FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme/specialisation: Petroleum Engineering / Natural Gas Engineering	Spring semester, 2019 Open
Author: Sergey Kim	………………………………………….. (signature of author)
Programme coordinator: Supervisor(s): Dr. Reidar Brumer Bratvold	
Title of master's thesis: **Probabilistic Portfolio Modeling in Python**	
Credits: 30 ECTS	
Keywords: Portfolio Modeling Python Sensitivity Analysis Portfolio Optimization Optimization Algorithm Probability	Number of pages: 72 + supplemental material/other: + 2 USB flash drives Stavanger, 15th June / 2019

Title page for Master's Thesis
Faculty of Science and Technology

Acknowledgments

This thesis would not have been possible without the guidance of my supervisor Dr. Reidar Brumer Bratvold. I am grateful for his academic support and constructive comments, remarks, and suggestions. I am truly fortunate to have had the opportunity to work with him.

I would like to extend my sincere thanks to Vladislav Legkokonets, who introduced me to Portfolio Optimization, and Mark Kim for his contribution to my programming skills.

Finally, I thank my parents and my best friend Umid Amontoshev, who have supported me throughout the entire process. I am deeply thankful for their constant help and encouragement.

Abstract

The portfolio selection problem has been known for centuries. However, Markowitz (1952) was the first to introduce a robust framework for optimized portfolios on financial markets. Later this approach was applied in the petroleum industry to increase the corporate performance of oil and gas companies and to manage associated risks (Hightower et al. (1991)).

Nevertheless, despite the lack of uncertainty optimization, simple portfolio selection techniques such as the Rank and Cut method remains popular in the industry (Wood (2016)). In this thesis, the advantages and disadvantages of this approach were briefly mentioned. Besides Markowitz Portfolio Theory and the Rank and Cut Method, a number of new portfolio selection methods were developed that not only improve the performance and minimize the risks but also can be used as processes and tools to deliver shareholder value or to achieve strategic corporate goals.

One such approach is the use of multi-objective time series portfolio optimization, where the corporate goals are defined as constraints, the level of constraint accomplishment is quantified in terms of probability of exceeding the constraint and net present value is set as the main objective. This method was used to select an optimal portfolio from the pool of petroleum projects. One of the main contributions of this work is to provide a tool and process that can be used by management teams to evaluate different portfolios quickly using multiple time-dependent corporate constraints. The tool can be used to evaluate the impact on the portfolio of changing constraints or weighting the constraints differently. The ability to do this interactively is essential as it allows the management team to evaluate and address the key elements of their portfolio decision problem.

A crucial part of the portfolio optimization problem is the choice of optimization algorithms. Several algorithms that facilitate the petroleum industry's needs of portfolio optimization were studied, and a brief overview of them was presented.

We also included a discussion of the choice of programming language for portfolio models. Although we built the project model in R, we ended up using Python as it provided significant computational speed improvements over R. We also argued why Excel, although very popular, is far from an optimal tool for portfolio modeling.

Contents

Acknowledgments **i**

Abstract **ii**

List of Figures **iv**

List of Tables **vi**

List of Listings **vii**

1 Introduction **1**

2 Portfolio Optimization Methods **3**
- 2.1 The Rank and Cut Method 3
- 2.2 Markowitz's Portfolio Theory (MPT) 5
- 2.3 Naive Diversification and Risk Parity Portfolio Optimization 10
- 2.4 Time Series Multi-Objective Portfolio Optimization (TSMPO) 12

3 Optimization Algorithms **16**
- 3.1 Linear Programming 16
- 3.2 Quadratic Programming 17
- 3.3 Evolutionary Algorithms 18
- 3.4 Penalty Function 20

4 Project Model **21**
- 4.1 Class "Projects" 21
- 4.2 Oil and Gas Price 22
- 4.3 Project Simulation 25

5 Portfolio Optimization Model **35**

6 Case Study **37**
- 6.1 Mean vs P10 41
- 6.2 Sensitivity Analysis 43
- 6.3 Probability Maximization 55

7 Python for Portfolio Optimization **57**

Conclusion **60**

References **64**

List of Figures

2.2.1 Efficient Frontier (adapted from Reilly (2003)) 10
2.4.1 Oil Production of Portfolio (TSMPO) 14
2.4.2 Gas Production of Portfolio (TSMPO) 14
2.4.3 CapEx of Portfolio (TSMPO) 15
3.3.1 Genetic Algorithm 19
4.2.1 Oil Price 24
4.2.2 Oil Price Year 8 24
4.3.1 Exploration Success [Bernoulli] 27
4.3.2 Length of Appraisal Program [Multinomial] 27
4.3.3 Reserves [PERT] 28
4.3.4 NPV Distribution 32
4.3.5 Development Well Forecast 33
4.3.6 Development Well Distribution Year 8 33
4.3.7 Mean Annual Production Forecast 34
4.3.8 Annual Production Distribution Year 8 34
6.1.1 Mean Optimization 41
6.1.2 P10 Optimization 42
6.2.1 Tornado Diagram 45
6.2.2 Gas Production Year 1 - Gas Production Year 2 48
6.2.3 G Prod Y2 Projection 48
6.2.4 G Prod Y1 Projection 48
6.2.5 Gas Production Year 3 - Gas Production Year 4 49
6.2.6 G Prod Y4 Projection 49
6.2.7 G Prod Y3 Projection 49
6.2.8 Gas Production Year 3 - CapEx Year 1 50
6.2.9 CapEx Y1 Projection 50
6.2.10 G Prod Y3 Projection 50
6.2.11 Gas Production Year 3 - NCF Year 1 51
6.2.12 NCF Y1 Projection 51
6.2.13 G Prod Y3 Projection 51
6.2.14 Gas Production Year 4 - CapEx Year 1 52
6.2.15 CapEx Y1 Projection 52
6.2.16 G Prod Y4 Projection 52
6.2.17 Gas Production Year 4 - NCF Year 1 53
6.2.18 NCF Y1 Projection 53
6.2.19 G Prod Y4 Projection 53
6.2.20 CapEx Year 1 - NCF Year 1 54
6.2.21 NCF Y1 Projection 54
6.2.22 CapEx Y1 Projection 54
6.3.1 Optimized Portfolios with respect to Probability of exceeding Gas Production Goal for Year 4 and constrained by NPV 55
7.0.1 Number of Active R Packages available on CRAN (Muenchen (2019)) 57
7.0.2 Number of Active Python Packages available on PyPi (adapted from Becker (2019)) 57

7.0.3 Programming Language Rankings 2019 (adapted from O'Grady (2019)) 58

List of Tables

2.1.1 Project Pool for the Rank and Cut Method 3
2.1.2 The Rank and Cut Method Applied (ranking with respect to Expected NPV) 4
2.1.3 The Rank and Cut Method Applied (ranking with respect to Capital Efficiency) 4
4.1.1 Input Parameters of Class "Projects" 22
4.2.1 Parameters for Oil and Gas Price Calculation 23
4.3.1 Parameters Constant Along the Timeline 25
4.3.2 Uncertainty Factors as Function of Project Phase 29
4.3.3 Most Likely Values as Function of Project Location 30
4.3.4 Values as Function of Project Reserves 31
4.3.5 Time-Series Parameters 32
6.0.1 Corporate Constraints Along the Timeline 37
6.0.2 NPV, Oil and Gas Reserves 38
6.0.3 Oil and Gas Production 39
6.0.4 CapEx and NCF 40
6.2.1 Portfolio Weights. Gas Production Year 1 46
6.2.2 Portfolio Weights. Gas Production Year 2 46
6.2.3 Portfolio Weights. Gas Production Year 3 46
6.2.4 Portfolio Weights. Gas Production Year 4 47
6.2.5 Portfolio Weights. CapEx Year 1 47
6.2.6 Portfolio Weights. NCF Year 1 47
6.3.1 Portfolio Weights of Gas Projects 56

List of Listings

4.1.1 Class "Projects" Definition 21
4.2.1 Oil and Gas Price Function 23
4.3.1 Project Simulation Method 25
5.0.1 Class "Portfolio Pool" Initiation 35
5.0.2 Objective Function Method 35
5.0.3 Constraint Method 35
5.0.4 Portfolio Optimization. QP Method 36
5.0.5 Portfolio Optimization. DE Method 36

1 Introduction

Diversification for a long time was considered as a desirable property of a good investment. In the early 18th century Daniel Bernoulli stated: "it is advisable to divide goods which are exposed to some danger into several portions rather than to risk them all " (Bernoulli (1954)). In addition, he gave an example that having possesses in foreign countries and given the fact that every tenth ship goes down, in terms of the expectation of commodities, it is beneficial to transport them by two different ships.

D. Bernoulli nevertheless was not the first one who showed advantages of diversification. Rubinstein (2002) has mentioned The Merchant of Venice, a play written by William Shakespeare, as another illustration of this concept:

> *"My ventures are not in one bottom trusted,*
> *Nor to one place; nor is my whole estate*
> *Upon the fortune of this present year:*
> *Therefore my merchandise makes me not sad."*
>
> Act I, Scene 1

The superiority of diversification comes from uncertainty mitigation of future outcomes, and in order to fully understand the economic risk, one should quantify it. Variance as a measure of risk was first proposed by Fisher (1906); Tobin (2006) suggested variance of return as investment portfolio risks quantification. In 1949 a well-known investor Graham (1949) chose a margin of safety to be a measure of risk and adviced to look for undervalued companies based on their historical performance. He also recommended diversification of investments to diminish uncertainties (S. Mitra (2009)).

The traditional method of portfolio selection is to allocate capital by ranking and funding projects in decreasing order of some economic measure (net present value (NPV), internal rate of return, discounted profit-to-investment ratio, etc.) until the budget is exhausted (Erdogan et al. (2007)). This method is known as "rank and cut" approach and is commonly used in Exploration and Production (E&P) industry (Lessard (2003)). A detailed review of the technique can be found in section 2.1.

Although concepts of investment risks and diversification were well known among financial economists, a clear mathematical framework of portfolio analysis was not well established until Harry Markowitz proposed his Portfolio Theory in 1952 (S. Mitra (2009)). It was a significant breakthrough in the theory of financial economics, and later in 1990 he was awarded Alfred Nobel Memorial Prize in Economic Sciences for having developed the theory of portfolio choice. This approach will be discussed later in section 2.2.

As it was stated earlier diversification, as a mean of managing risk, is a key property of an optimized portfolio. Selection of equal shares of each of the security is the easiest way to achieve it. This technique was suggested by DeMiguel et al. (2009) as a rule in which a fraction $1/N$ is allocated equally to each of the N securities (section 2.3).

Another approach is time series multi-objective portfolio optimization (TSMPO), where various constraints are set along the future timeline to facilitate established corporate strategy. Instead of portfolio variance, probabilities of constraint accomplishment are chosen as measures of risk. This approach is widely known in the petroleum industry and has been discussed in multiple papers: Howell et al. (2001); DuBois (2007); Faya et al. (2007). Further description of this portfolio optimization technique will be presented in section 2.4.

Optimization algorithm that was chosen to solve the problem beyond its application provide suboptimal results. In the case of portfolio optimization, poor selection of the optimization algorithm may lead to significant potential losses. To address this issue an overview of optimization algorithms regarding the portfolio selection problem and their application areas has been presented in chapter 3.

This research has two main contributions. The first is the use of Python programming language for portfolio modeling. TSMPO has been chosen to reflect the needs of the petroleum industry. We applied this method to develop a Python-based portfolio optimization model that was described in chapter 5. Project model (chapter 4) that provides data to portfolio optimization model was built separately in Python and in R. In chapter 7 we discussed the advantages of these programming languages and compared them to Microsoft Excel.

Literature overview of TSMPO method applied to a pool of petroleum projects has demonstrated a lack of processes and tools that describe the results of optimized portfolios. This was the motivation for the second contribution of this work. In chapter 6 we provided a case study and in section 6.2 presented a method to determine the most significant corporate constraints that impact the optimized portfolios.

2 Portfolio Optimization Methods

2.1 The Rank and Cut Method

The Rank and Cut approach is a basic optimizer imposing an objective set by the decision-maker. It's routinely applied across E&P industry, in particular concerning budget limitations (e.g., capital investment).

The algorithm of portfolio selection is described by Wood (2016):

1. Establish the objective metric to be used to rank the projects, normally an objective function to optimize (e.g., NPV, discounted profit-to-investment ratio, etc.).

2. Rank the projects and order them according to the significance of their contribution with respect to the chosen objective metric.

3. Select the constraint.

4. Select and accumulate the projects, starting with the ranked #1 project and consecutively adding projects one by one in decreasing order (i.e., rank#2, rank#3, etc.) until the constraint limit is reached.

5. In reaching the constraint limit, only a fraction of the project is selected, so that the constraint limit is not exceeded.

The method is applied to the project pool shown in table 2.1.1, where $E[NPV]$: expected NPV; $CapEx$: capital expenditures; $CapitalEfficiency$: a ratio $E[NPV]/CapEx$.

Table 2.1.1: Project Pool for the Rank and Cut Method

Project	**E[NPV]** *USD (million)*	**CapEx** *USD (million)*	**Capital Efficiency**
1	917	556	1.65
2	449	780	0.58
3	2232	1743	1.28
4	1388	889	1.56
5	929	387	2.40
6	1480	507	2.92
7	1629	1397	1.17
8	2117	729	2.90
9	254	153	1.66
10	673	296	2.27
11	1163	965	1.21
12	875	324	2.70

Table 2.1.2: The Rank and Cut Method Applied (ranking with respect to Expected NPV)

Project	**E[NPV]**	**CapEx**	**Capital Efficiency**	**Selected**
	USD (million)	*USD (million)*		%
3	2232	1743	1.28	100
8	2117	729	2.90	100
7	1629	1397	1.17	100
6	1480	507	2.92	100
4	1388	889	1.56	13.9
11	1163	965	1.21	0
5	929	387	2.40	0
1	917	556	1.65	0
12	875	324	2.70	0
10	673	296	2.27	0
2	449	780	0.58	0
9	254	153	1.66	0

Table 2.1.3: The Rank and Cut Method Applied (ranking with respect to Capital Efficiency)

Project	**E[NPV]**	**CapEx**	**Capital Efficiency**	**Selected**
	USD (million)	*USD (million)*		%
6	1480	507	2.92	100
8	2117	729	2.90	100
12	875	324	2.70	100
5	929	387	2.40	100
10	673	296	2.27	100
9	254	153	1.66	100
1	917	556	1.65	100
4	1388	889	1.56	100
3	2232	1743	1.28	37.8
11	1163	965	1.21	0
7	1629	1397	1.17	0
2	449	780	0.58	0

The projects are ranked with respect to expected NPV and sorted. The budget is set to 4500 USD (million), and the projects are selected based on the ranking (table

2.1.2). Accumulated capital expenditures of the four best projects are equal to 4376 USD (million), the remaining 124 USD (million) were allocated to project #4, which resulted in a share of 13.9%. The same procedure can be applied to Capital Efficiency as the objective function.

Unsurprisingly the rank and cut method based on capital efficiency does not produce the same results when ranking is done with respect to expected NPV (table 2.1.3). Projects # 1, 5, 9, 10, 12 are part of the portfolio, yet they were not selected by expected NPV ranking. The same applies to project #7 that is dropped from the portfolio in the case of capital efficiency ranking.

Moreover, the rank and cut method does not account for risk. Erdogan et al. (2007) showed that the technique leads to a higher risk in comparison to optimization approaches that explicitly take risk into account. The rank and cut furthermore fails to select projects that are subject to multiple objective functions and/or several limit constraints. Despite the aforementioned disadvantages, the method is very popular and can be used as a benchmark or starting point of optimization algorithms.

2.2 Markowitz's Portfolio Theory (MPT)

Before presenting MPT, it's necessary to introduce some definitions and assumptions. Uncertainty and risk are often used interchangeably, and this was illustrated in chapter 1. However, there is a noticeable difference in the formal definition.

Definition 2.2.1. Uncertainty is a subjective aspect and represents a lack of knowledge regarding the statement (Bratvold et al. (2010)).

Definition 2.2.2. Risk is an undesirable consequence of uncertainty (Bratvold et al. (2010)).

In the subsequent sections, this distinction between the terms of risk and uncertainty will be preserved.

The assumptions of the Markowitz model are based on the behavior of investor (Reilly (2003)):

- The measure of the uncertainty of the portfolio is the variance of expected returns.
- Investors decide solely on the basis of uncertainty and expected returns.
- For a given expected return, investors prefer less uncertainty to more uncertainty. Correspondingly, for a given level of uncertainty, investors select higher expected returns to lower.

On the aforementioned support, Markowitz (1952) stated that a single portfolio is efficient if no other portfolio with fixed uncertainty level has higher expected returns, or, similarly, no other portfolio with fixed expected returns has lower uncertainty. In addition, he performed a geometric analysis of three- and four-security pools to demonstrate the properties of efficient sets. The set of efficient mean-variance combinations of portfolios defined by Markowitz (1952) became known as the efficient frontier.

Later Markowitz (1959) derived and further defined an expected value - variance (E-V) efficient portfolios.

Stock Market vs. E&P Industry

Markowitz theory was developed to be applied to securities on the stock markets. Later Hightower et al. (1991) implemented the E-V portfolio optimization to find the optimized portfolio of petroleum assets.

We, therefore, would like to emphasize the key differences between stock and E&P investments listed by Ball et al. (1999):

- Stock portfolios depend only on uncertain returns. E&P projects face both local uncertainties (e.g., the discovery and production of oil at a given site), and global uncertainties (e.g., prices, politics, etc.). Moreover, stock returns' uncertainties typically have a bell-shaped distribution while E&P uncertainties are highly skewed and stress rare events.
- Uncertainty in stock portfolios is commonly measured in terms of volatility (variance). E&P portfolios must specifically get use of risk.
- E&P projects pay out over long periods. Stocks can be bought and sold at will.
- Stock portfolios mostly contain investments with a small fraction of shares. E&P portfolios, on the other hand, include projects with participation of 100%.

As it was stated in the "Stock Market vs. E&P Industry" note, Markowitz' portfolio optimization can be applied to petroleum assets; thus, the E-V optimization will be explained with respect to petroleum assets.

For a set of n assets and time period of t, every asset has a vector of returns (R) which is a vector of realizations drawn from the identified distribution:

$$R_1 = \begin{bmatrix} R_{11} \\ R_{12} \\ \vdots \\ R_{1t} \end{bmatrix}, \quad R_2 = \begin{bmatrix} R_{21} \\ R_{22} \\ \vdots \\ R_{2t} \end{bmatrix}, \quad \cdots, \quad R_n = \begin{bmatrix} R_{n1} \\ R_{n2} \\ \vdots \\ R_{nt} \end{bmatrix} \tag{2.2.1}$$

Given this historical data, one can calculate means (μ) and variances (σ^2) of returns for each of the asset:

$$\mu_1 = \frac{1}{t}\sum_{j=1}^{t} R_{1j}, \;\; \mu_2 = \frac{1}{t}\sum_{j=1}^{t} R_{2j}, \;\; \cdots, \;\; \mu_n = \frac{1}{t}\sum_{j=1}^{t} R_{nj} \tag{2.2.2}$$

$$\sigma_1^2 = \frac{1}{t}\sum_{j=1}^{t}(R_{1j}-\mu_1)^2, \;\; \sigma_2^2 = \frac{1}{t}\sum_{j=1}^{t}(R_{2j}-\mu_2)^2, \;\; \cdots, \;\; \sigma_n^2 = \frac{1}{t}\sum_{j=1}^{t}(R_{nj}-\mu_n)^2, \tag{2.2.3}$$

Then, based on the linearity of expectations, the expected value of portfolio return P can be written as:

$$E[P] = \sum_{i=1}^{n}(w_i R_i) \tag{2.2.4}$$

where w_i is a corresponding share of selected asset i.

However, variance doesn't have the same linearity property as the expected value. In order to calculate the variance of the portfolio returns the following formula should be used:

$$Var(P) = \sum_{i=1}^{n}(w_i \cdot \sigma_i^2) + 2\sum_{i=1,k=1}^{n}(w_i \cdot w_k \cdot cov(R_i, R_k)) \tag{2.2.5}$$

where $cov(R_i, R_k)$ is the covariance of R_i and R_k and is given by equation:

$$cov(R_i, R_k) = E[(R_i - \mu_i)(R_k - \mu_k)] \tag{2.2.6}$$

Presenting expected return $\boldsymbol{\mu}$ and weights $\boldsymbol{w}$ of n assets as vectors of form $n \times 1$:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}, \;\; \boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \tag{2.2.7}$$

and portfolio variance $Var(P)$ as covariance matrix $\boldsymbol{\Sigma}$:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{bmatrix} \tag{2.2.8}$$

portfolio expected return $E[P]$ and portfolio variance $Var(P)$ can be written as:

$$E[P] = \boldsymbol{\mu}^T \times \boldsymbol{w} \tag{2.2.9}$$

$$Var(P) = \boldsymbol{w}^T \times \boldsymbol{\Sigma} \times \boldsymbol{w} \tag{2.2.10}$$

Correlation Coefficient

From the equation (2.2.5) it can be concluded, that in order to minimize the variance of portfolio, not only variances of each of the assets should be considered, but also covariance between returns of the assets. Yet, covariance interpretation may not be intuitive, so a correlation coefficient can be used instead:

$$corr(R_i, R_k) = \frac{cov(R_i, R_k)}{\sigma_i^2 \sigma_k^2}$$

Correlation coefficient can take values in the range $[-1, 1]$. This set can be divided into 3 cases:

1. R_i and R_k are independent: $corr(R_i, R_k) = 0$
2. R_i and R_k are positevely correlated: $corr(R_i, R_k) > 0$. Meaning: increase of R_i implies increase of R_k and vice-versa.
3. R_i and R_k are negatively correlated: $corr(R_i, R_k) < 0$. Meaning: increase of R_i implies decrease of R_k and vice-versa.

It was showed that correlation coefficient provides a better interpretation of projects interaction. Nonetheless, in the derivation of E-V problem we will stick to covariance.

It's important to emphasize that portfolio return is the measure of uncertainty (see definition of uncertainty 2.2.1). Markowitz (1959) defined the semi-variance as an alternative measure of risk (notice the definition of risk 2.2.2).

Now, having defined the expected return of portfolio and portfolio variance, E-V portfolio optimization problem can be represented as follows:

- Portfolio expected return maximization with a portfolio variance as an upper constraint:

$$\max_{w \in W} \boldsymbol{\mu}^T \times \boldsymbol{w} \tag{2.2.11}$$

- Portfolio variance minimization with a portfolio expected return as a lower constraint:

$$\min_{w \in W} \boldsymbol{w}^T \times \boldsymbol{\Sigma} \times \boldsymbol{w} \tag{2.2.12}$$

- Alternatively, taking in account risk aversion coefficient λ:

$$\max_{w \in W} \boldsymbol{\mu}^T \times \boldsymbol{w} + \lambda \times \boldsymbol{w}^T \times \boldsymbol{\Sigma} \times \boldsymbol{w} \tag{2.2.13}$$

Risk Aversion Coefficient

The von Neumann-Morgenstern expected utility theory (see Von Neumann et al. (2007)) is commonly applied to find the optimized portfolio. The weights of the optimized portfolio maximize the expected utility of future wealth as well. The problem then is to find a functional form of the utility function. One of such functions is the exponential utility given by:

$$U_{exp}(R_w) = 1 - e^{-\gamma_{exp} R_w}$$

where
R_w: the return of the portfolio $R_w = \boldsymbol{R}^T \times \boldsymbol{w}$ (see equation 2.2.11)
γ_{exp}: the risk aversion coefficient, which is a parameter that defines the decision maker attitude towards risk.

If the asset returns are multivariate normally distributed then the maximization of $E(U_{exp}(R_w))$ is equal to maximization of mean-variance utility function (Okhrin et al. (2008)):

$$\boldsymbol{\mu}^T \times \boldsymbol{w} + \frac{\gamma_{mv}}{2} \times \boldsymbol{w}^T \times \boldsymbol{\Sigma} \times \boldsymbol{w}$$

where γ_{mv} is the risk aversion coefficient.

If $\frac{\gamma_{mv}}{2}$ is substituted with λ we get the equation 2.2.13. The quantification of the risk aversion coefficient is not obvious, and we would like to refer the reader to a paper written by Bodnar et al. (2018) where the problem of its estimation was discussed.

The problem of E-V portfolio optimization lies in the field of quadratic programming algorithms that are described in section 3.2.

Finally, constructed portfolios can be plotted with portfolio variance on the x-axis, and portfolio expected return on the y-axis (figure 2.2.1). The scatter plot represents the set of possible portfolios, and the curved line is an efficient frontier. From the plot, it's clearly seen that for a given expected return (line $A - C$) no other portfolio has lower variance, than point A. Equivalently, for a given variance (line $B - C$) no other portfolio has a higher expected return than that on the efficient frontier: point B.

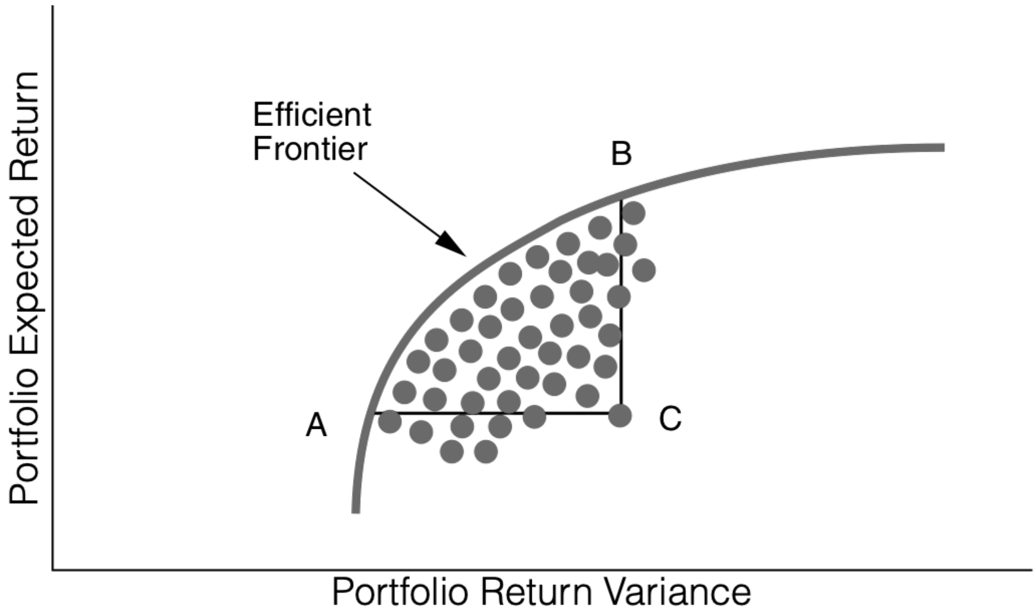

Figure 2.2.1 : Efficient Frontier (adapted from Reilly (2003))

2.3 Naive Diversification and Risk Parity Portfolio Optimization

In their work DeMiguel et al. (2009) mentioned the naive diversification as a simple and therefore widespread method to select an optimal portfolio. The $1/N$ strategy was chosen to be a benchmark in a series of comparisons to other optimizing models. Out of all the chosen techniques, the minimum-variance portfolio showed the best results in terms of Sharpe ratio (see definition 2.3.1).

Definition 2.3.1. Sharpe ratio S of portfolio is stated as follows(Reilly (2003)):

$$S = \frac{\bar{R} - \overline{RFR}}{\sigma} \tag{2.3.1}$$

where: $\bar{R}$ = the average rate of return for portfolio; $\overline{RFR}$ = the average rate of return for risk-free assets; σ = the standard deviation of the rate of return for portfolio.

On the basis of simulation results, DeMiguel et al. (2009) defined three required conditions of the optimizing models to outperform naive diversification:

1. Long estimation window.

2. True Sharpe ratio of the E-V efficient portfolio is considerably higher than that of the naive diversification.

3. Number of assets in the pool is small.

An obvious reason for the advantages of the $1/N$ method is that a very long time series of data is needed to estimate the consistent vector of expected returns and variance-covariance matrix of returns for mean-variance method. Resulting portfolio weights of these techniques are less optimal in comparison to naive diversification in a way that errors caused by the $1/N$ method are smaller (DeMiguel et al. (2009)).

Finally, DeMiguel et al. (2009) recommended choosing the naive diversification as a natural benchmark for its simplicity and low-cost implementation.

The post-Markowitz era gave birth to numerous measures of risk as well. In the overview of recent advances in portfolio theory, Kolm et al. (2014) listed risk contribution, marginal risk contribution and relative risk contribution.

Risk contribution has different definitions. J.P Morgan (1997) specified it as follows. Let's consider portfolio, where asset $\#i$ is removed. Then portfolio weights have the following representation:

$$\boldsymbol{w}_{-i} = \begin{bmatrix} w_1 \\ \vdots \\ w_{i-1} \\ 0 \\ w_{i+1} \\ \vdots \\ w_n \end{bmatrix} \tag{2.3.2}$$

The risk contribution σ_i of asset i in portfolio w thus is defined as:

$$\sigma_i(w) = \sigma(w) - \sigma(w_{-i}) \tag{2.3.3}$$

However, the asset-wise sum of risk contributions is not equal to the standard deviation of portfolio returns, which makes this approach less intuitive.

Alternatively, Kolm et al. (2014) advised to use risk contribution derived on the basis of marginal risk contribution $MRC_i(w)$, determined as a partial derivative of portfolio standard deviation $\sigma(w)$ with respect to weight w_i of selected asset i:

$$MRC_i(w) = \frac{\partial \sigma(w)}{\partial w_i} = \frac{(\Sigma \times w)_i}{\sigma(w)} \tag{2.3.4}$$

Then, risk contribution RC_i of asset i is calculated as weighted marginal risk contribution:

$$RC_i(w) = w_i \cdot MRC_i(w) \tag{2.3.5}$$

As opposed to the definition of J.P Morgan (1997) the analogue (2.3.5) defined by Kolm et al. (2014) has nice property such that:

$$\sum_{i=1}^{n} RC_i(w) = \sum_{i=1}^{n} \frac{w_i \cdot (\Sigma w)_i}{\sigma(w)} = \frac{\boldsymbol{\sigma}^T \boldsymbol{\Sigma} \boldsymbol{w}}{\sigma(w)} = \sigma(w) \tag{2.3.6}$$

Finally, relative risk contribution RRC_i is specified as a fraction of risk contribution RC_i of asset i and portfolio standard deviation $\sigma(w)$:

$$RRC_i(w) = \frac{RC_i(w)}{\sigma(w)} \tag{2.3.7}$$

On the basis of relative risk contribution RRC_i Asness et al. (2012) suggested a risk parity portfolio optimization method. In this approach, the optimal portfolio is constructed so that risk is evenly allocated between all the selected assets. To achieve this, the portfolio should satisfy the only condition:

$$RC_i(w) = \frac{\sigma(w)}{n} \tag{2.3.8}$$

To conclude, the naive diversification method and risk-parity method follows quite similar principles: an equal allocation of capital and total risk, respectively.

2.4 Time Series Multi-Objective Portfolio Optimization (TSMPO)

All oil and gas companies have corporate strategies that are explicitly defined in the form of goals, or implicitly in the form of historical business practices (Howell et al. (2001)). In the explicit case, a portfolio selection problem to achieve multiple goals, that are distributed across a future timeline, is known as time series multi-objective portfolio optimization. These goals are interpreted as constraints that are imposed on portfolio optimization.

To illustrate the process of portfolio optimization, let's consider the following set up. The corporate strategy consists of three constraints: oil production, gas production, and capital expenditures. Each of the constraints is set annually for m-year periods, and the NPV is defined as the main objective function. The problem set-up is as follows:

$$\max_{w} \sum_{i=1}^{n} NPV_i \cdot w_i \tag{2.4.1}$$

subject to:

$$\sum_{i=1}^{n} Q_{i,j}^{oil} \cdot w_i \geq C_j^{oil} \tag{2.4.2}$$

$$\sum_{i=1}^{n} Q_{i,j}^{gas} \cdot w_i \geq C_j^{gas} \tag{2.4.3}$$

$$\sum_{i=1}^{n} CapEx_{i,j} \cdot w_i \geq C_j^{CapEx} \tag{2.4.4}$$

$$0 \leq w_i \leq 1 \tag{2.4.5}$$

where:
$i = 1, ..., n$: project #
$j = 1, ..., m$: year #
w_i: portfolio weight for project #i
NPV_i: NPV for project #i
$Q_{i,j}^{oil}$: oil production of year #j for project #i
C_j^{oil}: oil production constraint for year #j
$Q_{i,j}^{gas}$: gas production of year #j for project #i
C_j^{gas}: gas production constraint for year #j
$CapEx_{i,j}$: CapEx of year #j for project #i
C_j^{CapEx}: CapEx constraint for year #j

Project metrics (NPV_i, $Q_{i,j}^{oil}$, $Q_{i,j}^{gas}$, $CapEx_{i,j}$) can be deterministic or probabilistic. In the probabilistic case, the parameters are defined as expected values of the corresponding probability density function/probability mass function or quantiles (e.g., P90, P50, P10) with respect to risk preferences of the decision-maker. Probability of constraint accomplishment is set as uncertainty measure.

The aforementioned problem is optimized using linear programming (LP) algorithms (section 3.1), as all project metrics have linear properties. However, in the highly constrained setup, when it's not possible to achieve at least one of the constraints, LP will fail to find a feasible solution. Penalty functions (section 3.4) help to overcome this issue. The results of optimization are presented on figures 2.4.1 , 2.4.2 , 2.4.3 , where constraints are depicted as red bars, means of the optimized portfolio as green bars, and probabilities of achieving constraints as blue lines.

Uncertainty minimization problem with respect to a certain constraint lays in the field of quadratic programming (section 3.2) or, in the case of non-convex/non-concave - evolutionary algorithms (section 3.3). The setup where the probability of exceeding oil production constraint for year 5 is the objective function, and portfolio NPV is constrained to be greater or equal to C^{NPV}, has the following form:

$$\max_{w} P\left[\sum_{i=1}^{n}(Q_{i,5}^{oil} \cdot w_i) \geq C_5^{oil}\right] \tag{2.4.6}$$

subject to

$$\sum_{i=1}^{n} NPV_i \cdot w_i \geq C^{NPV} \tag{2.4.7}$$

The TSMPO approach provides a way to set and evaluate corporate strategies (Howell et al. (2001)). It takes into account uncertainties in terms of probabilities of exceeding constraints, which is a more intuitive measure than variance or standard deviation.

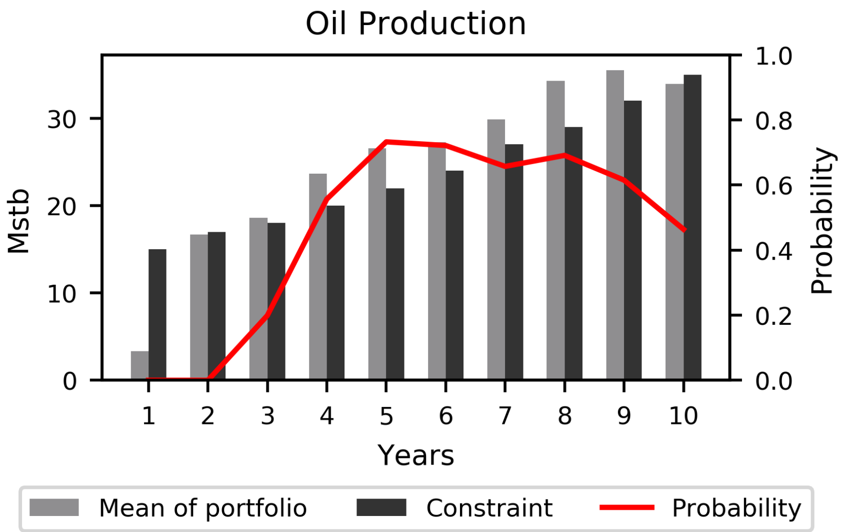

Figure 2.4.1 : Oil Production of Portfolio (TSMPO)

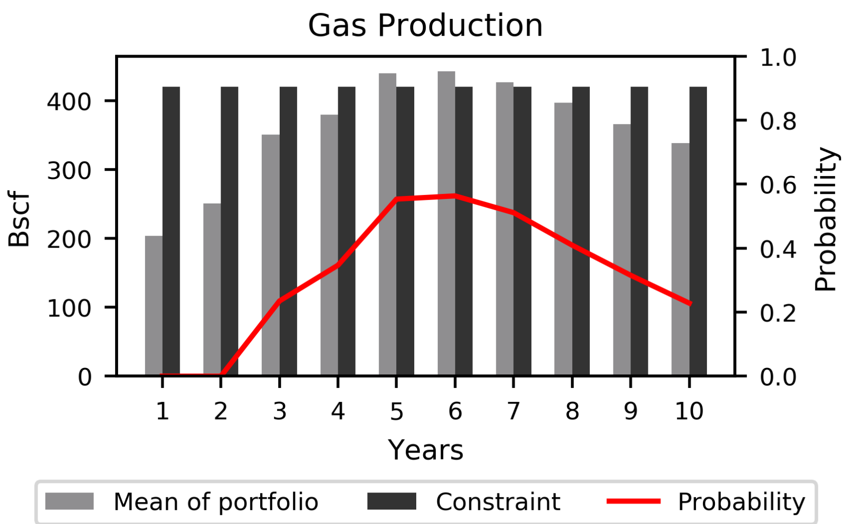

Figure 2.4.2 : Gas Production of Portfolio (TSMPO)

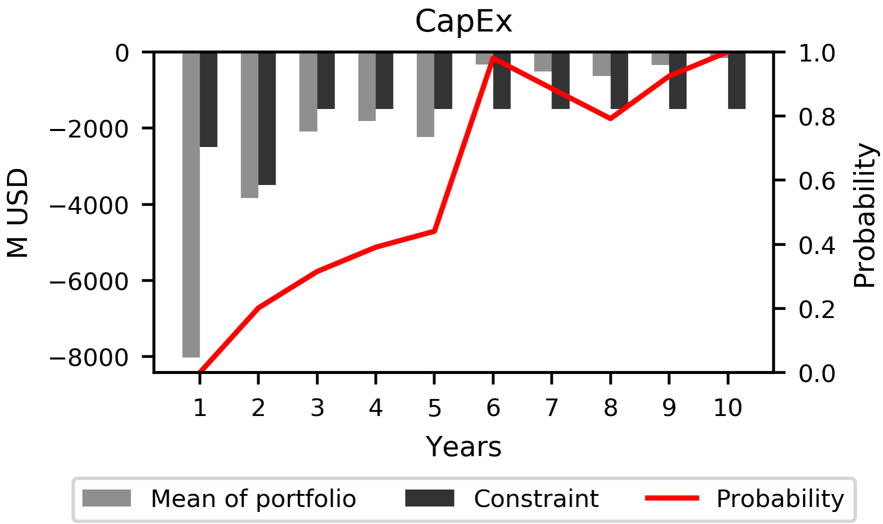

Figure 2.4.3 : CapEx of Portfolio (TSMPO)

3 Optimization Algorithms

3.1 Linear Programming

Linear programming (LP) optimizers are widely applied in the petroleum industry. The reason for that is performance metrics such as revenues, costs, cash flows, oil production, gas production which are aggregated linearly (Wood (2016)).

LP is a mathematical method to solve constrained optimization problems in the following form (Lessard (2003)):

$$\min_{x} \boldsymbol{p}^T \boldsymbol{x} \tag{3.1.1}$$

subject to:

$$\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{l} \tag{3.1.2}$$

$$\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b} \tag{3.1.3}$$

where:
$\boldsymbol{p}$: vector of linear coefficients of the objective function;
$\boldsymbol{x}$: decision variables vector (e.g., vector of portfolio weights);
$\boldsymbol{A}$: matrix of the inequality constraint coefficients;
$\boldsymbol{l}$: vector of coefficients imposed as bounds on Ax;
$\boldsymbol{C}$: matrix of the equality constraint coefficients;
$\boldsymbol{b}$: vector of coefficients imposed as equality constraint on Cx

Equation (3.1.1) is called an objective function; term (3.1.2) is defined as inequality constraint, and term (3.1.3) as equality constraint.

A well-known method to solve LP problems is simplex approach (Dantzig et al. (1955)).

However, LP algorithms fail to optimize the original Markowitz E-V portfolio, which is a quadratic programming problem. This comes from the fact that variance aggregation is not linear unless all selected assets are considered independent (see equations 2.2.5, 2.2.6, 2.2).

LP optimization can be easily applied in the petroleum industry-related problem:

$$\min_{x} \sum_{i=1}^{n} NPV_i \cdot w_i \tag{3.1.4}$$

$$\sum_{i=1}^{n} Q_i \cdot w_i \geq c \tag{3.1.5}$$

$$\sum_{i=1}^{n} w_i = 1 \tag{3.1.6}$$

where NPV_i is net present value and Q_i is oil production.

A number of Python and R packages using LP solvers have been developed: lp-Solve (Berkelaar et al. (2015)); APMonitor (Hedengren et al. (2014)); SciPy (Jones et al. (2001–)), etc.

However, the LP problem has numerous issues :

1. The majority of uncertainty or risk measures are not optimized unless they are linearly aggregated (some examples of LP computable risk and uncertainty measures are mentioned by Mansini et al. (2014))
2. Algorithm fails to optimize non-concave/non-convex functions
3. Algorithm may not find a solution in case of highly constrained set up

The listed disadvantages and shortcomings of LP problem are partially addressed by quadratic programming (see item 1 in the list of the disadvantages of LP) and are completely overcome by genetic algorithm along with introduction of the penalty function, which will be discussed in the next sections.

3.2 Quadratic Programming

As it was stated in section 3.1, the problem defined by Markowitz (1959) is a quadratic programming problem (QP). The general definition of the problem is as follows (G. Mitra et al. (2007)):

$$\min_x \boldsymbol{p}^T \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} \tag{3.2.1}$$

subject to:

$$\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{l} \tag{3.2.2}$$

$$\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b} \tag{3.2.3}$$

where $\boldsymbol{Q}$ is the symmetric matrix of the objective function coefficients, and other terms are defined as in the LP formulation (3.1).

Quadratic programming has a very nice property (Cottle et al. (2010)):

Theorem 3.2.1. *If LP problem:*

$$\min_x \boldsymbol{p}^T \boldsymbol{x}$$

subject to:

$$\begin{aligned} \boldsymbol{C}\boldsymbol{x} &= \boldsymbol{b}, \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$

has an optimal solution, then for every $\lambda \geq 0$, so does the QP problem:

$$\min_{x} \lambda \boldsymbol{p}^T \boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x}$$

subject to:

$$\begin{aligned} \boldsymbol{C}\boldsymbol{x} &= \boldsymbol{b}, \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$

The possibility to optimize **non-linear functions (i.e. uncertainty or risk measure) and theorem 3.2.1** make quadratic programming a good choice. The technique has been included in multiple Python and R packages: APMonitor (Hedengren et al. (2014)); SciPy (Jones et al. (2001–)); quadprog (Berwin et al. (2019)); Rmosek (Friberg (2019)), etc.

Despite its advantages relative to LP solutions, QP fails to optimize a highly constrained problem and fails to find a global maximum/minimum in case of optimization of non-convex/non-concave functions.

3.3 Evolutionary Algorithms

The genetic algorithm (GA) was inspired by Darwin's principle of survival of the fittest and was first discussed by Holland (1975). In GA, a randomly defined initial population undergoes the process of selection. Each individual (portfolio) has a chromosome subdivided into genes, which in the case of portfolio optimization are interpreted as asset weights. Selection is done with respect to fitness function $f(x)$:

$$f(x) = \boldsymbol{p}^T \boldsymbol{x} \tag{3.3.1}$$

where:
$\boldsymbol{p}$: vector of linear coefficients of the objective function;
$\boldsymbol{x}$: decision variables vector (e.g., portfolio weights);

Individuals that were chosen on the selection stage are called parents. They form pairs and proceed to the crossover step, where their offsprings inherit genes from parents. On the mutation stage offsprings are randomly altered. Final generation is composed of parents and offsprings. If the termination condition is reached, the algorithm stops, otherwise, all steps are repeated with the previous generation as the initial generation for the next step. Figure 3.3.1 illustrates the algorithm.

Selection, crossover and mutation stages filter individuals, thus the initial population will converge to the optimal solution. The stages can be adjusted and modified: one can set the genes ratio of parent 1 to parent 2 inherited by offspring; mutation can be set to follow random gaussian distribution, random uniform distribution or any other distribution; the magnitude of the alteration can be defined as a parameter of the probability density function, etc.

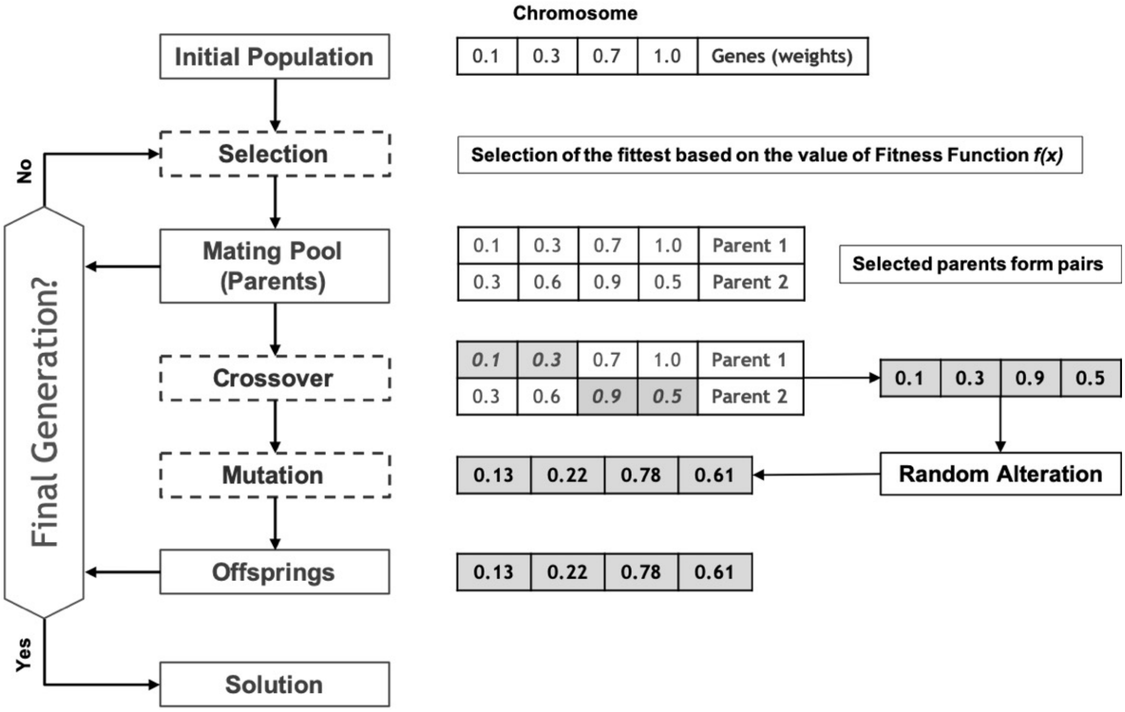

Figure 3.3.1 : Genetic Algorithm

Another evolutionary optimization method is differential evolution (DE) defined by Storn et al. (1995). Keller et al. (2017) stated that the differences between DE and GA are in the methods of crossover and mutation schemes. They also briefly reviewed the algorithm of differential evolution. While genetic algorithm applies addition or multiplication by random variable drawn from a distribution, the differential evolution approach performs mutation by adding the weighted difference of chromosomes of two randomly selected individuals to the third one. The crossover between mutated individuals and individuals from the past generation is executed, followed by the selection step to produce the generation of offsprings. The following parameters can be chosen to control DE process: population size, perturbation rate to define mutation scheme, and method to generate individuals for the solution.

A more detailed and comprehensive explanation of the genetic algorithm and differential evolution can be found in the books written by Shukla et al. (2010) and Price et al. (2006).

Successful application of evolutionary algorithms to solve **real optimization non-convex problems**, as well as robustness of this approach, made the technique attractive to researches and the subject to numerous packages in Python and R: distributed evolutionary algorithms in python DEAP (Rainville et al. (2012)); mystic optimization algorithm framework (McKerns et al. (2009)); GA package (Scrucca et al. (2013)); RFreak (Nunkesser (2008)), etc. However, the main disadvantage of the method is long execution times compared to QP and LP.

3.4 Penalty Function

Imposing equality/inequality constraints that are not feasible will lead to failure in finding solutions in LP and QP problems. One possible approach is to adjust constraints, another is to establish a penalty function that 'penalizes' the objective functions 3.1.1 or 3.2.1.

In the case of LP in section 3.1 the objective function 3.1.1 is transformed to (Venkataraman (2002)):

$$\min_x \boldsymbol{p}^T \boldsymbol{x} + P(\boldsymbol{x}) \tag{3.4.1}$$

where $P(x)$ is penalty function and is defined as follows:

$$P(x) = r_e \cdot \sum_{j=1}^{J} (h_e^2(\boldsymbol{x}))_j + r_i \cdot \sum_{k=1}^{K} max(0, (g_i(\boldsymbol{x}))_k)^2 \tag{3.4.2}$$

where:
$j = 1, 2, ..., J$: number of equality constraints
$k = 1, 2, ..., K$: number of inequality constraints
h_e: function of equality constraint;
r_e: penalty multiplier of equality constraint;
g_i: function of inequality constraint;
r_i: penalty multiplier of inequality constraint.

Function h_e is a reformulated equality constraint (3.1.3). Every time when equality (3.1.3) will not be satisfied the equality term will be greater than 0 and will contribute to the penalty function (3.4.2).

$$h_e = \boldsymbol{Cx} - \boldsymbol{b} \tag{3.4.3}$$

Function g_i is a representation of inequality constraint (3.1.2). While inequality constraints are not met, the function takes positive values, so does the inequality term in penalty function (3.4.2).

$$g_i = \boldsymbol{l} - \boldsymbol{Ax} \tag{3.4.4}$$

Parameters of control in the penalty function are r_e and r_i. Possible adjustment of the penalty function by varying the penalty multipliers will be useful in the portfolio optimization.

4 Project Model

4.1 Class "Projects"

We developed a probabilistic model for each project where the relevant uncertainties for portfolio analysis were generated using Monte Carlo simulation with 5000 iterations (`n=5000`) over a 100 year time period (`period=100`).

The model is developed using Python class `Projects`, where individual projects are represented as an instance of the class. To create the instance the following arguments (see table 4.1.1) should be provided to the class (see listing 4.1.1).

Object Orienting Programming (OOP)

In Python, *Objects* can be thought of as a representation of real-world objects: houses, cars, laptops, etc. At the same time, say, houses have own parameters: area, number of floors, number of windows, etc. These parameters are named *Attributes*.

Python class provides a framework for the object and its attributes. Different houses can be set as class House, and the individual house is called *Instance*.

Finally, we may want to have a function of the class Houses, that calculates the number of windows per unit of area. This function is called *method*.

In the example below, class `House` was defined, it has attributes `area`, `number_of_floors`, `number_of_windows`, and the method `windows_per_area`

```python
class House(object):
  def __init__(self, area, number_of_floors, number_of_windows):
    self.area = area
    self.number_of_floors = number_of_floors
    self.number_of_windows = number_of_windows

  def windows_per_area(self):
    return self.number_of_windows / self.area
```

The OOP is beyond the scope of this work, a detailed explanation of OOP application in Python can be found in the text-book written by Lutz (2009)

The input argument Hydrocarbon Price (`hc_price`), which is a matrix of size `period` $\times$ `n` (number of simulated years $\times$ number of realizations), can be generated by using the function Oil and Gas Price (`price()`) described in section 4.2.

Listing 4.1.1: Class "Projects" Definition

```python
Projects(life_cycle, locat, m_res, poes_expl, poas_appr,
   m_av_max_well_rate, capex_scaler, opex_scaler, well_cap_pot,
   hc_price, y_to_start_project=0, hc='oil', period=100, n=5000,
   start_of_sim=2009, pipe=0)
```

Table 4.1.1: Input Parameters of Class "Projects"

Parameter	**Units**	**Python Argument**	**Variable Type**
Current Stage of Project's Life Cycle	*"expl", "appr", "dev"*	`life_cycle`	string
Location of the Project	*"onshore", "shelf", "deep"*	`locat`	string
Units of Reserves	*Mstb; Bscf*	`m_res`	float
Probability of Success of Exploration Stage	-	`poes_expl`	float
Probability of Success of Appraisal Stage	-	`poas_appr`	float
Units of Average Maximum Well Rate	*Kbpd; Mscfpd*	`m_av_max_well_rate`	float
CapEx Scaler	-	`capex_scaler`	float
OpEx Scaler	-	`opex_scaler`	float
Well Capacity Potential	-	`well_cap_pot`	float
Hydrocarbon Price	*USD/bbl; USD/Kscf*	`hc_price`	numpy array
Years to Start Project	*years*	`y_to_start_project`	float
Hydrocarbon	*"oil", "gas"*	`hc`	string
Simulation Period	*years*	`period`	float
Number of Realisations	-	`n`	float
Start of Simulation	*year #*	`start_of_sim`	float
Pipeline Cost	*USD (million)*	`pipe`	float

4.2 Oil and Gas Price

Oil and gas prices were modeled as mean reverting (MR) processes. The MR model is a stochastic process, where prices in each time period follow a log-normal distribution, but the logarithmic price changes are related to each other and have constant long term equilibrium price and mean reversion rate (Begg et al. (2007)):

$$\frac{dP}{P} = \eta(P - P^*)dt + \sigma\epsilon\sqrt{dt} \tag{4.2.1}$$

where:
P: price
P^*: long term equilibrium price
t: time period
η: mean reversion rate (speed at which the price tends to revert to the mean)
σ: price volatility
ϵ: random variable with standard normal distribution

We will not discuss a detailed description of the process, for a comprehensive review of the MR the reader should consult Begg et al. (2007).

The MR price model was presented in the form of function `price` with default arguments `period=100`, `n=5000`: simulation periods in years and number of realizations respectively (see listing 4.2.1). Parameters used for price calculations are derived from historical prices. These parameters are shown in table 4.2.1, and the values are defined in the function's body. Prefix `o_` and `g_` at the beginning of parameters refers to oil and gas, respectively.

Listing 4.2.1: Oil and Gas Price Function

```
price(period=100, n=5000)
```

Table 4.2.1: Parameters for Oil and Gas Price Calculation

Parameter	**Unit**	**Python Variable**	**Oil**	**Gas**
Time Period	years	`period`	100	100
Realizations		`n`	5000	5000
Price Floor	USD/bbl; USD/Mscf	`o_price_floor`; `g_price_floor`	8	0.8
Simulation Time Step	years	`o_dt`; `g_dt`	1	1
Standard Deviation of Annual Increments	USD/bbl; USD/Mscf	`o_sd`; `g_sd`	5	0.7
Half Life	years	`o_half_life`; `g_half_life`	4	8
Initial Price	USD/bbl; USD/Mscf	`o_initial`; `g_initial`	70	5
Long Term Mean Price	USD/bbl; USD/Mscf	`o_l_term`; `g_l_term`	48.3333	5.8333

The function returns two arrays `oil_price` and `gas_price`, where each row represents the time period and each column is individual realization. Figure 4.2.1 shows the price of oil across the 100-year period, while figure 4.2.2 illustrates oil price distribution for year 8.

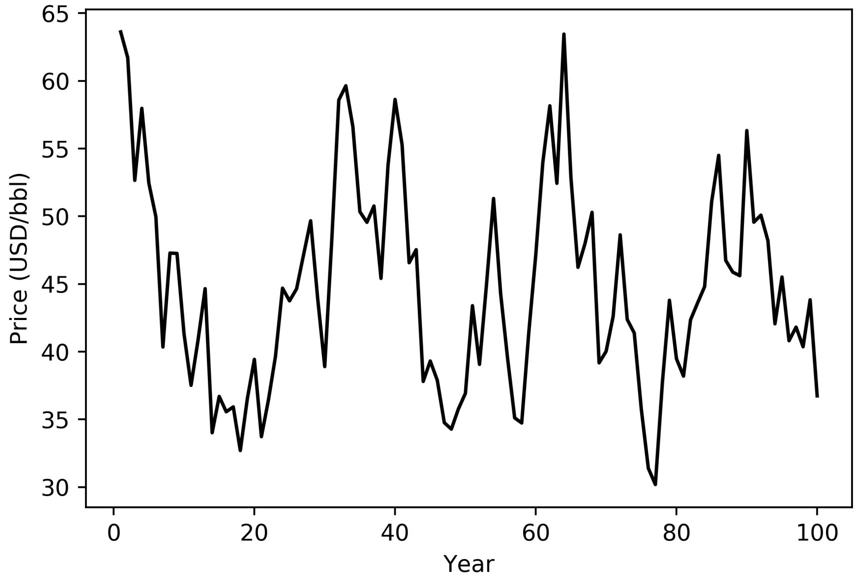

Figure 4.2.1 : Oil Price

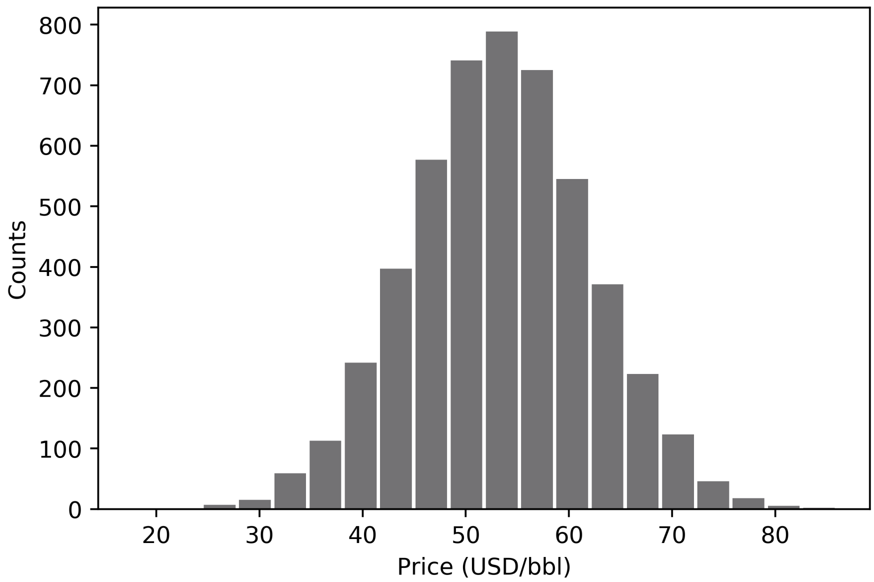

Figure 4.2.2 : Oil Price Year 8

4.3 Project Simulation

Project Simulation is developed as the method `proj_sim()` of class `Projects` (see listing 4.3.1). No arguments are required, as `proj_sim()` makes use of class attributes defined in section 4.1.

Calculated variables that are constant along the simulation timeline are summarized in table 4.3.1.

Listing 4.3.1: Project Simulation Method

```
Projects.proj_sim()
```

Table 4.3.1: Parameters Constant Along the Timeline

Parameter	**Unit**	**Python Variable**	**Distribution**
Reserves	Mstb; Bscf	`res`	PERT
Equivalent Reserves	Mboe	`res_for_case`	-
Reservoir Case	-	`case`	-
Average Maximum Well Rate	Kbpd; Mscfpd	`av_max_well_rate`	PERT
Average Annual Maximum Well Rate	Mbpa; Bscfpa	`well_max_prod`	-
Exploration Indicator	1 or 0	`do_expl_i`	-
Exploration Success	1 or 0	`expl_success`	Bernoulli
Appraisal Indicator	1 or 0	`do_appr_i`	-
Appraisal Success	1 or 0	`appr_success`	Bernoulli
Development Indicator	1 or 0	`do_dev_i`	-
Length of Exploration Program	1, 2 or 3	`p_l_expl`	Multinomial
Length of Appraisal Program	1, 2 or 3	`p_l_appr`	Multinomial
Exploration Program Cost	USD (million)	`expl_cost`	PERT
Appraisal Program Cost	USD (million)	`appr_cost`	PERT
Average Development Well Cost	USD (million)	`av_dev_well_cost`	PERT

Parameter	**Unit**	**Python Variable**	**Distribution**
Facility Cost	USD (million)	`facil_cost`	PERT
Fixed OpEx	USD (million)	`fixed_opex`	PERT
Variable OpEx	USD/bbl; USD/Kscf	`var_opex`	PERT
Abandonment Cost	USD (million)	`aband_cost`	-
Number of Wells	-	`n_of_wells`	-
Wells Ready for the 1st Production Year	-	`wells_ready_1st`	-
Years to Start Development	years	`y_to_start_dev`	-
Years of Development Phase	years	`y_of_dev`	-
Number of Wells Drilled per Year	wells/year	`wells_per_year`	-
Well Effective Production Rate	Mbblpa; Bscfpa	`well_eff_prod`	-
Processing Capacity	Mbblpa; Bscfpa	`proces_cap_year`	-
Years to First Production	years	`y_to_firs_prod`	-
Start of Exploration Program	year #	`expl_start`	-
End of Exploration Program	year #	`expl_end`	-
Start of Appraisal Program	year #	`appr_start`	-
End of Appraisal Program	year #	`appr_end`	-
Start of Development Program	year #	`dev_start`	-
End of Development Program	year #	`dev_end`	-

Eleven parameters (Reserves, Exploration Success, etc.) are simulated as random variables that follow PERT, Bernoulli and multinomial distributions. Figures 4.3.1 , 4.3.2 , 4.3.3 illustrate three parameters that follow each of the distributions.

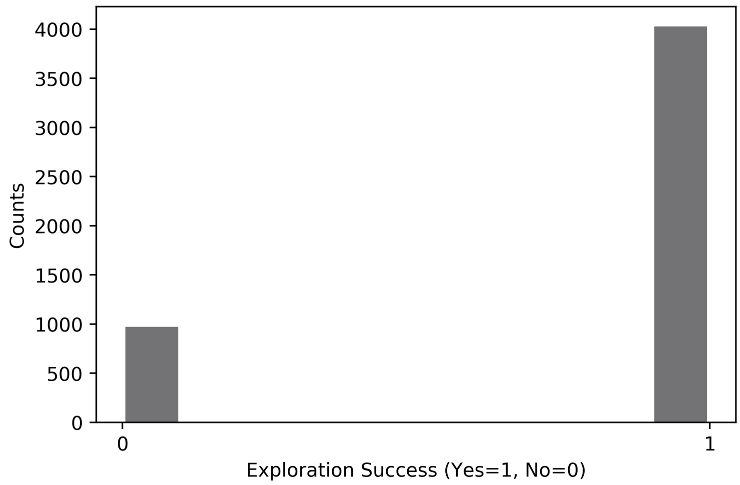

Figure 4.3.1 : Exploration Success [Bernoulli]

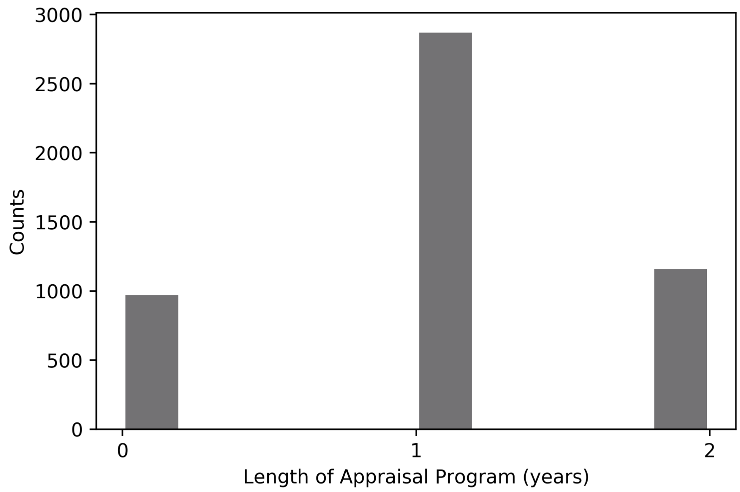

Figure 4.3.2 : Length of Appraisal Program [Multinomial]

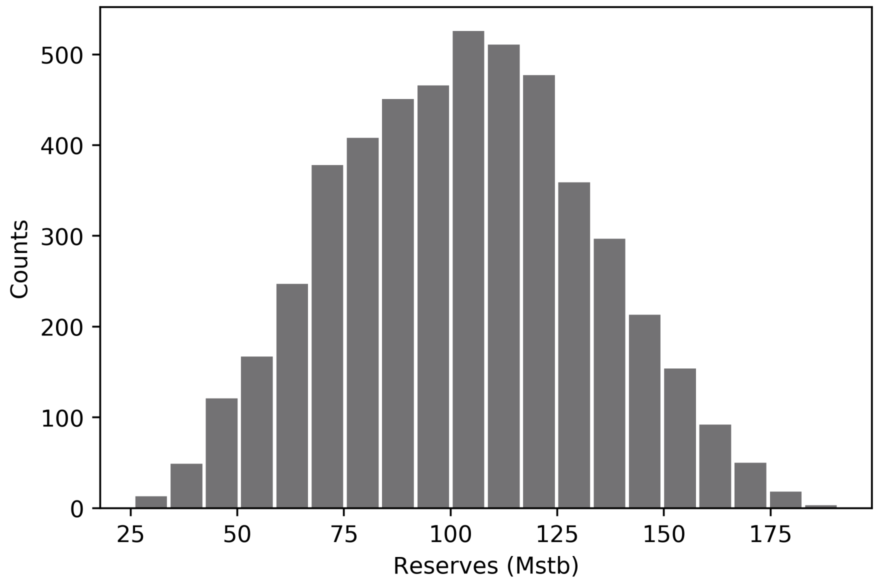

Figure 4.3.3 : Reserves [PERT]

Seven uncertain parameters (Reserves, Exploration Program Cost, Appraisal Program Cost, Average Development Well Cost, Facility Cost, Fixed Operating Expenditures (OpEx) and Variable OpEx) are calculated as functions of the project phase, project location, and fluid type. Table 4.3.2 and table 4.3.3 provide information needed for the calculations.

Average Maximum Well Rate and Abandonment Cost are considered as functions of the project phase and project location, respectively.

Years to Start Development, Wells Ready for the 1st Production Year, Start/End of Exploration Program, Start/End of Appraisal Program, and CapEx are defined as functions of gas/oil reserves. The calculations are based on the information from table 4.3.4.

Table 4.3.2: Uncertainty Factors as Function of Project Phase

Parameter	**Unit**	**Exploration**		**Appraisal**		**Development**	
		Min	Max	Min	Max	Min	Max
Reserves	-	0.20	2.00	0.50	1.50	0.75	1.50
Exploration Program Cost	*USD (million)*	5.00	30.00	-	-	-	-
Appraisal Program Cost	*USD (million)*	5.00	20.00	5.00	15.00	-	-
Average Development Well Cost	*USD (million)*	5.00	20.00	5.00	15.00	8.00	15.00
Average Maximum Initial Rate	-	0.25	1.75	0.50	1.50	0.75	1.25
Facilities Cost	*USD (million)*	5.00	20.00	5.00	20.00	8.00	20.00
Variable OpEx	*USD/bbl; USD/Kscf*	5.00	15.00	7.00	13.00	7.50	12.50
Fixed OpEx	*USD (million)/year*	5.00	15.00	7.00	13.00	9.00	11.00

Table 4.3.3: Most Likely Values as Function of Project Location

Parameter	**Unit**	**Oil**			**Gas**		
		Onshore	Shelf	Deep Water	Onshore	Shelf	Deep Water
Reserves per Well	*Mstb; Mboe*	1	10	20	1	10	20
Exploration Program Cost	*USD (million)*	100	200	500	100	200	500
Appraisal Program Cost	*USD (million)*	200	400	1000	200	400	1000
Average Development Well Cost	*USD (million)*	10	30	150	10	30	150
Abandonment as % of CapEx	*%*	5%	15%	20%	5%	15%	20%
Facilities Cost Multiplier	-	1	1.5	2	1.5	2.25	3
Variable OpEx	*USD/bbl; USD/Kscf*	10	15	20	1.645	2.468	3.290
Fixed OpEx	*%*	0.5%	1.0%	1.5%	0.5%	1.0%	1.5%

Table 4.3.4: Values as Function of Project Reserves

Gas Reserves	**Oil Reserves**	**Case**	**Years to Develop**	**Wells Ready for the 1st Year**	**Development Facilities CapEx Split**						**Probability of Appraisal Length**			**Probability of Exploration Length**		
Bscf	*Mstb*		*Years*	*%*	*Y1*	*Y2*	*Y3*	*Y4*	*Y5*	*Y6*	*1 year*	*2 years*	*3 years*	*1 year*	*2 years*	*3 years*
9.1	1	1	1	100%	1						1			1		
18.2	2	2	1	100%	1						1			1		
45.6	5	3	1	100%	1						1			1		
91.2	10	4	1	100%	1						1			1		
182.4	20	5	1	100%	1						1			1		
455.9	50	6	1	100%	1						0.7	0.3		0.8	0.2	
911.8	100	7	2	80%	0.4	0.6					0.7	0.3		0.8	0.2	
1823.5	200	8	2	70%	0.4	0.6					0.7	0.3		0.8	0.2	
4558.8	500	9	3	60%	0.2	0.3	0.5				0.5	0.5		0.6	0.4	
9117.6	1000	10	3	50%	0.2	0.3	0.3	0.2			0.5	0.5		0.6	0.4	
18235.3	2000	11	4	40%	0.2	0.3	0.3	0.2			0.5	0.5		0.6	0.4	
45588.2	5000	12	4	40%	0.1	0.2	0.2	0.2	0.3		0.2	0.5	0.3	0.5	0.5	
91176.5	10000	13	5	25%	0.1	0.1	0.2	0.2	0.2	0.2	0.2	0.5	0.3	0.5	0.5	
182352.9	20000	14	6	25%	0.05	0.1	0.2	0.2	0.25	0.2	0.2	0.5	0.3	0.5	0.5	

Once values from the table 4.3.1 are known, time-series parameters presented in table 4.3.5 are calculated. NPV, Development Wells, and Annual Production are shown on figures 4.3.4 , 4.3.5 , 4.3.6 , 4.3.7 , 4.3.8 as examples of projected parameters across the selected period of 100 years, and distributions of these parameters for a selected year.

Table 4.3.5: Time-Series Parameters

Parameter	**Unit**	**Python Variable**
Development Wells	# of wells	`eff_dev_well`
Annual Production	Mstb; Bscf	`ann_prod`
Cumulative Production	Mstb; Bscf	`cum_prod`
CapEx	USD (million)	`capex`
NCF	USD (million)	`bt_ncf`
NPV	USD (million)	`npv_year`

Time-series and distribution of three parameters are displayed on figures 4.3.4 , 4.3.6 , 4.3.8 . The peaks of distributions concentrated around zero are the results of project failures, which are driven by Probability of Success of Exploration Stage and Probability of Success of Appraisal Stage (table 4.1.1).

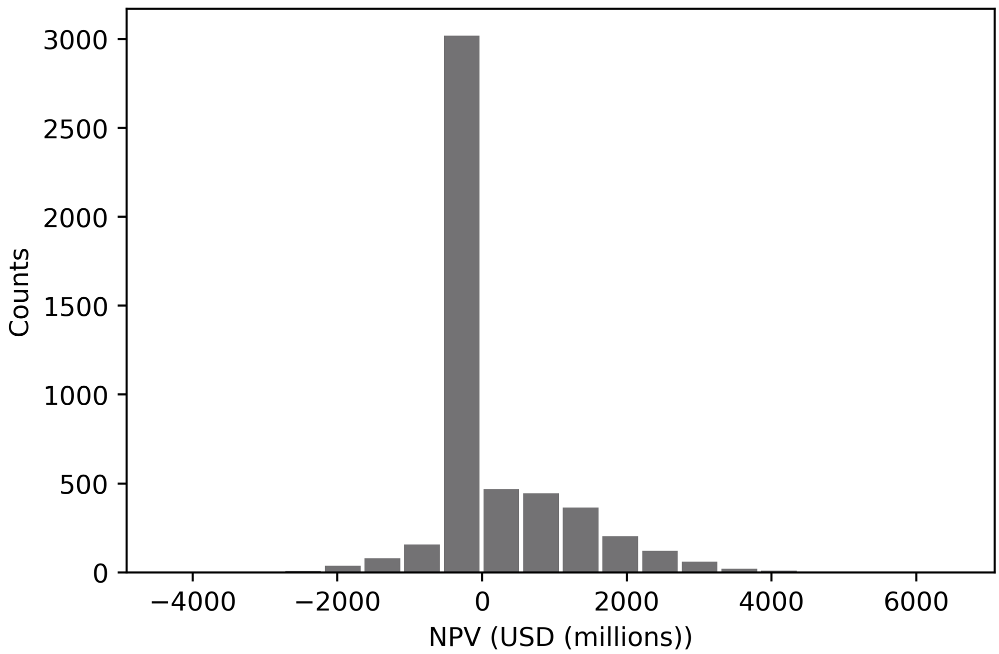

Figure 4.3.4 : NPV Distribution

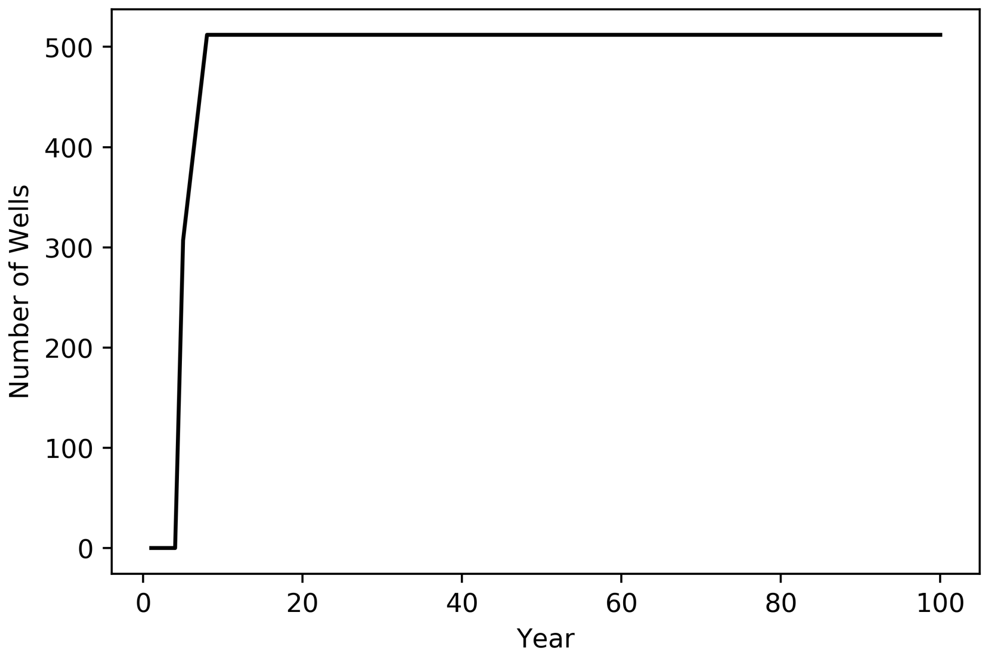

Figure 4.3.5 : Development Well Forecast

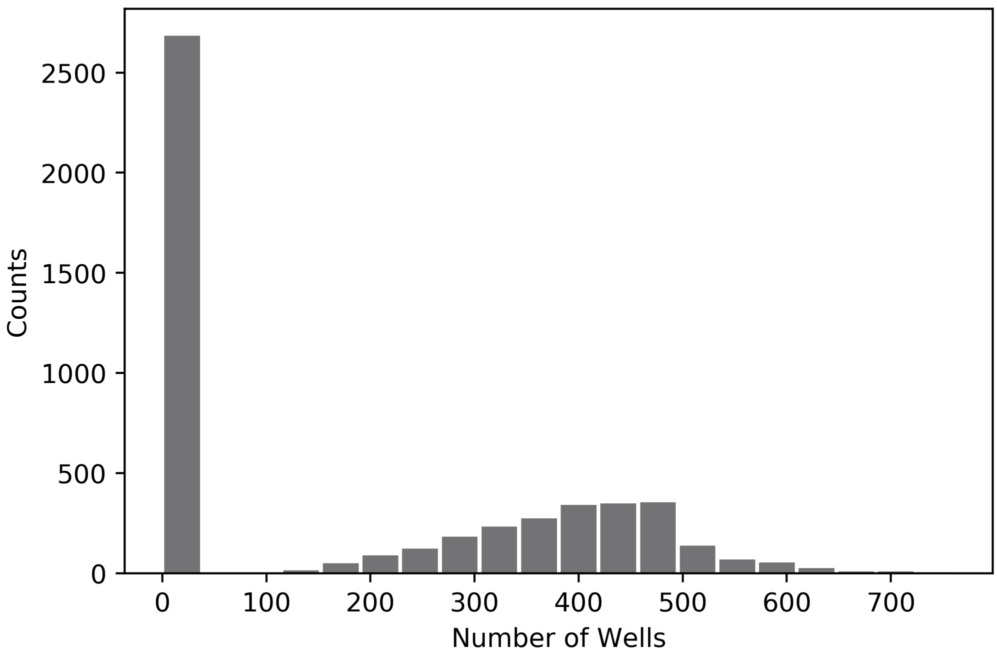

Figure 4.3.6 : Development Well Distribution Year 8

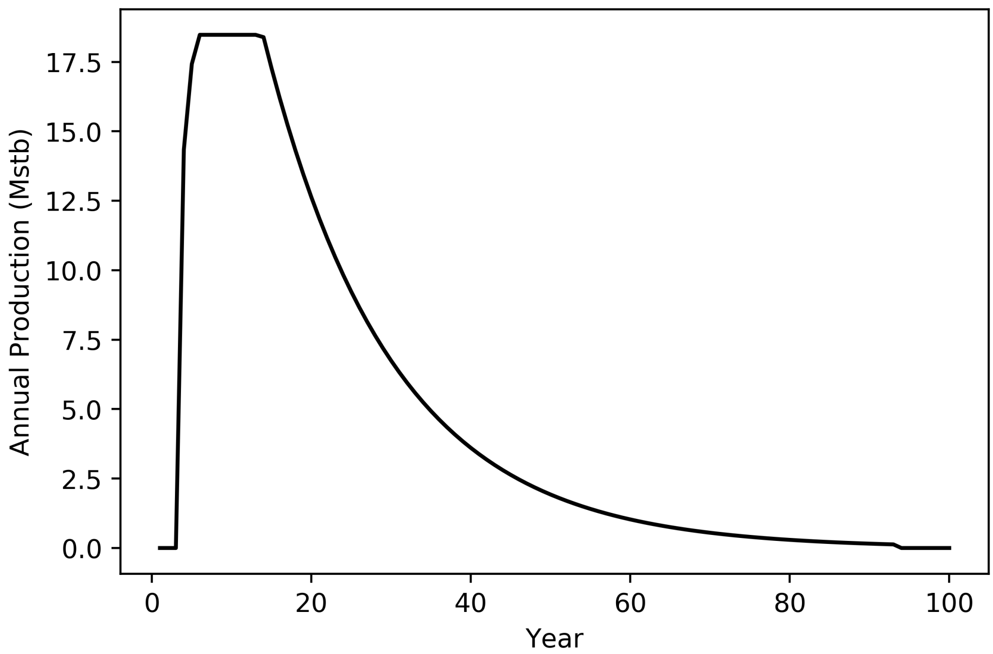

Figure 4.3.7 : Mean Annual Production Forecast

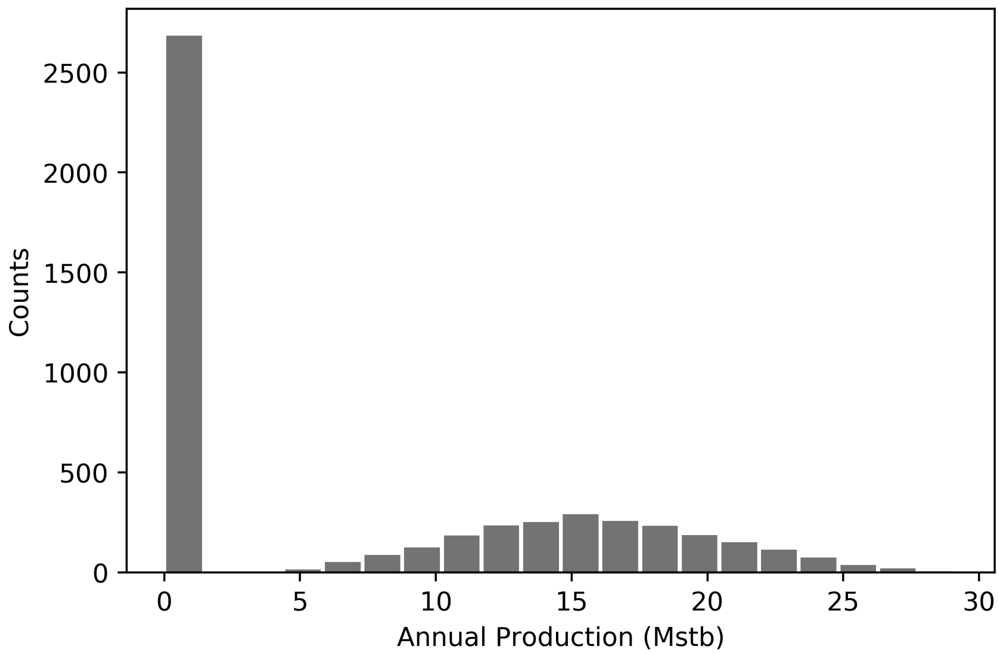

Figure 4.3.8 : Annual Production Distribution Year 8

The data, generated by project simulation is accessed through method `get_data()`. After calling the method the user will be asked to select the parameter that will be returned in the form of a dataframe table.

5 Portfolio Optimization Model

To perform portfolio optimization a pool of projects has to be available. In the model used here, this is done by establishing a `Portfolio_Pool` class, where a vector with the project's data is supplied as an argument (listing 5.0.1). An optimized portfolio is constructed by project selection from this portfolio pool.

Listing 5.0.1: Class "Portfolio Pool" Initiation

```
Portfolio_Pool(pool_data)
```

The portfolio optimization model is defined as the MTPO problem (section 2.4). Prior to optimization the objective function and constraints have to be specified. It's done by calling `obj_func()` and `constraint()` methods respectively (listings 5.0.2 and 5.0.3).

Listing 5.0.2: Objective Function Method

```
Portfolio_Pool.obj_func()
```

Listing 5.0.3: Constraint Method

```
Portfolio_Pool.constraint()
```

First, `obj_func()` method should be called. Once this is done the user will be asked to choose a representation of the main objective (NPV):

- P90
- P50
- P10
- Expected value

Although these parameter representations are not a consistent implementation of risk attitude, they help to provide quick information of optimized portfolios with respect to predefined cases (P90, P50, P10, expected value).

In the next step, the constraints have to be selected. The `constraint()` method allows us to choose any combination of the following constraint parameters:

- Oil Production
- Gas Production
- Oil Reserves
- Gas Reserves

- CapEx
- Net Cash Flow (NCF)

As in the case of `obj_func()` method, the user will be prompted to select between the following representations of constraints: P90, P50, P10, expected value.

Two methods of portfolio optimization were developed:

1. Main objective (NPV) optimization with corporate constraints
2. Maximization of the probability of exceeding the corporate goal constrained by NPV

The first problem is solved by the QP algorithm (section 3.2) and constraints are defined by the penalty function (section 3.4). To start optimization `qp()` method should be called (listing 5.0.4). The outcome of `qp()` method is a vector of portfolio weights.

Listing 5.0.4: Portfolio Optimization. QP Method

```
Portfolio_Pool.qp()
```

The second problem is solved by the differential evolution algorithm (section 3.3) where NPV constraint is defined by penalty function. To solve this problem `de()` method was developed (listing 5.0.5). Following the call of `de()` method uncertainty parameter (probability of exceeding a corporate goal) has to be chosen, and NPV constraint should be defined. The outcome of `de()` method is a vector of portfolio weights.

Listing 5.0.5: Portfolio Optimization. DE Method

```
Portfolio_Pool.de()
```

Optimized portfolio performance can be visualized in the same manner, that was shown on figures 2.4.1 , 2.4.2 , 2.4.3 .

6 Case Study

In this chapter, a pool of E&P projects will be studied. The time-series multi-objective portfolio optimization is used to construct optimized portfolios. Two portfolio optimizations will be performed and compared: optimization with respect to expected values, optimization with respect to the P90. In the next step, a sensitivity analysis will be conducted. It will provide information on which corporate constraints have the largest impact on portfolio selection. Finally, we will illustrate and discuss probability optimization.

The project pool includes 30 E&P projects. The decision-makers have to choose projects so that the resulting portfolio meets corporate constraints set by management (table 6.0.1). NPV is chosen as the main objective, and probabilities of exceeding corporate constraints are defined as uncertainty measures.

Table 6.0.1: Corporate Constraints Along the Timeline

Parameter	**Unit**	**Year**	**Corporate Constraint**
Oil Production	*Mstb*	1	15
		2	17
		3	18
		4	20
		5	22
		6	24
		7	27
		8	29
		9	32
		10	35
Gas Production	Mscfpd	1...10	420
CapEx	USD (million)	1	-2500
		2	-3500
		3...10	-1500
NCF	USD (million)	1...10	100
Oil Reserves	Mstb	-	600
Gas Reserves	Bscf	-	3500

Project simulation data is summarized and presented in terms of expected values (tables 6.0.2, 6.0.3, 6.0.4). Projects #1 - #6 and #11 - #21 are oil projects, Expected Gas Reserves and Expected Gas Production values for these projects are replaced with dashes (tables 6.0.2, 6.0.3), and vice versa for the gas projects (#7 - #10, #22 - #30). From table 6.0.3 it's clearly seen that expected oil production for individual projects in the first year is 0 except for project #19 that yields 3.4 Mstb. In contrast to the oil projects, three gas projects deliver approximately 450 Bscf of aggregated expected production at year 1. However, although projects #9 and #10 have substantial gas production they have poor expected NPVs of - USD 18,356 million and - USD 10 million, respectively. Additionally, the large exploration, appraisal, and development costs were the reasons for negative expected NCFs in year 1 and 2 for almost all the projects.

Table 6.0.2: NPV, Oil and Gas Reserves

Project	**Expected Oil Reserves**	**Expected Gas Reserves**	**Expected NPV**	**Project**	**Expected Oil Reserves**	**Expected Gas Reserves**	**Expected NPV**
	Mstb	Bscf	USD (million)		Mstb	Bscf	USD (million)
1	414	-	239	16	229	-	185
2	104	-	-424	17	108	-	277
3	829	-	-7141	18	125	-	-422
4	156	-	-1016	19	96	-	294
5	939	-	-2837	20	145	-	-432
6	312	-	724	21	268	-	146
7	-	714	-81	22	-	1009	-918
8	-	2062	1032	23	-	2027	-265
9	-	4171	-18356	24	-	554	68
10	-	2082	-10	25	-	630	-772
11	105	-	820	26	-	261	123
12	450	-	-273	27	-	709	-1074
13	762	-	-314	28	-	3048	-8153
14	649	-	-1436	29	-	119	-16
15	996	-	-5727	30	-	390	209

Table 6.0.3: Oil and Gas Production

Project #	**Expected Oil Production, Mstb**										**Expected Gas Production, Bscf**									
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
1	0.0	0.0	0.0	2.6	5.1	6.7	7.3	7.3	7.3	7.3	-	-	-	-	-	-	-	-	-	-
2	0.0	0.0	0.4	1.3	1.8	1.9	1.9	1.9	1.9	1.9	-	-	-	-	-	-	-	-	-	-
3	0.0	0.0	0.0	0.8	12.1	29.9	37.6	37.9	30.4	14.2	-	-	-	-	-	-	-	-	-	-
4	0.0	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1	-	-	-	-	-	-	-	-	-	-
5	0.0	0.0	23.0	23.0	23.0	23.0	23.0	23.0	23.0	23.0	-	-	-	-	-	-	-	-	-	-
6	0.0	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.1	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	0.0	0.0	27.5	27.5	27.5	27.5	27.3	26.5	24.2	18.2
8	-	-	-	-	-	-	-	-	-	-	0.0	0.0	70.7	70.7	70.7	70.7	70.7	70.7	70.5	70.0
9	-	-	-	-	-	-	-	-	-	-	319.7	319.7	319.7	319.7	319.7	316.6	229.7	88.4	11.8	0.1
10	-	-	-	-	-	-	-	-	-	-	107.3	107.3	107.3	107.3	107.3	107.3	107.3	107.3	107.0	106.4
11	0.0	0.0	1.9	6.8	9.8	10.2	10.1	9.9	9.2	7.5	-	-	-	-	-	-	-	-	-	-
12	0.0	0.0	0.0	3.3	7.2	10.0	11.0	11.1	11.1	11.1	-	-	-	-	-	-	-	-	-	-
13	0.0	0.0	0.0	0.0	0.0	1.4	11.7	27.1	33.7	34.2	-	-	-	-	-	-	-	-	-	-
14	0.0	0.0	0.0	1.2	5.5	12.1	14.8	15.0	15.0	15.0	-	-	-	-	-	-	-	-	-	-
15	0.0	0.0	0.0	0.0	37.1	76.8	78.8	79.0	79.0	79.0	-	-	-	-	-	-	-	-	-	-
16	0.0	0.0	4.1	5.8	5.8	5.8	5.8	5.8	5.8	5.8	-	-	-	-	-	-	-	-	-	-
17	0.0	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	-	-	-	-	-	-	-	-	-	-
18	0.0	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	-	-	-	-	-	-	-	-	-	-
19	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	-	-	-	-	-	-	-	-	-	-
20	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-	-	-	-	-	-	-	-	-	-
21	0.0	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	2.9	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	0.0	0.0	13.8	13.8	13.8	13.8	13.8	13.8	13.8	13.8
23	-	-	-	-	-	-	-	-	-	-	0.0	0.0	0.0	0.0	63.5	63.5	63.5	63.5	63.5	63.5
24	-	-	-	-	-	-	-	-	-	-	0.0	0.0	0.0	29.3	29.3	29.3	29.3	29.3	29.1	28.5
25	-	-	-	-	-	-	-	-	-	-	0.0	0.0	7.2	7.2	7.2	7.2	7.2	7.2	7.2	7.2
26	-	-	-	-	-	-	-	-	-	-	0.0	24.3	24.3	24.3	24.3	24.3	24.2	23.6	20.5	14.7
27	-	-	-	-	-	-	-	-	-	-	0.0	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7	9.7
28	-	-	-	-	-	-	-	-	-	-	0.0	141.1	141.6	141.6	141.6	141.6	141.6	141.6	141.6	141.6
29	-	-	-	-	-	-	-	-	-	-	0.0	10.5	10.5	10.5	10.5	10.5	10.5	10.3	9.3	6.9
30	-	-	-	-	-	-	-	-	-	-	41.2	41.2	41.2	41.2	37.5	41.1	41.0	36.9	26.4	16.6

Table 6.0.4: CapEx and NCF

Project #	Expected CapEx, USD (million)										Expected NCF, USD (million)									
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10
1	-110	-85	-159	-819	-792	-602	-285	-110	-71	-21	-110	-85	-159	-694	-558	-302	33	203	236	281
2	-224	-260	-293	-301	-134	-28	-7	-1	0	0	-224	-260	-278	-248	-63	43	62	66	66	64
3	-509	-381	-389	-520	-1108	-1365	-777	-377	-321	-244	-509	-381	-389	-543	-1433	-2215	-1882	-1533	-1277	-689
4	-142	-845	-79	-79	0	0	0	0	0	0	-142	-809	-54	-62	9	3	-1	-5	-9	-12
5	-481	-721	-2053	-189	-189	-189	0	0	0	0	-481	-721	-1840	-25	-65	-99	64	41	21	5
6	-1019	-3105	-394	-394	0	0	0	0	0	0	-1019	-2316	345	303	664	636	614	595	579	565
7	-124	-110	-499	0	0	0	0	0	0	0	-124	-110	-401	99	101	101	101	100	92	69
8	-247	-234	-431	0	0	0	0	0	0	0	-247	-234	-191	245	249	254	254	256	256	258
9	-7217	0	0	0	0	-1	-36	-58	-31	-5	-9364	-2131	-2108	-2087	-2069	-2035	-1504	-621	-105	-6
10	-2867	0	0	0	0	0	0	0	-1	-2	-2608	264	272	279	285	290	292	297	301	303
11	-112	-133	-346	-530	-309	-93	-28	-4	0	0	-112	-133	-250	-200	145	366	416	422	386	308
12	-241	-550	-28	-28	0	0	0	0	0	0	-216	-180	-261	-484	-346	-109	102	188	195	208
13	0	0	-513	-457	-543	-880	-1768	-2133	-1149	-539	0	0	-513	-457	-543	-832	-1388	-1275	-115	487
14	-519	-422	-449	-748	-1253	-1466	-732	-323	-222	-62	-519	-422	-449	-705	-1067	-1072	-265	134	227	375
15	0	-749	-834	-1467	-3762	-3749	-1341	-1010	-509	0	0	-749	-834	-1467	-3232	-2763	-412	-158	273	728
16	-342	-395	-916	-380	-82	-24	0	0	0	0	-342	-395	-740	-141	146	196	214	208	203	199
17	-189	-1067	-99	-99	0	0	0	0	0	0	-189	-888	71	63	157	152	148	144	142	139
18	-356	-818	-47	-47	0	0	0	0	0	0	-356	-738	28	24	68	65	63	61	60	58
19	-1435	0	0	0	0	0	0	0	0	0	-1249	176	167	160	154	149	146	142	139	137
20	-241	-550	-28	-28	0	0	0	0	0	0	-241	-512	7	5	32	30	29	28	27	26
21	-334	-769	-63	-63	0	0	0	0	0	0	-334	-636	63	57	115	111	108	105	102	100
22	-249	-300	-941	0	0	0	0	0	0	0	-249	-300	-910	32	33	33	34	34	35	35
23	0	0	-251	-240	-1775	0	0	0	0	0	0	0	-251	-240	-1571	207	208	211	213	215
24	0	-125	-122	-612	0	0	0	0	0	0	0	-125	-122	-477	136	138	138	139	140	138
25	-250	-269	-576	0	0	0	0	0	0	0	-250	-269	-562	15	16	16	16	17	17	17
26	-200	-296	0	0	0	0	0	0	0	0	-200	-208	90	91	93	94	94	93	81	58
27	-401	-1049	0	0	0	0	0	0	0	0	-401	-1031	18	19	19	20	20	20	21	21
28	-1000	-5724	-18	0	0	0	0	0	0	0	-1000	-6034	-317	-290	-282	-276	-273	-267	-262	-256
29	-196	-118	-1	0	0	0	0	0	0	0	-196	-76	41	43	43	44	44	44	40	30
30	-795	0	0	0	-2	0	0	0	0	0	-655	142	145	148	136	152	152	139	100	64

6.1 Mean vs P10

Using the project pool presented in chapter 6, two portfolio are constructed; one with the main objective being the maximization of expected values and the other with the main objective being the maximization of the P10 of the NPV. The reason for doing that is to provide alternatives to the decison-makers with respect to their preferences.

The results of the Mean Optimization and the P10 Optimization are shown on figures 6.1.1 and 6.1.2 .

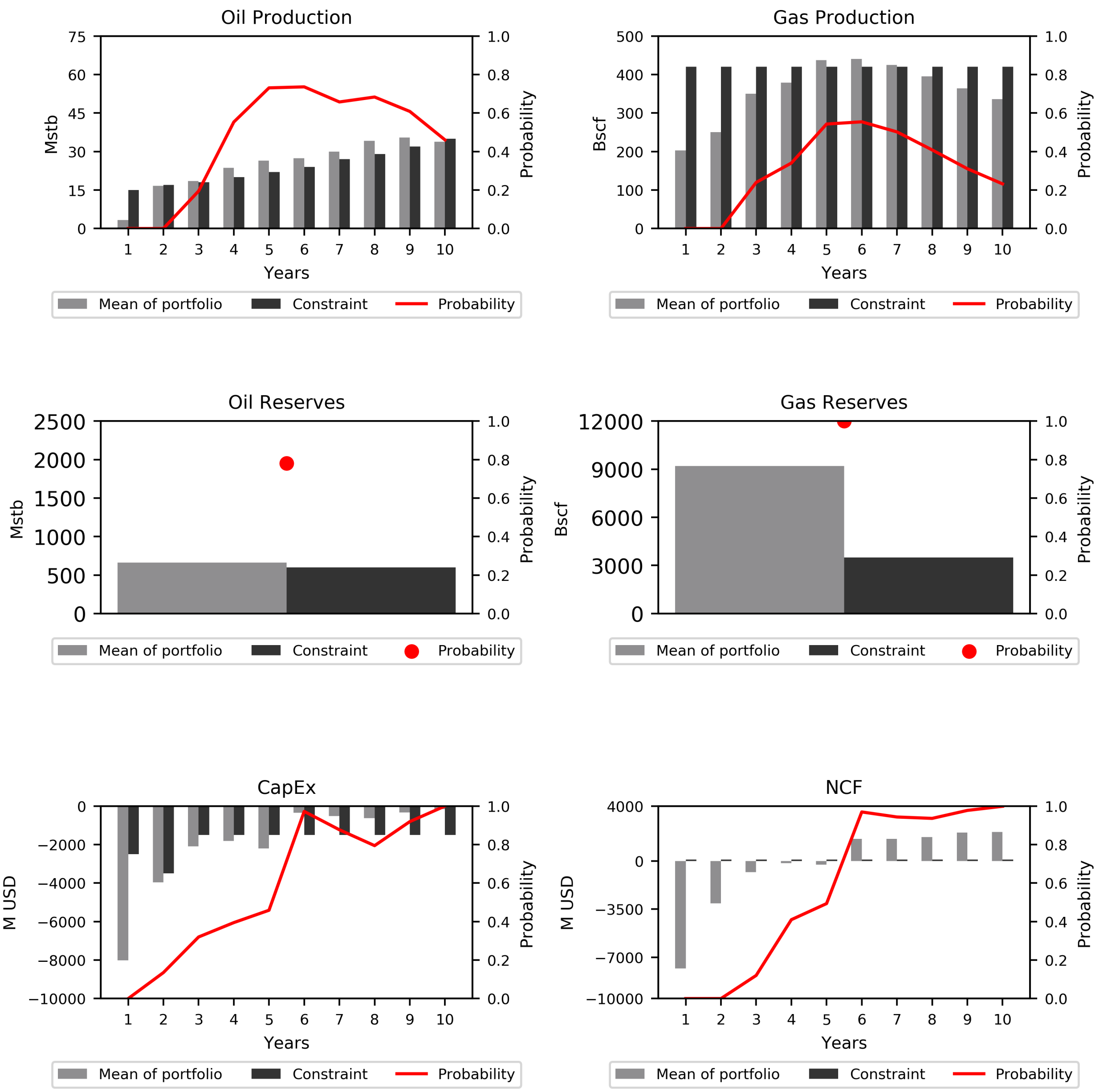

Figure 6.1.1 : Mean Optimization

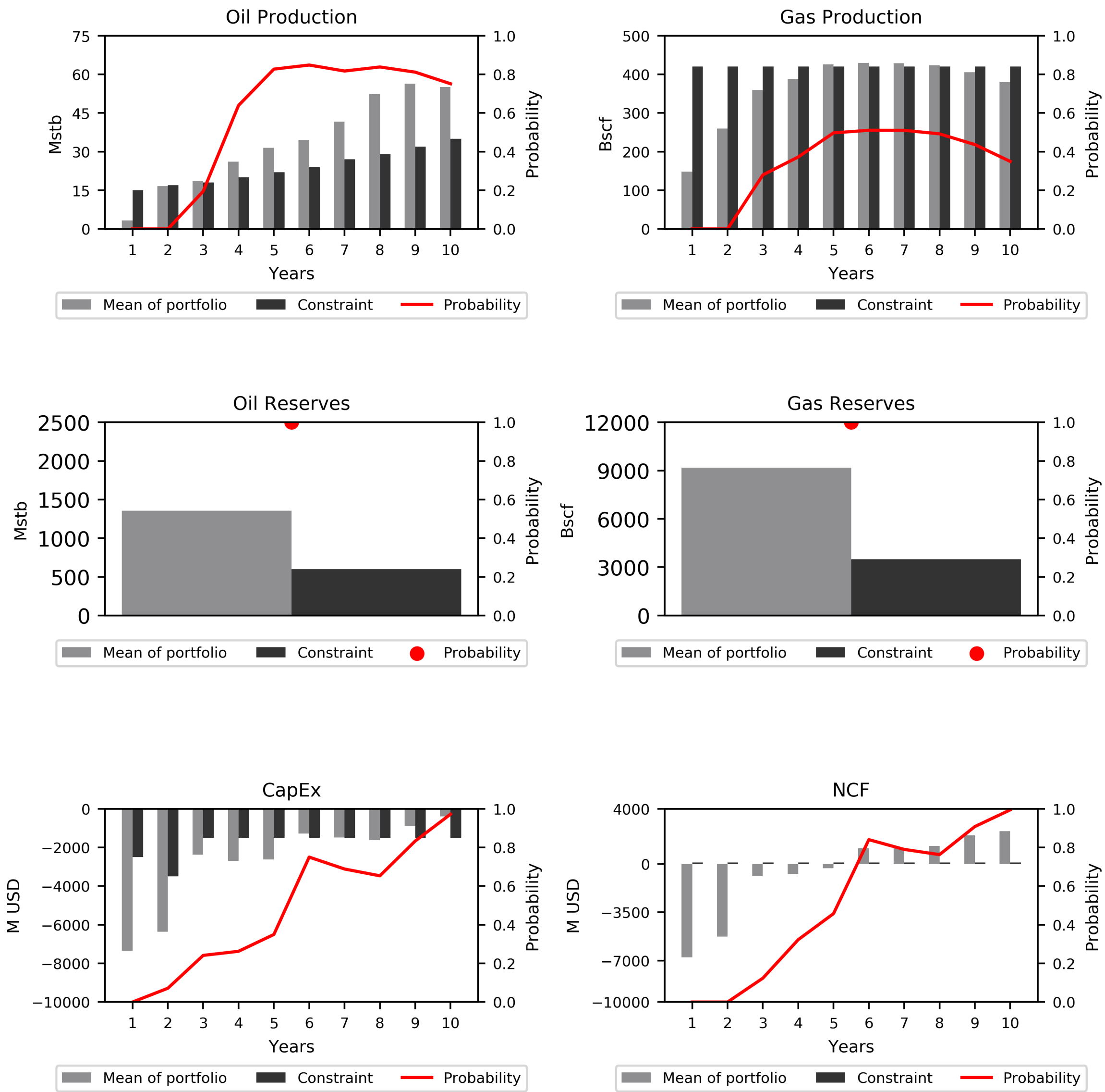

Figure 6.1.2 : P10 Optimization

P10-optimized portfolio outperforms Mean-optimized portfolio in terms of the probability of exceeding the corporate constraint. It's clearly seen that probabilities of exceeding Oil and Gas Production Constraints are noticeably greater for the period from year 6 to year 10. The same applies to oil reserves; however, probabilities of exceeding CapEx and NCF showed a slight decline. Expected portfolio NPV is - USD 1214 million and - USD 1581 million for Mean and P10 optimized portfolios, respectively. The negative outcomes can be explained by highly constrained optimization, especially considering constraints of Oil and Gas Production as well as CapEx and NCF for the period from year 1 to year 2, that have low probabilities of exceeding the corporate constraint. Nevertheless, the difference between mean NPV is substantial and it is the investor's task to determine the importance of meeting the corporate constraints with respect to the main objective (NPV).

6.2 Sensitivity Analysis

In section 6.1 it was shown that both optimized portfolios have negative mean NPVs. We also concluded that highly constrained optimization is the main factors that impact NPV.

In the next step to gain more information and support portfolio decision, a sensitivity analysis should be conducted. Penalty multipliers (see section 3.4) for all corporate constraints were set to be equal to 0.3. Then the penalty multiplier for each of the constraints was changed to 0.1 once at a time, while others were kept constant at 0.3. The procedure was then repeated with multipliers set to 0.5. The results were plotted as a tornado diagram (figure 6.2.1).

It's clearly seen from the plot that gas production corporate constraints for year 1 and 2 dominate other constraints. The second group of the most significant constraints in terms of their impact on expected NPV is the following: Gas Production Year 3, Gas Production Year 4, CapEx Year 1, NCF Year 1.

In the tables 6.2.1, 6.2.2, 6.2.3, 6.2.4, 6.2.5, 6.2.6 only projects with variations in portfolio weights with respect to the six constraints identified in the tornado graph are presented. The portfolio weight for project #9 is rising while constraints of gas production are improved by the increase of the penalty multiplier. This is happening because expected annual gas production of project #9 is the highest among all the projects in the pool. It also has the lowest expected NPV which prevents a 100% participation in this project.

Particular attention should be paid to the portfolio weight's difference of project #9 with respect to variations in penalty multipliers of Gas Production constraints from Year 1 to Year 4. The optimized portfolio with Gas Production constraint for Year 1 and its penalty multiplier of 0.1 does not include project #9, thus, portfolio weight is 0%. The optimized portfolio with the same constraint and its penalty multiplier of 0.5 has project #9 with portfolio weight of 28%. Therefore, the difference in portfolio weights of project #9 for two penalty multipliers of Gas Production constraint for Year 1 is 28%. Similarly, the weight's differences of project #9 for penalty multiplier variations of Gas Production constraints for Year 2, 3, 4 are 10%, 4%, and 3%, respectively. A probable explanation for this is a decreasing impact on the optimized portfolio of the Gas Production constraints from Year 1 to Year 4. However, the changes of portfolio weights for other projects (#6, #13, #17, #28) can not be easily explained, due to the combined effect of multiple constraints. To describe the interaction between the most impactful corporate constraints, a number of 3D graphs were plotted.

On the figure 6.2.2 a 3D graph of expected NPV with respect to changes in penalty multipliers of gas production constraints for year 1 and year 2 is presented. From the tornado chart, it is not clearly seen which corporate constraint has a larger impact on the main objective (NPV). Projections of the surface (figures 6.2.3 , 6.2.4) of figure 6.2.2 help to visualize the significance of one corporate constraint with respect to another. The isolines (curves with fixed penalty multiplier of the second corporate constraint) of the function ***Expected NPV(Penalty Multiplier of Gas Production for Year***

1) are steeper compared to the curves on figure 6.2.3 . Thus, the corporate constraint for Gas Production for Year 1 has a larger influence on the portfolio selection than that for Year 2.

The same analysis was done to distinguish the importance of Gas Production for Year 3 and Year 4. The projections (figures 6.2.6 , 6.2.7) showed the result that was expected after looking up at the tornado diagram (figure 6.2.1) - corporate constraint of Gas Production for Year 3 is indicated to have a larger effect on portfolio optimization.

Comparison of corporate constraints of Gas Production for Year 3 and CapEx for Year 1 is not that trivial as it is hard to decide solely on the basis of the tornado plot. However, figures 6.2.9 and 6.2.10 show us that the gas production constraint has a higher magnitude with respect to expected NPV. We arrived at the same conclusion, the relative importance of Gas Production constraint for Year 3 to NCF for Year 1, by inspecting the figures 6.2.12 and 6.2.13 .

Figures 6.2.15 , 6.2.16 and 6.2.18 , 6.2.19 showed the results such that Gas Production constraint for Year 4 is dominated by corporate constraints of CapEx Year 1 and NCF Year 1 respectively.

The similarities of the surfaces on the figures 6.2.8 and 6.2.11 as well as on the figures 6.2.14 and 6.2.17 come from the fact that NCF for Year 1 and CapEx for Year 1 are highly correlated (correlation coefficient = 0.9051). This is also clearly seen from the figures 6.2.20 , 6.2.21 , and 6.2.22 where NPV performance is the same considering variations of penalty multipliers of NCF Year 1 and CapEx Year 1.

In summary, the following was revealed in this section:

- The two most significant groups of corporate constraints regarding the main objective (NPV) were explored by plotting the tornado diagram (figure 6.2.1).
- Five projects that contribute to the changes in portfolio selection were displayed in tables from 6.2.1 to 6.2.6.
- The significance of two groups of corporate constraints has the following order (from largest to least impact): Gas Production Year 1 $>$ Gas Production Year 2 $>$ Gas Production Year 3 $>$ CapEx Year 1 / NCF Year 1 $>$ Gas Production Year 4.
- The surface graphs showed that most of the optimized portfolios yielded negative expected NPVs. To achieve positive values it's necessary to reduce the penalty multipliers of the most significant corporate constraints.

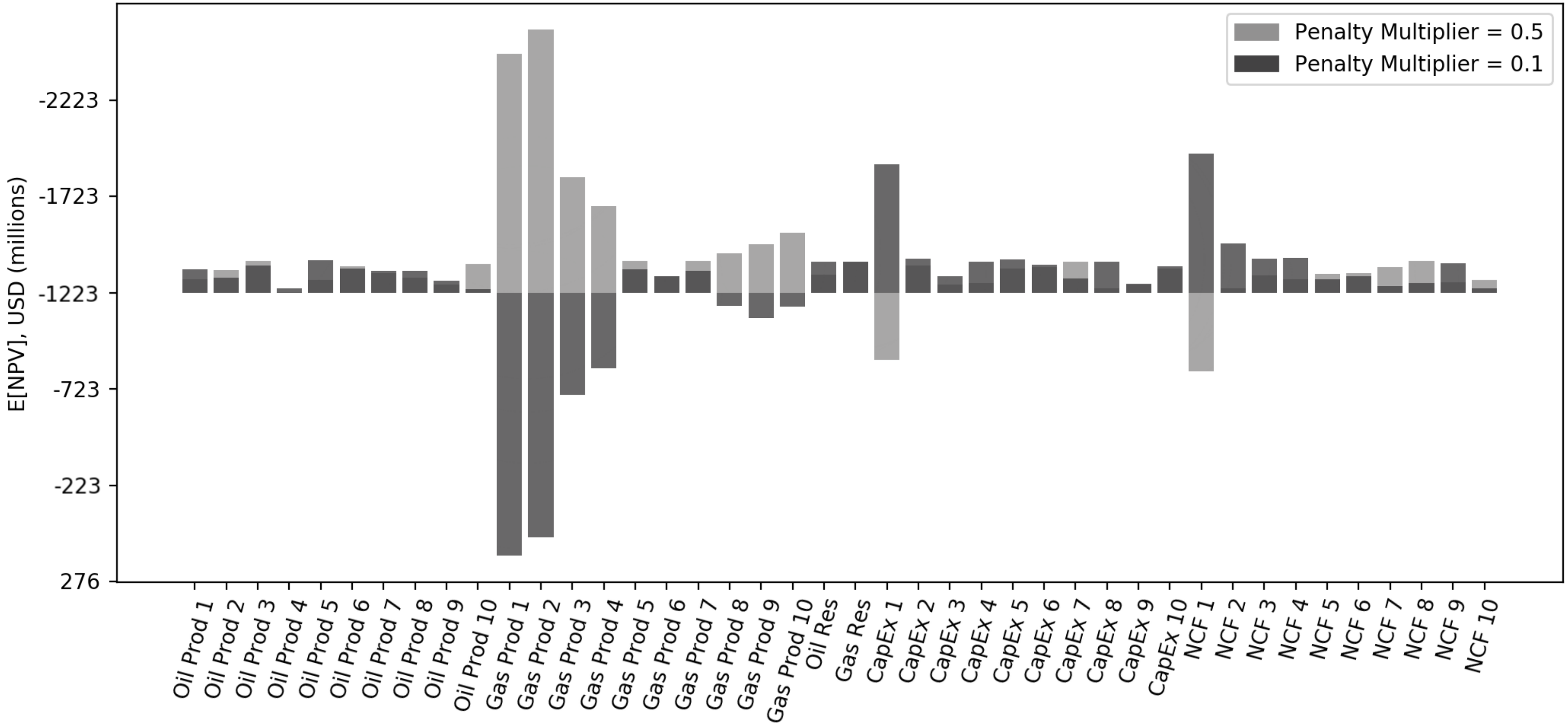

Figure 6.2.1 : Tornado Diagram

Table 6.2.1: Portfolio Weights. Gas Production Year 1

Project #	**E[Gas Prod] Y1**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-	724	68	68
9	319.7	-18356	0	28
13	-	-314	30	28
17	-	277	0	12
28	0	-8153	29	0

Table 6.2.2: Portfolio Weights. Gas Production Year 2

Project #	**E[Gas Prod] Y2**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-	724	57	69
9	319.7	-18356	12	22
13	-	-314	29	28
17	-	277	70	0
28	141.1	-8153	5	14

Table 6.2.3: Portfolio Weights. Gas Production Year 3

Project #	**E[Gas Prod] Y3**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-	724	61	65
9	319.7	-18356	15	19
13	-	-314	29	29
17	-	277	47	26
28	141.6	-8153	7	11

Table 6.2.4: Portfolio Weights. Gas Production Year 4

Project #	**E[Gas Prod] Y4**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-	724	58	60
9	319.7	-18356	16	19
13	-	-314	29	29
17	-	277	62	50
28	141.6	-8153	8	10

Table 6.2.5: Portfolio Weights. CapEx Year 1

Project #	**E[CapEx] Y1**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-1019	724	62	67
9	-7217	-18356	22	13
13	0	-314	28	29
17	-189	277	48	7
28	-1000	-8153	6	13

Table 6.2.6: Portfolio Weights. NCF Year 1

Project #	**E[NCF] Y1**	**E[NPV]**	**Portfolio Weights** (%)	
	Bscf	*USD (million)*	*Pen Mult = 0.1*	*Pen Mult = 0.5*
6	-1019	724	60	69
9	-9364	-18356	23	12
13	0	-314	28	29
17	-189	277	61	0
28	-1000	-8153	4	14

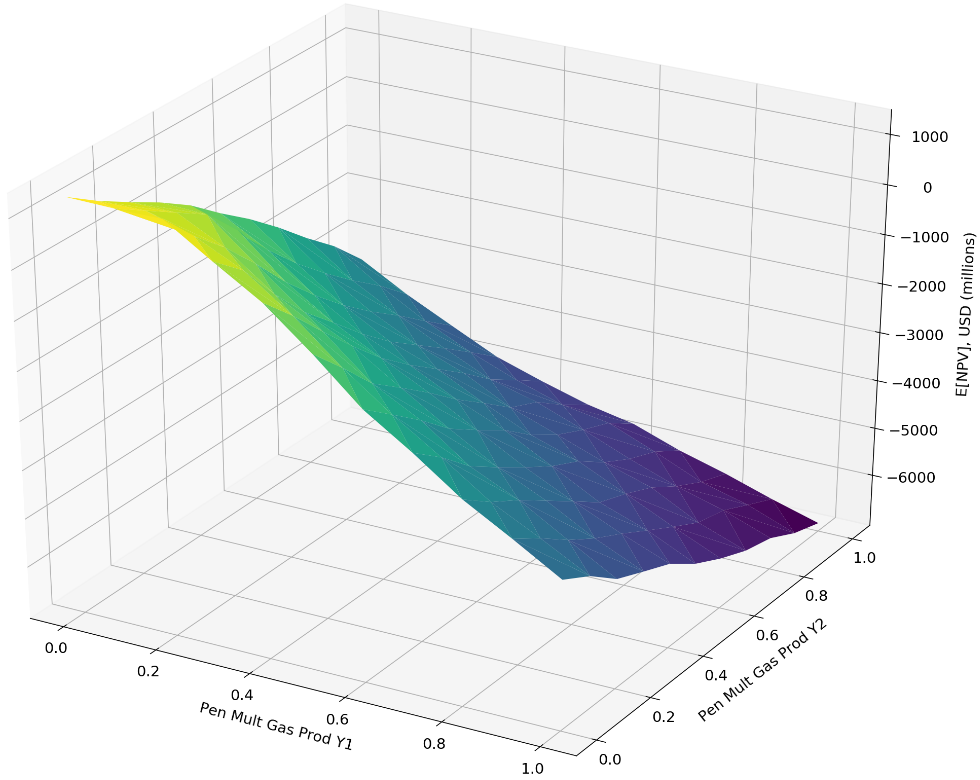

Figure 6.2.2 : Gas Production Year 1 - Gas Production Year 2

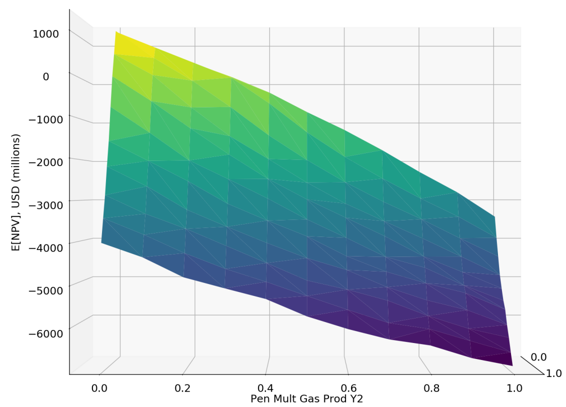

Figure 6.2.3 : G Prod Y2 Projection

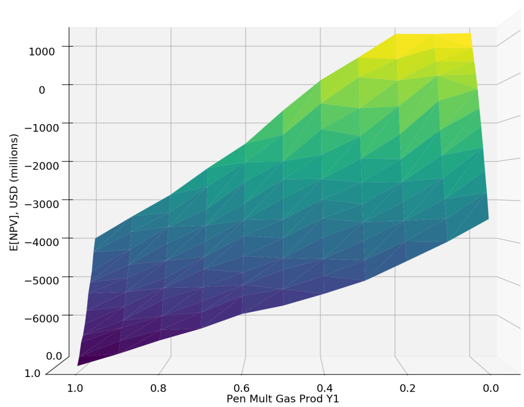

Figure 6.2.4 : G Prod Y1 Projection

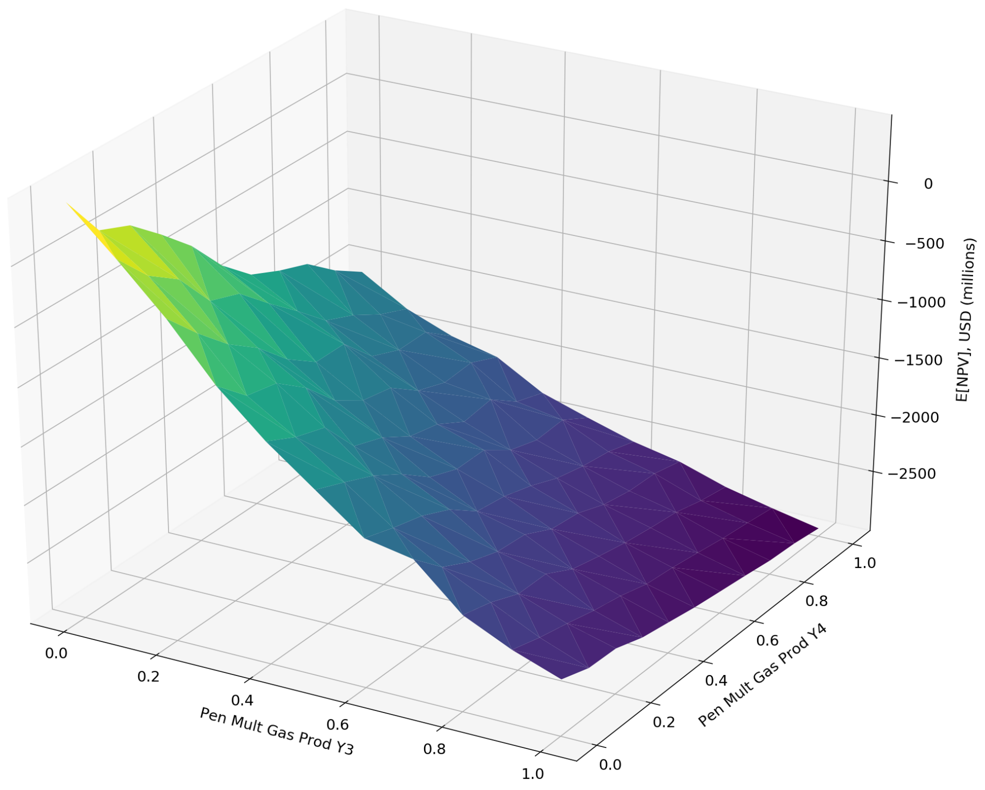

Figure 6.2.5 : Gas Production Year 3 - Gas Production Year 4

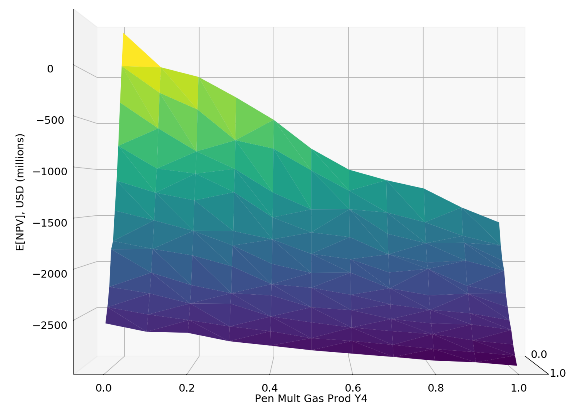

Figure 6.2.6 : G Prod Y4 Projection

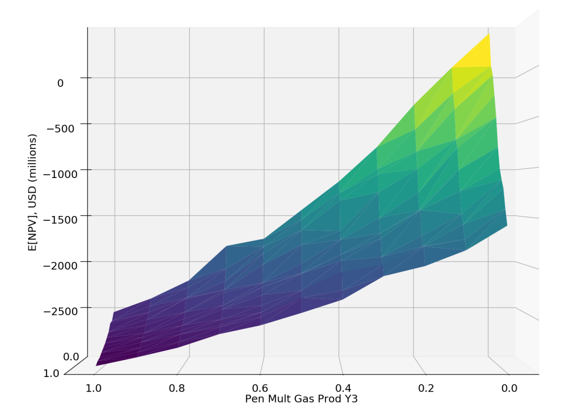

Figure 6.2.7 : G Prod Y3 Projection

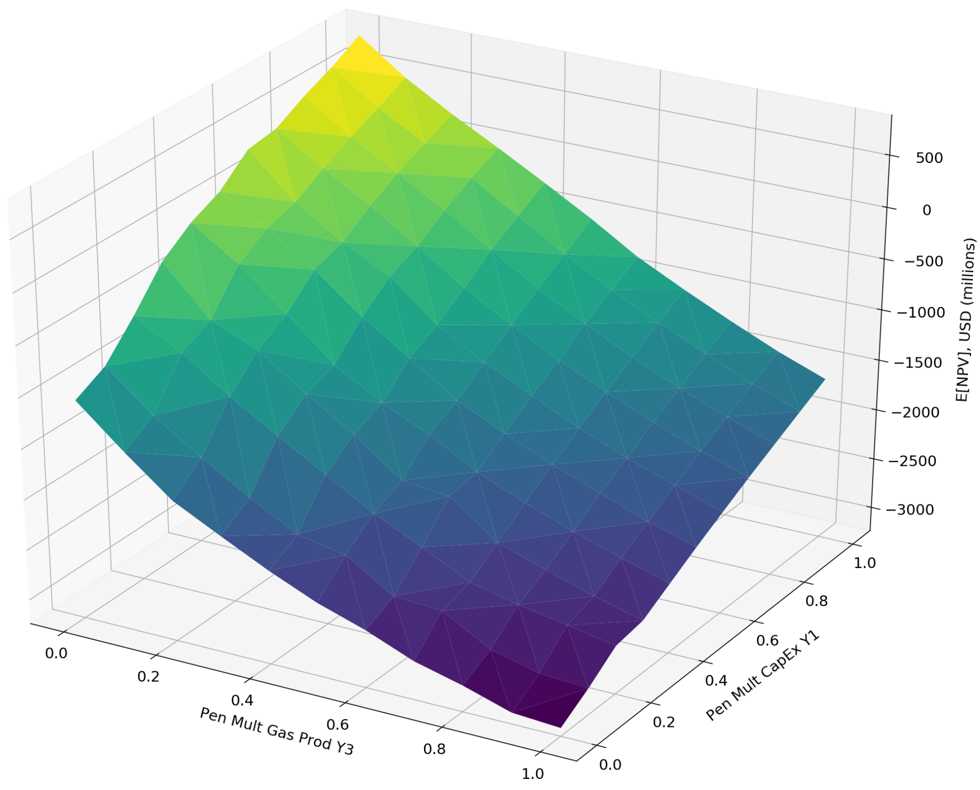

Figure 6.2.8 : Gas Production Year 3 - CapEx Year 1

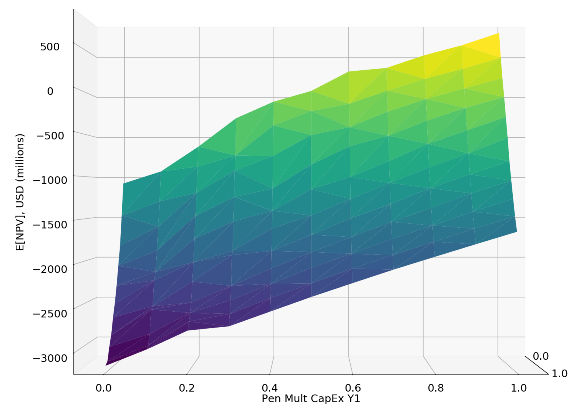

Figure 6.2.9 : CapEx Y1 Projection

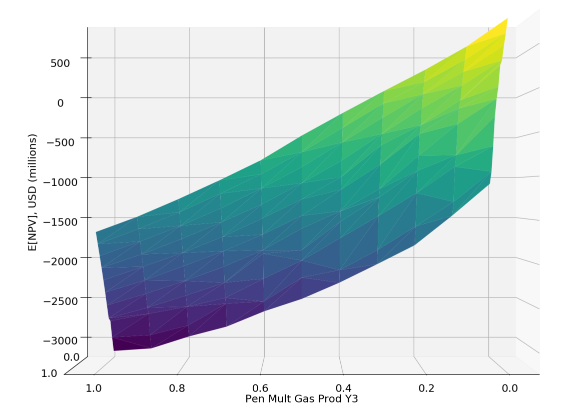

Figure 6.2.10 : G Prod Y3 Projection

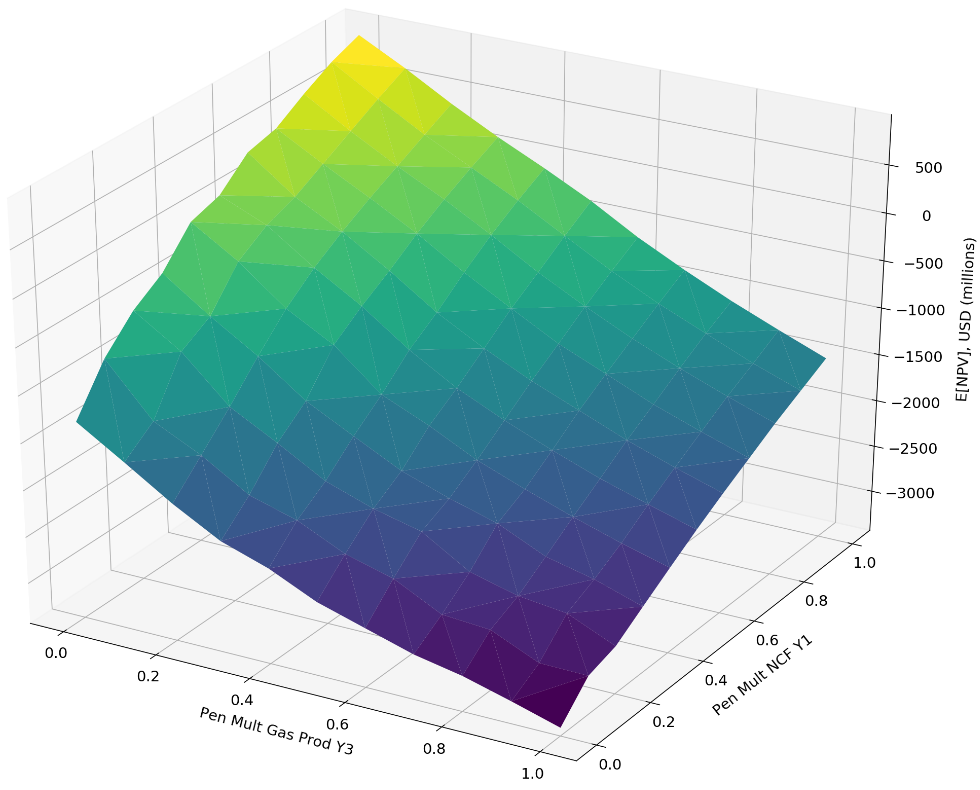

Figure 6.2.11 : Gas Production Year 3 - NCF Year 1

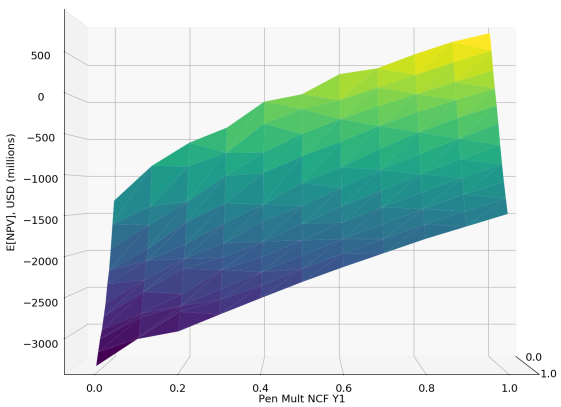

Figure 6.2.12 : NCF Y1 Projection

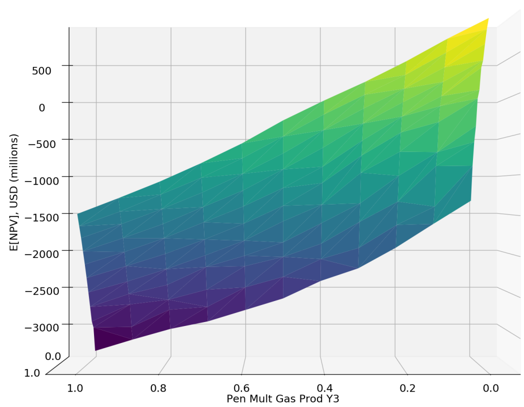

Figure 6.2.13 : G Prod Y3 Projection

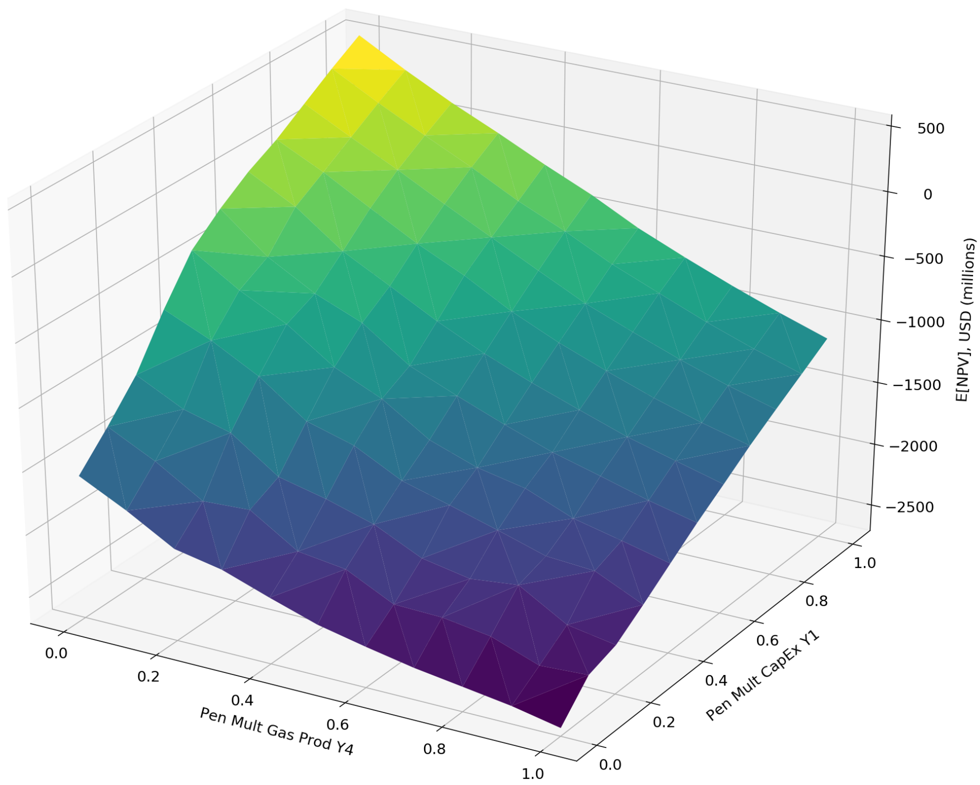

Figure 6.2.14 : Gas Production Year 4 - CapEx Year 1

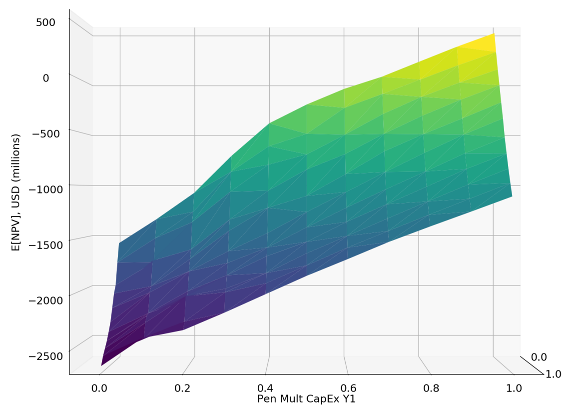

Figure 6.2.15 : CapEx Y1 Projection

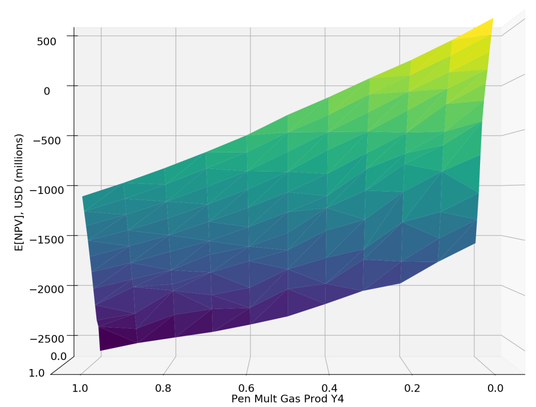

Figure 6.2.16 : G Prod Y4 Projection

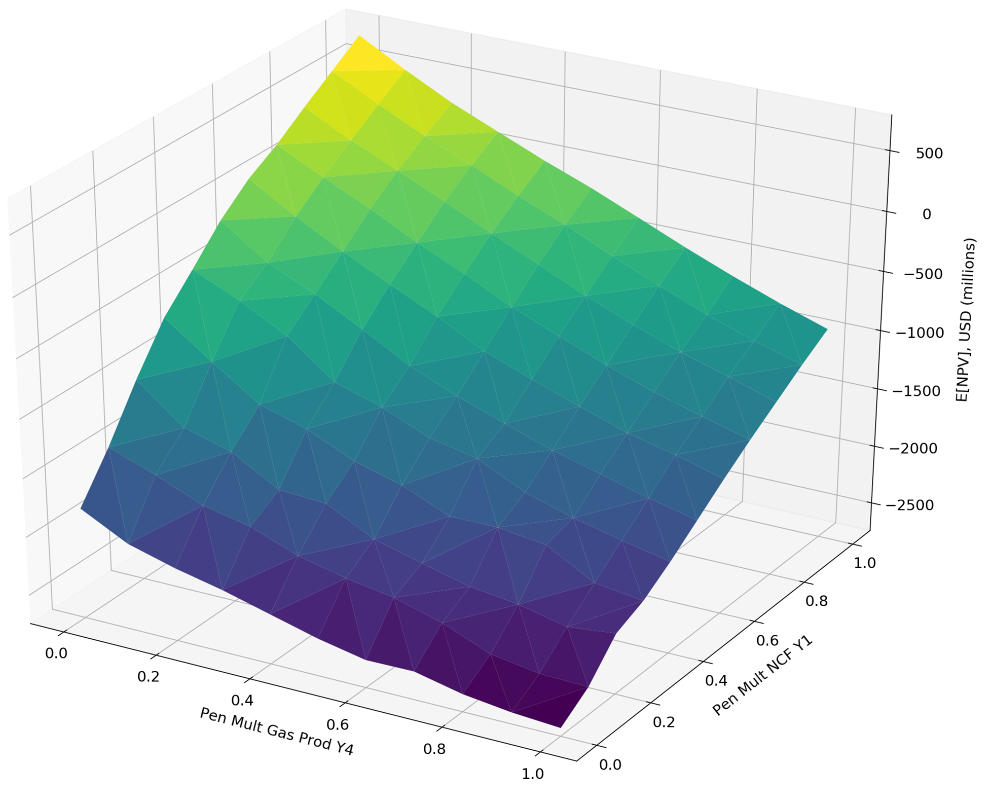

Figure 6.2.17 : Gas Production Year 4 - NCF Year 1

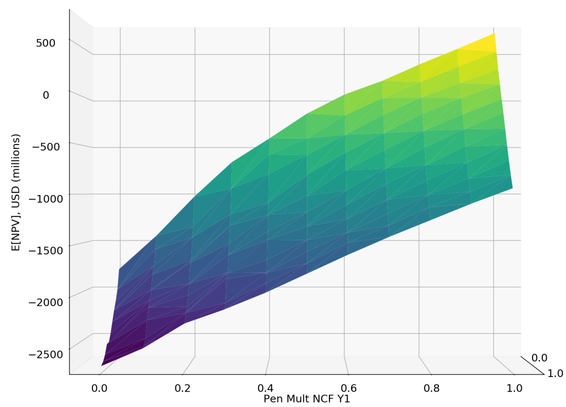

Figure 6.2.18 : NCF Y1 Projection

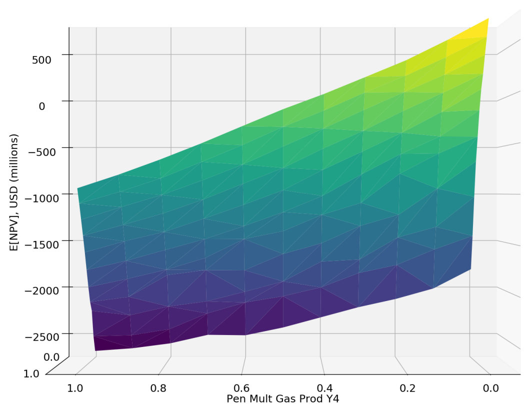

Figure 6.2.19 : G Prod Y4 Projection

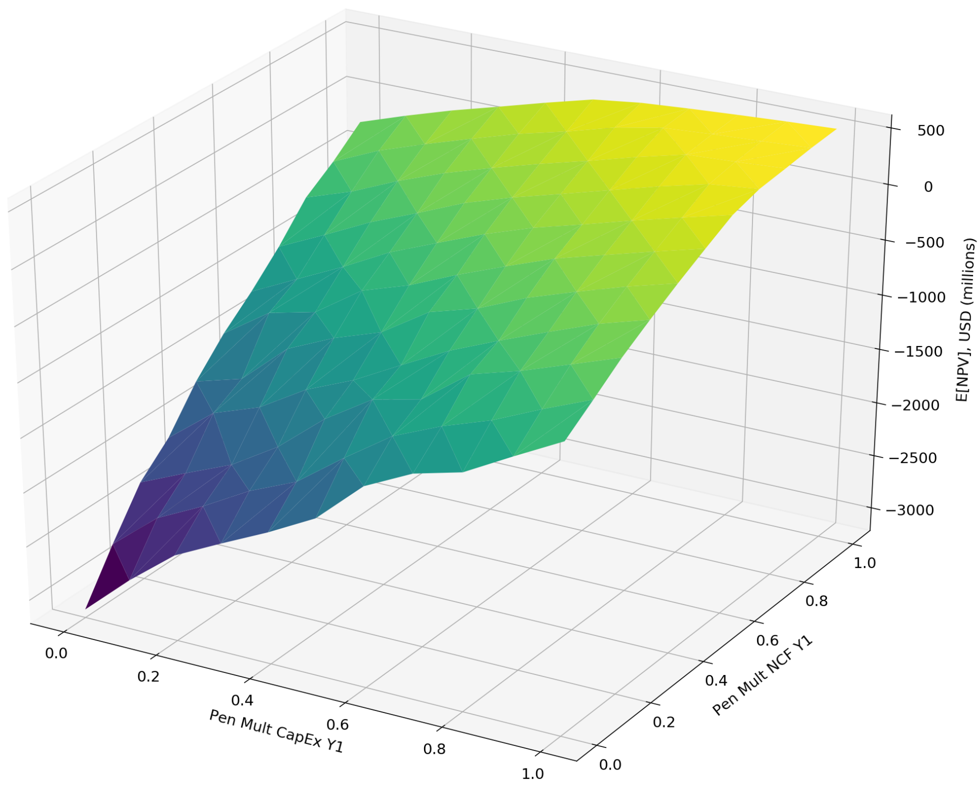

Figure 6.2.20 : CapEx Year 1 - NCF Year 1

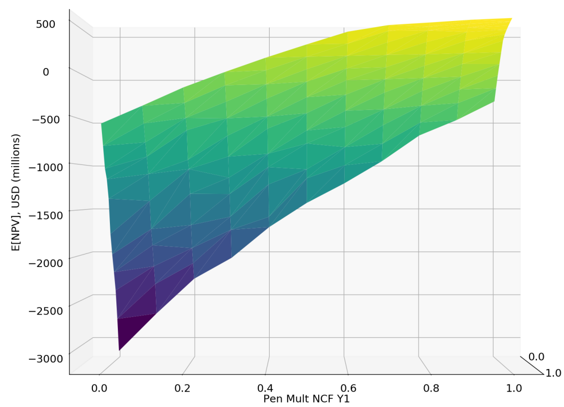

Figure 6.2.21 : NCF Y1 Projection

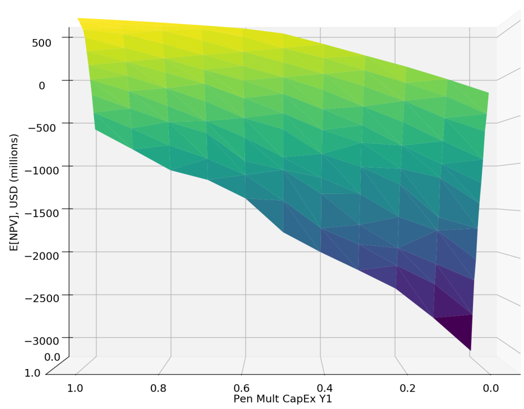

Figure 6.2.22 : CapEx Y1 Projection

6.3 Probability Maximization

In this section, maximization of probability of exceeding a certain corporate goal subject to NPV constraint was performed. Three portfolios were constructed with different NPV constraints: USD 100 million, USD 500 million, and USD 1000 million. The main objective in these cases was the probability of exceeding the gas production of 420 Bscf for year 4.

Figure 6.3.1 illustrates the performance of the probability optimized portfolios. As was stated in the previous sections, natural gas projects with high annual gas production generally have negative NPVs. Moreover, the two projects that produce the largest volumes of gas also have the lowest expected NPV among all projects in the pool (tables 6.0.2, 6.0.3). Thus, in order to facilitate "hard" NPV constraints, the portfolio selection algorithm has to lower the shares of the most productive gas project (table 6.3.1, projects #9, #28). This has a negative impact on the probability of exceeding the gas production target for year 4.

By constructing probability optimized portfolios we concluded that it's not possible to maximize the probability of exceeding corporate goal for Gas Production Year 4 and increase NPV constraints at the same time (see figure 6.3.1). This information can be the basis for reconsideration of the corporate strategy with respect to the company's natural gas business or acquiring suitable natural gas projects that can deliver a large amount of gas in the first years and have positive NPV.

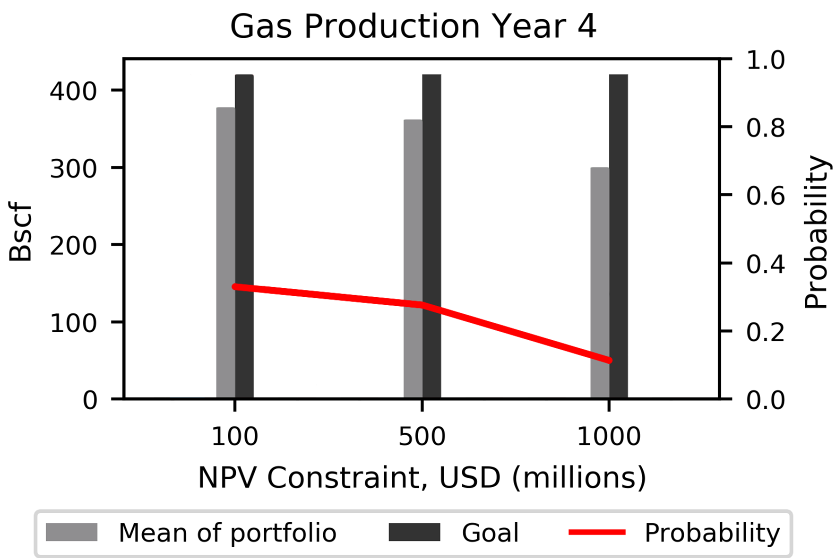

Figure 6.3.1 : Optimized Portfolios with respect to Probability of exceeding Gas Production Goal for Year 4 and constrained by NPV

Table 6.3.1: Portfolio Weights of Gas Projects

Project #	**Portfolio Weights** (%)			**E[Gas Production Y4]**	**E[NPV]**
	NPV constraint = 100 USD (million)	*NPV constraint = 500 USD (million)*	*NPV constraint = 1000 USD (million)*	*Bscf*	*USD (million)*
7	100	100	100	27.5	-81
8	100	100	100	70.7	1032
9	15	3	1	319.7	-18356
10	100	100	99	107.3	-10
22	1	99	37	13.8	-918
23	1	5	3	0	-265
24	99	100	100	29.3	68
25	34	56	0	7.2	-772
26	100	100	100	24.3	123
27	2	10	5	9.7	-1074
28	10	14	2	141.6	-8153
29	95	100	100	10.5	-16
30	99	100	42	41.2	209
E[Portf NPV], USD (million)	**27**	**370**	**958**		

7 Python for Portfolio Optimization

Spreadsheet-based software solutions, such as Microsoft Excel, remain popular among business analysts. A survey conducted annually by KDnuggets online platform for data science and business analytics showed that more than a third of respondents used Excel as their major analytics/data science tool (Piatetsky (2018)). The strengths of the Microsoft Office program is that a relatively small amount of time needed to learn how to perform analyses, the existence of add-on modules available to undertake various statistical tasks, formal user support, and integrations with other Microsoft products. The software was used to construct portfolios in multiple papers (Wood (2016), Willigers et al. (2010), Allan (2007), etc.). However, the use of Excel is impractical in case of large scale problems, as the process becomes time-consuming while the number of assets is increasing.

Contrary to Excel, Python and R are more efficient in the execution of large scale problems, especially considering the use of list comprehension and lapply package, respectively, that handle loops. Moreover, the capability of these programming languages has grown significantly as the number of packages increases over the years. To illustrate, Muenchen (2019) showed the number of add-ons available for R on CRAN package archive network (figure 7.0.1), while Becker (2019) presented active Python packages from the year 2005 to 2018.

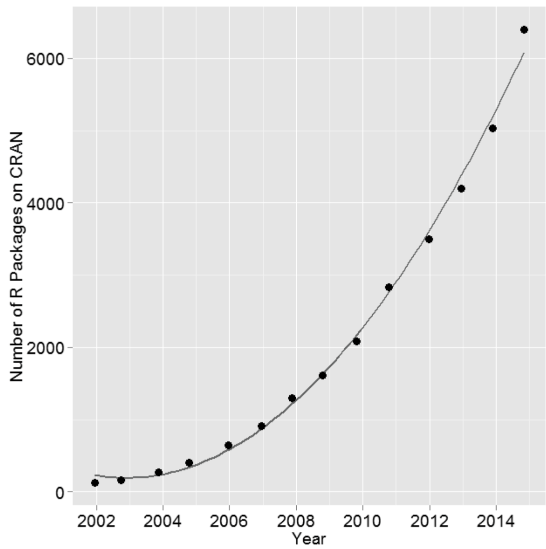

Figure 7.0.1 : Number of Active R Packages available on CRAN (Muenchen (2019))

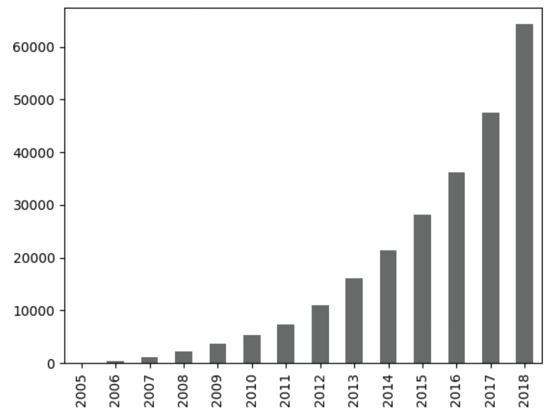

Figure 7.0.2 : Number of Active Python Packages available on PyPi (adapted from Becker (2019))

Nevertheless, the quality of these packages varies greatly. Some of the add-ons are not efficient, while others are limited to solving very specific problems. Furthermore, both programming languages require solid programming skills to build the required portfolio models. Another advantage of Python and R is community forums that provide solutions to a wide variety of problems, including those related to portfolio optimization. The developer-focused industry analyst firm RedMonk analyzed the popularity of programming languages by plotting the number of lines of code written using each language on the GitHub repository against the number of tagged comments on the discussion forum StackOverflow (Muenchen (2019)). According to the RedMonk Programming Language Rankings, Python took the third place, while R was in the top 20 (figure 7.0.3) not very far away from the leaders in terms of Stack Overflow requests. Visual Basic, a programming language used to automate tasks in Excel, is behind the two in terms of popularity on Stack Overflow and GitHub.

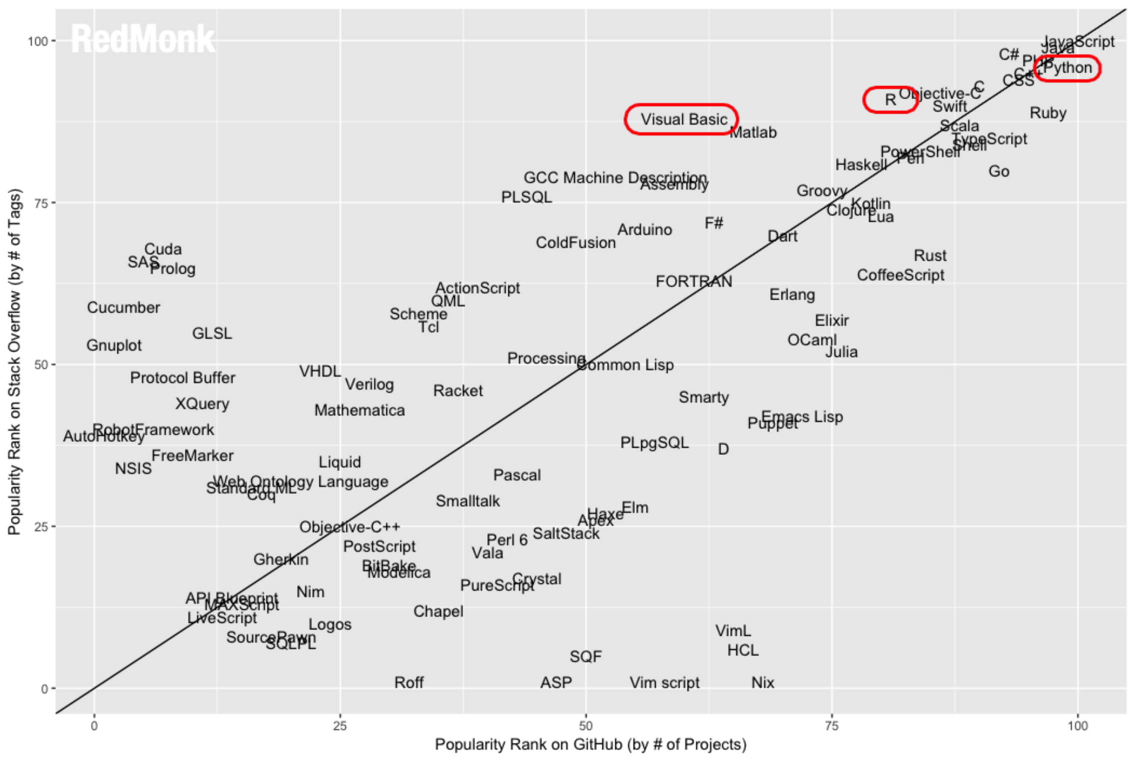

Figure 7.0.3 : Programming Language Rankings 2019 (adapted from O'Grady (2019))

The project model described in chapter 4.3 was built in R and then compared to the model constructed in Python. Despite the fact that the number of Python packages is significantly greater than the number of packages in R, a Python function to sample the values from PERT distributions had to be written from scratch. This is due to the fact that R was primarily designed for statistical computing, while Python is a general-purpose language. Another difference between the languages is relative ease of learning Python that is the main reason of its popularity. In a survey conducted by WP Engine Python was indicated by 9.0% of respondents as the easiest language to learn and took second place in the rating (*How do developers feel about programming languages?* (2017)). R was picked by only 4.4% of respondents and was even included in the list of most complicated languages with 3.6%.

However, the determining factor for the final choice was performance characteristics. Python based models required less than half the run time compared to R-based models. The main reason for this is slow execution of the for-loop in R. Optimized for-loops written in Python and R by using list comprehension and lapply-package respectively, did not improve the relative performance of the model built in R. Considering the number of Monte Carlo realizations and number of projects in the pool, it was decided to use the Python programming language.

Conclusion

The uncertain nature of assets' outcomes and a desire to mitigate these uncertainties were the main reasons why people throughout history have seen the benefits of diversifying their investments. Markowitz (1952) was the first to propose a mathematically sound framework to reduce uncertainties through diversification of financial market investments.

Petroleum companies face a very similar problem as upstream petroleum projects include a large number of uncertain parameters and events including oil and gas prices, exploration success, reserves, production, etc. New portfolio optimization techniques that are more suitable given the multi-objective nature of petroleum portfolio decision were developed. In this work, we have focused on multi-objective time series portfolio optimization, where the objective is NPV maximization subject to corporate constraints through a consistent and comprehensive portfolio selection process.

This method was applied to a set of petroleum projects with uncertain outcomes. We have argued and demonstrated that it is not only the specific investment suggestions of the mathematical optimization that are important but, maybe even more so, also the facilitation of crucial conversations in support of difficult investment decisions. The latter was the motivation for the sensitivity analysis that showed the most significant corporate constraints in terms of their impact on the main objective of the optimized portfolio. Additionally, the analysis helped to identify the projects that provide the largest contribution to the portfolio objective. Given this information, the management team can make the necessary changes in order to improve the portfolio performance, e.g. to adjust the corporate strategy or to acquire projects that will improve corporate performance or farm out a project that does not.

Using poor optimization algorithms may mislead the decision makers by providing results that are not global optima. To address this issue, an overview of optimization algorithms was provided. Applications of the algorithms were discussed, including their advantages and disadvantages.

The project model was built in R and Python. In terms of computational performance Python significantly surpassed R, which was the main reason for using Python for the continued work. A brief comparison of the two languages and Microsoft Excel was provided in the last chapter. We concluded that Excel is a suboptimal choice as the execution time of the portfolio problem increases in case of a realistic number of projects. Programming languages such as R and Python are more efficient, however, they may require time to gain the required programming knowledge and skills to implement and use the model.

References

Allan, Paul (2007). "Consistent Performance: Reducing the Impact of Price Uncertainty through Portfolio Management Practices". In: DOI: 10.2523/82001-ms.

Asness, Clifford S., Andrea Frazzini, and Lasse H. Pedersen (2012). "Leverage aversion and risk parity". In: *Financial Analysts Journal* 68(1), pp. 47–59. ISSN: 0015198X. DOI: 10.2469/faj.v68.n1.1.

Ball, Ben C and Sam L Savage (1999). "Notes on exploration and production portfolio optimization". In: *available for download from http://www. stanford. edu/˜ savage or http://www. ziplink. net/˜ benball.*

Becker, Alex (2019). "Python Packages Since 2005". In: *URL: https://pydist.com/blog/growth-of-python-ecosystem.* [Online; accessed <31.05.2019>].

Begg, Stephen and Nea Smit (2007). "Sensitivity of Project Economics to Uncertainty in Type and Parameters ofPrice Models". In: pp. 1–10. DOI: 10.2523/110812-ms.

Berkelaar, Michel et al. (2015). *lpSolve: Interface to Lpsolve to Solve Linear-Integer Programs.* [Online; accessed <08.05.2019>]. URL: https://CRAN.R-project.org/package=lpSolve.

Bernoulli, Daniel (1954). "Exposition of a New Theory on the Measurement of Risk". eng. In: *Econometrica* 22(1), pp. 23–36. ISSN: 00129682.

Berwin, Turlach, and Andreas Weingessel (2019). *quadprog: Functions to Solve Quadratic Programming Problems.* [Online; accessed <08.05.2019>]. URL: https://CRAN.R-project.org/package=quadprog.

Bodnar, Taras et al. (2018). "Determination and estimation of risk aversion coefficients". In: *Computational Management Science* 15(2), pp. 297–317.

Bratvold, Reidar and Steve Begg (2010). *Making good decisions.* Society of Petroleum Engineers.

Cottle, Richard W and Gerd Infanger (2010). "Harry Markowitz and the Early History of Quadratic Programming". eng. In: *Handbook of Portfolio Construction: Contemporary Applications of Markowitz Techniques.* Springer US: Boston, MA, pp. 179–211. ISBN: 9780387774381.

Dantzig, George B, Alex Orden, Philip Wolfe, et al. (1955). "The generalized simplex method for minimizing a linear form under linear inequality restraints". In: *Pacific Journal of Mathematics* 5(2), pp. 183–195.

DeMiguel, Victor, Lorenzo Garlappi, and Raman Uppal (2009). "The Society for Financial Studies Optimal versus Naive Diversification : How Inefficient Is the 1 / N Portfolio Strategy?" In: *Review of Financial Studies* 22(5), pp. 1915–1953. ISSN: 0893-9454.

DuBois, James (2007). "An Investigation of Risk and Probability in a Portfolio Management Context". In: DOI: 10.2523/71421-ms.

Erdogan, M. et al. (2007). "Optimization of Decision Tree and Simulation Portfolios: A Comparison". In: DOI: 10.2118/68575-ms.

Faya, L C et al. (2007). "SPE 107709 Beyond Portfolio Optimization". In:

Fisher, Irving (1906). *The nature of capital and income.* The Macmillan Company.

Friberg, Henrik A. (2019). *Rmosek: The R to MOSEK Optimization Interface.* [Online; accessed <08.05.2019>]. URL: https://CRAN.R-project.org/package=Rmosek.

Graham, Benjamin (1949). “The Intelligent Investor (Re-issue of the 1949 edition)”. In: *New York: Collins*.

Hedengren, John D. et al. (2014). “Nonlinear modeling, estimation and predictive control in APMonitor”. In: *Computers & Chemical Engineering* 70. Manfred Morari Special Issue, pp. 133–148. ISSN: 0098-1354. DOI: http://dx.doi.org/10.1016/j.compchemeng.2014.04.013. URL: http://www.sciencedirect.com/science/article/pii/S0098135414001306.

Hightower, ML, A David, et al. (1991). “Portfolio modeling: a technique for sophisticated oil and gas investors”. In: *SPE Hydrocarbon Economics and Evaluation Symposium*. Society of Petroleum Engineers.

Holland, JH (1975). “Adaptation in natural and artificial systems: an introductory analysis”. In: *Holland, JH*.

How do developers feel about programming languages? (2017). [Online; accessed <11.06.2019>]. URL: https://wpengine.com/developers-feel-programming-languages/.

Howell, John I. and Peter A. Tyler (2001). “Using Portfolio Analysis to Develop Corporate Strategy”. In: *SPE Hydrocarbon Economics and Evaluation Symposium*(Figure 1), pp. 1–8. DOI: 10.2118/68576-MS. URL: http://www.onepetro.org/doi/10.2118/68576-MS.

J.P Morgan (1997). “CreditMetricsTM – Technical Document”. In: *JP Morgan*, p. 212. URL: http://www.ma.hw.ac.uk/%7B~%7Dmcneil/F79CR/CMTD1.pdf.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. (2001–). *SciPy: Open source scientific tools for Python*. [Online; accessed <08.05.2019>]. URL: http://www.scipy.org/.

Keller, Andre A. and Andre A. Keller (2017). *Extended Multi-Objective Optimization Problems*, pp. 17–45. ISBN: 9783319585642. DOI: 10.2174/97816810856851170100004.

Kolm, Petter N., Reha Tütüncü, and Frank J. Fabozzi (2014). “60 Years of portfolio optimization: Practical challenges and current trends”. In: *European Journal of Operational Research* 234(2), pp. 356–371. ISSN: 03772217. DOI: 10.1016/j.ejor.2013.10.060. URL: http://dx.doi.org/10.1016/j.ejor.2013.10.060.

Lessard Rodney W, Schlumberger Information (2003). “SPE 82012 Portfolio Optimization Techniques for the Energy Industry”. In: (2), pp. 1–6.

Lutz, Mark (2009). *Learning Python*. eng. Beijing ; Sebastopol, California.

Mansini, Renata, Wlodzimierz Ogryczak, and M. Grazia Speranza (2014). “Twenty years of linear programming based portfolio optimization”. In: *European Journal of Operational Research* 234(2), pp. 518–535. ISSN: 03772217. DOI: 10.1016/j.ejor.2013.08.035. URL: http://dx.doi.org/10.1016/j.ejor.2013.08.035.

Markowitz, Harry (1952). “Portfolio Selection”. In: *The Journal of Finance* 7(1). ISSN: 00221082.

Markowitz, Harry (1959). *Portfolio selection : efficient diversification of investments*. eng. New Haven ;

McKerns, Michael, Patrick Hung, and Michael Aivazis (2009). “Mystic: a simple model-independent inversion framework”. In: *URL< http://dev. danse. us/trac/mystic*.

Mitra, Gautam, Frank Ellison, and Alan Scowcroft (2007). “Quadratic programming for portfolio planning: Insights into algorithmic and computational issues”. In: *Journal of Asset Management* 8(3), pp. 200–214. ISSN: 1470-8272. DOI: 10.1057/palgrave.jam.2250075.

Mitra, Sovan (2009). "Risk Measures in Quantitative Finance". In: pp. 1–13. arXiv: 0904.0870. URL: http://arxiv.org/abs/0904.0870.

Muenchen, Robert A (2019). "The popularity of data analysis software". In: *URL: http://r4stats. com/articles/popularity*. [Online; accessed <31.05.2019>].

Nunkesser, Robin (2008). *RFreak-An R-package for evolutionary computation*. Tech. rep. Technical Report//Sonderforschungsbereich 475, Komplexitätsreduktion in ...

O'Grady, Stephen (2019). "The RedMonk Programming Language Rankings: January 2019". In: *URL: https://redmonk.com/sogrady/2019/03/20/language-rankings-1-19/*. [Online; accessed <31.05.2019>].

Okhrin, Yarema and Wolfgang Schmid (2008). "Estimation of optimal portfolio weights". In: *International Journal of Theoretical and Applied Finance* 11(03), pp. 249–276.

Piatetsky, Gregory (2018). "Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis". In: *URl: https://www. kdnuggets. com/2018/05/poll-tools-analytics-data-sciencemachine-learning-results. html*. [Online; accessed <31.05.2019>].

Price, Kenneth, Rainer M Storn, and Jouni A Lampinen (2006). *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media.

Rainville, De et al. (2012). "Deap: A python framework for evolutionary algorithms". In: *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, pp. 85–92.

Reilly, Frank K (2003). *Investment analysis and portfolio management*. eng. Mason, Ohio.

Rubinstein, Mark (2002). "Markowitz's "Portfolio Selection": A Fifty-Year Retrospective". In: LVII(3), pp. 1041–1045.

Scrucca, Luca et al. (2013). "GA: a package for genetic algorithms in R". In: *Journal of Statistical Software* 53(4), pp. 1–37.

Shukla, A., R. Tiwari, and R. Kala (2010). "Genetic algorithm". In: *Studies in Computational Intelligence* 307, pp. 59–82. ISSN: 1860949X.

Storn, Rainer and Kenneth Price (1995). "Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces". In: *Technical Report TR-95-012*, pp. 1–12. DOI: 10.1.1.1.9696. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.9696.

Tobin, J. (2006). "Liquidity Preference as Behavior Towards Risk". In: *The Review of Economic Studies* 25(2), p. 65. ISSN: 00346527. DOI: 10.2307/2296205.

Venkataraman, P. (2002). *Applied Optimization with MATLAB Programming*. A Wiley-Interscience publication. Wiley. ISBN: 9780471349587. URL: https://books.google.no/books?id=sw93dY5JcDMC.

Von Neumann, John and Oskar Morgenstern (2007). *Theory of games and economic behavior (commemorative edition)*. Princeton university press.

Willigers, Bart J.A. and Felix Majou (2010). "Creating Efficient Portfolios That Match Competing Corporate Strategies". In: *SPE Economics & Management* 2(01), pp. 12–18. ISSN: 2150-1173. DOI: 10.2118/129259-pa.

Wood, David A. (2016). "Asset portfolio multi-objective optimization tools provide insight to value, risk and strategy for gas and oil decision makers". In: *Journal of Natural Gas Science and Engineering* 33, pp. 196–216. ISSN: 18755100. DOI:

10.1016/j.jngse.2016.03.103. URL: http://dx.doi.org/10.1016/j.jngse.2016.03.103.