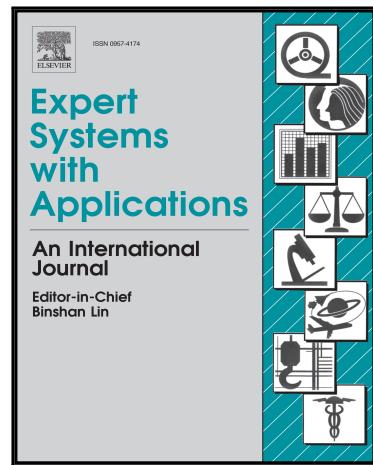


Accepted Manuscript

A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems

Yuri Laio T.V. Silva, Ana Beatriz Herthel, Anand Subramanian

PII: S0957-4174(19)30345-8
DOI: <https://doi.org/10.1016/j.eswa.2019.05.018>
Reference: ESWA 12671



To appear in: *Expert Systems With Applications*

Received date: 27 February 2019
Revised date: 21 April 2019
Accepted date: 14 May 2019

Please cite this article as: Yuri Laio T.V. Silva, Ana Beatriz Herthel, Anand Subramanian, A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems, *Expert Systems With Applications* (2019), doi: <https://doi.org/10.1016/j.eswa.2019.05.018>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- We solve several variants of the portfolio selection problem with a unified approach
- A novel adaptive ranking procedure based on three mechanisms is introduced
- Extensive computational experiments show the efficiency and robustness of the method

A multi-objective evolutionary algorithm for a class of mean-variance portfolio selection problems

Yuri Laio T. V. Silva^{a,c}, Ana Beatriz Herthel^a, Anand Subramanian^{b,*}

^a*Universidade Federal da Paraíba, Departamento de Engenharia de Produção, Centro de Tecnologia, Campus I, Cidade Universitária, João Pessoa-PB, CEP 58059-900, Brazil.*

^b*Universidade Federal da Paraíba, Departamento de Sistemas de Computação, Centro de Informática, Mangabeira, Rua dos Escoteiros s/n, CEP 58055-000, João Pessoa, Brazil.*

^c*Centro Universitário de João Pessoa, BR 230, 22, Água Fria, João Pessoa-PB, CEP 58053-000, Brazil.*

Abstract

The portfolio selection problem (PSP) concerns the resource allocation to a finite number of assets. In its classic approach, the problem aims at overcoming a trade-off between the risk and expected return of the portfolio. In recent years, additional constraints identified in financial markets have been incorporated into the literature, as an attempt to close the gap between theory and practice. In view of this, this paper introduces a unified multi-objective particle swarm optimization approach capable of solving a class of mean-variance PSPs. An adaptive ranking procedure is also developed, which is based on three mechanisms, including a new one. Extensive computational experiments were carried out in five PSP variants and the results obtained were compared with those found by problem-specific methods from the literature. The proposed approach was capable of finding highly competitive results in all problems and in most of the multi-objective metrics considered.

Keywords: Portfolio selection problem. Mean-variance. Multi-objective optimization. Particle swarm optimization. Unified algorithm.

1. Introduction

The portfolio selection problem (PSP) concerns the resource allocation to a finite number of assets. It is currently one of the most important topics regarding economical and financial issues. Due to the existing trade-off between investments risks and returns, the main purpose

*Corresponding author

Email addresses: yurilaio@gmail.com, yuri.laio@unipe.edu.br (Yuri Laio T. V. Silva), abherthel@gmail.com (Ana Beatriz Herthel), anand@ci.ufpb.br (Anand Subramanian)

5 of the portfolio selection is to decide in which assets to invest and in which proportions to do so, considering the amount of capital available.

Different returns are associated with various risk levels and an investor's risk acceptance is changeable. It can be influenced by social status, age or future prospects, therefore, there is not a common portfolio suitable to all investors ([Skolpadungket et al., 2007](#)). In light of
10 that, optimization presents itself as a tool to increase investors profitability, helping in the decision-making process as it takes into account their investment goals.

One of the first operations research-based approaches proposed for the PSP was the one by [Markowitz \(1952\)](#), and it was denoted as mean-variance. The formulation aims at maximizing the return and minimizing the risk of the portfolio. This model can also be
15 described as a quadratic programming (QP) problem (QPP), in which the portfolio risk is minimized and the lowest return acceptable is represented by means of a linear constraint. Markowitz's mean-variance model was essential for the dissemination of the investment portfolio optimization problem and its extensive studying.

Problems involving two objectives that cannot be simultaneously optimized do not have a
20 unique optimal solution. There is, in fact, a set of efficient solutions which is usually denoted as efficient frontier, or Pareto front. [Branke et al. \(2009\)](#) claim that the main motivation in achieving an efficient frontier is the importance of obtaining solutions bearing different trade-offs according to the optimization mechanisms. This allows for an optimal solution to be achieved, maximizing the expected return, based on a certain risk factor provided by the
25 investor. This frontier is obtained by solving QPP for all risk and return trade-offs.

Several exact and heuristic approaches have been employed to solve PSPs. Generally, they aim at assessing a large number of possible combinations, thus potentially preventing the problem to be solved within an acceptable CPU time ([Skolpadungket et al., 2007](#)). The issue might become even more complex when incorporating other constraints such as floor,
30 ceiling and cardinality constraints, giving rise to variants that are known to be \mathcal{NP} -hard ([Moral-Escudero et al., 2006](#)).

As a result of the high competitiveness of the financial market, there has been a growing attention in considering certain factors when approaching the PSP. [Metaxiotis & Liagkouras \(2012\)](#) pointed out that approximately 96.5% and 68.7% of the PSP studies considered up
35 to four and two constraints, respectively. Nevertheless, in observing their analysis, one can notice that if we take into account the last ten years, works with the previous characteristics dropped their percentages to 88.5% and 45.7%, respectively. This shows a trend in adding external factors when solving PSPs, as also discussed in [Kolm et al. \(2014\)](#).

Given the above, with the increasing amount of constraints to be considered, there is a broad array of PSP versions, so that they can be tailored according to the investor's current objectives and strategies. However, the vast majority of existing solution approaches focused on a particular PSP variant. In fact, there is clearly a gap in the literature when it comes to devising generic yet efficient PSP algorithms. Hence, the development of a unified method with the ability of tackling several variants is highly relevant, so that it can be applied in actual market situations.

The main contributions of this work are as follows.

- We propose a unified multi-objective particle swarm optimization (MOPSO) algorithm capable of solving a class of mean-variance PSPs. To our knowledge, this is one of the first algorithms designed to cope with different versions of this type of problems. This enables the investors to apply the developed methodology to solve problems that include a subset of financial market constraints of their particular interest. Extensive computational experiments were carried out in five variants, and the results obtained show that the proposed algorithm produces highly competitive results when compared to methods specifically devised to solve a particular variant.
- We introduce an adaptive ranking procedure that is based on three mechanisms: non-dominated sorting (Deb et al., 2002), crowding distance (Deb et al., 2002) and a new one called cost-benefit. We empirically evaluate the gains of incorporating such procedure and the results suggest that the newly proposed strategy improves the performance of the MOPSO algorithm based on a standard non-dominated sorting implementation.

The remainder of this work is organized as follows. Section 2 briefly explains multi-objective optimization. Section 3 describes the PSPs considered, including the most relevant constraints that appear in the literature. Section 4 reviews the related works. Section 5 introduces the proposed method, including the new ranking procedure. Section 6 presents the results of the computational experiments. Section 7 provides the concluding remarks.

2. Multi-objective optimization

In this section we define a general multi-objective (MO) problem of the form:

$$\min f(x) = (f_1(x), \dots, f_k(x)) \quad (1)$$

$$x \in \mathcal{X}, \quad (2)$$

where $\mathcal{X} = \{x \in \mathbb{R}^n : h(x) = 0, g(x) \geq 0\}$ is the set of feasible solutions and $f(x)$ is a vector composed of k conflicting objective functions. Such conflicting nature implies that there is not a feasible solution that minimizes all the objectives at once. For this reason, one seeks to
70 find a set of *efficient solutions*. We present below some definitions that enable us to define these solutions.

When solving a MO problem, solutions are often assessed according to a Pareto dominance relation. Let $x_1, x_2 \in \mathcal{X}$ be two feasible solutions obtained from the MO problem described previously. A vector $f(x_1)$ is said to dominate a vector $f(x_2)$ if, and only if:

$$f_i(x_1) \leq f_i(x_2) \quad \forall i = 1, \dots, k \quad (3)$$

$$f_i(x_1) < f_i(x_2) \quad \exists i = 1, \dots, k \quad (4)$$

This relation is denoted by \prec . In the case where only the (3) is satisfied, a relation \preceq arises.

A solution $x^* \in \mathcal{X}$ is said to be efficient if there does not exist any other feasible solution $x \in \mathcal{X}$ such that $f(x) \prec f(x^*)$.

75 The image of the solution x^* in the objective space \mathbb{R}^k is called a *non-dominated point*. In turn, the image of the efficient set (set containing all the efficient solutions of the problem) is the so-called Pareto front.

For further concepts on MO optimization, the interested reader is referred to [Ehrgott \(2005\)](#). Moreover, comprehensive surveys on MO methods and on MO evolutionary-based
80 techniques are presented in [Marler & Arora \(2004\)](#) and [Coello \(1999\)](#), respectively.

3. Mean-variance portfolio selection problems

This model was proposed by [Markowitz \(1952\)](#) and it consists of a bi-objective optimization problem with two functions representing return and risk regarding the portfolio. The variance measures the risk of the portfolio and it is minimized by objective function (5).

85 The expected return, on the other hand, is maximized through (6).

The problem constraints (7) and (8) limit the investments to 100% of the available amount, and that these investments cannot be lower than 0, respectively.

$$z_1 = \min \sum_{j \in A} \sum_{k \in A} w_j w_k \sigma_{jk} \quad (5)$$

$$z_2 = \max \sum_{j \in A} w_j r_j \quad (6)$$

Subject to:

$$\sum_{j \in A} w_j = 1 \quad (7)$$

$$w_j \geq 0, j \in A. \quad (8)$$

- Set A represents the number of assets available for the portfolio composition and σ_{jk} indicates the covariance between assets j and k . The investment proportion for each asset is represented by w_j , while r_j is the average return of asset j . Here the return on assets are considered as random variables, whose expected values and covariance matrix are known.
- 90 This leads to an important observation regarding the inputs: there might be parameter uncertainty or estimation errors because the actual model parameters are unknown and have to be estimated from data. In the current literature, the impact of estimation errors in a mean-variance portfolio optimization context is well understood. In fact, there is vast literature on how to mitigate the impact of estimation errors (see e.g. [Kan & Zhou 2007](#)).
- 95 Nevertheless, it has been shown that imposing certain constraints on portfolio weights can be equivalent to reducing or increasing the estimated covariances (see [Jagannathan & Ma 2003](#)). In addition, constraints on the risk contributions of individual assets in a portfolio can also mitigate the impact of estimation errors.

- To generate more realistic models, one can consider additional constraints related to
100 market preferences, resource allocation, transaction costs and cardinality are considered. However, some constraints might cause PSP to be potentially unsolvable by exact methods, even for small instances, as is stated in [Mishra \(2012\)](#).

3.1. Additional constraints

- In what follows, we present the additional constraints that may appear in the different
105 PSP variants solved by the proposed algorithm.

3.1.1. Cardinality constraints (CC and IC)

The cardinality constraints (CC) (9) are among the most commonly studied ones in the literature. Along with floor and ceiling constraints, they are among the major considerations made by investors in the decision-making process. These constraints impose limitations on the amount of assets in a portfolio:

$$\sum_{j \in A} z_j = K. \quad (9)$$

In this model, z_j represents a binary variable that assumes value 1, if the asset j is present in the portfolio, and 0, otherwise. K represents the amount of assets that must compose the portfolio. Most studies deal with cardinality as depicted in Equation (9). However it can be approached differently. Some works consider the cardinality constraint as an inequality (IC). They define lower (K_{min}) and upper (K_{max}) bounds (10) for the number of assets in the portfolio:

$$K_{min} \leq \sum_{j \in A} z_j \leq K_{max}. \quad (10)$$

3.1.2. Floor and ceiling constraints (FC)

FC constraints define lower (ϵ_j) and upper (δ_j) bounds on the investment proportion for each asset, as shown in Equation (11). Generally, they represent the investor's preferences to certain assets, sets of assets or to economic sectors.

$$\epsilon_j z_j \leq w_j \leq \delta_j z_j, j \in A \quad (11)$$

3.1.3. Round-lot constraints (RL)

The round-lot constraints (12) indicate that certain assets can only be acquired in multiples of a certain amount, i.e., in lots. Hence, the investment proportion per asset is equal to $l_j y_j$ where l_j represents the minimum investment proportion of asset j that must be acquired, and y_j is the integer number indicating the acquired amount of lots of asset j (Ponsich et al., 2013).

$$w_j = l_j y_j, \forall j \in A, y_j \in Z_+ \quad (12)$$

Several studies included the round-lot constraints in their models, especially recent ones
 110 (Skolpadungket et al., 2007; Soleimani et al., 2009).

3.1.4. Pre-assignment constraints (PA)

The PA constraints (13) indicate an investor's preference towards a particular set of assets (U) that must be selected for investment. Therefore, for each asset, there is an associated binary input u_j . In this case, $u_j = 1$ indicates that asset j must be selected for investment, whereas if $u_j = 0$, j may not necessarily be selected.

$$z_j \geq u_j, \forall j \in A \quad (13)$$

Note that the pre-assignment of assets limits the range of possibilities of the portfolio, especially when combined with cardinality and FC constraints as it reduces the remaining number of assets to be invested. From the financial viewpoint, in case the pre-assigned asset
115 is from a certain market sector, the risk is likely to be prominent if the decision variables indicate that there are more investments in the sector of such asset. Conversely, in case the sectors are more diversified, the risk is likely to be lower.

3.2. Problems considered

The PSP variants dealt in this work are summarized in [Table 1](#). For all of them, the risk
120 measurement is done according to the classic well-known model, which is based on variance and covariance of assets.

Table 1: PSPs considered

PSP	Additional Constraints				
	CC	IC	FC	RL	PA
Unrestricted PSP (UPSP) or mean-variance PSP					
Cardinality constrained PSP (CCPSP)	✓			✓	
Inequality cardinality constrained PSP (ICCPSP)		✓		✓	
Inequality cardinality pre-assignment constrained PSP (ICPAC-PSP)		✓		✓	✓
Cardinality round-lot pre-assignment constrained PSP (CRLPAC-PSP)	✓		✓	✓	✓

4. Literature review

The mean-variance PSP, as proposed by [Markowitz \(1952\)](#), is still the most studied variant of the problem. However, in the past decades, many authors have been considering
125 several additional constraints and risk measurement approaches.

[Speranza \(1996\)](#) implemented a heuristic procedure for a PSP variant with several constraints. [Chang et al. \(2000\)](#) used three different methods to solve CCPSP and the results achieved for the instances of OR-Library [Beasley \(1990\)](#) were promising at the time. They also concluded that the addition of CC implies a non-continuous efficient frontier.

[Lin & Wang \(2002\)](#) implemented a MO genetic algorithm (GA) for PSP with RL and transaction costs constraints. The latter constraints usually aim at quantifying the amount of bought or sold assets between two periods of time. [Schaerf \(2002\)](#) employed local search (LS) techniques to solve CCPSP. [Crama & Schyns \(2003\)](#) proposed a SA for a PSP variant with CC, FC, turnover and trading constraints. Turnover constraints limit the maximum and
135 minimum amount of purchase and sale of an asset j in the period t through $t + 1$. Trading constraints restrict the maximum and minimum purchase and sale values of a particular

asset, regardless of the period or its current investment proportion. Maringer & Kellerer (2003) devised a hybrid approach for CCPSP, with generic FC bounds (0 and 1).

Armananzas & Lozano (2005) applied three approaches for CCPSP: greedy LS (GLS), 140 SA and ant colony optimization (ACO). Moral-Escudero et al. (2006) implemented a hybrid genetic algorithm (HGA), combining GA and QP, to solve the same variant. Streichert & Tanaka-Yamawaki (2006) devised a MO memetic algorithm to solve CCPSP by combining a MO evolutionary algorithm (MOEA) and a QP-based LS.

Fernández & Gómez (2007) approached CCPSP by means of a neural network (NN) 145 heuristic. Lin & Liu (2008) developed methods based on GA for a PSP with RL constraints. This problem was also approached by Chiam et al. (2008), who proposed a MOEA with order-based representation. Branke et al. (2009) put forward a procedure called envelope-based MOEA (e-MOEA) to solve CCPSP. Cura (2009) implemented a particle swarm optimization (PSO) algorithm for the same problem.

CCPSP was also solved by Pai & Michel (2009) by means of a MOEA, and by Soleimani 150 et al. (2009) using a GA considering several additional constraints. Anagnostopoulos & Mamanis (2009) developed a non-dominated sorting genetic algorithm (NSGA) to solve the same variant. Later, Anagnostopoulos & Mamanis (2010) presented the PSP as a tri-objective optimization problem, involving expected return, risk and cardinality. The authors implemented three evolutionary algorithms and later proposed another five for CCPSP Anagnostopoulos & Mamanis (2011).

Zhu et al. (2011) presented a PSO for the classic PSP. Chen & Zhang (2010) approached a PSP with transaction costs and FC constraints. Woodside-Oriakhi et al. (2011) implemented 160 three algorithms (GA, TS and SA) for CCPSP and studied the performances of their efficient fronts. Deng & Lin (2010b) devised a PSO for the CCPSP. Golmakani & Fazel (2011) also proposed a PSO for the same problem, but with RL and sector capitalization constraints. The latter indicates the investor's preference in investing on assets from sectors with a higher market value, in order to reduce risks. In another work, Deng & Lin (2010a) proposed an ACO-based algorithm for the same variant. Two years later, Deng et al. (2012) implemented 165 an improved PSO for CCPSP. The results indicated a superior performance of their method. This problem was later reformulated as a standard QPP by Cesarone et al. (2013).

CCPSP was also approached by Liagkouras & Metaxiotis (2014) who proposed a procedure called probe guided mutation. Mishra et al. (2014a) put forward a MOPSO approach for the same variant. In another work, Mishra et al. (2014b) developed an algorithm based 170 on MOEA for a CCPSP with floor constraints. Salahi et al. (2014) proposed two heuristic

approaches for this variant, based on PSO and harmonic search (HS). [Sabar & Kendall \(2014\)](#) also applied HS to solve the same problem. [Baykasoglu et al. \(2015\)](#), on the other hand, used, for the first time, a greedy randomized adaptive search procedure (GRASP) for CCPSP.

¹⁷⁵ [Sadigh et al. \(2012\)](#) proposed a hybrid approach for CCPSP, combining PSO and NN. Similarly, [Lwin & Qu \(2013\)](#) associated EA and differential evolution, while [Corazza et al. \(2013\)](#) combined mixed-integer QP (MIQP) and PSO.

Some multi-criteria approaches have also been proposed. For [Xidonas et al. \(2011\)](#), their method was based on decision support system (DSS) and it was employed on a CCPSP with ¹⁸⁰ sector capitalization constraints. This variant was also solved by [Lim et al. \(2014\)](#) through a multi-criteria approach based on data envelopment analysis (DEA).

In the last few years, PSP variants considering portfolio rebalancing and multiperiod approaches have been more prominent. [Woodside-Oriakhi et al. \(2013\)](#) worked with a rebalancing problem regarding an existing portfolio. They considered V-Shaped transaction costs ¹⁸⁵ for each alteration in asset investment proportions. [Guo et al. \(2016\)](#) preferred to develop a GA to solve it.

[Bertsimas & Pachamanova \(2008\)](#) and [Liu et al. \(2015\)](#) used a robust optimization approach for the classic multiperiod PSP, whereas [Palczewski et al. \(2015\)](#) worked on the same problem with transaction costs. Recently, [Ruiz-Torrubiano & Suárez \(2015\)](#) approached the ¹⁹⁰ multiperiod CCPSP with transaction costs and trading constraints.

[Chen et al. \(2010\)](#) put forward a PSO for the PSP with fuzzy probabilities, whereas [Noroozi et al. \(2016\)](#) introduced the application of a probabilistic model (CVX) to the classic version of the problem with random and uncertain return.

In addition to the thorough review concerning deterministic models and applications for ¹⁹⁵ mean-variance portfolio optimization by [Kalayci et al. \(2019\)](#), the reader is also referred to the following surveys regarding specific type of methodologies proposed for PSPs: (i) MOEAs ([Metaxiotis & Liagkouras, 2012](#); [Ponsich et al., 2013](#)); (ii) linear programming-based approaches ([Mansini et al., 2014](#)); (iii) metaheuristics ([Jarraya, 2013](#)); and (iv) goal programming ([Azmi & Tamiz, 2010](#); [Aouni et al., 2014](#)). Moreover, some practical challenges ²⁰⁰ and potential trends concerning portfolio optimization were reviewed in [Kolm et al. \(2014\)](#).

5. Adaptive ranking multi-objective particle swarm optimization

As shown in Section 4, evolutionary algorithms have been widely employed to solve several PSP variants. One of these approaches, PSO, is a population-based metaheuristic

that simulates the social behavior of a swarm of particles in search for food. In this method,
205 the position of each particle represents one solution to the problem. Such particles may move in a multidimensional search space through velocity vectors, which are dynamically adjusted after each iteration (Kennedy et al., 2001; Deng et al., 2012).

At each iteration t , out of max_iter iterations, the i th particle of the swarm (population) P has its position $x_{i,j}^t$ updated through its velocity $v_{i,j}^{t+1}$ for all j dimensions. Thus, the position of the next iteration is updated according to Equation (14):

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, \quad (14)$$

where $x_{i,j}^{t+1}$ and $v_{i,j}^{t+1}$ represent, respectively, the position (solution) and the velocity of the particle i in dimension (asset) j of the search space at iteration $t + 1$. The velocity of the
210 particle is responsible for changing its position over the iterations and it can be obtained by means of Equation (15)

$$v_{i,j}^{t+1} = Wv_{i,j}^t + C_1r_1(p_{i,j} - x_{i,j}^t) + C_2r_2(p_{g,j} - x_{i,j}^t). \quad (15)$$

In this equation, $p_{i,j}$ indicates the best local position that particle i has already been in the search space. On the other hand, $p_{g,j}$ indicates the best global position, that is, the best position ever achieved by any particle since the first iteration. Since we are dealing with a
215 MO problem, determining $p_{i,j}$ and $p_{g,j}$ at each iteration is not a trivial task. Due to its large influence on the general performance of the method, the study of how to perform such task has received a special attention from several authors in recent years (Mostaghim & Teich, 2003; Qu et al., 2013; Rezaee Jordehi & Jasni, 2013).

Moreover, W corresponds to the inertial weight, which is responsible for weighing the
220 influence of previous iteration velocities while updating the current positions. It is important for converging the search. Values r_1 and r_2 are randomly obtained through an uniform distribution and they are, traditionally, between 0 and 1. These values represent the stochastic factor of the algorithm. Parameters C_1 and C_2 express particles' acceleration coefficients. They are used to weight the contribution of cognitive and social components. In this context,
225 C_1 indicates the confidence of a particle based solely on its position, whereas C_2 expresses the confidence of the particle regarding the other ones (Deng et al., 2012). Such values can be used and updated in different ways, depending on the strategy employed to solve the problem. Regarding the values of C_1 and C_2 , Shi & Eberhart (1998) concluded that promising values for these parameters lie between 1.90 and 2.10, whereas Eberhart & Shi

²³⁰ (2000) concluded that values around 0.90 appear to be promising. After performing some preliminary experiments, we selected the following values for these parameters: W varying from 0.95 to 0.85 decreasing in proportions of $\frac{0.10}{max_iter}$, at each iteration; C_1 and C_2 were defined as 2.05 and 1.95, respectively. Parameter max_iter was set to 500, as it is a standard value used in the PSO literature (see, e.g., Shi & Eberhart 1998; Handoyo et al. 2017).

²³⁵ In light of that, we developed a MOPSO that makes use of a newly proposed procedure based on three mechanisms: non-dominated sorting (NS) (Deb et al., 2002), crowding distance (CD) (Deb et al., 2002) and a new one called cost-benefit (CB) to perform the ranking of particles and the selection of the best global position.

²⁴⁰ In what follows, we describe the building blocks of the proposed algorithm, called adaptive ranking MOPSO (ARMOPSO). Next, we show how they are integrated by means of a pseudocode.

5.1. Initialization

²⁴⁵ Once the PSP is identified, the initialization randomly generates an initial swarm of particles for the problem, considering its objectives and constraints, i.e., the initial swarm of particles are always composed by feasible solutions.

²⁵⁰ In the case of CCPSP, the initialization procedure must respect the exact amount of assets in the portfolio (K_{eq}), whereas for ICCPSP, it must take into account the minimum (K_{min}) and maximum (K_{max}) amount of assets. As for ICPAC-PSP and CRLPAC-PSP, the PA constraints are satisfied by enforcing the amount invested in the corresponding pre-specified assets to be strictly greater than zero. Moreover, the procedure makes sure that RL constraints arising in CRLPAC-PSP are satisfied by employing the feasibility mechanism described in Section 5.5. Of course, the FC constraints are also respected in all cases when generating the initial swarm.

As shown in Section 5.6, the initialization is called each time the method restarts.

255 5.2. Evaluation

The evaluation phase seeks to determine the aptitude of the particles in the swarm. It calculates, for each particle, the values obtained for the m objectives of the problem. From these calculations, the set of non-dominated solutions (H) is updated, meaning that the solutions that will compose the Pareto front are stored. Therefore, after each iteration, if ²⁶⁰ any particle in the current swarm is not dominated by any other element of H , it is then added to this set. However, if any particle dominates any other one of H , the dominated ones will be removed from H .

Because our method was designed to solve a class of PSPs, and due to the various ways the results are reported in the literature, we also considered an alternative evaluation function, given by (16), to assess the quality of the particles:

$$f(p) = \lambda \left[\sum_{j \in A} \sum_{k \in A} w_j w_k \sigma_{jk} \right] - (1 - \lambda) \sum_{j \in A} w_j r_j, \quad (16)$$

where $\lambda \in [0, 1]$ is a value that is updated each time the method restarts. More precisely, λ begins with value 0 and it increases by 0.02 at each restart until it reaches value 1, resulting in 50 restarts. For each of them, all particles of H are evaluated through Equation (16). The one to obtain the smallest value is stored in a set denoted as V . Hence, the number of particles in V is limited to the amount of iterations that λ is updated. Once all restarts are performed, V is possibly updated so as to only keep the non-dominated solutions.

5.3. Proposed adaptive ranking procedure

The particle ranking procedure employed throughout the algorithm combines NS, CD and CB. NS is a very common scheme in MO genetic algorithms, and it was proposed by Deb et al. (2002). Mishra et al. (2014a) were the first to incorporate NS in a PSO, obtaining competitive results.

The proposed adaptive ranking procedure has two functions: (i) updating H by selecting which particles will be removed from the set, if the predefined particle limit is exceeded; (ii) determining the best global position on each iteration.

5.3.1. Updating H

The procedure starts with NS. Given a set of feasible solutions, the mechanism builds all possible Pareto fronts. Next, it sorts the frontiers in hierachic levels, based on their particular characteristics. Figure 1 depicts the procedure for a set of 13 random solutions.

The frontiers obtained by NS are used in the CB and CD mechanisms which will evaluate their corresponding points. From these assessments, two different rankings are built, which will serve as basis for obtaining the final ranking of solutions, as depicted in Figure 2. P_1 represents the current positions of the swarm, whereas P_2 represents H . Set F corresponds to the non-dominated frontiers generated after applying NS, F' denotes the frontiers re-sorted after applying CD and CB, and F^R represents the final ranking.

The CD value for a particle i , given by cd_i , is determined based on the distance between particle i and the two closest ones, $i + 1$ and $i - 1$. The mechanism allows for estimating the particle density around a certain particle, as shown in Figure 3.

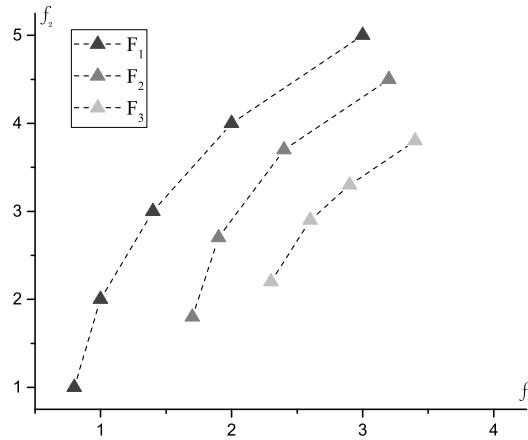


Figure 1: Non-dominated sorting

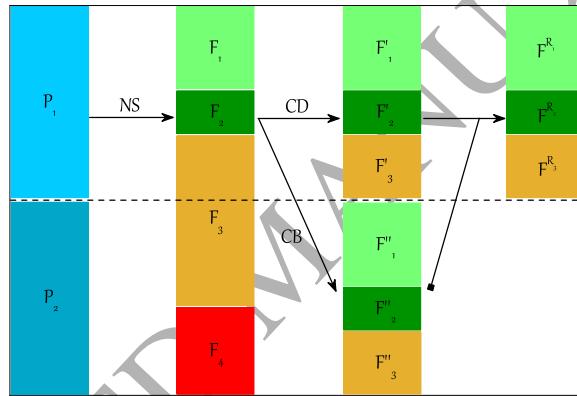


Figure 2: Procedure of ranking solutions

Initially, H is sorted in ascending order according to the values of each objective function (m). Next, the distance between the adjacent solutions $d_{i,m}$ is calculated for each particle and objective function value by means of Equation (17), and the final value of cd_i can be computed as shown in Equation (18). Lastly, RK_1 is generated by sorting the CD values obtained for each particle in ascending order.

$$d_{i,m} = \frac{f_{i+1,m} - f_{i-1,m}}{f_m^{\max} - f_m^{\min}} \quad (17)$$

$$cd_i = \sum_{m=1}^M d_{i,m}, \forall i \in H, \quad (18)$$

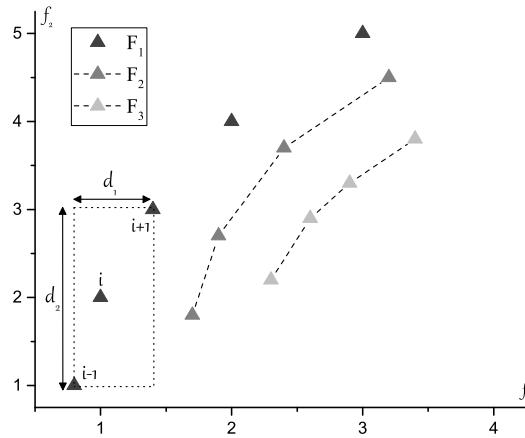


Figure 3: Crowding distance

where $f_{i,m}$ denotes the values achieved by solution i in each objective m , while f_m^{\max} and f_m^{\min} are the maximum and minimum values, respectively, achieved in H .

Now the proposed mechanism, CB, is used to generate a second and independent ranking (RK_2). It aims at selecting the best particles based on the cost-benefit relationship between the values obtained for the objective functions. The cost-benefit cb_i of each particle i is obtained by means of the ratio between the expected return and the risk associated with the position of the particle in the search space, as presented in Equation (19), where $f_{i,1}$ is the expected return and $f_{i,2}$ is the associated risk. A higher value of cb_i indicates a larger proximity of particle i to the optimal frontier. RK_2 is thus obtained by sorting the CB values computed for each particle in descending order. Figure 4 illustrates the process of determining cb_i .

$$cb_i = \frac{f_{i,1}}{f_{i,2}}, \forall i \in H \quad (19)$$

Once both RK_1 and RK_2 are determined, a final ranking procedure is built as follows. Each particle i receives a score associated with its position on each rank. These scores are then added and a final score is achieved. Next, the particles are sorted in ascending order of score. If two or more particles are tied in the ranking, the one with a higher CB value is assumed to be the better one. In case the predetermined size of H is surpassed, the particles ranked beyond this threshold are removed from H . The maximum size of H was set to 500.

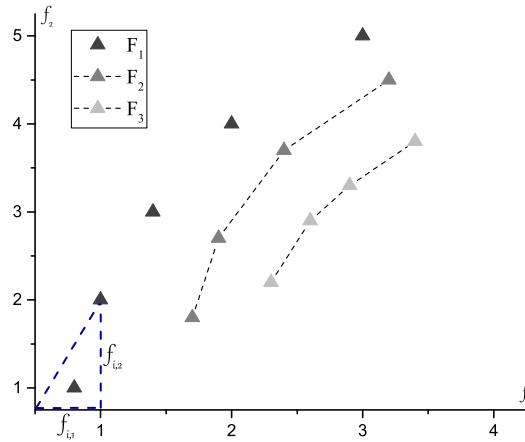


Figure 4: Cost-benefit

5.3.2. Selection of best local and global positions

By definition, the best local position (solution) of each particle represents the best position already assumed by that particle. Throughout the iterations, if there is any improvement, the best local position of the particle is updated. The improvement assessing process 305 is based on the concepts of dominance. If the position of particle i in iteration t ($x_{i,j}^t$) is dominant over $p_{i,j}$, the latter is updated. However, if there is no dominant position, a random procedure will determine if $p_{i,j}$ will be updated.

Selecting the best global position is one of the most important steps in a PSO algorithm. In the proposed method, at each iteration, H is used to determine the best global position, 310 which is obtained by randomly selecting a position among the top 15% ones after performing the ranking procedure.

5.4. Mutation

The mutation phase is responsible for the swarm diversification between iterations. For each particle, there is a small probability (in our case 5%) of occurring mutation, meaning 315 that its position might change in any dimension of the space.

Mutations can only be performed when the current iteration t satisfies the relationship: $t < \text{max_iter} \times p\text{Mut}$ and $p\text{Mut} = 0.5$. The main objective of parameter $p\text{Mut}$ is to ensure the method has more potential to explore the search space in the beginning. The convergence is only prioritized in the second half of iterations.

320 5.5. Feasibility mechanisms

Several feasibility mechanisms, based on Xu et al. (2010); Lwin (2015), were implemented to repair the infeasible solutions that may arise while executing the method. Generally, they modify the values regarding particle positions in every dimension, so that all problem constraints are met.

325 For example, let us consider a problem with 10 available assets, a solution is then found infeasible for not respecting constraint (7). The following feasibility procedure normalizes the values of w_j using Equation (20) and it is represented in Table 2.

$$w_j = \frac{w_j}{\sum_{j \in A} w_j}, \forall j \in A \quad (20)$$

Table 2: An example of solution feasibility via normalization

	z_j	0	1	1	0	1	1	0	1	\sum
Before	w_j	0.00	0.15	0.23	0.00	0.11	0.46	0.00	0.25	1.20
After	z_j	0	1	1	0	1	1	0	1	\sum
	w_j	0.00	0.13	0.19	0.00	0.09	0.38	0.00	0.21	1.00

After the normalization adjustments, violation of constraints regarding lower (ϵ_j) and 330 upper (δ_j) investment bounds per asset might occur. If that is the case, the following modifications on investment proportions should be carried out:

$$w_j = \begin{cases} w_j + \psi(\frac{\theta_j}{\delta^*}) & \text{if } \delta_j > w_j \\ \delta_j & \text{if } \delta_j < w_j \\ w_j - \phi(\frac{\varphi_j}{\epsilon^*}) & \text{if } w_j > \epsilon_j \\ \epsilon_j & \text{if } w_j < \epsilon_j, \end{cases}$$

where:

$$\theta_j = \delta_j - w_j, \quad \varphi_j = w_j - \epsilon_j, \quad \delta^* = \sum_{j=1}^A \theta_j, \theta_j > 0, \\ \psi = \sum_{j=1}^A |\theta_j|, \theta_j < 0, \quad \epsilon^* = \sum_{j=1}^A \varphi_j, \varphi_j > 0, \quad \phi = \sum_{j=1}^A |\varphi_j|, \varphi_j < 0.$$

In this context, δ^* indicates the sum of all gaps between the amount which was invested 335 and δ_j . The value of ψ represents the sum of all amounts exceeding δ_j . The sum of all

amounts exceeding ϵ_j is represented by ϵ^* . Lastly, ϕ describes the divergence between ϵ_j and the invested amounts.

In cases which there are RL constraints, one needs to adapt the mechanisms presented previously. The modified procedure calculates the remainder of the division between w_j and l_j , for each asset j . If the difference between w_j and the calculated remainder does not violate the RL constraint for j , then it is executed. Otherwise, the difference between l_j and the remainder is added to w_j :

$$w_j = \begin{cases} w_j - (w_j \bmod l_j) & \text{if } w_j - (w_j \bmod l_j) > \epsilon_j \\ w_j + l_j - (w_j \bmod l_j) & \text{if } w_j - (w_j \bmod l_j) < \epsilon_j. \end{cases}$$

After applying this procedure for all assets, the sum of remaining amounts to be invested in the portfolio is randomly distributed in single l_j lots.

Concerning PA and cardinality constraints, it was not necessary to develop a feasibility mechanism because the proposed algorithm never allows them to be violated throughout its execution.

5.6. Proposed ARMOPSO algorithm

Algorithm 1 shows the pseudocode of ARMOPSO describing how the main blocks of the method are put together.

Algorithm 1 ARMOPSO

```

1: Set the problem parameters
2:  $\lambda \leftarrow 0$ 
3: while  $\lambda < 1$  do
4:    $t \leftarrow 0$ 
5:   Initialize positions  $(x_{i,j}^t)$  for all  $N$  particles respecting to the problem's constraints.
6:   Initialize velocities  $(v_{i,j}^t \leftarrow 0)$ 
7:   Initialize the best local position for each particle  $(p_{i,j})$  and determine its best global position  $(p_{g,j})$ 
8:   Evaluate the particles
9:   Check the best candidate to be added to  $V$ 
10:  Store non-dominated solutions in  $H$ 
11:  while  $t < max\_iter$  do
12:    Sort  $H$  using non-dominated sorting
13:    Calculate the value of CD and the value of CB for each particle in  $H$ 
14:    Determine rankings  $RK_1$  and  $RK_2$  and re-sort  $H$  based on them
15:    Select  $p_{g,j}$ 
16:    Update  $v_{i,j}^t$  and  $x_{i,j}^t$ 
17:    Check particle feasibility and, if necessary, make adjustments
18:    if  $t < max\_iter \times pMut$  then
19:      Perform mutation on the particles
20:    Evaluate the particles
21:    Update  $H$  and  $p_{i,j}$ 
22:    Update  $V$ 
23:    Increase  $\lambda$ 
24: return  $H$  and  $V$ 

```

6. Computational experiments

The proposed approach was coded in C++ and it was executed on an Intel i7 3.40 GHz, 16 GB of RAM running Linux Mint 13. Only a single thread was used in our testing. The algorithm was executed 20 times for each instance and we report the average results.

355 6.1. Performance metrics

Results for MO problems can be found in several ways in the literature. Therefore, to compare the results obtained with those reported by other works, some performance metrics were used. They are presented in [Table 3](#) and described in detail in [Appendix A](#).

Table 3: Performance metrics

Metric	Assessment	Best outcome
Spacing (S)	Distance variation between solutions in the obtained frontier and in the optimal one	Lowest
Generational distance (GD)	Distance to the optimal frontier	Lowest
Diversity metric (Δ)	Dispersion of the solution set	Lowest
Hypervolume (HV)	Proximity to the optimal frontier and diversity of obtained solutions	Highest
Error ratio (Er)	Percentage of solutions that are not in the optimal frontier	Lowest
Mean return error (MRE)	Proximity to the optimal frontier	Lowest
Variance of return error (VRE)	Proximity to the optimal frontier	Lowest
Mean percentage error (MPE)		
Minimum (MinPE)	Proximity to the optimal frontier	Lowest
Maximum (MaxPE)		
Median (MedPE)		

6.2. Benchmark instances and comparative works

In order to evaluate the performance of the proposed method, we used the PSP instances ([Chang et al., 2000](#)) from the OR-Library, which were generated by collecting with data from major financial markets. They have been widely used in the literature and their optimal frontiers are detailed and accessible online. The authors made use of DATASTREAM for obtaining weekly price data during the period of March 1992 to September 1997. They collected 291 values for each stock. As shown in [Table 4](#), the benchmark is composed of five instances, each with 31, 85, 89, 98 and 225 assets. For each asset, an estimated return and standard deviation are provided. There is also a covariance matrix between all assets.

In order to assess the quality of our results, we compared them with those found by the works described in [Table 5](#), which contains the PSPs considered, approaches used, parameters adopted, metrics, sets and number of executions, as well as CPU specifications.

Table 4: Benchmark instances

Index	Country	#Assets
Hang Seng	Hong Kong	31
DAX 100	Germany	85
FTSE 100	UK	89
S&P 100	USA	98
Nikkei 225	Japan	225

Table 5: Detailed information on works considered for comparison

Work	PSPs	Methods	Parameters	Metrics	Sets	#Runs	CPU
Chang et al. (2000)¹	UPSP	GA; TS; SA CCPSP	$\epsilon_j = 0.00; \delta_j = 1$	MPE; MedPE	V	avg of 20	Workstation R4000, 100 MHz, 48 MB RAM
	CCPSP		-	MPE; Time	V, H	-	
Xu et al. (2010)²	UPSP	GA; PSO; PBIL	$\epsilon_j = 0.00; \delta_j = 1$	MPE; MedPE	V	avg of 15	-
Lwin & Qu (2013)³	UPSP	PBILDE CCPSP	$\epsilon_j = 0.00; \delta_j = 1$	MPE; Time; MedPE	V, H	avg of 20	Core 2 Duo 3.16 GHz, 2 GB RAM
	CCPSP		-	MPE; Time	-	-	
	CRLPAC-PSP	NSGA-II; SPEA2; PESA-II; PAES; MODEwAwL	$K = 10; \epsilon_j = 0.01;$ $\delta_j = 1; u = \{30\};$ $l_j = 0.008$	HV; GD; Δ	H	avg of 20	Core i7 3.2 GHz, 6 GB RAM
Mishra et al. (2014b)⁴	UPSP	PAES; PESA; Micro-GA; NSGA-II; MOBFO	$\epsilon_j = 0.00; \delta_j = 1$	GD, S, Δ , Er	V	avg of 25	Core 2 Duo 3 GHz, 4 GB RAM
	CCPSP		NSGA-II; MOBFO	VRE; MRE; S; GD; Δ		-	
	ICCPSP	PESA-II; SPEA2; NSGA-II; MOPSO PAES; PESA; Micro-GA; NSGA-II; MOBFO	$K = 10 - 15;$ $\epsilon_j = 0.01; \delta_j = 0.1$	S; GD; Δ ; Er	V, H	-	
	CCPSP		-	MPE; Time	H	-	
Fernández & Gómez (2007)⁵	CCPSP	NN	-	MPE; Time	H	-	-
Cura (2009)⁶	CCPSP	PSO	-	VRE; MRE; Time	V	-	Pentium M 2.13 GHz, 1 GB RAM
Woodside-Oriakhi et al. (2011)⁷	CCPSP	GA; TS; SA	-	MPE; Time	V	-	Core 2 Duo, 2.40 GHz, 3.24 GB RAM
Sadigh et al. (2012)⁸	CCPSP	PHNN	-	VRE; MRE	V	-	-
Salahi et al. (2014)⁹	CCPSP	ICPSO; IHS	-	VRE; MRE	V	-	Core CPU 2.70 GHz, 1 GB RAM
Mishra et al. (2014a)¹⁰	CCPSP	PESA-II; SPEA2; NSGA-II; MOPSO	-	VRE; MRE; S; GD; Δ	V	-	Core 2 Duo 3 GHz, 4 GB RAM
Baykasoglu et al. (2015)¹¹	CCPSP	GRASP	-	VRE; MRE	V, H	-	-
Anagnostopoulos & Mamanis (2011)¹²	ICCPSP	NSGA-II; SPEA2; E-NGPA2; PESA; e-MOEA; SOEA	$K = 1 - 10;$ $\epsilon_j = 0.01; \delta_j = 0.1$	HV	V, H	avg of 30	Core 2 Duo 2.1 GHz, 4 GB RAM
Lwin (2015)¹³	ICPAC-PSP	PESA-II; MOSSWA	$1 \leq K \leq 10;$ $\epsilon_j = 0.01; \delta_j = 1;$ $u = \{30\};$	GD; Δ	H	avg of 20	Core 2 Duo 2.79 GHz, 2 GB RAM

6.3. MOPSO vs. ARMOPSO

In this section, we compare the performances of the MOPSO developed in Mishra et al. (2014a) with ARMOPSO so as to better assess the impact of the CB mechanism included in the latter.

Table 6 presents the results of both approaches for the variant tackled in Mishra et al. (2014a) (CCPSP) considering the metrics adopted in the referred work. Note that the solution set V for metrics S, Δ and GD was used for evaluating the results of instance Nikkei 225. The results obtained show that ARMOPSO outperformed MOPSO in most metrics.

Table 6: ARMOPSO and MOPSO¹⁰ performances for all instances

Instance	Metric	MOPSO ¹⁰	ARMOPSO
Hang Seng	VRE	1.284	1.151
	MRE	0.602	0.574
	Time (s)	671	7
DAX 100	VRE	6.754	6.293
	MRE	1.267	1.098
	Time (s)	1566	55
FTSE 100	VRE	2.812	2.184
	MRE	0.325	0.307
	Time (s)	1582	67
S&P 100	VRE	3.476	2.406
	MRE	0.702	0.771
	Time (s)	1602	73
Nikkei 225	VRE	0.987	0.901
	MRE	0.327	0.322
	Time (s)	4700	589
S	Avg.	3.48E-6	3.15E-6
	Std.	8.50E-7	8.39E-7
	GD	Avg.	1.45E-4
		Std.	1.31E-4
Δ	Avg.	3.60E-5	3.45E-5
		Std.	0.133
			0.312
			0.043
			0.097

Furthermore, we followed the same procedure adopted in Mishra et al. (2014a) and we tested cardinality values between 20 and 180 on instance Nikkei 225. We then computed the average and standard deviation values for 20 runs of each cardinality value and report them in Table 7. One can observe that ARMOPSO outperformed MOPSO, as the first was capable of clearly achieving better results for S and GD, and slightly better ones (except for two cases) for Δ , for all cardinality values tested. This strongly suggests that the CB mechanism yielded a higher proximity to the optimal frontier, as well as higher stability in distance between points from that frontier and the obtained one.

The main improvement achieved by ARMOPSO is the increase in proximity to the optimal frontier. Figure 5a–Figure 5c illustrate the performances of the methods for S, Δ

Table 7: Comparing approaches based on PSO for S, Δ and GD – V – Nikkei 225

K	S				Δ				GD			
	MOPSO		ARMOPSO		MOPSO		ARMOPSO		MOPSO		ARMOPSO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
20	5.6E-6	1.4E-6	5.5E-6	1.46E-6	0.38	0.09	0.36	0.09	2.5E-4	8.2E-5	2.3E-4	6.1E-5
40	7.8E-6	2.1E-6	6.9E-6	1.5E-6	0.39	0.10	0.40	0.10	3.4E-4	1.0E-4	3.0E-4	7.2E-5
60	4.4E-5	1.2E-5	3.9E-5	8.6E-6	0.45	0.12	0.41	0.09	4.2E-4	1.3E-4	3.7E-4	1.1E-4
80	7.7E-5	1.9E-5	6.6E-5	1.7E-5	0.58	0.14	0.57	0.14	2.2E-3	5.5E-4	1.9E-3	5.3E-4
100	8.9E-5	2.5E-5	8.0E-5	1.9E-5	0.67	0.17	0.68	0.16	3.9E-3	9.7E-4	3.3E-3	9.8E-4
120	5.2E-4	1.3E-4	4.2E-4	9.3E-5	0.79	0.19	0.70	0.16	4.5E-3	1.4E-3	3.4E-3	1.2E-3
140	8.5E-4	2.1E-4	6.9E-4	1.6E-4	0.86	0.21	0.84	0.18	1.4E-2	3.3E-3	1.1E-2	3.2E-3
160	2.7E-3	8.9E-4	2.5E-3	7.6E-4	0.95	0.29	0.88	0.28	2.0E-2	5.2E-3	1.4E-2	3.7E-3
180	3.2E-3	9.1E-4	3.1E-3	6.8E-4	0.99	0.23	0.91	0.22	3.1E-2	1.0E-2	2.4E-2	9.3E-3

and GD. One can see that a larger cardinality produces a larger gap between the methods. For Δ , on the other hand, this tendency was not observed.

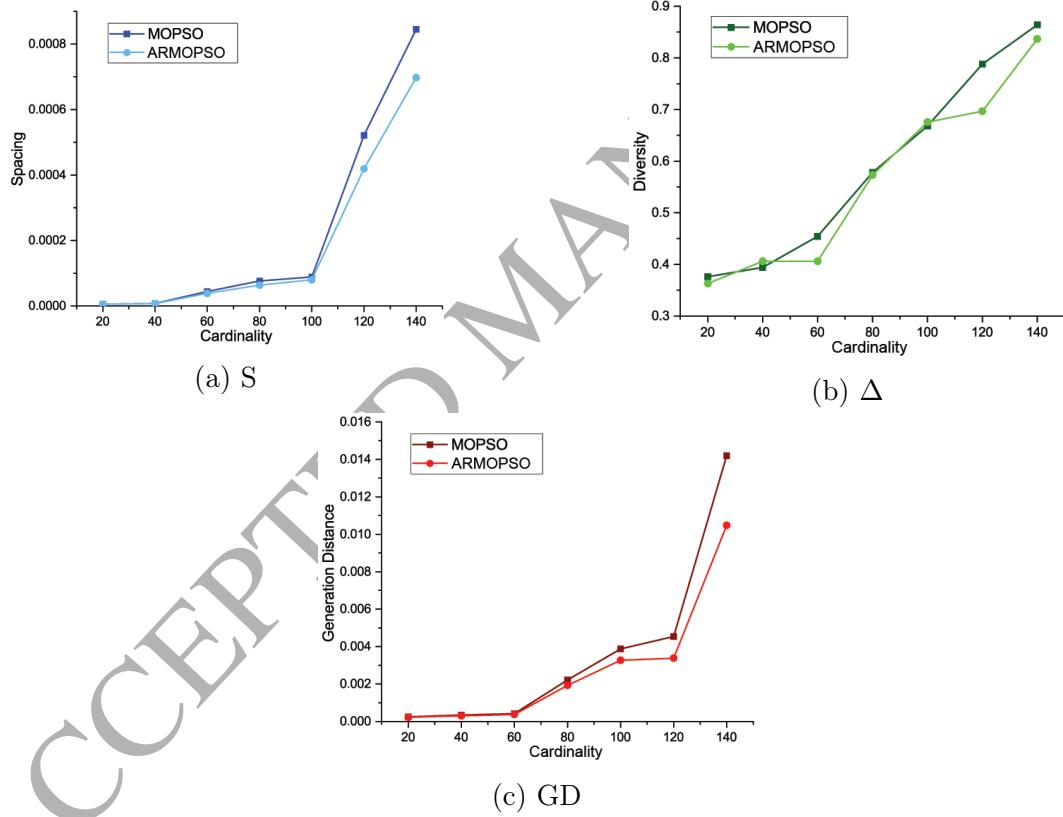


Figure 5: MOPSO and ARMOPSO performances – Nikkei 225

6.4. Comparison with the literature

This section presents a comparison between the results bound by ARMOPSO and those reported in the literature considering different performance metrics.

395 6.4.1. UPSP

In Table 8 we present comparison between the results found by the proposed algorithm and those achieved by methods proposed in Chang et al. (2000), Xu et al. (2010) and Lwin & Qu (2013). The performance metrics considered were MPE, MedPE and CPU time, in seconds. In general, GA (Chang et al., 2000), PBILDE (Lwin & Qu, 2013) and ARMOPSO 400 performed similarly in all instances, except for Nikkei 225, where GA (Chang et al., 2000) obtained a superior result. ARMOPSO outperformed the other methods in all instances and, with respect to CPU time, it appears to be faster than PBILDE (Lwin & Qu, 2013), despite the performance advantage of our CPU over theirs. Note that the runtime difference increases with the size of the instance.

Table 8: UPSP results for MPE and MedPE – V

Instance	Metric	GA ¹	TS ¹	SA ¹	GA ²	PSO ²	PBIL ²	PBILDE ³	ARMOPSO
Hang Seng	MPE	0.0202	0.8973	0.1129	0.0191	0.1422	0.0003	0.0002	0.0002
	MedPE	0.0165	1.0718	0.0161	0.0166	<0.0001	<0.0001	<0.0001	<0.0001
	Time (s)	-	-	-	-	-	-	109	15
DAX 100	MPE	0.0136	3.5645	0.0394	0.0350	1.1044	0.0023	0.0052	0.0050
	MedPE	0.0123	2.7816	0.0033	0.0124	<0.0001	<0.0001	<0.0001	<0.0001
	Time (s)	-	-	-	-	-	-	1445	126
FTSE 100	MPE	0.0063	3.2731	0.2012	0.0109	1.143	0.0186	0.0059	0.0064
	MedPE	0.0029	3.0238	0.0426	0.0020	0.0084	<0.0001	<0.0001	<0.0001
	Time (s)	-	-	-	-	-	-	1643	141
S&P 100	MPE	0.0084	4.4281	0.2158	0.0430	2.0249	0.0137	0.0078	0.0073
	MedPE	0.0085	4.2781	0.0142	0.0085	0.5133	<0.0001	<0.0001	0.0038
	Time (s)	-	-	-	-	-	-	2094	160
Nikkei 225	MPE	0.0085	15.9163	1.7681	0.3715	8.1781	0.0606	0.2733	0.0721
	MedPE	0.0084	14.2668	0.8107	0.0068	4.7023	<0.0001	<0.0001	0.0244
	Time (s)	-	-	-	-	-	-	24823	855
Avg.	MPE	0.0114	5.6158	0.4675	0.0959	2.5185	0.0191	0.0584	0.0182

405 The performance of ARMOPSO was also assessed by means of metrics S, GD, Δ and Er. The results were compared with the MO approaches by Mishra et al. (2014a). Table 9 shows these results for instance Nikkei 225, where it can be verified that our algorithm achieved the best average values for GD, Δ and Er. A better performance on GD and Er indicates more proximity between the frontier obtained by ARMOPSO and the optimal one. It also 410 means a greater capacity in finding points from the optimal frontier. On the other hand, for metric S, ARMOPSO was outperformed by MOBFO. This suggests a higher oscillation in terms of distance between points from the ARMOPSO frontier.

Table 9: UPSP results for S, GD, Δ and Er – V – Nikkei 225

Metric		PESA ⁴	Micro-GA ⁴	APAES ⁴	NSGA-II ⁴	MOBFO ⁴	ARMOPSO
S	Avg.	4.23E-5	2.67E-5	9.73E-6	5.74E-6	4.21E-6	4.35E-6
	Std.	1.03E-5	6.72E-6	1.44E-6	1.43E-6	1.05E-6	1.22E-6
GD	Avg.	1.17E-2	8.32E-3	1.28E-3	8.72E-4	7.15E-4	1.14E-4
	Std.	3.42E-3	2.08E-3	3.20E-4	2.18E-4	1.81E-4	6.45E-5
Δ	Avg.	0.812	0.803	0.786	0.596	0.553	0.308
	Std.	0.203	0.208	0.193	0.148	0.147	0.116
Er	Avg.	0.45	0.41	0.37	0.26	0.19	0.15
	Std.	0.19	0.16	0.12	0.08	0.06	0.06

6.4.2. CCPSP

The results reported in this section considered the parameters $K = 10$, $\epsilon = 0.01$ and $\delta = 1.00$ in their methods. The performances of the works for this variant, as listed in Table 5, considered various metrics. Therefore, many comparisons were necessary, both to correlate with the main results in the literature as well as to better asses the performance of ARMOPSO.

Table 10 reports the results obtained by ARMOPSO and by the methods from Table 5.

Metrics VRE and MRE, as well as CPU time, in seconds, were considered.

For the MRE metric, similar results were achieved by PHNN (Sadigh et al., 2012), IHS (Salahi et al., 2014), GRASP (Baykasoglu et al., 2015), NSGA-II, MOPSO (Mishra et al., 2014a) and ARMOPSO, but the latter was still capable of obtaining the best results for three instances. On the other hand, PHNN and IHS performed better for a single instance each. For VRE, ARMOPSO obtained the best results for all instances except Nikkei 225. The method performed with great superiority for smaller sized instances. Moreover, despite the CPU differences, ARMOPSO seems to run faster, on average, when compared to the MO approaches.

Table 11 compares the results found by the MO approaches for metrics S, GD and Δ .

In this case, set V and instance Nikkei 225 were considered.

For metrics S and GD, ARMOPSO outperformed the other methods. However, it was not able to obtain the best results for Δ , meaning that there seems to be a larger oscillation of points across the frontier. Nonetheless, such frontier was closest to the optimal one and it appeared to have more stability regarding the distances between the points of the fronts.

Table 12 shows the results for metric MPE, considering sets V and H . There was a better performance by ARMOPSO in most instances and sets, except for one combination of them (set V , instance Nikkei 225). The improvement in performance for set H is significantly more prominent than for set V .

Table 10: CCPSP results for VRE, MRE and Time – V

Instance	Metric	GA ¹	TS ¹	SA ¹	PSO ⁶	PHNN ⁸	ICPSO ⁹	IHS ⁹	PESAII ⁵	SPEA2 ¹⁰	NSGAII ¹⁰	MOPSO ¹⁰	GRASP ¹¹	ARMOPS/O
Hang Seng	VRE	1.644	1.657	1.662	2.242	2.590	1.900	1.804	1.523	1.487	1.326	1.284	1.640	1.151
	MRE	0.607	0.611	0.624	0.742	0.733	0.641	0.648	0.762	0.689	0.647	0.602	0.606	0.574
DAX 100	Time (s)	18	9	10	34	-	57	55	685	708	675	671	27	7
	VRE	7.218	9.031	8.548	6.856	5.758	7.206	7.380	9.282	8.243	7.121	6.754	6.759	6.293
FTSE 100	MRE	1.279	1.908	1.282	1.588	1.147	1.188	1.043	2.221	1.592	1.263	1.267	1.277	1.098
	Time (s)	99	42	52	179	-	254	159	1606	1653	1586	1566	86	55
S&P 100	VRE	2.866	4.012	3.820	3.059	5.414	3.381	3.248	5.238	3.765	2.987	2.812	2.430	2.184
	MRE	0.328	0.329	0.330	0.364	0.309	0.324	0.320	0.402	0.365	0.333	0.325	0.324	0.307
Nikkei 225	Time (s)	106	42	55	190	-	269	168	1621	1669	1601	1582	92	67
	VRE	3.480	5.714	5.425	3.914	5.146	4.589	3.902	7.012	5.432	3.763	3.476	2.521	2.406
	MRE	1.226	0.712	0.842	1.404	0.292	0.896	0.948	2.428	1.211	0.732	0.702	0.906	0.771
	Time (s)	126	51	66	214	-	323	186	1641	1680	1617	1602	96	73
	VRE	1.206	1.243	1.201	2.427	4.778	1.841	1.602	3.098	2.042	1.123	0.987	0.836	0.901
	MRE	5.327	0.421	0.413	0.799	0.704	0.433	0.403	1.231	0.865	0.432	0.327	0.418	0.322
	Time (s)	742	234	286	919	-	2676	659	4820	4960	4760	4700	409	589

Table 11: CCPSP results for S, GD, Δ – V – Nikkei 225

Metric		PESA-II ¹⁰	SPEA2 ¹⁰	NSGA-II ¹⁰	MOPSO ¹⁰	ARMOPSO
S	Avg.	2.33E-5	6.40E-6	4.70E-6	3.48E-6	3.15E-6
	Std.	5.80E-6	1.60E-6	1.50E-6	8.50E-7	8.39E-7
GD	Avg.	1.76E-2	1.02E-3	6.72E-3	1.45E-4	1.31E-4
	Std.	4.20E-3	2.80E-4	1.48E-3	3.60E-5	3.45E-5
Δ	Avg.	0.593	0.386	0.296	0.133	0.312
	Std.	0.148	0.093	0.078	0.043	0.097

Table 12: CCPSP results for MPE metrics – V and H

Instance	Metric	GA ¹		PBIL ²		PBILDE ³		ARMOPSO	
		V	H	V	H	V	H	V	H
Hang Seng	MPE	1.0974	0.9457	1.1026	0.8472	1.1431	0.6196	1.0520	0.5205
	MedPE	1.2181	1.1819	1.2190	1.1013	1.2390	0.4712	0.7917	0.4182
DAX 100	MPE	2.5424	1.9515	2.5163	2.0781	2.4251	1.5433	2.1570	1.1182
	MedPE	2.5466	2.1262	2.5739	2.2783	2.5866	1.0986	2.0184	1.0157
FTSE 100	MPE	1.1076	0.8784	0.996	0.7658	0.9706	0.8234	0.9128	0.5138
	MedPE	1.0841	0.5938	1.0841	0.4132	1.084	0.5134	0.6642	0.2170
S&P 100	MPE	1.9328	1.7157	2.232	1.634	1.6386	1.3902	1.6176	1.0192
	MedPE	1.2244	1.1447	1.1536	0.8453	1.1692	0.7303	1.2170	0.7712
Nikkei 225	MPE	0.7961	0.6431	1.0017	0.6451	0.5972	0.3996	0.6178	0.3209
	MedPE	0.6133	0.6062	0.5854	0.5596	0.5896	0.4619	0.2273	0.1730
Avg.	MPE	1.4953	1.2269	1.5697	1.1940	1.3549	0.9552	1.2914	0.6985
	MedPE	1.3373	1.1306	1.3232	1.0395	1.3337	0.6551	1.0437	0.5190

Some authors reported MPE for set H , in addition to MinPE, MaxPE and MedPE, as illustrated in Table 13. It is possible to observe an equivalent performance between the proposed method and those by Woodside-Oriakhi et al. (2011), i.e., GA, TS and SA. The best performing approaches, however, were GA, TS and ARMOPSO. Concerning the CPU time, ARMOPSO seems to run faster for the first three instances, whereas GA appears to be the fastest for Nikkei 225.

We were able to obtain the H frontiers of GA and NN generated in Fernández & Gómez (2007) for CCPSP. We compared them to the one achieved by ARMOPSO, as illustrated in Figure 6. We chose to depict the comparison for instance Nikkei 225, since it is the largest one. We can observe that the frontier generate by ARMOPSO has higher stability and proximity to the optimal one. The graph was created by depicting one in every ten consecutive points, so as to improve its visualization. Fernández & Gómez (2007) were the only authors to made their frontiers available for comparison.

Table 13: CCPSP results for MPE metrics – H

Instances	Metric	NN ⁵	GA ⁷	TS ⁷	SA ⁷	PBILDE ³	ARMOPSO
Hang Seng	MPE	1.2310	0.8501	0.8234	1.0589	0.6196	0.5205
	MedPE	-	0.5873	0.3949	0.5355	0.4712	0.4182
	MinPE	-	0.0036	0.0068	0.0349	0.2816	0.0307
	MaxPE	-	2.9034	4.6096	4.6397	0.6768	1.9327
DAX 100	Time (s)	47	76	85	99	-	7
	MPE	1.5770	0.7740	0.7190	1.0267	1.5433	1.1182
	MedPE	-	0.2400	0.4298	0.8682	1.0986	1.0157
	MinPE	-	0.0000	0.0149	0.0278	0.7537	0.0472
FTSE 100	MaxPE	-	4.6811	2.7770	4.4123	1.6804	2.4820
	Time (s)	162	74	113	293	-	55
	MPE	1.2510	0.1620	0.3930	0.8952	0.8234	0.5138
	MedPE	-	0.0820	0.2061	0.3944	0.5134	0.2170
S&P 100	MinPE	-	0.0000	0.0019	0.0230	0.4359	0.0000
	MaxPE	-	0.7210	3.4570	10.2029	0.8695	1.5912
	Time (s)	160	95	232	286	-	67
	MPE	1.7920	0.2922	1.0358	3.0952	1.3902	1.0192
Nikkei 225	MedPE	-	0.1809	1.0248	2.1064	0.7303	0.7712
	MinPE	-	0.0007	0.0407	0.8658	0.4816	0.0395
	MaxPE	-	1.6295	3.0061	8.6652	1.5726	2.1361
	Time (s)	178	100	222	371	-	73
Avg.	MPE	1.4730	0.3353	0.7838	1.1193	0.3996	0.3209
	MedPE	-	0.3040	0.6526	0.6877	0.4619	0.1730
	MinPE	-	0.0180	0.0085	0.0113	0.3739	0.0106
	MaxPE	-	1.0557	2.6082	3.9678	0.4965	1.1293
	Time (s)	570	104	414	604	-	589
	MPE	1.4640	0.4827	0.7510	1.4391	0.9552	0.6985
	MedPE	-	0.2788	0.5416	0.9184	0.6550	0.5190
	MinPE	-	0.0045	0.0146	0.1926	0.4653	0.0081
	MaxPE	-	2.1981	3.2916	6.3776	1.0591	1.3311
	Time (s)	223	90	213	331	-	158

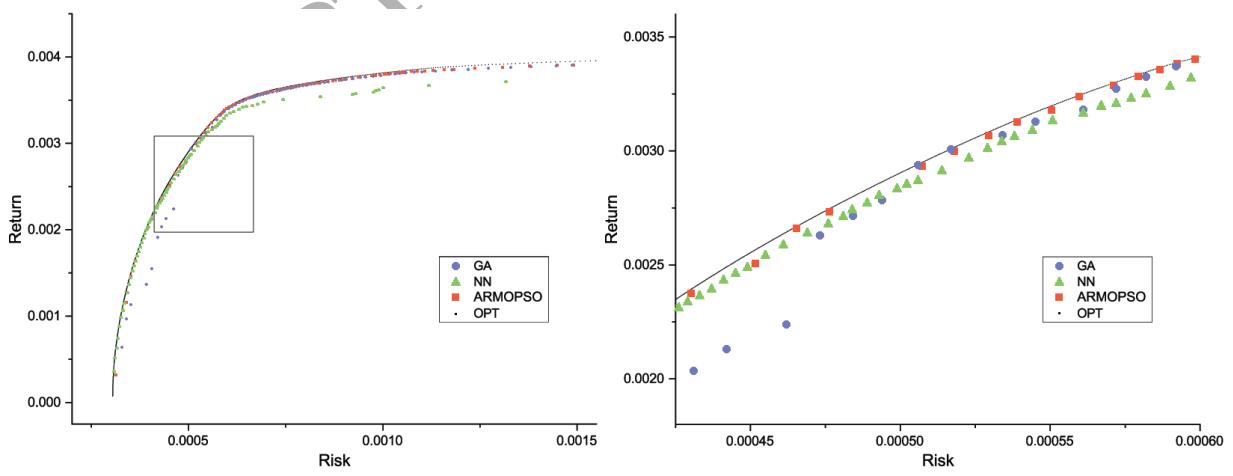


Figure 6: Comparing GA, NN and ARMOPSO frontiers – CCPSP – Nikkei 225

6.4.3. ICCPSP

According to [Anagnostopoulos & Mamanis \(2011\)](#), this variant has been more frequently studied due to the applicability of its constraints in the financial market. However, there
455 are not many results reported for it yet.

[Anagnostopoulos & Mamanis \(2011\)](#) reported performances of several MO approaches for ICCPSP using HV. These approaches include the well-known NSGA-II and SPEA2, as well as less explored strategies, such as E-NGPA2 and e-MOEA. The authors considered $1 < K < 10$ in their tests. From [Table 14](#), one can infer that ARMOPSO performed better
460 in four instances, but it was marginally outperformed by SPEA2 in instance DAX 100.

Table 14: ICCPSP results for HV

Instance	NSGA-II ^{T2}	SPEA2 ^{T2}	E-NGPA2 ^{T2}	PESA ^{T2}	e-MOEA ^{T2}	SOEA ^{T2}	ARMOPSO
Hang Seng	0.705	0.705	0.704	0.704	0.705	0.704	0.712
DAX 100	0.809	0.810	0.809	0.808	0.809	0.808	0.806
FTSE 100	0.718	0.718	0.717	0.717	0.718	0.717	0.723
S&P 100	0.791	0.791	0.790	0.780	0.791	0.790	0.798
Nikkei 225	0.806	0.806	0.805	0.805	0.806	0.804	0.812

Regarding S, GD, Δ and Er metrics, we compared ARMOPSO with traditional MO approaches, as well as with APAES, Micro-GA and MOBFO. The latter was proposed by [Mishra et al. \(2014a\)](#). [Table 15](#) shows results for instance Nikkei 225, where we can verify the superior performance by ARMOPSO considering S, GD and Er. However, for Δ and the
465 standard deviation of Er, the proposed method was outperformed by MOBFO and NSGA-II, respectively.

Table 15: ICCPSP results for S, GD, Δ and Er – Nikkei 225

Metric	PESA ⁴	Micro-GA ⁴	APAES ⁴	NSGA-II ⁴	MOBFO ⁴	ARMOPSO
S	Avg.	4.34E-4	4.07E-4	7.54E-5	6.54E-5	5.65E-5
	Std.	1.32E-4	1.21E-4	2.33E-5	2.33E-5	1.98E-5
GD	Avg.	6.17E-2	4.04E-2	7.32E-3	6.34E-3	8.57E-3
	Std.	1.87E-2	1.88E-2	2.01E-3	1.89E-3	2.52E-3
Δ	Avg.	0.932	0.888	0.839	0.866	0.755
	Std.	0.254	0.232	0.222	0.243	0.201
Er	Avg.	0.910	0.888	0.824	0.801	0.788
	Std.	0.299	0.245	0.288	0.234	0.235

6.4.4. ICPAC-PSP

For this PSP variant, we compared the results obtained with those reported in [Lwin \(2015\)](#). Detailed information of this work is provided in [Table 5](#), which also shows that asset
470 30 was pre-assigned to the portfolio.

Regarding GD, ARMOPSO performed better in all instances, specially those with fewer assets, as displayed in [Table 16](#). For Δ the performances were similar. The proposed

method managed to produce the best results for smaller instances. However, for the larger ones, MOSSwA was superior.

Table 16: ICPAC-PSP results for GD and Δ

Instance	Metric	PESA-II ¹³	MOSSwA ¹³	ARMOPSO
Hang Seng	GD	6.90E-3	5.50E-4	2.67E-4
	Δ	0.93	0.22	0.18
DAX 100	GD	6.70E-3	5.80E-4	3.09E-4
	Δ	0.91	0.21	0.19
FTSE 100	GD	8.10E-3	1.10E-3	5.78E-4
	Δ	1.19	0.26	0.34
S&P 100	GD	5.20E-3	1.20E-3	6.10E-4
	Δ	1.14	0.28	0.36
Nikkei 225	GD	2.70E-2	2.30E-3	9.73E-4
	Δ	1.28	0.31	0.43

475 6.4.5. CRLPAC-PSP

The results on CRLPAC-PSP were compared with several approaches proposed by Lwin & Qu (2013) of which the settings are provided in Table 5. The performances of the methods were assessed via HV, GD, Δ , as shown in Table 17.

Table 17: CRLPAC-PSP Avg. Results for HV, GD and Δ

Instance	Metric	NSGA-II ³	SPEA2 ³	PESA-II ³	PAES ³	MODEwAwL ³	ARMOPSO
Hang Seng	HV	0.65	0.62	0.61	0.47	0.67	0.66
	GD	2.66E-3	2.61E-3	2.93E-3	2.61E-3	2.85E-3	2.58E-3
	Δ	0.58	0.51	0.77	0.76	0.67	0.57
DAX 100	HV	0.69	0.68	0.66	0.19	0.67	0.66
	GD	4.90E-3	4.45E-3	5.60E-3	2.33E-3	2.70E-3	2.30E-3
	Δ	0.80	0.89	1.16	0.98	0.87	0.85
FTSE 100	HV	0.56	0.55	0.55	0.33	0.70	0.70
	GD	4.32E-3	4.55E-3	4.68E-3	3.75E-3	2.40E-3	2.72E-3
	Δ	0.68	0.63	1.03	0.89	0.60	0.60
S&P 100	HV	0.60	0.60	0.59	0.27	0.74	0.73
	GD	4.40E-3	4.50E-3	4.68E-3	3.72E-3	2.85E-3	2.71E-3
	Δ	0.71	0.71	1.08	0.97	0.60	0.63
Nikkei 225	HV	0.17	0.13	0.07	0.11	0.51	0.54
	GD	8.50E-3	1.32E-2	1.85E-2	6.35E-3	2.70E-3	2.82E-3
	Δ	0.98	1.20	1.27	1.22	0.60	0.64

The results demonstrate that the methods seem to have more difficulties in solving 480 CRLPAC-PSP, when compared to other PSP variants considered in this work, at least for the referred metrics. For GD, ARMOPSO performed slightly better than MODEwAwL for three instances, and better than the other procedures for all instances. Moreover, SPEA2, NSGA-II, MODEwAwL and ARMOPSO achieved competitive results regarding Δ , for all instances. For smaller instances, SPEA2 and NSGA-II were the best performing methods. On the other hand, for the larger ones, MODEwAwL and ARMOPSO were the most successful.

7. Concluding remarks

This work addressed several mean-variance portfolio selection problem (PSP) variants by means of a unified multi-objective (MO) particle swarm optimization (MOPSO) approach. The proposed method, called adaptive ranking MOPSO (ARMOPSO), introduced a ranking procedure based on non-dominated sorting, crowding distance and a new mechanism called cost-benefit. According to our extensive literature review on the PSP works published in the past two decades, this is one of the first generic procedures to tackle various PSPs using a same framework.

ARMOPSO was tested on five mean-variance PSP variants for which there are results reported in the literature. These problems include constraints that arise in financial markets such as floor and ceiling, cardinality, round-lot and pre-assignment. The results were compared using the specific metrics adopted by each work, which may include one or more among the following: spacing (S), generational distance (GD), diversity metric (Δ), hyper-volume (HV), error ratio (Er), mean return error (MRE), variance of returned error (VRE), as well as mean percentage error (MPE), minimum (MinPE), maximum (MaxPE) and median (MedPE). We compiled their description in [Appendix A](#) to serve as a self-contained reference for those interested in implementing such metrics when evaluating the performance of MO algorithms for PSPs.

The results obtained from the extensive computational experiments show that ARMOPSO achieved a highly competitive performance in all variants and in most of the assessed metrics, when compared to the problem-specific methods proposed in the literature. This highlights not only the efficiency of the unified method but also its remarkable robustness.

Future work may include the development of more effective ways to manage infeasibilities, thus helping the algorithm to find improved non-dominated solutions, and consequently a better quality frontier. Advances regarding this issue might also reduce the CPU time spent by the procedure, especially for large size instances. Furthermore, we believe that ARMOPSO can also be applied to solve other MO problems.

Acknowledgments

We would like to thank the anonymous reviewers for the insightful comments and Luciano Costa for the valuable discussions that helped improving the quality of the paper. This research was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), grants 305223/2015-1 and 428549/2016-0 and by Comissão de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Finance Code 001.

CRediT Author Statement

520 Yuri Laio T. V. Silva: Conceptualization, Methodology, Investigation, Data curation,
 Software, Validation. Ana Beatriz Herthel: Writing-Original draft preparation, Visualization.
 Anand Subramanian: Supervision, Resources, Writing-Reviewing and Editing.

DECLARATION OF INTEREST STATEMENT

525 We wish to confirm that there are no known conflicts of interest associated with this
 publication and there has been no significant financial support for this work that could have
 influenced its outcome.

530 We confirm that the manuscript has been read and approved by all named authors and
 that there are no other persons who satisfied the criteria for authorship but are not listed.
 We further confirm that the order of authors listed in the manuscript has been approved by
 all of us.

We confirm that we have given due consideration to the protection of intellectual property
 associated with this work and that there are no impediments to publication, including the
 timing of publication, with respect to intellectual property. In so doing we confirm that we
 have followed the regulations of our institutions concerning intellectual property.

535 We understand that the Corresponding Author is the sole contact for the Editorial process
 (including Editorial Manager and direct communications with the office). He/she is
 responsible for communicating with the other authors about progress, submissions of revisions
 and final approval of proofs. We confirm that we have provided a current, correct email
 address which is accessible by the Corresponding Author and which has been configured to
 540 accept email from anand@ci.ufpb.br.

Signed by all authors as follows:

Yuri Laio T. V. Silva – April, 21, 2019

Ana Beatriz F Herthel – April, 21, 2019

References

- 545 Anagnostopoulos, K., & Mamanis, G. (2009). Finding the efficient frontier for a mixed
 integer portfolio choice problem using a multiobjective algorithm. *iBusiness*, 1, 99–105.
 Anagnostopoulos, K., & Mamanis, G. (2010). A portfolio optimization model with three
 objectives and discrete variables. *Computers & Operations Research*, 37, 1285–1297.

- Anagnostopoulos, K., & Mamanis, G. (2011). The mean variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, 38, 14208–14217. doi:<http://dx.doi.org/10.1016/j.eswa.2011.04.233>.
- Aouni, B., Colapinto, C., & Torre, D. L. (2014). Financial portfolio management through the goal programming model: Current state-of-the-art. *European Journal of Operational Research*, 234, 536–545.
- Armananzas, R., & Lozano, J. A. (2005). A multiobjective approach to the portfolio optimization problem. In *2005 IEEE Congress on Evolutionary Computation* (pp. 1388–1395). IEEE volume 2.
- Azmi, R., & Tamiz, M. (2010). A review of goal programming for portfolio selection. In D. Jones, M. Tamiz, & J. Ries (Eds.), *New Developments in Multiple Objective and Goal Programming* (pp. 15–33). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Baykasoglu, A., Yunusoglu, M. G., & Ozsoydan, F. B. (2015). A grasp based solution approach to solve cardinality constrained portfolio optimization problems. *Computers & Industrial Engineering*, 90, 339–351.
- Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of the operational research society*, (pp. 1069–1072).
- Bertsimas, D., & Pachamanova, D. (2008). Robust multiperiod portfolio management in the presence of transaction costs. *Computers & Operations Research*, 35, 3–17.
- Branke, J., Scheckenbach, B., Stein, M., Deb, K., & Schmeck, H. (2009). Portfolio optimization with an envelope-based multi-objective evolutionary algorithm. *European Journal of Operational Research*, 199, 684–693. doi:<http://dx.doi.org/10.1016/j.ejor.2008.01.054>.
- Cesarone, F., Scozzari, A., & Tardella, F. (2013). A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, 205, 213–234. doi:[10.1007/s10479-012-1165-7](https://doi.org/10.1007/s10479-012-1165-7).
- Chang, T.-J., Meade, N., Beasley, J., & Sharaiha, Y. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27, 1271–1302. doi:[http://dx.doi.org/10.1016/S0305-0548\(99\)00074-X](http://dx.doi.org/10.1016/S0305-0548(99)00074-X).
- Chen, W., & Zhang, W.-G. (2010). The admissible portfolio selection problem with transaction costs and an improved pso algorithm. *Physica A: Statistical Mechanics and its Applications*, 389, 2070–2076.
- Chen, Y., Mabu, S., & Hirasawa, K. (2010). A model of portfolio optimization using time

- adapting genetic network programming. *Computers & Operations Research*, 37, 1697–1707. doi:<http://dx.doi.org/10.1016/j.cor.2009.12.003>.
- 585 Chiam, S., Tan, K., & Al Mamum, A. (2008). Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing*, 5, 67–80.
- Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1, 269–308.
- 590 Corazza, M., Fasano, G., & Gusso, R. (2013). Particle swarm optimization with non-smooth penalty reformulation, for a complex portfolio selection problem. *Applied Mathematics and Computation*, 224, 611–624.
- Crama, Y., & Schyns, M. (2003). Simulated annealing for complex portfolio selection problems. *European Journal of Operational Research*, 150, 546–571. doi:[http://dx.doi.org/10.1016/S0377-2217\(02\)00784-1](http://dx.doi.org/10.1016/S0377-2217(02)00784-1). Financial Modelling.
- 595 Cura, T. (2009). Particle swarm optimization approach to portfolio optimization. *Nonlinear Analysis: Real World Applications*, 10, 2396–2406. doi:<http://dx.doi.org/10.1016/j.nonrwa.2008.04.023>.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective 600 genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6, 182–197.
- Deng, G.-F., & Lin, W.-T. (2010a). Ant colony optimization for markowitz mean-variance portfolio model. In *Swarm, Evolutionary, and Memetic Computing* (pp. 238–245). Springer.
- Deng, G.-F., & Lin, W.-T. (2010b). Swarm intelligence for cardinality-constrained portfolio 605 problems. In *Computational Collective Intelligence. Technologies and Applications* (pp. 406–415). Springer.
- Deng, G.-F., Lin, W.-T., & Lo, C.-C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, 39, 4558–4566.
- 610 Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)* (pp. 84–88). IEEE volume 1.
- Ehrgott, M. (2005). *Multicriteria Optimization*. (2nd ed.). Berlin/Heidelberg: Springer-Verlag. URL: <http://link.springer.com/10.1007/3-540-27659-9>.
- 615 Fernández, A., & Gómez, S. (2007). Portfolio selection using neural networks. *Computers & Operations Research*, 34, 1177–1191.

- Golmakani, H. R., & Fazel, M. (2011). Constrained portfolio selection using particle swarm optimization. *Expert Systems with Applications*, 38, 8327–8335. doi:<http://dx.doi.org/10.1016/j.eswa.2011.01.020>.
- 620 Guo, S., Yu, L., Li, X., & Kar, S. (2016). Fuzzy multi-period portfolio selection with different investment horizons. *European Journal of Operational Research*, 254, 1–10. doi:[10.1016/j.ejor.2016.04.055](https://doi.org/10.1016/j.ejor.2016.04.055).
- Handoyo, S., Efendi, A., Jie, F., & Widodo, A. (2017). Implementation of particle swarm optimization (PSO) algorithm for estimating parameter of arma model via maximum likelihood method. *Far East Journal of Mathematical Sciences*, 102, 1337–1363.
- 625 Jagannathan, R., & Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *The Journal of Finance*, 58, 1651–1683.
- Jarraya, B. (2013). Asset allocation and portfolio optimization problems with metaheuristics: a literature survey. *Business Excellence and Management*, 3, 38–56.
- 630 Kalayci, C. B., Ertenlice, O., & Akbay, M. A. (2019). A comprehensive review of deterministic models and applications for mean-variance portfolio optimization. *Expert Systems with Applications*, 125, 345 – 368.
- Kan, R., & Zhou, G. (2007). Optimal portfolio choice with parameter uncertainty. *The Journal of Financial and Quantitative Analysis*, 42, 621–656.
- 635 Kennedy, J., Kennedy, J. F., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann.
- Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research*, 234, 356–371.
- 640 Liagkouras, K., & Metaxiotis, K. (2014). A new Probe Guided Mutation operator and its application for solving the cardinality constrained portfolio optimization problem. *Expert Systems with Applications*, 41, 6274–6290. doi:[10.1016/j.eswa.2014.03.051](https://doi.org/10.1016/j.eswa.2014.03.051).
- Lim, S., Oh, K. W., & Zhu, J. (2014). Use of DEA cross-efficiency evaluation in portfolio selection: An application to Korean stock market. *European Journal of Operational Research*, 236, 361–368. doi:[10.1016/j.ejor.2013.12.002](https://doi.org/10.1016/j.ejor.2013.12.002).
- 645 Lin, C.-C., & Liu, Y.-T. (2008). Genetic algorithms for portfolio selection problems with minimum transaction lots. *European Journal of Operational Research*, 185, 393–404.
- Lin, D., & Wang, S. (2002). A genetic algorithm for portfolio selection problems. *Advanced Modeling and Optimization*, 4, 13–27.
- 650 Liu, J., Jin, X., Wang, T., & Yuan, Y. (2015). Robust multi-period portfolio model based

- on prospect theory and almv-ps algorithm. *Expert Systems with Applications*, 42, 7252–7262.
- Lwin, K., & Qu, R. (2013). A hybrid algorithm for constrained portfolio selection problems. *Applied intelligence*, 39, 251–266.
- 655 Lwin, K. T. (2015). *Evolutionary approaches for portfolio optimization*. Ph.D. thesis University of Nottingham.
- Mansini, R., Ogryczak, W., & Speranza, M. G. (2014). Twenty years of linear programming based portfolio optimization. *European Journal of Operational Research*, 234, 518–535. doi:<http://dx.doi.org/10.1016/j.ejor.2013.08.035>.
- 660 Maringer, D., & Kellerer, H. (2003). Optimization of cardinality constrained portfolios with a hybrid local search algorithm. *OR Spectrum*, 25, 481–495. doi:[10.1007/s00291-003-0139-1](https://doi.org/10.1007/s00291-003-0139-1).
- Markowitz, H. (1952). Portfolio selection. *Journal of Finance*, 7, 77–91.
- Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for 665 engineering. *Structural and multidisciplinary optimization*, 26, 369–395.
- Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: a comprehensive literature review. *Expert Systems with Applications*, 39, 11685–11698.
- Mishra, S. K. (2012). *Robust and Constrained Portfolio Optimization using Multiobjective Evolutionary Algorithms*. Ph.D. thesis National Institute of Technology Rourkela.
- 670 Mishra, S. K., Panda, G., & Majhi, R. (2014a). A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. *Swarm and Evolutionary Computation*, 16, 38–51.
- Mishra, S. K., Panda, G., & Majhi, R. (2014b). Constrained portfolio asset selection using 675 multiobjective bacteria foraging optimization. *Operational Research*, 14, 113–145. doi:[10.1007/s12351-013-0138-1](https://doi.org/10.1007/s12351-013-0138-1).
- Moral-Escudero, R., Ruiz-Torrubiano, R., & Suárez, A. (2006). Selection of optimal investment portfolios with cardinality constraints. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 2382–2388). IEEE.
- 680 Mostaghim, S., & Teich, J. (2003). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE* (pp. 26–33). IEEE.
- Noroozi, A., Mokhtari, H., Bagherpour, M., & Sadjadi, S. J. (2016). A probabilistic portfolio budget allocation problem with cpi index under risk. *Journal of Industrial and Production*

- 685 *Engineering*, *33*, 236–246.
- Pai, G., & Michel, T. (2009). Evolutionary optimization of constrained-means clustered assets for diversification in small portfolios. *Evolutionary Computation, IEEE Transactions on*, *13*, 1030–1053.
- 690 Palczewski, J., Poulsen, R., Schenk-Hoppé, K. R., & Wang, H. (2015). Dynamic portfolio optimization with transaction costs and state-dependent drift. *European Journal of Operational Research*, *243*, 921–931. doi:[10.1016/j.ejor.2014.12.040](https://doi.org/10.1016/j.ejor.2014.12.040).
- Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2013). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on Evolutionary Computation*, *17*, 321–344.
- 695 Qu, B.-Y., Suganthan, P. N., & Das, S. (2013). A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, *17*, 387–402.
- Rezaee Jordehi, A., & Jasni, J. (2013). Parameter selection in particle swarm optimisation: a survey. *Journal of Experimental & Theoretical Artificial Intelligence*, *25*, 527–542.
- 700 Ruiz-Torrubiano, R., & Suárez, A. (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. *Applied Soft Computing*, *36*, 125–142.
- Sabar, N. R., & Kendall, G. (2014). Using harmony search with multiple pitch adjustment operators for the portfolio selection problem. *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, (pp. 499–503). doi:[10.1109/CEC.2014.6900384](https://doi.org/10.1109/CEC.2014.6900384).
- 705 Sadigh, A. N., Mokhtari, H., Iranpoor, M., & Ghomi, S. (2012). Cardinality constrained portfolio optimization using a hybrid approach based on particle swarm optimization and hopfield neural network. *Advanced Science Letters*, *17*, 11–20.
- Salahi, M., Daemi, M., Lotfi, S., & Jamalian, A. (2014). PSO and Harmony Search Algorithms for Cardinality Constrained Portfolio Optimization Problem. *Advanced Modeling and Optimization*, *16*, 559–573.
- 710 Schaerf, A. (2002). Local search techniques for constrained portfolio selection problems. *Computational Economics*, *20*, 177–190. doi:[10.1023/A:1020920706534](https://doi.org/10.1023/A:1020920706534).
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)* (pp. 69–73). IEEE.
- 715 Skolpadungket, P., Dahal, K., & Harnpornchai, N. (2007). Portfolio optimization using multi-objective genetic algorithms. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (pp. 516–523). IEEE.

- 720 Soleimani, H., Golmakani, H. R., & Salimi, M. H. (2009). Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications*, 36, 5058–5063. doi:<http://dx.doi.org/10.1016/j.eswa.2008.06.007>.
- Speranza, M. G. (1996). A heuristic algorithm for a portfolio optimization model applied to the Milan stock market. *Computers & Operations Research*, 23, 433–441.
- 725 Streichert, F., & Tanaka-Yamawaki, M. (2006). The effect of local search on the constrained portfolio selection problem. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on* (pp. 2368–2374). IEEE.
- Van Veldhuizen, D. A. (1999). *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Technical Report DTIC Document.
- 730 Van Veldhuizen, D. A., & Lamont, G. B. (1998). *Multiobjective evolutionary algorithm research: A history and analysis*. Technical Report Air Force Institute of Technology.
- Woodside-Oriakhi, M., Lucas, C., & Beasley, J. (2013). Portfolio rebalancing with an investment horizon and transaction costs. *Omega*, 41, 406–420.
- Woodside-Oriakhi, M., Lucas, C., & Beasley, J. E. (2011). Heuristic algorithms for the 735 cardinality constrained efficient frontier. *European Journal of Operational Research*, 213, 538–550.
- Xidonas, P., Mavrotas, G., Zopounidis, C., & Psarras, J. (2011). Ipssis: An integrated multi-criteria decision support system for equity portfolio construction and selection. *European Journal of Operational Research*, 210, 398–409.
- 740 Xu, R.-t., Zhang, J., Liu, O., & Huang, R.-z. (2010). An estimation of distribution algorithm based portfolio selection approach. *2010 International Conference on Technologies and Applications of Artificial Intelligence*, (pp. 305–313).
- Zhu, H., Wang, Y., Wang, K., & Chen, Y. (2011). Particle swarm optimization (pso) for the constrained portfolio optimization problem. *Expert Systems with Applications*, 38, 10161–10169.

Appendix A. Performance metrics

Results for multi-objective problems can be found in several ways in the literature. Therefore, in this appendix we present a brief explanation of the performance metrics used to compare the results.

750 Appendix A.1. Spacing

Spacing (S) measures the dispersion between the set of non-dominated solutions and the optimal frontier. More precisely, it assess the distance variation between each solution from the frontier obtained and the nearest solution from the optimal one, as given by Equation [Equation A.1](#):

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2}, \quad (\text{A.1})$$

where d_i is the Euclidean distance between point i of the frontier obtained and the nearest point j from the optimal one. The term \bar{d} represents the average of all d_i and n is the amount of solutions in the frontier obtained. The value of S should be as small as possible to indicate that the set of solutions are of high quality. Note that $S = 0$ means that all points are equally far from the optimal frontier.

755 Appendix A.2. Generational distance

Generational distance (GD) estimates the distance between the frontier obtained and the optimal one [Van Veldhuizen & Lamont \(1998\)](#); [Van Veldhuizen \(1999\)](#). It is determined by computing the average Euclidean distance between the elements from the set on non-dominated solutions and the corresponding nearest points from the optimal Pareto front, as shown in [Equation A.2](#):

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n}. \quad (\text{A.2})$$

A lower value of GD means a smaller distance from the obtained set to the optimal frontier. When $GD = 0$, all set solutions are present in the optimal frontier. [Figure A.7](#) illustrates the GD.

760 Appendix A.3. Diversity metric

Diversity metric (Δ) refers to the dispersion of the set, that is, how uniformly the points are distributed regarding the proximity of the set in the objective space. This metric does not

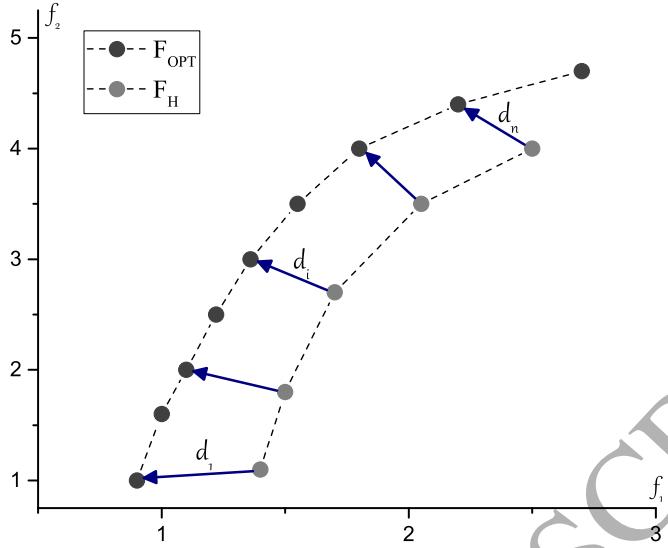


Figure A.7: Generational distance

necessarily require an optimal frontier, as it is related to the Euclidean distance between the solutions Deb et al. (2002). Equation Equation A.3 shows how the value of Δ is determined:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1)\bar{d}}, \quad (\text{A.3})$$

where d_i is the Euclidean distance between consecutive solutions in the frontier obtained and \bar{d} is the average of these distances. On the other hand, d_f and d_l are the Euclidean distances between the extreme solutions of the frontier obtained and the nearest points of the optimal frontier, respectively. A lower value of Δ means a better diversification of the non-dominated solutions set. A value of $\Delta = 0$ indicates that the set is as evenly distributed as possible.

Appendix A.4. Hypervolume

Hypervolume is obtained by calculating the area (or volume) that emerges from the relationship between the points of the obtained Pareto front and a given reference point, denoted as Nadir point (R). The metric allows for measuring the proximity between the frontier obtained and the ideal one, as well as the diversity of the solutions achieved.

The Nadir point can be determined in several ways. The most popular one uses a vector corresponding to the worst values obtained by the objective function. Hence, it is possible to ensure that all solutions of the frontier obtained will dominate the Nadir point.

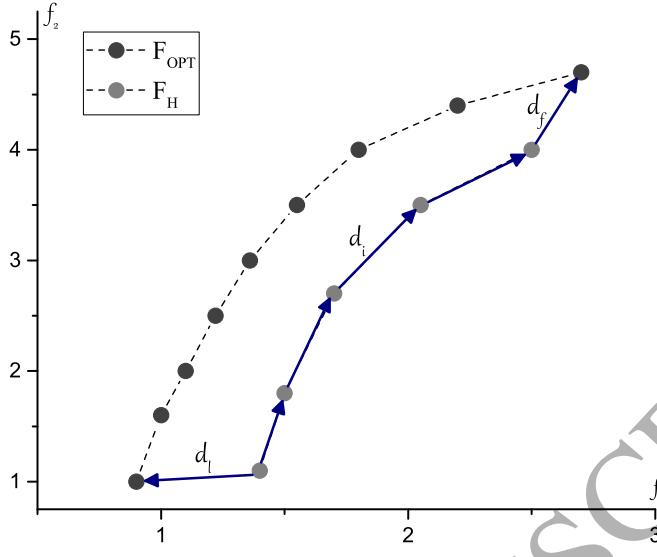


Figure A.8: Diversity metric

The HV of the set of non-dominated solutions (H) corresponds to the sum of the areas or volumes (V_x) formed between each solution $x_i \in H$ and R , as observed in Equation (A.4):

$$HV = \bigcup_{x_i \in H} V_x. \quad (\text{A.4})$$

A higher value of HV means that the frontier obtained is far from R and, consequently, closer to the optimal one. It also indicates a greater solution diversity. Since R is used as reference, the calculation of HV does not depend on the existence of an optimal frontier. This is one of the reasons for its popularity in MO problems. Figure A.9 depicts the representation of the HV for an arbitrary frontier.

780 Appendix A.5. Error ratio

Error Ratio (Er) indicates the percentage of solutions in H that are not part of the optimal frontier Van Veldhuizen & Lamont (1998). A lower value of Er indicates a better set of solutions, as it has more points as part of the optimal frontier. The value is computed as illustrated in Equation A.5:

$$Er = \frac{\sum_{i=1}^n e_i}{n}, \quad (\text{A.5})$$

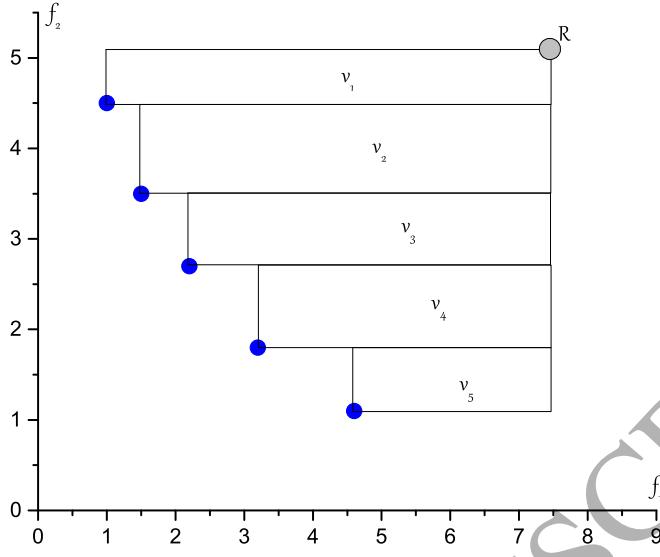


Figure A.9: Hypervolume

where e_i is a binary variable that assumes value 1 if solution i is not part of the optimal frontier, and 0 otherwise.

Appendix A.6. Mean return error, variance return error, mean percentage error

Apart from the metrics mentioned above, mean return error (MRE), variance of return error (VRE) and mean percentage error (MPE) were also considered. These metrics are mathematically detailed in [Chang et al. \(2000\)](#); [Fernández & Gómez \(2007\)](#).

Let (v_{ij}^s, r_{ij}^s) be a point in the optimal frontier closest located to point (v_j^h, r_j^h) from the obtained one, and let T be the number of points in the optimal frontier. In this context, i_j is defined as follows:

$$i_j = \arg \min_{i=1, \dots, T} \sqrt{(v_i^s - v_j^h)^2 + (r_i^s - r_j^h)^2}, \forall j = 1, \dots, n. \quad (\text{A.6})$$

VRE and MRE are defined, respectively, as follows:

$$VRE = \frac{(\sum_{j=1}^n 100|v_{ij}^s - v_j^h|/v_j^h)}{n} \quad (\text{A.7})$$

$$MRE = \frac{(\sum_{j=1}^n 100|r_{ij}^s - r_j^h|/r_j^h)}{n}. \quad (\text{A.8})$$

Some authors also used MPE to assess the quality of the frontiers obtained [Fernández &](#)

Gómez (2007); Woodside-Oriakhi et al. (2011). Consider (v_i^s, r_i^s) , $(i = 1, \dots, T)$, and (v_j^h, r_j^h) , $(j = 1, \dots, n)$. These terms represent, respectively, the variance and average return for the i th point in the optimal frontier and for the j th point in the obtained one. Therefore, in order to locate the point in the optimal frontier that is nearest to the average return (\hat{r}_{ij}^s) and return variance (\hat{v}_{ij}^s), the following calculations must be performed:

$$r_{m_j}^s = \min(r_i^s | r_i^s \geq r_j^h), j = 1, \dots, n \quad (\text{A.9})$$

$$r_{n_j}^s = \max(r_i^s | r_i^s \leq r_j^h), j = 1, \dots, n \quad (\text{A.10})$$

$$v_{m_j}^s = \min(v_i^s | v_i^s \geq v_j^h), j = 1, \dots, n \quad (\text{A.11})$$

$$v_{n_j}^s = \max(v_i^s | v_i^s \leq v_j^h), j = 1, \dots, n \quad (\text{A.12})$$

$$\hat{v}_{ij}^s = v_{n_j}^s + (v_{m_j}^s - v_{n_j}^s) \frac{r_j^h - r_{n_j}^s}{r_{m_j}^s - r_{n_j}^s}, j = 1, \dots, n \quad (\text{A.13})$$

$$\hat{r}_{ij}^s = r_{n_j}^s + (r_{m_j}^s - r_{n_j}^s) \frac{v_j^h - v_{n_j}^s}{v_{m_j}^s - v_{n_j}^s}, j = 1, \dots, n. \quad (\text{A.14})$$

The percentages regarding the variance deviation (φ_j), return (Ψ_j) and standard deviation (β) for any point obtained (v_j^h, r_j^h) are calculated as follows:

$$\varphi_j = 100 \frac{|v_j^h - \hat{v}_{ij}^s|}{\hat{v}_{ij}^s}, j = 1, \dots, n \quad (\text{A.15})$$

$$\Psi_j = 100 \frac{|r_j^h - \hat{r}_{ij}^s|}{\hat{r}_{ij}^s}, j = 1, \dots, n \quad (\text{A.16})$$

$$\beta_j = 100 \frac{|\sqrt{v_j^h} - \sqrt{\hat{v}_{ij}^s}|}{\sqrt{\hat{v}_{ij}^s}}, j = 1, \dots, n. \quad (\text{A.17})$$

Thus, from Equations (A.16) and (A.17) it is possible to determine the mean percentage error (MPE), as well as the minimum (MinPE), the maximum (MaxPE) and the median (MedPE):

$$MPE = \frac{\sum_{j=1}^n \min(\beta_j, \Psi_j)}{n} \quad (\text{A.18})$$

$$\text{MinPE} = \arg \min_{j=1, \dots, n} (\min(\beta_j, \Psi_j)) \quad (\text{A.19})$$

$$\text{MaxPE} = \arg \max_{j=1, \dots, n} (\min(\beta_j, \Psi_j)). \quad (\text{A.20})$$