, and obtained the best results with the proposed algorithms.

Keywords: Investment Portfolio Optimization; genetic algorithm; Simulated Annealing and Threshold Accepting; Sharpe Ratio; Markowitz model

check for updates

Citation: Frausto Solis, J.; Purata Aldaz, J.L.; González del Angel, M.; González Barbosa, J.; Castilla Valdez, G. SAIPO-TAIPO and Genetic Algorithms for Investment Portfolios. *Axioms* **2022**, *11*, 42. https://doi.org/10.3390/axioms11020042

Academic Editors: Witold Pedrycz, Laura Cruz-Reyes and Rafael Alejandro Espin Andrade

Received: 1 December 2021
Accepted: 17 January 2022
Published: 21 January 2022

1. Introduction

In finance, applying the diversification of assets that make up an investment portfolio aims to maximize profits and minimize risk. The mean-variance portfolio integration model developed by Harry Markowitz in 1952 has been a widely accepted tool for asset portfolio integration [1]. Markowitz's portfolio theory states that the investor should analyze the portfolio as a whole, studying the characteristics of risk and global return. The participation of each asset is chosen based on its expected return. In other words, volatility is treated as a risk factor, and the portfolio is integrated, considering the risks and seeking the maximum level of profitability available [2]. Some investigations have expanded the work of the Markowitz model, such as the mean semi-variance model [3], mean objective model [4], and portfolio optimization models with fuzzy logic [5]. Different metaheuristics algorithms have been applied to solve difficult problems, and recently important overviews have been published regarding these types of problems [6–8]. The reason for employing these algorithms is that they obtain suitable solutions within reasonable execution times [9]. For instance, a two-stage multi-attribute portfolio analysis framework using genetic algorithms (GA) to solve a multi-attribute portfolio selection problem was proposed [10]. Moreover, Genetic Algorithms have been used for selecting and evaluating investment portfolios [11]. As seen in [12], the mean-variance approach was used as a reference to propose a new model which included different constraints, using GA as an optimization method. Chen et al., applied machine learning algorithms to predict asset values of the Shanghai stock market; they used a GA to solve a multi-objective optimization problem [13]. Pekar et al., used differential evolution (DE) to solve the portfolio selection problem [14]; where they employed the Omega function and Sortino ratio to measure the portfolio's risk based on the Sharpe Ratio. Presently, DE algorithms have been applied in several portfolio applications: Dow Jones [14], S&P, BOFA, Frank Russell, MSCI [15], and National Stock Exchange [16]. The use of DE combined with other methods has increased the quality of results [15,17]. Notably, in a previous study [15], the following algorithms were evaluated: NSGA-II, Differential Evolution (GDE3), SMPSO, and SPEA2; where SPEA and Differential Evolution were reported as the best algorithms. A modified Particle Swarm Optimization (PSO) method is used in different indexes such as S&P 100, Hang Seng, DAX 100, FTSE 100, and Nikkei [18]; the mathematical model includes bounding and cardinality constraints. A Firefly Algorithm (FA) and an Imperialist Competitive Algorithm (ICA) are used to select the best set of assets using the mean semi-variance approach as the objective function [19]. In [20] a hybrid algorithm was proposed that combines Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) optimization, and GA for solving the cardinality constrained portfolio optimization problem.

Different classical and hybrid methods or techniques for portfolio integration have been applied, such as Threshold Accepting (TA) and Simulated Annealing (SA). TA was used to select sets of assets that minimized portfolio risks [21,22] and optimized tracking errors [23]. Chang et al., [24] attempted to find the efficient frontier by using GA, TA, and SA over several stock markets (DAX, FTSE, S&P, and Nikkei). Fogarasi and Levendovszky [25] explored a solution to sparse mean reverting portfolio selection using a Greedy method and an SA algorithm. In [26], the following algorithms were used: GA, Genetic network programming (GNP), SA, and PSO. In [27], the quality of SA and GA in rectifying this problem was performed by adding an aversion risk coefficient to the Markowitz model. The results were similar; however, SA obtained shorter runtimes. Wang et al., used a modified Generalized Simulated Annealing GSA and the Sharpe Ratio to construct optimal portfolios; they used stocks from S&P 500 [28]. A High-Speed Hill-Climbing algorithm, called HC-S-R, was proposed for the DAX stock exchange in [29]; where they applied a similar acceptance criterion to TA and reported better results than the classical algorithm. In [30], the Archive Multi-objective Simulated Annealing (AMOSA) was used to solve the two objectives of the mean-variance model with eight stock indexes that included NASDAQ, S&P 500, FTSE 100, and Hang Seng. In [31], a hybrid FA–SA algorithm was proposed; the experimental results showed this algorithm performed better than FA, SA, GA, and PSO when the transaction costs were considered. Kumar et al., applied a modified SA combined with ABC, Radial Basis Function Network (RBFN), and an Artificial Neural Network (ANN); where they integrated portfolios with stocks from NSE [32]. SA, GA, and Quantum Annealing was applied in [33], with stocks from S&P 500, Nasdaq, Russel 2000, and Wilshire 5000.

The algorithms GENPO, SAIPO, and TAIPO proposed in this work are based on the Genetic Algorithm (GA), Simulated Annealing (SA), and Threshold Accepting (TA). These proposed algorithms use the closing prices of the assets to calculate the indicators to obtain the Sharpe Ratio (SR) [34] value of the portfolio, which is used as an evaluation function in the search process of the algorithms. The algorithms used are focused on portfolio integration and aimed at both diversification of assets and minimization of portfolio risks; this was achieved through the model proposed by Sharpe [35] and is standard in using the correlation between candidate assets [36,37]. In addition, a constraint was used that allowed for the consideration of only the assets that had an expected return equal to or greater than the Minimum Acceptable Rate of Return (MARR). To test the proposed algorithms, assets from the Mexican Stock Exchange were used, then results were compared with the hybrid algorithms developed by the mathematical models of related works.

The rest of this paper is organized as follows. In Section 2, we review the related models, including the Markowitz model. Moreover, we describe the classical Simulated

Annealing, Threshold Accepting, and Genetic Algorithms. In Section 3, the algorithms proposed in this work are described. In Section 4, we present the experiments and results. Finally, in Section 5, the conclusions of this work are presented.

2. Background & Related Works

As we mentioned before, this work proposes the use of classical algorithms GA, TA, and SA. We present a general description of these algorithms in the first subsection. Moreover, new successful models are usually based on the Markowitz model; they are referred to here as Yu, Gilli, and Masese models. Then, we briefly present all these models in the Related Works subsection.

2.1. Classical Genetic and Simulated Annealing like Algorithms

2.1.1. Genetic Algorithms

John Holland proposed Genetic Algorithms (GA) in 1975 [38]. These algorithms imitate nature for solving complex problems [39–41]. They select solutions from the solution space of the problems considered. Basically, a GA evolves a population of individuals (solutions) by selecting operations similar to those in biological evolution (for instance, mutation and crossing over). The procedure of the Classic GA is shown in Algorithm 1.

Algorithm 1. Classic Genetic Algorithm

```
Set first gen $i = 0$
Begin /* Produce new generation */
  Begin /* Reproductive cycle */
    Select two individuals, $X_1$ y $X_2$ of P(i),
    Crossing point random selection.
    Cross $X_1$ y $X_2$ getting two offspring $X_{h1}$, $X_{h2}$.
    Insertion $X_{h1}$ y $X_{h2}$ in P(i)
    Mutation random $\tau$ elements of P(i);
    Compute fitness function (i) of $\tau$ elements,
    Order individuals best evaluated in P(i).
    Limit the population to its original size.
  End
  IF convergence = true or gen = max generation
End
```

2.1.2. Simulated Annealing and Threshold Accepting

Proposed by Kirkpatrick in the 1980s [42], the Simulated Annealing (SA) algorithm is based on the Metropolis algorithm [43], which is used in the heat treatment of metals. SA begins with an initial solution; then, a neighbor solution is selected randomly. A new solution is accepted if it is better than the old solution; otherwise, the new solution is accepted with a probability based on the Boltzmann distribution; with this criterion, the acceptance rate of incorrect solutions decreases during the execution of the algorithm. This strategy is used to escape from local optima. The parameters of this algorithm include an initial temperature T_i, final temperature T_f, cooling rate α, and the number of iterations for each metropolis cycle, as shown in Algorithm 2.

The Threshold Accepting (TA) algorithm was proposed by Dueck and Scheuer [44] and is similar to Simulated Annealing. The fundamental difference is the acceptance criterion of new solutions. In TA, a poor solution can be accepted if the decrease does not exceed a certain tolerance or threshold, which decreases during the execution of the algorithm. This criterion avoids calculating probabilities or making random decisions. The parameters of this algorithm are the number of iterations n_{iter}, number of steps n_{steps}, and a threshold sequence, as shown in Algorithm 3. Both SA and TA algorithms have been applied to successfully solve different problems [45,46].

Algorithm 2. Classic Simulated Annealing

```
Simulated Annealing $\left(T_i, T_f, \alpha\right)$
    $T_k = T_i$
    $s_{current} = random\ initial\ solution$
    while $T_k \geq T_f$ do
        while $Metropolis\ lenght$ do
            $s_{new} = generateNeighbor\ (s_{current})$
            $\Delta = E(s_{new}) - E(s_{current})$
            if $\Delta < 0$ then
                $s_{current} = s_{new}$
            else if $e^{\frac{-\Delta}{T_k}} < random(0,1)$ then
                $s_{current} = s_{new}$
            end if
        end while
        $T_k = \alpha * T_k$
    end while
end Simulated Annealing
```

Algorithm 3. Classic Threshold Accepting

```
Initialize $n_{iter}$ and $n_{steps}$
Compute Threshold Sequence $T_r,\ r = 1,2,\ldots n_{iter}$
Generate random initial solution $X_c \in X$
for $r = 1$ to $n_{iter}$ do
    for $i = 1$ to $n_{steps}$ do
        Generate $X_n \in N(X_c)$
        $\Delta = f(X_n) - f(X_c)$
        if $\Delta < T_r$ then
            $X_c = X_n$
        end if
    end for
end for
```

The process of parameter tuning for the SA and TA algorithms used in this paper is based on the analytical tuning method presented in [47]. This method states that the initial and final temperatures are functions of the maximum and minimum cost or energy values E_{max} and E_{min}. These values are used in the Boltzmann distribution criterion, which establishes that, in a temperature T, a poor solution is accepted if $random(0,\ 1) \leq e^{-\frac{\Delta E}{T}}$. The values ΔE_{max} and ΔE_{min} are used in the Boltzmann distribution for determining the initial and final temperatures. To obtain these values, a set of previous executions must be carried out [46]. The parameters used in the tuning process are shown in Table 1.

The probability of accepting a new solution is close to one at high temperatures, thus the deterioration of the energy is at a maximum. T_i is associated with ΔE_{max} and $\mathrm{P}(\Delta E_{max})$, and can be calculated with Equation (1), while $\mathrm{P}(\Delta E_{max})$ is given by Equation (2). The minimum deterioration $\mathrm{P}(\Delta E_{min}\ )$, calculated with Equation (3), is used to establish T_f, as shown in Equation (4).

The cooling scheme establishes the method behind decreases in temperature using a factor α. For rapid decrements, an α value equal to 0.7 is commonly used, and for slow decrements, the figure is 0.99 [46]. Given a current temperature T_k, the next temperature value T_{k+1} is calculated with Equation (5).

The parameter L_k, refers to the number of iterations of the Metropolis cycle at each temperature. In high temperatures, few iterations are required, since the equilibrium is reached quickly; for this reason, L_{min} is usually close to one. Nevertheless, at low temperatures, a more exhaustive search is required; therefore, a larger L_k is used. For a given value of L_k, then L_{k+1} can be calculated with Equation (6). Since Equations (5) and (6) are applied successively from T_i to T_f, L_{max} is calculated as $L_{max} = \beta^n L_{min}$, while $T_f = \alpha^n T_i$, where n and β are calculated with Equations (7) and (8), respectively [48].

Table 1. Parameters used for SA and TA Analytical Tuning.

Parameter	Description	Equation
$s_i,\ s_j$	Current and candidate solution	-
$E(s_i),\ E\left(s_j\right)$	Energy associated with s_i and s_j	-
$\Delta E_{max},\ \Delta E_{min}$	Maximum and minimum deterioration	-
T_i	Initial temperature	$T_i = \frac{-\Delta E_{max}}{ln(P(\Delta E_{max}\ ))}$ (1)
$\mathrm{P}(\Delta E_{max}),\ \mathrm{P}(\Delta E_{min})$	Probability of accepting a solution with maximum and minimum deterioration	$P(\Delta E_{max}\ ) = e^{\frac{-\Delta E_{\mathrm{max}}}{T_i}}$ (2) $P(\Delta E_{min}\ ) = e^{\frac{-\Delta E_{\mathrm{min}}}{T_f}}$ (3)
T_f	Final temperature	$T_f = \frac{-\Delta E_{min}}{ln(P(\Delta E_{min}))}$ (4)
α	Temperature decrease factor	-
T_k	Current temperature	$T_{k+1} = \alpha T_k$ (5)
L_k	Length of the Markov chain in Metropolis cycle	$L_{k+1} = \beta L_k$ (6)
L_{min}	Value of L_k at T_i	-
n	Number of steps from T_i to T_f	$n = \frac{ln(T_f)-ln(T_i)}{ln(\alpha)}$ (7)
β	Increment coefficient of L_k	$\beta = e^{\frac{ln(L_{\mathrm{max}})-ln(L_{\mathrm{min}})}{n}}$ (8)
V_{s_i}	Neighborhood of the solution s_i	-
$P\left(s_j\right)$	Probability of selecting s_j	$P\left(s_j\right) = 1 - e^{-\frac{N}{\lvert V_{si}\rvert}}$ (9)
N	Number of random samples in V_{Si}	$N = -\lvert V_{si}\rvert ln\left(1 - P\left(s_j\right)\right) = C\lvert V_{si}\rvert$ (10)
C	Exploration level	$C = ln\left(P\left(s_j\right)\right)$ (11)
L_{max}	Value of L_k at T_f	$L_{max} = C\lvert V_{si}\rvert$ (12)

On the other hand, the probability of selecting the solution s_j from N random samples in V_{Si} is given by Equation (9); for this expression, N is calculated with Equation (10), where C defines the exploration level calculated with Equation (11). To guarantee a good exploration level, the value of C must be between $1 \leq C \leq 4.6$. Finally, once C is defined, L_{max} can be established with Equation (12).

2.2. Related Works

2.2.1. Markowitz Model

A typical feature of Markowitz's theory is that it provides a quantitative solution to portfolio asset allocation, as it considers the possible trade-off between expected return and risk exposure between established securities or assets. Thus, the investment portfolio design problem is a multi-objective optimization issue with two main objectives [1]: maximize the expected return, and minimize the variance of the portfolio models with Equations (13) and (14), respectively. This model uses a mean-risk relationship, which is known as the mean-variance approach.

$$Max\ E(R) = \sum_{i=1}^{n} \mu_i x_i \tag{13}$$

$$Min\ \sigma^2(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i\ x_j\sigma_{ij} \tag{14}$$

$$\text{Subject to :}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{15}$$

$$x_i\ \geq 0(i = 1, \ldots, n) \tag{16}$$

where $E(R)$ is the expected return of the portfolio; μ_i is the expected return of asset i; x_i is the fraction or weight of asset i in the portfolio, this weight must be positive (Equation (16)), in addition, the sum of all weights must be equal to one as shown in Equation (15); n is the number of assets in the portfolio; $\sigma^2(R)$ is the variance of the portfolio; σ_{ij} is the covariance between asset i and j.

2.2.2. Yu Model

A genetic algorithm (GA) was proposed [10] for a two-stage multi-attribute portfolio analysis framework to solve the portfolio selection problem. First, a GA is used for evaluating asset quality with multiple asset attributes, and some are selected. Then, another GA is used to optimize capital allocation among the selected assets in the second stage. Finally, the fitness function $F(R)$ is used to make a rational trade-off between minimizing risk and maximizing expected return, as shown in Equation (17).

$$\min F(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i\ x_j\sigma_{ij} + \left(\sum_{i=1}^{n} \mu_i x_i - \gamma\right)^2 \tag{17}$$

where x_i, x_j are the portion of the stocks i, j integrating the portfolio, and σ_{ij}; is the covariance of them. Moreover, μ_i is the mean of the stock value and $\sum\limits_{i=1}^{n} \mu_i x_i$ is the expected return of the portfolio. Finally, γ represents the MARR.

In another section, we present the results obtained with this model.

2.2.3. Gilli Model

This model uses a single-objective minimization approach based on the classical Markowitz model. The Gilli model seeks to create optimal portfolios in terms of variance and expected return [21]. The mathematical model is shown in Equations (18) and (19).

$$\min F(R) = \sigma^2(R) + p(\gamma - E(R)) \tag{18}$$

$$p = \begin{cases} \frac{\sigma^2(R)_{max} - \sigma^2(R)_{min}}{\gamma - \overline{R}} & when\ \ \gamma > E(R) \\ 0 & \text{Otherwise} \end{cases} \tag{19}$$

where, the variables $\sigma^2(R)$ and $E(R)$ denote the variance and the expected return of the portfolio, respectively, while p is a penalty function. The values for $\sigma^2(R)\ _{max}$, $\sigma^2(R)\ _{min}$ are the maximum and minimum variance, and $\overline{R}$ is an average of the expected return value of numerous randomly generated portfolios. While γ is the required return and has been pre-defined. These parameters are used for defining the scaling constant p. The problem in this model is to define this scaling constant for achieving the best portfolios. Therefore, this model uses the scaling constant for achieving optimal portfolios.

2.2.4. Masese Model

This model is similar to the classical formulation of Markowitz for investment portfolios, and is an optimization problem defined by Equation (20). This model aims to minimize the variance of the portfolio [22]; they use the classical Threshold Accepting algorithm [44].

$$Min\ \sigma^2(R) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i\, x_j \sigma_{ij} \tag{20}$$

Subject to

$$x_i{}^{inf} \leq\ x_i \leq\ x_i^{sup} \tag{21}$$

$$n_{inf} \leq n_{sup} \tag{22}$$

where $x_i{}^{inf}$ and x_i^{sup}, shown in Equation (21), are the minimum and maximum weights for the stock in the portfolio, and n_{inf} and n_{sup} in Equation (22), are the constraints setting the minimum and a maximum number of stocks in the portfolio. The authors of this paper reported that their model outperforms the Markowitz Model for portfolio selection.

3. Proposed Algorithms

The three algorithms proposed in this work, GENPO [49], SAIPO, and TAIPO, are described in this section. These algorithms use optimization heuristics for obtaining the stocks and then integrating the portfolio that is modeled as an optimization problem. GENPO uses a Genetic Algorithm, SAIPO uses a Simulated Annealing Algorithm, and TAIPO uses a Threshold Accepting Algorithm.

3.1. Mathematical Model for the Evaluation Function

The Sharpe Ratio (SR) is a financial metric proposed by W. Sharpe [34]. This financial ratio indicates how suitable an investment is concerning its risk. This ratio can be used to determine which investment obtains the greatest return with the same risk. In other words, it determines the additional return achieved by investing in riskier financial assets. SR is calculated with Equation (23).

$$SR = \frac{E(R) - \gamma}{\sigma(R)} \tag{23}$$

where $E(R)$ is the portfolio's expected return, γ is the $MARR$ (Minimum Acceptable Rate of Return), and $\sigma(R)$ represents the portfolio's risk. The $MARR$ parameter means that any stock (or investment) should not be part of the portfolio if it does not surpass it.

Sharpe Ratio (SR) is based on the Markowitz model, and it is assumed that there are adequate statistics for determining: (a) the expected value of the shares and (b) the standard deviation (the risk measure) of the asset value over a given period. On the other hand, the problem of selecting the best stocks from a specific market can be formulated as those that maximize the expected return and, at the same time, minimize the risk value of the portfolio; a practical alternative centers on finding stocks which maximize the Sharpe ratio. This strategy allows the discovery of portfolios with assets from different financial areas selected when the SR is maximized.

The proposed model represented by Equation (24) seeks to maximize the Sharpe Ratio, subject to the expected value of the portfolio being greater or equal to the MARR ratio.

$$\max SR = \frac{E(R) - \gamma}{\sqrt{\sum_{i=1}^{n} \sigma_i{}^2 x_i{}^2 + 2\sum_{i=1}^{n}\sum_{j\neq i}^{n} \sigma_i \sigma_j x_i x_j \rho_{ij}}} \tag{24}$$

Subject to

$$\mu_i \geq \gamma \tag{25}$$

where $E(R) = \mu_i x_i$ is the expected return of the portfolio, μ_i is the expected return of asset i, and x_i is the weight of the asset i in the portfolio. This model includes a constraint shown in Equation (25), that allows only assets with an expected return equal to or greater than the MARR to be included in the portfolio. The MARR parameter is represented here by γ. As mentioned before [49], maximizing the value of the Sharpe Ratio allows solving the problem of two objectives in one. However, it is an NP-hard problem; thus, finding the optimal solution is not an easy task. We used Equation (26) to estimate the risk of the portfolio, while the average correlation of the portfolio was obtained with a simple process [37]. This correlation is provided in Equation (27), and we used it to check that the correlation of the entire portfolio had a low value. In [50], the correlation between assets and bonds was analyzed. Similarly, a previous study [37] used a strategy to evaluate portfolios and risks from different portfolios.

$$\sigma(R) = \sqrt{\sum_{i=1}^{n} {\sigma_i}^2 {x_i}^2 + 2\sum_{i=1}^{n}\sum_{j\neq i}^{n} \sigma_i \sigma_j x_i x_j \rho_{ij}} \tag{26}$$

$$\left|\overline{\rho}_p\right| = \frac{\sigma^2(R) - \sum_{i=1}^{n} {\sigma_i}^2 {x_i}^2}{2\sum_{i=1}^{n}\sum_{j\neq i}^{n} \sigma_i \sigma_j x_i x_j} \tag{27}$$

The interpretation of Equation (27) is as follows: if $\left|\overline{\rho}_p\right| \cong 1$ the portfolio has little diversification; on the other hand, if $\left|\overline{\rho}_p\right| \cong 0$ the portfolio is highly diversified.

The portfolio solution found with the model implemented in this paper, contained the weighted values of the assets x_i; the expected value portfolio's $E(R)$ and its risk $\sigma(R)$, in addition to the Sharpe Ratio and average correlation of the portfolio $\left|\overline{\rho}_p\right|$. This model and the implemented algorithms aim to find a balance between the expected value and the portfolio's risk. Moreover, the implemented algorithms seek to find an equilibrium between the Sharpe metric and the managing portfolio risk.

3.2. Genetic Portfolio Optimization Algorithm (GENPO)

Algorithm 4 shows the structure of the GENPO algorithm. This is a Genetic Algorithm where the initial population is conformed randomly by a set of weights in the range (0,1); additionally, the summation of these weights should be one. The fitness function was previously shown in Equation (24), and is applied for each population generated by the algorithm. Then, the subsequent generations are generated, choosing one or two solutions, depending on the genetic operator used: crossover or mutation, respectively.

From lines 1 to 5, a population is generated using a random number in the range (0,1).

The fitness Sharpe Ratio (SR) is the fitness function based on Sharpe Ratio described in Equation (24). Moreover, it establishes the population size and the generations.

Lines 6 to 16 integrate the selection step. Two individuals from the population are evaluated by the fitness function and two individuals from the binary tournament, evaluating from the higher expected value.

In the Crossover step, the x_i values (the portion of assets) are randomly selected to generate offspring from the selected individuals in the previous step. These offspring could be part of the population depending on their SR. They are sorted from highest to lowest. The last two individuals are discarded from the population to retain the population size. From line 17 to the end, lays up the best individual from each generation to store their variables and performance.

The best individual from generation i is compared with the previous individuals of generation $i - 1$. The process is repeated each time a new population is generated. This process is performed to determine if the number of generations achieves the maximum permitted number without enhancement, which is used as a stop criterion.

The result of the algorithm is a portfolio with the best assets and the investment proportion applied in each one. This combination of assets has the best profit–risk ratio and the lowest correlation, leading to the highest portfolio diversification.

Algorithm 4. GENPO

```
Begin /* GENPO */
  Set first gen $i = 0$
  Generate initial population (portfolio) $P(i)$
  Compute Sharpe Ratio (SR) evaluation initial for each individual SR(i)
  While Converge = False or i < MaxGen do
  Begin /* Produce new generation */
    Begin /* Reproductive cycle */
      Select two individuals (portfolios), $X_1$ y $X_2$ of P(i),
      Crossing point random selection.
      Cross $X_1$ y $X_2$ getting two offspring $X_{h1}$, $X_{h2}$.
      Insertion $X_{h1}$ y $X_{h2}$ in P(i)
      Mutation random $\tau$ elements of P(i);
      Compute fitness function (i) of $\tau$ elements,
      Order individuals best evaluated in P(i).
      Limit the population to its original size.
    End
    Save best individual of the population P(i)
    If SR(i + 1) < SR(i) + tolerance then //* if SR doesn't grow enough *//
      gamma = gamma + 1              //* gamma counter grow *//
    If gamma >= conv then //*conv=generations number without improv SR *//
      Convergence=TRUE
 End
 End while
  i = i + 1
If i > MaxGen then
Converge: = TRUE
End
```

3.3. Simulated Annealing for Investment Portfolio Optimization (SAIPO)

The SAIPO algorithm uses the negative value of the Sharpe Ratio, shown in Equation (24), as the objective function.

SAIPO (Algorithm 5) receives as initial parameters (line 1) an initial temperature (T_i), final temperature (T_f), a cooling rate α, an internal cycle length L_k, and an internal cycle increment coefficient β. In line 2, a random initial solution is generated, represented by X_c, defined as a portfolio with random weights. Then, in line 3, the negative value of the Sharpe Ratio is calculated. In line 4, the current temperature T_k takes the value of the initial temperature T_i. In line 5, the main cycle begins, which is executed until the current temperature is greater than or equal to the final temperature. In line 6, the Metropolis cycle begins. In line 7, a new solution is generated by applying a perturbation to the current solution. This new solution is compared with the existing solution, then calculating the difference between both solutions in line 8. When this difference is negative (line 9), the new solution is accepted in line 10; if this new solution is better than the best previously found, it is accepted as the best global solution in line 12. Alternatively, if the difference is greater or equal to zero, the new solution is accepted by applying the Boltzmann acceptance criterion in line 14. In line 19, the number of iterations of the Metropolis cycle (L_k) increases, multiplying its previous value by β. In line 20, the current temperature T_k is decreased by multiplying its value by the α value. Finally, in line 22, the Sharpe Ratio is calculated by multiplying its negative value by -1. Lastly, in line 23, the output of the algorithm is generated.

Algorithm 5. SAIPO

```
1:  Parameters ($T_i$, $T_f$, $\alpha$, $L_k$, $\beta$)
2:      Generate random initial solution $X_c = X_{best}$
3:      $s_c = f(X_n) = SRneg = s_{best}$
4:      $k = 0$; $T_i = T_k$
5:      while $T_k \geq T_f$ do
6:          while $k < L_k$ do
7:              $X_n = Perturbation\ (X_c)$; $s_n = f(X_n)$
8:              $\Delta = s_n - s_c$
9:              if $\Delta < 0$ then
10:                 $s_c = s_n$; $X_c = X_n$
11:                 if $s_c < s_{best}$ then
12:                     $s_{best} = s_c$; $X_{best} = X_c$
13:                 end if
14:             else if random (0,1)$< e^{-\Delta/T_k}$ then
15:                 $s_c = s_n$; $X_c = X_n$
16:             end if
17:             $k = k + 1$
18:         end while
19:         $L_{k+1} = \beta * L_k$
20:         $T_{k+1} = \alpha * T_k$
21:     end while
22:     $SR = -s_{best}$
23:     return $SR$, $X_{best}$
24: end SAIPO
```

Based on the neighborhood structure method presented in [23], in this work, a perturbation is applied to generate a neighbor solution X_n close to the current solution X_c. This procedure is detailed in Algorithm 6.

Algorithm 6. Perturbation Procedure

```
1:  Parameters ($q$, $X_c$)
2:      Select two assets $i$ and $j$ randomly
3:      If $wi \geq q$ then
4:          $wi = wi - q$
5:      else if $wi < q$ then
6:          $q = wi$
7:          $wi = 0$
8:      end if
9:      $wj = wj + q$
10:     $X_n \ni (wi,\ wj)$
```

In line 2, two assets i and j with weights w_i and w_j are randomly selected. Then, a quantity q is subtracted from asset i in line 4, and the same quantity is added to asset j in line 9. Therefore, the weight of asset i in the portfolio, when the neighbor solution is generated, will be $w_i - q$, and the weight of asset j will be $w_j + q$. This quantity q is obtained by experimentation.

3.4. Threshold Accepting for Investment Portfolio Optimization (TAIPO)

Algorithm 7 shows the TAIPO algorithm that shares the same structure as SAIPO. These two algorithms have a temperature cycle and a Metropolis cycle. TAIPO has an additional parameter in line 1: a decreasing tolerance rate δ. As in SAIPO, in the TAIPO algorithm, a perturbation is applied to the current solution to generate a new solution in line 7. However, in this algorithm, if the new solution is worse, the current solution is replaced if the difference does not exceed a certain tolerance (*Tol*) or threshold (line 9), which decreases by multiplying its previous value by δ in line 17. In the beginning, *Tol* has a value equal to T_k (line 4), which implies that at high temperatures, the new solution will be generally accepted as the current solution. That is, during the processing of 90% of temperatures, *Tol* has the same value of T_k; in this case, δ is equal to 1. For the remaining temperatures, δ takes the value of 0.96, obtained by experimentation, making the process more restrictive in the last iterations [46].

Algorithm 7. TAIPO

```
1:  Parameters ($T_i$, $T_f$, $\alpha$, $L_k$, $\beta$, $\delta$)
2:      Generate random initial solution $X_c = X_{best}$
3:      $s_c = f(X_n) = SRneg = s_{best}$
4:      $k = 0$; $Tol_k = T_i = T_k$
5:      while $T_k \geq T_f$ do
6:          while $k < L_k$ do
7:              $X_n = Perturbation\ (X_c);\ s_n = f(X_n)$
8:              $\Delta = s_n - s_c$
9:              if $\Delta < Tol_k$ then
10:                 $S_c = S_n;\ X_c = X_n$
11:                 if $s_c < s_{best}$ then
12:                     $s_{best} = s_c;\ X_{best} = X_c$
13:                 end if
14:             end if
15:             $k = k + 1$
16:         end while
17:         $Tol_{k+1} = \delta * Tol_k$
18:         $L_{k+1} = \beta * L_k$
19:         $T_{k+1} = \alpha * T_k$
20:     end while
21:     SR = $-s_{best}$
22:     return SR, $X_{best}$
23: end TAIPO
```

In this work, we used the classic TA algorithm and compared its performance with the algorithms proposed. As described in Algorithm 3, TA used a threshold sequence. The procedure shown in Algorithm 8, which is based on the method proposed in [23], was used to generate this sequence.

Algorithm 8. Generation of Threshold Sequence

```
1:  Initialize $n_{iter}$
2:  for i = 1 to $n_{iter}$ do
3:      $Generate\ random\ solution\ X_c$
4:      $X_n = Perturbation\ (X_c)$ // Generate neighbor solution
5:      $Compute\ \Delta_i = |f(X_n) - f(X_c)|$
6:  end for
7:  Sort $\Delta_1 \leq \Delta_2 \leq \Delta_3 \leq \ldots \leq \Delta_i$ and use as Threshold Sequence
```

In line 1, the size of the sequence is declared, which will be equal to the number of iterations of the TA classic algorithm. In line 3, a random solution is generated, then in line 4, a neighbor solution is generated, applying the perturbation described in Algorithm 6. In line 5, the difference between the new solution and the current solution is calculated. This process is repeated until the given number of iterations is generated. Finally, these differences are ordered from highest to lowest in line 7.

3.5. Proposed Algorithms Hybridized with Related Models

The Sharpe Ratio (Equation (24)) is regularly used as the objective function of GENPO, SAIPO, and TAIPO. To compare the performance of the algorithms proposed in this work with the related works described previously, we developed three hybrid algorithms using the last algorithm but replaced the objective function with that used by the reference models of Gilli, Yu, and Masese. These three hybrid algorithms are named: GENPO-X, SAIPO-X, and TAIPO-X, where X is the name of a reference model.

Hybrid Pseudocodes

The SAIPO Hybrid Algorithm shown in Algorithm 9 has a similar structure to the single SAIPO of Algorithm 5. In the beginning, the initial parameters are specified in line 1. Then, the objective function is defined in line 2. There are three alternatives or different models. Used in line 3 is the model presented in [10], the Yu model, shown in Equation (17). On the other hand, to use the Gilli model [21] calculated with Equation (18), it is necessary to compute the penalty p, established with Equation (19). This process is shown in lines 4–8. Finally, the function of the Masese model [22], calculated with Equation (20), is presented in line 10.

After selecting the model, a random feasible solution is generated in line 11. Then, the temperature cycle begins in line 13, and the Metropolis cycle begins in line 14. The Sharpe ratio is calculated at the end of the algorithm (lines 29–30). The rest of the process and parameters are similar to SAIPO.

Algorithm 9. SAIPO Hybrid Algorithm

```
1:  Parameters ($T_i$, $T_f$, $\alpha$, $L_k$, $\beta$)
2:      Define objective function $f(X)$
3:      if $f(X) = Yu\ model$ then $f(X) = \sigma^2(R) - ((E(R) - \gamma^2)$ end if //Objective function for Yu model
4:      else if $f(X) = Gilli\ model$ then
5:          Generate 1000 random portfolios and compute $\sigma^2(R)_{max}$ ; $\sigma^2(R)_{min}$ ; $\overline{R}$
6:          if $\gamma > E(R)$ then $p = \frac{\sigma^2(R)_{max} - \sigma^2(R)_{min}}{\gamma - \overline{R}}$ end if
7:          else $p = 0$
8:          $f(X) = \sigma^2(R) + p(\gamma - E(R))$                  //Objective function for Gilli model
9:      end if
10:     else if $f(X) = Masese\ model$ then $f(X) = \sigma^2(R)$ end if //Objective function for Masese model
11:     Generate a random feasible solution $X_c$
12:     $s_c = s_{best} = f(X_c); k = 0;\ X_c = X_{best};\ T_k = T_i$ .
13:     while $T_k \geq T_f$ do
14:         while $k < L_k$ do
15:             $X_n = Perturbation\ (X_c); s_n = f(X_n)$
16:             $\Delta = s_n - s_c$ ; r = random (0,1)
17:             if $\Delta < 0.$ then
18:                 $s_c = s_n; X_c = X_n$
19:                 if $s_c > s_{best}$ then $s_{best} = s_c; X_{best} = X_c$ end if
20:             else if $r < \hat{e}(-\Delta/T_k)$ then
21:                 $s_c = s_n; X_c = X_n$
22:             end if
23:             $k = k + 1$
24:         end while
25:         $L_{k+1} = \beta * L_k;$
26:         $T_{k+1} = \alpha * T_k$
27:     end while
28:     $SRup = E(R) - \gamma; SRdown = \sigma(R)$
29:     $SR = SRup/SRdown$
30:     return $s_{best}$, $X_{best}$, SR
31: end SAIPO Hybrid Algorithm
```

Algorithm 10, TAIPO Hybrid Algorithm, is similar to the SAIPO Hybrid Algorithm, changing the new solutions acceptance criterion, as shown in line 16.

Algorithm 11 illustrates the procedure of the GENPO Hybrid Algorithm. As with TAIPO and SAIPO hybrid algorithms, the fitness function is defined in line 4. In line 6, the *Yu model* is used as presented in [10], computed with Equation (17). On the other hand, the process behind the *Gilli model* [21] shown in Equations (18) and (19) is described in lines 9–14. Finally, the function of the *Masese model* [22], shown in Equation (20), is presented in lines 15–17. The remainder of the process is the same as in the GENPO algorithm.

Algorithm 10. TAIPO Hybrid Algorithm

```
Parameters ($T_i$, $T_f$, $\alpha$, $L_k$, $\beta$, $\delta$)
    Define objective function $f(X)$
    if $f(X) = Yu\ model$ then $f(X) = \sigma^2(R) - ((E(R) - \gamma^2)$ end if //Objective function for Yu model
    else if$f(X) = Gilli\ model$ then
        $Generate$ 1000 $random\ portfolios\ and\ compute\ \sigma^2(R)_{max}\ ;\ \sigma^2(R)_{min}\ ;\ \overline{R}$
        if $\gamma > E(R)$ then $p = \frac{\sigma^2(R)_{max} - \sigma^2(R)_{min}}{\gamma - \overline{R}}$ end if
        else $p = 0$
    $f(X) = \sigma^2(R) + p(\gamma - E(R))$ //Objective function for Gilli model
    end if
    else if $f(X) = Masese\ model$ then $f(X) = \sigma^2(R)$ end if //Objective function for Masese model
    Generate a random feasible solution $X_c$
    $s_c = s_{best} = f(X_c); k = 0; Tol_k = T_k;\ X_c = X_{best};\ T_k = T_i$
    while $T_k \geq T_f$ do
        while $k < L_k$ do
            $X_n = Perturbation\ (X_c);\ s_n = f(X_n); \Delta = s_n - s_c$
            if $\Delta < Tol_k$ then $s_c = s_n;\ X_c = X_n$
                if $s_c < s_{best}$ then $s_{best} = s_c;\ X_{best} = X_c$ end if
            end if
            $k = k + 1$
        end while
        $Tol_{k+1} = \delta * Tol_k$
        $L_{k+1} = \beta * L_k$
        $T_{k+1} = \alpha * T_k$
    end while
    $SRup = E(R) - \gamma;\ SRdown = \sigma(R)$
    $SR = SRup/SRdown$
    return $s_{best}$, $X_{best}$, SR
end TAIPO Hybrid Algorithm
```

Algorithm 11. GENPO Hybrid Algorithm

```
Begin /* GenPO */
    Set first gen $i = 0$
    Generate initial population (portfolio) $P(i)$
    Define objective function $f(X)$
    if $f(X) = Yu\ model$ then
        $f(X) = \sigma^2(R) - ((E(R) - \gamma^2)$
    end if //Objective function for Yu model
    else if$f(X) = Gilli\ model$ then
    $Generate$ 1000 $random\ portfolios\ and\ compute\ \sigma^2(R)_{max}\ ;\ \sigma^2(R)_{min}\ ;\ \overline{R}$
        if $\gamma > E(R)$ then
            $p = \frac{\sigma^2(R)_{max} - \sigma^2(R)_{min}}{\gamma - \overline{R}}$ end if
        else $p = 0$
            $f(X) = \sigma^2(R) + p(\gamma - E(R))$ //Objective function for Gilli model
    end if //Objective function for Gilli model.
    else if $f(X) = Masese\ model$ then
        $f(X) = \sigma^2(R)$
    end if //Objective function for Masese model
    While Converge = False or i<MaxGen do
    Begin /* Produce new generation */
        Begin /* Reproductive cycle */
                Select two individuals (portfolios), $X_1$ y $X_2$ of P(i),
                Crossing point random selection.
                Cross $X_1$ y $X_2$ getting two offspring $X_{h1}$, $X_{h2}$.
                Insertion $X_{h1}$ y $X_{h2}$ in P(i)
                Mutation random τ elements of P(i);
                Compute fitness function (i) of τ elements,
                Order individuals best evaluated in P(i).
                Limit the population to its original size.
            End
            Save best individual of the population P(i)
            If $f(X)$(i + 1) < $f(X)$(i) + tolerance then //* if SR doesn´t grow enough *//
                    gamma = gamma + 1            //* gamma counter grow *//
            If gamma >= conv then //* conv = generations number without improv SR *//
                Convergence = TRUE
        End
    End while
    i = i + 1
If i>MaxGen then
Converge: = TRUE
End
```

4. Results

The experiments performed with the algorithms presented in this paper, and the comparison with the hybrid algorithms described previously are shown in this section. To perform analytical tuning for SAIPO and TAIPO algorithms, some previous executions were required. Moreover, we showed the parameters used for those executions in Table 2.

Table 2. Tuning parameters for TAIPO and SAIPO algorithms.

Number of Executions	Initial Temperature T_i	Final Temperature Tf	L_k	Alpha Value
30	100	0.00001	100	0.95

4.1. Experiments

To test all algorithms, we used a dataset with 47 stocks taken from the Mexican Stock Exchange. The period evaluated was from 1 January 2017 to 30 June 2018.

In this paper, the performance of the proposed algorithms was compared with the Hybrid Algorithms described in Section 3.5. The MARR or risk-free rate used was 8%. The main metric that we used was the Sharpe Ratio, which allowed measurement of the relationship between expected return and risk. The performance of the algorithms was also measured with other metrics: expected return, risk, average portfolio correlation, and runtime. GENPO algorithm used an initial population of 50 individuals through 100 generations, and used 2 parents with a random selection cross point to generate 2 offspring, then applied an 8% mutation for the entire population. Finally, a stop criterion was implemented in the algorithm after 20 repetitions without improvement. Based on the evaluation, the individuals were ordered from best to worst and the least qualified individuals were eliminated from the population, leaving the population at its original size.

We compared the performance of the algorithms in two circumstances: first by applying the MARR constraint shown in Equation (25), which allowed only assets with an expected return equal to or greater than the MARR to be included in the portfolio; and then without this constraint. The average values of 30 runs are shown in Tables 3 and 4.

Table 3. Algorithm performance without MARR constraint.

Algorithm	Sharpe Ratio	Risk	Expected Return	Average Correlation	Stocks	Runtime (seconds)
TAIPO	**8.1**	0.0973	0.8683	0.0043	**7**	292.93
SAIPO	**8.1**	0.0973	0.8682	**0.0041**	**7**	301.63
TA	**8.1**	0.0976	0.871	0.0044	**7**	745.03
TAIPO-Yu	0.0015	0.0556	0.0801	0.0304	26	10.16
SAIPO-Yu	0.0055	0.0556	0.0803	0.0294	26	10.1
TA-Yu	−0.0103	0.0587	0.0788	0.0346	27	5.46
TAIPO-Gilli	0.7685	**0.0554**	0.1226	0.0296	27	11.01
SAIPO-Gilli	0.7605	**0.0554**	0.1221	0.0297	27	11.23
TA-Gilli	0.7636	**0.0554**	0.1223	0.0297	27	**5.78**
TAIPO Masese	0.1765	0.0656	0.0919	0.0134	10	9.84
SAIPO Masese	0.4795	0.0664	0.1124	0.0117	10	10.52
TA Masese	0.6323	0.0662	0.1229	0.0127	10	11.8
GENPO	6.95	0.087	0.7244	0.0187	21	462.14
GENPO-Yu	0.3874	0.0575	0.0803	0.0157	44	430.62
GENPO-Gilli	0.8077	**0.0563**	0.1255	0.0164	43	561.57
GENPO-Masese	0.6034	0.0564	0.1141	0.0207	20	487.96

Note in Table 3, we obtained the best result of the Sharpe Ratio using the algorithms TAIPO, SAIPO, and TA, while the second-best result was achieved by GENPO. When the asset constraint was not applied, the TA-Yu algorithm selected a portfolio with negative SR.

This negative value is a consequence of the expected lower return than the MARR (eight percent), which is the acceptable minimum value for accepting a portfolio.

Table 4. Algorithm performance with MARR constraint.

Algorithm	Sharpe Ratio	Risk	Expected Return	Average Correlation	Stocks	Runtime (seconds)
TAIPO	**8.1**	0.0980	**0.8716**	0.0048	7	17.43
SAIPO	**8.1**	0.0975	0.8699	0.0048	7	18.02
TA	**8.1**	0.0974	0.8687	0.0048	7	136.21
TAIPO-Yu	0.5915	0.0886	0.1324	0.0785	11	2.05
SAIPO-Yu	0.5915	0.0886	0.1324	0.0784	11	**2.04**
TA-Yu	0.5957	0.0884	0.1326	0.0785	11	2.81
TAIPO-Gilli	5.1604	**0.0673**	0.4273	0.0479	17	2.48
SAIPO-Gilli	5.1747	**0.0673**	0.4282	0.0478	17	2.53
TA-Gilli	5.148	**0.0673**	0.4264	0.048	17	3.15
TAIPO-Masese	3.3167	0.0714	0.317	0.0435	10	**1.8**
SAIPO-Masese	3.2771	0.0719	0.3154	0.0432	10	**2.0**
TA-Masese	3.3202	0.0718	0.3183	0.0427	10	3.79
GENPO	**7.96**	0.0951	0.8420	**0.0011**	9	68.64
GENPO-Yu	0.9061	0.0870	0.1569	0.0326	16	80.40
GENPO-Gilli	0.8663	**0.0673**	0.4215	0.0514	19	87.41
GENPO-Masese	4.9194	0.0676	0.4126	0.0384	13	88.25

In the risk metric, we obtained the best results with the hybrid algorithms SAIPO-Gilli, TAIPO-Gilli, TA-Gilli, and GENPO-Gilli. In terms of expected return, the portfolios obtained with TAIPO, SAIPO, TA, and GENPO, had higher values than the other algorithms. As shown in Tables 3 and 4, TAIPO, SAIPO, and TA algorithms achieved equivalent results in Sharpe Ratio with and without a constraint. However, the SR placed higher than the rest of the algorithms when the MARR constraint was applied.

In Table 4, we observe that the average correlation is higher when the MARR constraint is implemented than when it does not (as in Table 3). However, this difference is lower in TAIPO, SAIPO, and TA. We obtain the best average correlation value with the GENPO algorithm.

When the MARR constraint satisfies the assets, they can conform to the portfolio, and the algorithms may select them. In addition, portfolios with the lowest number of stocks are collected with TAIPO, SAIPO, and TA, which can be advantageous for these algorithms. We can observe that when the MARR is implemented, the runtime is reduced among all algorithms. Although the TA algorithm attains similar results to TAIPO and SAIPO in the Sharpe Ratio, the runtime is higher. Thus, the new TAIPO algorithm proposed in this paper represents a better alternative than the classical Threshold Accepting algorithm. However, SAIPO and TAIPO algorithms require a lower runtime than Genetic algorithms.

Table 5 shows the relationship between the result reached in the Sharpe Ratio and the runtime (seconds), that is $\frac{Sharpe\ Ratio}{Runtime}$, of each algorithm; a higher value represents a better result. When the constraint was applied, the best results were obtained with TAIPO-Gilli and SAIPO-Gilli.

When the constraint was not implemented, we obtained the largest $\frac{Sharpe\ Ratio}{Runtime}$, with the algorithm TA-Gilli. This metric can be used when seeking faster solutions, although faster solutions do not necessarily obtain the best quality results.

4.2. Statistical Test

To compare the results, a Friedman test [51] was applied, considering 30 observations, where each algorithm represented a treatment. Tables 6 and 7 show the results applied to the TA, TAIPO, and SAIPO algorithms and their hybrid versions. With a significance value of 5% and a p-value equal to zero, we can conclude that there are significant differences in at least one algorithm. In addition, those that reached the best results were the TAIPO, SAIPO, and TA algorithms, with the higher sum of ranks.

Table 5. Sharpe Ratio vs. Runtime.

	MARR Constraint	Without MARR Constraint
Algorithm	**Sharpe/Runtime**	**Sharpe/Runtime**
TAIPO	0.4647	0.0277
SAIPO	0.4495	0.0269
TA	0.0595	0.0109
TAIPO-Yu	0.2885	0.0001
SAIPO-Yu	0.2900	0.0005
TA-Yu	0.2120	−0.0019
TAIPO-Gilli	**2.0808**	0.0698
SAIPO-Gilli	**2.0453**	0.0677
TA-Gilli	1.6343	**0.1321**
TAIPO-Masese	1.8426	0.0179
SAIPO-Masese	1.6386	0.0456
TA-Masese	0.8760	0.0536
GENPO	0.1160	0.0150
GENPO-Yu	0.0113	0.0009
GENPO-Gilli	0.0099	0.0014
GENPO-Masese	0.0557	0.0012

Table 6. Friedman Test TA, TAIPO and SAIPO Algorithms without MARR constraint.

Algorithm	Median	Sum of Ranks
TA	8.1051	332
SAIPO	8.1051	330
TAIPO	8.1040	328
TAIPO-Gilli	0.7650	207
SAIPO-Gilli	0.7699	199
TA-Gilli	0.7557	197
TA-Masese	0.5874	163
SAIPO-Masese	0.4451	148
TAIPO-Masese	−0.3164	130
TA-Yu	−0.0088	111
SAIPO-Yu	0.0067	105
TAIPO-Yu	0.0025	90

Table 7. Friedman Test TA, TAIPO and SAIPO Algorithms with MARR constraint.

Algorithm	Median	Sum of Ranks
SAIPO	8.1000	332
TA	8.1000	330
TAIPO	8.0990	328
SAIPO-Gilli	5.1709	254
TAIPO-Gilli	5.1566	240
TA-Gilli	5.1480	226
TA-Masese	3.2775	155
SAIPO-Masese	3.2226	149
TAIPO-Masese	3.2577	146
TA-Yu	0.5956	84
SAIPO-Yu	0.5923	53
TAIPO-Yu	0.5889	43

In both cases, whether a constraint is applied or not, the TAIPO, SAIPO, and TA algorithms secured the best results in the Sharpe Ratio.

For the algorithms GENPO–Sharpe and its hybrid variants, considering a significance value of 5%, and with a p-value equal to zero, there were significant differences between them. Table 8 shows results for instances without constraint.

Table 8. Friedman Test Genetics Algorithms without MARR constraint.

Algorithm	Median	Sum of Ranks
GENPO	7.032	120
GENPO-Gilli	0.769	81.5
GENPO-Masese	0.617	68.5
GENPO-Yu	0.033	30

In the same way, a Friedman test was applied for the instance with constraint, obtaining a p-value equal to zero, showing a statistically significant difference. Results are displayed in Table 9.

Table 9. Friedman Test Genetics Algorithms with MARR constraint.

Algorithm	Median	Sum of Ranks
GENPO	8.086	120
GENPO-Masese	4.921	90
GENPO-Yu	0.924	51
GENPO-Gilli	0.866	39

Table 10 shows the comparison of the best algorithms when the constraint was not applied. With a p-value of zero, the null hypothesis can be rejected, thus if there are differences between the groups, the TAIPO, SAIPO, and TA algorithms show better performance.

Table 10. Friedman Test for best algorithms without MARR constraint.

Algorithm	Median	Sum of Ranks
TA	8.1049	92
SAIPO	8.1032	90
TAIPO	8.1037	88
GENPO	6.9434	30

Table 11 shows the results of the Friedman test undertaken to compare the performance of the best algorithms in each group when the MARR constraint was applied.

Table 11. Friedman Test for best algorithms with MARR constraint.

Algorithm	Median	Sum of Ranks
GENPO	8.1055	77
SAIPO	8.1009	76
TA	8.1006	75
TAIPO	8.1002	72

With a p-value of 0.964 and a significance value of 0.05, we can conclude that there are no significant differences.

5. Conclusions

This paper presents three algorithms for portfolio optimization: GENPO based on Genetic Algorithms, SAIPO based on Simulated Annealing, and TAIPO, which improves upon the classical Threshold Accepting algorithm (TA). We compared them with state-of-the-art algorithms, namely: Gilli, Yu, and Masese models. This comparison was undertaken by replacing the Sharpe ratio with the objective functions used in these models. The proposed algorithms maximize the Sharpe Ratio (SR) and apply the Minimum Acceptable Rate of Return (MARR) as an essential constraint. Therefore, we present three families of algorithms for portfolio optimization in this paper. For instance, the GA family constituents are GENPO, GENPO-Yu, GENPO-Gilli, and GENPO-Masese.

The proposed algorithms were designed to select the best combination of assets from any stock exchange such as S&P 500 and any other stock market. However, they were only tested with datasets from the Mexican stock exchange (BMV). The optimal result of these algorithms was defined as the proportion of each asset that maximized the objective function.

The experimental results revealed that the proposed algorithms, GENPO, SAIPO, and TAIPO, obtained the highest quality portfolio within each family. Upon application of the MARR constraint, the portfolio with the highest quality was obtained. Moreover, for the same algorithm, when this constraint was included, the execution time was significantly shorter. Regarding the GA family, the proposed GENPO algorithm showed the highest quality in both cases, with and without application of the MARR constraint. Moreover, GENPO did not require excessive tuning time to obtain high-quality portfolios. Similarly, the algorithms SAIPO, TAIPO, and TA showed superior performance with or without the MARR constraint. In addition, they had the lowest average correlation than that of their respective family.

Finally, in the statistical test, where the performance of the proposed algorithms was compared, no significant differences were found when the MARR constraint was applied. However, when the MARR constraint was not applied, SAIPO, TAIPO, and TA obtained a significant difference in quality compared with the GENPO algorithm. The difference was that the TAIPO and SAIPO algorithms converge faster than GENPO and TA algorithms. Therefore, TAIPO and SAIPO represent excellent alternatives for the portfolio integration problem.

The SAIPO and TAIPO algorithms contained more parameters than GENPO; their tuning process was longer and involved a series of previous executions; for this reason, the tuning of these algorithms was more complex than that of GENPO. However, once the appropriate parameters were obtained, the execution time of SAIPO and TAIPO was shorter than the time used by GENPO. However, with TAIPO, unlike SAIPO, for the acceptance criterion of new solutions, the calculation of probabilities was not required, which rendered the execution time of TAIPO slightly lower than SAIPO.

For future research, we propose an application of these algorithms to larger stock markets, including other objective functions and metrics. Finally, we propose the application and extension of these algorithms with other heuristics for design portfolios among several markets. Moreover, the applicability of these algorithms regarding derivatives and other stocks remains a problem that should be studied.

Author Contributions: Conceptualization, J.F.S. and G.C.V.; methodology, M.G.d.A. and J.L.P.A.; software, M.G.d.A. and J.L.P.A.; validation, G.C.V., J.G.B. and J.F.S.; formal analysis, J.F.S. and J.G.B.; investigation, M.G.d.A., J.L.P.A. and J.F.S.; writing—original draft preparation, M.G.d.A. and J.L.P.A.; writing—review and editing, J.G.B., J.F.S. and G.C.V.; supervision, J.F.S. and J.G.B.; project administration, J.F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this section, Data supporting are located in this paper.

Acknowledgments: The authors thank Conacyt for the scholarship for graduate studies that let us become researchers and/or full professors.

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

References

1. Markowitz, H. Portfolio Selection. *J. Financ.* **1952**, *7*, 77–91.
2. Markowitz, H.M. Foundations of Portfolio Theory. *J. Financ.* **1991**, *46*, 469–477. [CrossRef]
3. Mao, J.C.T. Models of Capital Budgeting, E-V Vs E-S. *J. Financ. Quant. Anal.* **1970**, *4*, 657. [CrossRef]
4. Fishburn, P.C. Mean-Risk Analysis with Risk Associated with Below-Target Returns. *Source Am. Econ. Rev.* **1977**, *67*, 116–126.
5. Fang, Y.; Lai, K.; Wang, S. *Fuzzy Portfolio Optimization Theory and Methods*; Springer: Berlin/Heidelberg, Germany, 2008.
6. Kalayci, C.B.; Ertenlice, O.; Akbay, M.A. A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Syst. Appl.* **2019**, *125*, 345–368. [CrossRef]
7. Doering, J.; Kizys, R.; Juan, A.A.; Fitó, À.; Polat, O. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Oper. Res. Perspect.* **2019**, *6*, 100121. [CrossRef]
8. Zanjirdar, M. Overview of Portfolio Optimization Models. *Adv. Math. Financ. Appl.* **2020**, *5*, 419–435.
9. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [CrossRef]
10. Yu, L.; Wang, S.; Lai, K.K. Multi-Attribute Portfolio Selection with Genetic Optimization Algorithms. *INFOR* **2009**, *47*, 23–30. [CrossRef]
11. Sefiane, S.; Benbouziane, M. Portfolio Selection Using Genetic Algorithm. *J. Appl. Financ. Bank.* **2012**, *2*, 143–154.
12. Hadi, A.S.; Naggar, A.A.E.; Bary, M.N.A. New model and method for portfolios selection. *Appl. Math. Sci.* **2016**, *10*, 263–288. [CrossRef]
13. Chen, B.; Zhong, J.; Chen, Y. A hybrid approach for portfolio selection with higher-order moments: Empirical evidence from Shanghai Stock Exchange. *Expert Syst. Appl.* **2020**, *145*, 113104. [CrossRef]
14. Pekár, J.; Čičková, Z.; Brezina, I. Portfolio performance measurement using differential evolution. *Central Eur. J. Oper. Res.* **2016**, *24*, 421–433. [CrossRef]
15. García, S.; Quintana, D.; Galván, I.M.; Isasi, P. Multi-objective algorithms with resampling for portfolio optimization. *Comput. Inform.* **2013**, *32*, 777–796.
16. Zaheer, H.; Pant, M. Solving portfolio optimization problem through differential evolution. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 3982–3987.
17. Liu, W.-H. Optimal computing budget allocation to the differential evolution algorithm for large-scale portfolio optimization. *J. Simul.* **2017**, *11*, 380–390. [CrossRef]
18. Ni, Q.; Yin, X.; Tian, K.; Zhai, Y. Particle swarm optimization with dynamic random population topology strategies for a generalized portfolio selection problem. *Nat. Comput.* **2016**, *16*, 31–44. [CrossRef]
19. Heidari, H.; Neshatizadeh, L. Stock Portfolio-Optimization Model by Mean-Semi-Variance Approach Using of Firefly Algorithm and Imperialist Competitive Algorithm. *Int. J. Bus. Dev. Stud.* **2018**, *10*, 115–143.
20. Kalayci, C.B.; Polat, O.; Akbay, M.A. An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm Evol. Comput.* **2020**, *54*, 100662. [CrossRef]
21. Gilli, M.; Këlezi, E. *A Heuristic Approach to Portfolio Optimization*; FAME Research Paper Series RP20; International Center for Financial Asset Management and Engineering, 2000; p. 14. Available online: https://ideas.repec.org/p/fam/rpseri/rp20.html (accessed on 3 January 2022).
22. Masese, J.M.; Othieno, F.; Njenga, C. Portfolio Optimization under Threshold Accepting: Further Evidence from a Frontier Market. *J. Math. Finance* **2017**, *7*, 941–957. [CrossRef]
23. Gilli, M.; Schumann, E. Heuristics for Portfolio Selection. *Int. Ser. Oper. Res. Manag. Sci.* **2017**, *245*, 225–253. [CrossRef]
24. Chang, T.-J.; Meade, N.; Beasley, J.; Sharaiha, Y. Heuristics for cardinality constrained portfolio optimisation. *Comput. Oper. Res.* **2000**, *27*, 1271–1302. [CrossRef]
25. Fogarasi, N.; Levendovszky, J. Sparse, mean reverting portfolio selection using simulated annealing. *Algorithmic Financ.* **2013**, *2*, 197–211. [CrossRef]
26. Salehpoor, I.B.; Molla-Alizadeh-Zavardehi, S. A constrained portfolio selection model at considering risk-adjusted measure by using hybrid meta-heuristic algorithms. *Appl. Soft Comput.* **2019**, *75*, 233–253. [CrossRef]
27. Kapiamba, J.N.; Ulungu, B.; Mubenga, P.K. Simulated Annealing vs Genetic Algorithm to Portfolio Selection. *IJSIMR* **2015**, *3*, 18–30.
28. Wang, X.; He, L.; Ji, H. Modified generalized simulated annealing algorithm used in data driven portfolio management. In Proceedings of the 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, 20–22 May 2016; pp. 1014–1017.
29. John, C. High Speed Hill Climbing Algorithm for Portfolio Optimization. *Tanzan. J. Sci.* **2021**, *47*, 1236–1242. [CrossRef]
30. Sen, T.; Saha, S.; Ekbal, A.; Laha, A.K. Bi-objective portfolio optimization using Archive Multi-objective Simulated Annealing. In Proceedings of the 2014 International Conference on High Performance Computing and Applications (ICHPCA), Bhubaneswar, India, 22–24 December 2014; pp. 1–6. [CrossRef]
31. Chen, W.; Wang, Y.; Mehlawat, M.K. A hybrid FA–SA algorithm for fuzzy portfolio selection with transaction costs. *Ann. Oper. Res.* **2018**, *269*, 129–147. [CrossRef]

32. Kumar, C.; Doja, M.N.; Baig, M.A. A Novel Framework for Portfolio Optimization Based on Modified Simulated Annealing Algorithm Using ANN, RBFN, and ABC Algorithms. In *Towards Extensible and Adaptable Methods in Computing*; Springer: Singapore, 2018; pp. 157–178.
33. Cohen, J.; Khan, A.; Alexander, C. Portfolio Optimization of 60 Stocks Using Classical and Quantum Algorithms. *arXiv* **2020**, 1–19.
34. Sharpe, W.F. Mutual Fund Performance. *J. Bus.* **1966**, *39*, 119–138. [CrossRef]
35. Choueifaty, Y.; Coignard, Y. Toward Maximum Diversification. *J. Portf. Manag.* **2008**, *35*, 40–51. [CrossRef]
36. Aneja, Y.P.; Chandra, R.; Gunay, E. A Portfolio Approach to Estimating the Average Correlation Coefficient for the Constant Correlation Model. *J. Financ.* **1989**, *44*, 1435–1438. [CrossRef]
37. Salazar, J.R.; Tella, R. Supplement Portfolio Construction Based on Implied Correlation. *EconoQuantum* **2014**, *12*, 125–144. [CrossRef]
38. Holland, J.H. *Adaptation in Natural and Artificial Systems*, 1st ed.; University of Michigan Press: Ann Arbor, MI, USA, 1975.
39. Kadar, A.; Faruque, F.; Mohammad, S.; Iqdal, H. Solving the Vehicle Routing Problem using Genetic Algorithm. *Int. J. Adv. Comput. Sci. Appl.* **2011**, *2*, 126–131. [CrossRef]
40. Gen, M.; Choi, J.; Ida, K. Improved genetic algorithm for generalized transportation problem. *Artif. Life Robot.* **2000**, *4*, 96–102. [CrossRef]
41. Werner, F. A survey of genetic algorithms for shop scheduling problems. *Math. Res. Summ.* **2017**, *2*, 15.
42. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
43. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of state calculations by fast computing machines. *Equ. State Calc. Fast Comput. Mach.* **1953**, *21*, 1087–1092. [CrossRef]
44. Dueck, G.; Scheuer, T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. *J. Comput. Phys.* **1990**, *90*, 161–175. [CrossRef]
45. Sánchez-Hernández, J.P.; Frausto-Solís, J.; González-Barbosa, J.J.; Soto-Monterrubio, D.A.; Maldonado-Nava, F.G.; Castilla-Valdez, G. A Peptides Prediction Methodology for Tertiary Structure Based on Simulated Annealing. *Math. Comput. Appl.* **2021**, *26*, 39. [CrossRef]
46. Frausto-Solis, J.; Hernández-Ramírez, L.; Castilla-Valdez, G.; González-Barbosa, J.J.; Sánchez-Hernández, J.P. Chaotic Multi-Objective Simulated Annealing and Threshold Accepting for Job Shop Scheduling Problem. *Math. Comput. Appl.* **2021**, *26*, 8. [CrossRef]
47. Frausto-Solis, J.; Román, E.F.; Romero, D.; Soberon, X.; Liñán-García, E. Analytically Tuned Simulated Annealing Applied to the Protein Folding Problem. *Comput. Vis.* **2007**, *4488*, 370–377. [CrossRef]
48. Frausto-Solis, J.; Sanvicente-Sánchez, H.; Imperial-Valenzuela, F. ANDYMARK: An Analytical Method to Establish Dynamically the Length of the Markov Chain in Simulated Annealing for the Satisfiability Problem. *Comput. Vis.* **2006**, *4247*, 269–276. [CrossRef]
49. Purata, L.; Frausto, S.; Gonzalez, J.; Castilla, G. Genpo-Sharpe: Stock selection for investing portfolio using a genetic algorithm with sharpe ratio applied to mexican stock exchange. In Proceedings of the 8th International Workshop on Numerical and Evolutionary Optimization, Ciudad de Mexico, Mexico, 8–10 September 2020; p. 100.
50. Dopfel, F.E. Asset Allocation in a Lower Stock-Bond Correlation Environment. *J. Portf. Manag.* **2003**, *30*, 25–38. [CrossRef]
51. Myles Hollander, E.C.; Douglas, A. *Wolfe, Nonparametric Statistical Methods*, 3rd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2013.