Computing equal risk contribution portfolios

H. Mausser
O. Romanko

For institutional investors, optimizing the trade-off between risk and reward poses significant modeling and computational challenges. Notably, small errors in the estimated returns of financial assets can lead to optimized portfolios that incur far too much risk for the returns they actually deliver. Given these adverse effects, portfolio construction techniques that are based exclusively on risk have grown in popularity. For instance, equal risk contribution (ERC) portfolios seek to equalize the risk contributions of all assets, so that the portfolio is fully diversified from a risk perspective. This paper reviews the nonlinear optimization models that underlie the ERC approach and, in response to the reported difficulty of solving such optimization problems, compares the performance of several nonlinear programming algorithms when constructing ERC portfolios. Our results suggest that performance worsens with a poor choice of algorithm or a bad problem formulation. We provide alternative formulations and also develop heuristic procedures that construct an approximation to the ERC portfolio in an efficient manner.

Introduction

Investors have long appreciated that a balanced portfolio, which invests capital in many different assets, lessens the chance of large losses. However, it was not until 1952 that Markowitz [1] quantified the benefits of diversification and, in doing so, laid the foundation of Modern Portfolio Theory (MPT). Markowitz proposed that investors allocate funds among financial assets according to the mean-variance rule: select a so-called efficient portfolio that delivers the largest expected (mean) return with the lowest possible risk, where the latter is measured by the variance of the return. Still widely-used today, mean-variance analysis nevertheless presents several practical challenges. Among them is the fact that the efficient portfolios, found by solving an optimization problem, lack robustness; the portfolios are sensitive to small changes in the problem’s inputs, which are the estimated variance-covariance (VCV) matrix of the asset returns and, most significantly, the estimated mean returns [2]. While various methods can make the mean-variance results more robust [3–5], another proposal is to exclude the main source of the instability, namely the estimated mean returns, from the analysis. Two examples of the mean-free approach, the minimum variance portfolio and the so-called 1/N portfolio (the first references only the estimated VCV matrix, while the second simply weights all N assets equally), were found to perform surprisingly well in practice [6]. Recently, another mean-free strategy has gained widespread attention. Known as the equal risk contribution (ERC) or the risk parity approach, it seeks to equalize the risk contributions of all assets so that the resulting portfolio is fully diversified from a risk perspective [7].

By nature, ERC portfolios tend to invest primarily in low-risk assets, i.e., those that deliver stable, rather than volatile, returns. Thus, relative to a traditional 60/40 balanced portfolio (60% capital weight in stocks, 40% capital weight in bonds), ERC portfolios significantly overweight bonds and underweight equities on a capital basis. This asset mix is particularly favorable when interest rates are low or falling, which has generally been the case since the launch of the first ERC fund in 1996. The strong performance of the ERC approach to date has led to a proliferation of such funds, estimated to hold up to $200 billion in assets as of June 2013 [8]. Since 2012, index providers including Stoxx, VelocityShares, and FTSE (Financial Times Stock Exchange) have introduced ERC versions of many well-known stock indices. So-called “risk parity plus” funds, which tactically adjust an ERC portfolio based on return
expectations, are also available (such funds do not use a mean-free approach in a strict sense).

At the same time, there is ongoing debate about the merits of the ERC strategy. Critics argue, for instance, that the strategy’s performance is highly dependent on asset selection [9] and that its use of leverage (i.e., borrowing to invest) is dangerous [10]. Moreover, in today’s economic environment, the potential impact of rising interest rates on ERC funds is a key uncertainty, one for which there are conflicting viewpoints [11, 12].

Rather than joining this debate, we consider only the computational aspects of finding ERC portfolios. Like the mean-variance and minimum variance approaches, the ERC strategy requires solving an optimization problem to obtain the desired asset allocations. However, the structure of the ERC problem makes it harder to solve than the others. Specifically, the mean-variance and minimum variance problems have a quadratic objective function and linear constraints, which allows them to be solved by quadratic programming (QP) algorithms. In contrast, the ERC problem includes more general nonlinear functions that entail the use of nonlinear programming (NLP) techniques. Although commercial solvers are available for both types of problems, QP algorithms are faster, more reliable, and generally easier to use than NLP algorithms. For example, while the gradient vector of first derivatives and the Hessian matrix of second derivatives are both given by the coefficients of quadratic problems, these derivatives must be approximated or calculated analytically (typically with a user-supplied procedure) within NLP algorithms. The source of derivatives is one of many computational parameters—others include the solution algorithm and various numerical tolerances—that must be specified when using an NLP solver. Given their sensitivity to these parameters, different NLP solvers often exhibit varied performance for the same problem.

In response to the numerical difficulties reported in [7] when finding ERC portfolios with NLP methods, several alternative techniques were proposed. These are essentially fast, iterative algorithms that either apply in special cases, such as when returns follow a linear factor model [13], or are not guaranteed to find a valid portfolio [14]. While the difficulty of computing ERC portfolios, particularly when the number of assets is large, is often mentioned in the literature, we are not aware of any actual computational results in this regard. Thus, our first objective is to apply several different NLP solvers to problems like those in [7], to determine whether the numerical difficulty is inherent to the nonlinear optimization problem or simply the result of using a particular NLP algorithm.

Second, we investigate alternative problem formulations for finding ERC portfolios. In particular, we obtain formulations that can be solved with second-order conic programming (SOCP) algorithms [15]. SOCP problems are intermediate to quadratic programs and general nonlinear programs in terms of difficulty. While such problems can be solved by NLP methods, SOCP algorithms are an attractive option for several reasons, including their increased reliability, lack of derivative calculations, and greater accessibility (e.g., solvers such as IBM CPLEX* [16] include SOCP, but not NLP, algorithms).

Finally, we discuss several simple heuristic methods for obtaining approximate ERC portfolios. The heuristics, which require only the ability to solve quadratic programs, generally do not yield equal risk contributions for all assets. However, unlike the methods developed in [13, 14], they can incorporate restrictions on the asset allocations, as might exist in practice, e.g., an upper limit on the size of the allocations [17].

While the ERC approach has been associated exclusively with financial investment decisions, it represents a potential alternative to the mean-variance model wherever MPT is applied. The use of the latter for constructing portfolios of non-financial assets, with “return” replaced by a suitably chosen reward metric, presents interesting opportunities for applying the ERC strategy. For example, McKenna [18] advocates selecting information technology (IT) projects with MPT. In this case, reward is measured in terms of Net Present Value or Expected Commercial Value. Similarly, Bisias, Lo, and Watkins [19] use MPT to analyze funding decisions in biomedical research, where reward is the improvement in years of life lost. Another novel application of MPT relates to information retrieval, where documents must be ranked in terms of their relevance in response to an information request. Since a document’s relevance cannot be assessed with certainty, Wang and Zhu [20] draw parallels between a document’s relevance and an asset’s return and solve a mean-variance model to find a combination of documents whose overall relevance score has a high expected value and a low variance. In this case, the ERC strategy might be used to produce a diverse set of documents so that at least one is likely to be relevant to the information request.

The layout of the paper is as follows. The next section presents the portfolio construction problem and briefly describes the minimum variance and mean-variance optimization models. We then explain the ERC approach and examine its computational aspects in three parts: existing nonlinear formulations, alternative nonlinear and SOCP formulations, and heuristic methods.

Finally, we draw conclusions and suggest topics for further study.

**Portfolio construction**

Suppose that a fund manager can invest in a set of $N$ financial assets (e.g., stocks, bonds, options, etc.) to construct a portfolio. Let us denote the weight of asset $i$, i.e., the monetary value of its position divided by the total monetary value of the portfolio (assumed to be positive), by $x_i$. 
A portfolio is identified by a vector $\mathbf{x}$, comprising the weights of its $N$ constituent positions. By definition, the weights sum to one, i.e., $\sum_{i=1}^{N} x_i = 1$, while $x_i$ may be positive or negative depending on whether the portfolio is, respectively, long or short asset $i$ (in a long position, the asset is owned by the portfolio, while a short position results from the sale of a borrowed asset).

In general, portfolios may be subject to various restrictions, which we call structural constraints. Accordingly, we denote the set of acceptable weights by $\Omega \subseteq \mathbb{R}^N$ and say that a portfolio $\mathbf{x}$ is feasible if and only if $\mathbf{x} \in \Omega$. For example, if the structural constraints prohibit short positions but impose no further restrictions, a special case which we call “long-only”, then $\Omega = \{ \mathbf{x} \in \mathbb{R}^N \mid \sum_{i=1}^{N} x_i = 1; x_i \geq 0 \text{ for } i = 1, \ldots, N \}$.

For convenience, we assume that all structural constraints are linear functions of the weights, which guarantees that $\Omega$ is convex.

If asset $i$ delivers a return of $r_i$ during a specified time interval then the return of a portfolio $\mathbf{x}$ is $r(\mathbf{x}) = \mathbf{x}^T \mathbf{r} = \sum_{i=1}^{N} r_i x_i$, where $\mathbf{T}$ denotes the transpose. For investors, who are concerned with future performance, the challenge is to construct portfolios in the face of uncertain returns, i.e., in practice, $\mathbf{r}$ is a random vector and $r(\mathbf{x})$ is a random variable.

In contrast to the equal-weighting approach, which simply sets $x_i = N^{-1}$ for $i = 1, \ldots, N$, other portfolio construction techniques require estimating one or more characteristics of the joint asset returns. For example, the variance of $r(\mathbf{x})$ is $\sigma^2(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$, where $\mathbf{Q}$ is the estimated VCV matrix of the asset returns. Note that $Q_{ij} = \rho_{ij} \sigma_i \sigma_j$, where $\sigma_i$ is the standard deviation, or the volatility, of $r_i$ and $\rho_{ij}$ is the correlation between $r_i$ and $r_j$. Given these estimates, the minimum variance portfolio is found by solving

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} x_i x_j$$

s.t.

$$\mathbf{x} \in \Omega.$$  

(1)

The mean-variance approach requires also estimating the expected returns, $\mu = E(\mathbf{r})$, and then solving

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} x_i x_j$$

s.t.

$$\sum_{i=1}^{N} \mu_i x_i \geq \tilde{\mu}$$

$$\mathbf{x} \in \Omega.$$  

(2)

for a return level, $\tilde{\mu}$, consistent with the investor’s preferences.

Problems 1 and 2 are quadratic programs, which can be solved easily in practice. However, they both yield portfolios that tend to hold only a small number of assets. ERC portfolios, discussed in the next section, are typically more diversified but harder to compute.

The **equal risk contribution approach**

Let $R(\mathbf{x})$ be some measure of the risk of portfolio $\mathbf{x}$ and define $C_i(\mathbf{x})$ to be the risk contribution of asset $i$. By definition, it holds that $\sum_{i=1}^{N} C_i(\mathbf{x}) = R(\mathbf{x})$. If the portfolio’s risk is measured by the variance of its return then $R(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$ and $C_i(\mathbf{x}) = x_i (\mathbf{Q} \mathbf{x})_i$, where $(\mathbf{Q} \mathbf{x})_i = \sum_{j=1}^{N} Q_{ij} x_j$. Similarly, using the standard deviation of the return as the risk measure yields $R(\mathbf{x}) = \sqrt{\mathbf{x}^T \mathbf{Q} \mathbf{x}}$ and $C_i(\mathbf{x}) = (\mathbf{x}^T \mathbf{Q} \mathbf{x})_i / \sqrt{\mathbf{x}^T \mathbf{Q} \mathbf{x}}$.

An ERC portfolio $\mathbf{x}^{\text{ERC}}$ satisfies $C_i(\mathbf{x}^{\text{ERC}}) = (R(\mathbf{x}^{\text{ERC}}) / N)$ for $i = 1, \ldots, N$. Since $x_i (\mathbf{Q} \mathbf{x})_i = (\mathbf{x}^T \mathbf{Q} \mathbf{x})_i$ if and only if

$$x_i (\mathbf{Q} \mathbf{x})_i = \frac{\sqrt{\mathbf{x}^T \mathbf{Q} \mathbf{x}}}{N},$$

it follows that the ERC portfolios for the variance and the standard deviation risk measures are the same. Hereafter, we consider only the variance risk measure, recognizing that all results apply equally to the standard deviation.

Existing long-only formulations

The difficulty of finding an ERC portfolio depends on the problem characteristics, namely the VCV matrix $\mathbf{Q}$ and the set of feasible portfolios $\Omega$. In [7], the authors consider long-only portfolios, i.e., $\Omega = \{ \mathbf{x} \in \mathbb{R}^N \mid \sum_{i=1}^{N} x_i = 1; x_i \geq 0 \text{ for } i = 1, \ldots, N \}$. First, they show that if all correlations are the same then the ERC portfolio is given by the formula

$$x_i^{\text{ERC}} = \frac{\sigma_i^{-1}}{\sum_{j=1}^{N} \sigma_j^{-1}}, \quad i = 1, \ldots, N,$$

(3)

which guarantees that the weights are non-negative and sum to one.

More generally, when the correlations are different, the ERC portfolio must be found using numerical methods. In this case, Maillard, Roncalli, and Teiletche [7] propose to solve

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} (x_i (\mathbf{Q} \mathbf{x})_j - x_j (\mathbf{Q} \mathbf{x})_i)^2$$

s.t.

$$\mathbf{x} \in \Omega.$$  

(4)

Observe that Problem 4 minimizes the total squared differences between the risk contributions of all pairs of assets. Clearly, if a feasible ERC portfolio exists then the optimal value of Problem 4 is zero.
In [7], the authors also present an alternative approach. The Karush–Kuhn–Tucker optimality conditions imply that if \( y^* \) solves

\[
\begin{align*}
\text{min}_y & \quad y^T Q y \\
\text{s.t.} & \quad \sum_{i=1}^N \log(y_i) \geq c \\
 & \quad y_i \geq 0 \quad i = 1, \ldots, N
\end{align*}
\] (5)

then \( y_i^*(Qy^*) = \lambda_i \) for \( i = 1, \ldots, N \), where \( \lambda_i \) is the Lagrange multiplier of the sum of logarithms constraint. The ERC portfolio then is obtained by normalization, i.e.,

\[
x_{\text{ERC}}^i = y_i^* / \sum_{j=1}^N y_j^*. \]

If an algorithm only references the values, and not the derivatives, of the objective function and the constraints in its calculations, we refer to it as a "no derivative" algorithm. In practice, most NLP algorithms can utilize derivative information to speed up the solution procedure and require the gradient of the objective function and the Jacobian matrix of partial derivatives of the constraints to be supplied by the user or computed inside the algorithm. If analytic derivatives are not provided, most NLP solvers can compute derivatives by automatic differentiation or finite differences. In addition, many NLP algorithms can use not only first-order derivative information, but also second derivatives (Hessian matrix). Accordingly, second derivatives can be either approximated by Quasi-Newton techniques such as BFGS or a user-supplied analytic (exact) Hessian can be used.

We consider three different NLP solvers: the FMINCON function in MATLAB [21], an open-source code IPOPT.
Table 2  Results of solving Problems 4 and 5 to find long-only equal risk contribution (ERC) portfolios with 500 assets. (IPM, interior point method; SQP, sequential quadratic programming.)

<table>
<thead>
<tr>
<th>Solver</th>
<th>Q</th>
<th>Problem 4</th>
<th>Problem 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time (s)</td>
<td>Max error</td>
</tr>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>unscaled</td>
<td>4.57</td>
<td>3.66E-03</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td></td>
<td>2.04</td>
<td>3.56E-16</td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td></td>
<td>1.97</td>
<td>3.68E-03</td>
</tr>
<tr>
<td>KNITRO (IPM, exact Hessian)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, no derivatives)</td>
<td></td>
<td>2.10</td>
<td>3.68E-03</td>
</tr>
<tr>
<td>FMINCON (IPM, approx. Hessian)</td>
<td></td>
<td>1.42</td>
<td>3.68E-03</td>
</tr>
<tr>
<td>FMINCON (IPM, exact Hessian)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FMINCON (SQP, no derivatives)</td>
<td></td>
<td>3.81</td>
<td>3.68E-03</td>
</tr>
<tr>
<td>FMINCON (SQP, derivatives)</td>
<td></td>
<td>1.31</td>
<td>3.68E-03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solver</th>
<th>Q = scale 1E+04</th>
<th>Problem 4</th>
<th>Problem 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time (s)</td>
<td>Max error</td>
</tr>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>scale 1E+04</td>
<td>48.89</td>
<td>2.51E-12</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td></td>
<td>20.11</td>
<td>5.46E-09</td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td></td>
<td>87.61</td>
<td>4.10E-08</td>
</tr>
<tr>
<td>KNITRO (IPM, exact Hessian)</td>
<td></td>
<td>83.01</td>
<td>4.15E-08</td>
</tr>
<tr>
<td>FMINCON (IPM, no derivatives)</td>
<td></td>
<td>83.01</td>
<td>4.15E-08</td>
</tr>
<tr>
<td>FMINCON (IPM, approx. Hessian)</td>
<td></td>
<td>2048.28a</td>
<td>5.54E-06</td>
</tr>
<tr>
<td>FMINCON (IPM, exact Hessian)</td>
<td></td>
<td>2444.33a</td>
<td>8.40E-06</td>
</tr>
</tbody>
</table>

In Table 1, good (i.e., low error) solutions are obtained for Problem 5 in all cases. IPOPT finds slightly better solutions for the unscaled version of Problem 4 than KNITRO and FMINCON, but this is apparently due to numerical tolerances. All solvers produce good solutions for the scaled version of Problem 4. Since the solution times are generally less than one second, there is no evidence of computational difficulties for this problem.

Next, we find a long-only ERC portfolio comprising 500 stocks, selected at random from the Morgan Stanley Capital International (MSCI) world index. Elements of the VCV matrix range in magnitude from $10^{-5}$ to $10^{-3}$. In this case, scaling the VCV matrix by $10^4$ is an effective choice.
Thus, Table 2 reports the results for two instances, when the VCV matrix is unscaled and multiplied by 10^4. Once again, good solutions are obtained for Problem 5 with the scaled VCV matrix in all cases, but now the solution times vary widely. While interior point algorithms with exact Hessians solve the problem in roughly one second, the SQP algorithm in FMINCON takes more than half an hour to find a solution, regardless of how derivatives are obtained. We do not solve Problem 4 with an exact Hessian since its calculation takes too long with 500 assets. Good solutions are found in the remaining cases, with SQP again taking much longer than the interior point methods. Overall, Problem 5 can be solved faster than Problem 4, suggesting that the former has better numerical properties.

Table 3  Results of solving Problem 6 to find long-only equal risk contribution (ERC) portfolios with 500 assets. (IPM, interior point method; SQP, sequential quadratic programming.)

<table>
<thead>
<tr>
<th>Solver</th>
<th>Q</th>
<th>Time (s)</th>
<th>Max error</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>unscaled</td>
<td>0.07</td>
<td>3.68E-03</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td>7.05</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td>0.16</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>KNITRO (IPM, exact Hessian)</td>
<td>1.85</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, no derivatives)</td>
<td>1.08</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, approx. Hessian)</td>
<td>0.67</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, exact Hessian)</td>
<td>3.82</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>FMINCON (SQP, no derivatives)</td>
<td>0.47^a</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>FMINCON (SQP, derivatives)</td>
<td>0.42^a</td>
<td>3.68E-03</td>
<td></td>
</tr>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>scale 1E+04</td>
<td>1.38</td>
<td>2.86E-09</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td>24.42</td>
<td>3.33E-07</td>
<td></td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td>1.76</td>
<td>2.54E-06</td>
<td></td>
</tr>
<tr>
<td>KNITRO (IPM, exact Hessian)</td>
<td>11.19</td>
<td>1.32E-06</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, no derivatives)</td>
<td>27.85</td>
<td>5.11E-05</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, approx. Hessian)</td>
<td>10.66</td>
<td>5.11E-05</td>
<td></td>
</tr>
<tr>
<td>FMINCON (IPM, exact Hessian)</td>
<td>25.54</td>
<td>5.11E-05</td>
<td></td>
</tr>
<tr>
<td>FMINCON (SQP, no derivatives)</td>
<td>66.95^a</td>
<td>1.02E-06</td>
<td></td>
</tr>
<tr>
<td>FMINCON (SQP, derivatives)</td>
<td>183.27^a</td>
<td>1.12E-06</td>
<td></td>
</tr>
</tbody>
</table>

Table 4  Results of solving Problems 11 and 14 to find long-only equal risk contribution (ERC) portfolios with 500 assets. (IPM, interior point method; SQP, sequential quadratic programming.)

<table>
<thead>
<tr>
<th>Solver</th>
<th>Q</th>
<th>Time (s)</th>
<th>Max error</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>unscaled</td>
<td>7.05</td>
<td>1.52E-01</td>
</tr>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>877.73^a</td>
<td>13.72</td>
<td>5.42E-13</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td>899.27</td>
<td>2708.34^1</td>
<td>3.99E-03</td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td>1133.19</td>
<td>73.06</td>
<td>4.74E-15</td>
</tr>
<tr>
<td>KNITRO (IPM, exact Hessian)</td>
<td>scale 1E+04</td>
<td>7.31</td>
<td>3.14E-07</td>
</tr>
<tr>
<td>CPLEX</td>
<td>scale 1E+04</td>
<td>5.28</td>
<td>1.25E-15</td>
</tr>
<tr>
<td>IPOPT (IPM, approx. Hessian)</td>
<td>589.33</td>
<td>25.22</td>
<td>5.42E-13</td>
</tr>
<tr>
<td>IPOPT (IPM, exact Hessian)</td>
<td>31.22^a</td>
<td>2072.85^1</td>
<td>3.85E-03</td>
</tr>
<tr>
<td>KNITRO (IPM, approx. Hessian)</td>
<td>15.33^a</td>
<td>22.09</td>
<td>1.12E-13</td>
</tr>
</tbody>
</table>

Thus, Table 2 reports the results for two instances, when the VCV matrix is unscaled and multiplied by 10^4. Once again, good solutions are obtained for Problem 5 with the scaled VCV matrix in all cases, but now the solution times vary widely. While interior point algorithms with exact Hessians solve the problem in roughly one second, the SQP algorithm in FMINCON takes more than half an hour to find a solution, regardless of how derivatives are obtained. We do not solve Problem 4 with an exact Hessian since its calculation takes too long with 500 assets. Good solutions are found in the remaining cases, with SQP again taking much longer than the interior point methods. Overall, Problem 5 can be solved faster than Problem 4, suggesting that the former has better numerical properties.
Alternative long-only formulations

The way that an optimization problem is formulated can dramatically affect the performance of solution algorithms. In light of this, consider the following alternative to Problem 4

$$\min_{x, \tau} \sum_{i=1}^{N} (x_i(Qx)_i - \tau)^2$$

s.t.

$$x \in \Omega.$$  \hspace{1cm} (6)

Observe that Problem 6 minimizes the total squared differences between the risk contributions of all assets and some unconstrained variable $\tau$ (using calculus, it is easy to show that $\tau = (x^TQx/N)$ in an optimal solution). Like Problem 4, Problem 6 attains a minimal value of zero when $x = x^{\text{ERC}}$, so these problems are effectively equivalent in the long-only case, when an ERC portfolio is known to exist.

As shown in Table 3, good solutions to the 500-asset problem with a scaled VCV matrix are obtained in all cases when solving Problem 6. Moreover, the solution times are much lower than those for Problem 4 (see Table 2), with IPOPT and KNITRO taking about one second when using an interior point method and an approximate Hessian. Evidently, simplifying the objective function reduces the computational cost of calculating derivatives and results in fewer nonlinearities [25].

Since Problem 6 has a fourth-order polynomial objective function, it is still a general nonlinear program that must be solved with NLP methods. In fact, it is possible to reformulate Problems 4 and 5 so that they can be solved with an SOCP algorithm. The new formulations rely in part on transformations of constraints involving products of non-negative variables into second-order cone constraints (e.g., [15, 26, 27]). For instance, given real variables $s_1 \geq 0$, $s_2 \geq 0$ and $t$, the so-called hyperbolic constraint $s_1s_2 \geq t^2$ is equivalent to the second order cone constraint

$$(2t)^2 + (s_1 - s_2)^2 \leq (s_1 + s_2)^2.$$  \hspace{1cm} (7)

Consider the expression

$$\sqrt{x^TQx/N} = \sqrt{\min_{1 \leq i \leq N} \{x_i(Qx)_i\}}$$

which computes the difference between the square root of the average risk contribution and the square root of the smallest risk contribution. Observe that Equation (8) is zero if $x = x^{\text{ERC}}$ and positive otherwise. Thus, minimizing Equation (8) is a suitable objective function for finding long-only ERC portfolios. Adapting Problem 4 to use this objective function, we first introduce non-negative variables $z_i = (Qx)_i$ for $i = 1, \ldots, N$. Next, we introduce non-negative variables $t$ and $p$ and add the constraints

$$x^TQx \leq Np^2$$  \hspace{1cm} (9)

and

$$x_i z_i \geq t^2, \quad i = 1, \ldots, N.$$  \hspace{1cm} (10)

Note that $p^2$ is an upper bound for the average risk contribution, while $t^2$ is a lower bound for the individual risk contributions, so that $p \geq t$ always. Since $p = t$ for an ERC portfolio, our goal is to minimize $p - t$, which corresponds to Equation (8). This leads to the following SOCP problem:

$$\min_{x, z, p, t} \quad p - t$$

s.t.

$$z_i = (Qx)_i, \quad i = 1, \ldots, N$$

$$x^TQx \leq Np^2$$

$$x_i z_i \geq t^2, \quad i = 1, \ldots, N$$

$$x \in \Omega$$

$$z_i \geq 0, \quad i = 1, \ldots, N$$

$$p, t \geq 0.$$  \hspace{1cm} (11)
Turning to Problem 5, observe that
\[ \sum_{i=1}^{N} \log(y_i) \geq c \] (12)
is equivalent to
\[ y_1 y_2 \ldots y_N \geq e^c. \] (13)
Given real variables \( t \) and \( s_k \geq 0, k = 1, \ldots, 2^K \), the constraint \( s_k y_1 y_2 \geq t^k \) can be expressed as a set of \( 2^K - 1 \) partial products, each of which is a hyperbolic constraint. These, in turn, can be transformed into second order cone constraints per Equation (7). Thus, it follows that an alternative formulation of Problem 5 is
\[
\begin{align*}
\min & \quad y^T Q y \\
\text{s.t.} & \quad y_{2(i-1)+j} y_{2i} \geq t_j^2 \\
& \quad t_{2(j-1)+k} y_{2j} \geq t_{2^{k+1}+j}^2 \\
& \quad y_{2^{k-1}} \geq e^{2s_k} \\
& \quad y_i \geq 0, \quad i = 1, \ldots, N \\
& \quad y_i = 1, \quad i = N + 1, \ldots, 2^K \\
& \quad y_j \geq 0, \quad j = 1, \ldots, 2^K - 1
\end{align*}
\] (14)
where \( K \) is the smallest value satisfying \( 2^K \geq N \).

For the 500-asset problem, Table 4 reports the results of solving Problems 11 and 14 with the NLP algorithms in IPOPT and KNITRO, and the SOCP algorithm in CPLEX. Some of the NLP algorithms fail to obtain good solutions, either exceeding the allowed number of iterations or terminating with an error due to numerical difficulties. Surprisingly, the SOCP algorithm is not the fastest overall; the best performance is obtained when solving Problem 5 with an interior point method and an exact Hessian, followed closely by solving Problem 6 with an interior point method and an approximate Hessian. This is related to the fact that the SOCP complementary slackness conditions yield a dense matrix, so that the Cholesky factorization performed by the SOCP algorithm takes a long time.

**Additional constraints**

All of the problems considered thus far have included the long-only restriction, which is of practical interest because many fund managers cannot take short positions. Moreover, as noted above, an ERC portfolio is guaranteed to exist in the long-only case. Removing this restriction and/or adding more structural constraints complicates the problem of finding ERC portfolios. Notably, Problems 5 and 14 are no longer valid if the long-only restriction is modified.

When short positions are allowed, multiple ERC portfolios can exist for a given set of assets. In this case, Problems 4 and 6 remain valid but there are multiple local minima; alternative ERC portfolios can be found by specifying different starting points for the NLP algorithm. While interesting, the construction of ERC portfolios with short positions is beyond the scope of this paper (see [25]). Conversely, if the long-only restriction is augmented by further structural constraints then a feasible ERC portfolio may not exist. However, Problems 4, 6, and 11 each can be solved to find a portfolio that is “close” to the ERC portfolio, in the sense that their respective objective functions are as small as possible. Note that the specialized algorithms in [13, 14] are not applicable in this case.

To illustrate, we add a constraint to the 500-asset problem so that the long-only ERC portfolio is no longer feasible. Specifically, in the long-only ERC portfolio, the 50 assets having the smallest weights collectively represent about 5% of the portfolio value. Our constraint requires these same 50 assets to make up at least 30% of the value.

Table 5 shows the results when Problems 4, 6, and 11 are solved with CPLEX and IPOPT. Instead of a single error number, we now report both the largest over- and under-contributions (again, versus the ideal relative risk contribution of 0.002) to highlight the effects of the different objective functions. For instance, Problems 4 and 6, by virtue of minimizing squared differences, penalize over- and under-contributions more or less equally. In contrast, Problem 11 penalizes large under-contributions without regard for over-contributions. Evidently, Problems 4 and 6 yield the same portfolio in this case, with relative risk contributions ranging from 0 to 0.0133 (the two formulations are not equivalent, however, as their solutions differ for other problems). Conversely, for Problem 11 the optimal portfolio has relative risk contributions ranging from 0.001225 to 0.260. From a performance standpoint, Problem 6 represents the best formulation, while the interior point method with approximate Hessian in IPOPT is the fastest algorithm.

**Heuristic procedures**

Our results suggest that, when constructing ERC portfolios of practical size, the associated nonlinear optimization problems can be solved readily with a well-selected NLP or SOCP algorithm. We now present heuristic methods, requiring nothing beyond QP capability, that can be used to construct approximate ERC portfolios.

**Convex combination heuristic**

In the long-only case, an ERC portfolio can be viewed as being intermediate to the minimum variance and the 1/N portfolios [7]. This motivates the following simple heuristic procedure: given any two feasible portfolios, find the convex combination thereof that minimizes the total squared differences between all pairs of risk.
contributions (i.e., solve Problem 4 with the added constraint that the portfolio be a convex combination of two “endpoint” portfolios). Since all such convex combinations are feasible, provided that the endpoints are feasible and \( \Omega \) is convex, it is possible to solve the problem analytically.

Let \( \tilde{x} \in \Omega \) and \( \check{x} \in \Omega \) denote the two endpoints and consider the portfolio \( x(\lambda) = \lambda\tilde{x} + (1 - \lambda)\check{x} \) for \( 0 \leq \lambda \leq 1 \), so that \( x(\lambda) \) is a convex combination of \( \tilde{x} \) and \( \check{x} \). Denote the variance contributions of asset \( i \) to portfolios \( \tilde{x} \) and \( \check{x} \) by \( C_i \) and \( \bar{C}_i \), respectively. The variance contribution of asset \( i \) to portfolio \( x(\lambda) \) is

\[
C_i(\lambda) = [\lambda \tilde{x}_i + (1 - \lambda)\check{x}_i] \sum_{j=1}^{n} \sigma_j [\lambda \tilde{x}_j + (1 - \lambda)\check{x}_j] \\
= \lambda \left[ \tilde{x}_i \bar{C}_i + (1 - \lambda) \check{x}_i \bar{C}_i \right] \\
+ (1 - \lambda) \left[ \tilde{x}_i C_i + (1 - \lambda) \check{x}_i \bar{C}_i \right].
\]

The difference between the contributions of assets \( i \) and \( j \) is

\[
f_{ij}(\lambda) = C_i(\lambda) - C_j(\lambda) = \lambda^2 (\bar{C}_i - \bar{C}_j) + (1 - \lambda)^2 (\bar{C}_i - \bar{C}_j) \\
+ \lambda(1 - \lambda) \left( \frac{\tilde{x}_i \bar{C}_i}{\tilde{x}_j} - \frac{\tilde{x}_i \bar{C}_i}{\check{x}_j} + \frac{\check{x}_i \bar{C}_i}{\tilde{x}_j} - \frac{\check{x}_i \bar{C}_i}{\check{x}_j} \right).
\]

We want to find \( \lambda^* \) that minimizes

\[
F(\lambda) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}(\lambda)^2.
\]

Define \( \alpha_{ij} = \tilde{x}_i - \bar{C}_i, \beta_{ij} = (\bar{C}_i - \bar{C}_j) \) and \( \delta_{ij} = ((\check{x}_i \bar{C}_i/\tilde{x}_j) - (\check{x}_i \bar{C}_i/\check{x}_j) + (\check{x}_i \bar{C}_i/\tilde{x}_j) - (\check{x}_i \bar{C}_i/\check{x}_j)) \). It is straightforward to show that

\[
(\frac{dF}{d\lambda}) = 2(a\lambda^3 + b\lambda^2 + c\lambda + d)
\]

where

\[
a = \sum_{i=1}^{n} \sum_{j=1}^{n} 2(\alpha_{ij} + \beta_{ij} - \delta_{ij})^2
\]

\[
b = \sum_{i=1}^{n} \sum_{j=1}^{n} 3(\alpha_{ij} + \beta_{ij} - \delta_{ij})(-2\beta_{ij} + \delta_{ij})
\]

\[
c = \sum_{i=1}^{n} \sum_{j=1}^{n} 2\beta_{ij}(\alpha_{ij} + \beta_{ij} - \delta_{ij}) + (-2\beta_{ij} + \delta_{ij})^2
\]

\[
d = \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij}(-2\beta_{ij} + \delta_{ij}).
\]

To find \( \lambda^* \), we solve the cubic equation

\[
a\lambda^3 + b\lambda^2 + c\lambda + d = 0.
\]

If there are one or more real roots between 0 and 1 then \( \lambda^* \) equals the root that minimizes Equation (17). If there are no real roots between 0 and 1, then one of the endpoints is the best approximation to the ERC portfolio. In this case, \( \lambda^* \) equals 0 if \( F(0) < F(1) \) or 1 otherwise.
The quality of the heuristic solution, $x(\lambda^*)$, depends significantly on the endpoint portfolios. Good candidates might be the $1/N$ portfolio, the approximate ERC portfolio that assumes equal correlations [via Equation (3)] or the minimum variance portfolio. Ideally, the endpoint portfolios would be obtained analytically or by solving relatively simple (e.g., linear or quadratic) optimization problems.

**Trade-off heuristic**

Instead of considering only convex combinations of two endpoint portfolios, one can adopt more of a mean-variance approach and construct intermediate portfolios by optimal trade-off. Like Problem 2, we now minimize variance but replace the mean return constraint with an upper bound on the linear distance from some target portfolio, $\bar{x}$, i.e.,

$$\min_x \sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} x_i x_j$$

s.t.

$$\sum_{i=1}^{N} |x_i - \bar{x}_i| \leq U$$

$$\mathbf{x} \in \Omega.$$  \hspace{1cm} (22)

Problem 22 yields the target portfolio when $U = 0$ and the minimum variance portfolio when $U$ is sufficiently large. Since Problem 22 is a quadratic program, it can be solved efficiently for multiple values of $U$ to generate a “frontier” of portfolios intermediate to the minimum variance and target portfolios. We then compute the total squared differences between all pairs of risk contributions for each of these portfolios and select the one having the minimal value.

**Computational results**

We use the heuristics to find an approximate long-only ERC portfolio for the 9-asset problem. The convex combination heuristic fixes the minimum variance portfolio as one endpoint and one of the $1/N$ portfolio or the equal correlation ERC portfolio as the other endpoint. The trade-off heuristic uses the $1/N$ portfolio as the target portfolio in one case and the equal correlation ERC portfolio as the target portfolio in the other.

**Figure 1** plots the ERC error, or the square root of the total squared differences between all pairs of risk contributions, against the standard deviation of return for all intermediate portfolios obtained by the heuristics and for the actual ERC portfolio. In this case, the best approximation is a convex combination of the minimum variance and the equal correlation ERC portfolios. As shown in **Figure 2**, the heuristic solutions exhibit more diversified risk contributions than the minimum variance portfolio,
with the best approximation coming quite close to the ERC portfolio. In Figure 2, the risk contribution of each asset is shown by a different color. Also, “mVar” denotes the minimum variance portfolio, “eERC” denotes the equal correlation ERC portfolio, “(c)” stands for convex combination heuristic, and “(t)” stands for trade-off heuristic.

**Conclusions**

The presence of general nonlinear functions makes the optimization problems for constructing ERC portfolios harder to solve than the quadratic programming problems that yield mean-variance or minimum variance portfolios. If solving ERC optimization problems proves difficult, our investigation suggests that computational performance can be improved by changing the solution algorithm and/or the problem formulation. In particular, SOCP formulations provide practical alternatives to general NLP models.

NLP solvers have many parameters that affect their speed and solution quality. In our experiments, varying the solution algorithm and the method of calculating numerical derivatives had a significant impact on the performance of three NLP solvers—FMINCON, IPOPT, KNITRO—when constructing a long-only ERC portfolio with 500 assets. In particular, the SQP algorithm (which had numerical difficulties in [7]) was typically much slower than the IPM algorithm.

Our computational results suggest that minimizing the total squared differences between all pairs of risk contributions, the objective function typically cited in the literature, is rather ineffective for constructing large ERC portfolios. A different problem formulation, which combines a quadratic objective function with a sum-of-logarithms constraint, is better as it can be solved efficiently by IPM algorithms with an exact Hessian matrix. However, the latter formulation is limited to long-only portfolios. In light of this, we provided efficient nonlinear and SOCP formulations for the more general case of arbitrarily-constrained long ERC portfolios. Due to their greater reliability and fewer parameters, SOCP solvers are generally easier to deploy than NLP solvers. We also proposed heuristic methods that find approximations to ERC portfolios without using SOCP or NLP solvers. While not considered in this paper, constructing ERC portfolios with short positions is an interesting problem that deserves further study.

Finally, we emphasize that the intent of our computational experiments is not to recommend a particular solver, but to show that nonlinear ERC optimization problems can be solved effectively if the problem formulation and the solution algorithm are chosen appropriately. Indeed, further scaling of the problem and additional parameter tuning may significantly improve the performance of the solvers used in this study.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**References**

package for nonlinear optimization,” in Large-Scale Nonlinear
Optimization, G. di Pillo and M. Roma, Eds. New York, NY,
24. B. Bruder and T. Roncalli, “Managing risk exposures using the
risk parity in portfolio selection,” in Optimization Online e-print
org/DB_FILE/2013/10/4089.pdf
26. M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret,
“Applications of second-order cone programming,” Linear
27. F. Alizadeh and D. Goldfarb, “Second-order cone programming,”

Received October 15, 2013; accepted for publication
November 19, 2013

H. Mausser IBM Software Group. Toronto, Canada M5T 2C6
(helmut.mausser@ca.ibm.com). Dr. Mausser is a Lead Mathematician
in the Quantitative Research Group, Risk Analytics at IBM Canada.
He received a B.Math. degree in mathematics in 1987 and an M.A.Sc.
degree in management sciences in 1992, both from the University
of Waterloo, and a Ph.D. degree in operations research from the
University of Colorado in 1997. He then joined Algorithmics
Incorporated in Toronto, where he has developed decision support tools
for financial risk management. Dr. Mausser has coauthored 30
publications and two patents, and he is a member of INFORMS
(Institute for Operations Research and the Management Sciences).

O. Romanko IBM Software Group. Toronto, Canada M5T 2C6
(oleksandr.romanko@ca.ibm.com). Dr. Romanko is a Senior Research
Analyst, Risk Analytics - Business Analytics at IBM Canada
(Quantitative Research Group). He holds Ph.D. degree in computer
science from McMaster University. Research interests of Dr. Romanko
include financial and risk modeling, portfolio optimization, business
analytics, operations research, multiobjective and parametric
optimization, and computational algorithms. He has published
a number of research papers and won several awards from IBM,
Canadian Operational Research Society, Mitacs, and INFORMS
(Institute for Operations Research and the Management Sciences).