Portfolio Selection with Multiple Criteria

Bernhard Pfaff  
bernhard_pfaff@fra.invesco.com

Invesco Asset Management GmbH  
Frankfurt am Main

R in Finance, Chicago IL, 20 and 21 May 2016
Overview

- Definition of multiple (conflicting) criteria objectives with respect to portfolio optimization problems.
- Concept of non-dominated solutions (Pareto efficiency).
- Solutions/points can determined by:
  1. GA/EMO: elitist non-dominated sorting algorithms (e.g. NSGA-II)
  2. MCDM: (classical) optimization (e.g. weighted-sum method)
- Applied to multi-asset portfolio with objectives for:
  1. Mean return,
  2. Total Covariance Risk,
  3. Diversification with respect to assets’ risk contributions.
Problem formulation

\[
\begin{align*}
\text{minimize} & \quad f_m(\omega), & m = 1, 2, \ldots, M; \\
\text{subject to} & \quad g_j(\omega) \geq 0, & j = 1, 2, \ldots, J; \\
& \quad h_k(\omega) = 0, & k = 1, 2, \ldots, K; \\
& \quad \omega_i^{(L)} \leq \omega_i \leq \omega_i^{(U)}, & i = 1, 2, \ldots, n.
\end{align*}
\]

- Problem: \( M \) (conflicting) objective functions and \( n \) (constrained) variables.
- A solution \( \hat{\omega} \in \Omega \) is efficient (Pareto optimal or non-dominated) if there is no \( \omega \in \Omega \) such that \( f_k(\omega) \leq f_k(\hat{\omega}) \) for \( k = 1, \ldots, p \) and \( f_i(\omega) < f_i(\hat{\omega}) \) for some \( i \in \{1, \ldots, k\} \).
Goal: find solutions which lie on Pareto-efficient front and encompass its entire range.

Solutions can be determined by genetic NSGA-II algorithm.

However, no guarantee in finding optimal points on (close to) frontier.

NSGA-II consists of the following steps (a) create random population, (b) selection (non-dominant/constraint-dominant), (c) applying variation (crossover, mutation), (d) elitism (crowding distances).

An implementation is provided in the R package mco (see Mersmann, 2014).
Multi Criteria Optimization

MCDM: Weighted-sum method

\[
\begin{align*}
\text{minimize} & \quad \sum_{m=1}^{M} \lambda_m f_m(\omega), \quad \text{with} \quad \lambda_m \geq 0; \\
\text{subject to} & \quad g_j(\omega) \geq 0, \quad j = 1, 2, \ldots, J; \\
& \quad h_k(\omega) = 0, \quad k = 1, 2, \ldots, K; \\
& \quad \omega^{(L)}_i \leq \omega_i \leq \omega^{(U)}_i, \quad i = 1, 2, \ldots, n.
\end{align*}
\]

- Scaling/normalization of objective functions is important.
- Goals and/or satisficing levels can be included.
- Possibility of a hybrid approach, where some objectives are specified as \( \epsilon \)-constraints.
Multi Criteria Optimization

Synopsis of portfolio applications

- Bi-criterion EMO problem formulation with discontinuities in search space (risk & return with (i) either zero or within bounds allocations and (ii) cardinality constraint on count of included assets): Deb et al. (2011).
- Tri-criterion problems (quad-lin-lin):
  - risk, return and transaction costs: Steuer et al. (2005), Steuer et al. (2013).
Example: Multi-asset class portfolio

Specification

- MCO consisting of three objectives:
  1. mean return,
  2. volatility, and
  3. dispersion of risk contributions.
- Targeted return of 6% p.a.
- Targeted volatility of 4% p.a.
Example: Multi-asset class portfolio

R code: Initializing

```r
> library(FRAPO)
> library(mco)
> ## Loading of data
> data(MultiAsset)
> Prices <- timeSeries(MultiAsset,
+     charvec = rownames(MultiAsset))
> NAssets <- ncol(Prices)
> R <- returns(Prices, method = "discrete", percentage = TRUE)
> ## Defining parameters
> TargetRpa <- 6 ## percentage p.a.
> TargetR <- 100 * ((1 + TargetRpa / 100)^(1 / 12) - 1)
> TargetVol <- 4 ## percentage p.a.
> l <- rep(1, 3) ## goal weighting
> WeightedSum <- FALSE
> mu <- colMeans(R)
> S <- cov(R)
```
Example: Multi-asset class portfolio

R code: Multiple criteria objective and budget constraint

```r
> f <- function(x){
+   y <- numeric(3)
+   y[1] <- -1.0 * l[1] * drop(crossprod(x, mu)) / TargetR
+   y[2] <- l[2] * drop(sqrt(t(x) %*% S %*% x)) *
+           sqrt(12) / TargetVol
+   if(WeightedSum){
+     return(sum(y))
+   } else {
+     return(y)
+   }
+ }
> g <- function(x){
+   c(1.01 - sum(x), sum(x) - 0.99)
+ }
```
Example: Multi-asset class portfolio

R code: Determining Pareto efficient solutions

```r
> ans <- nsga2(f, NAssets, 3,
+    lower.bounds = rep(0, NAssets),
+    upper.bounds = rep(1, NAssets),
+    constraints = g, cdim = 2, popsize = 500)
> names(ans)
[1] "par"      "value"   "pareto.optimal"

> ## Preparing objective values for graphics
> mco <- data.frame(ans$value[ans$pareto.optimal, ])
> mco[, 1] <- ((1 + (-1.0 * mco[, 1] * TargetR) / 100)^12
+    - 1.0) * 100
> mco[, 2] <- mco[, 2] * TargetVol
> colnames(mco) <- c("Return", "Risk", "Diversification")
```
Example: Multi-asset class portfolio

R code: 3D scatterplot

```r
> library(scatterplot3d)
> scatterplot3d(mco,
+       main = "Pareto Efficient Solutions",
+       sub = "Pareto Frontier (Surface)",
+       xlab = "Return Objective",
+       ylab = "Risk Objective",
+       zlab = "Dispersion of MRC",
+       angle = 15,
+       highlight.3d = FALSE,
+       box = TRUE,
+       color = "steelblue",
+       pch = 19, type = "p",
+       cex.symbols = 0.6)
```

Pareto Efficient Solutions

Pareto Frontier (Surface)
Example: Multi-asset class portfolio

R code: image plot

```r
> library(akima)
> library(fields)
> s <- interp(mco[, 2], mco[, 1], mco[, 3],
+          xo = seq(min(mco[, 2]), max(mco[, 2]), length = 100),
+          yo = seq(min(mco[, 1]), max(mco[, 1]), length = 100),
+          duplicate = "mean"
+          )
> par(mar = c(5, 6, 5, 6))
> image.plot(s, nlevel = 50,
+          main = "Image plot of efficient set",
+          legend.lab = "Dispersion of MRC",
+          xlab = "Risk Objective",
+          ylab = "Return Objective",
+          legend.mar = 4,
+          horizontal = TRUE,
+          legend.shrink = 0.7,
+          col = topo.colors(50))
> contour(s, add = TRUE, nlevels = 20, labcex = 0.8)
> points(mco[, 2], mco[, 1], pch = 18, cex = 0.4, col = "orange")
```
Example: Multi-asset class portfolio

R code: weighting of objectives

```r
grid <- expand.grid(x = seq(0.05, 0.95, by = 0.05),
                    y = seq(0.05, 0.95, by = 0.05))
grid <- grid[which(rowSums(grid) <= 1.0), ]
wobj <- as.matrix(cbind(grid, 1 - rowSums(grid)),
                   nrow = nrow(grid), ncol = 3)
W <- matrix(NA, nrow = nrow(wobj), ncol = NAssets)
WeightedSum <- TRUE
IneqA <- matrix(1, nrow = 1, ncol = NAssets)
ev <- rep(1 / NAssets, NAssets)
library(fPortfolio) ## for donlp2NLP
for(i in 1:nrow(wobj)){
  l <- c(wobj[i, ])
  W[i, ] <- donlp2NLP(start = ev, objective = f,
                      par.lower = rep(0, NAssets),
                      ineqA = IneqA, ineqA.lower = 1.0,
                      ineqA.upper = 1.0)$solution
}
```
Example: Multi-asset class portfolio

R code: weighting of objectives & ternary plot

```r
> library(PerformanceAnalytics)
> library(ggtern) ## Wahrschau! version < 2.0.1
> Es95Mod <- apply(W, 1, function(x){
+   r <- timeSeries(R %*% x / 100, time(R))
+   -100 * ES(r)
+ })
> terndat <- data.frame(cbind(wobj, Es95Mod))
> colnames(terndat) <- c("x", "y", "z", "value")
> ## Theme for ternary plot
> terntheme <- function(){
+   list(theme_rbg(),
+        theme(legend.position = c(0, 1),
+              legend.justification = c(0, 1),
+              plot.margin=unit(c(0, 2,0, 2), "cm"))
+   )
+ }
```
Example: Multi-asset class portfolio

R code: ternary plot, cont’d

```r
## ternary plot
> ggtern(terndat, aes(x = x, y = y, z = z, value = value)) +
  geom_interpolate_tern(aes(value = value, color = ..level..),
                       binwidth = 1.0) +
  terntheme() +
  theme_hidegrid_minor() +
  theme_showgrid_major() +
  Lline(0.2, color = "blue", linetype = 2) +## x
+ Tline(0.3, colour = "red2", linetype = 2) +## y
+ Rline(0.5, color = "brown", linetype = 2) +## z
+ scale_color_gradient(low = "green", high = "red") +
+ labs(x = "Return", y = "Risk", z = "MRC",
      title = "Ternary Plot with ES Contour Lines",
      color = "Level")
```

Ternary Plot with ES Contour Lines

Pfaff (Invesco)
Example: Multia-set class portfolio

R code: backtest, part I

```r
# library(cccp) ## for ERC portfolio
# backtest, extending window
# ep <- time(R)[-c(1:59)]
# bs <= length(ep)
# sp <= rep(start(R), bs)
# initialising object
# Wmco <= matrix(NA, nrow = bs, ncol = NAssets)
# Wmsr <= Wmdp <= Wgmv <= Werc <= Wmco
# l <= c(0.2, 0.1, 0.7) ## goal weighting
```
Example: Multi-asset class portfolio

R code: backtest, part II

```r
> for(i in 1:bs){
+    rdat <- window(R, start = sp[i], end = ep[i])
+    mu <- colMeans(rdat)
+    S <- cov(rdat)
+    Wmco[i, ] <- donlp2NLP(start = ew, objective = f,
+                           par.lower = rep(0, NAssets),
+                           ineqA = IneqA, ineqA.lower = 1.0,
+                           ineqA.upper = 1.0)$solution
+    ans <- tangencyPortfolio(rdat)
+    Wmsr[i, ] <- getWeights(ans)
+    ans <- PMD(rdat)
+    Wmdp[i, ] <- FRAPO::Weights(ans) / 100
+    ans <- PGMV(rdat)
+    Wgmv[i, ] <- FRAPO::Weights(ans) / 100
+    ans <- rp(ew, S, ew, optctrl = ctrl(trace = FALSE))
+    Werc[i, ] <- c(getx(ans))
+ }
```
Example: Multi-asset class portfolio

R code: backtest, part III

```r
> W <- list("MCO" = Wmco, "MSR" = Wmsr, "MDP" = Wmdp,
+ "GMV" = Wgmv, "ERC" = Werc)
> E <- lapply(W, function(x){
+ wTs <- timeSeries(x, charvec = ep)
+ wTsL1 <- lag(wTs, 1)
+ RetFac <- 1 + rowSums(R[ep, ] * wTsL1) / 100.0
+ RetFac[1] <- 100
+ timeSeries(cumprod(RetFac), charvec = ep)
+ })
> cols <- topo.colors(6)
> plot(E[[1]], lwd = 2,
+ ylab = "Index", xlab = "", col = cols[1],
+ main = "Comparison of Allocation Strategies")
> lines(E[[2]], col = cols[2])
> lines(E[[3]], col = cols[3])
> lines(E[[4]], col = cols[4])
> lines(E[[5]], col = cols[5])
> legend("topleft",
+ legend = c("MCO", "MSR", "MDP", "GMW", "ERC"),
+ col = cols, lty = 1, lwd = 2)
> abline(h = 100, col = "gray")
```
Example: Multi-asset class portfolio

R code: backtest, part IV

```r
> Rstrat <- matrix(unlist(lapply(E, Return.calculate)), ncol = 5)
> RstratTs <- na.omit(xts(Rstrat, order.by = as.Date(ep)))
> Bench <- xts(rep(0, nrow(RstratTs)), order.by = as.Date(ep)[-1])
> S1 <- as.matrix(table.AnnualizedReturns(RstratTs, Rf = Bench,
+ scale = 12))
> S2 <- VaR(RstratTs)
> ans <- rbind(S1, -100 * S2)
> colnames(ans) <- c("MCO", "MSR", "MDP", "GMV", "ERC")
> rownames(ans) <- c("Return (p.a.)", "StdDev. Risk (p.a.)",
+ "Sharpe Ratio", "VaR (p.a.)")
> round(ans, 3)

<table>
<thead>
<tr>
<th></th>
<th>MCO</th>
<th>MSR</th>
<th>MDP</th>
<th>GMV</th>
<th>ERC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return (p.a.)</td>
<td>0.061</td>
<td>0.060</td>
<td>0.058</td>
<td>0.051</td>
<td>0.060</td>
</tr>
<tr>
<td>StdDev. Risk (p.a.)</td>
<td>0.038</td>
<td>0.039</td>
<td>0.037</td>
<td>0.034</td>
<td>0.034</td>
</tr>
<tr>
<td>Sharpe Ratio</td>
<td>1.605</td>
<td>1.532</td>
<td>1.585</td>
<td>1.513</td>
<td>1.746</td>
</tr>
<tr>
<td>VaR (p.a.)</td>
<td>1.301</td>
<td>1.380</td>
<td>1.308</td>
<td>1.211</td>
<td>1.017</td>
</tr>
</tbody>
</table>
```
Summary

- Aiding decision makers by making portfolio choices for conflicting objectives (*a posteriori analysis*).
- Allows amendment of classical portfolio optimization formulations (e.g. GMV, ERC, MDP and/or MSR) by additional goals.
- MCDM: For tri-criterion formulations, depiction of solutions by a fourth portfolio characteristic/measure is feasible by means of ternary plots.
- Caveat/strength of EMO: It is at the user’s discretion to chose his ‘optimal’ allocation out of the Pareto efficient set, which might be a challenge on its own.
Bibliography


Bibliography III


