Two Optimization Problems:

A Transshipment Problem and A Risk Minimizing Problem

Introduction

This report presents a comprehensive and in-depth analysis of optimization methodologies applied to two distinct, real-world problems. The primary objective of this report is to provide a detailed examination of the context, mathematical formulations, and solutions for both parts of the project, demonstrating the effectiveness and applicability of optimization techniques in addressing complex challenges across various domains.

The first part of the project focuses on the Rockhill Shipping & Transport Company's logistics problem involving hazardous waste transportation and transshipment. Linear programming techniques will be employed to optimize shipping routes and minimize costs. The second part tackles the challenge of investment allocation in a diverse portfolio, using quadratic programming techniques to derive optimal investment allocations while minimizing risk.

***Keywords*:** Optimization, Linear Programming, Quadratic Programming, Python.

Part 1 Rockhill Shipping & Transport Company

This part examines a logistics problem faced by the Rockhill Shipping & Transport Company. The company is negotiating a contract with Chimotoxic, which involves transporting hazardous waste products from six manufacturing plants to three waste disposal sites. Given the sensitive nature of the cargo, transportation costs and route constraints must be carefully considered. This section of the report thoroughly explores the complexities of the shipping problem, the constraints associated with transporting hazardous materials, and the need for efficient route optimization thoroughly. By employing linear programming techniques, we seek to determine the optimal shipping routes that minimize Rockhill's total transportation costs while adhering to the constraints imposed by the problem.

I. Define the decision variables

The problem involves the transportation of waste from six plants, namely Denver, Morganton, Morrisville, Pineville, Rockhill, and Statesville, to three waste disposal sites, Orangeburg, Florence, and Macon. In addition to direct shipping between plants and sites, transshipment is also taken into account, implying the transfer of waste between plants and between waste disposal sites. To formulate the problem mathematically, three sets of decision variables are introduced. The first set, denoted by x[i][j], represents the number of waste barrels transported directly from plant i to waste disposal site j. The second set, denoted by y[i][j], refers to the number of barrels transported from plant i to plant j or waste disposal site j. The third set, denoted by z[i][j], represents the number of barrels shipped between waste disposal site i and waste disposal site j. All decision variables are stored in a matrix and displayed below.

$$X = \begin{bmatrix} x_{00} & x_{01} & x_{02} \\ x_{10} & x_{11} & x_{12} \\ x_{20} & x_{21} & x_{22} \\ x_{30} & x_{31} & x_{32} \\ x_{40} & x_{41} & x_{42} \\ x_{50} & x_{51} & x_{52} \end{bmatrix} \qquad Y = \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} & y_{04} & y_{05} \\ y_{10} & y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ y_{20} & y_{21} & y_{22} & y_{23} & y_{24} & y_{25} \\ y_{30} & y_{31} & y_{32} & y_{33} & y_{34} & y_{35} \\ y_{40} & y_{41} & y_{42} & y_{43} & y_{44} & y_{45} \\ y_{50} & y_{51} & y_{52} & y_{53} & y_{54} & y_{55} \end{bmatrix} \qquad Z = \begin{bmatrix} z_{00} & z_{01} & z_{02} \\ z_{10} & z_{11} & z_{12} \\ z_{20} & z_{21} & z_{22} \end{bmatrix}$$

The transportation costs associated with waste shipments from the plants to the waste disposal sites, between the plants, and between the waste disposal sites are represented by the matrices C1, C2, and C3, respectively. These matrices contain the transportation costs associated with each possible route.

$$C1 = \begin{bmatrix} 12 & 15 & 17 \\ 14 & 9 & 10 \\ 13 & 20 & 11 \\ 17 & 16 & 19 \\ 7 & 14 & 12 \\ 22 & 16 & 18 \end{bmatrix} \quad C2 = \begin{bmatrix} - & 3 & 4 & 9 & 5 & 4 \\ 6 & - & 7 & 6 & 9 & 4 \\ 5 & 7 & - & 3 & 4 & 9 \\ 5 & 4 & 3 & - & 3 & 11 \\ 5 & 9 & 5 & 3 & - & 14 \\ 4 & 7 & 11 & 12 & 8 & - \end{bmatrix} \quad C3 = \begin{bmatrix} - & 12 & 10 \\ 12 & - & 15 \\ 10 & 15 & - \end{bmatrix}$$

Additionally, the number of waste generated by each plants are stored in the vector WastePerweek and the disposal capacities of each waste proposal site are stored in the vector DisposalCapacities.

$$WastePerweek = \begin{bmatrix} 45 & 26 & 42 & 53 & 29 & 38 \end{bmatrix}$$

$$DisposalCapacities = \begin{bmatrix} 65 & 80 & 105 \end{bmatrix}$$

II. Case without transshipment

In this case, only the direct shipping between plants and sites are considered.

1. Objective function

The objective is to minimize the total transportation cost, which is the sum of the products of the number of barrels shipped between each pair of locations and the corresponding shipping cost. In this case, the objective function is

*Minimize: C1 * X*

where C1 is the cost matrix for shipping waste from plants to sites,

X is the matrix storing the numbers of waste barrels transported directly from plant i to waste disposal site j.

2. Constraints

In this case, the constraints and the mathematical formulation are listed below:

Constraint 1:

The total waste shipped from each plant equal to the waste generated by the plant. In other words, for each plant i (i = 0, 1, 2, 3, 4, 5), the sum of the waste shipped to all waste disposal sites (j = 0, 1, 2) should be equal to the weekly waste production at that plant.

$$\textit{For i in \{0,1,2,3,4,5\}},\ \sum x[i,j] \text{ for j in } \{0,1,2\} = \textit{WastePerweek}[i]$$

Constraint 2:

The total waste received at each waste disposal site cannot exceed its maximum capacity. Namely, for each waste disposal site j, the sum of the waste received from all plants (i = 0, 1, 2, 3, 4, 5) should not exceed the capacity of that disposal site.

$$\textit{For j in \{0,1,2\}},\ \sum x[i,j] \text{ for i in } \{0,1,2,3\ ,4,5\} \le \textit{DisposalCa pacities}[j]$$

Constraint 3:

The amount of waste shipped from plant i to waste disposal site j is integer and cannot be negative.

3. Implement the Linear Programming model

Python can be used to implement linear programming models for optimization problems. To solve the problem, the Pulp library in Python can be utilized. The first step is to import the Pulp library and define the problem as minimizing the overall shipping costs. Next, the decision variables are established using the LpVariable function from Pulp, which specifies the integer decision variables for each waste-producing location and disposal facility. The code then identifies the shipping costs associated with the problem and stores them in a matrix, while also defining the waste production and disposal site capacities in lists. The objective function is then formulated by summing the product of the shipping costs and the decision variables. The code defines two constraints, which is identified as Constraint 1 and Constraint 2 in the previous part. Finally, the code solves the problem and presents the optimal solution, which shows the values of the decision variables that minimize the total shipping costs. The resultant optimal solution is presented in Table 1, with an optimal cost of $2,988.

Table 1 Optimal Solution for the Case without Transshipment

Unit: barrels	Orangeburg	Florence	Macon
Denver	45	0	0
Morganton	0	0	26
Morrisville	0	0	42
Pineville	0	53	0
Rockhill	20	0	9
Statesville	0	27	11

The specific distribution of waste is as follows:

Denver needs to convey 45 barrels of waste to Orangeburg;

Morganton needs to convey 26 barrels of waste to Macon;

Morrisville needs to convey 42 barrels of waste to Macon;

Pineville needs to convey 53 barrels of waste to Florence;

Rockhill needs to convey 20 barrels of waste to Orangeburg and 9 barrels of waste to Macon;

Statesville needs to convey 27 barrels of waste to Florence and 11 barrels of waste to Macon.

III. Case with transshipment

In this case, transshipment is considered in the waste management process, referring to the transfer of waste between different plants and waste disposal sites. This means that waste generated at one plant may be transferred to another plant for further processing, and waste from one site may be transported to another site for final disposal. Transshipment can help to optimize the waste management process by ensuring that waste is processed and disposed of in the most efficient and cost-effective manner.

1. Objective function

The objective is to minimize the total transportation cost. In this case, the objective function is

$$\text{Minimize: } C1 * X + C2 * Y + C3 * Z$$

where C1 is the cost matrix for shipping waste from plants to sites,

C2 is the cost matrix for shipping waste between the plants,

C3 is the cost matrix for shipping waste between the waste disposal sites,

X is the matrix storing the numbers of waste barrels transported directly from plant i to waste disposal site j, *Y is the matrix storing the number of waste barrels transported form plant i to plant k,*

Z is the matrix storing the number of waste barrels transported from site j to site l.

2. Constraints

In this scenario, the constraints and the mathematical formulation are listed below:

Constraint 1:

The total waste sent from each plant (i) to disposal sites, combined with the waste sent to other plants, should be equal to the waste generated by each plant plus the waste received from other plants. This constraint ensures that the waste management process is balanced and all waste is accounted for in the transportation plan.

For i in {0,1,2,3,4,5},

$$\sum x[i,j] \text{ for j in } \{0,1,2\} + \sum \mathrm{y[i,k]} \text{ for k in } \{0,1,2,3,4,5\} = WastePerweek[i] + \sum \mathrm{y[k,i]} \text{ for k in } \{0,1,2,3,4,5\}$$

Constraints 2:

The total waste received at each waste disposal site cannot exceed its maximum capacity. In other words, for each waste disposal site j, the sum of the waste received from all plants (i = 0, 1, 2, 3, 4, 5) plus the sum of the waste received from other sites should not exceed the capacity of that disposal site. This constraint is represented mathematically as follows:

$$\textit{For j in \{0,1,2\},} \quad \sum x[i,j] \text{ for i in } \{0,1,2,3\ ,4,5\} + \sum \mathrm{z[l,j]} \text{ for l in } \{0,1,2\} \le DisposalCapacities[j]$$

Constraint 3:

The amount of waste shipped are integers and cannot be negative.

3. Implement the Linear Programming model in Python

Similarly, after implementing the Linear Programming model in Python, the optimal solution is listed in the Table 2, with the optimal cost of $2,674. The solution includes transshipping between plants; however, there is no transshipment between waste proposal sites.

Table 2 Optimal Solution for the Case with Transshipment

Unit: barrels	Denver	Morganton	Morrisville	Pineville	Rockhill	Statesville
Denver	0	45	0	0	0	0
Morganton	0	0	0	0	0	0
Morrisville	0	0	0	0	0	0
Pineville	0	17	0	0	36	0
Rockhill	0	0	0	0	0	0
Statesville	0	0	0	0	0	0

Unit: barrels	Orangeburg	Florence	Macon
Denver	0	0	0
Morganton	0	42	46
Morrisville	0	0	42
Pineville	0	0	0
Rockhill	65	0	0
Statesville	0	38	0

The specific distribution of waste is as follows:

Denver needs to convey 45 barrels of waste to Morganton plant;

Morganton needs to convey 42 barrels of waste to Florence and 46 barrels of waste to Macon;

Morrisville needs to convey 42 barrels of waste to Macon;

Pineville needs to convey 17 barrels of waste to Morganton plant and 36 barrels of waste to Rockhill plant;

Rockhill needs to convey 65barrels of waste to Orangeburg;

Statesville needs to convey 38 barrels of waste to Florence.

The solution obtained from the Linear Programming model reveals that only two intermediate shipping points, Morganton and Rocjhiis, are utilized in the optimal solution. It is notable that Morganton is the more frequently used intermediate shipping point, suggesting that it is likely more cost-effective to transport waste through this plant. The relative lower cost of transportation from these two plants to the waste proposal sites may account for their prominence in the optimal solution. This finding has practical implications for waste management decision-making, as it highlights the importance of considering transportation costs when determining the optimal waste management strategy.

Moreover, the comparison of the two optimal costs reveals that the case with transshipment is more cost-effective than the one without transshipment.

Conclusion for Part 1

Through the application of linear programming methodology, this study has determined the optimal shipping routes for two cases: with and without transshipment. The optimal solution for the case without transshipment is presented in Table 1, yielding an optimal cost of $2,988. Alternatively, the optimal solution for the case with transshipment is presented in Table 2, with an optimal cost of $2,674. Comparing the optimal costs, it can be concluded that incorporating transshipment prove to be a more economical option than not using it, as it results in lower total transportation costs.

These findings suggest that the incorporation of transshipment in the waste management process has a significant impact on the total transportation cost. The use of transshipment can help to optimize the waste management process by ensuring that waste is processed and disposed of in the most efficient and cost-effective manner. Thus, the results of this study emphasize the importance of considering the transshipment option in the waste management process for Rockhill Shipping & Transport Company.

Furthermore, the optimal solutions obtained from the linear programming model provide a useful insight into the optimal shipping routes and their respective costs, which can inform the development of a contract proposal to submit to Chimotoxic for waste proposal. The findings of this study can guide decision-making in waste management, enabling Rockhill Shipping & Transport Company to minimize costs while adhering to constraints associated with transporting hazardous materials.

Part 2 Investment Allocations

In this part, we shift our focus to the financial domain, specifically addressing the challenge of allocating investments across various asset types to optimize risk and return. The problem involves determining the optimal distribution of a $10,000 investment in a portfolio comprising diverse assets such as bonds, high-tech stocks and gold. The primary goal is to achieve a minimum baseline expected return while minimizing the associated risk. This section of the report will explore the intricacies of investment allocation, the role of the covariance matrix in quantifying risk, and the utilization of quadratic programming to obtain the optimal investment allocations. By solving the problem using this optimization technique, we also aim to gain a deeper understanding of the relationship between risk and return in investment portfolios.

I. Define the decision variables, objective functions and constraints.

The decision variables are the proportions invested in the six investments: bonds, high tech stocks, foreign stocks, call options, put options and gold, which are denoted by x1,x2, x3, x4, x5 and x6.

The objective is to minimize the risk and the objective function is :

$$Z = X^T * S * X$$

where X is a vector representing allocation in percentage of the investment amounts

X^T is the transpose of X

S is the covariance matrix of asset returns

In the objective function f(x) = X^T * S * X, the product of x and its transpose is used to compute the portfolio risk, which is represented as a quadratic form. This operation generates a matrix, which is then multiplied by the covariance matrix S to yield a scalar value indicating the portfolio risk. This approach is frequently employed in portfolio optimization to estimate the risk of a portfolio based on its allocation.

In addition, the expected returns for all the investment are stored in the vector R:

$$R = \begin{bmatrix} 0.07 & 0.12 & 0.11 & 0.14 & 0.14 & 0.09 \end{bmatrix}$$

All constraints are listed below:

1. $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 1$

2. *For i in {0,1,2,3,4,5},* $\sum x_i r_i \geq 0.11$

3. $x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$

II. Implement the Quadratic Programming model in Python

The SciPy library in Python includes a quadratic programming solver for optimization problems. To implement a quadratic programming model in Python, one needs to define the objective function and constraints as Python functions and use the minimize() function from the scipy.optimize module to solve the optimization problem. The objective function is a quadratic function that balances the trade-off between risk and return, while the constraints ensure that the allocations of the portfolio add up to 100% and meet the minimum expected return requirement. The minimize() function can be called with the objective function, constraints, and bounds on the variables, which will return a set of optimal variable values that minimize the objective function subject to the constraints.

Table 3 shows the optimal allocation outcome, resulting in an overall expected return of 11%.

Table 3 Optimal Solution for the Allocation

	Allocation in Percentage
Bonds	19.54%
High Tech Stocks	13.95%
Foreign Stocks	18.66%
Call Options	11.18%
Put Options	20.80%
Gold	15.87%

Hence, to pursue the minimum expected return of 11% at a minimum risk, our investor are supposed to invest $1,954 in bonds, $1,395 in high tech stocks, $1,866 in foreign stocks, $1,118 in call options and $1,587 in gold.

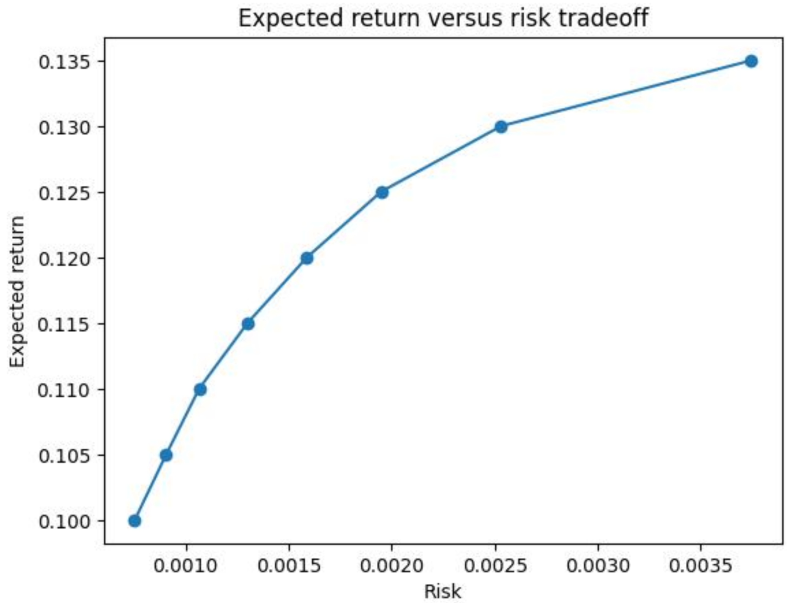
III. Risk-Return Trade-off in Portfolio Optimization: A Pattern Analysis.

This section analyzes the relationship between risk and expected return in a portfolio by solving a quadratic programming model for successive baseline return values. The obtained solution pairs are then plotted to examine any patterns or relationships that may exist between the minimized risk and expected portfolio return.

Likewise, the SciPy library is utilized. In addition to the previous operation, a series of minimum returns are stored in an array and the risk-return pairs are then plotted, with risks and returns extracted from the optimization results.

The optimization results show that the risks increase from 0.00075 to 0.0037 as the expected portfolio returns increase from 0.1 to 0.135. This is demonstrated by the pairs of results (risk, return) generated from successive values of 10%, 10.5%, 11%, 11.5%, 12%, 12.5%, 13%, and 13.5% used as the baseline return values. Figure 1 shows the relationship between the expected returns and the risks. The pattern shows a positive correlation between risk and return, suggesting that investors can expect higher returns as they take on more risk.

Figure 1. Expected Return with the Risk



Conclusion for Part 2

Part 2 of this report focuses on the challenge of allocating investments across various asset types to optimize risk and return. The primary goal is to achieve a minimum baseline expected return while minimizing the associated risk. Additionally, a pattern analysis is conducted to examine the relationship between risk and expected return in a portfolio by solving a quadratic programming model for successive baseline return values. The obtained solution pairs are plotted, and the relationship between the minimized risk and expected portfolio return is examined.

In conclusion, the utilization of quadratic programming in investment allocation problems enables investors to achieve their investment goals while optimizing the risk and return trade-off. By implementing the model in Python using the SciPy library, we are able to obtain the optimal allocation and further analyze the relationship between risk and expected return in a portfolio.

References

1. Fischer, M. (2021). Linear Programming in Python. Real Python. Retrieved April 3, 2023, from https://realpython.com/linear-programming-python

2. Machine Learning Geek. (n.d.). Transshipment problem in Python using PuLP. Retrieved April 3, 2023, from https://machinelearninggeek.com/transshipment-problem-in-python-using-pulp/

3. 2.Mitchell, J. E., & Jean-Paul, W. (2021). Pulp documentation. COIN-OR. Retrieved April 3, 2023, from https://coin-or.github.io/pulp/main/index.html

4. Real Python. (n.d.). Python scipy cluster optimize. Retrieved April 3, 2023, from https://realpython.com/python-scipy-cluster-optimize/