Constrained Risk Budgeting Portfolios Theory, Algorithms, Applications & Puzzles[*]

Jean-Charles Richard
Quantitative Research
Eisler Capital
jcharles.richard@gmail.com

Thierry Roncalli
Quantitative Research
Amundi Asset Management, Paris
thierry.roncalli@amundi.com

January 2019

Abstract

This article develops the theory of risk budgeting portfolios, when we would like to impose weight constraints. It appears that the mathematical problem is more complex than the traditional risk budgeting problem. The formulation of the optimization program is particularly critical in order to determine the right risk budgeting portfolio. We also show that numerical solutions can be found using methods that are used in large-scale machine learning problems. Indeed, we develop an algorithm that mixes the method of cyclical coordinate descent (CCD), alternating direction method of multipliers (ADMM), proximal operators and Dykstra's algorithm. This theoretical body is then applied to some investment problems. In particular, we show how to dynamically control the turnover of a risk parity portfolio and how to build smart beta portfolios based on the ERC approach by improving the liquidity of the portfolio or reducing the small cap bias. Finally, we highlight the importance of the homogeneity property of risk measures and discuss the related scaling puzzle.

Keywords: Risk budgeting, large-scale optimization, Lagrange function, cyclical coordinate descent (CCD), alternating direction method of multipliers (ADMM), proximal operator, Dykstra's algorithm, turnover, liquidity, risk parity, smart beta portfolio.

JEL classification: C61, G11.

1 Introduction

Since the 2008 Global Financial Crisis, the development of risk budgeting (RB) techniques has marked an important milestone in portfolio management by putting diversification at the center of portfolio construction (Qian, 2005; Maillard *et al.*, 2010). In particular, the equal risk contribution (ERC) portfolio has been very popular and has significantly impacted the asset management industry. For instance, this allocation approach is extensively implemented in risk parity funds, factor investing and alternative risk premia (Roncalli, 2017).

The main advantages of RB portfolios are the stability of the allocation, and the diversification management principle, which appear more robust than the diversification mechanism of mean-variance optimized portfolios (Bourgeron *et al.*, 2018). This is why we don't need

[*]We would like to thank Thibault Bourgeron, Joan Gonzalvez, Edmond Lezmi, Jean-Tristan Marin, Sarah Perrin, Roman Rubsamen and Jiali Xu for their helpful comments.

to add constraints in order to regularize the solution. This advantage is also its drawback. Indeed, there are some situations where portfolio managers have to impose constraints. For example, they may want to impose a minimum investment weight, a sector-neutrality or some liquidity thresholds. The goal of this paper is then to define what does a constrained risk budgeting portfolio mean, since adding constraints will change the risk budgets that are targeted, meaning that ex-post risk contributions are not equal to ex-ante risk budgets.

This paper is organized as follows. Section Two illustrates the bridge between risk budgeting and portfolio optimization. In Section Three, we present the right mathematical formulation of constrained risk budgeting portfolios, and develop the corresponding numerical algorithms. In Section Four, we consider some applications in asset allocation, in particular the management of turnover or the consideration of liquidity. Finally, we discuss the compatibility puzzle of the homogeneity property of coherent risk measures.

2 The original risk budgeting portfolio

2.1 Definition of the risk budgeting portfolio

We consider a universe of n risky assets. Let μ and Σ be the vector of expected returns and the covariance matrix of asset returns. We have $\Sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j$ where σ_i is the volatility of asset i and $\rho_{i,j}$ is the correlation between asset i and asset j. Following Roncalli (2015), we consider the standard deviation-based risk measure defined as follows:

$$\mathcal{R}(x) = -x^\top(\mu - r) + c \cdot \sqrt{x^\top \Sigma x} \tag{1}$$

where c is a scalar that measures the trade-off between the expected return of the portfolio and its volatility. We deduce that the risk contribution of Asset i is given by:

$$\mathcal{RC}_i(x) = x_i \cdot \left(-(\mu_i - r) + c \frac{(\Sigma x)_i}{\sqrt{x^\top \Sigma x}} \right)$$

Following Maillard *et al.* (2010), Roncalli (2013) defines the risk budgeting (RB) portfolio using the following non-linear system:

$$\left\{ \begin{array}{l} \mathcal{RC}_i(x) = b_i \mathcal{R}(x) \\ b_i > 0 \\ x_i \geq 0 \\ \sum_{i=1}^n b_i = 1 \\ \sum_{i=1}^n x_i = 1 \end{array} \right. \tag{2}$$

where b_i is the risk budget of Asset i expressed in relative terms. The constraint $b_i > 0$ implies that we cannot set some risk budgets to zero. This restriction is necessary in order to ensure that the RB portfolio is unique.

Remark 1 *Roncalli (2015) shows that the existence of the RB portfolio depends on the value taken by c. In particular, the RB portfolio exists and is unique if $c > \mathrm{SR}^+$ where SR^+ is the maximum Sharpe ratio of the asset universe:*

$$\mathrm{SR}^+ = \max\left(\sup_{x \in [0,1]^n} \mathrm{SR}(x \mid r), 0 \right)$$

Remark 2 *The original ERC portfolio is obtained by considering the volatility risk measure and the same risk budgets. It is equivalent to seting $\mu_i = r$, $c = 1$ and $b_i = 1/n$. In this case, we have:*

$$\mathcal{RC}_i(x) = \frac{x_i \cdot (\Sigma x)_i}{\sqrt{x^\top \Sigma x}} = \frac{1}{n}\sqrt{x^\top \Sigma x}$$

2.2 The associated optimization problem

2.2.1 The wrong formulation

System (2) is equivalent to solving n non-linear equations with n unknown variables. Therefore, we can use the Newton or Broyden methods to find the numerical solution. We also deduce that:

$$\frac{1}{b_i}\mathcal{RC}_i(x) = \frac{1}{b_j}\mathcal{RC}_j(x) \qquad \text{for all } i, j$$

In order to find the solution, an alternative approach is to solve the optimization problem:

$$\begin{aligned} x_{\text{RB}} &= \arg\min \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{1}{b_i}\mathcal{RC}_i(x) - \frac{1}{b_j}\mathcal{RC}_j(x)\right)^2 \\ & \text{s.t.} \quad \begin{cases} \mathbf{1}^\top x = 1 \\ x \geq \mathbf{0} \end{cases} \end{aligned} \tag{3}$$

At the optimum x_{RB}, the objective function $f(x)$ must be equal to zero. This approach was originally proposed by Maillard *et al.* (2010) in the case $b_i = b_j$.

At first sight, Problem (3) seems to be easy to solve because it resembles how a quadratic functions in n variables is defined and we can analytically compute the gradient vector and the hessian matrix of the objective function. In fact, it is not a convex problem (Feng and Palomar, 2015), and the optimization is tricky when the number of assets is large. From a theoretical point of view, the objective function is not well defined because the solution is only valid if the zero can be reached: $f(x_{\text{RB}}) = 0$. But the most important issue is that the optimization problem is driven by the equality constraint: $\mathbf{1}^\top x = 1$. Indeed, if we remove it, the solution is $x_{\text{RB}} = \mathbf{0}$.

2.2.2 The right formulation

Roncalli (2013) shows that the RB portfolio is the solution of the following optimization problem:

$$\begin{aligned} x_{\text{RB}} &= \arg\min \mathcal{R}(x) \\ & \text{s.t.} \quad \begin{cases} \sum_{i=1}^{n} b_i \ln x_i \geq \kappa^\star \\ \mathbf{1}^\top x = 1 \\ x \geq \mathbf{0} \end{cases} \end{aligned} \tag{4}$$

where $\kappa^\star$ is a constant to be determined. This optimization program is equivalent to finding the optimal solution $x^\star(\kappa)$:

$$\begin{aligned} x^\star(\kappa) &= \arg\min \mathcal{R}(x) \\ & \text{s.t.} \quad \begin{cases} \sum_{i=1}^{n} b_i \ln x_i \geq \kappa \\ x \geq \mathbf{0} \end{cases} \end{aligned} \tag{5}$$

where κ is an arbitrary constant and to scale the solution:

$$x_{\text{RB}} = \frac{x^\star(\kappa)}{\mathbf{1}^\top x^\star(\kappa)}$$

Using the Lagrange formulation, we obtain an equivalent solution:

$$\begin{aligned} x^\star(\lambda) &= \arg\min \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i \\ & \text{s.t.} \quad x \geq \mathbf{0} \end{aligned} \tag{6}$$

where λ is an arbitrary positive scalar and:

$$x_{\mathrm{RB}} = \frac{x^{\star}(\lambda)}{\mathbf{1}^{\top} x^{\star}(\lambda)}$$

$x^{\star}(\lambda)$ is the solution of a standard logarithmic barrier problem, which has very appealing characteristics. First, it defines a unique solution. Second, the constraint $\mathbf{1}^{\top} x = 1$ is removed, meaning that the optimization exploits the scaling property. Finally, the constraint $x \geq \mathbf{0}$ is redundant since the logarithm is defined for strictly positive numbers.

We claim that Problem (6) is the right risk budgeting problem. For instance, Maillard *et al.* (2010) used this formulation to show that the ERC portfolio exists and is unique. Roncalli (2013) also noticed that there is a discontinuity when one or more risk budgets b_i are equal to zero. In this case, we can find several solutions that satisfy $\mathcal{RC}_i(x) = b_i \mathcal{R}(x)$ or Problem (3), but only one solution if we consider the logarithmic barrier program.

2.3 Numerical solution

2.3.1 The Newton algorithm

Spinu (2013) proposes solving Problem (6) by using the Newton algorithm[1]:

$$x^{(k+1)} = x^{(k)} - \eta^{(k)} \left(\frac{\partial^2 f\left(x^{(k)}\right)}{\partial x \, \partial x^{\top}} \right)^{-1} \frac{\partial f\left(x^{(k)}\right)}{\partial x}$$

where $\eta^{(k)} \in [0,1]$ is the step size and k is the iteration index. Generally, we set $\eta^{(k)} = 1$. Spinu (2013) noticed that the Newton algorithm may be improved because the risk measure is self-concordant. In this case, we can use the results of Nesterov (2004) to determine the optimal size $\eta^{(k)}$ at each iteration.

2.3.2 The CCD algorithm

The descent algorithm is defined by the following rule:

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \Delta x^{(k)} \\ &= x^{(k)} - \eta D^{(k)} \end{aligned}$$

At the k^{th} Iteration, the current solution $x^{(k)}$ is updated by going in the opposite direction to $D^{(k)}$. For instance, $D^{(k)}$ is equal respectively to $\partial_x f\left(x^{(k)}\right)$ in the gradient algorithm and $\left(\partial^2_{x,x} f\left(x^{(k)}\right)\right)^{-1} \partial_x f\left(x^{(k)}\right)$ in the Newton algorithm. Coordinate descent is a modification of the descent algorithm by minimizing the function along one coordinate at each step:

$$\begin{aligned} x_i^{(k+1)} &= x_i^{(k)} + \Delta x_i^{(k)} \\ &= x_i^{(k)} - \eta D_i^{(k)} \end{aligned}$$

[1]The first and second derivatives are computed using the following analytical expressions:

$$\begin{aligned} \frac{\partial f(x)}{\partial x_i} &= -(\mu_i - r) + c \frac{(\Sigma x)_i}{\sqrt{x^{\top} \Sigma x}} - \lambda \frac{b_i}{x_i} \\ \frac{\partial^2 f(x)}{\partial x_i \, \partial x_j} &= \frac{c}{\sqrt{x^{\top} \Sigma x}} \left(\rho_{i,j} \sigma_i \sigma_j - \frac{(\Sigma x)_i (\Sigma x)_j}{x^{\top} \Sigma x} \right) \\ \frac{\partial^2 f(x)}{\partial x_i^2} &= \frac{c}{\sqrt{x^{\top} \Sigma x}} \left(\sigma_i^2 - \frac{(\Sigma x)_i^2}{x^{\top} \Sigma x} \right) + \lambda \frac{b_i}{x_i^2} \end{aligned}$$

The coordinate descent algorithm becomes a scalar problem, and we know that minimizing a function with respect to one variable is easier than with n variables. Concerning the choice of the variable i, there are two approaches: random coordinate descent or RCD (Nesterov, 2012) and cyclical coordinate descent or CCD (Tseng, 2001). In the first case, we assign a random number between 1 and n to the index i. In the second case, we cyclically iterate through the coordinates:

$$x_i^{(k+1)} = \arg\min_x f\left(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$$

This ensures that all the indices are selected during one cycle. In the CCD algorithm, k is the cycle index while i is the iteration index within a cycle.

Griveau-Billion *et al.* (2013) propose applying the CCD algorithm to find the solution of the objective function:

$$f(x) = -x^\top \pi + c\sqrt{x^\top \Sigma x} - \lambda \sum_{i=1}^n b_i \ln x_i$$

where $\pi = \mu - r$. The first-order condition is:

$$\frac{\partial\, \mathcal{L}(x;\lambda)}{\partial\, x_i} = -\pi_i + c\frac{(\Sigma x)_i}{\sigma(x)} - \lambda \frac{b_i}{x_i}$$

At the optimum, we have $\partial_{x_i} \mathcal{L}(x;\lambda) = 0$ or:

$$c\sigma_i^2 x_i^2 + \left(c\sigma_i \sum_{j\neq i} x_j \rho_{i,j}\sigma_j - \pi_i \sigma(x)\right) x_i - \lambda b_i \sigma(x) = 0$$

By definition of the RB portfolio we have $x_i > 0$. We notice that the polynomial function is convex because we have $\sigma_i^2 > 0$. Since the product of the roots is negative, we always have two solutions with opposite signs. It can be deduced that the solution is the positive root of the second-degree equation. For the cycle $k+1$ and the i^{th} coordinate, we have:

$$x_i = \frac{-c\left(\sigma_i \sum_{j\neq i} x_j\rho_{i,j}\sigma_j\right) + \pi_i\sigma(x) + \sqrt{\left(c\left(\sigma_i \sum_{j\neq i} x_j\rho_{i,j}\sigma_j\right) - \pi_i\sigma(x)\right)^2 + 4\lambda c b_i \sigma_i^2 \sigma(x)}}{2c\sigma_i^2}$$

In this equation, we have the following correspondence: $x_i \to x_i^{(k+1)}$, $x_j \to x_j^{(k+1)}$ if $j < i$, $x_j \to x_j^{(k)}$ if $j > i$, and $x \to \left(x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}, x_i^{(k)}, x_{i+1}^{(k)}, \ldots, x_n^{(k)}\right)$. If the values of $(x_1, \ldots, x_n)$ are strictly positive and if $c > \mathrm{SR}^+$, $x_i^{(k+1)}$ should be strictly positive. The positivity of the solution is then achieved after each iteration and each cycle if the starting values are positive. Therefore, the coordinate-wise descent algorithm consists in iterating the previous equation, and we can show that it always converges (Roncalli, 2015).

Remark 3 *As noted by Griveau-Billion et al. (2013), the previous algorithm can be simplified by setting λ equal to* 1 *and by rescaling the solution once the convergence is obtained. Our experience shows that it is better to rescale the solution once the CCD algorithm has converged rather than after each cycle. Moreover, Griveau-Billion et al. (2013) derive analytical formulas in order to update $\sigma(x)$ and $\sum_{j\neq i} x_j\rho_{i,j}\sigma_j$ at each iteration.*

3 Theory of constrained risk budgeting portfolio

3.1 Mathematical issues

If we consider the definition of Roncalli (2013), introducing constraints leads to the following formulation of the constrained risk budgeting portfolio[2]:

$$\begin{cases} \mathcal{RC}_i(x) = b_i \mathcal{R}(x) \\ x \in \mathcal{S} \\ x \in \Omega \end{cases}$$

where $\mathcal{S}$ is the standard simplex:

$$\mathcal{S} = \left\{ x_i \geq 0 : \sum_{i=1}^{n} x_i = 1 \right\}$$

and $x \in \Omega$ is the set of additional constraints. Let $x^\star(\mathcal{S})$ be the risk budgeting portfolio, i.e. the solution such that $x \in \mathcal{S}$, and $x^\star(\mathcal{S}, \Omega)$ be the constrained risk budgeting portfolio, i.e. the solution such that $x \in \mathcal{S}$ and $x \in \Omega$. Since $x^\star(\mathcal{S})$ is unique, we deduce that the solution $x^\star(\mathcal{S}, \Omega)$ exists only if $x^\star(\mathcal{S}) \in \Omega$, and we have $x^\star(\mathcal{S}, \Omega) = x^\star(\mathcal{S})$. Since we generally have $x^\star(\mathcal{S}) \notin \Omega$, we deduce that there is almost certainly no solution.

This is why professionals generally replace the equality constraint by an approximate equality:

$$\begin{cases} \mathcal{RC}_i(x) \approx b_i \mathcal{R}(x) \\ x \in \mathcal{S} \\ x \in \Omega \end{cases}$$

Therefore, the optimization problem becomes:

$$\begin{aligned} x^\star(\mathcal{S}, \Omega) &= \arg\min \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{1}{b_i} \mathcal{RC}_i(x) - \frac{1}{b_j} \mathcal{RC}_j(x) \right)^2 \\ \text{s.t.} \quad & x \in \mathcal{S} \cap \Omega \end{aligned} \tag{7}$$

Bai *et al.* (2016) show that this optimization problem can be simplified as follows:

$$\begin{aligned} \{x^\star(\mathcal{S}, \Omega), \theta^\star\} &= \arg\min \sum_{i=1}^{n} \left(\frac{1}{b_i} \mathcal{RC}_i(x) - \theta \right)^2 \\ \text{s.t.} \quad & x \in \mathcal{S} \cap \Omega \end{aligned} \tag{8}$$

Example 1 *We consider a universe of four assets. Their volatilities are equal to* 10%, 15%, 20% *and* 30%. *The correlation matrix of asset returns is given by the following matrix:*

$$\rho = \begin{pmatrix} 1.00 & & & \\ 0.50 & 1.00 & & \\ 0.50 & 0.50 & 1.00 & \\ 0.50 & 0.50 & 0.75 & 1.00 \end{pmatrix}$$

By using the volatility risk measure, we compute the ERC portfolio and the RB portfolio corresponding to the risk budgets $(30\%, 30\%, 19.5\%, 20.5\%)$. In Table 1, we report the solution of ERC and RB portfolios. We also indicate the marginal risk $\mathcal{MR}_i$, the absolute

[2]We assume that $b_i > 0$ and $\sum_{i=1}^{n} b_i = 1$.

Table 1: Computation of ERC and RB portfolios

Asset	ERC portfolio				RB portfolio			
	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	41.01	7.79	3.19	25.00	45.05	8.06	3.63	30.00
2	27.34	11.68	3.19	25.00	30.04	12.09	3.63	30.00
3	18.99	16.82	3.19	25.00	14.67	16.10	2.36	19.50
4	12.66	25.23	3.19	25.00	10.24	24.23	2.48	20.50
$\sigma(x)$			12.78				12.11	

Table 2: Computation of ERC and RB portfolios when $x_i \leq 30\%$

Asset	ERC portfolio				RB portfolio			
	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	30.00	7.19	2.16	15.50	30.00	7.19	2.16	15.48
2	30.00	11.60	3.48	24.98	30.00	11.60	3.48	24.96
3	24.57	17.43	4.28	30.74	24.43	17.42	4.26	30.52
4	15.43	25.98	4.01	28.78	15.57	26.01	4.05	29.04
$\sigma(x)$			13.93				13.94	

risk contribution $\mathcal{RC}_i$ and the relative risk contribution $\mathcal{RC}_i^\star$. Let us now introduce the constraint $x_i \leq 30\%$ and solve the optimization problem (8). Since this constraint is not satisfied by the previous unconstrained portfolio, it has an impact as shown in Table 2. As expected, the relative risk contributions (or ex-post risk budgets) are completely different from the ex-ante risk budgets. The concept of "equal risk contribution" does not make any sense. Moreover, we observe that the ordering relationship between risk budgets are not preserved when we introduce constraints. For example, in the case of the RB portfolio, we have $b_3 < b_4$ (19.50% versus 20.50%) but $\mathcal{RC}_3^\star > \mathcal{RC}_4^\star$ (30.52% versus 29.04%) although the allocation in Assets 3 and 4 does not reach the upper bound constraint. We also notice that the constrained ERC portfolio is very close to the constrained RB portfolio. This gives us the feeling that the choice of risk budgets has little impact, and the solution is mainly driven by the constraints.

3.2 Formulation of the optimization problem

Like for the unconstrained risk budgeting portfolio, we argue that the right optimization problem is:

$$\begin{aligned} x^\star(\mathcal{S},\Omega) &= \arg\min \mathcal{R}(x) \\ & \text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^n b_i \ln x_i \geq \kappa^\star \\ x \in \mathcal{S} \cap \Omega \end{array} \right. \end{aligned} \tag{9}$$

where $\kappa^\star$ is a constant to be determined. We notice that the previous problem can be simplified because:

1. the logarithmic barrier constraint imposes that $x_i \geq 0$;
2. Roncalli (2013) shows that there is only one constant $\kappa^\star$ such that the constraint $\mathbf{1}^\top x = 1$ is satisfied.

We can then consider the following optimization problem:

$$\begin{aligned} x^\star(\Omega,\kappa) &= \arg\min \mathcal{R}(x) \\ &\text{s.t.} \quad \left\{ \begin{array}{l} \sum_{i=1}^n b_i \ln x_i \geq \kappa \\ x \in \Omega \end{array} \right. \end{aligned} \tag{10}$$

and we have:

$$x^\star(\mathcal{S},\Omega) = \left\{ x^\star(\Omega,\kappa^\star) : \sum_{i=1}^n x_i^\star(\Omega,\kappa^\star) = 1 \right\}$$

This new formulation is appealing since the constraint $x \in \mathcal{S}$ is not explicit, but it is implicitly embedded in the optimization problem. From a computational point of view, this reduces the complexity of the numerical algorithm. When the constraint $x \in \Omega$ vanishes, we retrieve the previous scaling rule. Otherwise, we consider the Lagrange formulation:

$$\begin{aligned} x^\star(\Omega,\lambda) &= \arg\min \mathcal{R}(x) - \lambda \sum_{i=1}^n b_i \ln x_i \\ &\text{s.t.} \quad x \in \Omega \end{aligned} \tag{11}$$

Again, we have:

$$x^\star(\mathcal{S},\Omega) = \left\{ x^\star(\Omega,\lambda^\star) : \sum_{i=1}^n x_i^\star(\Omega,\lambda^\star) = 1 \right\}$$

Formulations (9), (10) and (11) have the advantage of revealing the true nature of risk budgeting. The objective is to minimize the risk measure subject to a penalization (Richard and Roncalli, 2015). A risk budgeting portfolio is then a minimum risk portfolio subject to hard risk budgeting, constraints, whereas a constrained risk budgeting portfolio is a minimum risk portfolio subject to soft risk budgeting constraints. Because of the convexity of the optimization problem (10), it follows that:

$$\kappa_2 \geq \kappa_1 \Rightarrow \mathcal{R}(x^\star(\Omega,\kappa_2)) \geq \mathcal{R}(x^\star(\Omega,\kappa_1)) \tag{12}$$

for a given set of constraints Ω. This property is fundamental since it is the essence of risk budgeting, and it explains the relationships between long-only minimum variance, risk budgeting and equally-weighted portfolios obtained by Maillard *et al.* (2010) and the relationships between long-only minimum risk, risk budgeting and weight-budgeting portfolios obtained by Roncalli (2013). This property is also necessary to impose the continuity of risk budgeting portfolios in particular when some risk budgets tend to zero. Without this property, it is impossible to show that the RB portfolio is unique, and to determine the true solution in the case where there are several solutions to Problem (2) when $b_i = 0$.

Remark 4 *Imposing tighter constraints does not necessarily increase the risk measure*[3]*:*

$$\Omega_2 \subset \Omega_1 \nRightarrow \mathcal{R}(x^\star(\mathcal{S},\Omega_2)) \geq \mathcal{R}(x^\star(\mathcal{S},\Omega_1))$$

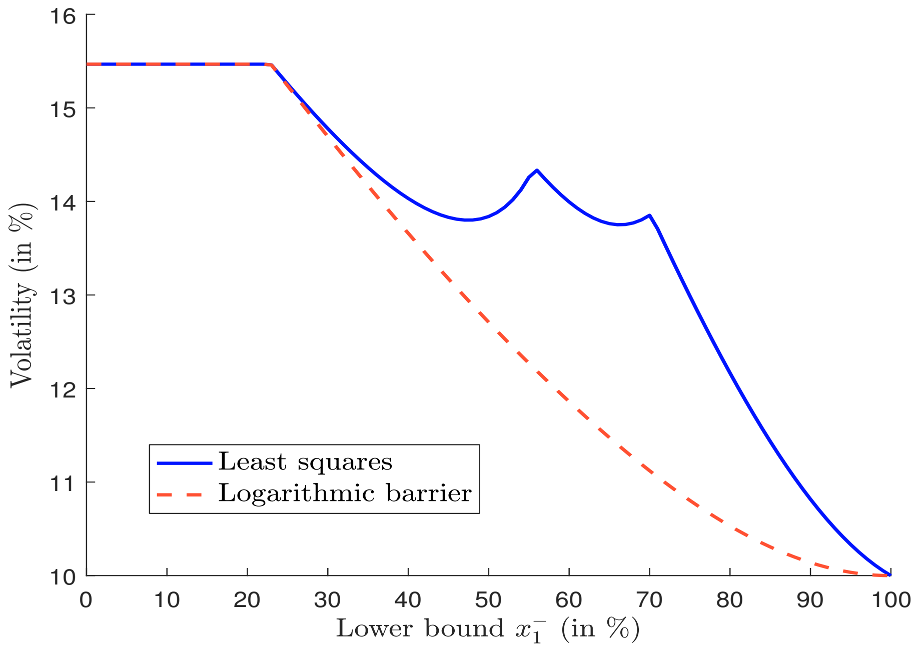
In order to illustrate the risk measure issue, we consider Example 1 with the risk budgets $b = (10\%, 20\%, 30\%, 40\%)$. Moreover, we impose that the weight x_1 of Asset 1 is greater than a given lower bound x_1^-. In Figure 1, we have reported the portfolio volatility $\sigma(x^\star(\mathcal{S},\Omega))$ of the constrained risk budgeting with respect to x_1^-. If we consider the least squares problem (8), the volatility is non-monotonous. This is not the case if we consider the logarithmic barrier problem (11). Indeed, the least squares formulation only considers the dimension of risk contribution matching, but not the dimension of risk measure minimization.

[3]Indeed, we have:

$$\Omega_2 \subset \Omega_1 \Rightarrow \mathcal{R}(x^\star(\Omega_2,\kappa)) \geq \mathcal{R}(x^\star(\Omega_1,\kappa))$$

for a given value of κ. However, the value $\kappa^\star$ to obtain the optimal portfolio $x^\star(\mathcal{S},\Omega)$ depends on Ω.

Figure 1: Volatility of the constrained RB portfolio when $x_1 \leq x_1^-$



3.3 Numerical algorithms

The optimization function becomes:

$$\mathcal{L}(x;\lambda) = \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i + \mathbb{1}_{\Omega}(x) \tag{13}$$

where $\mathbb{1}_{\Omega}(x)$ is the convex indicator function of Ω, meaning that $\mathbb{1}_{\Omega}(x) = 0$ for $x \in \Omega$ and $\mathbf{1}_{\Omega}(x) = +\infty$ for $x \notin \Omega$. The choice of λ is very important, since the constrained RB portfolio is obtained for the optimal value $\lambda^{\star}$ such that the sum of weights is equal to one. This can be done using the Newton-Raphson or the bisection algorithm. If we note $x^{\star}(\lambda)$ the solution of the minimization problem (13), we obtain Algorithm 1 in the case of the bisection method.

3.3.1 ADMM algorithm

In order to solve Problem (13), we exploit the separability of $\mathcal{L}(x;\lambda)$. For example, we can write:

$$\mathcal{L}(x;\lambda) = \underbrace{\mathcal{R}(x) - \lambda \sum\nolimits_{i=1}^{n} b_i \ln x_i}_{f(x)} + \underbrace{\mathbb{1}_{\Omega}(x)}_{g(x)} \tag{14}$$

or:

$$\mathcal{L}(x;\lambda) = \underbrace{\mathcal{R}(x) + \mathbb{1}_{\Omega}(x)}_{f(x)} + \underbrace{-\lambda \sum\nolimits_{i=1}^{n} b_i \ln x_i}_{g(x)} \tag{15}$$

Algorithm 1 General algorithm for computing the constrained RB portfolio

```
The goal is to compute the optimal Lagrange multiplier $\lambda^{\star}$ and the solution $x^{\star}(\mathcal{S},\Omega)$
We consider two scalars $a_{\lambda}$ and $b_{\lambda}$ such that $a_{\lambda} < b_{\lambda}$ and $\lambda^{\star} \in [a_{\lambda}, b_{\lambda}]$
We note $\varepsilon_{\lambda}$ the convergence criterion of the bisection algorithm (e.g. $10^{-8}$)
repeat
    We calculate $\lambda = \frac{a_{\lambda}+b_{\lambda}}{2}$
    We compute $x^{\star}(\lambda)$ the solution of the minimization problem:
        $x^{\star}(\lambda) = \arg\min \mathcal{L}(x;\lambda)$
    if $\sum_{i=1}^{n} x_i^{\star}(\lambda) < 1$ then
        $a_{\lambda} \leftarrow \lambda$
    else
        $b_{\lambda} \leftarrow \lambda$
    end if
until $\left| \sum_{i=1}^{n} x_i^{\star}(\lambda) - 1 \right| \leq \varepsilon_{\lambda}$
return $\lambda^{\star} \leftarrow \lambda$ and $x^{\star}(\mathcal{S},\Omega) \leftarrow x^{\star}(\lambda^{\star})$
```

We notice that we have:

$$\begin{aligned} \{x^{\star}(\lambda), z^{\star}(\lambda)\} &= \arg\min f(x) + g(z) \\ \text{s.t.} \quad & x - z = 0 \end{aligned} \tag{16}$$

It follows that we can use the alternative direction method of multipliers (ADMM) to solve this optimization problem. Algorithm 2 describes the different steps.

In the case of the Lagrange function (14), the x-update is equivalent to solving a penalized risk budgeting problem whereas the z-update corresponds to a proximal operator. Therefore, we can use the Newton algorithm[4] to find $x^{(k)}$. For the z-update, we have:

$$z^{(k)} = \mathbf{prox}_{g/\varphi}(v) = \arg\min_z \left\{ g(z) + \frac{\varphi}{2} \left\| z - v_z^{(k)} \right\|_2^2 \right\}$$

where $v_z^{(k)} = x^{(k)} + u^{(k-1)}$. If we assume that $g(z) = \mathbb{1}_{\Omega}(z)$ where Ω is a convex set, we obtain:

$$\begin{aligned} z^{(k)} &= \arg\min_z \left\{ \mathbb{1}_{\Omega}(z) + \frac{\varphi}{2} \left\| z - v_z^{(k)} \right\|_2^2 \right\} \\ &= \mathcal{P}_{\Omega}\left(v_z^{(k)}\right) \end{aligned}$$

[4]The first and second derivatives of $f^{(k)}(x) = f(x) + \frac{\varphi}{2} \left\| x - z^{(k-1)} + u^{(k-1)} \right\|_2^2$ are equal to:

$$\begin{aligned} \frac{\partial f^{(k)}(x)}{\partial x_i} &= \frac{\partial f(x)}{\partial x_i} + \varphi\left(x_i - z^{(k-1)} + u^{(k-1)}\right) \\ \frac{\partial^2 f^{(k)}(x)}{\partial x_i \partial x_j} &= \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \\ \frac{\partial^2 f^{(k)}(x)}{\partial x_i^2} &= \frac{\partial^2 f(x)}{\partial x_i^2} + \varphi \end{aligned}$$

where the derivatives of $f(x)$ are given in Footnote 1 on page 4.

Algorithm 2 ADMM algorithm for computing the portfolio $x^{\star}(\lambda)$

```
The goal is to compute the solution $x^{\star}(\Omega, \lambda)$ for a given value of $\lambda$
We initialize $x^{(0)}$ and we choose $0 \leq \varphi \leq 1$
We set $z^{(0)} = x^{(0)}$ and $u^{(0)} = \mathbf{0}$
We note $\varepsilon$ the convergence criterion of the ADMM algorithm (e.g. $10^{-8}$)
repeat
    $$\begin{array}{rcl} x^{(k)} & = & \arg\min \left\{ f(x) + \frac{\varphi}{2} \left\| x - z^{(k-1)} + u^{(k-1)} \right\|_2^2 \right\} \\ z^{(k)} & = & \arg\min \left\{ g(z) + \frac{\varphi}{2} \left\| x^{(k)} - z + u^{(k-1)} \right\|_2^2 \right\} \\ u^{(k)} & = & u^{(k-1)} + \left( x^{(k)} - z^{(k)} \right) \end{array}$$
until $\left\| x^{(k)} - z^{(k)} \right\| \leq \varepsilon$
return $x^{\star}(\lambda) \leftarrow x^{(k)}$
```

where $\mathcal{P}_{\Omega}(v)$ is the standard projection. In Appendix A.2 on page 29, we give the results for the generic constraints that we encounter in portfolio optimization. We develop some special cases in the next section.

For the Lagrange function (15), the x-update corresponds to a constrained risk minimization problem whereas the z-update is equivalent to solving a penalized logarithmic barrier problem. The x-update can be done using constrained non-linear optimization methods[5], whereas the z-step corresponds to the proximal operator of the logarithmic barrier function[6]:

$$z_i^{(k)} = \frac{\varphi \left(x_i^{(k)} + u_i^{(k-1)} \right) + \sqrt{\varphi^2 \left(x_i^{(k)} + u_i^{(k-1)} \right)^2 + 4\varphi\lambda b_i}}{2\varphi}$$

If we consider the volatility risk measure $\mathcal{R}(x) = \sqrt{x^{\top}\Sigma x}$ instead of the standard deviation-based risk measure given by Equation (1), the ADMM algorithm is simplified as follows[7]:

$$\begin{array}{rcl} x^{(k)} & = & \arg\min \frac{1}{2} x^{\top} \left(\Sigma + \varphi I_n \right) x - \varphi x^{\top} \left(z^{(k-1)} - u^{(k-1)} \right) \\ & \text{s.t.} & x \in \Omega \\ z^{(k)} & = & \dfrac{\varphi \left(x_i^{(k)} + u_i^{(k-1)} \right) + \sqrt{\varphi^2 \left(x_i^{(k)} + u_i^{(k-1)} \right)^2 + 4\varphi\lambda b}}{2\varphi} \\ u^{(k)} & = & u^{(k-1)} + \left(x^{(k)} - z^{(k)} \right) \end{array}$$

[5]In order to accelerate the convergence, we can implement analytical derivatives, which are the same than those given in Footnote 4 on page 10 by setting $\lambda = 0$ in the derivatives of the function $f(x)$.

[6]See Appendix A.4 on page 33.

[7]We have:

$$\begin{array}{rcl} f^{(k)}(x) & = & \frac{1}{2} x^{\top} \Sigma x + \frac{\varphi}{2} \left(x - z^{(k-1)} + u^{(k-1)} \right)^{\top} \left(x - z^{(k-1)} + u^{(k-1)} \right) \\ & = & \frac{1}{2} x^{\top} \Sigma x + \frac{\varphi}{2} \left(x^{\top} x - 2 x^{\top} v_x^{(k)} + \left(v_x^{(k)} \right)^{\top} v_x^{(k)} \right) \end{array}$$

where $v_x^{(k)} = z^{(k-1)} - u^{(k-1)}$.

This algorithm exploits the property that minimizing the portfolio volatility is equivalent to minimizing the portfolio variance, even if this last risk measure does not satisfy the Euler decomposition. If Ω is a set of linear (equality and inequality) constraints, the x-update reduces to a standard QP problem.

3.3.2 CCD algorithm

Another route for solving Problem (14) is to consider the CCD algorithm. Convergence of coordinate descent methods requires that the function is strictly convex and differentiable. However, Tseng (2001) has extended the convergence properties to a non-differentiable class of functions:

$$f(x) = f_0(x) + \sum_{i=1}^{n} f_i(x_i)$$

where f_0 is strictly convex and differentiable and the functions f_i are non-differentiable. Dealing with convex constraints is equivalent to writing the constraints in the sum term of $f(x)$. If we consider the formulation (14) and the standard deviation-based risk measure, we have:

$$\mathcal{L}(x;\lambda) = \mathcal{L}_0(x;\lambda) + \mathbf{1}_{\Omega}(x)$$

where $\mathcal{L}_0(x;\lambda)$ is defined as follows:

$$\mathcal{L}_0(x;\lambda) = -x^\top \pi + c\sqrt{x^\top \Sigma x} - \lambda \sum_{i=1}^{n} b_i \ln x_i$$

The case of separable constraints If we assume that the set of constraints is separable with respect to all the variables:

$$\Omega = \bigcap_{i=1}^{n} \Omega_i$$

where Ω_i is the constraint on x_i, we have $f_i(x_i) = \mathbb{1}_{\Omega_i}(x_i)$. We deduce that the CCD algorithm consists in two steps. We first solve the minimization problem of $\mathcal{L}_0(x;\lambda)$ for one coordinate, and then we compute the projection onto Ω_i. For the first step, the first-order condition is:

$$\frac{\partial\,\mathcal{L}_0(x;\lambda)}{\partial\,x_i} = -\pi_i + c\frac{(\Sigma x)_i}{\sqrt{x^\top \Sigma x}} - \lambda \frac{b_i}{x_i} = 0$$

It follows that $cx_i(\Sigma x)_i - \pi_i x_i \sigma(x) - \lambda b_i \sigma(x) = 0$ or equivalently:

$$\alpha_i x_i^2 + \beta_i x_i + \gamma_i = 0$$

where:

$$\begin{cases} \alpha_i = c\sigma_i^2 \\ \beta_i = c\sigma_i \sum_{j\neq i} x_j \rho_{i,j}\sigma_j - \pi_i\sigma(x) \\ \gamma_i = -\lambda b_i \sigma(x) \end{cases}$$

We deduce that the coordinate solution is the positive root of the second-degree equation:

$$x_i = \frac{-\beta_i + \sqrt{\beta_i^2 - 4\alpha_i\gamma_i}}{2\alpha_i}$$

The second step is the projection into the set Ω_i:

$$x_i = \mathcal{P}_{\Omega_i}(x_i)$$

Finally, we obtain Algorithm 3.

Algorithm 3 CCD algorithm for computing the portfolio $x^{\star}(\lambda)$ when the set of constraints is separable with respect to the variables x_i

```
The goal is to compute the solution $x^{\star}(\Omega, \lambda)$ for a given value of $\lambda$
We initialize the vector $x$
We note $\varepsilon$ the convergence criterion of the CCD algorithm (e.g. $10^{-8}$)
repeat
  $x' \leftarrow x$
  for $i = 1 : n$ do
    $\sigma_x \leftarrow \sigma(x)$
    We update $x_i$ as follows:
        $$x_i \leftarrow \frac{-\beta_i + \sqrt{\beta_i^2 - 4\alpha_i\gamma_i}}{2\alpha_i}$$
    where:
        $$\left\{\begin{array}{l} \alpha_i = c\sigma_i^2 \\ \beta_i = c\sigma_i \sum_{j \neq i} x_j \rho_{i,j} \sigma_j - \pi_i \sigma_x \\ \gamma_i = -\lambda b_i \sigma_x \end{array}\right.$$
    $x_i \leftarrow \mathcal{P}_{\Omega_i}(x_i)$
  end for
until $\sum_{i=1}^{n} (x'_i - x_i)^2 \leq \varepsilon$
return $x^{\star}(\lambda) \leftarrow x^{(k)}$
```

Remark 5 *Our algorithm differs from the one given by Nesterov (2012) and Wright (2015). Let $\eta > 0$ be the stepsize of the gradient descent. The coordinate update is:*

$$x_i^{\star} = \arg\min (x - x_i) g_i + \frac{1}{2\eta} (x - x_i)^2 + \xi \cdot \mathbb{1}_{\Omega_i}(x)$$

where ξ is a positive scalar and:

$$g_i = -\pi_i + c\frac{(\Sigma x)_i}{\sqrt{x^{\top}\Sigma x}} - \lambda \frac{b_i}{x_i}$$

In our case, this algorithm is very simple[8]*, because it reduces to calculate the proximal of $x_i - \eta g_i$ associated with the function $\mathbb{1}_{\Omega_i}(x)$. However, we prefer to use the previous algorithm in order to exploit the analyticity of the solution.*

The case of non-separable constraints A first idea is to replace the projection step $x_i \leftarrow \mathcal{P}_{\Omega_i}(x_i)$ by something equivalent that ensures that the constraints are verified. The natural approach is to apply the proximal operator or equivalently the projection: $x \leftarrow \mathcal{P}_{\Omega}(x)$. In practice, we observe that the CCD solution does not always converge to the true solution. It will depend on how the constraints and the variables are ordered. In fact, the true approach is to use a block-coordinate algorithm if constraints are separable by blocks. This is not always the case. This is why we prefer to use the previous ADMM algorithms or the ADMM-CCD algorithm, which is described in the next paragraph.

3.3.3 Mixed ADMM-CCD algorithm

If we consider the formulation (14), we have already shown that the x-update of the ADMM algorithm corresponds to a regularized risk budgeting problem. Therefore, we can use the

[8]See Appendix A.5 on page 34.

Algorithm 4 ADMM-CCD algorithm for computing the portfolio $x^{\star}(\lambda)$

```
The goal is to compute the solution $x^{\star}(\Omega, \lambda)$ for a given value of $\lambda$
We initialize $x^{(0)}$ and we choose $0 \leq \varphi \leq 1$
We set $z^{(0)} = x^{(0)}$ and $u^{(0)} = \mathbf{0}$
We note $\varepsilon$ and $\varepsilon'$ the convergence criterion of ADMM and CCD algorithms
We note $k_{\max}$ the maximum number of ADMM iterations
for $k = 1 : k_{\max}$ do
                                I. $x$-update
  $v_x^{(k)} \leftarrow z^{(k-1)} - u^{(k-1)}$
  $\tilde{x} \leftarrow x^{(k-1)}$
  repeat
    $\tilde{x}' \leftarrow \tilde{x}$
    for $i = 1 : n$ do
      We update the volatility $\sigma_x \leftarrow \sigma(\tilde{x})$ and calculate:
        $$\left\{\begin{array}{l} \alpha_i = c\sigma_i^2 + \varphi\sigma_x \\ \beta_i = c\sigma_i \sum_{j \neq i} \tilde{x}_j \rho_{i,j} \sigma_j - \left(\pi_i + \varphi v_{x_i}^{(k)}\right) \sigma_x \\ \gamma_i = -\lambda b_i \sigma_x \end{array}\right.$$
      We update $\tilde{x}_i$ as follows:
        $$\tilde{x}_i \leftarrow \frac{-\beta_i + \sqrt{\beta_i^2 - 4\alpha_i\gamma_i}}{2\alpha_i}$$
    end for
  until $\sum_{i=1}^{n} (\tilde{x}_i' - \tilde{x}_i)^2 \leq \varepsilon'$
  $x^{(k)} \leftarrow \tilde{x}$
                                II. $z$-update
  $v_z^{(k)} \leftarrow x^{(k)} + u^{(k-1)}$
  $z^{(k)} \leftarrow \mathcal{P}_{\Omega}\left(v_z^{(k)}\right)$
                                III. $u$-update
  $u^{(k)} \leftarrow u^{(k-1)} + x^{(k)} - z^{(k)}$
                                IV. Convergence test
  if $\left\| x^{(k)} - z^{(k)} \right\| \leq \varepsilon$ then
    Break
  end if
end for
return $x^{\star}(\lambda) \leftarrow x^{(k)}$
```

CCD algorithm to find the solution $x^{(k)}$. We remind that:

$$f^{(k)}\left(x\right) = -x^{\top}\pi + c\sqrt{x^{\top}\Sigma x} - \lambda \sum_{i=1}^{n} b_i \ln x_i + \frac{\varphi}{2}\left\| x - v_x^{(k)} \right\|_2^2$$

where $v_x^{(k)} = z^{(k-1)} - u^{(k-1)}$. The first-order condition is:

$$\frac{\partial\, f^{(k)}\left(x\right)}{\partial\, x_i} = -\pi_i + c\frac{\left(\Sigma x\right)_i}{\sqrt{x^{\top}\Sigma x}} - \lambda\frac{b_i}{x_i} + \varphi\left(x_i - v_{x_i}^{(k)}\right) = 0$$

It follows that:

$$cx_i\left(\Sigma x\right)_i - \pi_i x_i \sigma\left(x\right) - \lambda b_i \sigma\left(x\right) + \varphi x_i^2 \sigma\left(x\right) - \varphi x_i v_{x_i}^{(k)} \sigma\left(x\right) = 0$$

or:

$$\alpha_i x_i^2 + \beta_i x_i + \gamma_i = 0$$

where:

$$\left\{\begin{array}{l} \alpha_i = c\sigma_i^2 + \varphi\sigma\left(x\right) \\ \beta_i = c\sigma_i \sum_{j\neq i} x_j \rho_{i,j}\sigma_j - \left(\pi_i + \varphi v_{x_i}^{(k)}\right)\sigma\left(x\right) \\ \gamma_i = -\lambda b_i \sigma\left(x\right) \end{array}\right.$$

We deduce that the coordinate solution is the positive root of the previous second-degree equation:

$$x_i = \frac{-\beta_i + \sqrt{\beta_i^2 - 4\alpha_i\gamma_i}}{2\alpha_i}$$

Finally, we obtain the ADMM-CCD algorithm, which is described on the previous page.

3.3.4 Efficiency of the algorithms

We may investigate the efficiency of the previous algorithms. Firstly, our experience shows that traditional constrained optimization algorithms (SQP, trust region, constrained interior point, etc.) fail to find the solution. It is somewhat surprising because we have the feeling that the optimization problem of constrained risk budgeting portfolios seems to be standard. Unfortunately, this is not the case, because the mixing of the logarithmic barrier and constraints is not usual. This is why it is important to implement the previous algorithms. Secondly, all the algorithms are not equal and the implementation is key in particular when we consider large problems with more than one hundred assets. In Table 3, we have reported the computational time we have obtained for solving the example described on page 18. For that, we consider five different methods: ADMM-Newton, ADMM-BFGS, ADMM-QP, ADMM-CCD and CCD. For each method, we consider three implementations:

1. The first one considers that the primal variable φ is constant ($\varphi = 1$) and we use the classical bisection method described in Algorithm 1.

2. The second one considers that the penalization variable φ is constant ($\varphi = 1$), and we use an accelerated bisection method. The underlying idea is to choose a starting value $x^{(0)}$ for the ADMM/CCD algorithm, which is not constant and depends on the Lagrange coefficient λ. The corresponding method is described in Appendix A.6 on page 35.

3. The third uses the accelerated bisection algorithm, and considers the adaptive method for the variable $\varphi^{(k)}$, which is given in Appendix A.7 on page 35. The underlying idea is to accelerate the convergence of the ADMM algorithm by using the right scale of the primal residual variable $u^{(k)}$.

Results using our Matlab implementation are reported in Table 3. Absolute figures are not interesting, because they depend on the processing power of the computer. Relative figures show that computational times can differ dramatically from one algorithm to another, from one implementation to another. The best algorithms are ADMM-Newton and CCD, followed by ADMM-CCD. Curiously, the ADMM-QP method is less efficient[9]. Finally, the worst algorithm is the ADMM-BFGS algorithm. However, we notice a large improvement in this algorithm if we implement the accelerated bisection and the adaptive method for scaling the regularization parameter. Indeed, the computational time is divided by a factor of 15!

Table 3: Computational time using our Matlab implementation (relative value)

Algorithm	x-update	(1)	(2)	(3)
ADMM	Newton	2	1	1
ADMM	BFGS	380	280	25
ADMM	QP	220	120	110
ADMM	CCD	10	9	8
CCD		1	1	

3.4 Special cases

3.4.1 Box constrained optimization

In portfolio optimization, imposing lower and upper bounds is frequent:

$$\Omega = \left\{x \in \mathbb{R}^n : x^- \leq x \leq x^+\right\}$$

For example, box constraints are used when limiting single exposures because of regulatory constraints or controlling the turnover of the portfolio.

The optimization framework In the box constrained case, the Lagrange function becomes:

$$\begin{aligned}\mathcal{L}\left(x;\lambda,\lambda^-,\lambda^+\right) &= -x^\top\pi + c\sqrt{x^\top\Sigma x} - \lambda\sum_{i=1}^n b_i \ln x_i - \\ & \sum_{i=1}^n \lambda_i^-\left(x_i - x_i^-\right) - \sum_{i=1}^n \lambda_i^+\left(x_i^+ - x_i\right)\end{aligned} \tag{17}$$

The first-order condition is:

$$\frac{\partial\,\mathcal{L}\left(x;\lambda,\lambda^-,\lambda^+\right)}{\partial\,x_i} = -\pi_i + c\frac{\left(\Sigma x\right)_i}{\sqrt{x^\top\Sigma x}} - \lambda\frac{b_i}{x_i} - \lambda_i^- + \lambda_i^+ = 0$$

We deduce that:

$$\mathcal{RC}_i\left(x\right) = \lambda b_i + \lambda_i^- x_i - \lambda_i^+ x_i$$

Since the Kuhn-Tucker conditions are:

$$\left\{\begin{array}{l}\min\left(\lambda_i^-, x_i - x_i^-\right) = 0\\ \min\left(\lambda_i^+, x_i^+ - x_i\right) = 0\end{array}\right.$$

we obtain three cases:

[9] This is due to the `quadprog` procedure of Matlab. Indeed, we do not observe same results in Python.

1. If no bound is reached, we have $\lambda_i^- = 0$ and $\lambda_i^+ = 0$, and we retrieve the RB portfolio x_{RB};
2. If the lower bound is reached, $\lambda_i^- > 0$ and the risk contribution of Asset i is higher than b_i;
3. If the upper bound is reached, $\lambda_i^+ > 0$ and the risk contribution of Asset i is lower than b_i.

Remark 6 *When we compare the logarithmic barrier solution with the least squares solution of the ERC portfolio, we observe that the first one preserves the "equal risk contribution" property for all the assets that do not reach lower or upper bounds. This is not the case with the least squares solution, since the risk contribution is different for all the assets.*

The previous algorithms require the proximal operator to be computed:

$$\begin{aligned} x^\star &= \mathbf{prox}_g(\tilde{x}) \\ &= \mathcal{P}_\Omega(\tilde{x}) \end{aligned}$$

where $\tilde{x} = x^{(k)} + u^{(k-1)}$ is the value of v_z^k in the ADMM procedure. In Appendix A.2 on page 29, we show that the proximal operator corresponds to the truncation operator:

$$\mathbf{prox}_g(\tilde{x}) = \mathcal{T}\left(\tilde{x}; x^-, x^+\right)$$

where:

$$\mathcal{T}\left(\tilde{x}; x^-, x^+\right) = \begin{cases} x_i^- & \text{if } \tilde{x}_i < x_i^- \\ \tilde{x}_i & \text{if } x_i^- \leq \tilde{x}_i \leq x_i^+ \\ x_i^+ & \text{if } \tilde{x}_i > x_i^+ \end{cases}$$

For the CCD algorithm, the projection $x_i \leftarrow \mathcal{P}_{\Omega_i}(x_i)$ reduces to apply the truncation operator to the single coordinate x_i.

At the optimum, we deduce that:

$$\lambda_i^{-\star} = \max\left(\frac{\mathcal{RC}_i\left(x^\star(\mathcal{S},\Omega)\right) - \lambda^\star b_i}{x_i^\star(\mathcal{S},\Omega)}, 0\right)$$

and:

$$\lambda_i^{+\star} = \max\left(\frac{\lambda^\star b_i - \mathcal{RC}_i\left(x^\star(\mathcal{S},\Omega)\right)}{x_i^\star(\mathcal{S},\Omega)}, 0\right)$$

where $\lambda^\star$ is the solution of the bisection algorithm.

Remark 7 *In order to find the optimal value $\lambda^\star$, we need an initial guess λ_0 for the Newton-Raphson algorithm or an interval $[a_\lambda, b_\lambda]$ for the bisection method. Let x_{RB} be the RB portfolio without constraints. The optimal Lagrange coefficient associated with x_{RB} is equal to*[10]*:*

$$\lambda_{\mathrm{RB}} = \sum_{i=1}^n \mathcal{RC}_i(x_{\mathrm{RB}}) = \mathcal{R}(x_{\mathrm{RB}})$$

[10]Indeed, the first-order condition is:

$$\begin{aligned} \frac{\partial\,\mathcal{R}(x)}{\partial\,x_i} - \lambda\frac{b_i}{x_i} = 0 \quad &\Leftrightarrow \quad x_i\frac{\partial\,\mathcal{R}(x)}{\partial\,x_i} - \lambda b_i = 0 \\ &\Leftrightarrow \quad \sum_{i=1}^n x_i\frac{\partial\,\mathcal{R}(x)}{\partial\,x_i} - \lambda\sum_{i=1}^n b_i = 0 \\ &\Leftrightarrow \quad \lambda = \sum_{i=1}^n x_i\frac{\partial\,\mathcal{R}(x)}{\partial\,x_i} = \mathcal{R}(x) \end{aligned}$$

because $\sum_{i=1}^n b_i = 1$.

It follows that a good initial guess is $\lambda_0 = \mathcal{R}(x_{\mathrm{RB}})$. *We also notice that:*

$$\begin{aligned}
\lambda^\star &= \sum_{i=1}^n \mathcal{RC}_i\left(x^\star(S,\Omega)\right) + \sum_{i=1}^n \left(\lambda_i^{+\star} - \lambda_i^{-\star}\right) x_i^\star(S,\Omega) \\
&= \mathcal{R}\left(x^\star(S,\Omega)\right) + \sum_{i=1}^n \left(\lambda_i^{+\star} - \lambda_i^{-\star}\right) x_i^\star(S,\Omega)
\end{aligned}$$

We deduce that $a_\lambda = m_a \cdot \mathcal{R}(x_{\mathrm{RB}})$ *and* $b_\lambda = m_b \cdot \mathcal{R}(x_{\mathrm{RB}})$ *where the parameters* m_a *and* m_b *depends on the tightness of constraints*[11]. *Generally, we have* $\mathcal{R}(x^\star(S,\Omega)) \approx \mathcal{R}(x_{\mathrm{RB}})$, *implying that the values* $m_a = 0.5$ *and* $m_b = 2.0$ *are sufficient.*

An example of dynamic allocation We consider a universe of five assets. Their volatilities are equal to 15%, 20%, 25%, 30% and 10%. The correlation matrix of asset returns is given by the following matrix:

$$\rho = \begin{pmatrix} 1.00 & & & & \\ 0.10 & 1.00 & & & \\ 0.40 & 0.70 & 1.00 & & \\ 0.50 & 0.40 & 0.80 & 1.00 & \\ 0.50 & 0.40 & 0.05 & 0.10 & 1.00 \end{pmatrix}$$

Let us assume that the current portfolio is $x_0 = (25\%, 25\%, 10\%, 15\%, 30\%)$. In Table 4, we report the volatility breakdown of this portfolio. We notice that there are some large differences in terms of risk contributions. In particular, the second asset has a volatility contribution of 31.1%. We would like to obtain a more balanced portfolio. Table 5 shows the results of the ERC portfolio.

Table 4: Volatility breakdown (in %) of the current portfolio

Asset	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	25.00	10.00	2.50	20.21
2	25.00	15.40	3.85	31.10
3	10.00	20.30	2.03	16.41
4	10.00	22.24	2.22	17.98
5	30.00	5.90	1.77	14.30
$\sigma(x)$			12.37	

We notice that the ERC portfolio is relatively far from the current portfolio. In particular, the turnover is equal to 22.18%. In order to obtain a solution closer to the current allocation, we impose that the weights cannot deviate from the current ones by 5%:

$$x_0 - 5\% \leq x \leq x_0 + 5\%$$

The underlying idea is to move from the initial portfolio to a risk budgeting portfolio, which presents the risk parity property as much as possible. In this case, we obtain the results in Table 6. We notice that three assets (#1, #3 and #4) present the same contributions (2.35%) because they do not reach the bounds. On the contrary, the second and fifth assets have respectively a higher and lower risk contribution (2.98% and 2.10%) because the lower and upper bounds are reached. This solution helps to reduce the turnover, since it is equal to 14.22%.

[11] We have $m_a \leq 1$ and $m_b \geq 1$.

Table 5: Volatility breakdown (in %) of the ERC portfolio

Asset	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	22.40	10.61	2.38	20.00
2	16.51	14.39	2.38	20.00
3	12.03	19.74	2.38	20.00
4	10.51	22.60	2.38	20.00
5	38.54	6.16	2.38	20.00
$\sigma(x)$			11.88	

Table 6: Volatility breakdown (in %) of the constrained RB portfolio

Asset	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	λ_i^-	λ_i^+
1	22.89	10.28	2.35	19.39	0.00	0.00
2	20.00	14.90	2.98	24.55	3.13	0.00
3	11.69	20.13	2.35	19.39	0.00	0.00
4	10.42	22.57	2.35	19.39	0.00	0.00
5	35.00	6.00	2.10	17.29	0.00	0.73
$\sigma(x)$			12.14		$\lambda = 11.76$	

A naive solution to obtain a risk parity portfolio that matches the constraints would be to identify the assets that reach the lower and upper bounds and to allocate the remaining weight between the other assets by imposing the same risk contribution. In this example, the second and fifth assets do not satisfy the constraints. Therefore, we have to allocate 45% of the allocation between the first, third and fourth assets. This naive solution is given in Table 7. The ERC portfolio between the three unconstrained assets is equal to $(22.84\%, 12.34\%, 9.83\%)$. In this case, the risk contributions are the same and are equal to 2.65%. However, the equal risk contribution property does not hold if we consider the full portfolio, when we take into account the constrained assets. Indeed, we obtain $\mathcal{RC}_1 = 2.34\%$, $\mathcal{RC}_3 = 2.49\%$ and $\mathcal{RC}_4 = 2.21\%$. The reason is that risk budgeting portfolios are sensitive to the asset universe definition (Roncalli and Weisang, 2016). This is why this two-step naive procedure does not give the right answer. In Table 7, we also report the least squares solution corresponding to the optimization problem (8). Again, no assets verify the equal risk contribution property.

Table 7: Volatility breakdown (in %) of naive and least squares solutions

	Naive solution				Least squares solution			
Asset	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$	x_i	$\mathcal{MR}_i$	$\mathcal{RC}_i$	$\mathcal{RC}_i^\star$
1	22.84	10.25	2.34	19.30	23.13	10.32	2.39	19.70
2	20.00	14.98	3.00	24.70	20.00	14.86	2.97	24.53
3	12.34	20.18	2.49	20.53	11.39	20.07	2.29	18.87
4	9.83	22.46	2.21	18.20	10.48	22.55	2.36	19.51
5	35.00	5.99	2.10	17.28	35.00	6.02	2.11	17.39
$\sigma(x)$			12.13				12.11	

Remark 8 *The previous example shows how to take into account a current allocation when building a risk budgeting portfolio. This approach is particularly relevant when considering dynamic rebalancing and tactical asset allocation. One of the main advantages of the RB approach is that it produces a stable allocation. However, it does not enable us to consider investment constraints such as the current allocation. Introducing weight constraints allows the fund manager to better control the portfolio construction.*

3.4.2 Risk budgeting with linear constraints

We now consider general linear constraints:

$$\Omega = \left\{x \in \mathbb{R}^n : Ax = B, Cx \leq D, x^- \leq x \leq x^+\right\}$$

These constraints generalize the case of lower and upper bounds. For example, it is common to add some (lower and upper) limits in terms of exposures by asset classes, sectors, ratings, etc. These types of limits are implemented thanks to inequality constraints $Cx \leq D$. Equality constraints $Ax = B$ are less common in portfolio optimization. In Appendix A.2 on page 29, we provide some closed-form formulas to calculate $\mathcal{P}_\Omega(x)$, when Ω corresponds to $Ax = B$ or $c^\top x \leq d$ or $x^- \leq x \leq x^+$. First, we notice that the analytical formula only exists for the half-space constraint $c^\top x \leq d$, but not for multiple inequality constraints $\Omega = \{x \in \mathbb{R}^n : Cx \leq D\}$ when $\operatorname{card}\Omega = m > 1$. The underlying idea is then to break down Ω as the intersection of m half-space sets:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \cdots \cap \Omega_m$$

where $\Omega_j = \left\{x \in \mathbb{R}^n : c_{(j)}^\top x \leq d_{(j)}\right\}$, $c_{(j)}^\top$ corresponds to the j^{th} row of C and $d_{(j)}$ is the j^{th} element of D. In this case, we can apply the Dykstra's algorithm for computing the proximal operator of $\mathbb{1}_\Omega(x)$. This algorithm is given on page 33. Second, mixing the constraints is not straightforward. Again, we can break down Ω as the intersection of three basic convex sets:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$$

where $\Omega_1 = \{\in \mathbb{R}^n : Ax = B\}$, $\Omega_2 = \{x \in \mathbb{R}^n : Cx \leq D\}$ and $\Omega_3 = \{x \in \mathbb{R}^n : x^- \leq x \leq x^+\}$. Since we know how to project each basic set, we calculate the projection $\mathcal{P}_\Omega(x)$ with the Dykstra's algorithm, which is described on page 34.

Table 8: Volatility and correlation matrix of asset returns (in %)

		1	2	3	4	5	6	7	8
σ_i		5.0	5.0	7.0	10.0	15.0	15.0	15.0	18.0
$\rho_{i,j}$	1	100							
	2	80	100						
	3	60	40	100					
	4	−20	−20	50	100				
	5	−10	−20	30	60	100			
	6	−20	−10	20	60	90	100		
	7	−20	−20	20	50	70	60	100	
	8	−20	−20	30	60	70	70	70	100

Let us consider an example of multi-asset allocation[12]. We consider a universe of eight asset classes: (1) US 10Y Bonds, (2) Euro 10Y Bonds, (3) Investment Grade Bonds, (4) High Yield Bonds, (5) US Equities, (6) Euro Equities, (7) Japan Equities and (8) EM Equities. In Table 8, we indicate the statistics used to compute the optimal allocation.

Using these figures, we calculate the risk parity portfolio, which corresponds to the second column in Table 9. We notice that the bond allocation is equal to 76.72% whereas the equity allocation is equal to 23.28%. In order to increase the equity allocation, we impose that the weight of the last four assets is greater than 30%. The solution is given in the fourth column. Finally, we overweight the allocation in European assets with respect to American assets by 5% (sixth column).

Table 9: The case of inequality constraints

$x_5+x_6+x_7+x_8 \geq 30\%$			✓		✓	
$x_2+x_6 \geq x_1+x_5+5\%$					✓	
Asset	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$
1	26.83	12.50	25.78	8.64	24.52	8.16
2	28.68	12.50	27.41	8.64	28.69	9.13
3	11.41	12.50	9.51	8.64	9.52	8.61
4	9.80	12.50	7.29	8.64	7.27	8.61
5	5.61	12.50	7.06	15.91	6.97	15.69
6	5.90	12.50	7.71	16.58	7.80	16.82
7	6.66	12.50	9.23	18.14	9.23	18.16
8	5.11	12.50	6.00	14.82	6.00	14.81
$\sigma(x)$ (in %)	4.78		5.20		5.19	

Remark 9 *Algorithm 3 is no longer valid when the coordinates are coupled via constraints. This is generally the case when we impose $Ax = B$ and $Cx \leq D$. This is why we use the ADMM-Newton or ADMM-CCD algorithms for solving this type of constrained risk budgeting problem.*

4 Applications

We consider two applications that are based on our professional experience. The first application is the design of risk-based equity indices. Building an ERC portfolio on the Eurostoxx 50 universe is straightforward. This is not the case if we consider the universe of the Eurostoxx index. Indeed, this universe contains many small cap stocks, and having the same risk contribution for small cap and large cap stocks may induce some liquidity issues. In particular, this type of problem occurs when considering a large universe of non-homogenous stocks. The second application is the control of rebalancing effects. This issue happens when we consider short-term covariance matrices. In this case, the allocation can be very reactive, implying large turnovers. For example, this type of situation is observed when we implement multi-asset risk parity strategies with daily or weekly rebalancing and empirical covariance matrices that are estimated with less than one year of historical data. These two applications are illustrated below.

[12]This example is taken from Roncalli (2013) on page 287.

4.1 Risk-based indexation and smart beta portfolios

We now consider a capitalization-weighted index composed of seven stocks. The weights are equal to 34%, 25%, 20%, 15%, 3%, 2% and 1%. We assume that the volatilities of these stocks are equal to 15%, 16%, 17%, 18%, 19%, 20% and 21%, whereas the correlation matrix of stock returns is given by:

$$\rho = \begin{pmatrix} 1.00 & & & & & & \\ 0.75 & 1.00 & & & & & \\ 0.73 & 0.75 & 1.00 & & & & \\ 0.70 & 0.70 & 0.75 & 1.00 & & & \\ 0.65 & 0.68 & 0.69 & 0.75 & 1.00 & & \\ 0.62 & 0.65 & 0.63 & 0.67 & 0.70 & 1.00 & \\ 0.60 & 0.60 & 0.65 & 0.68 & 0.75 & 0.80 & 1.00 \end{pmatrix}$$

As shown by Demey *et al.* (2010), the ERC portfolio defined by Maillard *et al* (2010) is a good candidate for building a risk-based equity index. However, an ERC index does not take into account liquidity constraints. For instance, we notice that the ERC allocation does not take into account the size of stocks in Table 10. We may assume that the three last assets are small cap stocks. In this case, it can be more realistic to distinguish small cap and large cap stocks. A first idea is to keep the CW weights on the small cap universe and to apply the ERC on the large cap universe. This solution called LC-ERC (for Large Cap ERC) is presented in Table 10. We face an issue here, because the ERC portfolio on large cap stocks does not depend on the full correlation matrix. Therefore, this solution assumes that the two universes of stocks are not correlated. We have the same issue if we consider the least squares solution (LS-ERC). A better approach is to find the ERC portfolio by imposing that the weights of small cap stocks are exactly equal to the corresponding CW weights. Let $\Omega_{\mathcal{SC}}$ be the universe of small cap stocks. We have $x_i = x_{\mathrm{cw},i}$ if $i \in \Omega_{\mathcal{SC}}$. This is equivalent to imposing the following weight constraints:

$$\begin{cases} 0 \le x_i & \text{if } i \notin \Omega_{\mathcal{SC}} \\ x_{\mathrm{cw},i} \le x_i \le x_{\mathrm{cw},i} & \text{if } i \in \Omega_{\mathcal{SC}} \end{cases}$$

The result corresponds to the C-ERC portfolio. Again, we notice that the property of equal risk contribution is satisfied at the global level for large cap stocks, contrary to LC-ERC and LS-ERC portfolios.

Table 10: Volatility breakdown (in %) of constrained ERC portfolios

Asset	CW		ERC		LC-ERC		LS-ERC		C-ERC	
	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$
1	34.00	32.08	17.22	14.29	25.81	23.39	26.62	24.23	25.87	23.46
2	25.00	24.82	15.90	14.29	24.06	23.44	24.20	23.63	24.07	23.46
3	20.00	20.92	14.78	14.29	22.44	23.44	22.09	23.08	22.46	23.46
4	15.00	16.01	13.83	14.29	21.69	23.57	21.09	22.89	21.59	23.46
5	3.00	3.10	13.17	14.29	3.00	3.10	3.00	3.10	3.00	3.10
6	2.00	2.03	12.86	14.29	2.00	2.02	2.00	2.02	2.00	2.02
7	1.00	1.05	12.23	14.29	1.00	1.05	1.00	1.05	1.00	1.05
$\sigma(x)$	14.50		15.23		14.68		14.66		14.68	

4.2 Managing the portfolio turnover

On page 18, we have considered a dynamic allocation and imposed some constraints in order to control the turnover of the portfolio. For that, we have used lower and upper bounds in order to impose a maximum deviation between the current allocation and the new allocation. This is a way of controlling the turnover. However, we can directly impose a turnover control. Let x_0 be the current allocation. The two-way turnover of Portfolio x with respect to Portfolio x_0 is defined by:

$$\begin{aligned} \tau\left(x \mid x_0\right) &= \sum_{i=1}^{n} \left|x_i - x_{0,i}\right| \\ &= \left\|x - x_0\right\|_1 \end{aligned}$$

It corresponds to the ℓ_1-norm of x with respect to the centroid vector x_0. Therefore, the corresponding Lagrange function is:

$$\mathcal{L}\left(x;\lambda\right) = \mathcal{R}\left(x\right) - \lambda \sum_{i=1}^{n} b_i \ln x_i + \mathbb{1}_{\Omega}\left(x\right)$$

where $\Omega = \left\{x \in R : \tau\left(x \mid x_0\right) \leq \tau^{\star}\right\}$ and $\tau^{\star}$ is the turnover limit. If we use the previous algorithms, the only difficulty is calculating the proximal operator[13] of $g\left(x\right) = \mathbb{1}_{\Omega}\left(x\right)$:

$$\mathbf{prox}_g\left(x\right) = \mathbf{prox}_f\left(x - x_0\right) + x_0$$

where $f\left(x\right) = \mathbb{1}_{\Omega'}\left(x\right)$ and $\Omega' = \left\{x \in R : \left\|x\right\|_1 \leq \tau^{\star}\right\}$. Finally, we deduce that:

$$\mathbf{prox}_g\left(x\right) = x - \mathbf{prox}_{\tau^{\star} \max}\left(\left|x - x_0\right|\right) \odot \operatorname{sign}\left(x - x_0\right)$$

where $\mathbf{prox}_{\lambda \max}\left(v\right)$ is the proximal operator given by Equation (24) on page 31.

We consider the example of multi-asset allocation on page 20. Let us assume that the current allocation is a 50/50 asset mix policy, where the weight of each asset class is 12.5%. In Table 11, we have reported the solution for different turnover limits $\tau^{\star}$. If the turnover limit is very low, the optimized RB portfolio is close to the current allocation. We verify that the optimized portfolio tends to the ERC portfolio when we increase the turnover limit.

Table 11: Constrained RB portfolios (in %) with turnover control

Asset	$\tau^{\star}$							
	0.00	10.00	20.00	30.00	40.00	50.00	60.00	70.00
1	12.50	14.86	17.28	19.68	22.01	24.28	26.58	26.83
2	12.50	15.14	17.72	20.32	22.99	25.72	28.42	28.68
3	12.50	12.50	12.50	12.50	12.50	12.50	11.65	11.41
4	12.50	12.50	12.50	12.50	12.50	11.50	9.90	9.80
5	12.50	11.20	9.70	8.49	7.27	6.28	5.66	5.61
6	12.50	12.02	10.36	9.02	7.69	6.63	5.95	5.90
7	12.50	12.50	11.72	10.16	8.66	7.47	6.71	6.66
8	12.50	9.28	8.22	7.33	6.39	5.62	5.14	5.11
$\tau\left(x^{\star} \mid x_0\right)$	0.00	10.00	20.00	30.00	40.00	50.00	60.00	61.02

[13] We use the properties of proximal operators described in Appendix A.2 on page 29.

5 Discussion and limitations of constrained risk budgeting portfolios

In this section, we discuss the concept of constrained risk budgeting allocation, and shows that it is not natural and has some limitations.

5.1 Coherent risk measures and the homogeneity property

Many people believe that the risk budgeting allocation is only related to the Euler decomposition:

$$\mathcal{R}(x) = \sum_{i=1}^{n} x_i \frac{\partial\,\mathcal{R}(x)}{\partial\,x_i}$$

implying that the risk measure is "convex". However, this concept is not always well-defined, and is often confused with the concept of coherent risk measure. Following Artzner *et al.* (1999), a risk measure $\mathcal{R}(x)$ is said to be coherent if it satisfies the following properties:

1. Subadditivity

   $$\mathcal{R}(x_1 + x_2) \leq \mathcal{R}(x_1) + \mathcal{R}(x_2)$$

   The risk of two portfolios should be less than adding the risk of the two separate portfolios.

2. Homogeneity

   $$\mathcal{R}(\lambda x) = \lambda \mathcal{R}(x) \quad \text{if } \lambda \geq 0$$

   Leveraging or deleveraging of the portfolio increases or decreases the risk measure in the same magnitude.

3. Monotonicity

   $$\text{if } x_1 \prec x_2 \text{, then } \mathcal{R}(x_1) \geq \mathcal{R}(x_2)$$

   If Portfolio x_2 has a better return than Portfolio x_1 under all scenarios, risk measure $\mathcal{R}(x_1)$ should be higher than risk measure $\mathcal{R}(x_2)$.

4. Translation invariance

   $$\text{if } m \in \mathbb{R} \text{, then } \mathcal{R}(x + m) = \mathcal{R}(x) - m$$

   Adding a cash position of amount m to the portfolio reduces the risk by m.

Föllmer and Schied (2002) propose replacing the homogeneity and subadditivity conditions by a weaker condition called the convexity property:

$$\mathcal{R}(\lambda x_1 + (1-\lambda) x_2) \leq \lambda \mathcal{R}(x_1) + (1-\lambda)\mathcal{R}(x_2)$$

This condition means that diversification should not increase the risk. Saying that the risk measure is convex is ambiguous. For some people, this means that $\mathcal{R}(x)$ satisfies the convexity property of Föllmer and Schied (2002), while for other people, this means that $\mathcal{R}(x)$ satisfies the Euler decomposition. It is true that these two concepts are related (Tasche, 2008), but they recover two different things (Kalkbrener, 2005). First, there is a confusion between the Euler decomposition and the Euler allocation principle (Tasche, 2008). Second, the Euler allocation principle does make sense only if the risk measure is subadditive (Kalkbrener, 2005). But another property is very important. Roncalli (2015) shows that risk budgeting is valid only if the homogeneity property is satisfied — $\mathcal{R}(\lambda x) = \lambda \mathcal{R}(x)$

— because this property ensures that there is a solution and the solution is unique. By definition, this property is related to the scaling property of the RB portfolio when there are no constraints. When we impose some constraints, it is obvious that the homogeneity property is valid only if the constraints Ω are compatible with the scaling property. This is not generally true except for some special cases. However, we don't need the homogeneity property to be satisfied for all values $\lambda \geq 0$. We need the homogeneity property to be met for a range of λ around the unconstrained risk budgeting portfolio. This is why imposing tight constraints can lead to a numerical solution without knowing if it corresponds to the true constrained risk budgeting portfolio.

5.2 The scaling puzzle

We consider the example given on page 20. If we consider the optimized portfolio subject to the constraint $\sum_{i=5}^{8} x_i \geq 30\%$, we may think that it is equivalent to the optimized portfolio subject to the constraint $\sum_{i=1}^{4} x_i \leq 70\%$. Results are given in Table 12, when we use the equally-weighted portfolio as the starting value $x^{(0)}$ in the ADMM algorithms. It is surprising to obtain two different solutions. Nevertheless, they are very close. If we do the same exercise with a minimum allocation of 40% in the equity asset class, the two solutions are very different (see columns 6 and 8 in Table 12). The problem is that we assume that the constraint $\sum_{i=1}^{n} x_i = 1$ is managed by the set Ω. This is not the case, because the constraint $\sum_{i=1}^{n} x_i = 1$ is managed by the Lagrange multiplier λ associated to the logarithmic barrier. Here, we face an important issue called the scaling compatibility problem. This means that a solution is acceptable if and only if the constraints Ω are "compatible" with the homogeneity property[14].

Table 12: Illustration of the scaling puzzle

$\sum_{i=5}^{8} x_i \geq 30\%$	✓							
$\sum_{i=1}^{4} x_i \leq 70\%$			✓					
$\sum_{i=5}^{8} x_i \geq 40\%$					✓			
$\sum_{i=1}^{4} x_i \leq 60\%$							✓	
Asset	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$	x_i	$\mathcal{RC}_i^\star$
1	25.78	8.64	23.39	6.50	24.09	4.35	18.73	2.01
2	27.41	8.64	24.34	6.11	25.09	4.35	19.08	1.68
3	9.51	8.64	12.46	10.98	6.57	4.35	12.48	7.84
4	7.29	8.64	9.81	12.07	4.24	4.35	9.71	10.43
5	7.06	15.91	7.30	16.09	8.74	18.59	9.82	19.51
6	7.71	16.58	7.66	16.09	10.75	21.87	10.27	19.51
7	9.23	18.14	8.46	16.09	14.09	27.32	11.15	19.51
8	6.00	14.82	6.57	16.09	6.42	14.82	8.76	19.51
$\sigma(x)$ (in %)	5.20		5.43		5.98		6.56	
$\mathcal{L}(x^\star;\lambda^\star)$ (in %)	13.29		20.86		10.68		28.27	
Global minimum	✓				✓			

How do we explain these results? The first reason is the choice of the starting value for initializing the algorithm. It is obvious that the bisection algorithm takes a road that depends on the initialization step. However, this reason is not the primary answer, because we observe

[14]For example, $\Omega = \{x \in \mathbb{R}^n : x_2 \geq 2x_1\}$ is compatible with the scaling property.

that we converge to the same solution whatever the starting value or the algorithm when we consider box or pointwise constraints. In fact, the main reason is the scaling property. In this case, the convergence path is crucial, because we can obtain local minima.

When the constraints are incompatible with the homogeneity property of the risk measure $\mathcal{R}(x)$, we may wonder if one solution is better than the others. In order to answer this question, we recall that the numerical algorithms solve the minimization problem $x^\star(\lambda) = \arg\min \mathcal{L}(x;\lambda)$ where:

$$\mathcal{L}(x;\lambda) = \mathcal{R}(x) - \lambda \sum_{i=1}^{n} b_i \ln x_i + \mathbb{1}_\Omega(x)$$

The idea is then to calculate $\mathcal{L}(x^\star(\lambda^\star);\lambda^\star)$ for the different solutions and to take the solution that gives the lowest value. In Table 12, the first and third portfolios are the best solutions. They have the lowest volatility and Lagrange function.

Remark 10 *Let us assume that the set of constraints includes the standard simplex: $\mathcal{S} \subset \Omega$. By construction, the sum of weights is always equal to* 1 *whatever the value of the Lagrange multiplier λ:*

$$\sum_{i=1}^{n} x_i^\star(\Omega,\lambda) = 1$$

It gives the impression that there are two solutions. However, there is only one solution which corresponds to the portfolio with the lowest risk measure.

6 Conclusion

In this paper[15], we propose an approach to find the risk budgeting portfolio when we impose some constraints. The underlying idea is to consider the logarithmic barrier problem and to use recent optimization algorithms in order to find the numerical solution. In particular, we use cyclical coordinate descent (CCD), alternative direction method of multipliers (ADMM), proximal operators and Dykstra's algorithm.

We provide different examples that are focused on the ERC portfolio. This portfolio is very interesting since it imposes the same risk contribution between the assets of the investment universe. We may then wonder what does an ERC portfolio mean when we impose some constraints. Most of the times, we observe that the ERC property continues to be satisfied for the assets that are not impacted by the constraints. This type of approach is very appealing when we would like to manage the liquidity of a portfolio, the small cap bias of risk-based indices or the turnover of a risk parity fund.

This study also highlights the importance of the homogeneity property of RB portfolios. Roncalli (2015) has already pointed out that the risk measure must be coherent, which can be incompatible when imposing constraints. The homogeneity property really asks the question of the compatibility between risk budgeting allocation and imposing some constraints. However, even if the calibration of the Lagrange multiplier λ remains an issue in some cases, our approach extends the optimization framework defined by Richard and Roncalli (2015), and reinforces that idea that risk budgeting and risk minimization are highly connected.

[15]This aim of this paper is also to help to popularize large-scale optimization algorithms that are very popular in machine learning, but are not well-known in finance. However, we think that they are relevant for many financial applications, in particular for portfolio optimization. Our paper which is focused on risk-budgeting optimization can then be seen as a companion work of Bourgeron *et al.* (2018) which focuses on mean-variance optimization.

References

[1] Artzner, A., Delbaen, F., Eber, J.-M., and Heath, D. (1999), Coherent Measures of Risk, *Mathematical Finance*, 9(3), pp. 203-228.

[2] Bai, X., Scheinberg, K. and Tutuncu, R. (2016), Least-squares Approach to Risk Parity in Portfolio Selection, *Quantitative Finance*, 16(3), pp. 357-376.

[3] Bauschke, H.H., and Borwein, J.M. (1994), Dykstra's Alternating Projection Algorithm for Two Sets, *Journal of Approximation Theory*, 79(3), pp. 418-443.

[4] Bourgeron, T., Lezmi, E., and Roncalli, T. (2018), Robust Asset Allocation for Robo-Advisors, *SSRN*, `www.ssrn.com/abstract=3261635`.

[5] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2010), Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends® in Machine learning*, 3(1), pp. 1-122.

[6] Combettes, P.L., and Müller, C.L. (2018), Perspective Functions: Proximal Calculus and Applications in High-dimensional Statistics, *Journal of Mathematical Analysis and Applications*, 457(2), pp. 1283-1306.

[7] Combettes, P.L., and Pesquet, J.C. (2011), Proximal Splitting Methods in Signal Processing, in Bauschke, H.H., Burachik, R.S., Combettes, P.L., Elser, V., Luke, D.R., and Wolkowicz, H. (Eds), *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, 48, pp. 185-212, Springer.

[8] Demey, P., Maillard, S, and Roncalli, T. (2010), Risk-Based Indexation, *SSRN*, `www.ssrn.com/abstract=1582998`.

[9] Dykstra, R.L. (1983), An Algorithm for Restricted Least Squares Regression, *Journal of the American Statistical Association*, 78(384), pp. 837-842.

[10] Feng, Y., and Palomar, D.P. (2015), SCRIP: Successive Convex Optimization Methods for Risk Parity Portfolio Design, *IEEE Transactions on Signal Processing*, 63(19), pp. 5285-5300.

[11] Föllmer, H., and Schied, A. (2002), Convex Measures of Risk and Trading Constraints, *Finance and Stochastics*, 6(4), pp. 429-447.

[12] Friedman, J., Hastie, T., and Tibshirani, R. (2010), Regularization Paths for Generalized Linear Models via Coordinate Descent, *Journal of Statistical Software*, 33(1), pp. 1-22.

[13] Gabay, D., and Mercier, B. (1976), A Dual Algorithm for the Solution of Nonlinear Variational Problems via Finite Element Approximation, *Computers & Mathematics with Applications*, 2(1), pp. 17-40.

[14] Griveau-Billion, T., Richard, J-C., and Roncalli, T. (2013), A Fast Algorithm for Computing High-dimensional Risk Parity Portfolios, *SSRN*, `www.ssrn.com/abstract=2325255`.

[15] He, B.S., Yang, H., and Wang, S.L. (2000), Alternating Direction Method with Self-Adaptive Penalty Parameters for Monotone Variational Inequalities, *Journal of Optimization Theory and applications*, 106(2), pp. 337-356.

[16] Kalkbrener M., (2005), An Axiomatic Approach to Capital Allocation, *Mathematical Finance*, 15(3), pp. 425-437.

[17] Maillard, S., Roncalli, T. and Teïletche, J. (2010), The Properties of Equally Weighted Risk Contribution Portfolios, *Journal of Portfolio Management*, 36(4), pp. 60-70.

[18] Nesterov, Y. (2004), *Introductory Lectures on Convex Optimization: A Basic Course*, Applied Optimization, 87, Kluwer Academic Publishers.

[19] Nesterov, Y. (2012), Efficiency of Coordinate Descent Methods on Huge-scale Optimization Problems, *SIAM Journal on Optimization*, 22(2), pp. 341-362.

[20] Parikh, N., and Boyd, S. (2014), Proximal Algorithms, *Foundations and Trends® in Optimization*, 1(3), pp. 127-239.

[21] Qian, E. (2005), Risk Parity Portfolios: Efficient Portfolios Through True Diversification, *Panagora Asset Management*, September.

[22] Richard, J-C., and Roncalli, T. (2015), Smart Beta: Managing Diversification of Minimum Variance Portfolios, in Jurczenko, E. (Ed.), *Risk-based and Factor Investing*, ISTE Press – Elsevier.

[23] Roncalli, T. (2013), *Introduction to Risk Parity and Budgeting*, Chapman & Hall/CRC Financial Mathematics Series.

[24] Roncalli, T. (2015), Introducing Expected Returns into Risk Parity Portfolios: A New Framework for Asset Allocation, *Bankers, Markets & Investors*, 138, pp. 18-28.

[25] Roncalli, T. and Weisang, G. (2016), Risk Parity Portfolio with Risk Factors, *Quantitative Finance*, 16(3), pp. 377-388.

[26] Spinu F. (2013), An Algorithm for the Computation of Risk Parity Weights, *SSRN*, `www.ssrn.com/abstract=2297383`.

[27] Tasche, D. (2008), Capital Allocation to Business Units and Sub-Portfolios: The Euler Principle, in Resti, A. (Ed.), *Pillar II in the New Basel Accord: The Challenge of Economic Capital*, Risk Books, pp. 423-453.

[28] Tibshirani, R.J. (2017), Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions, in Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (Eds), *Advances in Neural Information Processing Systems*, 30, pp. 517-528.

[29] Tseng, P. (1990), Dual Ascent Methods for Problems with Strictly Convex Costs and Linear Constraints: A Unified Approach, *SIAM Journal on Control and Optimization*, 28(1), pp. 214-242.

[30] Tseng, P. (2001), Convergence of a Block Coordinate Descent Method for Nondifferentiable Minimization, *Journal of Optimization Theory and Applications*, 109(3), pp. 475-494.

[31] Wang, S.L., and Liao, L.Z. (2001), Decomposition Method with a Variable Parameter for a Class of Monotone Variational Inequality Problems, *Journal of Optimization Theory and Applications*, 109(2), pp. 415-429.

[32] Wright, S.J. (2015), Coordinate Descent Algorithms, *Mathematical Programming*, 151(1), pp. 3-34.

Appendix

A Optimization algorithms

A.1 ADMM algorithm

The alternating direction method of multipliers (ADMM) is an algorithm introduced by Gabay and Mercier (1976) to solve problems which can be expressed as[16]:

$$\begin{aligned} \{x^\star, z^\star\} &= \arg\min f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz - c = 0 \end{aligned} \tag{18}$$

where $A \in \mathbb{R}^{p\times n}$, $B \in \mathbb{R}^{p\times m}$, $c \in \mathbb{R}^p$, and the functions $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions. Boyd *et al.* (2011) show that the ADMM algorithm consists of three steps:

1. The x-update is:
$$x^{(k)} = \arg\min \left\{ f(x) + \frac{\varphi}{2} \left\| Ax + Bz^{(k-1)} - c + u^{(k-1)} \right\|_2^2 \right\} \tag{19}$$

2. The z-update is:
$$z^{(k)} = \arg\min \left\{ g(z) + \frac{\varphi}{2} \left\| Ax^{(k)} + Bz - c + u^{(k-1)} \right\|_2^2 \right\} \tag{20}$$

3. The u-update is:
$$u^{(k)} = u^{(k-1)} + \left(Ax^{(k)} + Bz^{(k)} - c \right) \tag{21}$$

In this approach, $u^{(k)}$ is the dual variable of the primal residual $r = Ax + Bz - c$ and φ is the ℓ_2 penalty variable. In the paper, we use the notations $f^{(k)}(x)$ and $g^{(k)}(z)$ when referring to the objective functions that are defined in the x- and z-steps.

A.2 Proximal operator

In what follows, we give the main results that are summarized in Bourgeron *et al.* (2017). Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper closed convex function. The proximal operator $\mathbf{prox}_f(v) : \mathbb{R}^n \to \mathbb{R}^n$ is defined by:

$$\mathbf{prox}_f(v) = x^\star = \arg\min_x \left\{ f(x) + \frac{1}{2} \|x - v\|_2^2 \right\} \tag{22}$$

Since the function $f_v(x) = f(x) + \frac{1}{2}\|x - v\|_2^2$ is strongly convex, it has a unique minimum for every $v \in \mathbb{R}^n$ (Parikh and Boyd, 2014). For example, if we consider the logarithmic barrier function $f(x) = -\ln x$, we have:

$$\begin{aligned} f(x) + \frac{1}{2}\|x - v\|_2^2 &= -\ln x + \frac{1}{2}(x - v)^2 \\ &= -\ln x + \frac{1}{2}x^2 - xv + \frac{1}{2}v^2 \end{aligned}$$

[16] We follow the standard presentation of Boyd *et al.* (2011) on ADMM.

The first-order condition is $-x^{-1}+x-v=0$. We obtain two roots with opposite signs. Since the logarithmic function is defined for $x>0$, we deduce that the proximal operator is:

$$\mathbf{prox}_f(v)=\frac{v+\sqrt{v^2+4}}{2}$$

More generally, if we consider $f(x)=-\lambda\sum_{i=1}^n \ln x_i$, we have:

$$\left(\mathbf{prox}_f(v)\right)_i=\frac{v_i+\sqrt{v_i^2+4\lambda}}{2}$$

Let us now consider some special cases. If we assume that $f(x)=\mathbb{1}_\Omega(x)$ where Ω is a convex set, we have:

$$\begin{aligned}\mathbf{prox}_f(v) &= \arg\min_x \left\{\mathbb{1}_\Omega(x)+\frac{1}{2}\|x-v\|_2^2\right\} \\ &= \mathcal{P}_\Omega(v) \end{aligned} \tag{23}$$

where $\mathcal{P}_\Omega(v)$ is the standard projection. Here, we give the results of Parikh and Boyd (2014) for some simple polyhedra:

Ω	$\mathcal{P}_\Omega(v)$
$Ax=B$	$v-A^\dagger(Av-B)$
$a^\top x=b$	$v-\dfrac{\left(a^\top v-b\right)}{\|a\|_2^2}a$
$c^\top x\leqslant d$	$v-\dfrac{\left(c^\top v-d\right)_+}{\|c\|_2^2}c$
$x^-\leqslant x\leqslant x^+$	$\mathcal{T}(v;x^-,x^+)$

where $A^\dagger$ is the Moore-Penrose pseudo-inverse of A, and $\mathcal{T}(v;x^-,x^+)$ is the truncation operator:

$$\begin{aligned}\mathcal{T}\left(v;x^-,x^+\right) &= v\odot\mathbb{1}\left\{x^-\leqslant v\leqslant x^+\right\}+ \\ & \quad x^-\odot\mathbb{1}\left\{v<x^-\right\}+ \\ & \quad x^+\odot\mathbb{1}\left\{v>x^+\right\}\end{aligned}$$

In the case of complex polyhedra, the reader can find analytical formulas and numerical algorithms in Parikh and Boyd (2014), and Combettes and Pesquet (2011).

We also have:

$$\begin{aligned}x^\star &= \arg\min \frac{1}{2}\|x-v\|_2^2 \\ \text{s.t.} & \quad x\in\Omega\end{aligned}$$

If we define Ω as follows:

$$\Omega=\left\{x\in\mathbb{R}^n : Ax=B, Cx\leq D, x^-\leq x\leq x^+\right\}$$

we obtain:

$$\begin{aligned}x^\star &= \arg\min \frac{1}{2}x^\top x-v^\top x \\ \text{s.t.} & \quad \left\{\begin{array}{l}Ax=B\\ Cx\leq D\\ x^-\leq x\leq x^+\end{array}\right.\end{aligned}$$

Imposing linear constraints is then equivalent to solving a standard QP problem.

We now consider the case of norm functions. For that, we need a preliminary result. In the case of the pointwise maximum function $f(x) = \max x$, we have:

$$\mathbf{prox}_{\lambda f}(v) = \min(v, s^\star) \tag{24}$$

where $s^\star$ is the solution of the following equation:

$$s^\star = \left\{ s \in \mathbb{R} : \sum_{i=1}^{n} (v_i - s)_+ = \lambda \right\}$$

If we assume that $f(x) = \|x\|_p$, we obtain[17]:

p	$\mathbf{prox}_{\lambda f}(v)$
$p = 1$	$S_\lambda(v) = (\lvert v\rvert - \lambda \mathbf{1})_+ \odot \operatorname{sign}(v)$
$p = 2$	$\left(1 - \dfrac{1}{\max(\lambda, \|v\|_2)}\right) v$
$p = \infty$	$\mathbf{prox}_{\lambda \max}(\lvert v\rvert) \odot \operatorname{sign}(v)$

An important property of the proximal operator is the Moreau decomposition theorem:

$$\mathbf{prox}_f(v) + \mathbf{prox}_{f^*}(v) = v$$

where f^* is the convex conjugate of f. If $f(x)$ is a $\boldsymbol{\ell}_p$-norm function, then $f^*(x) = \mathbb{1}_{\mathcal{B}_p}(x)$ where $\mathcal{B}_p$ is the $\boldsymbol{\ell}_p$ unit ball. Since we have $\mathbf{prox}_{f^*}(v) = \mathcal{P}_{\mathcal{B}_p}(v)$, we deduce that:

$$\mathbf{prox}_f(v) + \mathcal{P}_{\mathcal{B}_p}(v) = v$$

More generally, we have:

$$\mathbf{prox}_{\lambda f}(v) + \lambda \mathcal{P}_{\mathcal{B}_p}\left(\frac{v}{\lambda}\right) = v$$

It follows that the projection on $\boldsymbol{\ell}_p$ ball can be deduced from the proximal operator of the $\boldsymbol{\ell}_p$-norm function. Let $\mathcal{B}_p(c, \lambda) = \left\{ x \in \mathbb{R}^n : \|x - c\|_p \leq \lambda \right\}$ be the $\boldsymbol{\ell}_p$ ball with center c and radius λ. We obtain:

p	$\mathcal{P}_{\mathcal{B}_p(\mathbf{0},\lambda)}(v)$
$p = 1$	$v - \mathbf{prox}_{\lambda \max}(\lvert v\rvert) \odot \operatorname{sign}(v)$
$p = 2$	$v - \mathbf{prox}_{\lambda \|\cdot\|_2}(\lvert v\rvert)$
$p = \infty$	$\mathcal{T}(v; -\lambda, \lambda)$

In the case where the center c is not equal to $\mathbf{0}$, we consider the translation property:

$$\mathbf{prox}_g(v) = \mathbf{prox}_f(v + c) - c$$

where $g(x) = f(x + c)$. Since we have the equivalence $\mathcal{B}_p(\mathbf{0}, \lambda) = \{x \in \mathbb{R}^n : f(x) \leq \lambda\}$ where $f(x) = \|x\|_p$, we deduce that:

$$\mathcal{P}_{\mathcal{B}_p(c,\lambda)}(v) = \mathcal{P}_{\mathcal{B}_p(\mathbf{0},\lambda)}(v - c) + c$$

[17] The proximal operator $S_\lambda(v)$ is known as the soft thresholding operator. In particular, it is used for solving lasso regression problems (Friedman *et al.*, 2010).

A.3 Dykstra's algorithm

We consider the following proximal problem:

$$x^{\star} = \mathbf{prox}_{f}\left(v\right)$$

where $f\left(x\right) = \mathbb{1}_{\Omega}\left(x\right)$ and:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \cdots \cap \Omega_m$$

The solution can be found thanks to Dykstra's algorithm (Dykstra, 1983; Bauschke and Borwein, 1994), which consists in the following two steps until convergence:

1. The x-update is:
$$x^{(k)} = \mathcal{P}_{\Omega_{\operatorname{mod}(k,m)}}\left(x^{(k-1)} + z^{(k-m)}\right)$$

2. The z-update is:
$$z^{(k)} = x^{(k-1)} + z^{(k-m)} - x^{(k)}$$

where $x^{(0)} = v$, $z^{(k)} = \mathbf{0}$ for $k < 0$ and $\operatorname{mod}\left(k, m\right)$ denotes the modulo operator taking values in $\{1, \ldots, m\}$.

Let us consider the case $\Omega = \{x \in \mathbb{R}^n : Cx \leq D\}$ where the number of inequality constraints is equal to m. We can write:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \cdots \cap \Omega_m$$

where $\Omega_j = \left\{x \in \mathbb{R}^n : c_{(j)}^{\top} x \leq d_{(j)}\right\}$, $c_{(j)}^{\top}$ corresponds to the j^{th} row of C and $d_{(j)}$ is the j^{th} element of D. We follow Tibshirani (2017) to define the corresponding algorithm. In particular, we introduce two iteration indices j and k. The index j refers to the constraint number, whereas the index k refers to the main loop. Algorithm 5 describes the Dykstra's approach for solving this proximal problem.

If we define Ω as follows:

$$\Omega = \left\{x \in \mathbb{R}^n : Ax = B, Cx \leq D, x^{-} \leq x \leq x^{+}\right\}$$

we decompose Ω as the intersection of three basic convex sets:

$$\Omega = \Omega_1 \cap \Omega_2 \cap \Omega_3$$

where $\Omega_1 = \{x \in \mathbb{R}^n : Ax = B\}$, $\Omega_2 = \{x \in \mathbb{R}^n : Cx \leq D\}$ and $\Omega_3 = \{x \in \mathbb{R}^n : x^{-} \leq x \leq x^{+}\}$. Using Dykstra's algorithm is equivalent to formulating Algorithm 6.

Remark 11 *An alternative approach is to write the constraints in the following way*[18]*:*

$$\Omega = \{x \in \mathbb{R}^n : C^{\star} x \leq D^{\star}\}$$

where $C^{\star} = \left(A, -A, C, -I_n, I_n\right)$ *and* $D^{\star} = \left(B, -B, D, -x^{-}, x^{+}\right)$. *Therefore, we can use Algorithm 5 to find the solution.*

[18] We use the following properties:

$$Ax = B \Leftrightarrow Ax \leq B \text{ and } Ax \geq B$$

and:

$$x^{-} \leq x \leq x^{+} \Leftrightarrow -x \leq -x^{-} \text{ and } x \leq x^{+}$$

Algorithm 5 Dykstra's algorithm for solving the proximal problem with inequality constraints

```
The goal is to compute the solution $x^\star = \mathbf{prox}_f(v)$ where $f(x) = \mathbb{1}_\Omega(x)$ and $\Omega = \{x \in \mathbb{R}^n : Cx \le D\}$
We initialize $x_m^{(0)} \leftarrow v$
We set $z_1^{(0)} \leftarrow \mathbf{0}, \ldots, z_m^{(0)} \leftarrow \mathbf{0}$
We note $k_{\max}$ the maximum number of iterations
for $k = 1 : k_{\max}$ do
    $x_0^{(k)} \leftarrow x_m^{(k-1)}$
    for $j = 1 : m$ do
        The $x$-update is:
            $x_j^{(k)} = \mathcal{P}_{\Omega_j}\left(x_{j-1}^{(k)} + z_j^{(k-1)}\right) = x_{j-1}^{(k)} + z_j^{(k-1)} - \frac{\left(c_{(j)}^\top x_{j-1}^{(k)} + c_{(j)}^\top z_j^{(k-1)} - d_{(j)}\right)_+}{\left\|c_{(j)}\right\|_2^2} c_{(j)}$
        The $z$-update is:
            $z_j^{(k)} = x_{j-1}^{(k)} + z_j^{(k-1)} - x_j^{(k)}$
    end for
    if $x_m^{(k)} = x_0^{(k-1)}$ then
        Break
    end if
end for
return $x^\star \leftarrow x_m^{(k)}$
```

A.4 Proximal operator of the risk budgeting logarithmic barrier

We have:

$$z^{(k)} = \arg\min g^{(k)}(z)$$

where:

$$\begin{aligned} g^{(k)}(z) &= g(z) + \frac{\varphi}{2}\left\|x^{(k)} - z + u^{(k-1)}\right\|_2^2 \\ &= -\lambda \sum_{i=1}^n b_i \ln z_i + \frac{\varphi}{2}\sum_{i=1}^n \left(x_i^{(k)} - z_i + u_i^{(k-1)}\right)^2 \\ &= \sum_{i=1}^n \left(\frac{\varphi}{2}\left(x_i^{(k)} - z_i + u_i^{(k-1)}\right)^2 - \lambda b_i \ln z_i\right) \end{aligned}$$

The first-order condition is:

$$\frac{\partial\, g^{(k)}(z)}{\partial\, z_i} = -\varphi\left(x_i^{(k)} - z_i + u_i^{(k-1)}\right) - \lambda b_i \frac{1}{z_i} = 0$$

We deduce that $z_i^{(k)}$ is the solution of the quadratic equation:

$$\left\{\begin{array}{l} \varphi z_i^2 - \varphi\left(x_i^{(k)} + u_i^{(k-1)}\right) z_i - \lambda b_i = 0 \\ z_i > 0 \end{array}\right.$$

Algorithm 6 Dykstra's algorithm for solving the proximal problem with general linear constraints

```
The goal is to compute the solution $x^{\star} = \mathbf{prox}_f(v)$ where $f(x) = \mathbb{1}_{\Omega}(x)$ and $\Omega = \{x \in \mathbb{R}^n : Ax = B, Cx \leq D, x^- \leq x \leq x^+\}$
We initialize $x_m^{(0)} \leftarrow v$
We set $z_1^{(0)} \leftarrow \mathbf{0}$, $z_2^{(0)} \leftarrow \mathbf{0}$ and $z_3^{(0)} \leftarrow \mathbf{0}$
We note $k_{\max}$ the maximum number of iterations
for $k = 1 : k_{\max}$ do
    $x_0^{(k)} \leftarrow x_m^{(k-1)}$
    For the set $\Omega_1$, we have:
        $$\left\{\begin{array}{l} x_1^{(k)} \leftarrow x_0^{(k)} + z_1^{(k-1)} - A^{\dagger}\left(Ax_0^{(k)} + Az_1^{(k-1)} - B\right) \\ z_1^{(k)} \leftarrow x_0^{(k)} + z_1^{(k-1)} - x_1^{(k)} \end{array}\right.$$
    For the set $\Omega_2$, we have[19]:
        $$\left\{\begin{array}{l} x_2^{(k)} \leftarrow \mathcal{P}_{\Omega_2}\left(x_1^{(k)} + z_2^{(k-1)}\right) \\ z_2^{(k)} \leftarrow x_1^{(k)} + z_2^{(k-1)} - x_2^{(k)} \end{array}\right.$$
    For the set $\Omega_3$, we have:
        $$\left\{\begin{array}{l} x_3^{(k)} \leftarrow \mathcal{T}\left(x_2^{(k)} + z_3^{(k-1)}; x^-, x^+\right) \\ z_3^{(k)} \leftarrow x_2^{(k)} + z_3^{(k-1)} - x_3^{(k)} \end{array}\right.$$
    if $x_3^{(k)} = x_0^{(k)}$ then
        Break
    end if
end for
return $x^{\star} \leftarrow x_3^{(k)}$
```

We have:

$$\Delta = \varphi^2\left(x_i^{(k)} + u_i^{(k-1)}\right)^2 + 4\varphi\lambda b_i$$

Since $\Delta > 0$ and $-\lambda\varphi b_i < 0$, we have two roots with opposite signs. Therefore, the solution is equal to:

$$z_i^{(k)} = \frac{\varphi\left(x_i^{(k)} + u_i^{(k-1)}\right) + \sqrt{\varphi^2\left(x_i^{(k)} + u_i^{(k-1)}\right)^2 + 4\varphi\lambda b_i}}{2\varphi}$$

A.5 CCD algorithm with separable constraints

The coordinate update proposed by Nesterov (2012) and Wright (2015) is:

$$x_i^{\star} = \arg\min\,(x - x_i)\,g_i + \frac{1}{2\eta}(x - x_i)^2 + \xi \cdot \mathbb{1}_{\Omega_i}(x_i)$$

[19]This step is done using Algorithm 5.

where ξ is a positive scalar, $\eta > 0$ is the stepsize of the gradient descent and g_i is the first-derivative of the function with respect to x_i:

$$g_i = -\pi_i + c\frac{(\Sigma x)_i}{\sqrt{x^\top \Sigma x}} - \lambda \frac{b_i}{x_i}$$

The objective function is equivalent to:

$$\begin{aligned}
(*) &= (x - x_i)\, g_i + \frac{1}{2\eta}(x - x_i)^2 + \xi \cdot \mathbb{1}_{\Omega_i}(x_i) \\
&= \frac{1}{2\eta}\Big((x - x_i)^2 + 2(x - x_i)\,\eta g_i\Big) + \xi \cdot \mathbb{1}_{\Omega_i}(x_i) \\
&= \frac{1}{2\eta}(x - x_i + \eta g_i)^2 + \xi \cdot \mathbb{1}_{\Omega_i}(x_i) - \frac{\eta}{2} g_i^2
\end{aligned}$$

By taking $\xi = \eta^{-1}$, we deduce that:

$$\begin{aligned}
x_i^\star &= \arg\min \mathbb{1}_{\Omega_i}(x_i) + \frac{1}{2}\left\| x - (x_i - \eta g_i) \right\|^2 \\
&= \mathbf{prox}_{\psi}(x_i - \eta g_i)
\end{aligned}$$

where $\psi(x) = \mathbb{1}_{\Omega_i}(x)$.

A.6 Accelerated bisection algorithm

The classic bisection algorithm consists in calculating the solution $x^\star(\lambda) = \arg\min \mathcal{L}(x;\lambda)$ and updating the bounds of the interval $[a_\lambda, b_\lambda]$ that contains the solution $\lambda^\star$ such that $\sum_{i=1}^{n} x^\star(\lambda) = 1$. One of the issues is that $x^\star(\lambda)$ is obtained by an optimization algorithm and is not an analytical formula. This means that we can write:

$$x^\star\left(\lambda; x^{(0)}\right) = \arg\min \mathcal{L}\left(x; \lambda, x^{(0)}\right)$$

where $x^{(0)}$ is the starting value of the ADMM or CCD algorithm. Therefore, we can update the starting value $x^{(0)}$ of the algorithm at each iteration of the bisection method. This helps to accelerate the convergence of the x-update. In practice, we can replace Algorithm 1 by Algorithm 7.

A.7 Adaptive penalization parameter

The convergence of the ADMM algorithm holds regardless of the value of the penalization parameter $\varphi > 0$. But the choice of φ affects the speed of convergence. In practice, the penalization parameter φ may be changed at each iteration, implying that φ is replaced by $\varphi^{(k)}$. This may improve the convergence and make the performance of the ADMM algorithm less dependent of the initial choice $\varphi^{(0)}$. To update $\varphi^{(k)}$ in practice, He *et al.* (2000) and Wang and Liao (2001) provide a simple and efficient scheme. Let $r^{(k)} = Ax^{(k)} + Bz^{(k)} - c$ and $s^{(k)} = \varphi A^\top B\left(z^{(k)} - z^{(k-1)}\right)$ be the primal and dual residual variables (Boyd *et al.*, 2011). On the one hand, the x and z-updates essentially comes from placing a penalty on $\left\| r^{(k)} \right\|_2^2$. As a consequence, if $\varphi^{(k)}$ is large, $\left\| r^{(k)} \right\|_2^2$ tends to be small. On the other hand, $s^{(k)}$ depends linearly on φ. As a consequence, if $\varphi^{(k)}$ is small, $\left\| s^{(k)} \right\|_2^2$ is small and $\left\| r^{(k)} \right\|_2^2$ may be large. To keep $\left\| r^{(k)} \right\|_2^2$ and $\left\| s^{(k)} \right\|_2^2$ within a factor μ, one may consider the following scheme:

Algorithm 7 Accelerated bisection method

```
The goal is to compute the optimal Lagrange multiplier $\lambda^\star$ and the solution $x^\star(\mathcal{S},\Omega)$
We consider two scalars $a_\lambda$ and $b_\lambda$ such that $a_\lambda < b_\lambda$ and $\lambda^\star \in [a_\lambda, b_\lambda]$
We note $\varepsilon_\lambda$ the convergence criterion of the bisection algorithm (e.g. $10^{-8}$)
We note $x^{(0)}$ the starting value of the ADMM/CCD algorithm
repeat
  We calculate $\lambda = \frac{a_\lambda + b_\lambda}{2}$
  We compute $x^\star\left(\lambda; x^{(0)}\right)$ the solution of the minimization problem:
      $x^\star\left(\lambda; x^{(0)}\right) = \arg\min \mathcal{L}\left(x; \lambda, x^{(0)}\right)$
  if $\sum_{i=1}^n x_i^\star\left(\lambda; x^{(0)}\right) < 1$ then
    $a_\lambda \leftarrow \lambda$
  else
    $b_\lambda \leftarrow \lambda$
  end if
  $x^{(0)} \leftarrow x^\star\left(\lambda; x^{(0)}\right)$
until $\left|\sum_{i=1}^n x_i^\star\left(\lambda; x^{(0)}\right) - 1\right| \leq \varepsilon_\lambda$
return $\lambda^\star \leftarrow \lambda$ and $x^\star(\mathcal{S},\Omega) \leftarrow x^\star\left(\lambda^\star; x^{(0)}\right)$
```

1. If $\left\|r^{(k)}\right\|_2^2 > \mu \left\|s^{(k)}\right\|_2^2$, the φ-update is:

$$\left\{\begin{array}{l} \varphi^{(k+1)} \leftarrow \tau\varphi^{(k)} \\ u^{(k+1)} \leftarrow \dfrac{u^{(k+1)}}{\tau} \end{array}\right.$$

2. If $\left\|s^{(k)}\right\|_2^2 > \mu \left\|r^{(k)}\right\|_2^2$, we have:

$$\left\{\begin{array}{l} \varphi^{(k+1)} \leftarrow \dfrac{\varphi^{(k)}}{\tau'} \\ u^{(k+1)} \leftarrow \tau' u^{(k+1)} \end{array}\right.$$

3. Otherwise, the penalization parameter remains the same $\varphi^{(k+1)} \leftarrow \varphi^{(k)}$, implying that we do not rescale the dual variable $u^{(k+1)}$.

The previous scheme corresponds to the φ-update and must be placed after the u-update of the ADMM algorithm. In practice, we use the following default values: $\varphi^{(0)} = 1$, $u^{(0)} = 0$, $\mu = 10^6$ and $\tau = \tau' = 2$.

B Implementation

A Python implementation is available on the following webpage:

`https://github.com/jcrichard`