OPTIMIZING OMEGA

S.J. Kane and M.C. Bartholomew-Biggs
School of Physics Astronomy and Mathematics, University of Hertfordshire
Hafield AL10 9AB, United Kingdom
M. Cross and M. Dewar
Numerical Algorithms Group, Oxford

Abstract
This paper considers the Omega function, proposed by Cascon, Keating & Shadwick, as a performance evaluation measure for comparison between financial assets. We discuss the optimization problems which arise when the Omega function is used as a basis for portfolio selection. We show that the problem of choosing invested fractions to maximize Omega typically has many local solutions and we describe some preliminary computational experience using a global optimization technique from the NAG library.

1 Introduction

The Omega function was introduced in [1] as a measure to be used in comparing the performance of financial assets. A brief description is as follows. Suppose we have a history of returns for an asset, measured over m days. Let r_{min} and r_{max} be, respectively, the worst and best values of observed return. The cumulative distribution of these returns is a monotonically non-decreasing curve like the one shown in Figure 1. The horizontal axis represents the observed returns, r ($r_{min} \leq r \leq r_{max}$) and the vertical scale shows $f(r)$ as the fraction of the observations for which the measured return was less than or equal to r.

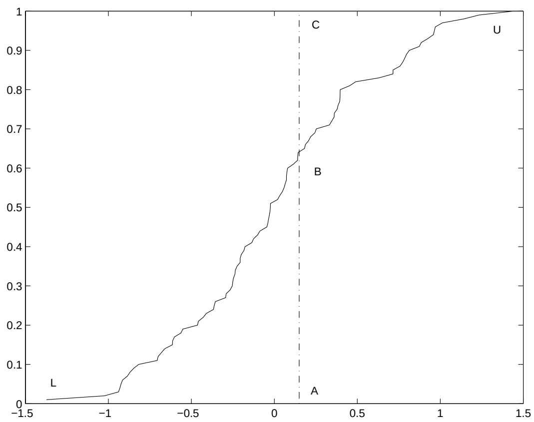

Figure 1: Typical cumulative distribution of asset returns

In Figure 1, the point **A** represents some threshold value for return r_t. The Omega function associated with r_A is defined as

$$\Omega = \frac{\int_{r_t}^{r_{max}} (1 - f(r))dr}{\int_{r_{min}}^{r_t} f(r)dr} \tag{1}$$

Geometrically this represents the ratio

$$\Omega = \frac{\text{Area } \mathbf{BCU}}{\text{Area } \mathbf{LAB}}$$

If r_t is close to r_{min} then area **BCU** is much larger than area **LAB** and so Ω is large. Conversely, $\Omega \to 0$ as $r_t \to r_{max}$. Thus, if r_t represents a desired rate of return, we can take Ω as a measure of the extent to which the historical performance of an asset has exceeded this target. Hence we would regard an asset which gives a large value of $\Omega(r_t)$ as a better investment than one for which $\Omega(r_t)$ is small.

Figure 2 shows how Ω varies with the threshold value r_t for the asset with the cumulative distribution in Figure 1.

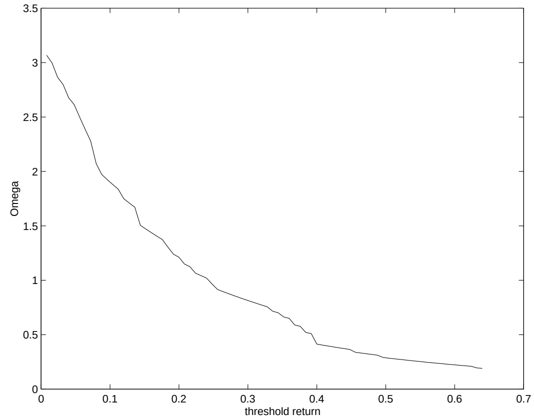

Figure 2: An Omega plot for $0.0\% \leq r_t \leq 0.65\%$

Given a specified level of threshold return, the Omega function can be used to compare two or more assets. Figure 3 shows cumulative distributions of returns for three assets and Figure 4 shows their Omega functions plotted for returns in the range 0.4% to 0.8%.

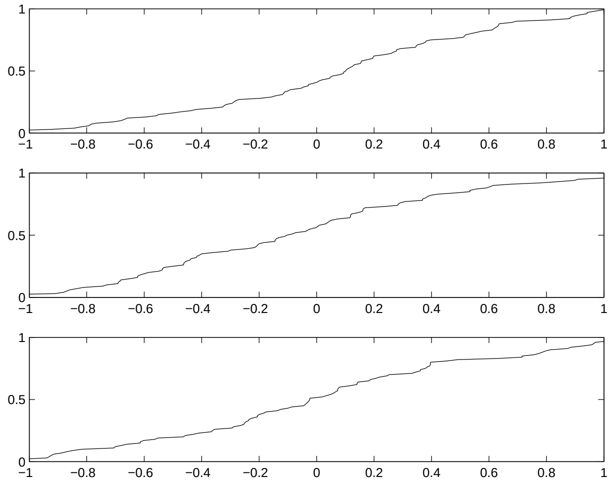

Figure 3: Cumulative distributions of returns on assets A, B and C (top to bottom)

Figure 4 indicates that asset A is always to be preferred to asset B because its Ω value is higher for all values of return. If the threshold return is less than about 0.6% then asset A is also better than asset C. However this situation changes as the threshold is increased and asset C becomes the best investment when target return is between 0.6% and 0.8%.

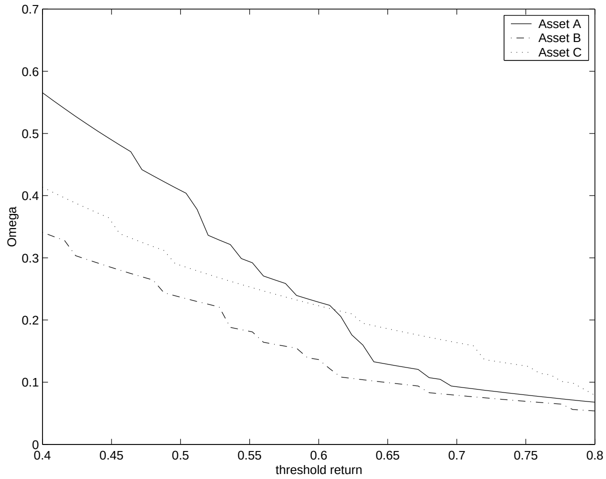

Figure 4: Omega plots for assets A,B and C

In this paper we regard Ω not just as a means of comparing individual assets but as a basis for portfolio selection. This has also been discussed by Favre-Bulle & Pache [2]. To explain how this is done, consider a portfolio involving three assets in which y_1, y_2 and y_3 denote the invested fractions. For any values of y_1, y_2, y_3 we can use historical data to determine how the portfolio would have performed over the previous m days; and from this we can construct a cumulative return distribution for the portfolio, as in Figure 1. For any threshold return r_t we can then evaluate $\Omega(r_t)$. The value of $\Omega(r_t)$ derived in this way obviously depends on y_1, y_2 and y_3 and so a possible way of optimizing the portfolio would be to choose y_1, y_2, y_3 to *maximize* $\Omega(r_t)$. This represents a significantly different strategy from the familiar Markowitz approach which involves choosing the invested fractions to minimize a measure of portfolio risk based on the variance-covariance matrix of the asset returns.

2 Portfolio optimization using Ω – a simple example

Figures 5–7 show some artifical 50-day return histories. Figure 5 relates to an asset that experiences modest gains and losses quite frequently while Figures 6, 7 correspond to assets which make bigger gains and losses less often. The cumulative return distributions for assets 1,2 and 3 are shown in Figures 8–10.

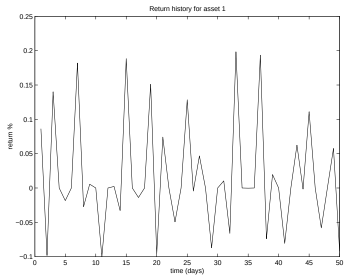

Figure 5: Return history for asset 1

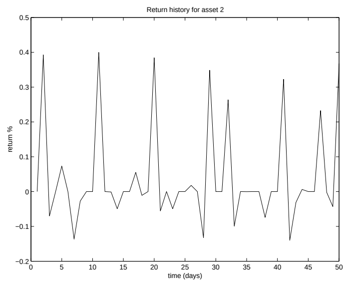

Figure 6: Return history for asset 2

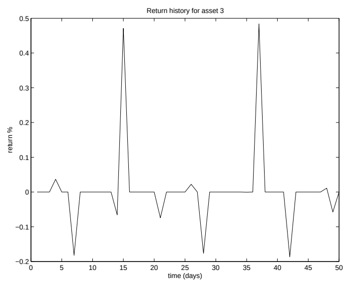

Figure 7: Return history for asset 3

To obtain a maximum-Ω portfolio from these assets we use $y_1 + y_2 + y_3 = 1$ and calculate daily returns as functions of y_1 and y_2 only, i.e.,

$$R_i = y_1 r_{1i} + y_2 r_{2i} + (1 - y_1 - y_2) r_{3i}, \quad i = 1, 2, .., 50.$$

In this expression r_{ji} denotes the observed return from asset j on day i. We can then generate a cumulative distribution from the $R_1, .., R_{50}$ and hence, for a specified return r_t, we can calculate the corresponding $\Omega(r_t)$. Obviously $\Omega(r_t)$ is also a function of the invested fractions and we want to choose y_1 and y_2 to maximize

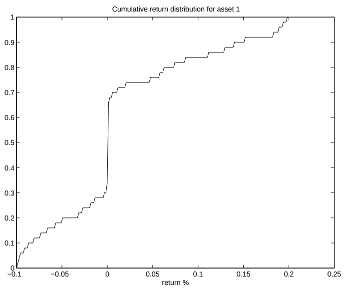

Figure 8: Cumulative distribution of returns for asset 1

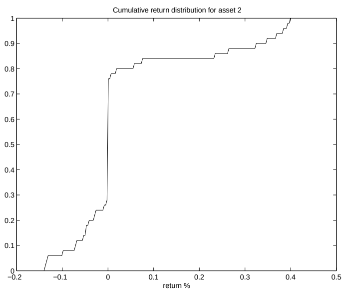

Figure 9: Cumulative distribution of returns for asset 2

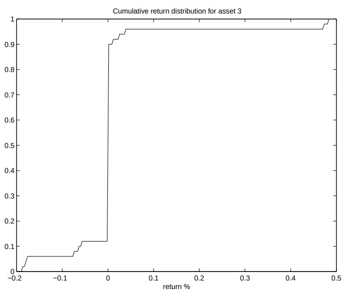

Figure 10: Cumulative distribution of returns for asset 3

it. We shall exclude solutions which involve short-selling and so we can apply a minimization algorithm to a penalty function such as

$$F(y_1,y_2) = -\Omega(r_t) + \rho\{[\min(0,y_1)]^2 + [\min(0,y_2)^2] + [\min(0,1-y_1-y_2)]^2\} \quad (2)$$

where ρ is a positive weighting parameter.

In the first instance, the optimization method that we choose is the Nelder and Mead Simplex algorithm [3] as implemented in the MATLAB procedure `fminsearch`. This direct search technique is a suitable method since the objective function F is non-differentiable w.r.t. the variables y_1, y_2.

Suppose first that we take the threshold value $r_t = 0.01$ and use $\rho = 2$ in (2). Figure 11 shows the contours of (2). (The jagged nature of these contours is a reflection of the fact that the cumulative distribution functions in Figures 8—10 are nonsmooth and hence the numerical integral involved in computing Ω does not change smoothly as the invested fractions change.)

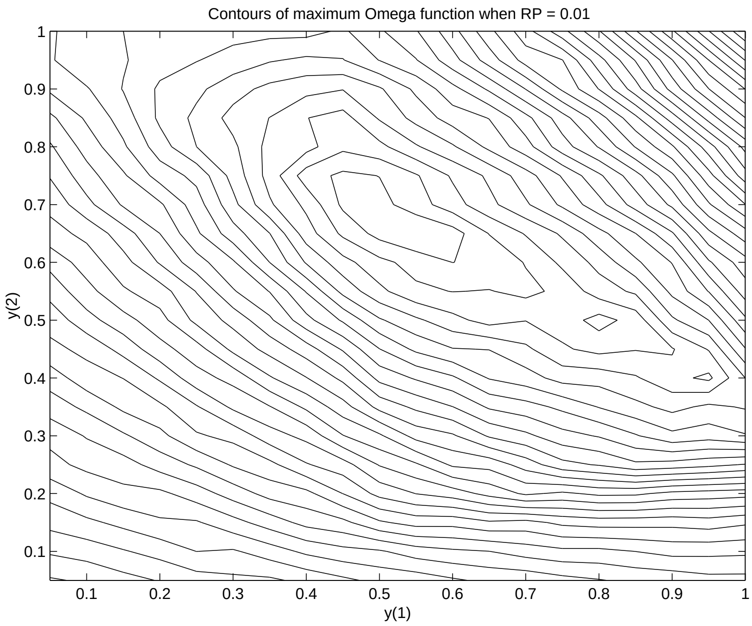

Figure 11: Contours of (2) when $r_t = 0.01$

Results from the simplex method are consistent with this figure because the optimum is found with $y_1 \approx 0.45,\ y_2 \approx 0.72$ giving $F \approx -2.89$. This is not a unique local optimum, however. Figure 11 suggests that there might be other local minima in the region of (0.8, 0.5) and (0.95, 0.4). These apparent optima are not artefacts of the contour-plotting process: there are genuine minima at $y_1 \approx 0.77,\ y_2 \approx 0.51$ (with $F \approx -2.78$) and at $y_1 \approx 0.95,\ y_2 \approx 0.4$ (with $F \approx -2.67$). This observation immediately makes the problem more challenging.

The simplex method is only a local optimization procedure and therefore, to seek a global optimum of F we could try a multistart process which essentially involves running `fminsearch` from a large number of randomly distributed starting guesses. Of course, there are other approaches to the global optimization problem and we shall consider one of them in a later section.

As well as showing the possible presence of multiple local solutions problem of maximizing Ω, the above example also shows that the choice $\rho = 2$ for the penalty

parameter is inadequate because it permits a substantial negative value for the third invested fraction $y_3 = 1 - y_1 - y_2 \approx -0.17$. To obtain a solution which is closer to being acceptable we use $\rho = 100$ and then, using the simplex method with multi-starts, we get three solutions which effectively set y_3 to zero, namely:
$y_1 \approx 0.5,\ y_2 \approx 0.5,\ \ F \approx -2.49$
$y_1 \approx 0.4,\ y_2 \approx 0.6,\ \ F \approx -2.52$
$y_1 \approx 0.45,\ y_2 \approx 0.55,\ \ F \approx -2.54$

Figure 12 shows the contour plot of (2) in the region of these local solutions and confirms the validity of the results.

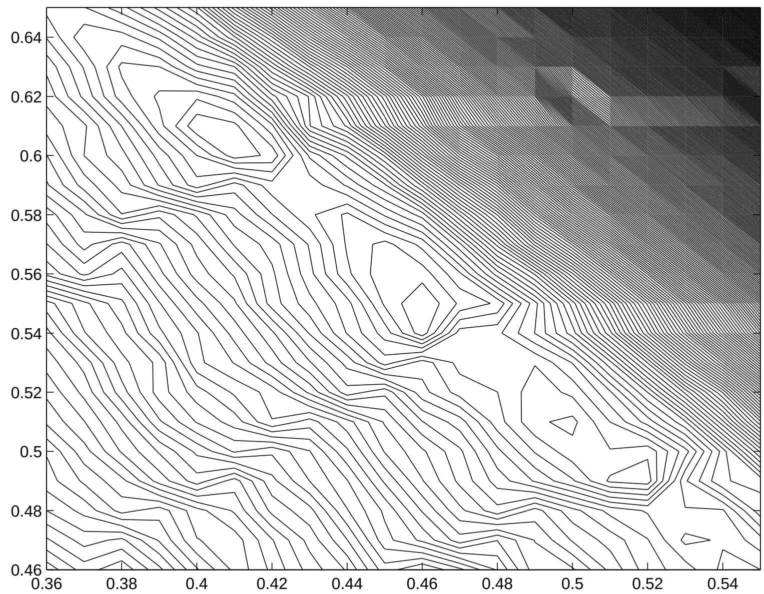

Figure 12: Contours of (2) for $0.35 \leq y_1 \leq 0.55, 0.45 \leq y_2 \leq 0.65$ when $r_t = 0.01$

The fact that the solution sets y_3 to zero is quite easily explained when we look at the shape of the cumulative distribution for the third asset in Figure 10. It is clear that the value of $\Omega(0.01)$ for this asset will be quite small compared with the corresponding values for assets 1 and 2 and so asset 3 is unlikely to contribute to the portfolio which yields the maximum value of Ω.

3 Using a multilevel algorithm to solve larger problems

We now want to consider larger problems involving a n-asset portfolios based on an m-day performance history. As before, we use the relation $\sum_j y_j = 1$ and express daily portfolio returns in terms of invested fractions as

$$R_i = \sum_{j=1}^{n-1} r_{ji} y_j + r_{jn}(1 - \sum_{k=1}^{n-1} y_k), \quad \text{for } i = 1,...,m.$$

We can calculate the cumulative distribution function for these R_i and then evaluate $\Omega(r_t)$ for a specified threshold return r_t. We shall calculate Max-Ω portfolios by finding $y_1,...,y_9$ to minimize an exact penalty function

$$E(y) = \frac{1}{\Omega(r_t)} + \rho \sum_{j=1}^{n-1} \min(0, y_j) + \rho \min(0, 1 - \sum_{k=1}^{n-1} y_k). \tag{3}$$

The fact that we use a nonmooth penalty term is of no great consequence since we already know the function Ω is nonsmooth.

In order to find a global optimum of the function (3) we shall use the the MCS algorithm of Huyer and Neumaier [5] which is described briefly in the next sub-section. This algorithm is implemented in the NAG library routine E05JBF [4]. Numerical tests with the MCS method have been carried out in MATLAB using a NAG-supplied interface to E05JBF.

3.1 The MCS algorithm and its E05JBF implementation

The MCS algorithm seeks a global minimum of an n-variable function $F(x)$ in a hyperbox defined by $l \leq x \leq u$. (In what follows we shall simply use the term *box* when strictly we mean an n-dimensional hyperbox.)

MCS searches for a global minimizer using branching recursively in order to divide the search space in a nonuniform manner. It divides, or *splits*, the root box $[l,\ u]$ into smaller sub-boxes. Each sub-box contains a basepoint at which the objective function is sampled. The splitting procedure biases the search in favour of sub-boxes where low function values are expected. The global part of the algorithm explores sub-boxes that enclose large unexplored territory, while the local part splits sub-boxes that have good function values.

A balance between the global and local parts of the method is achieved using a multilevel approach, where every sub-box is assigned a level $s \in \{0, 1, \ldots, s_{max}\}$. The value of s_{max} can be specified by a user. A sub-box with level 0 has already been split; a sub-box with level s_{max} will be split no further. Whenever a sub-box of intermediate level $0 < s < s_{max}$ is split a descendant will be given the level $s+1$ or $\min(s+2, s_{max})$. The child with the better function value is given the larger fraction of the splitting interval because then it is more likely to be split again more quickly.

An initialization procedure generates a preliminary set of sub-boxes, using points input by the user or derived using a default generation procedure. The method ranks each coordinate based on an estimated variability of the objective function, computed by generating quadratic interpolants through the points used in the initialization. Then the algorithm begins sweeping through levels.

Each sweep starts with the sub-boxes at the lowest level, this process being the global part of the algorithm. At each level the sub-box with the best function value is selected for splitting; this forms the local part of the algorithm. A box is split either by *rank* (when it reaches a sufficiently high level; in particular, as $s_{max} \to \infty$ this ensures each coordinate is split arbitrarily often) or by *expected gain* (along a coordinate where a maximal gain in function value is expected, again computed by fitting quadratics).

The splitting procedure as a whole is a variant of the standard coordinate search method: MCS splits along a single coordinate at a time, at adaptively chosen points. In most cases one new function evaluation is needed to split a sub-box into two or even three children. Each child is given a basepoint chosen to differ from the basepoint of the parent in at most one coordinate, and safeguards are present to ensure a degree of symmetry in the splits. If the optional parameter **Local Searches** is set to 'OFF', then MCS puts the basepoints and function values of sub-boxes of maximum level s_{max} into a 'shopping basket' of candidate minima. Turning **Local Searches** 'ON' (the default setting) will enable local searches to be started from these basepoints before they go into the shopping basket.

Local searches go ahead providing the basepoint is not likely to be in the basin of attraction of a previously-found local minimum. The search itself uses linesearches along directions that are determined by minimizing quadratic models, all subject to bound constraints. In particular, triples of vectors are computed using coordinate searches based on linesearches. These triples are used in *triple search* procedures to build local quadratic models for the objective, which are then minimized using a trust-region-type approach. The quadratic model need not be positive definite, so it is minimized using a general nonlinear optimizer.

3.2 Numerical experiments with E05JBF

We now consider a ten-asset portfolio based on a 100-day history. We can summarise the properties of the ten assets by showing their mean returns and displaying plots of their Ω functions. The mean returns are

$$\overline{r}_1 = 0.078\%,\ \overline{r}_2 = -0.066\%,\ \overline{r}_3 = 0.01\%,\ \overline{r}_4 = 0.048\%,\ \overline{r}_5 = -0.07\%$$

$$\overline{r}_6 = 0.044\%,\ \overline{r}_7 = 0.003\%,\ \overline{r}_8 = 0.02\%,\ \overline{r}_9 = 0.040\%,\ \overline{r}_{10} = -0.045\%.$$

For clarity we show the Ω-plots in groups of four. Figure 13 shows the four highest Ω curves. Asset 1 gives consistently the best Ω value for threshold returns in the range $0 \le r_t \le 0.1$. Assets 9, 6 and 4 are the next best performers in terms of Ω and they remain consistently ranked in this order although the curves for assets 6 and 4 touch near $r_t = 0.04\%$ and the curves for assets 6 and 9 touch near $r_t = 0.03\%$.

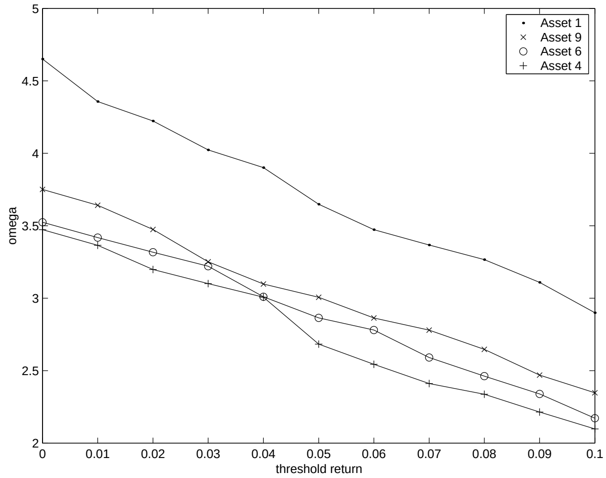

Figure 13: Ω plots for example assets 1,9,6 and 4

Figure 14 shows Ω plots for Assets 4,7,3 and 8. There is some changing of ranking between these assets as r_t increases. Thus, for instance, asset 8 is the worst performer when r_t is near-zero but is almost equal to asset 7 in being the best performer of this group as r_t approaches 0.1%.

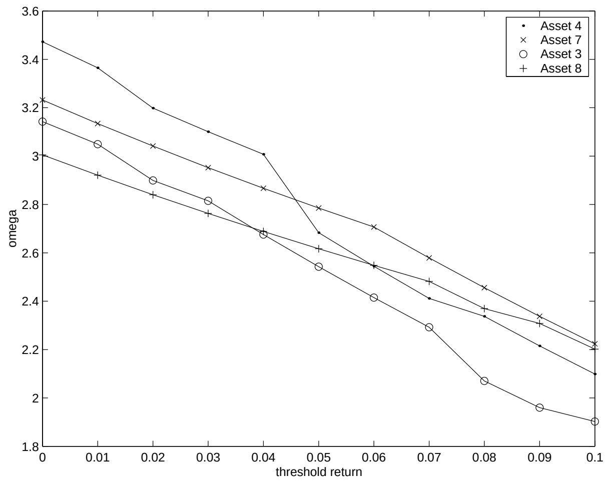

Figure 14: Ω plots for example assets 4,7,3 and 8

Finally, Figure 15 shows that assets 2,10 and 5 are well below asset 8. Assets 2 and 10 change places once or twice over the range of r_t but asset 5 is uniformly the worst choice.

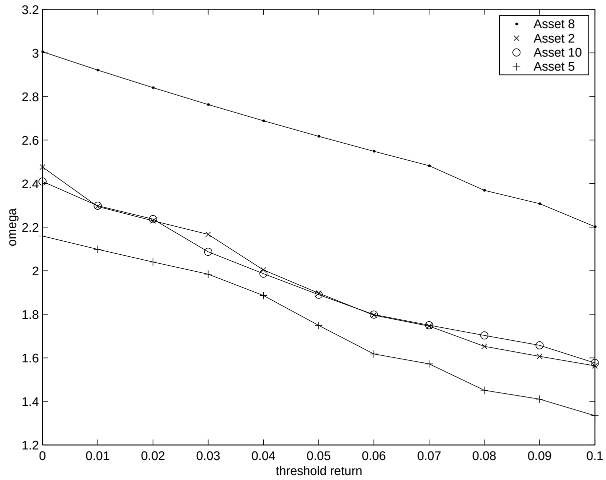

Figure 15: Ω plots for example assets 8,2,10 and 5

Table 1 shows Max-Ω portfolios obtained by applying E05JBF to the function (3) formed using data from our sample set of 10-assets and with various values for threshold return r_t. We seek the global minimum of (3) in the search box defined by $0 \leq y_i \leq 1$ for $i = 1, \ldots 9$. In the column for invested fractions we show only those which are non-zero.

r_t	Invested fractions	Ω_{max}
0%	$y_1 = 0.147,\ y_2 = 0.040,\ y_3 = 0.213,\ y_4 = 0.208$ $y_6 = 0.286,\ y_7 = 0.007,\ y_9 = 0.099$	7.08
0.025%	$y_1 = 0.118,\ y_2 = 0.036,\ y_4 = 0.225,\ y_6 = 0.361$ $y_7 = 0.003,\ y_9 = 0.219$	5.16
0.05%	$y_1 = 0.235,\ y_4 = 0.099,\ y_6 = 0.357,\ y_7 = 0.051$ $y_9 = 0.21,\ y_{10} = 0.046$	4.11
0.075%	$y_1 = 0.884,\ y_5 = 0.035,\ y_7 = 0.039,\ y_9 = 0.043$	3.48
0.1%	$y_1 = 0.977,\ y_5 = 0.014,\ y_8 = 0.009$	2.97

Table 1: Max-Ω portfolios for varying r_t

For the smaller values of r_t the solutions generally favour assets 1,4,6 and 9 which is consistent with the relative positions of their Ω curves in the graphs in Figures 13 – 15. For $r_t = 0.075\%$ and $r_t = 0.1\%$ the portfolios are strongly dominated by asset 1.

Table 2 gives more information about the Max-Ω solutions in terms of their portfolio expected return R_p and risk V.

r_t	R_p	V
0%	0.038%	0.058
0.025%	0.040%	0.064
0.05%	0.045%	0.061
0.075%	0.068%	0.211
0.1%	0.075%	0.261

Table 2: Expected return and risk for Max-Ω portfolios

We can compare Max-Ω portfolios with more conventional minimum-risk ones (denoted by Min-V) if we calculate the Min-V portfolio corresponding to the same expected portfolio returns R_p as achieved by the Max-Ω solutions in Table 2. Results are shown in Table 3. Clearly the invested fractions y_i are substantially different from those in Table 1. The Min-V solutions tend to use non-zero contributions from more of the assets; and almost the only point of similarity is the growing dominance of asset 1 as R_p increases. Comparing the values V_{min} with the values of V in Table 2 we can see that, for a given level of portfolio return, the Max-Ω are appreciably more risky than Min-V ones with the ratio

$$\frac{\text{risk of a Max-}\Omega\text{ portfolio}}{\text{risk of a Min-}V\text{ portfolio}}$$

lying between 1.4 and 1.7.

R_p	Invested fractions	V_{min}
0.038%	$y_1 = 0.201,\ y_3 = 0.069,\ y_4 = 0.115,\ y_6 = 0.171$ $y_7 = 0.122,\ y_8 = 0.09,\ y_9 = 0.191,\ y_{10} = 0.039$	0.035
0.040%	$y_1 = 0.214,\ y_3 = 0.065,\ y_4 = 0.118,\ y_6 = 0.173$ $y_7 = 0.116,\ y_8 = 0.095,\ y_9 = 0.192,\ y_{10} = 0.025$	0.037
0.045%	$y_1 = 0.260,\ y_3 = 0.049,\ y_4 = 0.133,\ y_6 = 0.174$ $y_7 = 0.101,\ y_8 = 0.091,\ y_9 = 0.193$	0.043
0.068%	$y_1 = 0.740,\ y_3 = 0.015,\ y_4 = 0.08,\ y_6 = 0.085$ $y_8 = 0.035,\ y_9 = 0.046$	0.156
0.075%	$y_1 = 0.817,\ y_4 = 0.027,\ y_6 = 0.085$ $y_7 = 0.029,\ y_8 = 0.044,\ y_9 = 0.083$	0.184

Table 3: Min-V portfolios giving same expected return as Max-Ω portfolios

It is interesting to compare the cumulative distributions of returns for Max-Ω and Min-V portfolios. Figure 16 shows the two curves for the case when $R_p = 0.038\%$. (Figures for other values of R_p are generally similar to this one.) Both curves are quite similar in the central part but the dotted curve for the Min-V portfolio has a

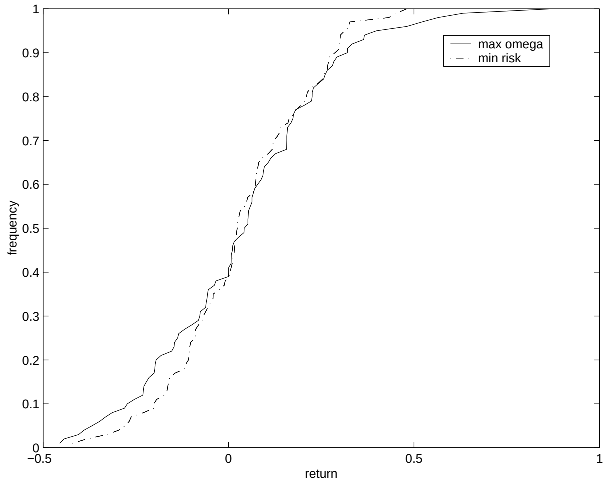

Figure 16: Comparing Max-Ω and Min-V for $R_p = 0.038\%$

shorter tail towards the upper end of the range of observed returns. The longer tail of the solid curve for the Max-Ω solution reflects the fact that it seeks to adjust the areas above and below the curve defined by (1) in order to maximize the ratio Ω.

A Min-V portfolio is calculated to minimize deviations both above and below expected portfolio return R_p. A Max-Ω solution, on the other hand, is concerned with increasing the chances that a portfolio will obtain *more than* the threshold return r_t. Hence a Max-Ω portfolio might be expected to have more in common with one which is designed to minimize *downside* risk – that is the risk of obtaining less than target return. Table 4 shows the portfolios obtained by minimizing downside risk DV for the values of R_p in Tables 2 and 3. Table 4 quotes the value DV_Ω of downside risk at the corresponding Max-Ω solution. The invested fraction distri-

R_p	Invested fractions	DV_{min}	DV_Ω
0.038%	$y_1 = 0.193$, $y_2 = 017$, $y_3 = 0.062$, $y_4 = 0.150$, $y_6 = 0.172$ $y_7 = 0.119$, $y_8 = 0.083$, $y_9 = 0.18$, $y_{10} = 0.023$	0.017	0.027
0.040%	$y_1 = 0.204$, $y_2 = 0.006$, $y_3 = 0.065$, $y_4 = 0.155$, $y_6 = 0.172$ $y_7 = 0.118$, $y_8 = 0.083$, $y_9 = 0.18$, $y_{10} = 0.017$	0.018	0.029
0.045%	$y_1 = 0.253$, $y_3 = 0.064$, $y_4 = 0.165$, $y_6 = 0.17$ $y_7 = 0.096$, $y_8 = 0.073$, $y_9 = 0.178$	0.021	0.031
0.068%	$y_1 = 0.782$, $y_4 = 0.148$, $y_6 = 0.008$ $y_8 = 0.015$, $y_9 = 0.021$, $y_{10} = 0.026$	0.09	0.11
0.075%	$y_1 = 0.932$, $y_2 = 0.001$, $y_4 = 0.017$, $y_6 = 0.048$	0.13	0.14

Table 4: Min-DV portfolios giving same expected return as Max-Ω portfolios

bution in the Min-*DV* portfolios bears more resemblance to that of the Min-*V* ones than to that of the Max-*Omega* solutions. It is worth noting, however, that for the larger values of R_p the Max-Ω portfolios have a downside risk that is not very much inferior to $DV - min$.

3.3 Comments on the performance of E05JBF

The global optimization algorithm implemented in E05JBF has performed quite successfully on the problems we have considered so far. It has certainly been considerably more efficient than the rather crude multistart approach which we tried initially in which the simplex method in `fminsearch` was applied from 50-100 random starting points. This procedure was reasonably useful for the demonstration three-variable example in the previous section; but for the ten-variable case it was very time-consuming and seldom yielded as good an estimate of the global solution as E05JBF.

Notwithstanding these positive comments, however, it must be noted that we have had to do some trial-and-error parameter-tuning in order to obtain satisfactory results. E05JBF requires a user to make a number of choices, listed below.

• Selection of initial points. These are values which are used by the algorithm in choosing points along the coordinate axes at which to split the original box. These points may be user-defined; but there are a number of default options for selecting them automatically. We had most success with the default approach in which initial splitting points are selected on the basis of local searches along each coordinate axis.

• Balance between global and local searching. The user can specify how many iterations of a local minimization method are to be used to refine each point that is identified by the splitting procedure as being a candidate global solution. These local searches use a trust-region approach based on quadratic interpolation. While this is likely to be a good approach in general, it may not be very suitable for our application since the objective function (3) is nonsmooth. Consequently we have chosen to use rather few local search iterations compared with the suggested default. We have sought to compensate for this by increasing the rigour of the global search and setting the number of splitting points per variable to be about twice the suggested default value.

• Setting termination conditions. The main tests for successful termination of E05JBF are based either on the search reaching a pre-specified target function value or on there being no decrease in the best function value for a pre-specified number of sweeps (the *static limit*). Furthermore E05JBF may terminate unsuccessfully – i.e. with an error flag – if a specified number of function evaluations or box splits is exceeded. We found that the suggested value for the static limit ($3n$ sweeps) was often too small and led to the method stopping at points well short of the global solution. We had to increase this to at least $5n$ to obtain acceptable solutions consistently. Basing successful termination on reaching a target function value is a more reliable way of ensuring that a satisfactory stopping point is reached: but often it is not possible to know in advance what the global minimum fnction is likely to be. We were largely able to avoid unsuccessful terminations by taking the maximum number of splits as being 120 and the maximum number of function calls as 15000.

Because of the under-use of local searches mentioned above it is unlikely that best point returned by E05JBF will give the global optimum to high precision. We have confirmed this by running the MATLAB implementation of the Nelder and Mead simplex method (`fminsearch`) from the best point found by the MCS algorithm and observing that it is usually able to obtain a small further reduction in function value. It is these refined estimates of global solutions quoted in Table 1.

4 Discussion

We have given a preliminary account of an investigation of the use of the Omega ratio [1] as a performance measure for portfolio selection by seeking to choose invested fractions which maximize Ω as defined in (1). We have considered some Max-Ω portfolios produced using E05JBF and compared them with portfolios produced using the well-known Markowitz approach. Our small-scale sample calculations suggest that, for a given set of assets, a Max-Ω portfolio can be quite different from portfolios based on minimizing risk or downside risk. The practical differences between Max-Ω and minimum-risk portfolios need to be investigated further using back-testing and real-life asset data.

Maximizing Ω has proved to involve a non-convex and nonsmooth function. We have tackled this problem using the MCS algorithm of Huyer & Neumaier [5] as implemented in the NAG routine E05JBF.

In view of the non-smoothness of the Ω function it might be worth comparing the performance of the MCS algorithm (which uses quadratic interpolation and hence assumes smoothness) with that of the DIRECT method proposed by Jones *et al.* [6]. DIRECT is also a box-splitting approach which does not use quadratic models and hence might be more suitable for our nonsmooth problem. The issue of nonsmoothness of Ω may be somewhat alleviated if we deal with assets for which a long performance history is available this will make the cumulative density functions appear less jagged.

References

[1] A. Cascon, C. Keating and W. Shadwick, ”The Omega Function”, The Finance Development Centre, London, 2002.

[2] A. Favre-Bulle and S. Pache, ”The Omega Measure: Hedge Fund Portfolio Optimization”, MBF Master’s Thesis, University of Lausanne - Ecole des HEC, 2003.

[3] J.A. Nelder and R. Mead, ”A simplex method for function minimization”, Computer Journal, **7** (1965), 308-313.

[4] NAG Ltd - Private Communication

[5] W. Huyer and A. Neumaier, ”MCS: Global optimization by multilevel coordinate search”, J. Global Optimization **14** (1999), 331–355.

[6] D.R.Jones, C.D. POerttunen and B.E.Stuckman, ”Lipschitzian Optimization without the Lipschitz Constant”, J. Opt Theory Applics, **79** (1993) pp 157-181.