Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 3945–3961
computers & operations research
www.elsevier.com/locate/cor

Efficient implementation of an active set algorithm for large-scale portfolio selection

Michael Stein*, Jürgen Branke, Hartmut Schmeck

Institute for Applied Informatics and Formal Description Methods (AIFB), University of Karlsruhe (TH), Germany

Available online 2 June 2007

Abstract

This paper deals with the efficient implementation of parametric quadratic programming that is specialized for large-scale mean-variance portfolio selection with a dense covariance matrix. The aim is to calculate the whole Pareto front of solutions that represent the trade-off between maximizing expected return and minimizing variance of return.

We describe and compare in a uniform framework several techniques to speed up the necessary matrix operations, namely the initial matrix decomposition, the solution process in each iteration, and the matrix updates. Techniques considered include appropriate ordering of the matrix rows and columns, reducing the size of the system of linear equations, and dividing the system into two parts. Regarding implementation, we suggest to simultaneously use two different matrix representations that are specifically adapted to certain parts of the algorithm and propose a technique that prevents algorithm stalling due to numerical errors. Finally, we analyse and compare the runtime of these algorithm variants on a set of benchmark problems. As we demonstrate, the most sophisticated variant is several orders of magnitude faster than the standard implementation on all tested problem instances.

Keywords: Parametric quadratic programming; Mean-variance portfolio selection; Dense covariance matrix; Efficient implementation

1. Introduction

Classical portfolio optimization aims at both maximizing the expected return of a portfolio and minimizing its variance. It has been shown that under the assumption of multivariate normally distributed asset returns or, in the case of arbitrary returns, a quadratic utility function (see Markowitz [1–3] and especially [4]), the optimal portfolio for the investor lies on the mean-variance Pareto front or—as it is often called—the *Efficient Frontier*. A portfolio is efficient if no other feasible portfolio exists that either improves both optimization criteria or improves just one but does not worsen the other. The Efficient Frontier consists of all efficient portfolios.

In the absence of any information about an investor's risk aversion, or when the investor wants to take a look at all "interesting" portfolios, an investment consultant needs to be able to provide the complete Pareto front. This might also be required if there is a large group of investors with similar constraints but different risk-return preferences. Algorithms for the calculation of all portfolios on the Pareto front belong to the category of parametric quadratic programming

* Corresponding author.
E-mail addresses: stein@aifb.uni-karlsruhe.de (M. Stein), branke@aifb.uni-karlsruhe.de (J. Branke), schmeck@aifb.uni-karlsruhe.de (H. Schmeck).

0305-0548/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2007.05.004

algorithms—or shorter: PQP algorithms—and are presented in several publications, the first of them was the so-called *critical line algorithm* mentioned in the seminal work by Markowitz [2].

To our knowledge, all these algorithms that are able to calculate the whole Efficient Frontier assume that the search space is convex, usually by requiring that all constraints are linear in nature. The reason for our interest in the efficiency of the PQP algorithms is based on research in constraints that do not fit into the linear framework, and therefore, PQP algorithms are not applicable directly. However, by selecting different convex subsets in the nonconvex search space, we are able to use a PQP algorithm to calculate a Pareto front for each subset. These Pareto fronts are then merged into a solution for the problem with nonconvex restrictions. For a more detailed description on how the convex subsets are chosen and on how a PQP algorithm can be integrated into an evolutionary algorithm framework, the reader is referred to Branke et al. [5]. For such applications, efficiency is particularly crucial, as the PQP algorithm has to be executed multiple times, and a faster PQP algorithm therefore permits a larger number of convex subsets to be calculated.

Only very few publications that describe algorithms for PQP give any hints on how to implement these algorithms in an efficient and numerically stable way for large portfolio sizes, which is our main focus in the remainder of this paper.

The subsequent sections of the paper are organized as follows: Section 2 presents the algorithm framework that is the basis for our implementation. In Section 3 we compare other existing approaches for parametric and nonparametric quadratic programming. Based on our framework from Section 2 several modifications that are intended to shorten algorithm runtime are described in Section 4. The test results for these algorithm variants are presented in Section 5. In Section 6 two different matrix representations are presented, and we show how both are incorporated into the implementation of the algorithm variant with the best runtime properties. We also highlight one implementation detail that is crucial to achieve correct solutions. We finish with a short conclusion in Section 7.

2. An algorithm for PQP

We have used a modified version of the active set algorithm for PQP presented by Best [6] and have adapted it for portfolio selection. In order to introduce all the variables and the used methodology, we summarize the algorithm in a short way and refer the reader to Best [6] for a more detailed description. The implemented algorithm solves the following problem for the parameter λ_e in the interval $[0, +\infty)$:

$$\min\{\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} - \lambda_e\boldsymbol{\mu}^T\boldsymbol{x} \mid \boldsymbol{A}_I\boldsymbol{x} \leqslant \boldsymbol{b}_I,\ \boldsymbol{A}_E\boldsymbol{x} = \boldsymbol{b}_E\} \tag{1}$$

with the element x_i of the vector $\boldsymbol{x}$ denoting the fraction of the budget invested in asset i. $\boldsymbol{C}$ is the covariance matrix, $\boldsymbol{\mu}$ denotes the vector of expected returns of all assets. $\boldsymbol{A}_I$ and $\boldsymbol{A}_E$ are the coefficient matrices of inequalities and equalities; $\boldsymbol{b}_I$ and $\boldsymbol{b}_E$ denote the respective right-hand sides.

To start the parametric programming routine, at least one portfolio on the Pareto front has to be known. Due to the fact that it is easier and computationally cheaper to solve an optimization problem with a linear objective function instead of a quadratic objective function, our PQP algorithm starts at the portfolio with the highest possible expected return, which is the solution of the following optimization problem:

$$\max\{\boldsymbol{\mu}^T\boldsymbol{x} \mid \boldsymbol{A}_I\boldsymbol{x} \leqslant \boldsymbol{b}_I,\ \boldsymbol{A}_E\boldsymbol{x} = \boldsymbol{b}_E\}. \tag{2}$$

If this solution is unique, as is nearly always the case for "normal" portfolio selection problems, the portfolio lies at the end of the Efficient Frontier that is associated with the highest λ_e.

Otherwise, there are infinitely many portfolios that achieve the highest possible expected return, and it is necessary to select from all these solutions to problem (2) the portfolio with the lowest variance:

$$\min\{\boldsymbol{x}^T\boldsymbol{C}\boldsymbol{x} \mid \boldsymbol{\mu}^T\boldsymbol{x} = E_m,\ \boldsymbol{A}_I\boldsymbol{x} \leqslant \boldsymbol{b}_I,\ \boldsymbol{A}_E\boldsymbol{x} = \boldsymbol{b}_E\}. \tag{3}$$

Here, E_m denotes the maximum expected return calculated in (2). This is a quadratic programming problem that can be solved with any of the available standard codes or packages for this problem class (see e.g. the NEOS Optimization Software Guide [7]).

If not all portfolio weights have an upper and lower bound, it is possible that the expected return is unbounded and therefore a solution to problem (2) does not exist. In this case it is necessary to compute the minimum variance