Pesquisa Operacional (2023) 43: e266962 p.1-23
doi: 10.1590/0101-7438.2023.043.00266962

Printed version ISSN 0101-7438 / Online version ISSN 1678-5142
www.scielo.br/pope
ARTICLES

MULTIOBJECTIVE EVOLUTIONARY METAHEURISTIC APPROACH TO THE CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM

Massika Ikhlef[1] and Méziane Aïder[2*]

Received August 17, 2022 / Accepted November 11, 2022

ABSTRACT. In this paper, we propose a multi-objective evolutionary metaheuristic approach based on the Pareto Ant Colony Optimization (P-ACO) metaheuristic and the non-dominated genetic sorting algorithms (NSGA II and NSGA III) to solve a bi-objective portfolio optimization problem. P-ACO is used to select the best assets composing the efficient portfolio. Then, NSGA II and NSGA III are separately used to find the proportional weights of the budget allocated to the selected portfolio. The results we obtained by these two algorithms were compared to designate the best performing algorithm. Finally, we performed another comparison between our results and those of an exact method used for the same problem. The numerical experiments performed on a set of instances from the literature revealed that the combination of the ant colony optimization metaheuristic and the NSGA III genetic algorithm that we proposed most often gave much better results than both the combination of the ant colony optimization metaheuristic and NSGA II on the one hand and the iterative approach on the other hand.

Keywords: multiobjective optimization, portfolio selection, Pareto ant colony optimization, non-dominated sorting genetic algorithm.

1 INTRODUCTION

The first mathematical models related to the portfolio selection problem under uncertainty are due to Markowitz (1952). Such a problem was first considered as an optimization problem in which a combination of assets of minimum variance is chosen for any given level of expected return and, simultaneously, of maximum expected return for any given level of portfolio variance. Later, with the rise of multi-objective optimization, these models gave way to the mean-variance model, one of the most important portfolio optimization models (Markowitz, 1991), which consists of simultaneously optimizing two objectives, maximizing the expected return and minimizing the risk measured by the variance, by searching for feasible portfolios that offer the best compromise between risk and return. These trade-off portfolios are usually referred to as efficient portfolios or

*Corresponding author
[1]LaROMaD, Fac. Maths, USTHB, Algiers, & Fac. Sciences, UMB-Boumerdes, Algeria – E-mail: m.ikhlef@univ-boumerdes.dz http://orcid.org/0000-0002-3805-9716
[2]LaROMaD, Fac. Maths, USTHB, PB 32 Bab Ezzouar, 16111 Algiers, Algeria – E-mail: meziane.aider@usthb.edu.dz http://orcid.org/0000-0001-7195-810X

Pareto fronts, composed of non-dominated solutions to a multi-objective optimization problem. While the importance of the mean-variance model is widely recognized, the basic model has certain shortcomings in taking into account certain essential assumptions inherent to the practical world context.

To comply with the reality of the real world, we need to consider various additional constraints. Many papers have then been published extending the basic model by adding and imposing real-world constraints.

Budget constraints were considered by Smith et al. (2012) to impose the amount to invest. Floor and ceiling constraints were introduced by Streichert et al. (2004) to specify lower and/or upper bounds on the weight of each asset to be included in the portfolio. The turnover constraint was used by Krink & Paterlini (2008) to control the amount of money that can be traded for buy as well as sell. This constraint is used to control transaction costs. Mansini & Speranza (1999) considered the case of markets where the assets can have different minimum tradable lots, and their buying and selling must be done in a multiple of the minimum transaction lots and defined the round lot (or minimum lot) constraints. Lwin & Kendal (2014) considered the cardinality constraint that restricts the total number of assets to be included in the portfolio, the quantity constraint that restricts the minimum and maximum proportions of assets held in the portfolio, the pre-assignment constraint that requires some specific assets to be included in the portfolio and round lot constraint that requires to invest the assets in units of a certain size respectively.

Cardinality constraint, introduced by Chang et al. (2000), restricts the total number of assets to be included in a portfolio. The Basic Markowitz problem is an NP-hard problem and is complicated by the incorporation of a cardinality constraint. It can be formulated as a mixed integer quadratic optimization problem that can be solved using exact methods for small instances only. This is why most of the methods and approaches proposed in the field to solve the portfolio optimization problem are based on approximative algorithms such as Genetic Algorithms (GA), Ant Colony (AC), Artificial Bee Colony (ABC), and Particle Swarm (PS) (Bezoui et al., 2018).

Doerner et al. (2002) proposed a metaheuristic approach based on the Ant Colony to solve the multiobjective portfolio selection problem and compared its performance to those of other heuristic approaches, namely Pareto simulated annealing, and Non-dominated Sorting Genetic Algorithm.

Anagnostopoulos & Mamanis (2010) used three evolutionary multiobjective optimization techniques: Non-dominated Sorting Genetic Algorithm II (NSGA II), Pareto Envelope based Selection Algorithm (PESA), and Strength Pareto Evolutionary Algorithm 2 (SPEA 2), to solve the portfolio optimization problem with three objectives and discrete variables.

Kumar & Mishra (2017) solved the Portfolio optimization problem using a novel covariance-guided Artificial Bee Colony algorithm. Macedo et al. (2017) used Multiobjective Evolutionary Algorithms (MOEA) and Technical Analysis Rules to solve the Mean-Semi-variance Portfolio Optimization problem.

Lwin & Kendal (2014) proposed an efficient learning-guided hybrid multi-objective evolutionary algorithm to solve the constrained portfolio optimization problem in the extended mean-variance framework.

Fernandez et al. (2007) presented several potential advantages of the mean-variance paradigm over other methods used to solve the portfolio optimization problem, like linear programming and greedy algorithms.

Liu & Xiao (2021) established several optimization schemes to study the portfolio problem and showed that the genetic algorithm model is superior to the quadratic programming method.

Finally, we mention the case study by Fernandez et al. (2007) that proposed a decision model using both decision analysis and Bayesian risk analysis concepts in the design of a portfolio for production planning in the sugarcane industry in Brazil.

In this paper, we present a bi-objective portfolio optimization problem with three constraints. The first constraint is the budget constraint that means that all available capital is invested and that all portfolios have non-negative weights. The second is the cardinality constraint that requires fixing the number of assets in the portfolio. The third constraint, called the pre-allocation constraint, consists in fixing pairs of assets that cannot both be selected in the portfolio (at most one of them can be considered in the portfolio).

To solve this problem, we decomposed our work into two steps. The first step consists to find and select the best candidates for assets constituting the efficient portfolio that offers the best trade-off between risk and return. For this purpose, we use Pareto Ant Colony Optimization as a special metaheuristic to solve the portfolio selection problem.

In the second step of this work, we apply two versions of the Non-Dominated Sorting Genetic Algorithm, NSGA II and NSGA III, to find the proportion weights of the budget that will be allocated to the selected portfolio. The method is implemented and applied to compare its performance with that of the iterative method proposed by Bezoui et al. (2018) to solve a bi-objective portfolio optimization problem under constraints. Numerical experimentation is performed with real-world data.

The rest of the paper is structured as follows: After this brief introduction, Section 2 describes the multi-objective portfolio optimization problem. Section 3 presents the Ant Colony Optimization approach, the two Genetic Algorithms NSGA II and NSGA III, and how they were implemented to solve this problem. Empirical results are reported and discussed in Sections 4 and 5. In Section 6, we have compared the obtained results with those of an exact method. In section 7, we present the major conclusions of this study.

2 PROBLEM DESCRIPTION

Portfolios can be described as subsets of the set of all n asset propositions, they are modeled as vectors $y = (y_1, y_2, \ldots, y_n)$ where the binary variable y_i indicates whether the i asset is included in the portfolio $(y_i = 1)$ or not $(y_i = 0)$. This work consists in determining efficient asset portfolios for which there is no other possible alternative that promises better values in at least one of the objectives (expected return that should be maximized and risk that should be minimized) and that offers at least the same value in all the others, under the cardinality and pre-assignment constraints described below.

Cardinality Constraint (CC): It is expressed as follows:

$$\sum_{i=1}^{n} y_i = K, \tag{1}$$

where K is a fixed number of assets that a portfolio should include, y_i is the binary variable that equals 1 if asset i is included in the portfolio and 0 otherwise. This constraint is used to facilitate the management of the portfolio and to reduce its management costs.

Pre-allocation constraint: In some cases, generally at the request of the investor, specific assets must be included or excluded from the portfolio. In this work, we considered a set F of asset pairs (i, j) such that only one asset of each pair can be retained in the portfolio. This constraint can be expressed as follows:

$$\begin{array}{rcll} y_i . y_j & = & 0, \quad \forall (i,j) \in F, & (2.1) \\ & & \text{or, equivalently, since } y_i, y_j \in \{0,1\} & \\ y_i + y_j & \leq & 1, \quad \forall (i,j) \in F. & (2.2) \end{array} \tag{2}$$

Budget constraint: It is also called the summation constraint and is expressed as follows:

$$\sum_{i=1}^{n} x_i = 1. \tag{3}$$

This constraint requires that all portfolios have non-negative weights that amount to 1.

Mathematical model: After finding the efficient asset portfolio, we must determine the proportion budget weights $x = (x_1, \ldots, x_n)$ that will be allocated to the selected assets to maximize the expected return and minimize the risk under the three constraints above.

The proposed model in this paper is formulated as follows:

$$
\left\{
\begin{array}{rcllr}
\max \sum_{i=1}^{n} \mu_i x_i & & & & (4.1)\\
\min \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_i x_j & & & & (4.2)\\
\sum_{i=1}^{n} x_i & = & 1 & & (4.3)\\
\sum_{i=1}^{n} y_i & = & K & & (4.4)\\
y_i + y_j & \leq & 1 & \forall (i,j) \in F & (4.5)\\
x_i & \leq & y_i & \forall i \in E & (4.6)\\
x_i & \geq & 0 & \forall i \in E & (4.7)\\
y_i & \in & \{0,1\} & \forall i \in E & (4.8)
\end{array}
\right. \qquad (4)
$$

where n is the number of assets, x_i is the weight of asset i in the portfolio, μ_i is the expected return of asset i, ρ_{ij} is the correlation between assets i and j.

Equation (4.1) is the first objective function representing the maximization of the portfolio return, Equation (4.2) is the second objective function representing the minimization of the portfolio risk, Equations (4.3), (4.4) and (4.5) are respectively the budget, the cardinality and the pre-allocation constraints already described. Inequality (4.6) means that the weight of each asset is non-null only if the asset is retained. Constraints (4.7) and (4.8) represent the variation domains of the decision variables x_i and y_i respectively.

3 SOLUTION PROCEDURES

In this section, we describe the two approaches used to solve the above problem: the Pareto Ant Colony Optimization (P-ACO) is used to determine efficient asset portfolios and the Non-Dominant Sorting Genetic Algorithm II (NSGA II) allows to determine the proportion weights of the budget allocated to the efficient asset portfolio.

3.1 Pareto Ant Colony Optimization

The Ant Colony approach imitates the behavior shown by real ants when searching for food (Ibáñez, 2004). They communicate information about food sources via pheromone, which they secrete as they move along. When an ant finds a food source, it returns to the nest (Dorigo & Stutzle, 2004). As ants on short paths will return to the nest faster, more pheromone will be deposited on the shortest paths. Moving ants accordingly choose their path with a probability that depends on the amount of pheromone detected and, consequently, paths that are more frequently traveled become more attractive and, through this self-reinforcing behavior, will be used more often. Further, the pheromone evaporates over time, so that pheromone trails of infrequently traveled paths become weaker while attractive paths are reinforced (Doerner et al., 2006).

While such artificial ant colony systems have been successfully applied to various single-objective problems, several extensions have been necessary in order to be able to tackle multiobjective project portfolio selection problems at hand (for a detailed discussion of these modifications (Gambardella et al., 1999; Gravel et al., 2002)). Therefore, algorithms used to solve multi-objective problems and derived from the ACO metaheuristic are called Pareto Ant Colony Optimization (P-ACO) algorithms.

In the initialization phase of the P-ACO algorithm, we generate Γ ants and every ant has an empty portfolio denoted by $y = (0,\ldots,0)$. The lifespan Ξ and the objective weights $w = (w_1, w_2)$ are determined randomly from $[0,1]$ for each ant (Doerner et al., 2006).

In the construction phase of the algorithm, each ant tries to construct its own portfolio y by using the decision rule based on pheromone information τ_i. A feasible asset i is selected to be added to the current portfolio y according to the probability distribution p_i given by:

$$p_i = \frac{\left[\sum_{a=1}^{2}(w_a \tau_i^a)\right]^{\alpha}}{\sum_{h=1}^{n}\left[\sum_{a=1}^{2}(w_a \tau_h^a)\right]^{\alpha}}. \tag{5}$$

This probability is biased by the positive parameter α, which determines the relative influence of the trails.

After the construction of the portfolio by each ant, we have to test the feasibility and efficiency of each portfolio. If the considered portfolio is feasible and efficient, then it will be stored and saved.

3.1.1 Pheromone information

The pheromone information in the algorithm is represented by a matrix of A rows and n columns, where A is the number of objectives (and it is equal to 2 in our case) and n is the number of assets. Each element of this matrix is denoted by τ_i^a and represents the current pheromone information.

A local pheromone update is performed once an artificial ant has added an asset to the portfolio. When an ant selects an asset i, the amount of pheromone on the element τ_i^a of the pheromone vector is decreased for each objective a. The local pheromone update rule for these elements is given as follows:

$$\tau_i^a = (1-\rho)\tau_i^a + \rho\tau_0, \tag{6}$$

where τ_0 is the initial value of trails and ρ is the evaporation rate.

The proposed P-ACO algorithm to determine the efficient asset portfolio follows Doerner et al. (2006).

Algorithm 1 Procedure P-ACO

Step 0: Initialization of P-ACO.
  - Create Γ ants;
  - Initialize pheromone vectors with τ_0;

Step 1: Iteration:
for each $p \in P$ **do**
  - Fix the lifespan of the ant $\Xi = K$; ▷ K is the fixed number of assets to be selected;
  - Set $y = (0, \ldots, 0)$; ▷ Create an empty portfolio y for each ant, $\sum_i y_i = 0$ and $|y| = n$;
  - Set sum $:= 0$; ▷ Create the variable sum which gives the number of assets included to y;
  - Determine, randomly from the interval $[0,1]$, the objective weight w_a for each objective a;
  while sum $\leq K$ **do**
    Select an asset i according to the probability distribution and add it to y;
    Update local pheromone information;
    sum $:=$ sum $+ 1$;
  end while
end for

Step 2: Checking feasibility of portfolio y.
if portfolio y is feasible **then**
  check efficiency of portfolio y;
end if
if portfolio y is efficient **then**
  store portfolio y and remove dominated ones.
end if

3.2 Overview of NSGA II and NSGA III

Over the last fifteen years, we have observed a substantial development of multiobjective evolutionary algorithms (Metaxiotis & Liagkouras, 2012; Deb, 2001; Coello et al., 2007) and some of these algorithms have reached a high level of acceptance as efficient means to obtain good solutions for complex problems within a reasonable amount of time. For the most popular, attempts to enhance their performance have been successful and, consequently, new versions of the algorithms have emerged. This is the case of the two versions of the Non-Dominated Sorting Genetic Algorithm: NSGA II and NSGA III (Macedo et al., 2017).

NSGA II was introduced by Deb et al. (2002) as an improvement on the original NSGA. Several studies have highlighted the good performance of this algorithm compared to other MOEAs. In NSGA II, we start by creating a random parent population, P_t of size Z that is sorted using a non-dominated sort. The genetic operators (selection, crossing and mutation) are applied to P_0 to create an offspring population Q_t of size Z. The two populations P_t and Q_t are combined together to form R_t of size $2Z$. Next, a non-dominated sorting procedure is used to sort the entire population R_t (Ghosh & Das, 2008) and identify all non-dominated fronts. First, the non-dominated solutions are selected to be the first non-dominated front F_1 and given a rank of 1. Then, these solutions are ignored and a second non-dominated front F_2 is determined and assigned a rank 2. In the same way, all solutions are sorted. The crowding distance is used to classify solutions having the same rank and is applied in the following way: in each front, the solutions are sorted according to the value of each objective function and the extreme solutions are given a large distance so that they are always selected. The remaining solutions are assigned a distance value equal to the normalized absolute difference of the function values of two adjacent solutions (Anagnostopoulos & Mamanis, 2010).

NSGA III was proposed by Deb & Jain (2014), in which the crowded distance of NSGA-II is replaced by reference points. The NSGA III obtains the $(t+1)$ generation by combining the parent and offspring populations $R_t = P_t \cup Q_t$ where the size of R_t is $2Z$ (as in NSGA II). According to the non-dominated sorting rules, R_t is then divided into different levels (fronts), denoted by $F_1, F_2, \ldots$. Starting from F_1, each level is selected one at a time to construct a new population S_t, and the size of S_t is equal to or larger than Z for the first time. If the last level included is the v^{th} level, solutions in S_t/F_v (levels before F_v and solutions composing the v^{th} level) are chosen for the next parent population P_{t+1} while solutions in the remaining levels are rejected. However, when the size of the new population exceeds Z and thus the last level selected F_v cannot be fully included in this population, NSGA III uses a selection process to decide which r solutions from F_v will be included in this population.

To select the remaining r solutions from level F_v, NSGA III applies a selection process based on reference points. The process considers a set of reference points widely and uniformly distributed on the normalized hyperplane inherent to the optimization objectives of the problem addressed by the algorithm. Then, the process emphasizes the selection of solutions from F_v which are associated with each of these reference points. This process promotes the selection of diverse and well-distributed non-dominated solutions, with the aim of preserving the diversity and distribution of the new population.

NSGA III considers the same termination criterion used by NSGA II to finish its execution. After achieving such a criterion, NSGA III provides the Pareto set of the population corresponding to the last generation.

3.3 Implementation of NSGA II in the portfolio problem

Each non-dominated solution generated by the P-ACO algorithm is injected into the NSGA II algorithm to find the proportions of the budget allocated to the efficient asset portfolio found previously. The following steps show how the approach is implemented:

1. Consider an efficient asset portfolio generated by P-ACO algorithm under the cardinality constraint.

2. Create a vector of size K, containing the return of each asset i composing the portfolio (K values of return).

3. Create a matrix of dimension $K \times K$, containing the covariance of each couple (i, j) (i and j are two assets composing the portfolio).

4. Create N_{pop} vectors of size K containing the initial proportions x_i randomly chosen from $[0, 1]$ and normalize them by using the formula:

$$x_i' = \frac{x_i}{\sum_{k=1}^{K} x_k}. \tag{7}$$

   To make easy the implementation of the algorithm, it will be preferable to regroup these vectors in a matrix of dimension $N_{\text{pop}} \times K$, so that each line of this matrix represents an initial solution. In this work, we have $N_{\text{pop}} = 50$.

5. Calculate the expected return on each solution using the formula:

$$\sum_{i=1}^{K} \mu_i x_i, \tag{8}$$

   the value of the expected return on each asset μ_i is downloaded from Beasley (1990).

6. Calculate the risk of each solution using the formula:

$$\sum_{i=1}^{K} \sum_{j=1}^{K} x_i x_j \rho_{ij}, \tag{9}$$

   the correlation ρ_{ij} is downloaded from Beasley (1990).

7. Sort the solutions by using the non-domination notion and select N_{pop} solutions from the first ones. We obtain a matrix of N_{pop} rows, each row of this matrix represents a selected solution. We denote this matrix by M.

8. Create N_{pop} new solutions using crossover operator. The crossover operator used in this work is called arithmetic crossover, it consists in choosing some pairs of the selected solutions according to the crossover probability p_c and then combining the two solutions composing each pair using :

$$(x, x') \longmapsto \alpha x + (1 - \alpha)x', \quad \alpha \sim U[0, 1]. \tag{10}$$

   In this work, we have $\alpha = 0.46$, we denote the matrix containing these new solutions by MC.

9. After applying the arithmetic crossover, the mutation operator occurs with a mutation probability denoted p_m. In this work, we use the following mutation method:

   - Create a matrix of dimension $N_{\text{pop}} \times K$ composed of random values from $[0, 1]$, then check each value m_{ij} of this matrix.

- If m_{ij} is less than the mutation probability p_m, then the element of the i^{th} row and the j^{th} column of the matrix MC is replaced by another value from $[0,1]$.

We thus generate a matrix containing an offspring population.

10. Normalize the matrix of the offspring and combine it with the initial matrix. We thus obtain a combined matrix of $2N_{\text{pop}}$ solutions.

11. Calculate the expected return and risk of the $2N_{\text{pop}}$ solutions and classify all solutions according to the notion of non-domination. The non-dominated solutions are selected to constitute the first non-dominated front and are assigned a rank of 1. Next, these solutions are ignored and the second front is determined and assigned a rank of 2. In the same way, all solutions are classified to constitute all fronts.

12. Compute the crowding distance between solutions belonging to the same front in the following way:

    On each front, the solutions are sorted according to the value of each objective function and the extreme solutions are given a large distance so that they are always selected. The remained solutions are assigned a distance value equal to the absolute normalized difference of the function values of two adjacent solutions.

13. Select the first N_{pop} solutions and perform the crossover and mutation operators to generate the new offspring.

14. Repeat 10, 11,12,13 until the maximum number of iterations is reached.

3.4 Implementation of NSGA III in the portfolio problem

After determining all non-dominated fronts $F_1,\ldots,F_v$, NSGA III applies a selection process based on reference points to select the remaining solutions from the last selected front F_v and obtain the parent population P_{t+1}. The key implementation steps of the NSGA III process are as follows.

Step 1 (*Normalization of the Objective Values*). The objective values of the population members are normalized using the ideal and extreme points. In a population S_t, use the minimum values of all objectives to construct the ideal point $z^{\min} = (z_1^{\min},\ldots,z_A^{\min})$. The objective value of each solution is translated by subtracting the ideal point $z^{\min}$.

$$f'_a(x) = f_a(x) - z_a^{\min}, \tag{11}$$

where $a = 1,\ldots,A$. In our case $A = 2$, and $f_a(x)$ is the a^{th} objective value of the solution x.

The extreme point is identified by finding the solution that minimizes the following achievement scalarization function (ASF) with the weight vector w:

$$\min \text{ASF}(x,w) = \max_{a=1}^{A} \frac{f'_a(x)}{w_a}, \tag{12}$$

where $w = (w_1, \ldots, w_A)$ is the weight vector. For finding the a^{th} extreme point, we set $w_a = 1$, while the other weights are set to a small value 10^{-6}. We use A extreme points to obtain an A dimensional linear hyperplane. The $f'_a(x)$ can be normalized as follows:

$$f^n_a(x) = \frac{f'_a(x)}{c_a - z^{\min}_a}, \tag{13}$$

where c_a is the intercept of the a^{th} objective axis.

Step 2 (*Generation of Reference Points*): Reference points are commonly generated on a normalized hyperplane using Das and Dennis's systematic approach (Das & Dennis, 1998). For A objectives and k divisions of each objective, the total number H of reference points is:

$$H = \begin{pmatrix} k+A-1 \\ k \end{pmatrix}. \tag{14}$$

Step 3 (*Perpendicular Distance Computation*): After normalizing the objective values and generating reference points, the perpendicular distance between the objective value of each solution and a reference line (joining the origin with a reference point) is computed. For a population S_t, a solution is associated with the reference point of the minimum perpendicular distance.

Step 4 (*Niche-Preservation Operation*): The niche count ρ_h is equal to the number of solutions in S_t/F_v associated with the h^{th} reference point. The minimum niche count is $J_{\min} = \min \rho_h$. The reference point with $J_{\min}$ is chosen. If $J_{\min} > 1$, one reference point is chosen randomly. We set the chosen reference point as the l^{th} reference point.

If $\rho_l \geq 1$ and the l^{th} reference point is associated with one or more solutions in F_v, a solution in F_v is randomly selected into population P_{t+1}, and the value of ρ_l is incremented by one. If $\rho_l \geq 1$ and no solution in F_v with the l^{th} reference point, this reference point is not considered in the t^{th} generation.

If $\rho_l = 0$ and the l^{th} reference point is associated with one or more solutions in F_v, the solution with the minimum perpendicular distance is selected into population P_{t+1} and the count ρ_l will add one. If $\rho_l = 0$ and no solution in F_v is associated with the l^{th} reference point, this reference point is not considered in the t^{th} generation (Liu et al., 2019).

Step 5 (*Genetic Operations*): After the parent population P_{t+1} is obtained, the offspring population Q_{t+1} is generated by applying the arithmetic crossover and the mutation operator mentioned in Steps 8 and 9 of NSGA II. In other way, we have also used a mutation operator based on local search algorithm.

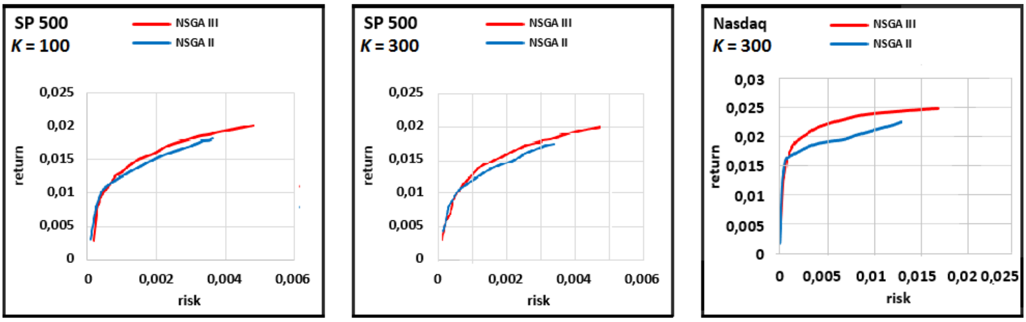
Algorithm 2 Local search algorithm

1: **for** each solution x from MC **do** ▷ MC is the matrix containing all solutions obtained by applying the crossover operator;
2: Calculate the objective value of the solution x;
3: **repeat**
4: - Set ε; ▷ a vector randomly chosen on the interval [0,1];
5: - Calculate the new solution $x' = x + \varepsilon$;
6: - Calculate the objective value of the new solution x';
7: **if** x' is dominated by x **then**
8: $x := x$;
9: **repeat** 4, 5, 6;
10: **end if**
11: **if** x is dominated by x'; **then**
12: **repeat** 4, 5, 6;
13: **end if**
14: **if** x' and x are equivalent **then**
15: **save** x' and x;
16: $x := x'$;
17: **repeat** 4, 5, 6;
18: **end if**
19: **until** termination criterion is **true**.
20: **end for**

4 EXPERIMENTAL RESULTS OF P-ACO AND DISCUSSION

In this section, we report the experimental results we obtained using a public data set of three stock markets downloaded from the Beasley's OR Library Beasley (1990): Hong Kong Hang Seng with 31 assets, American S&P 100 with 98 assets, Japanese Nikkei 225 with 225 assets. We also used two other stock markets that were described by Cesarone et al. (2014), available from Cesarone & Tardella (2017). These authors reported 263 weekly prices from March 2003 to March 2008 of American S&P 500 with 276 assets and European-American NASDAQ with 2196 assets. We performed the numerical study on a personal computer equipped with Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz 1.80 GHz, 8th gen, 08 Go RAM and the operational Windows 8. All procedures are implemented in R.

Recall that, in this work, we want to solve a portfolio optimization problem under the cardinality constraint and the pre-allocation constraint. For example, with the Hang Seng index, we set a single value to the cardinal ($K = 10$). Regarding the pre-allocation constraint, we have fixed, for the Hang Seng market, three pairs of assets (16,17), (17,18) and (16,18), the two assets composing each pair cannot both be selected and only one asset of each pair can be considered in



(a) (b) (c)

Figure 1 – Non-dominated solutions of SP500 index for $K = 100$ and $K = 300$ and Nasdaq for $K = 300$.

the portfolio. For the Nasdaq market, we set three K values (10, 100, 300) and six pairs of assets for the pre-allocation constraint.

We note that the P-ACO parameters for the computation experiments are $\alpha = 1, \rho = 0.7, \Gamma = 10, 20, 30, q_0 = 0$, and $\tau_0 = 0.9$. Table 1 shows the results.

Table 1 shows the risk and expected return of each non-dominated solution generated by the P-ACO algorithm according to each value of K for the different markets. For Hang Seng index, the algorithm uses 10 artificial ants and each ant constitutes its own solution (portfolio), after which the algorithm checks the feasibility and efficiency of the ten constructed portfolios to save the efficient ones (non-dominated solutions) and eliminate the others (dominated solutions). We see in the table that the P-ACO algorithm generated only three non-dominated solutions (three efficient portfolios) under the cardinality constraint $K = 10$. We tried to increase the number of ants (30 ants, 50 ants, 100 ants, ...) to have more than three non-dominated solutions but this did not change the result as we expected. We also notice in Table 1 that there is not much difference between these three efficient portfolios, they have practically the same expected return and risk. In this case, the investor can easily choose the best efficient portfolio among the three. For SP100, SP500 and Nasdaq, we note that the higher the cardinality value, the higher the return and risk. In other words, if the investor favors return, he must choose a portfolio with a high cardinality value, but if the investor favors low risk, he must be satisfied with studying small portfolios. Table 2 shows the three efficient asset portfolios obtained for the Hang Seng index using the P-ACO algorithm under the cardinality constraint $K = 10$. We notice that the best portfolio, according to the expected return, is port 1. After checking the data set corresponding to the Hang Seng Index, we find that this portfolio is composed of assets with the highest return.

5 EXPERIMENTAL RESULTS OF NSGA II AND NSGA III

In this section, we present the results obtained by the implementation of the two genetic algorithms NSGA II and NSGA III. Figures 1, 2 and 3 show the frontiers of non-dominated solutions generated to SP500, Nasdaq, Hang-Seng and SP100 according to several values of cardinality.

Table 1 – Risk and return of the efficient asset portfolios obtained by P-ACO.

SP 100 with 98 assets			**SP 500 with 476 assets**		
K	**Return**	**Risk**	*K*	**Return**	**Risk**
50	0.213265	512.678372	10	0.1354794	0.1376716
	0.112512	422.242630		0.09525867	0.10177683
	0.190705	503.768738		0.08501251	0.08373711
	0.116.808	423.820148		0.06799201	0.03822819
	0.176744	497.690060		0.05084089	0.03679313
	0.171635	495.680658		0.04417696	0.03518582
	0.131665	429.933680		0.04118156	0.03308424
	0.135661	433.616562		0.02510106	0.02298574
	0.149273	433.901566		0.02734874	0.02596494
	0.157965	441.118896		0.02871795	0.0266791
	0.161059	443.883738	100	0.7540985	5.2578543
	0.164927	450.406884		0.4622423	3.2788443
	0.168713	456.368526		0.3448385	2.8524634
80	0.278026	1207.943510		0.2410349	2.3486477
	0.256027	1159.959632		0.6114916	4.1390230
	0.234959	1115.002572		0.4323154	3.2468762
	0.262656	1165.628104		0.3193307	2.6280283
	0.249885	1131.073054		0.547265	3.500980
	0.271474	1184.533588		0.050428	49.152482
	0.255227	1159.089108		0.2950339	2.4596917
	0.234178	1110.485858		0.5023697	3.3834249
	0.248012	1121.461844		0.3770722	2.8849828
	0.224832	1108.411672	300	1.561179	32.089310
	0.266096	1178.966284		1.174359	28.197236
	0.255227	1138.903924		0.9588885	23.8943005
	0.214503	1097.030830		1.415065	31.665333
				1.150750	25.919244
				1.389743	28.851918
				1.211427	28.791523

Nasdaq with 2196 assets			Hang Seng with 31 assets		
K	**Return**	**Risk**	*K*	**Return**	**Risk**
10	0.2080628	0.214315	10	0.058008	54.094946
	0.1503612	0.17065663		0.053634	50.604180
	0.1428687	0.1422794		0.4052092	3.1003410
	0.11326854	0.09698731			
	0.09275083	0.08668862			
	0.08419325	0.0847406			
300	3.263283	55.474549			
	2.021897	44.972060			
	1.4616685	42.463714			
	1.065910	32.841021			
	0.7308886	29.8021161			
	0.4063562	27.2410551			
	2.270448	46.702699			
	1.618908	43.829508			

Table 2 – Efficient asset portfolios obtained by P-ACO for Hang Seng index.

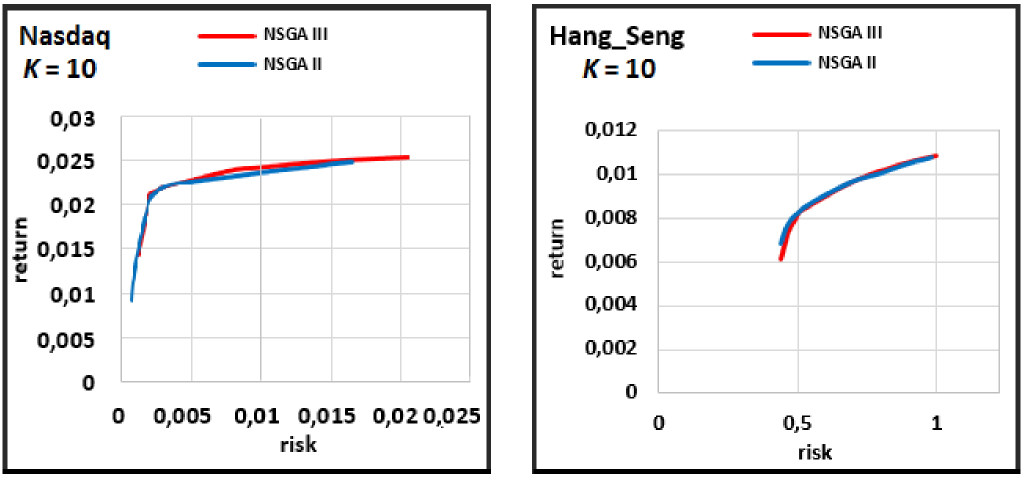
	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}
port 1	0	0	0	1	1	0	0	1	1	0	0	1	0	0	0	0
port 4	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1
port 7	1	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0

	p_{17}	p_{18}	p_{19}	p_{20}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{26}	p_{27}	p_{28}	p_{29}	p_{30}	p_{31}
port 1	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0
port 4	0	0	1	1	0	0	1	0	0	1	0	0	1	0	0
port 7	1	0	1	1	0	0	0	0	0	1	0	0	1	0	0

Table 3 – Return and Risk corresponding to the Efficient asset portfolios obtained by P-ACO for Hang Seng index.

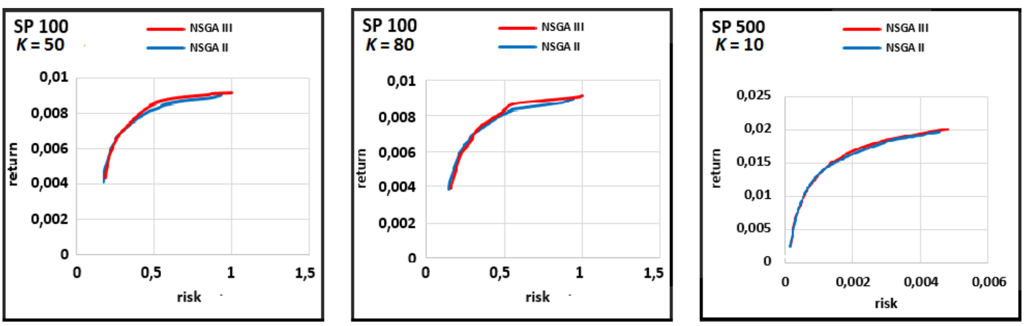
	Return	**Risk**
port 1	0.058008	54.094946
port 4	0.053634	50.604180
port 7	0.050428	49.152482

Two most important factors greatly impact the determination of these graphs. The first factor is the choice of the efficient asset portfolio that we need to incorporate into the NSGA II and NSGA III algorithms to find the proportional weights of the budget allocated to the portfolio. The second factor is the probability of crossover and mutation fixed in the two genetic algorithms. The portfolios with the lowest risk obtained by P-ACO for the Nasdaq index for each K value, the crossover probability $p_c = 0.90$ and the mutation probability $p_m = 0.7$ allowed us to draw the efficient frontier between the intervals $]0, 0.001]$ for risk and $]0.002, 0.022]$ for return. However, portfolios with the highest returns allowed us to draw the efficient frontier between the intervals $[0.002, 0.016]$ for risk and $[0.015, 0.025]$ for return. For SP500 index, the portfolios with the lowest risk allowed us to draw the frontier between the intervals $[0.00010, 0.0008]$ for risk and $[0.002, 0.0126]$ for return. The portfolios with the highest returns allowed us to draw the rest of the curve, with the probability of crossover and mutation equal to 0.90 and 0.70 respectively. For SP100, we chose only the portfolios with the highest return, but with a crossover probability p_c equal to 0.7, we got poor results and therefore reduced this probability to 0.2, which gave us much better results. For the mutation probability, we kept the same value $p_m = 0.7$. For the Hang-Seng index, we obtained three efficient asset portfolios using the P-ACO algorithm: port 1, port 2, and port 3 (see Table 2), but only port 1, which has the highest return, gave us a good frontier. The crossover and mutation probabilities are 0.2 and 0.70 respectively. We can say that portfolio and asset selection, crossover probability, and mutation probability played an essential role in exploring the research space, the diversity of solutions, and achieving good results. On the other hand, we can see from the frontiers of non-dominated solutions generated to SP500 and Nasdaq (Figures 1 to 3) that the NSGA III algorithm performs at least as well as NSGA II. Indeed, NSGA III algorithm performs better than NSGA II algorithm in the cases represented by



(a) (b)

Figure 2 – Non-dominated solutions of Nasdaq index for $K = 10$ and Hang_Seng index for $K = 10$.

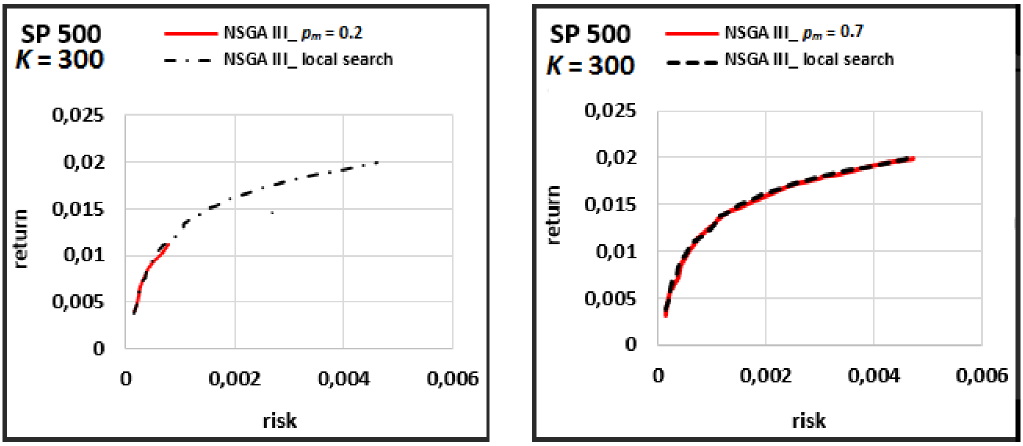


(a) (b) (c)

Figure 3 – Non-dominated solutions of SP 100 index for $K = 50$ and $K = 80$ and SP 500 index for $K = 10$.

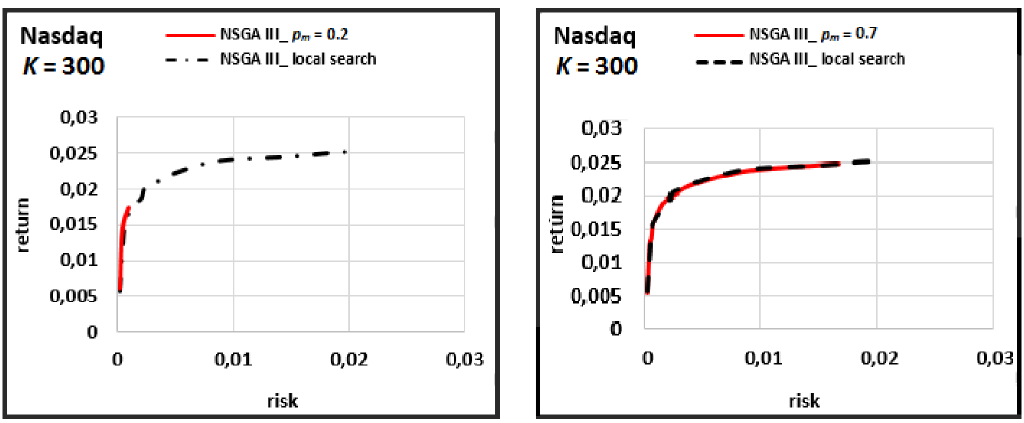
Figure 1, and the performances for the two algorithms are equivalent in the cases of Figures 2 and 3.

Mutation probability and local search : To improve the performance of NSGA III algorithm, we have tested different values of the mutation probability, we have also replaced the mutation method used before by another technique based on a local search algorithm. The results have been reported in Figures 4-9.



(a) (b)

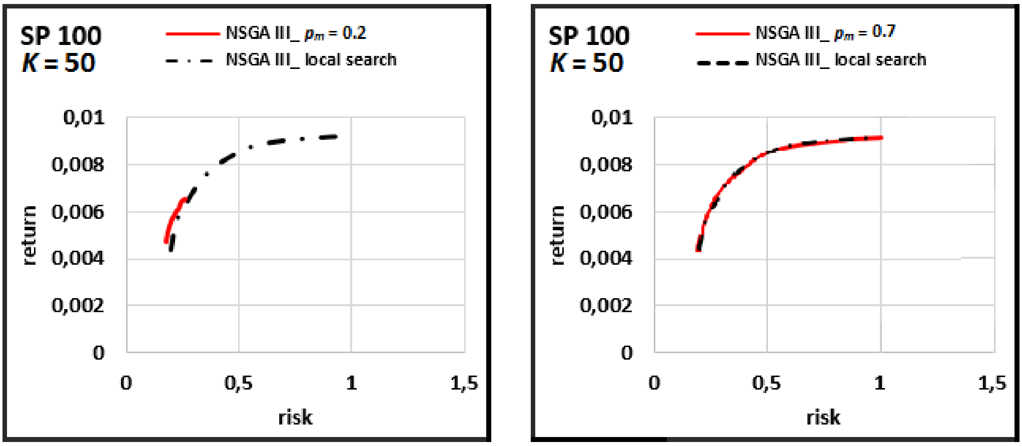
Figure 4 – Frontiers of non-dominated solutions obtained by NSGA III, using $p_m = 0.2$ and $p_m = 0.7$, with a local search, applied to SP500.



(a) (b)

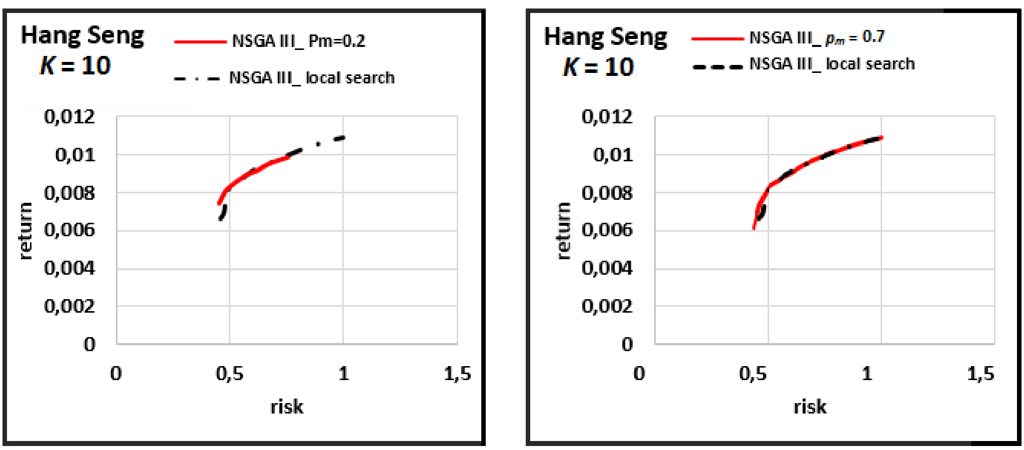
Figure 5 – Frontiers of non-dominated solutions obtained by NSGA III, using $p_m = 0.2$ and $p_m = 0.7$, with a local search, applied to Nasdaq.

Figures 4 to 7 show that when the mutation probability is high ($p_m = 0.7$), the new mutation operator based on the local search algorithm produces the same solution quality compared to the first mutation technique. However, if the mutation probability is small ($p_m = 0.2$), the local search technique is much better in terms of diversity. So we can say that the local search represents a good alternative because it does not require additional parameters (like the mutation probability) compared to the other mutation mechanisms.



(a) (b)

Figure 6 – Frontiers of non-dominated solutions obtained by NSGA III, using $p_m = 0.2$ and $p_m = 0.7$, with a local search, applied to SP 100.



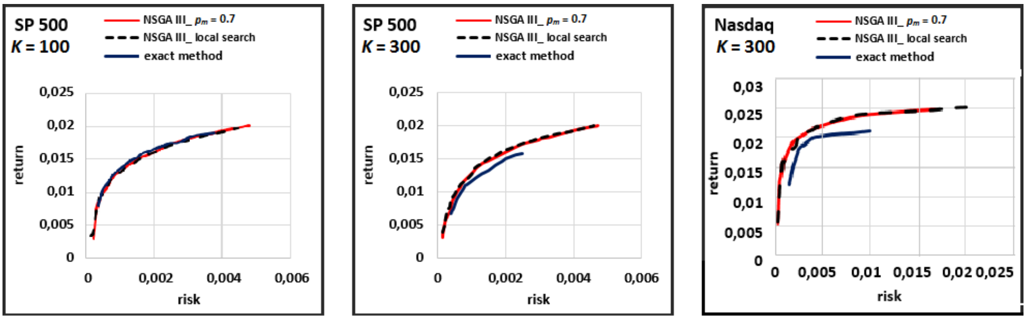
(a) (b)

Figure 7 – Frontiers of non-dominated solutions obtained by NSGA III, using $p_m = 0.2$ and $p_m = 0.7$, with a local search, applied to Hang Seng.

6 COMPARISON WITH AN EXACT METHOD

In this section, we compare the results obtained by NSGA III with those of an exact method used for the same problem.

Bezoui et al. (2018) proposed a variant of the Epsilon constraint method for solving a constrained portfolio optimization problem and presented the computational results obtained by performing experiments on a publicly available dataset. The study was conducted on seven benchmark



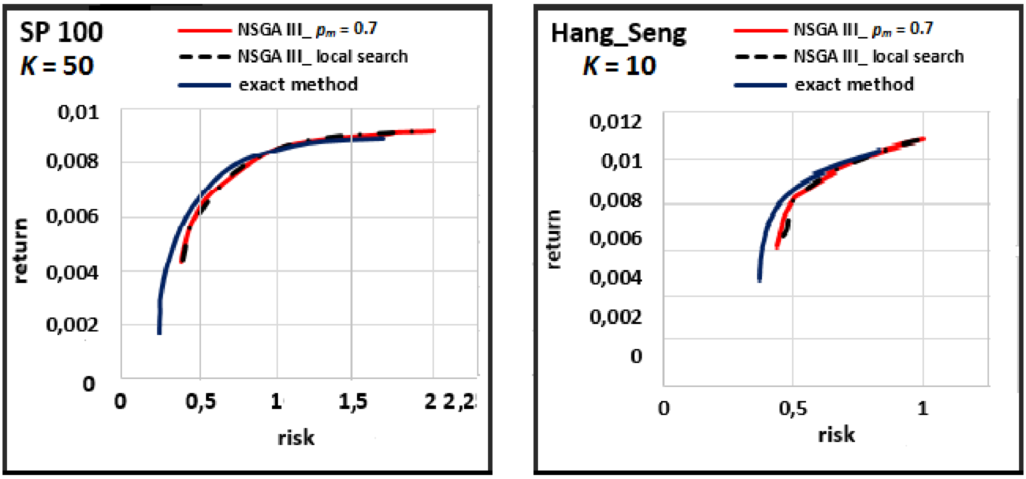
(a) (b) (c)

Figure 8 – Frontiers of non-dominated solutions obtained by the two approaches applied to SP500 for $K = 100$ and $K = 300$ and to Nasdaq for $K = 300$.

datasets. Five of them are available in the Beasley OR library Beasley (1990). These data provide the necessary input data for various assets in different stock market indices: Hong Kong Hang Seng with 31 assets, American S&P 100 with 98 assets, and two additional datasets described by Cesarone et al. (2014), available at Cesarone & Tardella (2017) (accessed January 12, 2017). These authors reported 263 weekly prices from March 2003 to March 2008 from the U.S. S&P 500 containing 476 assets and the European U.S. NASDAQ containing 2196 assets. We have tried to copy as much as possible the graphs showing the boundaries of the non-dominated solutions, obtained by the Epsilon constraints method, corresponding to the Hang Seng, SP100, SP500, and Nasdaq indexes Bezoui et al. (2018) and to compare them with our results obtained by applying the NSGA III genetic algorithm.

Figure 8 shows the efficient frontiers obtained by the two approaches for SP500 and Nasdaq according to two values of the cardinality: $100, 300$. Our approach is much better than the iterative approach. In fact, with $K = 300$ for SP500, the maximum return obtained by the iterative approach is less than 0.016 but with NSGA III we found portfolios with a value of return equal to 0.02. Similarly for Nasdaq, the maximum return obtained by NSGA III is 0.025, which is not the case with the iterative method.

Figure 9 shows the frontiers of the non-dominated solutions obtained by the two approaches: the genetic algorithm NSGA III and the iterative approach proposed by Bezoui et al. (2018). Both approaches were applied to SP100 index with a fixed cardinality value equal to 50 and to the Hang Seng index with a fixed cardinality value equal to 10. Thus, if we favor portfolios with an expected return belonging to the interval [0.008, 0.01], generated for Hang Seng, and belonging to the interval [0.005, 0.009], generated for SP100, we can say that the two approaches give the same results. Furthermore, the NSGA III can generate some portfolios with an expected return greater than 0.01079 for Hang Seng and than 0.009 for SP 100, which is not the case with the iterative approach. We also note that the iterative approach makes it possible to find portfolios with a low risk value.



(a) (b)

Figure 9 – Frontiers of non-dominated solutions obtained by the two approaches applied to SP100 for $K = 50$ and to HANG SENG for $K = 10$.

Finally, we can say that the combination of the ant colony optimization metaheuristic and the genetic algorithm NSGA III gives absolutely better results than the iterative approach for the SP500 index with two values for cardinality $K = 100$ and $K = 300$. We also have better results for the Nasdaq index with $K = 300$. For Hang Seng and SP100 our approach is only better if we focus on portfolios with a high return value, whereas the iterative approach obtained better solutions only if we favor portfolios with lower values of risk.

7 CONCLUSION

In this paper, we proposed two approaches: an optimization of the colony of ants metaheuristic that was used to select the best candidates of assets constituting the efficient portfolio that offers the best trade-off between risk and return under budget constraint, cardinality constraint, and pre-allocation constraint. Next, we used two versions of the Non-Dominated Sorting Genetic Algorithm (NSGA II and NSGA III) to find the proportions allocated to the selected portfolio and compare its performance to an exact method proposed by Bezoui et al. (2018).

Experimental results reveal that the proposed approaches can give a high quality of efficient portfolios compared to the exact method. The generated portfolios obtained by using our approaches can make a trade-off between return, risk and cardinality, which means that the combination of Ant colony optimization ACO and the Non-dominated Sorting Genetic Algorithm NSGA III is a good Multiobjective Approach to the Portfolio Optimization Problem.

As for future work, we are working to include the time factor in the model and to study this problem over several time periods

Acknowledgements

The authors thank the anonymous referees for their helpful comments and suggestions which contributed to the improvement of the contents of this paper.

The authors wish to thank the Directorate-General of the Scientific Research and the Technological Development of Algeria for its institutional support.

References

ANAGNOSTOPOULOS KP & MAMANIS G. 2010. A portfolio optimization model with three objectives and discrete variable. *Computers & Operations Research*, **378**(7): 1285–1297.

BEASLEY JE. 1990. OR-Library: Distributing Test Problems by Electronic Mail. *J Oper Res Soc*, **41**: 1069–1072.

BEZOUI M, MOULAÏ M, BOUNCEUR A & EULER R. 2018. An iterative method for solving a bi-objective constrained portfolio optimization problem. *Computational Optimization and Applications*, **72**(2): 479–498.

CESARONE F, SCOZZARI A & TARDELLA F. 2014. Linear vs. quadratic portfolio selection models with hard real-world constraints. *Computational Management Science*, **12**(3): 1–26.

CESARONE F & TARDELLA F. 2017. Data sets for portfolio selection problems. https://host.uniroma3.it/docenti/cesarone/datasetsw3_tardella.html.

CHANG T, MEADE N, BEASLEY JE & SHARAIHA YM. 2000. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, **27**(13): 1271–1302.

COELLO CAC, LAMONT GB & VELDHUIZEN DAV. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Second edition.

DAS I & DENNIS JE. 1998. Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM J. Optim.*, **8**(3): 631–357.

DEB K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley.

DEB K & JAIN H. 2014. An Evolutionary Many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput*, **18**(4): 577–601.

DEB K, PRATAP A, AGARWAL S & MEYARIVAN T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6**(2): 182–197.

DOERNER K, GUTJAHR WJ, HARTL R, STRAUSS C & STUMMER C. 2002. *Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection*. Kluwer Acadeemic Publishers.

DOERNER KF, GUTJAHR WJ, HARTL RF, STRAUSS C & STUMMER C. 2006. Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, **171**(3): 830–841.

DORIGO M & STUTZLE T. 2004. *Ant Colony Optimization*. MIT Press, Cambridge, MA.

FERNANDEZ PJ, D S LAURETTO M, D B PEREIRA CA & STERN JM. 2007. OR-Library: Distributing Test Problems by Electronic Mail. *Pesquisa Operacional*, **27**(3): 427–456.

GAMBARDELLA LM, TAILLARD ED & AGAZZI G. 1999. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: *New Ideas in Optimization*. pp. 63–76.

GHOSH A & DAS MK. 2008. Non-dominated Rank based Sorting Genetic Algorithms. *Fundamenta Informaticae*, **83**(231–252).

GRAVEL M, PRICE WL & GAGNE C. 2002. Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic. *European Journal of Operational Research*, **143**(1): 218–229.

IBÁÑEZ ML. 2004. *Multi-objective Ant Colony Optimization*. Ph.D. thesis. Technische Universität Darmstadt, Germany, and Universidat de Granadan Spain.

KRINK T & PATERLINI S. 2008. *Differential evolution formultiobjective portfolio optimization*.

KUMAR D & MISHRA KK. 2017. Portfolio optimization using novel co-variance guided Artificial Bee Colony algorithm. *Swarm and Evolutionary Computation*, **33**(C): 119–130.

LIU Q, LIU X, WU J & LI Y. 2019. An Improved NSGA-III Algorithm Using Genetic K-Means Clustering Algorithm. *IEEE Access*, **7**: 185239–185249.

LIU S & XIAO C. 2021. Application and Comparative Study of Optimization Algorithms in Financial Investment Portfolio Problems. *Mobile Information Systems*, **2021**: 3462715.

LWIN K & KENDAL RQG. 2014. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, **24**: 757–772.

MACEDO LL, GODINHO P & ALVES MJ. 2017. Mean-Semivariance Portfolio Optimization with Multiobjective Evolutionary Algorithms and Technical Analysis Rules. *Expert Systems With Applications*, **79**: 33–43.

MANSINI R & SPERANZA MG. 1999. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *Eur. J. Oper. Res.*, **114**(2): 219–233.

MARKOWITZ HM. 1952. Portfolio Selection. *The Journal of American Finance Association*, **7**(1): 77–91.

MARKOWITZ HM. 1991. *Portfolio Selection: Efficient Diversification of Investments*. 2nd ed., MA: Basil Blackwellk.

METAXIOTIS K & LIAGKOURAS K. 2012. Multiobjective Evolutionary Algorithms for Portfolio Management: A comprehensive literature review. *Expert Systems with Applications*, **39**(14): 11685—11698.

SMITH P, FERRINGER M, KELLY R & MIN I. 2012. Budget-constrained portfolio trades using multiobjective optimization. *Systems Engineering*, **4**(15): 461–470.

STREICHERT F, ULMER H & ZELL A. 2004. Comparing Discrete and Continuous Genotypes on the Constrained Portfolio Selection Problem. In: *Genetic and Evolutionary Computation – GECCO 2004*. pp. 1239–1250. Springer Berlin Heidelberg.

How to cite

IKHLEF M & AÏDER M. 2023. Multiobjective Evolutionary Metaheuristic Approach to the Constrained Portfolio Optimization Problem. *Pesquisa Operacional*, **43**: e266962. doi:10.1590/0101-7438.2023.043.00266962.