Improved iterative methods for solving risk parity portfolio

Jaehyuk Choi[a,*], Rong Chen[b]

[a] *Peking University HSBC Business School, Shenzhen, China*
[b] *Shenzhen Exchange, Shenzhen, China*

Abstract

Risk parity (or equal risk contribution) has recently gained increasing attention as a portfolio allocation method, but solving portfolio weights must resort to numerical methods as analytic solution is not available. This study improves two existing iterative methods: the cyclical coordinate descent (CCD) and Newton methods. We enhance the CCD method by formulating with the correlation matrix and imposing an additional rescaling step. We also suggest an improved initial guess inspired from the CCD method for the Newton method. Numerical experiments show that the improved CCD method performs the best. It is several times faster than the original CCD method, saving 40% of iteration steps.

Keywords: Risk parity, Equal risk contribution, cyclical coordinate descent, Newton method

1. Introduction

Optimal portfolio selection has been an important question in academia and financial industry alike. While there are several traditional methods, such as mean-variance, minimum variance, and equally weighted ($1/N$) portfolio, the risk parity (or equal risk contribution) model has recently gained popularity in asset management industry. Under the risk parity model, the portfolio weights are selected in such a way to equalize the contribution from each asset to the portfolio volatility become equal. Like $1/N$ portfolio, risk parity aims diversification, overcoming the concentration or sensitivity issues found in mean-variance or minimum variance portfolio. However, risk parity is the $1/N$ portfolio in terms of risk allocation rather than capital allocation.

While it is unclear who invented the risk parity idea for the first time, Maillard et al. (2010) and Qian (2011) are widely cited as one of the papers that introduce risk parity model. There has been growing literature on various aspects of the risk parity allocation method. For example, see Chaves et al. (2011) and Clarke et al. (2013) to see how risk parity compares to other asset allocation methods. Kim et al. (2020) study the risk parity model where the covariance is estimated from the XGBoost algorithm. Kim and Kim (2021) studies the risk parity under covariance estimation error. There are controversies over the actual performance of the funds implementing the strategy. See Grind (2013), for example.

*Corresponding author *Tel:* +86-755-2603-0568, *Address:* Rm 755, Peking University HSBC Business School, University Town, Nanshan, Shenzhen 518055, China

Email addresses: `jaehyuk@phbs.pku.edu.cn` (Jaehyuk Choi), `1801212829@pku.edu.cn` (Rong Chen)

Among the academic research topics around the risk parity model, this paper is primarily concerned with the numerical methods to solve the risk parity portfolio weights. Given the return covariance, it is challenging to solve the portfolio weight. As analytic solution is not available, one must resort to numerical methods. There has been several methods available so far. Maillard et al. (2010) formulate the risk parity problem as sequential quadratic programming (SQP) method. Chaves et al. (2012) uses the Newton method to solve the multidimensional root. This methods use the multidimensional root with the Newton method taking advantage of the analytical Jacobian matrix. While the original Newton method cannot guarantee the weights to be positive, Spinu (2013) later resolve the issue by adjusting the learning rate in the iteration steps. A competing method is the cyclical coordinate descent (CCD) algorithm (Griveau-Billion et al., 2013). As the CCD method uses the quadratic iteration steps, it does not rely on the Jacobian matrix. Bai et al. (2016) propose alternating linearization methods (ALMs) for solving the risk parity weight in a generalized setting.

The performance of the two methods seem comparable according to the literature. The CCD method Griveau-Billion et al. (2013) claim that the CCD method outperforms the Newton method when number of asset is lager than 250. Bouzida (2014) reports that. while CCD method is faster than the Newton method of Spinu (2013), the CCD method lacks the robustness for a pool of assets.

This study reviews and improves the two algorithms for solving the risk parity portfolio allocation: the CCD and Newton methods. We improve the CCD method into a numerically efficient form and the newton method by suggesting a new initial guess. Numerical experiment with randomly generated covariance matrix shows that our improved CCD method outperforms other methods, including the improved Newton method, for a wide range of portfolio sizes.

The remainder of this paper is organized as follows. Section 2 introduces the risk parity portfolio and its properties. Section 3 presents the improved root-finding method, and Section 4 demonstrates the computational gain of the new methods with numerical experiments. Finally, Section 5 concludes.

2. Risk parity portfolio

2.1. Notations and conventions

We define several notations and operations regarding vectors and matrices to be used for the rest of the paper.

- For vector $\boldsymbol{x}$ (in boldface), the i-th element is denoted by x_i or $(\boldsymbol{x})_j$.
- For matrix $\boldsymbol{A}$ (in boldface), the (i, j) element is denoted by A_{ij} or $(\boldsymbol{A})_{ij}$.
- Vectors are assume to be column vectors unless otherwise specified.
- $\mathbf{1}_N$ is the $N \times 1$ column vector filled with 1's. $\boldsymbol{I}_N$ is the $N \times N$ identity matrix.

- The operations, $*$ and $/$, between vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are defined to be the element-wise multiplication and division, respectively:

$$\boldsymbol{x} * \boldsymbol{y} = (x_1 y_1, \cdots, x_N y_N)^T \quad \text{and} \quad \boldsymbol{y}/\boldsymbol{x} = (y_1/x_1, \cdots, y_N/x_N)^T$$

- The operation $*$ between a (column) vector $\boldsymbol{x}$ and a matrix $\boldsymbol{A}$ is defined as

$$\boldsymbol{x} * \boldsymbol{A} = \boldsymbol{A} * \boldsymbol{x} = \begin{pmatrix} x_1 A_{11} & \cdots & x_1 A_{1N} \\ \vdots & \ddots & \vdots \\ x_N A_{N1} & \cdots & x_N A_{NN} \end{pmatrix}$$

2.2. Condition for risk parity portfolio

Let $\boldsymbol{\sigma}$ and $\boldsymbol{C}$ be the standard deviation vector and covariance matrix, respectively, of the return of N assets in a unit time period. The covariance $\boldsymbol{C}$ is symmetric and positive semi-definite, and its diagonal elements are related to $\boldsymbol{\sigma}$ via $C_{ii} = \sigma_i^2$. The portfolio of the N assets, invested with weight $\boldsymbol{w}$, has the return volatility:

$$V(\boldsymbol{w}) = \sqrt{\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w}}.$$

From Euler's homogeneous function theorem, the volatility $V(\boldsymbol{w})$ can be decomposed into the sum of the contribution from each asset,

$$V(\boldsymbol{w}) = \sum_i v_i(\boldsymbol{w}) \quad \text{where} \quad v_i(\boldsymbol{w}) = w_i \frac{\partial V(\boldsymbol{w})}{\partial w_i} = \frac{w_i (\boldsymbol{C}\boldsymbol{w})_i}{V(\boldsymbol{w})}.$$

Let b_i such that $\sum_i b_i = 1$ and $b_i > 0$ be the relative contribution to the portfolio volatility from the i-th asset. Then, we aim to find the weight $\boldsymbol{w}$ that satisfies

$$v_i(\boldsymbol{w}) = V(\boldsymbol{w})\, b_i \quad \text{for all } i.$$

The risk parity portfolio is the special case of the problem above where the risk contributions are equally divided among the assets,

$$b_i = 1/N \quad \text{for all } i.$$

Although we deal with the risk parity case exclusively, we will use b_i not to lose the generality. Therefore, the risk parity portfolio weight must satisfy the condition:

$$w_i (\boldsymbol{C}\boldsymbol{w})_i = V^2(\boldsymbol{w})\, b_i = (\boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w})\, b_i \quad \text{subject to} \quad w_i \geq 0. \tag{1}$$

Here, we impose $w_i \geq 0$ because we are concerned with the long-only portfolio. For the risk parity with unconstrained portfolio, we refer to Bai et al. (2016).

The solution is not unique because of the homogeneous property of the condition; if $\boldsymbol{w}$ is a solution, $\mu\boldsymbol{w}$ for $\mu > 0$ is also a solution. The degree of freedom can be fixed by imposing $\sum_i w_i = 1$. The normalized weight $\bar{\boldsymbol{w}}$ is obtained by

$$\bar{\boldsymbol{w}} = \frac{\boldsymbol{w}}{\lambda} \quad \text{for} \quad \lambda = \sum_i w_i.$$

2.3. Risk parity condition with correlation matrix

The condition for risk parity portfolio can equivalently be stated in terms of the correlation matrix (Spinu, 2013). Let $\boldsymbol{R}$ be the correlation matrix whose element is computed from the covariance matrix $\boldsymbol{C}$:

$$R_{ij} = \frac{C_{ij}}{\sigma_i \sigma_j} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} \quad (R_{ii} = 1)\,. \tag{2}$$

Then, the risk parity condition with the correlation matrix is given by

$$w_i(\boldsymbol{R}\boldsymbol{w})_i = (\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w})\, b_i \quad \text{subject to} \quad w_i \geq 0. \tag{3}$$

If $\boldsymbol{w}$ is the solution to the correlation condition, Eq. (3), $\boldsymbol{w}/\boldsymbol{\sigma}$ is the solution to the covariance condition, Eq. (1).

Moreover, Spinu (2013) pins down the degree of freedom in $\boldsymbol{w}$ by taking advantage of the fact that $\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}$ is a scalar although the value is unknown yet. Without loss of generality, we set

$$\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w} = 1, \tag{4}$$

and simply solve

$$w_i(\boldsymbol{R}\boldsymbol{w})_i = b_i \quad \text{subject to} \quad w_i \geq 0. \tag{5}$$

Once we find the unique root $\boldsymbol{w}$ of Eq. (5), the normalized portfolio weight $\bar{\boldsymbol{w}}$ can be obtained by

$$\bar{w}_i = \frac{w_i}{\lambda\sigma_i} \quad \text{for} \quad \lambda = \sum_i \frac{w_i}{\sigma_i}.$$

Note that Eq. (4) is consistent with Eq. (5) because it can be obtained by summing up Eq. (5) for all i,

$$\boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w} = \sum_i w_i(\boldsymbol{R}\boldsymbol{w})_i = \sum_i b_i = 1. \tag{6}$$

We will use both Eqs. (4) and (5) in Section 3.1 to improve the original CCD method.

2.4. A special case solution and initial guess for iterative methods

The general solution to the risk parity portfolio is not available analytically. An analytic solution, however, exists for a special condition on the correlation matrix. When the correlation matrix has the same row sums,

$$\textstyle\sum_j R_{ij} = r \quad \text{for all } i \text{ and some constant } r,$$

the constant vector, $\boldsymbol{w} = \mathbf{1}_N$, satisfies Eq. (3) with $b_i = 1/N$. Note that the special condition is typically satisfied when $\boldsymbol{R}$ has the same off-diagonal elements, $R_{jk} = \rho$ for $j \neq k$ and some ρ. Therefore, $\boldsymbol{w} = \mathbf{1}_N$ is the solution for the allocation among two assets. In terms of the simple condition, Eq. (5), $w_i = 1/\sqrt{nr}$ is the corresponding solution for the special case. The actual portfolio weights satisfying the original condition, Eq. (1), are given by the weights inversely proportional to the standard deviation,

$$\bar{w}_i = \frac{1/\sigma_i}{\sum_k 1/\sigma_k}. \tag{7}$$

Even though $\boldsymbol{R}$ does not satisfy the row sum condition, the special case solution serves as a first-order approximation. Chaves et al. (2012) and Griveau-Billion et al. (2013) uses Eq. (7) as the initial guess for the iterative methods. The Newton method of Spinu (2013), based on Eq. (5), uses an improved version for the initial guess. Based on $w_i = 1/\sqrt{nr}$, Spinu (2013, Theorem 3.4) generalizes the initial guess to

$$w_i = 1/\sqrt{\textstyle\sum_{j,k} R_{jk}} \quad \text{for all } i. \tag{8}$$

Note that $\sum_{j,k} R_{jk} = \mathbf{1}_N^T \boldsymbol{R} \mathbf{1}_N > 0$ if $\boldsymbol{R}$ is positive definite.

3. Methods for solving risk parity

In this section, we review the CCD and Newton methods, and improve them. As stated in Section 1, there exist other methods, such as, the SQP algorithm (Maillard et al., 2010) and ALM (Bai et al., 2016). Although such methods may be able to handle the risk parity under more generalized setting, it is reported that they are slower than the CCD and Newton methods in handling the standard long-only risk parity model. Therefore, we focus on the two methods.

3.1. The improved CCD method

The original CCD method (Griveau-Billion et al., 2013) aims to solve

$$w_i(\boldsymbol{C}\boldsymbol{w})_i = \sqrt{\boldsymbol{w}^T \boldsymbol{C}\boldsymbol{w}}\, b_i = V(\boldsymbol{w})\, b_i \quad \text{for all } i. \tag{9}$$

Here, note that this condition is yet different from Eq. (1) as $V^2(\boldsymbol{w})$ in Eq. (1) is replaced with $V(\boldsymbol{w})$. Although the intention is not explicitly stated in the reference, the purpose seems to fix the degree of freedom in the solution. The equation has two root: $\boldsymbol{w} = \mathbf{0}$ and the other one with $\boldsymbol{w} > 0$. Because of the homogeneous property, the nonzero root $\boldsymbol{w}$ is also a proper risk parity weight. In fact, the same holds for any nonzero scalar in the place of $V^2(\boldsymbol{w})$, in the same way we obtain Eq. (5).

The equation for the i-th component can be written as a quadratic form of w_i,

$$C_{ii} w_i^2 + \left(\sum_{j\neq i} C_{ij} w_j\right) w_i - V(\boldsymbol{w})\, b_i = 0,$$

and we use the root formula as an iteration step for w_i (Griveau-Billion et al., 2013, Eq. (4))

$$w_i \leftarrow \frac{\sqrt{a_i^2 + C_{ii} V(\boldsymbol{w})\, b_i} - a_i}{C_{ii}} \quad \text{for} \quad a_i = \frac{1}{2} \sum_{j\neq i} C_{ij} w_j = \frac{(\boldsymbol{C}\boldsymbol{w})_i - C_{ii} w_i}{2}. \tag{10}$$

Here, we select the positive one among the two roots of the quadratic equation to ensure that $w_i \geq 0$ (and to avoid the trivial root $\boldsymbol{w} = \boldsymbol{0}$). In CCD method, one iteration is composed of cyclically updating w_i for $i = 1, \cdots, N$. Therefore, updating w_i makes use of w_j for $1 \leq j < i$ which are previously updated in the same iteration step. Because of this, the CCD method is known to be more effective than the *batch* coordinate descent. Griveau-Billion et al. (2013) starts the iterations with the initial guess Eq. (7).

We improve the original CCD method in two ways. First, we formulate the CCD with the correlation condition, Eq. (5), instead. The equation for the i-th component of Eq. (5) can be written as a simpler quadratic form of w_i,

$$w_i^2 + \left(\sum_{j\neq i} R_{ij} w_j\right) w_i - b_i = 0.$$

The corresponding iteration step is simplified to

$$w_i \;\leftarrow\; \sqrt{a_i^2 + b_i} - a_i \quad \text{for} \quad a_i = \frac{1}{2} \sum_{j\neq i} R_{ij} w_j = \frac{(\boldsymbol{R}\boldsymbol{w})_i - w_i}{2}. \tag{11}$$

This new iteration brings two advantages because $V(\boldsymbol{w})$ disappears in the new CCD method. One obvious advantage is to save the computation time for $V(\boldsymbol{w})$. Because $V(\boldsymbol{w})$ requires $O(N^2)$ operations and it has to be updated when w_i is updated, the calculation can be time-consuming. The other advantage is not obvious, but more important. In the original CCD, the new w_i depends on the old value through $V(\boldsymbol{w})$ on the right-hand side of Eq. (10). The updated w_i is not the true root of Eq. (10). In the new CCD method, however, the new w_i is the exact root of Eq. (11) because old w_i does not appear on the right-hand side. Therefore, we expect that the convergence will be faster in the improved CCD iteration.

Second, we rescale $\boldsymbol{w}$ by

$$\boldsymbol{w} \;\leftarrow\; \frac{\boldsymbol{w}}{\sqrt{\boldsymbol{w}^T \boldsymbol{R} \boldsymbol{w}}}. \tag{12}$$

at the end of each iteration to ensure Eq. (4). This rescaling step is expected to make the convergence faster by adjusting $\boldsymbol{w}$ on average. The new CCD method will use the generalized initial guess, Eq. (8), borrowed from Spinu (2013). In fact, it can be understood as the result of the rescaling from the equal weight $\boldsymbol{w} = \boldsymbol{1}_N$.

Finally we summarize the improved CCD algorithm in Algorithm 1.

Algorithm 1 The improved CCD algorithm

Input: covariance matrix $\boldsymbol{C}$ and error tolerance ε
Calculate $\boldsymbol{R}$ and $\boldsymbol{\sigma}$
Initialize $\boldsymbol{w} \leftarrow \mathbf{1}_N / \sqrt{\sum_{ij} R_{ij}}$
while $\max_i |w_i(\boldsymbol{R}\boldsymbol{w})_i - b_i| > \varepsilon$ **do**
  for $i \leftarrow 1, \cdots, N$ **do**
    $w_i \leftarrow \sqrt{a_i^2 + b_i} - a_i \quad \text{for} \quad a_i = \frac{1}{2}\sum_{j\neq i} R_{ij} w_j$
  $\boldsymbol{w} \leftarrow \boldsymbol{w}/\sqrt{\boldsymbol{w}^T \boldsymbol{R}\boldsymbol{w}}$
return $(\boldsymbol{w}/\boldsymbol{\sigma})/(\sum_i w_i/\sigma_i)$

3.2. Newton method with an improved initial guess

Chaves et al. (2012) and Spinu (2013) use the multidimensional Newton method to find the risk parity portfolio. From Eq. (5), they set the objective function:

$$\boldsymbol{F}(\boldsymbol{w}) = \boldsymbol{R}\boldsymbol{w} - \frac{\boldsymbol{b}}{\boldsymbol{w}} \quad \text{or} \quad F_i(\boldsymbol{w}) = \sum_j R_{ij} w_j - \frac{b_i}{w_i}.$$

Then, the root of $\boldsymbol{F}(\boldsymbol{w}) = 0$ is the risk parity weight. The Jacobian of $\boldsymbol{F}(\boldsymbol{w})$ is readily available as

$$\nabla \boldsymbol{F}(\boldsymbol{w}) = \boldsymbol{R} + \boldsymbol{I}_N * \frac{\boldsymbol{b}}{\boldsymbol{w}^2} \quad \text{or} \quad \frac{\partial F_i(\boldsymbol{w})}{\partial w_j} = R_{ij} + \delta_{ij}\frac{b_i}{w_i^2},$$

where δ_{ij} is the Kronecker delta. Therefore, the iteration under the Newton method is given by

$$\boldsymbol{w} \;\leftarrow\; \boldsymbol{w} + \Delta\boldsymbol{w} \quad \text{for} \quad \Delta\boldsymbol{w} = -\left[\nabla \boldsymbol{F}(\boldsymbol{w})\right]^{-1} \boldsymbol{F}(\boldsymbol{w}) \tag{13}$$

Unlike the CCD method, however, the Newton method iteration cannot guarantee that the converged weights are positive. Spinu (2013) overcomes the problem with the dampened Newton method,

$$\boldsymbol{w} \;\leftarrow\; \boldsymbol{w} + \eta\Delta\boldsymbol{w}$$

where $\eta \leq 1$ is determined as a function of $\boldsymbol{w}$ and $\Delta\boldsymbol{w}$. While we do not discuss exact procedure, basic idea is to use $\eta < 1$ in early stage when $\boldsymbol{w}$ is away from the solution to ensure $w_i > 0$, and to use $\eta = 1$ later when $\boldsymbol{w}$ is close enough to the solution for faster convergence. Spinu (2013) use Eq. (8) for the initial guess of the Newton method.

Our enhancement on the Newton method is on the initial guess. By using an initial guess closer to the solution, we aim to use $\eta = 1$ all through the iteration yet without converging to negative weight. If this can be achieved, one can use generic Newton method routines available in many numerical analysis packages, which is highly optimized for the system.

We improve the original initial guess, Eq. (8), by updating it through the one-step CCD iteration, Eq. (11). Instead of slow cyclical update, however, we use the *batch* update where the old w_i values are used on the right-hand side:

$$w_i = \sqrt{a_i^2 + b_i} - a_i \quad \text{for} \quad a_i = \frac{\sum_{j\neq i} R_{ij}}{2\sqrt{\sum_{j,k} R_{jk}}}.$$

This new initial guess is more efficiently compured in the vetorized form,

$$\boldsymbol{w} = \sqrt{\boldsymbol{a} * \boldsymbol{a} + \boldsymbol{b}} \; - \; \boldsymbol{a} \quad \text{for} \quad \boldsymbol{a} = \frac{(\boldsymbol{R}\mathbf{1}_N - \mathbf{1}_N)}{2\sqrt{\mathbf{1}_N^T \boldsymbol{R}\mathbf{1}_N}}. \tag{14}$$

Numerical experiments in next Section demonstrates that our new initial guess is effective so that the method converges to a positive weight for almost all randomly generated test cases.

4. Numerical experiment

We test the numerical performance of the improved algorithms.[1] We implemented the following three algorithms for comparison.

- The original CCD method, Eq (10)
- The improved CCD method, Algorithm 1
- The Newton method, Eq (13), with the improved initial guess, Eq (14)

We implemented all methods in Python. For the Newton method, we use the generic root solver, `scipy.optimize.root` function[2] in Python SciPy package. For all methods, we consistently use the error tolerance $\varepsilon = 10^{-6}$.

In the test, we solve the risk parity portfolio for the correlation matrices randomly generated with `scipy.stats.random_correlation` class[3] in Python SciPy package. The routine takes the nonnegative eigenvalues as inputs. Then, it uses the algorithm of Davies and Higham (2000) to generate the correlation matrix. We use two methods of generating the eigenvalues to test both positive definite and positive semi-definite correlation matrix:

- **Test 1**: all eigenvalues sampled from independent uniform random variables between 0 and 1.
- **Test 2**: 80% of eigenvalues sampled from independent uniform random variables between 0 and 1, and 20% set to zeros.

---

[1] The test are performed in Python on a computer running the Windows 10 operating system with an Intel Core i5–6500 (3.2 GHz) CPU.

[2] See https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.html. We use `method='hybr'` (default) option.

[3] See https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.random_correlation.html.

While other literature (Griveau-Billion et al., 2013; Bai et al., 2016) typically test only the positive definite cases, we think the positive semi-definite case is also important because it is often encountered in practice when the covariance is estimated from time series. When the covariance between N assets is estimated from from M time periods with $M < N$, the estimated covariance matrix has rank at most M nonzero eigenvalues.

Figure 1 shows the computation time and number of iterations for **Test 1**. Several observations are in order. First, our improved CCD method performs the best in terms of both CPU time and iterations. Although the Newton method is marginally faster than the improved CCD method for $N \leq 150$, the required computation time is not long anyway for those N.

Second, our improved CCD method is three to four times faster than the original CCD, saving about 6.5 iterations on average. Although not reported in the figure, we also run the improved CCD method without the rescaling step, Eq. (12), to measure the gain from the rescaling step. It turns out that the rescaling step saves about 1.5 iterations on average.

Third, the Netwon method successfully converges to positive weights in all case cases. It confirms that the damped Newton method is not necessary with the improved initial guess, Eq. (14). Nevertheless, the Newton method is inferior to the improved CCD method. The log-log plot clearly shows that the computation time scales as $O(N^3)$ in the newton method, but scales as $O(N)$ for both the CCD methods. The $O(N^3)$ scale of the Newton method seems to be related to the computation for the Jacobian inversion, $[\nabla \boldsymbol{F}(\boldsymbol{w})]^{-1}$. In the optimized multidimensional Newton method, the inversion is not computed in every iteration. In our test, inversion is typically computed only once, i.e., at the initial condition. Nevertheless, the Jacobian inversion makes the Newton method slow for large N.

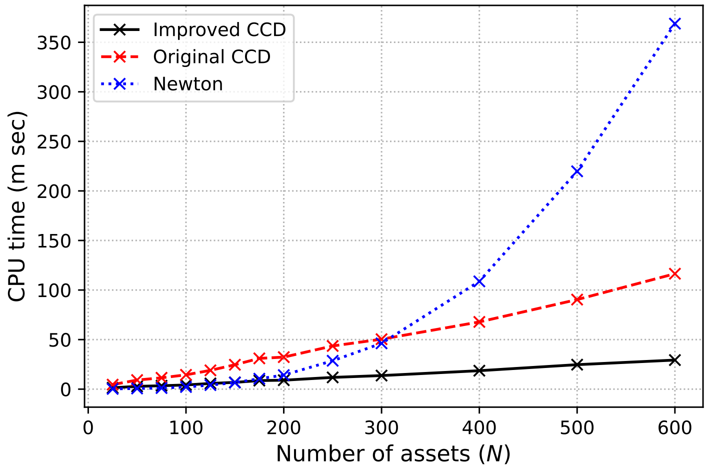
Figure 2 shows the results for **Test 2**. While the relative performance between the three methods are similar, the overall computation become slower than **Test 1**, requiring more iterations. This indicates difficulty of solving the risk parity portfolio against the positive semi-definite covariance matrices. Moreover, the Newton method shows instability. The method fails in convergence for 2 cases, and converges to negative weights for 13 cases. Conversely, the CCD methods stably converged to positive weights for cases.

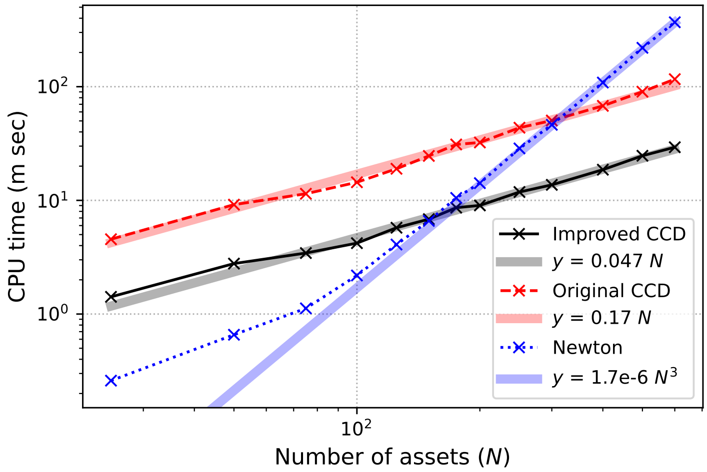
From the numerical tests, we believe that our improved CCD method is the fast and stable method for solving the risk parity weights.

5. Conclusion

Along the growing popularity of the risk parity model, several numerical methods has been proposed to solve the portfolio allocation. We present improvements on two existing methods based on iteration: the cyclical coordinate descent (CCD) and Newton methods. Numerical experiment shows that the improved CCD method performs the best in terms of speed and stability.

Figure 1: The computation time in linear(top) and log-log scales (middle), and the number of iterations (bottom) for randomly generated positive definite correlation matrices (**Test 1**). The values are averages over 200 tests for each N.





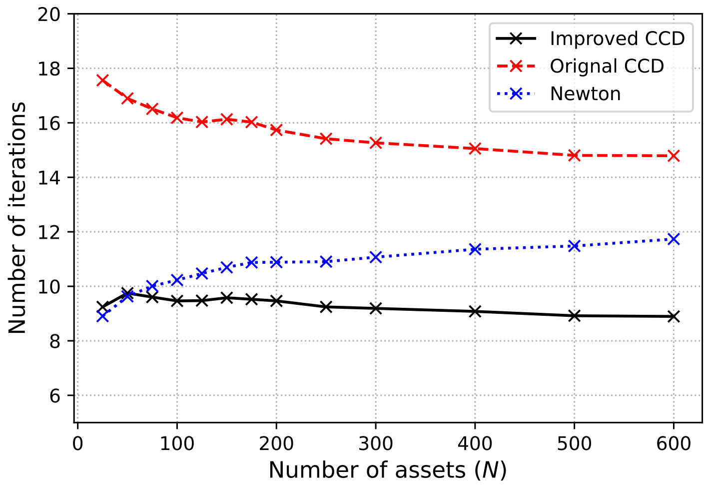
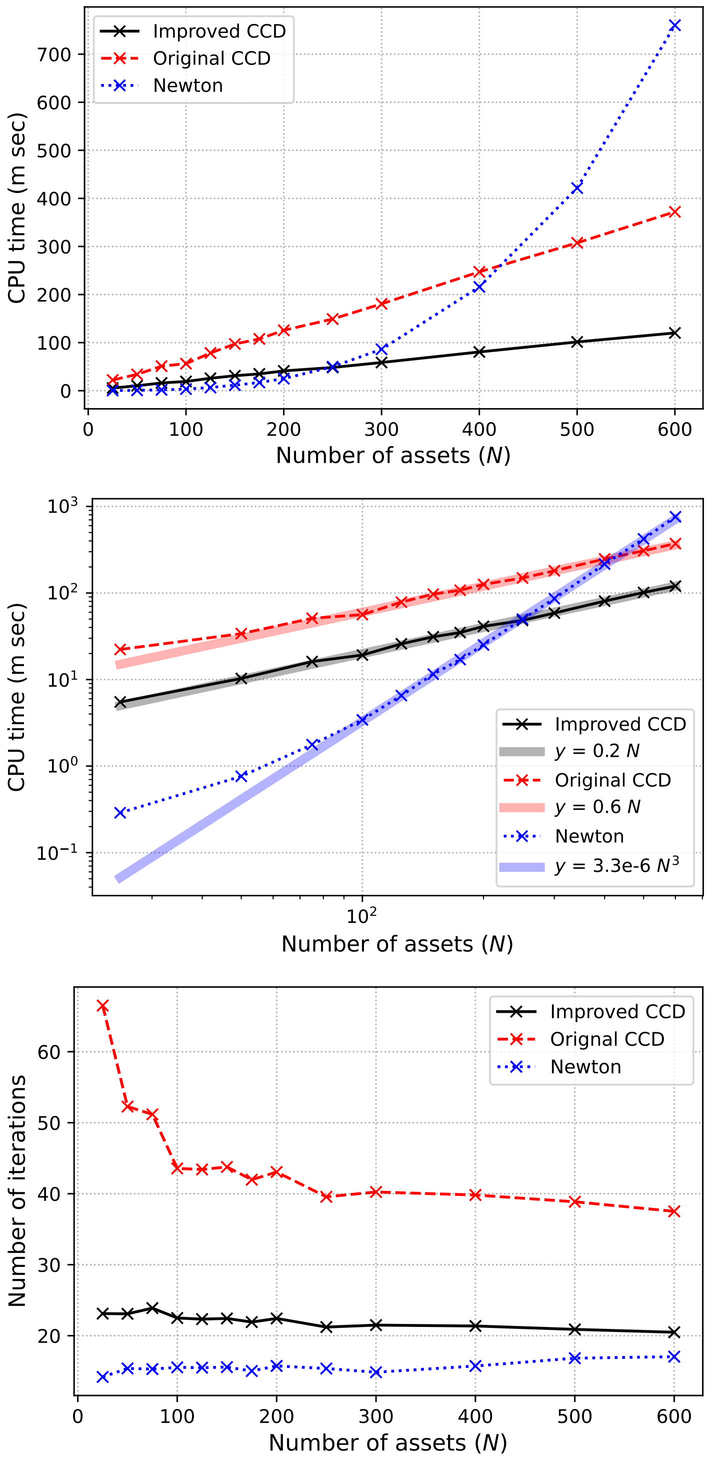


Figure 2: The computation time in linear (top) and log-log scales (middle), and the number of iterations (bottom) for randomly generated positive semi-definite correlation matrices (**Test 2**). The values are averages over 200 tests for each N.



Acknowledgements

This research was supported by the 2021 Korea Derivatives Association & FnGuide research grant.

References

Bai, X., Scheinberg, K., Tutuncu, R., 2016. Least-squares approach to risk parity in portfolio selection. Quantitative Finance 16, 357–376. doi:10.1080/14697688.2015.1031815.

Bouzida, F., 2014. Robust Risk Budgeting Algorithms in R. SSRN Scholarly Paper ID 3453218. Social Science Research Network. Rochester, NY. doi:10.2139/ssrn.3453218.

Chaves, D., Hsu, J., Li, F., Shakernia, O., 2011. Risk Parity Portfolio vs. Other Asset Allocation Heuristic Portfolios. The Journal of Investing 20, 108–118. doi:10.3905/joi.2011.20.1.108.

Chaves, D., Hsu, J., Li, F., Shakernia, O., 2012. Efficient Algorithms for Computing Risk Parity Portfolio Weights. The Journal of Investing 21, 150–163. doi:10.3905/joi.2012.21.3.150.

Clarke, R., de Silva, H., Thorley, S., 2013. Risk Parity, Maximum Diversification,and Minimum Variance: An Analytic Perspective. The Journal of Portfolio Management 39, 39–53. doi:10.3905/jpm.2013.39.3.039.

Davies, P.I., Higham, N.J., 2000. Numerically Stable Generation of Correlation Matrices and Their Factors. BIT Numerical Mathematics 40, 640–651. doi:10.1023/A:1022384216930.

Grind, Carolyn Cui and Kirsten, M.C., 2013. Fashionable 'Risk Parity' Funds Hit Hard. Wall Street Journal .

Griveau-Billion, T., Richard, J.C., Roncalli, T., 2013. A Fast Algorithm for Computing High-dimensional Risk Parity Portfolios. arXiv:1311.4057 [q-fin] arXiv:1311.4057.

Kim, H., Kim, S., 2021. Reduction of estimation error impact in the risk parity strategies. Quantitative Finance 21, 1351–1364. doi:10.1080/14697688.2021.1881599.

Kim, Y., Choi, H., Kim, S., 2020. A Study on Risk Parity Asset Allocation Model with XGBoos. Journal of Intelligence and Information Systems 26, 135–149. doi:10.13088/jiis.2020.26.1.135.

Maillard, S., Roncalli, T., Teïletche, J., 2010. The Properties of Equally Weighted Risk Contribution Portfolios. The Journal of Portfolio Management 36, 60–70. doi:10.3905/jpm.2010.36.4.060.

Qian, E., 2011. Risk Parity and Diversification. The Journal of Investing 20, 119–127. doi:10.3905/joi.2011.20.1.119.

Spinu, F., 2013. An Algorithm for Computing Risk Parity Weights. SSRN Scholarly Paper ID 2297383. Social Science Research Network. Rochester, NY. doi:10.2139/ssrn.2297383.