Annals of Operations Research
https://doi.org/10.1007/s10479-026-07149-2

ORIGINAL RESEARCH

Fast hierarchical risk parity methods for portfolio selection

Francisco Salas-Molina[1] · **Jordi Nin**[2]

Received: 23 January 2025 / Accepted: 24 February 2026

Abstract

Hierarchical Risk Parity methods address some of the limitations of the classical mean-variance approach to portfolio selection by deriving a hierarchical structure. These methods are based on hierarchical clustering techniques and the recursive bisection of an ordered list of assets. When the number of assets is large, computational time becomes a limitation. This paper finds invariants of the allocation produced by simple asset permutations. We also study the size of the decision space to improve the understanding of the allocation algorithm. Building on these results, we propose a fast hierarchical risk parity portfolio selection method that reduces computational time while ensuring a similar performance.

Keywords Asset allocation · Permutations · Decision space · Complexity · Computational time

1 Introduction

Asset allocation deals with the problem of selecting the proportion of a given budget that should be allocated to assets to obtain a portfolio such that the expected return is maximized for a given level of risk. Modern portfolio theory was initiated with the mean-variance model by Markowitz (1952) and has become a popular and very competitive research topic with many recent proposals from different perspectives.

The relevant literature related to this article includes, but is not limited to, the following works. Kan et al. (2022) proposed an optimal combination strategy to mitigate the risk of estimation of the classical mean-variance portfolio in the case without a risk-free asset. Cesarone et al. (2023) formulated a mixed-integer quadratic programming problem to solve the portfolio selection model characterized by three criteria: expected return, variance, and VaR at a specified confidence level. Qi and Steuer (2025) followed an analytical approach to derive a closed-form formula for the calculation of the properly efficient and weakly efficient sets of portfolio selection problems with multiple objectives restricted to equality

✉ Francisco Salas-Molina
frasamo@upv.es

Jordi Nin
jordi.nin@esade.edu

1 Department of Economics and Social Sciences, Universitat Politènica de València, Alcoy 03801, Spain

2 Universitat Ramon Llull, Esade, Avinguda de la Torre Blanca, 59, 08173 Sant Cugat del Vallès, Catalonia, Spain

Published online: 09 March 2026

constraints. Lakshmi and Kumara (2024) combined the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) technique and the Randomized Weighted Fuzzy Analytic Hierarchy Process (AHP) to improve the robustness of selecting the best stocks for a portfolio. Georgantas et al. (2024) also followed a robust optimization approach by incorporating uncertainty through a formal and analytical approach into the modeling to perform a comparative analysis of their performance using data from the US market during the period 2005–2020. Finally, López de Prado et al. (2025) showed how to improve portfolio management processes by combining traditional econometrics with machine learning techniques. This last work is closely related to the approach in our proposal.

López de Prado (2016) introduced the Hierarchical Risk Parity (HRP) method to build a diversified portfolio through hierarchical clustering, a well established unsupervised machine learning technique. The author claimed to solve some limitations of Markowitz's Critical Line Algorithm (Markowitz, 1959; Markowitz et al., 1956), such as instability, concentration, and out-of-sample underperformance (López De Prado, 2018). With regard to instability, small changes in any entry in the covariance matrix due to estimation errors may lead to unstable solutions when computing its inverse. This problem is especially prevalent when considering highly correlated investments. In addition, forecasting returns with sufficient accuracy is often considered a very difficult task, leading to the development of methods based only on the covariance matrix (Jurczenko, 2015; López De Prado, 2018; Roncalli, 2013). Recent applications of the HRP method are described in Molyboga (2020); Burggraf (2021); Sen and Dutta (2022), and Cho and Song (2023).

HRP is part of the class of portfolio selection models based on risk parity (Roncalli, 2013). Risk parity methods are heuristics based on building a balanced portfolio so that the risk contribution is the same for different assets. As a result, only covariance estimators are used to compute asset allocations, while return estimations are ignored. More precisely, the HRP algorithm described in López de Prado (2016) is based on three steps: 1) tree clustering, 2) quasi-diagonalization, and 3) recursive bisection. The output of the second step is a sorted list of assets that, by recursive bisection, is used to obtain a solution to the problem through an inverse-variance allocation procedure.

To improve the understanding of the HRP algorithm, our aim is to find transformations of the list of assets that keep the solution proposed by the HRP algorithm unchanged. To this end, we rely on the concept of invariance. Invariant theory dates back to Gauss and Hilbert (1890) and deals with the explicit description of polynomial functions that do not change or are invariant under some transformations from a given group (Derksen & Kemper, 2015; Hilbert, 1993). Here, we use a broad definition of invariance to examine how permutations of a sorted asset list function as a group of transformations that leave the HRP results unchanged. Our initial finding demonstrates that HRP weights remain invariant as long as the sorted list of assets is preserved, regardless of other parameters in the algorithm. Specifically, we show that any commutative permutation of the list does not affect the allocation when the number of assets in the list is even.

This result leads us to study the size of the decision space explored by the HRP algorithm in direct comparison to alternative portfolio selection models such as the mean-variance model (Markowitz, 1952), and the inverse-variance portfolio (López De Prado, 2018), and the equally weighted portfolio (DeMiguel et al., 2009). In theory, the decision space for the mean-variance model is infinite because there are an infinite number of possible combinations of N real numbers in the interval $[0, 1]$ expressing the allocation for N assets as a potential solution. In contrast, the decision space of an inverse-variance portfolio and an equally weighted portfolio has a single possible solution corresponding to assigning the same proportion to all assets. We show that the HRP algorithm explores solutions in an intermediate-sized decision space.

These insights influence the algorithm's efficiency, particularly in terms of computational time, directly affecting scalability and real-time decision-making. The initial step of the traditional HRP algorithm, hierarchical tree clustering, is computationally intensive. We hypothesize that a more efficient version of HRP can be developed without sacrificing performance. To address this, we introduce the Fast Hierarchical Risk Parity (FHRP) method, which reduces computational time by replacing hierarchical clustering with a correlation-based ranking. To validate our approach, we demonstrate that FHRP significantly reduces computational time while delivering performance comparable to the classic HRP.

We summarize the main contributions of this paper as follows:

- We show that the HRP portfolio weights remain unchanged under specific permutations of the sorted asset list, deepening the understanding of the algorithm.
- This result enables us to define the decision space of the HRP algorithm, providing insights into its computational complexity.
- Finally, we introduce the Fast Hierarchical Risk Parity (FHRP), which significantly reduces computational time while preserving the performance of traditional HRP.

In addition to this introduction, we provide the background on HRP methods in Sect. 2 to contextualize our contribution. Later, we describe our theoretical results in Sect. 3. Building on these results, we introduce a fast hierarchical risk parity algorithm in Sect. 4. Finally, in Sect. 5, we conclude this paper by analyzing the implications of our findings and suggesting potential directions for future research.

2 Background on hierarchical risk parity methods

The Hierarchical Risk Parity (HRP) method, introduced by López de Prado (2016); López De Prado (2018), offers an alternative approach to portfolio allocation that addresses some of the limitations of traditional optimization techniques. The HRP method is based on three steps, each designed to ensure a more stable and diversified portfolio. These steps include hierarchical tree clustering, quasi-diagonalization, and recursive bisection. In this section, we explore each of these components in detail and explain their role in the overall HRP approach.

2.1 Hierarchical tree clustering

This first step combines a matrix of observations, such as returns series of N variables over T periods, into a hierarchical structure of clusters. Hierarchical clustering aims to group similar data points based on a similarity metric. Specifically, HRP uses the single linkage clustering strategy, where the distance between two clusters is defined as the shortest distance between any two points in the two clusters. In other words, the linkage criterion is based on the minimum pairwise distance between the members of different clusters. Here is a step-by-step overview of how it works:

1. Begin with each asset as its cluster.
2. Build the proximity matrix by computing the pairwise distances between all clusters.
3. Merge the two clusters that have the smallest pairwise distance, forming a new cluster.

4. Update the proximity matrix by deleting the rows and columns corresponding to the elements of the new cluster and adding a row and column corresponding to the newly formed cluster following the single-linkage clustering strategy.
5. Repeat steps 2 and 3 until all assets are part of a single cluster.

HRP performs the clustering algorithm by calculating the pairwise distance between assets using the formula $d_{ij} = \sqrt{0.5(1 - \rho_{ij})}$, where ρ_{ij} is the Pearson correlation coefficient between assets i and j. Once two assets are merged into a cluster, their corresponding columns are removed from the correlation matrix, and a new column is added, representing the minimum correlation between the newly formed cluster and the remaining assets. The main result of the tree clustering algorithm in HRP is a $(N - 1) \times 4$ linkage matrix with the following structure:

$$Y = \{(y_{m,1}, y_{m,2}, y_{m,3}, y_{m,4})\}_{m=1,\dots,N-1} \tag{1}$$

where $y_{m,1}$ and $y_{m,2}$ are the constituents of the cluster, $y_{m,3} = \tilde{d}_{y_{m,1},y_{m,2}}$ is the distance between $y_{m,1}$ and $y_{m,2}$, and $y_{m,4}$ is the number of original items included in cluster m.

2.2 Quasi-diagonalization

The second step reorganizes the rows and columns of the covariance matrix V, aligning the largest values along the main diagonal. This process retains the hierarchical clustering order, producing a sorted list of the original (unclustered) items. Specifically, the quasi-diagonalization operates as follows:

1. Initialize a list L_0 with the constituents of the last row of the linkage matrix Y.
2. While the maximum value in L_0 is less than N do:
   (a) Identify the cluster labeled with a number that is greater than or equal to the total number of original items.
   (b) Add the constituents of the identified cluster to the list.
3. Return list L_0 with the assets reorganized.

This step of the HRP method takes a hierarchical clustering linkage matrix, identifies its clusters and their constituents, and returns a reorganized list of assets.

2.3 Recursive bisection

Finally, the reorganized list L_0 of assets from the quasi-diagonalization step is used to obtain the portfolio weights w_n for all $1 \leq n \leq N$ through the following recursive inverse-variance allocation procedure:

1. Initialization:
   (a) Set the list of items $L = \{L_0\}$, with $L_0 = \{n\}_{n=1,\dots,N}$.
   (b) Assign unit weights to all items: $w_n = 1, \forall n = 1, \dots, N$.
2. If $|L_i| = 1, \forall L_i \in L$, then stop.
3. For each $L_i \in L$ such that $|L_i| > 1$:
   (a) Bisect L_i into two subsets, $L_i^{(1)} \cup L_i^{(2)}$, where $|L_i^{(1)}| = \text{int}[\frac{1}{2}|L_i|]$, and the order is preserved.

(b) Define the variance of $L_i^{(j)}$, $j = 1, 2$, as the quadratic form:

$$\tilde{V}_i^j \equiv \tilde{w}_i^{(j)'} V_i^{(j)} \tilde{w}_i^{(j)} \tag{2}$$

where $V_i^{(j)}$ is the covariance matrix between the constituents of the $L_i^{(j)}$ bisection, and weights are set to:

$$\tilde{w}_i^{(j)} = \text{diag}[V_i^{(j)}]^{-1} \frac{1}{\text{tr}[\text{diag}[V_i^{(j)}]^{-1}]} \tag{3}$$

where diag[.] and tr[.] are, respectively, the diagonal and trace operators.

(c) Compute split factor α_i so that $0 \leq \alpha_i \leq 1$:

$$\alpha_i = 1 - \frac{\tilde{V}_i^{(1)}}{\tilde{V}_i^{(1)} + \tilde{V}_i^{(2)}}. \tag{4}$$

(d) Re-scale allocations w_n by a factor of α_i, $\forall n \in L_i^{(1)}$.

(e) Re-scale allocations w_n by a factor of $(1 - \alpha_i)$, $\forall n \in L_i^{(2)}$.

4. Loop to step 2.

In summary, the recursive bisection step determines the variance of each partition using inverse-variance weights, with split weights inversely proportional to the cluster's variance. This approach ensures that the total sum of weights equals one while maintaining non-negativity.

2.4 Computational complexity analysis

To assess the overall computational complexity of the standard Hierarchical Risk Parity (HRP) algorithm, it is necessary to analyze the cost of each of its constituent steps. The final asset allocation is performed via recursive bisection, which has a logarithmic complexity of $\mathcal{O}(\log n)$ per split. The quasi-diagonalization step, which determines the reordering of assets based on the hierarchical structure, requires traversing the clustering tree and repeatedly sorting an increasingly large list. Since sorting has a complexity of $\mathcal{O}(\log n)$ and this operation is performed over n assets, the total cost of quasi-diagonalization is approximately $\mathcal{O}(n \log n)$.

However, the most computationally intensive step is the construction of the hierarchical clustering tree. When single-linkage clustering is used, this step has a time complexity of $\mathcal{O}(n^2)$ (Sibson, 1973). As this dominates the overall cost, the total computational complexity of the standard HRP algorithm is $\mathcal{O}(n^2)$.

3 Invariants in HRP methods

After a brief motivation, in this section we define invariants and related concepts within the context of HRP and portfolio selection. Next, we derive theoretical results from these definitions, which help practitioners better understand the HRP algorithm and its implications for portfolio selection.

3.1 Motivation

By analyzing the steps of the HRP method described in Sect. 2, we realize that the ordered list of assets derived from hierarchical clustering and the quasi-diagonalization of the covariance matrix have several implications in the final set of weights that form the solution to a given portfolio selection problem. First, if the order of assets from quasi-diagonalization of the covariance matrix determines the solution, an interesting research question is whether a solution remains invariant for two (or more) different orders. Second, as a direct consequence of invariant solutions, it is important to explore the size of the decision space from a theoretical and practical perspective. Finally, analysis of the decision space is closely related to the potential for improving computational efficiency.

3.2 Definitions

The study of invariants (Derksen & Kemper, 2015; Hilbert, 1993) is closely related to the classification of mathematical objects. In our context, these objects are the N-tuples of assets that the HRP algorithm uses to generate a weight vector, which serves as the solution to the portfolio selection problem.

Definition 1 An **invariant** is a mapping $\phi : \mathcal{Q} \to \mathcal{W}$ of a set $\mathcal{Q}$ of mathematical objects endowed with a fixed equivalence relation R into another set $\mathcal{W}$ that is constant on the equivalence classes of $\mathcal{Q}$ for R.

In other words, an invariant is a property that remains unchanged under a specific transformation or operation. For example, the determinant and the trace of a square matrix are invariant under a change of basis.

When considering the HRP model, $\mathcal{Q}$ is the set of all ordered N-tuples $\boldsymbol{q} = (a_1, \ldots, a_N) \in \mathcal{Q}$ for the N different assets produced by the quasi-diagonalization step of the HRP algorithm. The set $\mathcal{W}$ is a subset of $\mathbb{R}^N$ with all possible weight solutions to the portfolio selection problem. To obtain these solutions, the HRP algorithm maps a sorted list of securities encoded in the vector $\boldsymbol{q} \in \mathcal{Q}$ to $\boldsymbol{w} \in \mathcal{W}$ through an inverse-variance allocation procedure by the recursive bisection of $\boldsymbol{q}$, given a covariance matrix $V \in \mathbb{R}^{N \times N}$.

Definition 2 An **equivalence relation** R in $\mathcal{Q}$ in the context of HRP can be defined by the following rule: $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathcal{Q}$ are R equal if and only if $w(\boldsymbol{q}_1) = w(\boldsymbol{q}_2)$, where $w : \mathcal{Q} \times \mathbb{R}^{N \times N} \to \mathcal{W}$ is the HRP function mapping vectors in $\mathcal{Q}$ and covariance matrices V in $\mathbb{R}^{N \times N}$ to the decision space $\mathcal{W}$.

For example, the vector $\boldsymbol{q}_1 = (a_1, a_2, a_3)$ is equivalent to $\boldsymbol{q}_2 = (a_1, a_3, a_2)$ if and only if $w(\boldsymbol{q}_1) = w(\boldsymbol{q}_2)$. Typically, the equivalence relation is constrained to a specific group of transformations. In this paper, we define this group as the set of permutations of the N-tuples in $\mathcal{Q}$.

Definition 3 The **group of transformations** G is defined by the permutation function $\pi : \mathcal{Q} \to \mathcal{Q}$, such that for every integer $i \in \{1, 2, \ldots, N\}$, there is exactly one integer $j \in \{1, 2, \ldots, N\}$ for which $\pi(j) = i$.

Consequently, there is always $\pi \in G$ such that $\boldsymbol{q}_1 = \pi(\boldsymbol{q}_2)$ for all $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathcal{Q}$, and the invariants that arise are called invariants of transformation $\pi \in G$. Following our example with vectors $\boldsymbol{q}_1 = (a_1, a_2, a_3)$, with $i = 1, 2, 3$, and $\boldsymbol{q}_2 = (a_1, a_3, a_2)$, with $j = 1, 3, 2$, we observe that $\boldsymbol{q}_1 = \pi(\boldsymbol{q}_2)$ when $\pi(1) = 1, \pi(3) = 2$, and $\pi(2) = 3$.

3.3 Results

HRP allocations w_n depend on the list of assets L equivalent to an N-dimensional vector $\boldsymbol{q}$. Our first result related to invariants states that the same ordered set of assets derived from the quasi-diagonalization step of the HRP algorithm produces the same solution weights for a given input covariance matrix.

Lemma 1 *Given an arbitrary list of N assets* $\boldsymbol{q} = \{a_1, \ldots, a_n, \ldots, a_N\}$*, characterized by the covariance matrix* $V = \left[V_{ij}\right]$*, where* V_{ij} *is the covariance of assets i and j in* $\boldsymbol{q}$*, the HRP recursive bisection of two equal permutations* $\boldsymbol{q}_1 = \pi_1(\boldsymbol{q})$ *and* $\boldsymbol{q}_2 = \pi_2(\boldsymbol{q})$ *such that* $\pi_1(\boldsymbol{q}) = \pi_2(\boldsymbol{q})$ *produces the same resulting weights* $w_n(\boldsymbol{q}_1) = w_n(\boldsymbol{q}_1)$*.*

Proof This lemma follows directly from the HRP algorithm, which is fundamentally a function of the covariance matrix, V, and the ordered set of assets q derived from the quasi-diagonalization of this matrix. The recursive bisection procedure produces identical results, as we assumed the same covariance matrix V and the same asset ordering q from the quasi-diagonalization process. □

From Lemma 1, the following result establishes a commutative property for the recursive bisection of a given ordered list of assets.

Theorem 1 *Given covariance matrix V and set* $\boldsymbol{q} = \{a_1, \ldots, a_n, \ldots, a_N\}$*. If N is even, define the bisections* $\boldsymbol{q}^{(1)} = \{a_1, \ldots, a_\beta\}$ *and* $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$*, where* $\beta = int[N/2]$*, such that* $\boldsymbol{q} = \{\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}\}$*. If N is odd, define the bisections* $\boldsymbol{q}^{(1)} = \{a_0, a_1, \ldots, a_\beta\}$ *and* $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$*, where* a_0 *is a dummy asset with infinite variance and zero covariance with the rest of the assets in* $\boldsymbol{q}$*. Subsequently, if we define a commutative set* $\boldsymbol{q}' = \{\boldsymbol{q}^{(2)}, \boldsymbol{q}^{(1)}\}$*, then* $w_n(\boldsymbol{q}) = w_n(\boldsymbol{q}')$*, for every asset* a_n*.*

Proof The number N of assets can be even or odd:

- N even. Consider the case $N = 2$ with $\boldsymbol{q} = \{a_1, a_2\}$. We know that w_1 and w_2 are invariant, disregarding the order of the elements of $\boldsymbol{q}$ since weights depend only on variances V_1 and V_2. For $N \geq 4$, define $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ as virtual assets derived from the HRP algorithm and use Lemma 1 to observe that the same ordered list of assets produces the same weights also for $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ taken independently. Likewise for case $N = 2$, $w_n(\boldsymbol{q}^{(1)})$ and $w_n(\boldsymbol{q}^{(2)})$ map the same weights to the same assets disregarding that $L_1^{(1)} = \boldsymbol{q}^{(1)}$ or $L_1^{(1)} = \boldsymbol{q}^{(2)}$ in the first bisection of $\boldsymbol{q}$. After $N - 1$ bisections, a different factor α_i re-scales allocations of the first-order bisections $\boldsymbol{q}^{(1)}$ and $\boldsymbol{q}^{(2)}$ to ultimately obtain $w_n(\boldsymbol{q}) = w_n(\boldsymbol{q}')$.
- N odd. Consider the case of $N = 1$ with $\boldsymbol{q} = \{a_0, a_1\}$. The recursive bisection algorithm produces $w_0 = 0$ and $w_1 = 1$, because of the infinite variance and zero correlation for the dummy asset a_0. For $N = 3$, define $\boldsymbol{q}^{(1)} = \{a_0, a_1\}$ and $\boldsymbol{q}^{(2)} = \{a_2, a_3\}$, and apply the case $N = 4$ above. For $N \geq 5$, define $\boldsymbol{q}^{(1)} = \{a_0, a_1, \ldots, a_\beta\}$ and $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$, and follow the same reasoning to observe that $w_n(\boldsymbol{q}) = w_n(\boldsymbol{q}')$.

□

It is important to note that standard HRP does not include any dummy asset. The case for N odd in Theorem 1 when a dummy asset is added is purely theoretical. We follow this approach to consider both cases, even though the theoretical results are different, as we further elaborate.

Corollary 1 *HRP weights are invariant for the commutative permutation of first-order bisections* $\boldsymbol{q} = \{\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}\}$ *described by the transformation* $\boldsymbol{q}' = \pi(\boldsymbol{q}) = \{\boldsymbol{q}^{(2)}, \boldsymbol{q}^{(1)}\}$*, where* $\boldsymbol{q}^{(1)} = \{a_1, \ldots, a_\beta\}$ *and* $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$ *for even N values, and* $\boldsymbol{q}^{(1)} = \{a_0, a_1, \ldots, a_\beta\}$ *and* $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$ *for N odd.*

Proof Directly from Theorem 1. □

Corollary 2 *HRP weights are not invariant for the commutative permutation of first-order bisections* $\boldsymbol{q} = \{\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}\}$ *described by transformation* $\boldsymbol{q}' = \pi(\boldsymbol{q}) = \{\boldsymbol{q}^{(2)}, \boldsymbol{q}^{(1)}\}$*, where* $\boldsymbol{q}^{(1)} = \{a_1, \ldots, a_\beta\}$ *and* $\boldsymbol{q}^{(2)} = \{a_{\beta+1}, \ldots, a_N\}$ *for N odd.*

Proof Directly from Theorem 1. □

In the case of corollary 2 is consistent with Theorem 1 because we are dealing with an odd number of assets without considering any dummy asset. The previous results lead to the analysis of the size of the decision space for the HRP algorithm in direct comparison with alternative portfolio selection models. At least in theory, the size of the decision space for the mean-variance portfolio selection model by Markowitz (1952) is equal to infinite because there are infinite possible combinations of N real numbers in the interval [0, 1]. However, the inverse variance portfolio is computed as follows:

$$w_i = \frac{1/V_i}{\sum_{j=1}^{N} 1/V_j} \tag{5}$$

where w_i is the weight assigned to the i-th asset, and V_i is the estimated variance of the i-th asset. Given the covariance matrix V, the size of the decision space is equal to one because there is a unique possible solution encoded in a vector of size N. Similarly, the size of the decision space for an equally weighted portfolio is one because the solution is given by a single scalar $1/N$.

Between the infiniteness of the decision space for the mean-variance portfolio selection model by Markowitz (1952) and the reduced size of an inverse-variance portfolio and an equally weighted portfolio, the HRP algorithm tends to look for solutions in an intermediate-sized decision space as the following results show.

Lemma 2 *Given an* $N \times N$ *covariance matrix V for N different assets, the size of the decision space for the HRP algorithm is less than* $N!$.

Proof It follows directly from Lemma 1 because the number of permutations of N distinct objects is $N!$ because there is always at least a permutation of the last bisection for a cluster of size two that produces the same results according to Theorem 1. □

However, we can provide more precise results for $2 \leq N \leq 4$. Given an $N \times N$ covariance matrix V, if $N = 2$, then the size of the decision space is one because there is only one possible solution for each of the two assets as a direct consequence of Theorem 1. For $N = 3$, we find the following invariants:

Theorem 2 *Given an* $N \times N$ *covariance matrix V, if* $N = 3$*, then the HRP weights are invariant for the permutations* $\boldsymbol{q}_1 = \{a_1, a_2, a_3\}$ *and* $\boldsymbol{q}_2 = \{a_1, a_3, a_2\}$*, for the permutations* $\boldsymbol{q}_3 = \{a_2, a_1, a_3\}$ *and* $\boldsymbol{q}_4 = \{a_2, a_3, a_1\}$*, and for the permutations* $\boldsymbol{q}_5 = \{a_3, a_1, a_2\}$ *and* $\boldsymbol{q}_6 = \{a_3, a_2, a_1\}$.

Table 1 HRP weights before and after the commutative reorder of the resulting list of 34 assets in IBEX35

Index	1	2	3	...	17	18	19	20	...	34
Ticker	PRS	OHL	TRE	...	TL5	TEF	MAP	BBVA	...	GRF
Weight	0.013	0.008	0.042	...	0.019	0.027	0.026	0.018	...	0.025
Ticker	TEF	MAP	BBVA	...	GRF	PRS	OHL	TRE	...	TL5
Weight	0.027	0.026	0.018	...	0.025	0.013	0.008	0.042	...	0.019

Proof The HRP weights are invariant for the permutations $\boldsymbol{q}_1 = \{a_1, a_2, a_3\}$ and $\boldsymbol{q}_2 = \{a_1, a_3, a_2\}$, because after the first bisection for $\boldsymbol{q}_1$, we have $L_0^{(1)} = \{a_1\}$ and $L_0^{(2)} = \{a_2, a_3\}$. For $\boldsymbol{q}_2$, we have $L_0^{(1)} = \{a_1\}$ and $L_0^{(2)} = \{a_3, a_2\}$. From Theorem 1, the weight allocation is the same for $L_0^{(2)} = \{a_2, a_3\}$ and $L_0^{(2)} = \{a_3, a_2\}$. A similar reasoning applies to the pairs $\boldsymbol{q}_3$ and $\boldsymbol{q}_4$, and $\boldsymbol{q}_5$ and $\boldsymbol{q}_6$. □

Corollary 3 *For a covariance matrix V of size $N \times N$, if $N = 3$, the decision space consists of exactly three possible solutions, each corresponding to selecting one of the three individual assets.*

Proof Directly from Theorem 2. □

Theorem 3 *For a covariance matrix V of size $N \times N$, if $N = 4$, the decision space has three options, as there are only three possible solutions corresponding to the four assets.*

Proof When $N = 4$, HRP weights are invariant for the commutative permutations of the first-order bisections. That is, $L_0^{(1)} = \{a_1, a_2\}$ and $L_0^{(1)} = \{a_2, a_1\}$ produce the same results according to Theorem 1 disregarding the order in $L_0^{(2)}$. For each of these two bisections, $L_0^{(2)} = \{a_3, a_4\}$ and $L_0^{(2)} = \{a_4, a_3\}$ also produce the same results. As a result, we have four permutations with the same results. A similar reasoning allows us to identify four more invariants when $L_0^{(1)} = \{a_1, a_3\}$ or $L_0^{(1)} = \{a_3, a_1\}$, and four more when $L_0^{(1)} = \{a_1, a_4\}$ or $L_0^{(1)} = \{a_4, a_1\}$. As a result, from the $4! = 24$ possible permutations of four elements, only three produce different results. □

We leave the theoretical study of ordered sets of assets for $N > 4$ for further research. However, we include some extra analytical results in the following subsection.

3.4 Numerical illustration of the theoretical results

To illustrate Theorem 1 for commutative bisections of an ordered list of assets with even cardinality, we use real-world data from the Spanish market index IBEX35. Using 259 weekly return observations from 2014 to 2019 for 34 assets, we calculate both the covariance and correlation matrices, which serve as key inputs for the HRP algorithm. The resulting ordered list of assets (represented by their tickers) and their corresponding weights are partially shown in the first block of Table 1. Next, we perform a commutative reordering by bisecting the list, placing the second bisection first and the first bisection second. When we recompute the asset weights, the allocation remains unchanged, as predicted by Theorem 1.

To illustrate Theorem 2 for ordered sets of size $N = 3$, we follow López de Prado (2016) to generate synthetic correlated returns from a Gaussian distribution with mean and standard

Table 2 HRP weights for all permutations of a list of assets of size $N = 3$

Permutation	w_1	w_2	w_3
(1, 2, 3)	0.98576	0.00478	0.00946
(1, 3, 2)	0.98576	0.00478	0.00946
(2, 1, 3)	0.97651	0.00786	0.01563
(2, 3, 1)	0.97651	0.00786	0.01563
(3, 1, 2)	0.97645	0.00789	0.01566
(3, 2, 1)	0.97645	0.00789	0.01566

Table 3 HRP weights for all permutations of a list of assets of size $N = 4$.

Permutation	w_1	w_2	w_3	w_4
(2, 4, 1, 3)	0.52623	0.00410	0.00814	0.46152
(2, 4, 3, 1)	0.52623	0.00410	0.00814	0.46152
(1, 3, 2, 4)	0.52623	0.00410	0.00814	0.46152
(1, 3, 4, 2)	0.52623	0.00410	0.00814	0.46152
(3, 1, 2, 4)	0.52623	0.00410	0.00814	0.46152
(3, 1, 4, 2)	0.52623	0.00410	0.00814	0.46152
(4, 2, 1, 3)	0.52623	0.00410	0.00814	0.46152
(4, 2, 3, 1)	0.52623	0.00410	0.00814	0.46152
(1, 4, 2, 3)	0.52824	0.00231	0.00457	0.46488
(1, 4, 3, 2)	0.52824	0.00231	0.00457	0.46488
(2, 3, 1, 4)	0.52824	0.00231	0.00457	0.46488
(3, 2, 1, 4)	0.52824	0.00231	0.00457	0.46488
(2, 3, 4, 1)	0.52824	0.00231	0.00457	0.46488
(3, 2, 4, 1)	0.52824	0.00231	0.00457	0.46488
(4, 1, 2, 3)	0.52824	0.00231	0.00457	0.46488
(4, 1, 3, 2)	0.52824	0.00231	0.00457	0.46488
(3, 4, 1, 2)	0.52954	0.00414	0.00806	0.45826
(3, 4, 2, 1)	0.52954	0.00414	0.00806	0.45826
(4, 3, 1, 2)	0.52954	0.00414	0.00806	0.45826
(4, 3, 2, 1)	0.52954	0.00414	0.00806	0.45826
(1, 2, 3, 4)	0.52954	0.00414	0.00806	0.45826
(1, 2, 4, 3)	0.52954	0.00414	0.00806	0.45826
(2, 1, 3, 4)	0.52954	0.00414	0.00806	0.45826
(2, 1, 4, 3)	0.52954	0.00414	0.00806	0.45826

deviation set to historical values of the S&P 500 market index. We compute the weights for all permutations in Table 2. Despite the slight differences in the weights, we observe that there are three distinct solutions. Moreover, the weights remain unchanged when the positions of the last two elements in the list are swapped.

We follow a similar approach to verify Theorem 3 for ordered sets of size $N = 4$. In this case, we organize Table 3 into blocks, making it clear that only three distinct solutions exist. Once again, we confirm that Theorem 1 holds for commutative permutations by comparing, for example, the first and third rows of Table 3.

Table 4 Number of distinct HRP solutions for $N \geq 3$ assets

N	Distinct solutions	Permutations
3	3	6
4	3	24
5	30	120
6	90	720
7	315	5,040
8	315	40,320
9	11,340	362,880
10	113,400	3,628,800

To extend our numerical analysis, we generated 50 synthetic datasets using the same data generation process described in López de Prado (2016). For each dataset, we enumerated all possible asset ordering permutations and applied the recursive bisection step of the HRP algorithm to compute the corresponding portfolio weights. We then determined the total number of distinct weight configurations resulting from these permutations. The results with $N \leq 10$ are presented in Table 4.

An analysis of these results reveals that the number of unique solutions produced by the HRP algorithm grows significantly more slowly than the total number of possible asset permutations. This suggests that the recursive bisection step, rather than the hierarchical clustering tree structure, plays a dominant role in shaping the final allocation. It is important to recall that the clustering step is the most computationally intensive part of the HRP algorithm, with complexity $\mathcal{O}(n^2)$. These findings indicate that exploring alternative, less costly methods for generating asset orderings could be a promising direction, as the recursive bisection step appears to be the primary driver of the resulting portfolio weights.

3.5 Need for a faster algorithm

Both theoretical and numerical reductions in the decision space of the HRP algorithm offer significant potential to improve computational efficiency. This approach is grounded in two key insights: first, the theoretical results show that certain permutations of an asset list yield identical weight allocations, reducing the number of permutations that need evaluation and enabling faster computations. Second, a smaller decision space eliminates redundant calculations tied to equivalent permutations, further reducing computational complexity. These efficiency gains are particularly interesting when working with large datasets or scenarios that require real-time portfolio adjustments.

4 Fast hierarchical risk parity models

Based on the findings in Sect. 3, we hypothesize that a faster HRP algorithm can be developed without sacrificing performance. To address this, we propose the Fast Hierarchical Risk Parity (FHRP) model, which replaces hierarchical clustering with a correlation-based asset ranking. Our results show that this approach significantly reduces computational time while maintaining comparable performance to the traditional HRP method.

4.1 Fast correlation-based HRP model

As noted in López De Prado (2018), the HRP allocation step described in Sect. 2.3 has a logarithmic computational complexity of $\mathcal{O}(\log_2[n])$. However, as discussed in Sect. 2, solving the complete allocation problem using HRP involves the calculation of a hierarchical tree clustering, which has a computational complexity of $\mathcal{O}(n^2)$ when using the single-linkage clustering algorithm (Sibson, 1973) as proposed in López de Prado (2016). While the full calculation of the HRP model is computationally feasible for medium-sized datasets, and occasionally for larger datasets, the significantly longer execution times pose a limitation in real-world scenarios, where managers often need to process large datasets efficiently.

In this paper, we introduce a Fast Hierarchical Risk Parity (FHRP) method to reduce computational time by replacing hierarchical clustering with a correlation-based ranking. Given a correlation matrix where the element (i, j) represents the correlation coefficient ρ_{ij}, the procedure is as follows:

1. Initialize the list of assets $L^* \leftarrow \{\emptyset\}$
2. For each asset $i = 1, \ldots, N$ find the maximum off-diagonal correlation

$$v_i \leftarrow \max_{i \neq j} \rho_{ij}.$$

3. Sort vector $\boldsymbol{v} = [v_i]$ in ascending order.
4. Set index k to the last element of sorted $\boldsymbol{v}$ with the highest correlation.
5. While $|L^*| < N$:
    - If asset i of v_k is not in list L^*, then add i to L^*.
    - If asset j of v_k is not in list L^*, then add j to L^*.
    - Set k to $k - 1$.
6. Return $L_0 = L^*$ as the input for the recursive bisection step.

Consequently, our FHRP algorithm generates a weight allocation that we will compare with the classical HRP algorithm in terms of both computational time and financial performance.

4.2 Computational complexity analysis

As in the complexity analysis of the HRP algorithm, we assume that the correlation matrix of the assets is already computed. Under this assumption, constructing vector $\boldsymbol{v} \in \mathbb{R}^N$, where each entry is defined as the maximum off-diagonal correlation in the corresponding row ($v_i \leftarrow \max_{j \neq i} \rho_{ij}$), does not introduce any additional asymptotic cost beyond reading the input correlation matrix itself. This operation can be performed on the fly while reading matrix ρ and therefore does not affect the complexity of the subsequent ordering step. In addition, the only additional memory required by this step is $\mathcal{O}(N)$, which corresponds to storing vector $\boldsymbol{v}$ with length N. The next step in the FHRP algorithm involves sorting vector $\boldsymbol{v}$. Standard comparison-based sorting algorithms require $\mathcal{O} = n \log n$ time.

Finally, the algorithm iterates through sorted vector $\boldsymbol{v}$ to construct the initial asset ordering L^*. This step adds at most one asset per iteration and terminates once all assets have been added, resulting in a complexity of $\mathcal{O} = n$.

Therefore, the overall theoretical complexity of the FHRP algorithm is dominated by the sorting step and is $\mathcal{O} = n \log n$, under the assumption that the correlation matrix is already available. Additional steps, such as building and traversing $\boldsymbol{v}$, contribute lower-order terms to the total cost.

4.3 Computational time and performance analysis

In the following analysis, we evaluate the advantages of four different HRP models using two datasets and two performance measures. Specifically, we generate synthetic returns to construct multiple correlation and covariance matrices for asset sets ranging from 10 to 2,000. Each correlation is randomly sampled from a uniform distribution with a minimum value (C_{min}) and a maximum value (C_{max}), which are based on historical data from real-world sources like the S&P 500 index. These synthetic correlation matrices are then used for the estimation the portfolio selection models. Additionally, we analyze a real-world dataset consisting of weekly returns for 530 assets within the S&P 500 index, covering the period from 2021-01-10 to 2024-12-31.

In our empirical study, we focus on comparing the computational time required to construct various portfolio selection models. Since the hierarchical clustering construction in the HRP algorithm is computationally intensive compared to simpler and faster models, such as the Inverse Variance Portfolio (IVP), we hypothesize that within the broader family of HRP models, it is possible to develop alternatives that significantly reduce computational time without compromising risk performance. As the primary purpose of HRP models is to manage risk, we evaluate risk performance based on the variance of out-of-sample returns. In summary, we assess the overall performance of these alternative models focusing on two key factors: computational time and return variance.

In addition to the classical HRP model, we consider three alternative models: our FHRP model, the Inverse Variance Portfolio (IVP) selection model, and a naive random HRP model. The HRP and the FHRP model were introduced in Sections 2 and 4.1, respectively. As a result, the IVP (López De Prado, 2018) is computed as follows:

$$w_i = \frac{1/\sigma_i^2}{\sum_{j=1}^{n} 1/\sigma_j^2} \tag{6}$$

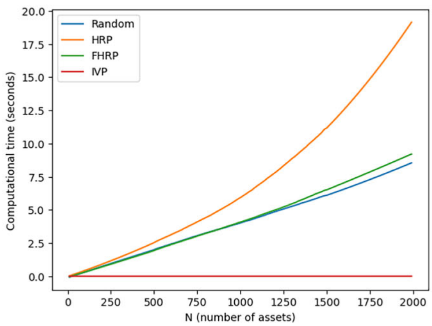
where w_i is the weight assigned to the i-th asset, and σ_i^2 is the estimated variance of the i-th asset. Finally, the random HRP model is a naive portfolio selection model in which the order of assets used by the recursive bisection step is established randomly.

Using the synthetic data set, we proceed as follows:

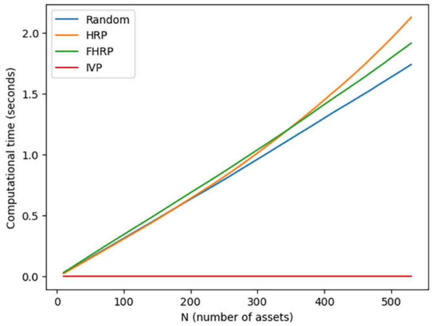
1. Set correlation parameters C_{min}, C_{max}, and maximum variance V_{max} for each asset.
2. Set number N of assets between 10 and 2, 000.
3. Generate a synthetic correlation matrix C of size $N \times N$ with elements sampled from a uniform distribution between C_{min} and C_{max}. Set elements of the main diagonal to 1.
4. Generate a synthetic covariance matrix V from C and V_{max}.
5. Find portfolio weights $\boldsymbol{w}$ with model m from C and V.
6. Record the computational time required to build the model and the resulting portfolio variance.

Using the S&P 500 data set, we proceed as follows:

1. Set number N of assets between 10 and 530 with steps of 10.
2. Sample returns for N assets from the data set.



(a) Synthetic data set



(b) S&P 500 data set

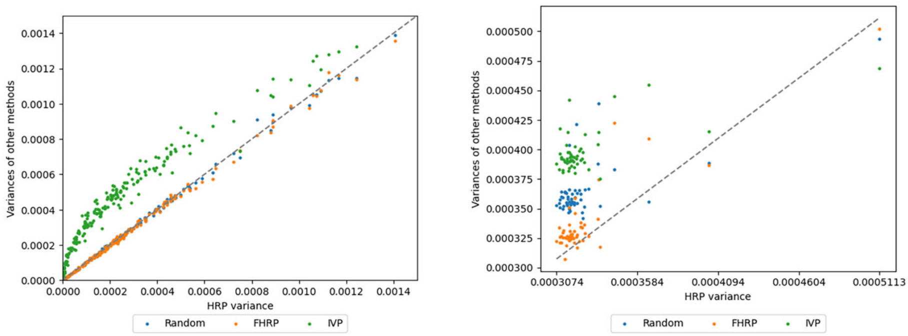
Fig. 1 Computational time for alternative HRP models

3. Select in-sample observations for training purposes to $F = 52$ weeks (equivalent to one year of data).
4. Compute portfolios in-sample for any method m using in-sample observations.
5. Select out-of-sample observations from time $t = F + 1$ to $t = F + B$ as the test set, where $B = 4$ weeks as the rebalance period for the portfolio.
6. Evaluate the out-of-sample variance of returns for all methods.
7. Move B steps forward and repeat from step 1 until the whole is covered.
8. Record the average computational time required to build the model and the resulting portfolio variance.

Computational time is affordable with a small number of assets, but becomes a critical factor when handling hundreds of assets or requiring frequent recalculations of the allocation. Figure 1 shows the computational time required to build portfolios for four different HRP models: the classical HRP model, our FHRP proposal, IVP, and random model as naive benchmarks. Focusing on synthetic data for a wide range of assets, we observe that both our FHRP method and the random benchmark show an almost linear increase in computational time as the number of assets increases. In contrast, as anticipated by the computational complexity of constructing a hierarchical tree clustering, the classical HRP model exhibits a significantly higher computational time, with a quadratic increase as the number of assets grows. As expected, the IVP computational time does not depend on the number of assets because only one computation is required to build the model.

When analyzing the results using the S&P 500 dataset, where the number of assets is limited to 530, we observe that the computational cost of the standard HRP algorithm begins to exhibit quadratic growth as the number of assets exceeds 400. In contrast, all other models, including our proposed FHRP method, continue to scale linearly. These findings, which are consistent across both real and synthetic datasets, support the conclusion that our FHRP method improves computational efficiency over the classical HRP model, particularly when constructing portfolios with a large number of assets (more than 400).

In addition to computational time, we assess the financial performance of the different HRP models, focusing on risk control, which is central to HRP. Specifically, we analyze the variance of the returns produced by each model as a measure of the performance of portfolio selection models aimed at minimizing risk. Figure 2 compares the global variance performance of the FHRP, Random, and IVP methods against the classical HRP. Each point



(a) Synthetic data set (b) S&P 500 data set

Fig. 2 Global variance performance comparison of alternative methods

represents an experiment with a varying number of assets, and the increasing diagonal reflects the performance of the HRP method.

For the synthetic dataset with a large number of assets, the IVP method consistently performs the worst, while the HRP, Random, and FHRP models show comparable results. In the S&P 500 dataset, which involves fewer assets, IVP again shows the weakest performance, and there are only slight differences between the FHRP and Random methods, largely due to the small scale of values on both axes.

Overall, our results indicate that the FHRP model performs similarly to the classical HRP in terms of risk (variance), particularly when dealing with a large number of assets, as seen in the synthetic dataset becoming an interesting alternative for practitioners who have to deal with large datasets or many clients.

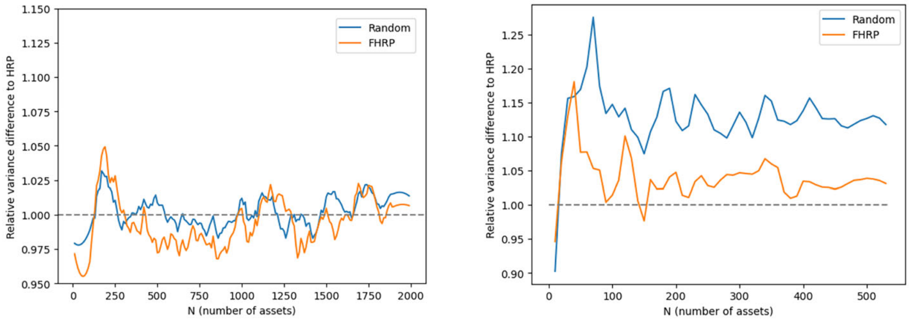
Figure 3 provides a detailed comparison of performance based on the number of assets. In this figure, the classical HRP method is represented by a dashed horizontal line, while the remaining lines depict the variance (risk) performance of alternative models. The IVP method is not included due to its poor performance, which is off the scale.

Our analysis reveals only minor differences across the HRP, Random, and FHRP models when using the synthetic dataset, with variations confined to the range [0.95, 1.05] across nearly all asset numbers. In addition, our FHRP method shows a performance advantage over the Random model and demonstrates improved results with the S&P 500 dataset. On the contrary, the classical HRP method outperforms the FHRP model, though this difference in performance remains consistent across the range of asset numbers.

As a result of the experiments described in this section, we conclude that our FHRP can significantly increase the computational efficiency of the classical HRP methodology without performance loss when the number of assets under consideration is up to 400.

4.4 General discussion

With the rapid growth in data generation and our enhanced ability to collect and store it, the need to reduce the computational costs of portfolio management algorithms has become increasingly critical. Advancing these algorithms is essential to address the challenges posed by this data-intensive reality in real-world scenarios (Shkolnik et al., 2025). Our research makes a significant contribution in this area with two key advancements.



(a) Synthetic data set

(b) S&P 500 data set

Fig. 3 Relative variance performance in terms of number of assets

First, from a theoretical perspective, we have increased the understanding of the HRP decision space, uncovering insights that enable the development of more computationally efficient alternatives. This theoretical framework lays the foundation for optimizing portfolio management methodologies. Second, from a practical perspective, we introduce a new, faster portfolio management algorithm designed for practitioners that is capable of delivering faster results for near real-time decision-making, making it highly relevant in dynamic market environments.

While our research is grounded on theoretical foundations, we also validate its applicability through a series of rigorous experiments using both synthetic and real-world datasets. These experiments demonstrate the practical utility and scalability of our proposed methods, reducing the gap between theoretical advancements and real-world implementation.

As a limitation, we must highlight that the computational improvements reported in this work are relevant when the number of assets under consideration is important (more than 400 assets). Furthermore, the improvements in computational time are within the range of seconds. As a result, our results become more relevant when the number of assets increases over a given threshold and when the frequency required to solve the problem is really an issue as in high-frequency trading.

5 Concluding remarks

In this paper, we study the invariants of the HRP algorithm for portfolio selection when alternative ordered lists are used to improve the algorithm's understanding and explore the size of the decision space. Our findings suggest that the size of the decision space derived from the HRP algorithm is located midway between the sizes of the classical mean-variance and other portfolio selection models. We refine this general result to specific small values for the size of the ordered list of assets and leave the study of more complex sets for further research. Finally, we must highlight that identifying invariants in a given algorithm for a group of transformations represents a line of work that can be generalized across different domains to open up opportunities for a broader range of applications.

These insights affect algorithmic efficiency (computational time), directly impacting scalability and real-time decision-making. As a result, we propose a fast hierarchical risk parity model (FHRP) based on a correlation-based ranking of assets, and we show that the computational time is reduced while keeping a similar performance to the classical HRP when the number of assets under consideration is above 400. Additional experiments on the stability of risk performance using larger real-world data sets with more assets represent an interesting future line of research.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was partially funded by the Fundación Ramon Areces through the project "Energy Aware Generative AI Model Deployment".

Declarations

Conflicts of Interest The authors declare that they have no conflict of interest.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

References

Burggraf, T. (2021). Beyond risk parity-a machine learning-based hierarchical risk parity approach on cryptocurrencies. *Finance Research Letters, 38*, Article 101523.

Cesarone, F., Martino, M. L., & Tardella, F. (2023). Mean-variance-var portfolios: MIQP formulation and performance analysis. *OR Spectrum, 45*(3), 1043–1069.

Cho, Y., & Song, J. W. (2023). Hierarchical risk parity using security selection based on peripheral assets of correlation-based minimum spanning trees. *Finance Research Letters, 53*, Article 103608.

DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *The Review of Financial Studies, 22*(5), 1915–1953.

Derksen, H., & Kemper, G. (2015). *Computational invariant theory*. Springer.

Georgantas, A., Doumpos, M., & Zopounidis, C. (2024). Robust optimization approaches for portfolio selection: A comparative analysis. *Annals of Operations Research, 339*(3), 1205–1221.

Hilbert, D. (1890). Ueber die theorie der algebraischen formen. Mathematische Annalen (pp. 473–534).

Hilbert, D. (1993). *Theory of algebraic invariants*. Cambridge University Press.

Jurczenko, E. (2015). *Risk-based and factor investing*. Amsterdam: Elsevier.

Kan, R., Wang, X., & Zhou, G. (2022). Optimal portfolio choice with estimation risk: No risk-free asset case. *Management Science, 68*(3), 2047–2068.

Lakshmi, K. V., & Kumara, K. U. (2024). A novel randomized weighted fuzzy ahp by using modified normalization with the topsis for optimal stock portfolio selection model integrated with an effective sensitive analysis. *Expert Systems with Applications, 243*, Article 122770.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91.

Markowitz, H. M. (1959). *Portfolio selection: Efficient diversification of investments*. New York: John Wiley and Sons.

Markowitz, H. M., et al. (1956). The optimization of a quadratic function subject to linear constraints. *Naval research logistics Quarterly, 3*(1–2), 111–133.

Molyboga, M. (2020). A modified hierarchical risk parity framework for portfolio management. *The Journal of Financial Data Science, 2*(3), 128–139.

López de Prado, M. (2016). Building diversified portfolios that outperform out of sample. *The Journal of Portfolio Management, 42*(4), 59–69.

López de Prado, M. (2018). Advances in financial machine learning. John Wiley & Sons.

López de Prado, M., Simonian, J., Fabozzi, F. A., & Fabozzi, F. J. (2025). Enhancing Markowitz's portfolio selection paradigm with machine learning. *Annals of Operations Research, 346*, 319–340.
Qi, Y., & Steuer, R. E. (2025). An analytical derivation of properly efficient sets in multi-objective portfolio selection. *Annals of Operations Research, 346*, 1573–1595.
Roncalli, T. (2013). *Introduction to risk parity and budgeting*. Boca Raton: CRC Press.
Sen, J., & Dutta, A. (2022). A comparative study of hierarchical risk parity portfolio and eigen portfolio on the NIFTY 50 stocks. In: Computational Intelligence and Data Analytics: Proceedings of ICCIDA 2022. Springer, (p. 443–460).
Shkolnik, A., Kercheval, A., Gurdogan, H., Goldberg, L. R., & Bar, H. (2025). Portfolio selection revisited. *Annals of Operations Research, 346*, 137–155.
Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal, 16*(1), 30–34.