

2021

Constrained Portfolio Optimisation

Moh Tahir, M.A.

Moh Tahir, M.A. (2021) 'Constrained Portfolio Optimisation', The Plymouth Student Scientist, 14(2), pp. 429-464.

<http://hdl.handle.net/10026.1/18510>

The Plymouth Student Scientist
University of Plymouth

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Constrained Portfolio Optimisation

Mohammad Alam Tahir

Project Advisor: Dr. Matthew Craven, School of Computing, Electronics, and Mathematics, University of Plymouth, Drake Circus, Plymouth, PL4 8AA

Abstract

Portfolio optimisation is an important problem in finance; it allows investors to manage their investments effectively. This paper considers finding the efficient frontier associated with the mean-variance portfolio optimisation (unconstrained) problem. We then extend the mean-variance model to include cardinality constraints (resulting in an NP-Hard problem) that limits the number of assets in a portfolio. We discuss different types of algorithms that one can use for finding the optimal portfolios, implementing a meta-heuristic genetic algorithm technique to solve the unconstrained and cardinality constrained problems. Finally, we improve our solutions by altering the crossover and mutation probabilities in the genetic algorithm method. For finding the efficient frontier associated with both problems, we examine a dataset involving 55 assets from the US stock exchange.

Keywords: Mean-variance portfolio optimisation, cardinality constrained portfolio optimization, NP-Hard problem, quadratic programming, meta-heuristic, genetic algorithm.

Introduction

In financial mathematics, portfolio optimisation is an important topic for investment management techniques. It helps investors select optimum portfolios, given the level of return they want with respect to a desired level of risk. Markowitz (1952, 1959) introduces the standard mean-variance approach to portfolio selection. It involves tracing out an efficient frontier, a continuous curve illustrating the trade-off between the expected return and risk (variance) [6].

The standard Markowitz mean-variance approach assumes that asset returns follow a multivariate normal distribution [6]. This means the return on our portfolio assets can be expressed by the expected return and the variance of the returns. Markowitz mean-variance is the unconstrained part of the portfolio optimisation problem. In this case, as we will see later, the objective function is quadratic, the constraints are linear, and the efficient frontier can be easily found via quadratic programming.

Adding more constraints to the problem makes it difficult to trace the efficient frontier. In this paper, we will look at the cardinality constrained problem, and in this case, the objective function is not quadratic. It allows us to restrict the number of assets in a portfolio. As we decrease the number of assets, it becomes more difficult to trace the efficient frontier, as there will be fewer assets to use to minimize risk to the same level as previously. As a result, heuristic methods have been introduced, where finding the optimum solution is not guaranteed. However, heuristic methods are efficient in finding the optimum solution or near-optimal solution in a reasonable amount of time. Therefore, we will find the efficient frontier for the cardinality constraints problem via a meta-heuristic algorithm (genetic algorithm).

The genetic algorithm and quadratic programming techniques are not the only two methods to find optimal portfolios. There are several procedures proposed in the literature for the mean-variance and cardinality constrained portfolio optimisation problem. Mansini and Speranza present three different heuristic algorithms to find the optimal portfolios in a portfolio optimisation problem with round lot constraint [20]. Chang et al. present three meta-heuristic algorithms. There are based upon a genetic algorithm, tabu search, and simulated annealing for finding the cardinality constrained efficient frontier, involving up to 225 assets [6]. Streichert et al. used evolutionary algorithms (e.g. GA). They compared the results on the constrained, and unconstrained portfolio optimisation problem [24]. Cura presents a population-based meta-heuristic algorithm using particle swarm optimisation technique for cardinality constrained mean-variance model. Cura compared the results with other methods such as genetic algorithms, simulated annealing and tabu search [8]. Deng et al. used an improved particle swarm optimisation technique for cardinality constrained problem. They showed most of the time, the particle swarm optimisation technique outperformed the genetic algorithm, simulated annealing, and tabu search [10]. There also exist many studies which applied a multi-objective evolutionary approach to the portfolio optimisation problem. In multi-objective portfolio optimisation, we maximise the return and minimise the risk at the same time. Diosan used Pareto Archived Evolution Strategy (PESA), Non-dominated Sorting Genetic Algorithm (NSGA II) and Strength Pareto Evolutionary Algorithm (SPEA 2) for solving the bi-objective portfolio optimisation problem [11]. In another study, Lwin et al. used a hybrid multi-objective evolutionary algorithm for cardinality, quantity, pre-assignment and round lot constraints. They showed the hybrid multi-objective evolutionary algorithm significantly outperforms PESA, NSGA II, and

SPEA 2 algorithms [18].

Many publications had discussed solving portfolio optimisation problems with exact algorithms. The exact algorithms aim to give us exact solutions compared to heuristic algorithms. However, it will take more time to find the optimum solution. Exact solution methodologies include Jobst, for the cardinality constrained portfolio selection problem, using a quadratic programming-based branch-and-bound approach [16]. Lejeune and Bonami used a nonlinear branch and bound algorithm to find the optimal portfolios with various real-world constraints such as buying stocks by lots [4]. Bienstock applied a branch-and-cut algorithm for the cardinality constrained mean-variance model with side constraints [3]. There are also other examples of exact methods, such as Bertsimas and Cory-Wright. They used cutting-plane methods for solving portfolio selection problem [2].

Portfolio optimisation is a complex problem, and it can not be easily solved numerically. So, we will be using R programming language software to do the complex calculations throughout this paper.

Efficient Frontier

The efficient frontier is a two-dimensional curve representing the set of optimal portfolios with the minimum risk for a given level of expected return. In other words, the efficient frontier comprises portfolios that offer the highest expected return for a specific level of risk. As we add more constraints to the standard mean-variance portfolio optimisation problem, various shapes of the efficient frontier will appear. For the unconstrained problem in which the objective function is quadratic, we will have a quadratic curve for the efficient frontier. However, we do not know where the position of the curve is in the risk and expected return space. The correlation between assets will roughly tell us this. On the other hand, the cardinality constraint's objective function is not quadratic, so that the efficient frontier will have varied shapes.

The efficient frontier for the constrained problem might be a discontinuous curve depending on the objective function of the problem. If we are dealing with a single objective function, the discontinuities will appear in the efficient frontier. However, if we are dealing with multi-objective functions, discontinuity will not emerge in the efficient frontier. In this paper, we are only dealing with the single objective function, and we will see a few examples where discontinuities appear in the efficient frontier. This is a consequence of a single-objective optimisation approach.

Mean Variance Portfolio optimisation

Formulation of unconstrained problem

Now we can formulate the mean-variance portfolio optimisation problem and then use the R computer programming language to solve the problem.

Notations: Let:

n be the number of assets.

w_i be the proportion held in asset i ($i = 1, 2, \dots, n$)

R be the expected rate return of asset i ($i = 1, 2, \dots, n$)

R_0 be the target return on our investment.

σ_{ij} be the co-variance of assets i and j
 $(i, j = 1, 2, \dots, n)$

μ_i be the expected return on our portfolio.

The unconstrained portfolio optimisation problem is as follows [6]:

$$\text{Minimise } \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \quad (1)$$

subject to:

$$\sum_{i=1}^n w_i = 1 \quad (2)$$

$$R = \sum_{i=1}^n w_i \mu_i \quad (3)$$

where $0 \leq w_i \leq 1$, $(i = 1, 2, \dots, n)$, and $R_i \geq R_0$.

Equation 1 is the variance formula, and it means to minimise the total risk that an investor is exposed to on their portfolio. Equation 2 means the proportions of investment in i asset is equal to one. Finally, equation 3 is the expected return on our portfolio.

Further analysis

The proportion of assets that we invest in must equal to one.

$$\sum_{i=1}^n w_i = 1$$

where w are the weights.

The collection of weights in vector form:

$$(w_1, w_2, \dots, w_n)^T$$

The return on the individual asset in a portfolio is equal to the proportion of the portfolio invested in an asset multiple the rate of return on a our investment for the individual asset. Thus, the total return on a portfolio is simply the sum of returns on individual assets in the portfolio.

$$R_1 = \sum_{i=1}^n w_i \tau_i$$

Where τ_i is the rate of return for asset i .

Calculating the mean and variance of the portfolio return:

$$R_1 = \sum_{i=1}^n \tau_i w_i = \tau_1 w_1 + \tau_2 w_2 + \dots + \tau_n w_n$$

Using the expectation properties:

$$E(R_1) = w_1 E(\tau_1) + w_2 E(\tau_2) + \dots + w_n E(\tau_n)$$

This means that:

$$R = \sum_{i=1}^n w_i \mu_i$$

$$R = \sum_{i=1}^n w_i \mu_i \tag{3}$$

Using the definition of variance:

$$\begin{aligned} \text{Var}(R_1) &= E[(R_1 - R)^2] = E \left[\left(\sum_{i=1}^n w_i \tau_i - \sum_{i=1}^n w_i \mu_i \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^n w_i (\tau_i - \mu_i) \right)^2 \right] \\ &= E \left[\left(\sum_{i=1}^n w_i (\tau_i - \mu_i) \right) \left(\sum_{j=1}^n w_j (\tau_j - \mu_j) \right) \right] \\ &= E \left[\sum_{i=1}^n \sum_{j=1}^n w_i w_j (\tau_i - \mu_i) (\tau_j - \mu_j) \right] \\ &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j E [(\tau_i - \mu_i) (\tau_j - \mu_j)] \\ &= \sum_{i=1}^n \sum_{j=1}^n w_i w_j \text{Cov}(\tau_i, \tau_j) \end{aligned}$$

This means that risk(σ^2) is equal to:

$$\sigma^2 = \sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \tag{1}$$

Co-variance is a statistical measure used to analyse the relationships between two variables: the relationships between assets i and j . Note that $\text{Cov}(r_i, r_i) = \text{var}(r_i)$. The variance co-variance matrix Σ is

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22}^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \dots & \sigma_n^2 \end{bmatrix}.$$

We can write the diagonal and off-diagonal terms in the co-variance matrix separately as follows:

$$\sigma^2 = \sum_{i=1}^n w_i^2 \sigma_i^2 + \sum_{i=1}^n \sum_{j=1, i \neq j}^n w_i w_j \sigma_{ij} \tag{4}$$

Next we will look at the strength of the relationships between assets i and j .

Correlation between assets

The material in this section have been adopted from [12]

Correlation is a statistical measure that determines how assets move in relation to each other. We can use the correlation to determine the relationship between the returns of two assets and then use that to determine the efficient frontier curve in risk and expected return space.

The correlation coefficients, which we use to measure correlation, range between -1 and +1 and we will denote the correlation coefficients between two assets i and j as ρ_{ij} .

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

where σ_i and σ_j are the standard deviation asset i and asset j respectively.

Now let us investigate coefficients between two assets, say m and n . Then, the expected return is:

$$R = w_m \mu_m + w_n \mu_n$$

The proportion in asset m and n must be equal to one:

$$w_m + w_n = 1$$

From equation (4), the formula for the risk becomes:

$$\begin{aligned} \sigma_{mn}^2 &= w_m^2 \sigma_m^2 + w_n^2 \sigma_n^2 + 2w_m w_n \rho_{mn} \sigma_m \sigma_n \\ &= w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2 + 2w_m (1 - w_m) \rho_{mn} \sigma_m \sigma_n \end{aligned} \quad (5)$$

Equation (5) is the risk formula for asset m and n .

Perfect Positive Correlation

A correlation of +1 suggest that the two assets have a perfect positive correlation. This means the expected return on two assets m and n move in the same direction.

When $\rho_{mn} = +1$, the equation (5) for the risk on the portfolio become:

$$\begin{aligned} \sigma_{mn}^2 &= w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2 + 2w_m (1 - w_m) \sigma_m \sigma_n \\ \sigma_{mn} &= [w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2 + 2w_m (1 - w_m) \sigma_m \sigma_n]^{1/2} \end{aligned}$$

The term in the bracket is a quadratic function and so

$$\sigma_{mn} = w_m\sigma_m + (1 - w_m)\sigma_n.$$

Also, the expected return on our portfolio is:

$$R_{mn} = w_m\mu_m + (1 - w_m)\mu_n$$

Solving w_m in the function of standard deviation (σ_{mn}) will give us:

$$w_m = \frac{\sigma_{mn} - \sigma_n}{\sigma_m - \sigma_n}$$

Substituting w_m into the expression for R gives us the equation for a straight line (see Figure 1):

$$R_{mn} = \frac{\sigma_{mn} - \sigma_n}{\sigma_m - \sigma_n}\mu_m + \left(1 - \frac{\sigma_{mn} - \sigma_n}{\sigma_m - \sigma_n}\right)\mu_n$$

When the correlation +1 the investment opportunity set a straight line and this means that there is very little room for diversification.

Perfect Negative Correlation

A correlation of -1 suggests that the two assets have a perfect negative correlation. This means the expected return on two assets m and n move in the opposite direction to each other. That is, when $\rho_{mn} = -1$, the equation (5) for the risk on the portfolio become:

$$\sigma_{mn}^2 = w_m^2\sigma_m^2(1 - w_m)^2\sigma_n^2 - 2w_m(1 - w_m)^2\sigma_m\sigma_n$$

Solving for σ will give us two solutions:

$$\sigma_{mn} = w_m\sigma_m - (1 - w_m)\sigma_n$$

or

$$\sigma_{mn} = -w_m\sigma_m + (1 - w_m)\sigma_n$$

We can perform the same calculation as before, finding w_m from the expression of standard deviation and substituting it into R_{mn} . By doing this we will get two straight lines equation that connecting m and n in the expected return and standard deviation space .

Therefore we have

$$w_m = \frac{\sigma_{mn} + \sigma_n}{\sigma_m + \sigma_n}$$

or

$$w_m = \frac{\sigma - \sigma_n}{-(\sigma_m + \sigma_n)}.$$

Substituting w_m into the expression for R_{mn} gives us:

$$R_{mn} = \frac{\sigma + \sigma_n}{\sigma_m + \sigma_n} \mu_m + \left(1 - \frac{\sigma + \sigma_n}{\sigma_m + \sigma_n}\right) \mu_n$$

or

$$R_{mn} = \frac{\sigma - \sigma_n}{-\sigma_m - \sigma_n} + \mu_m + \left(1 - \frac{\sigma - \sigma_n}{-\sigma_m - \sigma_n}\right) \mu_n$$

In this case the investment opportunity set becomes two straight lines that connects assets m and n and both lines move in opposite direction; (see Figure 1).

Correlation of Assets Between $\rho = 0$ and $\rho = 0.5$

A correlation of 0 suggests that there is no correlation between two assets. In this case the co-variance part drops out and we will be left with the return on a portfolio.

$$\sigma^2 = w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2$$

The above risk equation is quadratic, and therefore the assets m and n are connected by a quadratic curve (see Figure 1).

A correlation of 0.5 suggests that there is some relationships between the risk and expected return of assets m and n .

In theory, we can have a perfect positive and negative correlation between two or more assets; however, that is impossible to have in the real world. The correlation between any two or more real assets is almost always greater than zero or considerably less than 1. This would mean we can never have zero risks. As a result, let us find the minimum risk by minimising the risk formulae via differentiation.

The equation (5) for the risk on the portfolio :

$$\sigma_{mn} = (w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2 + 2w_m(1 - w_m)^2 \rho_{mn} \sigma_m \sigma_n)^{1/2}$$

Calculating the derivative of the risk formula with respect to w_m :

$$\begin{aligned} \frac{d\sigma_{mn}}{dw_m} &= \frac{1}{2} (w_m^2 \sigma_m^2 + (1 - w_m)^2 \sigma_n^2 + 2w_m(1 - w_m)^2 \rho_{mn} \sigma_m \sigma_n)^{-1/2} \\ &= (2w_m \sigma_m^2 - 2\sigma_n^2 + 2w_m \sigma_n^2 + 2\sigma_m \sigma_n \rho_{mn} - 4w_m \sigma_m \sigma_n \rho_{mn}) \end{aligned}$$

Setting $\frac{d\sigma_{mn}}{dw_m} = 0$:

$$\begin{aligned} 2w_m \sigma_m^2 - 2\sigma_n^2 + 2w_m \sigma_n^2 + 2\sigma_m \sigma_n \rho_{mn} - 4w_m \sigma_m \sigma_n \rho_{mn} &= 0 \\ 2w_m \sigma_m^2 + 2w_m \sigma_n^2 - 4w_m \sigma_m \sigma_n \rho_{mn} &= 2\sigma_n^2 - 2\sigma_m \sigma_n \rho_{mn} \end{aligned}$$

Therefore:

$$w_m = \frac{\sigma_n^2 - \sigma_m \sigma_n \rho_{mn}}{\sigma_m^2 + \sigma_n^2 - 2\sigma_m \sigma_n \rho_{mn}}$$

This means the weight in assets m achieves the minimum variance. We can do a similar calculation to obtain the minimum variance for asset n . However, in this case, the risk formula needs to be in terms of w_n .

The optimum efficient frontier curve normally lies between, $\rho_{mn} = 0$ and $\rho_{mn} = 0.5$. The Figure 1 shows the relationships between assets m and n for different values of ρ_{mn} .

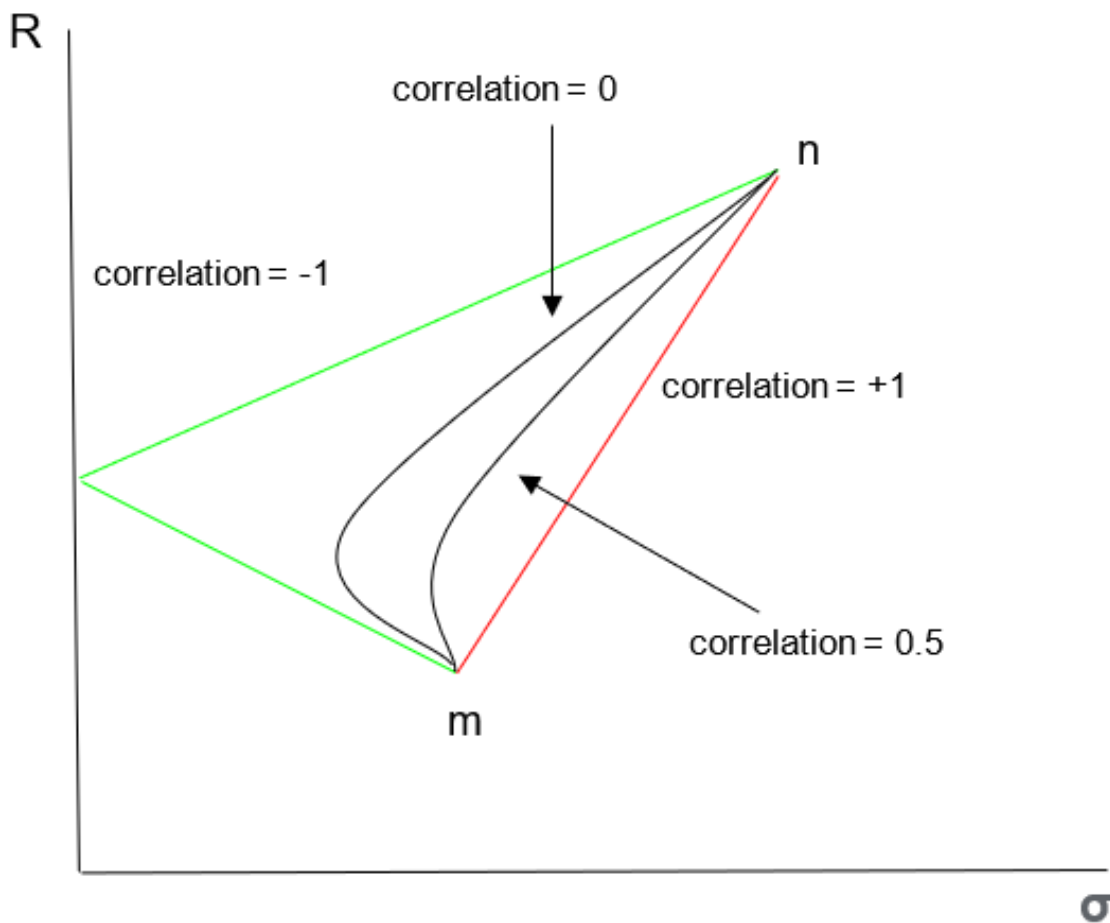


Figure 1: Strength of the relationship between risk and expected return for various correlation coefficients.

Solving the unconstrained problem

There are several techniques that we can use to solve the portfolio optimisation problem. In this paper, we will only focus on quadratic programming and genetic algorithm techniques. Also, we will examine the risk and expected return of 55 assets between January 2016 and October 2020. We choose this period of time as there was no messing off assets data.

Quadratic Programming

Quadratic Programming involves problems where the constraints are linear functions, and the objective is a quadratic function of decision variables [14]. We can use quadratic programming to solve the unconstrained part of the problem because the objective function is quadratic, and constraints are linear functions.

A quadratic programming problem in its standard form can be written as follows

[25].

$$\min \frac{1}{2} X^T Q X - c^T X$$

subject to both linear and linear inequality constraints

$$A^T x \geq b$$

$$x \geq 0$$

where Q is a $n \times n$ symmetric matrix and it is the Hessian matrix of the objective function. c^T is the the gradient of the objective function, A is a $R^{m \times n}$ matrix, X is a vector in R^n and b is a vector in R^m .

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \text{ and } b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$Q = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix}$$

A numerical example of quadratic programming optimisation follows. Consider:

$$\text{Minimise } f = 2x_1^2 + 2x_2^2 - 4x_1 - 10x_2$$

subject to:

$$8x_1 - x_2 \geq 4$$

$$2x_1 - 2x_2 \geq -2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

From the above information:

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, c = \begin{bmatrix} 4 \\ 10 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

Also:

$$A^T = \begin{bmatrix} 8 & 1 \\ 2 & -2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } b = \begin{bmatrix} 4 \\ -2 \\ 0 \\ 0 \end{bmatrix}$$

The following R code can be used to solve the above question [25]:

```
library(quadprog)
# Matrix appearing in the quadratic function to be minimised.
Dmat <- matrix(c(4, 0, 0, 4), nrow = 2)
# vector appearing in the quadratic function to be minimised.
dvec <- c(4, 10)
```

```
# Matrix defining the constraints
Amat <- t(matrix(c(8, 2, 1, 0, -1, -2, 0, 1), nrow = 4))
# vector holding the values of b
bvec <- c(4, -2, 0, 0)
# meq indicates how many constraints are equality
# No constraint is equality in this example, so meq = 0 (by default)
QP <- solve.QP(Dmat, dvec, Amat, bvec, meq=0, factorized=FALSE)
QP
```

R Output:

```
$solution
[1] 1.25 2.25

$value
[1] -14.25

$unconstrained.solution
[1] 1.0 2.5

$iterations
[1] 2 0

$Lagrangian
[1] 0.0 0.5 0.0 0.0

$iact
[1] 2
```

The R code output suggests the function f has a minimum at the point $(1.25, 2.25)$, and the minimum value is -14.25 with respect to the given constraints. The solution to the unconstrained problem is -14.5 , and it is at the point $(1, 2.5)$. In Figure 2, we can see the 3D plot of the function f and the location of the minimum point (green dot).

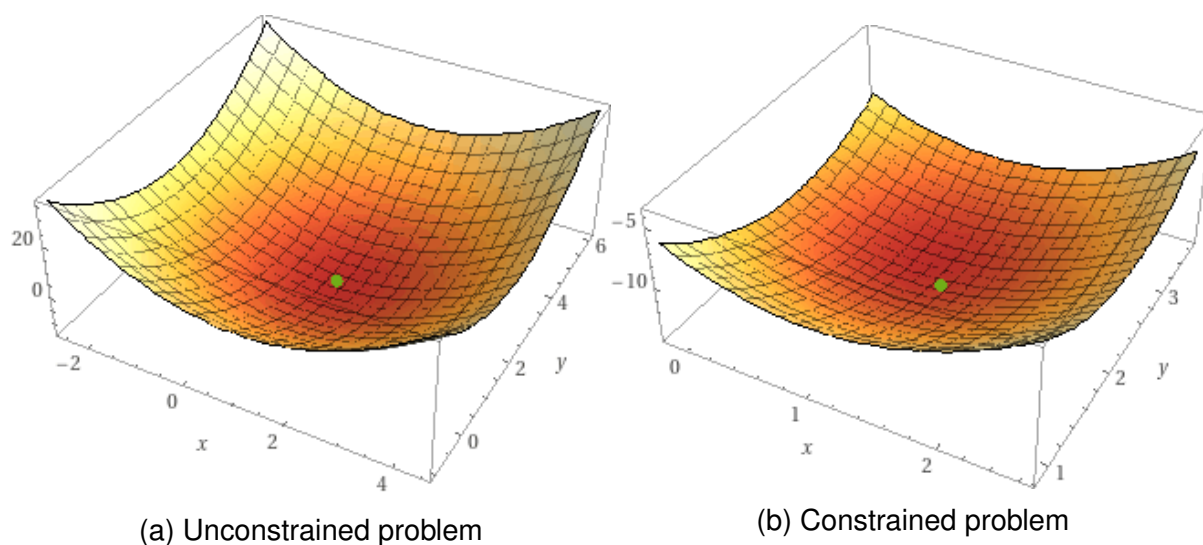


Figure 2: Shows the plot of the function f .

Figure 3 shows the contour plot of the objective function, and it shows how far the

minimum point for the constrained problem is away from the minimum point for the unconstrained problem. Note that both Figures 2 and 3 were created using <https://www.wolframalpha.com/>.

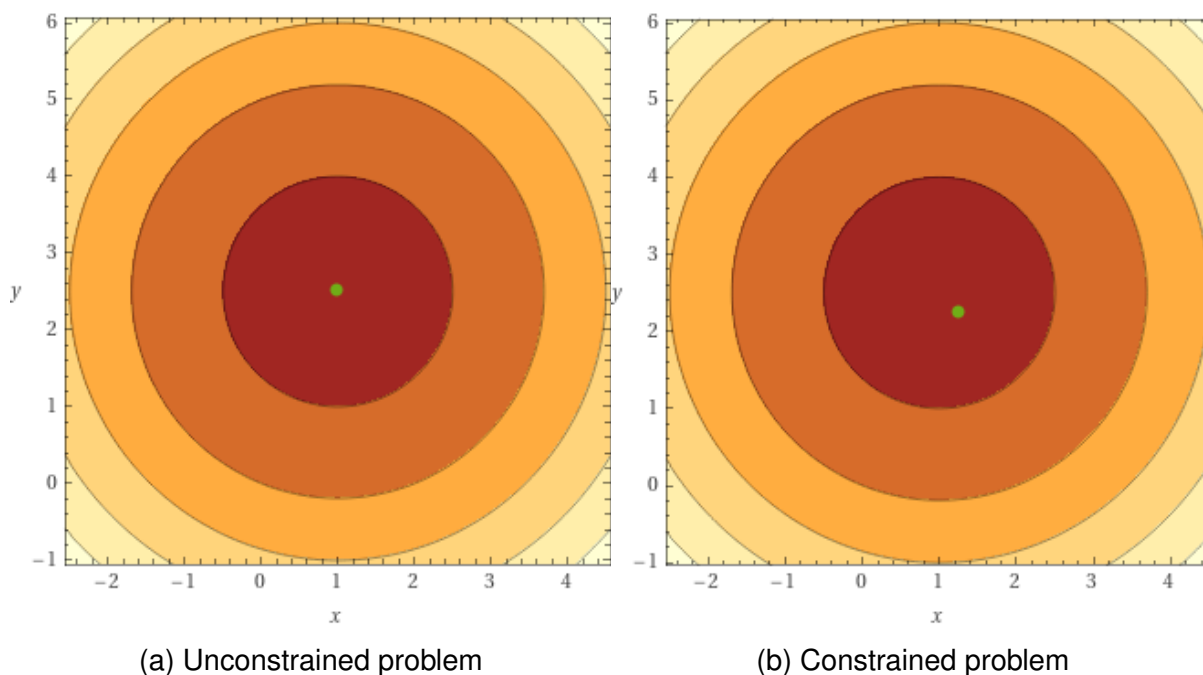


Figure 3: Contour plot

R code output

Figure 4 shows the efficient frontier for the unconstrained portfolio optimisation problem. We can plot a heat map of the Portfolio weights by the following R code.

```
library(lattice)
levelplot(weights_list, col.regions = rev(terrain.colors(100)),
          xlab='Portfolio', ylab='Stock_numbers')
```

Note: For the above code to work, the R code for solving mean variance portfolio optimisation needs to be run first.

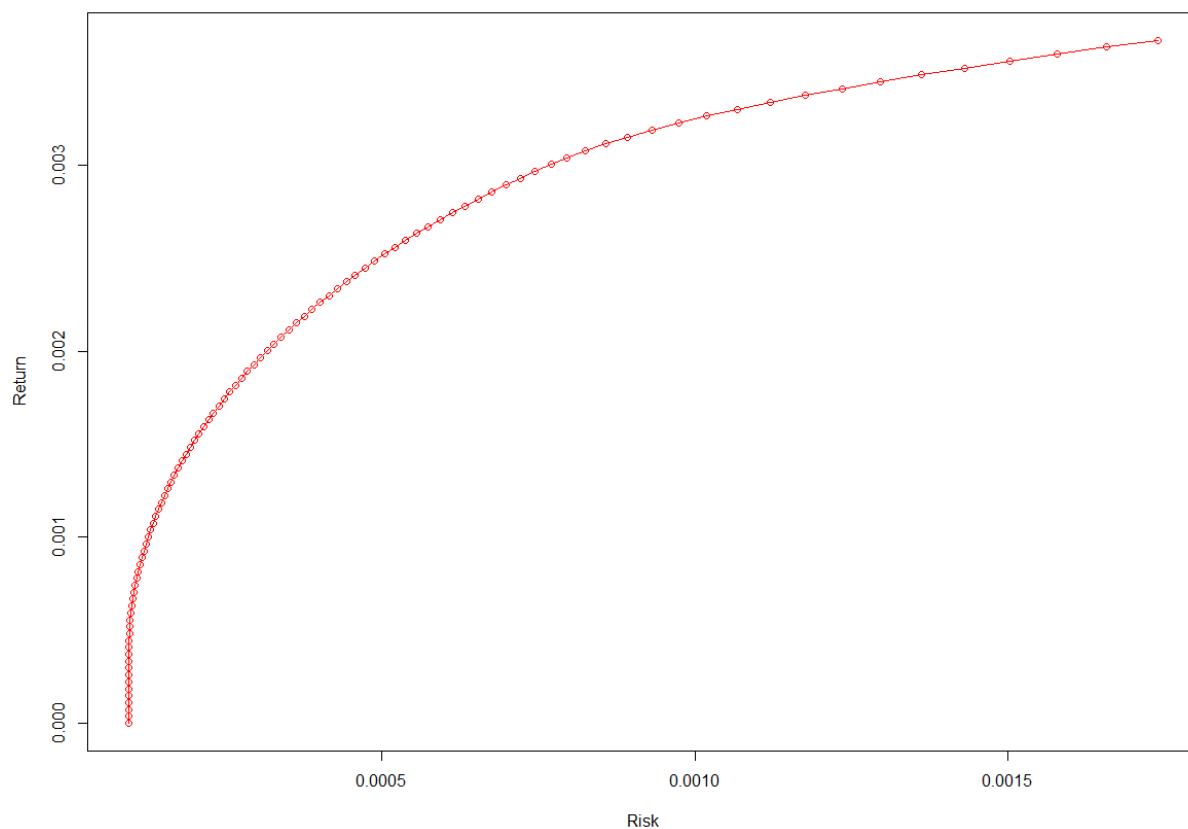


Figure 4: The efficient frontier.

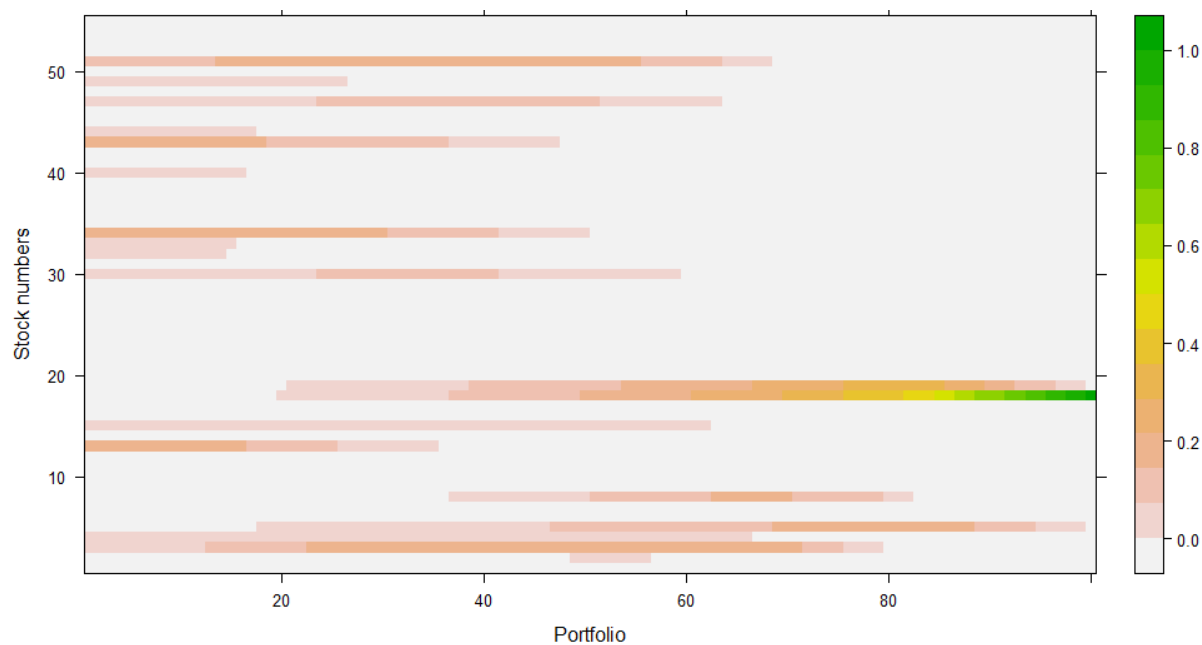


Figure 5: Weights heat map.

We can also find the portfolio that gives us the minimum return and minimum risk or vice versa, given the return tolerance. We can do that in R:

```
for ( point in 1:100 ) {
  print(tickers[which(weights_list[point,]>10-4)])
}
```

Table 1 shows the portfolio with the minimum return and minimum risk. As expected, the additions of proportions of assets in Table 1 is equal to one. The asset that gives us the maximum return is “AMD”, we invest 100% of our money in this asset. We can also conclude that out of those 55 assets to obtain any portfolios, we only need 14 assets, which means we do not need to invest in all 55 assets.

Table 1: The portfolio that give us the minimum return.

Assets	Proportions	Assets	Proportions
AMZN	0.06959706	MRK	0.2119145
NFLX	0.003974694	RES	0.01134017
JMP	0.1761246	D	0.1771800
BABA	0.03171242	MMM	0.05592903
COO	0.01712344	BR	0.02427032
MD	0.01534202	EA	0.05478582
MDT	0.02449597	AON	0.1262099

We can plot the time series of the portfolio return of the portfolio the minimum risk over the investment investment period.

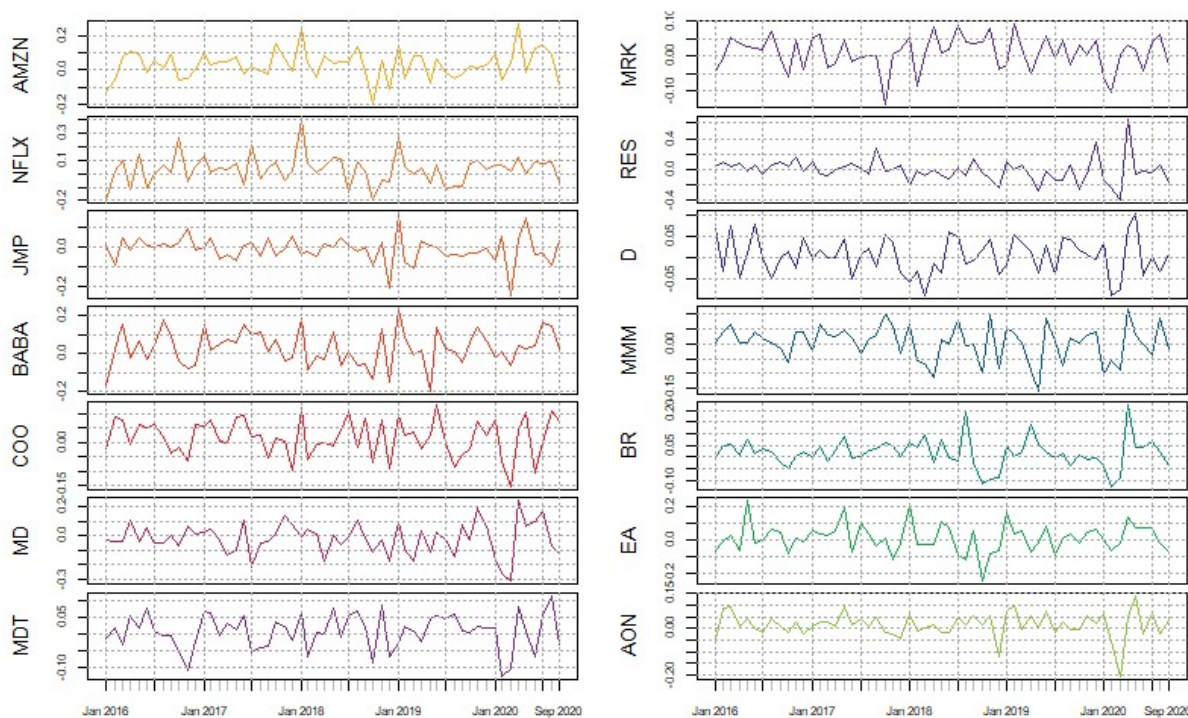


Figure 6: Time series of portfolio return.

Genetic Algorithm

Genetic algorithms (GAs) are heuristic search algorithms inspired by the theory of natural selection. They were first initiated by Holland in 1975 and have quickly become the best known evolutionary techniques [7]. In a genetic algorithm, an initial population containing chromosomes is selected randomly as the first generation. The individual chromosomes in the population are encoded into a string that consists of several feasible solutions in solution space. In a portfolio optimisation problem, each chromosome represents the weight of individual assets in a portfolio and is optimised to find a possible solution [7].

The fittest parents are chosen from the initial population in order to produce offspring [23]. The produced offspring inherit the characteristics of the parents and will be added to the next generation. The new generation consists of the better chromosomes and represented by fitness function which is originated from the objective function of the model [23]. The fitness function can be used to assess the fitness for each chromosome and define how good a solution the chromosome represents. Individuals with better fitness are more successful in adapting to their environment. They will have a greater chance of surviving and reproducing, whilst individuals who are less fit will be eliminated.[6]. This procedure keeps on iterating and, at the end, a generation with the fittest individuals will be found.

Before going through the basic steps of a simple GA, we need to understand some of the biological terminology such as selection, crossover and mutation operators.

Selection operator:

Selection operators determine which representatives of the existing population will be carried on to the next generation. There are different functions implementing selection operator in R. We are using the `lrSelection` selection operator. This selection operator decides the possibility of advancing to the next generation based on the fitness rank in the current population [15].

The crossover operator:

The population resulting from the selection is typically divided into two parts. Then, the genetic information of two parents is combined to generate new offspring. We are doing this with the aim of producing new solutions that are better than previous solutions. Many different types of crossover exist, e.g. single-point crossover, two-point crossover, and arithmetic crossover [22]. There are various functions implementing crossover operator in R. We are using the `LaplaceCrossover` operator. The Laplace crossover operator is based on the Laplace distribution. A full description of the Laplace crossover operator can be found in the work of Deep and Thakur [9].

Example of crossover: A chromosome can be divided into genes, which continue information about certain traits, and each locus in the chromosome has two possible genes: 0 and 1. In this example let us look at 8-bit strings.

Parent 1

0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

Parent 2

1	1	0	1	0	1	1	1
---	---	---	---	---	---	---	---

We can create two offspring by randomly selecting each bit (gene) from one of the corresponding genes of the parent chromosomes.

Offspring 1

0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---

Offspring 2

1	1	0	1	1	1	0	1
---	---	---	---	---	---	---	---

Here, we have two individuals that we can say have inherited the characteristics of the parents.

The mutation operator:

After the crossover process, the GA produces new offspring by modifying the existing individual chromosomes in this section of the process. This can be done by randomly flipping some of the bits in a chromosome with some probability; usually, very small [21]. We maintain genetic diversity in the population, which means we will have more

potential candidate solutions with different genetic information. This will allow the GA to find better solutions at each iteration.

Example of mutation: In this example we have an 8-bit string after and before mutation. We flip the bit in position 5.

Offspring 1 before mutation.

0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---

Offspring 1 after mutation.

0	1	1	0	1	1	1	0
---	---	---	---	---	---	---	---

When we find the optimum profiles portfolios for the cardinality constrained portfolio optimisation problem in R, it is crucial to test different values for the crossover and mutation probability and set those probabilities equal to a reasonable value. We do this because setting the mutation probability to a higher number will lead to a random search (which is no more beneficial than searching for a needle in a haystack). Also, setting the crossover probability to a lower probability will lead to poor results. We will also show this in Section 5.4, where we attempt to improve our solutions by altering the crossover and mutation probabilities.

The steps in a genetic algorithm are as follows:

Algorithm 1 Genetic Algorithm

1. Generate an initial population of chromosomes.
 2. Evaluate the fitness of individuals chromosomes in the population.
 3. Select a pair of parent chromosomes from the current population.
 4. Combine parents to produce offspring using crossover operator.
 5. Mutate the two offspring at each locus with with lower probability.
 6. Evaluate fitness of the offspring and select the Individuals with better fitness to be added in the next generation.
 7. Replace some or all of the population by the offspring.
 8. **if** a satisfactory solution has been found stop, **else** go to 3.
-

Other termination conditions exist, e.g. running each point for a given number of iteration to achieve a satisfactory solution. However, what is a satisfactory solution or how many iterations are sufficient to achieve such a solution is a subject for debate. Recall, as before we are examining a data set involving 55 assets between January 2016 and October2020.

Results from using the GA to solve the unconstrained problem

We solved the unconstrained problem using quadratic programming as the objective function is quadratic. Genetic algorithm techniques can also be used to solve both the unconstrained and constrained portfolio optimisation problem. **R Output:**

Figure 8 shows the R output for solving the unconstrained problem using both quadratic programming and genetic algorithm techniques. As we can see, we have obtained precisely the same efficient frontier using two different approaches.

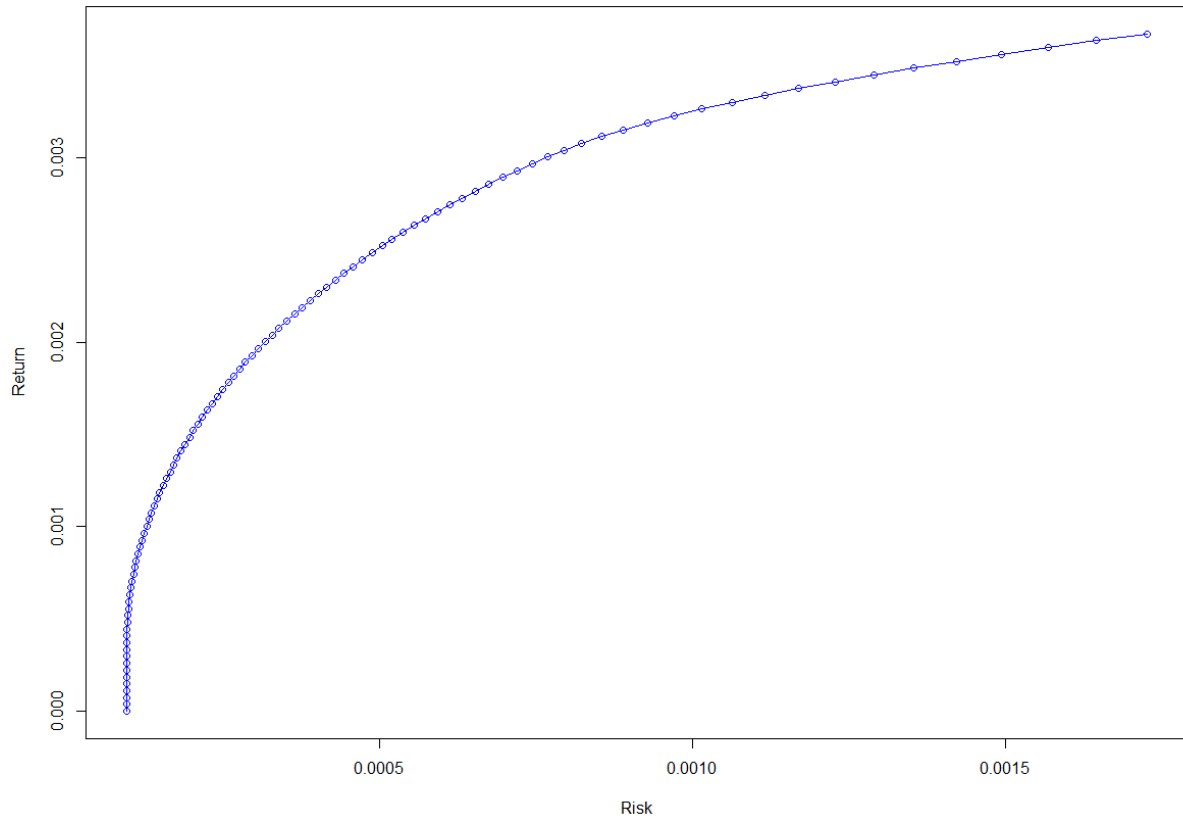


Figure 7: Unconstrained problem efficient frontier using the GA technique.

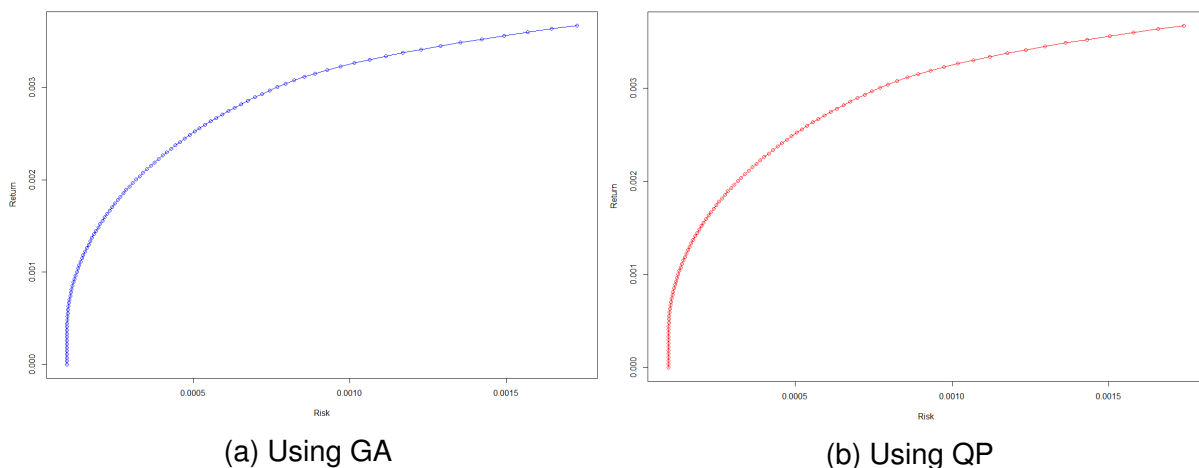


Figure 8: Solution to unconstrained problem using QP and GA.

We can use both GA and quadratic programming to solve the unconstrained portfolio optimisation problem. However, the quadratic programming technique is much faster than GA to find the optimal portfolios in R (see Figure 9). For example, it takes R 0.08 seconds to compute the first point using the quadratic programming technique; however, it takes R 10 seconds to compute the same point using the GA.

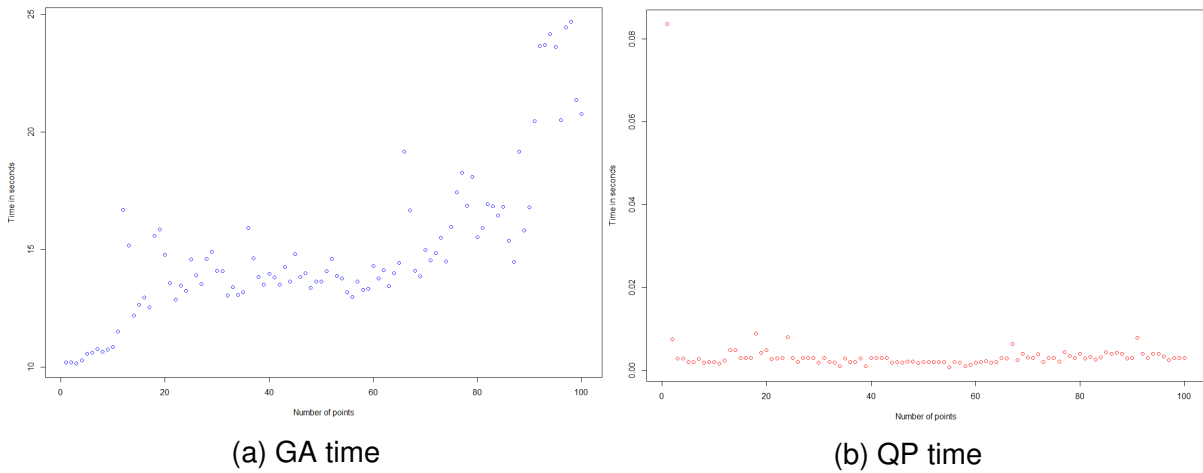
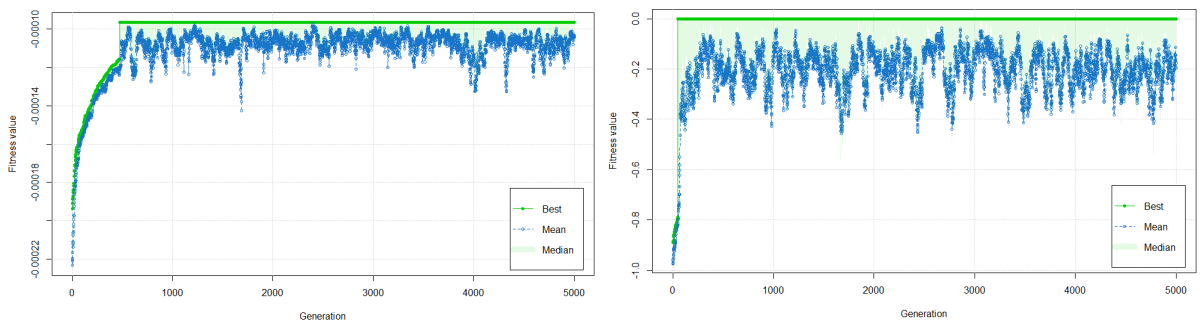


Figure 9: A time comparison of GA and QP for solving the unconstrained problem.

Figure 10 shows the generation-fitness plot for the first and last point of the efficient frontier. As we can see, for the first point, it takes more iterations to reach the optimum solution compared to the point at the top of the efficient frontier. From this, we can conclude that points at the bottom of the efficient frontier are harder to compute; hence, more iteration needed to reach the optimum solution. Also, it takes more time for R to find the optimum solutions. On the other hand, the points at the top of the efficient frontier are easy to find and take less time.



(a) Generation-fitness plot for the first point. (b) Generation-fitness plot for the last point.

Figure 10: Generation-fitness comparison of the first and last points for the unconstrained problem.

The cardinality constrained portfolio optimisation problem

In the previous section, we saw how to solve the mean-variance portfolio optimisation problem. Investors face some restrictions such as floor-ceiling constraints, pre-assignment constraints, and cardinality constraints in the real world. The floor-ceiling constraint restricts the proportion of each asset in the portfolio to lie between certain lower and upper limits [17]. This help investors to avoid very large or small position which means lower transaction cost. The pre-assignment constraint allows investors

who may intuitively favour a specific set of assets in their portfolio, with its proportion either fixed or determined [19].

There are other constraints, such as class constraints, class limit constraints and round lot constraints. However, in this section, we will only look at the cardinality constrained portfolio optimisation problem.

The cardinality constraint

The cardinality constraint imposes a limit on the number of assets in the portfolio. These limitations arise because investors have management issues of their portfolio since managing many assets in a portfolio may be hard to monitor. They may also seek to reduce transaction costs and/or to assure a certain degree of diversification by limiting the maximum number of assets in their portfolios [19]. Therefore cardinality constrained portfolio optimisation is a significant practical problem. By introducing the cardinality constraint into the mean-variance model, the problem becomes a mixed-integer quadratic programming problem which is an NP-hard problem. A problem is NP-hard if the algorithm for solving it is at least as hard as any NP-problem (non-deterministic polynomial time) [26]. The cardinality constrained portfolio optimisation problem is as follows [1]:

$$\text{Minimise } \lambda \left(\sum_{i=1}^n \sum_{j=1}^n w_i w_j \sigma_{ij} \right) - (1 - \lambda) \left(\sum_{i=1}^n w_i \mu_i \right) \quad (6)$$

subject to:

$$\sum_{i=1}^n w_i = 1 \quad (7)$$

$$\sum_{i=1}^n z_i \leq k \quad (8)$$

$$R = \sum_{i=1}^n w_i \mu_i \quad (9)$$

where:

$\lambda \in [0, 1]$ is the risk preference of the investor.

k is the desired number of assets in our portfolio.

z_i is a binary variable denote whether an asset is selected or not:

$$z_i = \begin{cases} 1, & \text{if any of asset } i \text{ is held, for } i = 1, 2, \dots, n \\ 0, & \text{otherwise} \end{cases}$$

Also, we have the analogous constraints as in page 3.

$$0 \leq w_i \leq 1 \text{ for } i = 1, 2, \dots, n$$

$$R \geq R_0$$

By introducing the cardinality constraint into the mean-variance model, finding the solution (the efficient frontier) becomes harder. If the value of k is equal to the total

number of assets in a portfolio, then the problem is identical to the unconstrained problem. The problem becomes harder to solve as we reduce the value of k . This is because there will be fewer assets each time to choose from the total number of assets, yet the same returns are needed.

Solving the cardinality constrained problem using a GA

Adding the cardinality constraint to the mean-variance portfolio optimisation problem makes the constraints non-linear. Hence, we can not use the quadratic programming technique to solve the problem. We can also see this in Figures 11 and 12 as the efficient frontier is starting to become a non-quadratic graph as we decrease the value of k . The following subsections give the results of running our GA on the problem. As before, we are examining the risk and expected return of 55 assets between January 2016 and October 2020.

R Outputs for different values of k

We will analyse the efficient frontier and their generation-fitness plots for different values of k . We will first look at $k = 20$ and $k = 10$ and then gradually decrease the value of k .

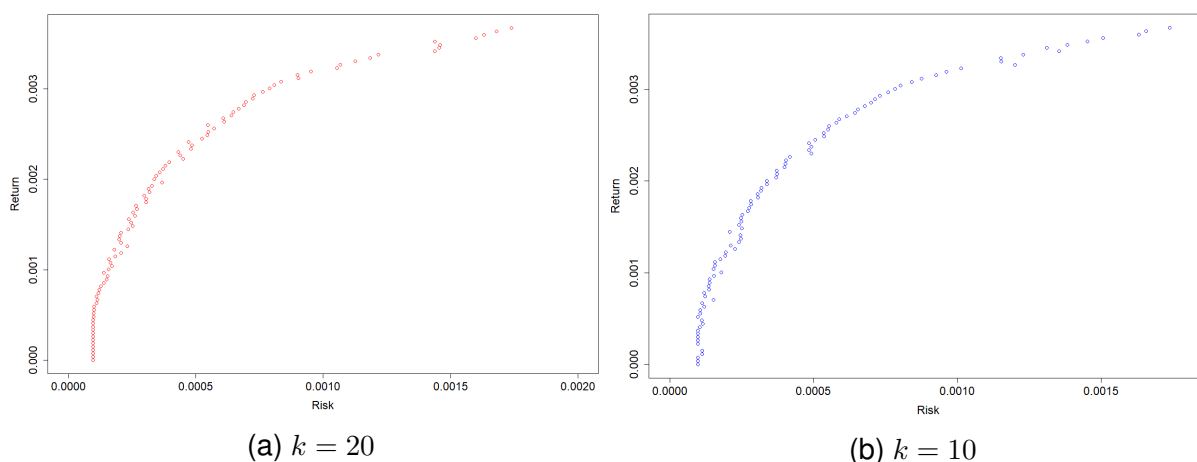
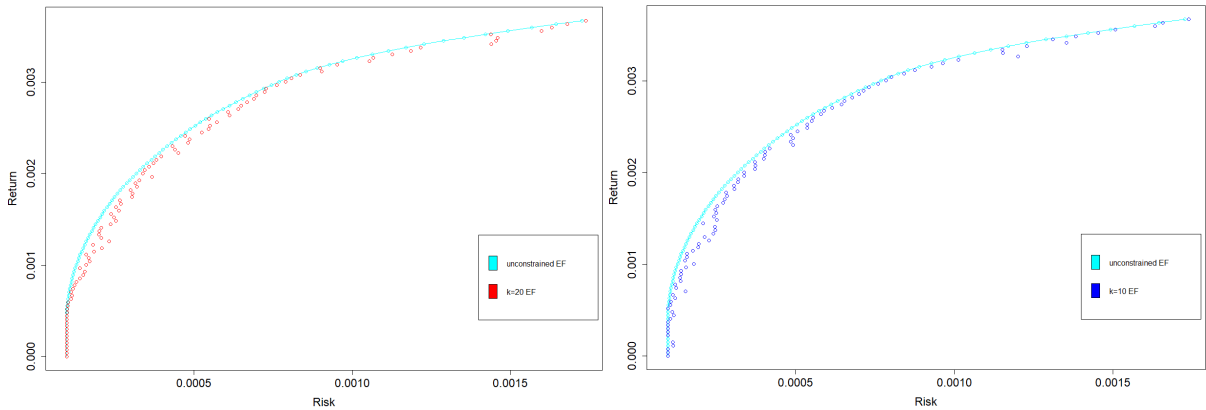


Figure 11: The GA-computed efficient frontier for $k = 20$ and $k = 10$.

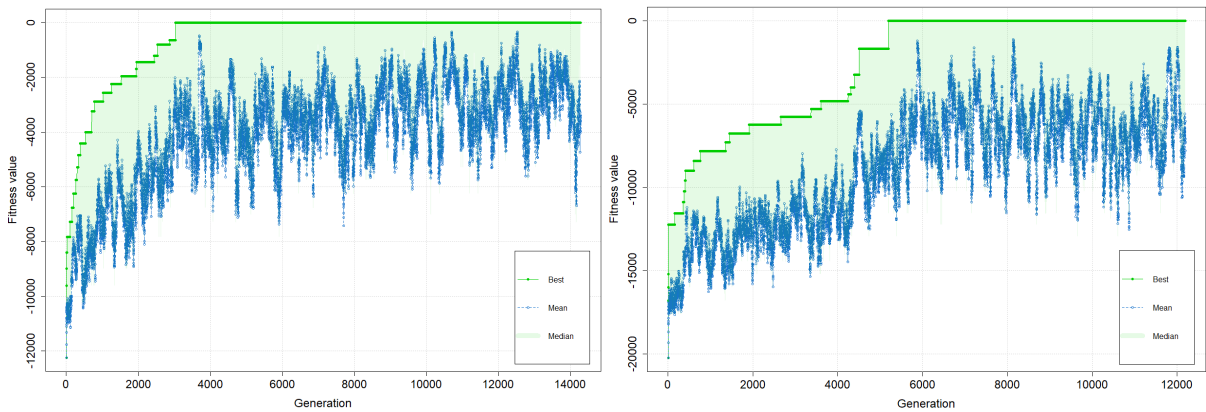
Figure 11 shows the efficient frontier for the cardinality constrained portfolio optimisation problem when $k = 20$ and $k = 10$. Let us compare the efficient frontier for both k and the unconstrained efficient frontier (shown in Figure 12). There is not much difference between the three curves, especially at the top of the curve. However, there is a small deviation below the average return of 0.002. When $k = 20$, the lower left part of the efficient frontier is a better approximation of the unconstrained efficient frontier than $k = 10$. We can conclude that as we decrease the value of k , we might see progressively larger deviations in the cardinality constrained efficient frontier compare to the unconstrained efficient frontier.



(a) Unconstrained and $k = 20$.

(b) Unconstrained and $k = 10$.

Figure 12: Comparison of the unconstrained efficient frontier with the constrained efficient frontier using $k = 20$ and $k = 10$.



(a) Generation-fitness plot for $k = 20$

(b) Generation-fitness plot for $k = 10$

Figure 13: Comparative sample generation-fitness plots for $k = 20$ and $k = 10$.

The generation-fitness plot shows how a particular point reaches the optimum solution and how many iterations it takes for that to happen. In Figure 13, we can see the generation-fitness plots for $k = 20$ and $k = 10$ for the point where the risk is at a minimum. For each point in the efficient frontier, the generation-fitness plot will be different as some points might reach the optimum solution faster or slower than others.

In Figure 10, we looked at the generation-fitness plot for the unconstrained portfolio optimisation problem. We saw that it takes the GA a few iterations to find the optimal solution; hence it is straightforward to find the efficient frontier - this is expected since the unconstrained problem is a quadratic optimisation problem. In Figure 13, we can see the generation-fitness plots for the cardinality constrained optimisation problem. As we can see, it takes more iterations to solve the cardinality constrained problem compared to the unconstrained problem. For example, it takes approximately 600 iterations for the first point (the lowest left point) in the efficient frontier to reach the optimum solution. However, for the same point, it takes around 3000 iterations when $k = 20$ and 5000 iterations when $k = 10$ to get to the optimum solution.

Before looking at Figure 13 in more detail, we need to know how each point reaches the optimum solution in GAs. The horizontal lines within the generation-fitness plot are

called local optima. The horizontal line at the top of the generation-fitness plot is the global optimum. In the GA, the mutation operator attempts to improve the solution to one of a higher fitness. When it reaches the local optimum, the crossover operator takes over, and the graph jumps; hence, we see the staircase shape. At the top of each stair, the GA uses the mutation operator for all these generations within the horizontal line. Therefore, if the horizontal line continues for long enough, the obtained solution is the optimum solution (global optimum).

Previously we said that the difficulty of the cardinality constrained problem increases as we decrease the value of k each time. We can see this by comparing the generation-fitness plots of $k = 20$ and $k = 10$ (see Figure 13). It takes fewer iterations for $k = 20$ compare to $k = 10$ to reach the optimum solution, and less of a staircase shape is apparent. However, it takes more iterations when $k = 10$ compare to $k = 20$ to reach the optimum solution, hence more staircase shape. We also had to increase the run parameter in the R code from 5000 (where $k = 20$) to 7000 (where $k = 10$) because of the increase in the problem difficulty and make sure we get better solutions.

Figure 14 compares the efficient frontiers of $k = 20$ and $k = 10$ to the unconstrained optimisation problem efficient frontier on one plot. We can conclude that the efficient frontiers for $k = 20$ and 10 are a good approximation of the unconstrained problem efficient frontier. Also, as we decrease the value of k , we should see more deviation at the lower left and middle of the efficient frontier than the unconstrained efficient frontier

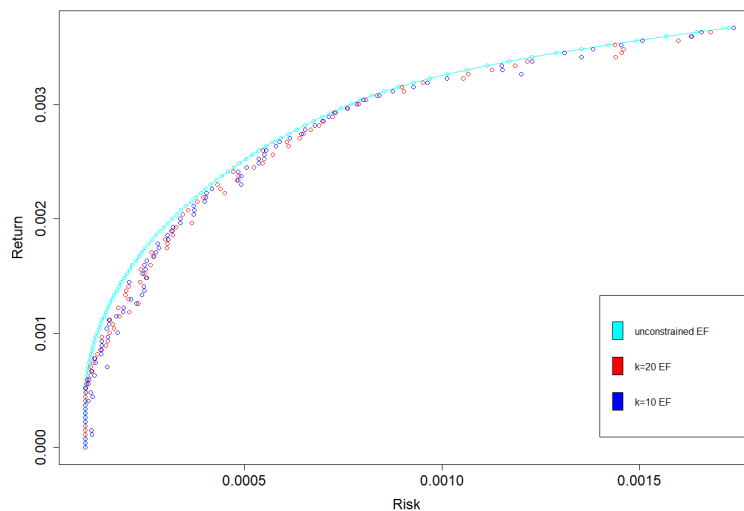


Figure 14: Comparison of the efficient frontiers produced using the GA with that of the unconstrained problem (light blue).

Let us look at the efficient frontier and the generation-fitness plots for $k = 5$ and $k = 3$. We did not study the efficient frontier for $k = 4$ because we do not expect much difference between $k = 5$, $k = 4$ and $k = 3$. When $k = 5$, there is slightly more deviation at the bottom of the curve than $k = 10$ and $k = 20$ compare to the unconstrained problem. As we decrease the value of k to 3, we observe more deviation (as expected). Sub-figures (b) in Figures 15 and 16 show the approximate traced efficient frontier for both values of k . The traced efficient frontier was produced by removing the non-efficient points from Sub-figures (a) and then tracing the approximate efficient frontier of the remaining points. We can otherwise remove the non-efficient

points using the code in the paper of [13]. We can also see discontinuities in the traced efficient frontier for both values of k . This is because we are only minimising the risk and not maximising the return simultaneously - that is, we are treating it as a one-objective optimisation problem. An example of this can be found in Figures 1 and 3 of the work of [5].

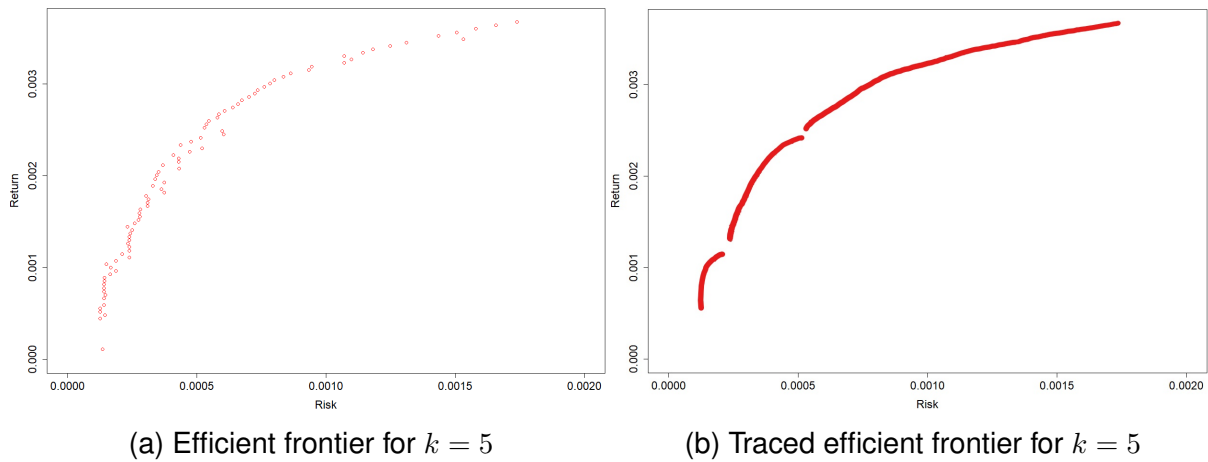


Figure 15: Shows the efficient frontier for $k = 5$.

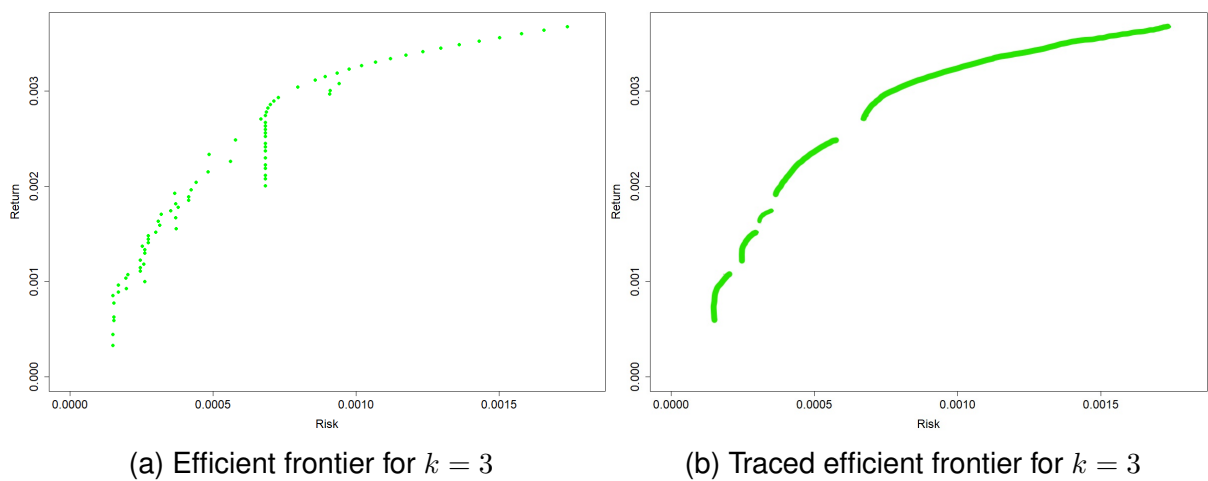
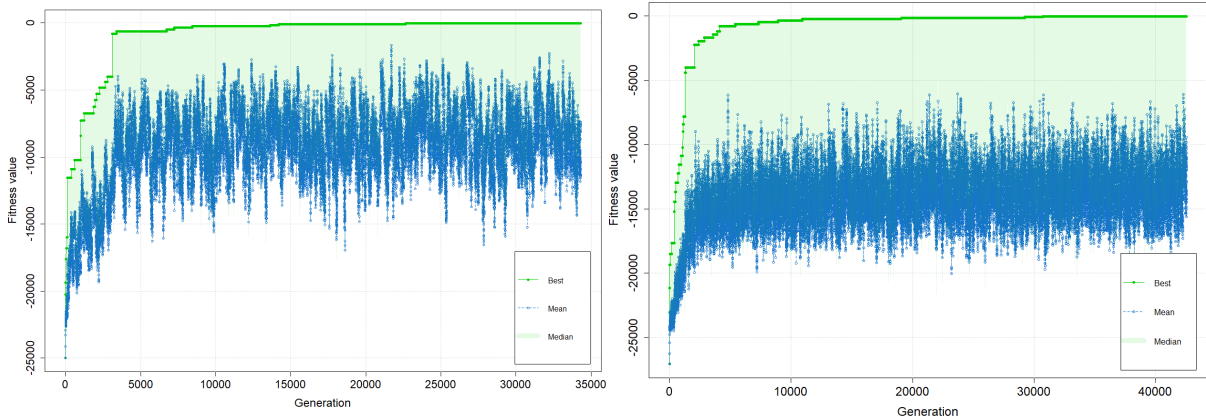


Figure 16: The efficient frontier for $k = 3$.

For $k = 5$ and $k = 3$, the number of iterations for the risk level below 0.0020 is 40000, for the risk level larger than 0.0020 is 90000, and the run parameter is equal to 10000.

In Figure 17 we can see the generation-fitness plots for $k = 5$ and $k = 3$. As expected, it takes the GA a large number of iterations (approximately 30000) to find the optimum solution for the first point when $k = 3$ compare to 13000 iterations when $k = 5$. We also had to increase the run parameter to 10000 as the problem difficulty increases as we decrease the value of k . Another way to see the increases in complexity of the problem is to analyse the average fitness in the population in the generation-fitness plot. For $k = 5$, the average fitness within the population higher than $k = 3$ (see Figure 17). Hence, the average fitness in the population would be higher for an easy problem.

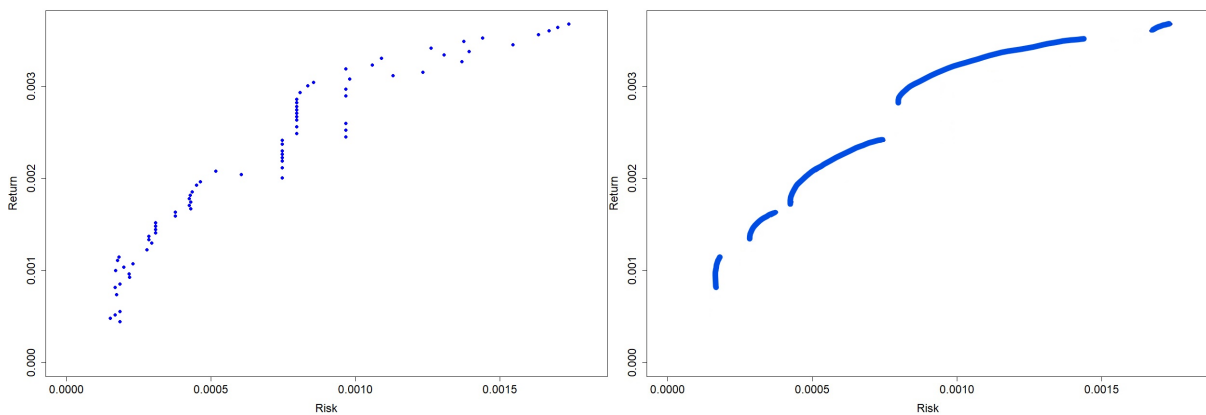


(a) Generation-fitness plot for $k = 5$

(b) Generation-fitness plot for $k = 3$

Figure 17: Comparative sample generation-fitness plots for $k = 5$ and $k = 3$.

Finally, we look at $k = 2$, and we can see the approximation of the traced efficient frontier after removing the non-efficient points in Figure 18. There is even more deviation at the left and middle part of the efficient frontier than other values of k 's that we saw previously compare to the unconstrained efficient frontier. As the value of k decreases, the efficient frontier is starting to move away from the unconstrained efficient frontier. However, the top right of the efficient frontier is a good approximation of the unconstrained efficient frontier when $k = 2$ (see Figure 20).



(a) Efficient frontier for $k = 2$

(b) Traced efficient frontier for $k = 2$

Figure 18: The efficient frontier for $k = 2$.

For $k = 2$, the number of iterations for the risk level below 0.0020 is 40000, for the risk level larger than 0.0020 is 90000, and the run parameter is equal to 12000.

Figure 19, shows the generation-fitness plot when $k = 2$. As expected, it takes the GA a larger number of iterations to find the optimum solution for the same point that we looked at in the previous generation-fitness plots. As expected, the average fitness in the population is lower for $k = 2$ compare to $k = 3$ and other values of k greater than 3. Note, it is essential to know that the generation-fitness plot may differ depending on the individual run.

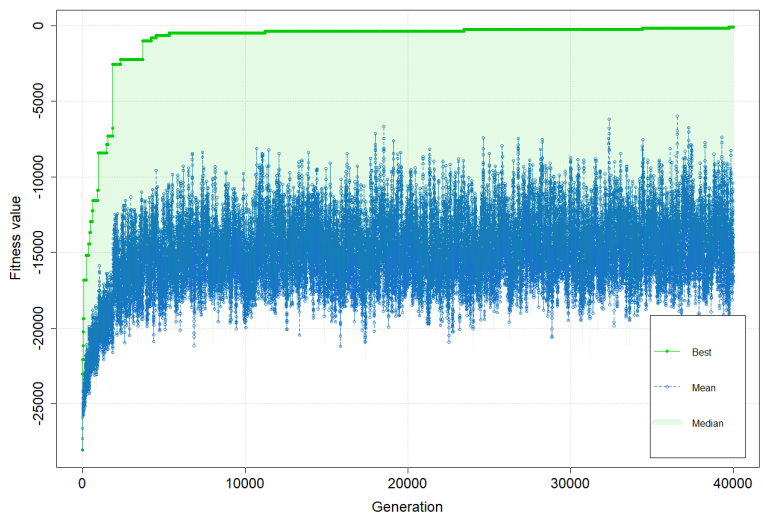


Figure 19: Generation-fitness plot for $k = 2$.

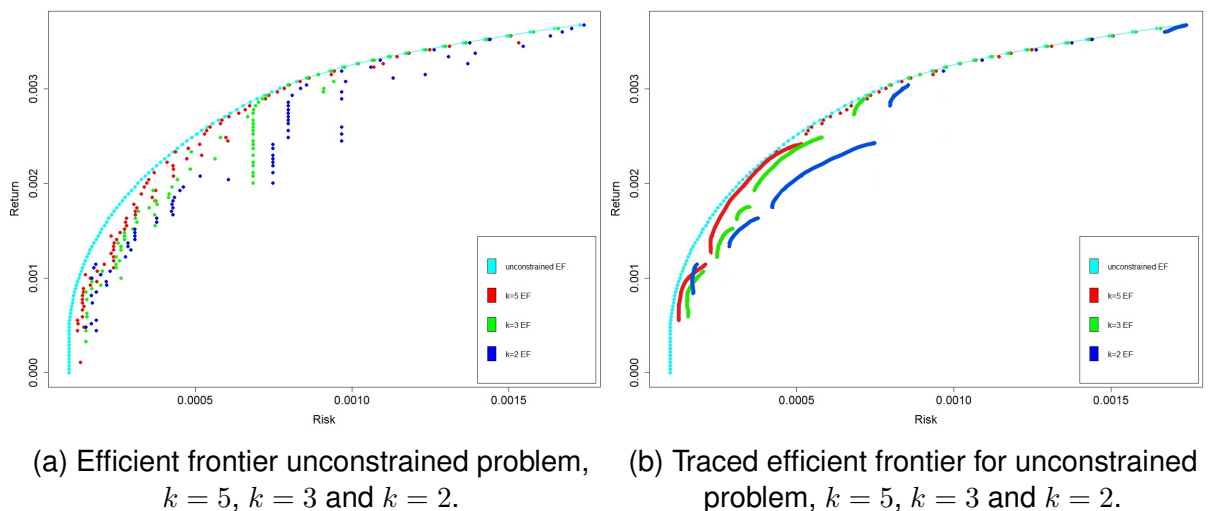


Figure 20: Comparison of the efficient frontiers produced using the GA techniques for the unconstrained problem and constrained problem $k = 5$, $k = 3$ and $k = 2$.

From the approximately traced efficient frontiers in Figure 20, as we decrease the value of k , we are starting to move away from the unconstrained efficient frontier. However, the top part of the graph for all values of k is a good approximation of the unconstrained efficient frontier - this is not surprising since those portfolios at the top part of the efficient frontier are commonly composed of a deficient number of assets. For example, the top right-hand efficient frontier point represents 100% investment into the single highest return asset. Also, as we decrease the value of k , the jumps between the curve discontinuity become larger.

Figure 21 shows the heat map of the portfolio weights, for $k = 5$ and $k = 3$.

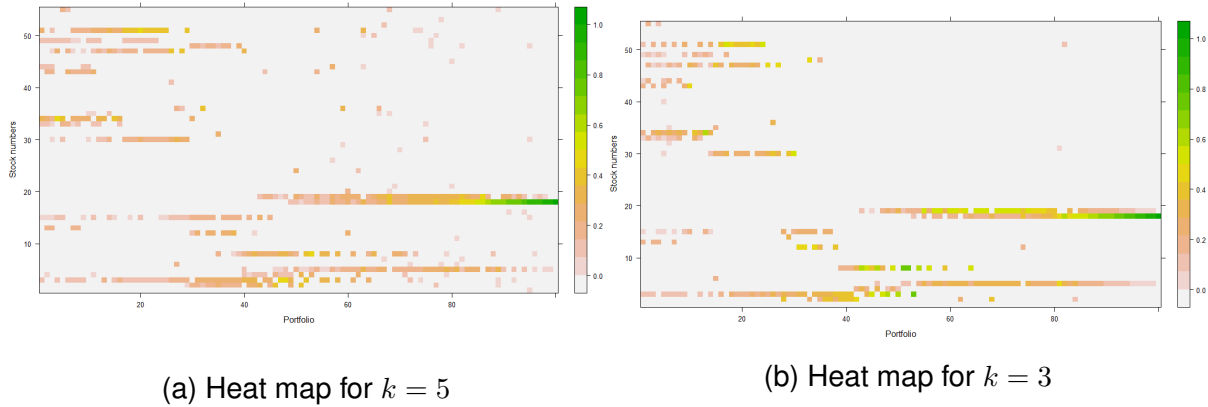


Figure 21: Heat map plot of asset weights by number of asset, for $k = 5$ and $k = 3$.

Figure 22 shows the heat map of the portfolio weights for $k = 2$.

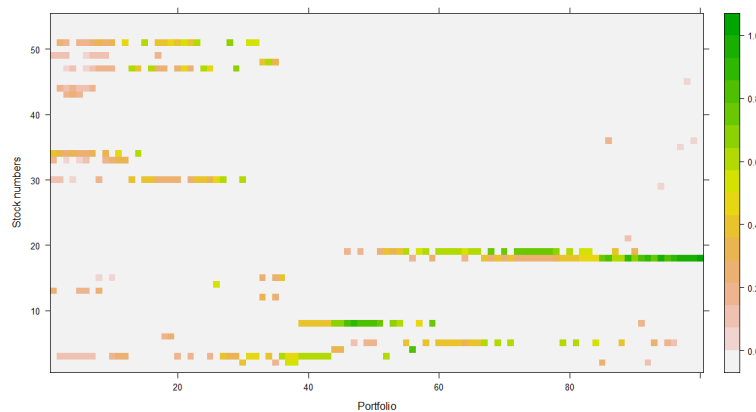


Figure 22: Heat map plot of asset weights by number of asset for $k = 2$.

Figures 21 and 22 show that asset 17 will give us the highest return, and the assets at the leftmost column of the heat map will give us the minimum return. Since asset 18 is not in the minimum return region, it represents the point with the highest risk in the efficient frontier. We can also find the portfolio that gives the minimum return and minimum risk or maximum return and maximum risk using the R code on page 14.

For example, Table 2 shows the portfolio that gives the minimum return when $k = 5$. As expected, the additions of proportions of assets in Table 2 is equal to one. The number of assets that give us the minimum return/risk is greater than K . This might

be that the GA is a heuristic method, which means our solutions are approximations of the ideal solutions. The asset that gives us the maximum return is “AMD”, and we invest 100% of our money in this asset.

Table 2: The portfolio that give us the minimum return.

Assets	Proportions
AMZN	0.09876884
BABA	0.03850131
MDT	0.10769915
MRK	0.31089986
MMM	0.13739378
EA	0.08825112
AON	0.21848595

Now let us look at a portfolio that is located somewhere in the middle of the efficient frontier for $k = 5$.

Table 3: Portfolio number 30 where $k = 5$.

Assets	Proportions
AAPL	0.1997389
MA	0.1428969
BABA	0.2144142
MRK	0.2741906
MSFT	0.1687594

As we go from the bottom to the top of the efficient frontier (i.e. left to right on the heat map), the number of assets in a portfolio will start equal to the value of k (see Table 3). This also confirms that the points at the bottom of the efficient frontier are more difficult to compute.

Figure 23 shows the number of portfolios against the number of assets in each portfolio.

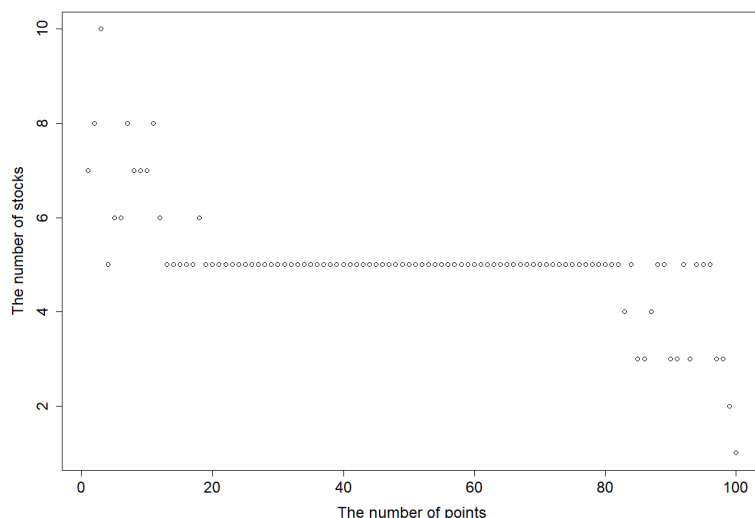


Figure 23: The number of assets in each portfolios when $k = 5$.

Table 4 shows the portfolio that gives the minimum return when $k = 3$. As expected, the number of assets in this portfolio is greater than the value of k (discussed previously).

Table 4: The portfolio that give us the minimum return when $k = 3$.

Assets	Proportions
AMZN	0.07718752
JMP	0.17901866
BABA	0.03674124
MDT	0.04713987
MRK	0.21296334
D	0.17722890
MMM	0.07249896
EA	0.05793089
AON	0.13929062

If we look at the portfolios between point 85 and 99, we have similar assets in each portfolio and they are “TSLA”, “AMD” and “WIX”. However, the value of proportion is different in each portfolio.

Table 5 shows the portfolio that gives the minimum return when k is equal to 2. The assets that give us the maximum return is “AMD” and we invest 100% of our money in this asset.

Table 5: The portfolio that give us the minimum return when $k = 2$.

Assets	Proportions
JMP	0.22855982
COO	0.08211846
MDT	0.18836451
MRK	0.36831630
EA	0.13264091

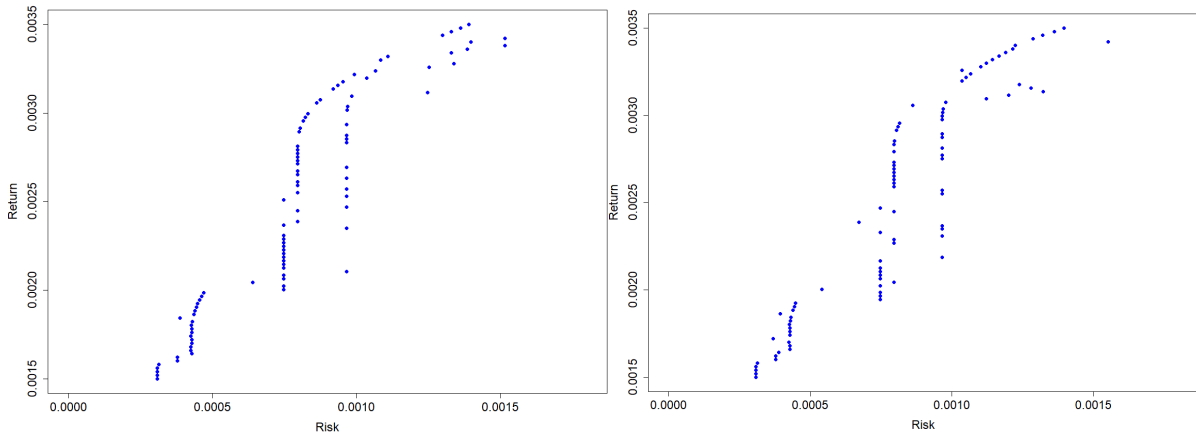
Next, we will look at how the change in crossover and mutation probabilities affect the efficient frontier’s shape.

Probability of crossover and mutation

Increasing or decreasing the crossover and mutation probabilities may improve our solutions to the cardinality constrained portfolio optimisation problem. We saw from the previous results that most non-efficient points are located in the middle part of the efficient frontier. In GAs, the mutation probability is usually small to avoid a random search for the solutions. We decided to set the probability of mutation and crossover to 0.4 and 0.6, respectively, for the efficient frontier in Figure 18.

Now, we will look at how the decrease in the mutation probability to 0.1 will change our solutions (the probability of the crossover is left unchanged). Since most of the non-efficient points are in the middle part of the frontier, we will only examine them for $k = 2$ between the return values of 0.0015 and 0.0035. We will not look at other values of k that we have seen because the crossover and mutation probability in these cases are 0.4 and 0.1, respectively.

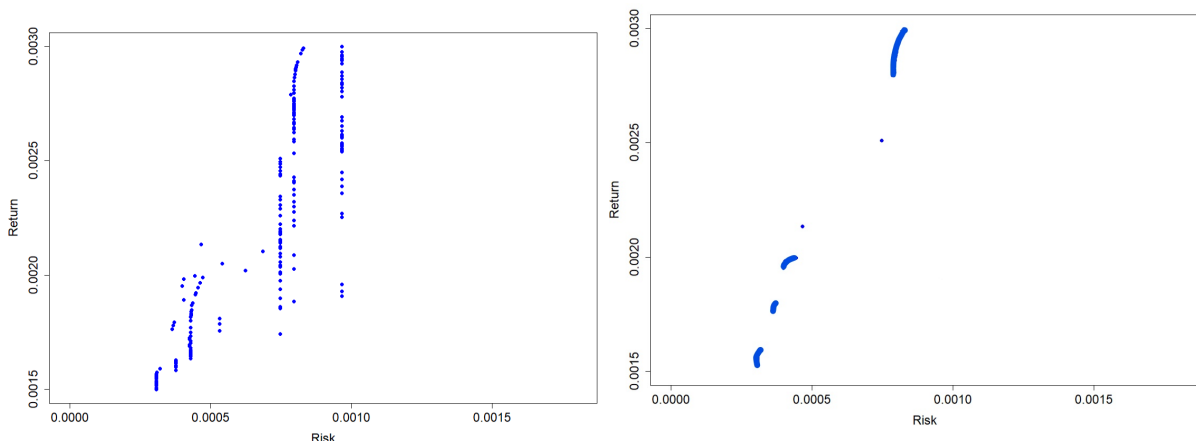
In Figure 24, CP and MP represent the crossover and mutation probabilities, respectively. By changing the mutation probability to 0.1, we can see some improvement in the efficient frontier as there are now fewer non-efficient points in the efficient frontier. However, what is happening between the return of 0.0022 and 0.0026 is unclear. To understand this better, we increased the number of points from 100 to 200 and increased the number of iterations.



(a) CP = 0.6 and MP = 0.4 efficient frontier. (b) CP = 0.6 and MP = 0.1 efficient frontier.

Figure 24: Efficient frontiers for $k = 2$ for varying crossover and mutation probabilities.

Figure 25 shows the efficient frontier and the approximate traced efficient frontier where the crossover and mutation probability are 0.6 and 0.1 using 200 points. The number of iterations for the risk level below 0.0020 is 90000, for the risk level larger than 0.0020 is 150000, and the run parameter is equal to 20000.



(a) CP = 0.6 and MP = 0.1 efficient frontier. (b) Traced CP = 0.6 and MP = 0.1 efficient frontier.

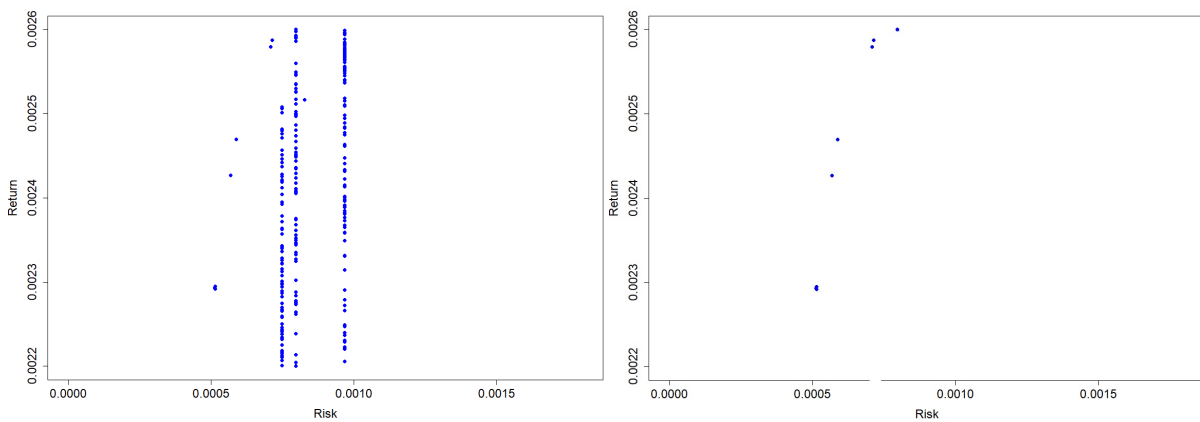
Figure 25: The efficient frontier for $k = 2$ using 200 points for return values between 0.005 and 0.0030.

By increasing the number of points and iterations, we can see slightly more of an improvement. But, we are still not sure what is happening between the return values of 0.0022 and 0.0026. We can conclude that Sub-figures (a) and (b) suggest that another discontinuity between return values 0.0022 and 0.0026 has been revealed by altering the crossover and mutation probability. However, there is not enough evidence to validate this claim; hence further investigation is needed to make a coherent conclusion.

We believe there are one or more discontinuities between the return values of 0.0022 and 0.0026. We narrowed the range of returns to between 0.0022 and 0.0026 in the efficient frontier to investigate this further. The number of iterations was unchanged, except for risk levels larger than 0.0075 (where the number of iterations is 150000). In Figure 26, the numbers of points are 350. Sub-figure (a) in Figure 26

displays the efficient frontier between the return values where we believe the discontinuities exist for $k = 2$ (it took R approximately 12 hours to produce this sub-figure).

In Sub-figure (b), we concentrate on efficient points by removing the non-efficient points by hand. Although there are few efficient points, in this case, we have enough evidence to conclude that there are one or more discontinuities between the return values of 0.0022 and 0.0026, and this is revealed by altering the crossover and mutation probabilities. If we increase the number of points and iterations to a large number (e.g., 500 points), the discontinuities in that region might be more visible as there may be more efficient points. By modifying the crossover and mutation probabilities, it seems the efficient frontier for $k = 2$ is now slightly closer to the efficient frontier for $k = 3$ between the return values of 0.0018 and 0.0026 (that is, we have a better result inside the range) than was apparent in Figure 20. This illustrates graphically improvements in the quality of solution that may be had through optimisation of GA parameters.



(a) Efficient and non-efficient points.

(b) Efficient points.

Figure 26: The efficient frontier for $k = 2$ using 350 points for return values between 0.0022 and 0.0026.

The crossover operator plays an essential role in the genetic algorithm process. So, modifying the crossover probability can have an impact on the quality of our solution. For example, in Figure 27, we can compare the unconstrained and $k = 2$ efficient frontier when the crossover probability is equal to zero and the mutation probability unchanged ($MP = 0.1$). Setting the crossover probability equal to zero makes it quite challenging to compute the points at the bottom of the efficient frontier. This is because the zero crossover turns the GA effectively into a hillclimber - such methods typically have issues with strong local minima. However, the top of the efficient frontier for $k = 2$ is still a good approximation of the unconstrained efficient frontier.

Earlier, we observed that by setting the crossover and mutation probability to 0.6 and 0.1, respectively, we obtained better results between the return values of 0.0018 and 0.0026. However, by changing the crossover probability to zero, the obtained results are very poor.

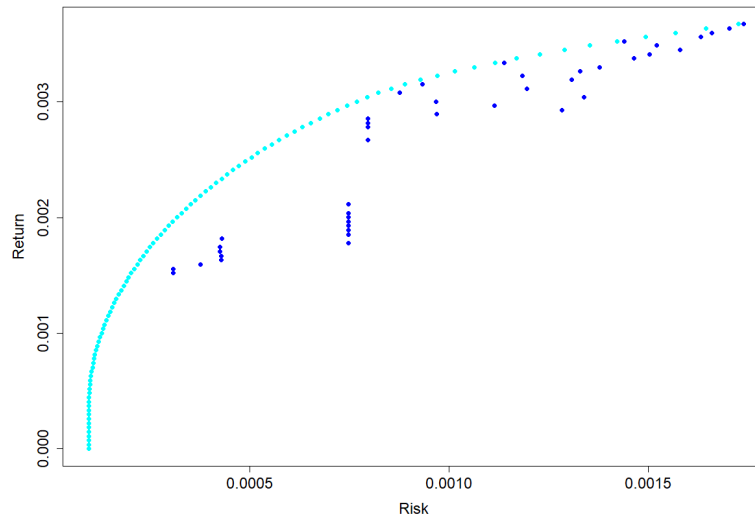
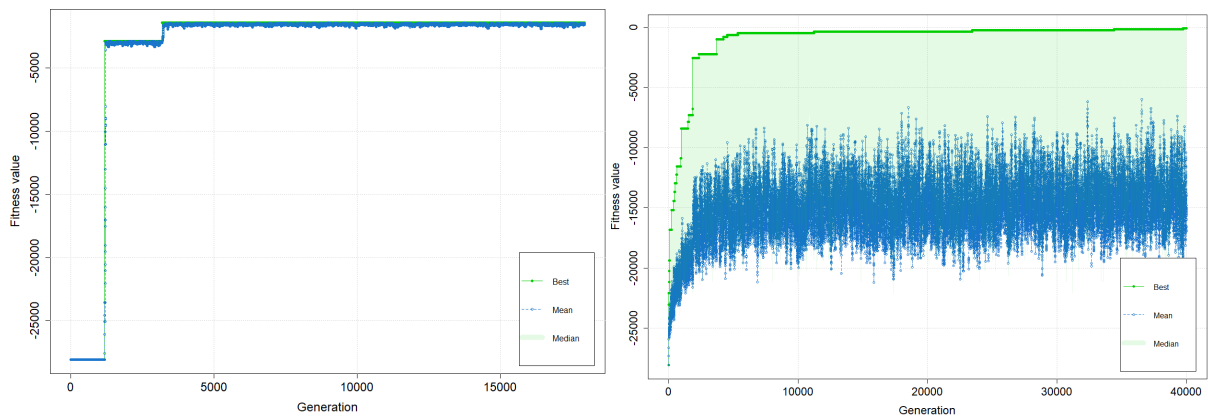


Figure 27: Unconstrained and $k = 2$ efficient frontier where $CP = 0$ and $MP = 0.1$ using 100 point

The generation-fitness plot for $k = 2$ for various crossover probabilities is shown in Figure 28. When the probability of the cross over is zero, the mean fitness value is much greater than the mean fitness value when the crossover probability is not zero. This suggests, by setting the crossover probability to zero, we will only find the points that are easy to compute and fail to find the points that are computationally very difficult to compute. Hence, it is crucial to test different values of crossover probability and select the best value.



(a) Generation-fitness plot where $CP=0$ and $MP = 0.1$ (b) Generation-fitness plot where $CP = 0.6$ and $MP = 0.1$

Figure 28: Generation-fitness plot for $k = 2$.

Conclusion and further work

Conclusion

In this paper, quadratic programming and genetic algorithm methods were applied to solve the optimal portfolio selection problem. We looked at both unconstrained and

cardinality constrained problems and obtained efficient frontiers. The unconstrained problem is straightforward as the objective function is quadratic and the constraints are linear. Hence, we can solve the problem via quadratic programming techniques. However, the cardinality constrained problem has non-linear constraints. Therefore a meta-heuristic method such as a genetic algorithm is a suitable method to use. The genetic algorithms are meta-heuristic search algorithms inspired by the theory of natural selection. The genetic algorithm has heuristic operators such as selection, crossover and mutation. It uses the operator step by step to find the optimum solutions in the portfolio optimisation problem. In the end, we tried to modify the crossover and mutation probabilities. This is important because it has a direct impact on the quality of the solutions. For example, a higher mutation probability will lead to a random search. We found that setting the crossover and mutation probabilities to 0.6 and 0.1 improves our solutions.

Further work

Many different experiments and studies have been left for the future due to a lack of time. Future work concerns a deeper analysis of other real-world constraints such as pre-assignment, class and floor-ceiling constraints using a meta-heuristic algorithm (genetic algorithm) in R. It will also be interesting to look at a different meta-heuristic algorithm and compare the results with a genetic algorithm.

Acknowledgements

I would like to thank Dr Matthew Craven for his advice and assistance throughout this project. His advice and overall insights in this field have made this an inspiring experience for me.

References

- [1] A. Baykasoğlu, M. G. Yunusoglu, and F. B. Özsoydan. A grasp based solution approach to solve cardinality constrained portfolio optimization problems. *Computers & Industrial Engineering*, 90:339–351, 2015.
- [2] D. Bertsimas and R. Cory-Wright. A scalable algorithm for sparse portfolio selection. *arXiv preprint arXiv:1811.00138*, 2018.
- [3] D. Bienstock. Computational study of a family of mixed-integer quadratic programming problems. *Mathematical programming*, 74(2):121–140, 1996.
- [4] P. Bonami and M. A. Lejeune. An exact solution approach for portfolio optimization problems under stochastic and integer constraints. *Operations research*, 57(3):650–670, 2009.
- [5] F. Cesarone, A. Scozzari, and F. Tardella. A new method for mean-variance portfolio optimization with cardinality constraints. *Annals of Operations Research*, 205(1):213–234, 2013.

- [6] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.
- [7] T.-J. Chang, S.-C. Yang, and K.-J. Chang. Portfolio optimization problems in different risk measures using genetic algorithm. *Expert Systems with applications*, 36(7):10529–10537, 2009.
- [8] T. Cura. Particle swarm optimization approach to portfolio optimization. *Nonlinear analysis: Real world applications*, 10(4):2396–2406, 2009.
- [9] K. Deep and M. Thakur. A new crossover operator for real coded genetic algorithms. *Applied mathematics and computation*, 188(1):895–911, 2007.
- [10] G.-F. Deng, W.-T. Lin, and C.-C. Lo. Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, 39(4):4558–4566, 2012.
- [11] L. Diosan. A multi-objective evolutionary approach to the portfolio optimization problem. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 183–187. IEEE, 2005.
- [12] E. J. Elton, M. J. Gruber, S. J. Brown, and W. N. Goetzmann. *Modern portfolio theory and investment analysis*. John Wiley & Sons, 2009.
- [13] D. I. Graham and M. J. Craven. An exact algorithm for small-cardinality constrained portfolio optimisation. *Journal of the Operational Research Society*, pages 1–17, 2020.
- [14] J. Heider. Quadratic programming. https://optimization.mccormick.northwestern.edu/index.php/Quadratic_programming, 2015. [Online; accessed 02-March-2021].
- [15] R. Jagdhuber, M. Lang, A. Stenzl, J. Neuhaus, and J. Rahnenführer. Cost-constrained feature selection in binary classification: adaptations for greedy forward selection and genetic algorithms. *BMC bioinformatics*, 21(1):1–21, 2020.
- [16] N. J. Jobst, M. D. Horniman, C. A. Lucas, G. Mitra, et al. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative finance*, 1(5):489–501, 2001.
- [17] K. Lwin and R. Qu. A hybrid algorithm for constrained portfolio selection problems. *Applied intelligence*, 39(2):251–266, 2013.
- [18] K. Lwin, R. Qu, and G. Kendall. A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Applied Soft Computing*, 24:757–772, 2014.
- [19] K. T. Lwin, R. Qu, and B. L. MacCarthy. Mean-var portfolio optimization: A non-parametric approach. *European Journal of Operational Research*, 260(2):751–766, 2017.

- [20] R. Mansini and M. G. Speranza. Heuristic algorithms for the portfolio selection problem with minimum transaction lots. *European Journal of Operational Research*, 114(2):219–233, 1999.
- [21] M. Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.
- [22] S. Sefiane and M. Benbouziane. Portfolio selection using genetic algorithm. 2012.
- [23] H. Soleimani, H. R. Golmakani, and M. H. Salimi. Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm. *Expert Systems with Applications*, 36(3):5058–5063, 2009.
- [24] F. Streichert, H. Ulmer, and A. Zell. Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In *Operations Research Proceedings 2003*, pages 253–260. Springer, 2004.
- [25] M. B. A. Turlach. Package ‘quadprog’. 2019.
- [26] E. Weisstein. Np-hard problem. <https://mathworld.wolfram.com/NP-HardProblem.html>, 2021. [Online; accessed 12-April-2021].