# Decision Community Challenge July 2024 – Smart Investment

Rob Parker

## Introduction

The decision management community posed a monthly challenge. Given a set of stocks, the current price and future expected price, build a portfolio to maximise the return subject to risk management constraints.

## Approach

This approach used generative AI to generate a mixed integer model directly from the problem statement. In particular, ChatGPT (3.5.x) was used to generate a ZIMPL[1] model directly from the problem description. ZIMPL is the Zurich Institute's Mathematical Programming Language. The problem description was copied directly from the Decision Community website into a ChatGPT dialogue with the added line requesting generation of a ZIMPL model to optimize the portfolio. The generated ZIMPL model is shown in Appendix B.

The raw generated ZIMPL model was essentially semantically correct with some syntactic errors. The syntactic errors were typically of the form;

set STOCKS := ABC XYZ TTT LMN; -> set STOCKS := {"ABC", "XYZ", "TTT", "LMN"};

param current_price := ABC 25... -> param current_price[STOCKS] := <"ABC"> 25,...

var invest{STOCKS} >= 0; -> var invest[STOCKS] >= 0;

sum{s in STOCKS} -> sum <s> in STOCKS :

The output from ChatGPT was corrected for the syntax errors as shown above. The syntactically corrected ZIMPL model is shown in Appendix A.

## Results

The problem is readily solved using a mixed integer solver. In this case the solution was found using SCIP.

```
SCIP> solution status: optimal solution found
objective value:                                  4650
invest$ABC                                        3000        (obj:0.4)
invest$XYZ                                        1000        (obj:0.2)
invest$TTT                                        1000        (obj:0.25)
invest$LMN                                        5000        (obj:0.6)
```

## Appendix (A) Revised ZIMPL Model

```
# Define the stocks
set STOCKS := {"ABC", "XYZ", "TTT", "LMN"};
# Parameters
param current_price[STOCKS] :=
    <"ABC"> 25,
    <"XYZ"> 50,
    <"TTT"> 100,
    <"LMN"> 25;
param projected_price[STOCKS] :=
    <"ABC"> 35,
    <"XYZ"> 60,
    <"TTT"> 125,
    <"LMN"> 40;
param total_investment := 10000;
param max_investment_per_stock := 5000;
param min_investment_per_stock := 1000;
# Decision variables
var invest[STOCKS] >= 0;
```

**cont…**

## Appendix (A) Revised ZIMPL Model (Continued…)

```
# Objective: Maximize the total projected gain
maximize Total_Gain:
sum <s> in STOCKS :
  (projected_price[s] - current_price[s]) * invest[s] / current_price[s];

# Constraints
subto Investment_Limit:
sum <s> in STOCKS :
  invest[s] <= total_investment;

subto Max_Investment:
forall <s> in STOCKS:
 invest[s] <= max_investment_per_stock;

subto Min_Investment:
forall <s> in STOCKS:
 invest[s] >= min_investment_per_stock;
```

## Appendix (B) ChatGPT ZIMPL Model

```
# Define the stocks
set STOCKS := ABC XYZ TTT LMN;

# Parameters
param current_price :=
   ABC 25
   XYZ 50
   TTT 100
   LMN 25;

param projected_price :=
   ABC 35
   XYZ 60
   TTT 125
   LMN 40;

param total_investment := 10000;
param max_investment_per_stock := 5000;
param min_investment_per_stock := 1000;

# Decision variables
var invest{STOCKS} >= 0;
```

**cont…**

## Appendix (B) ChatGPT ZIMPL Model (Continued…)

```
# Objective: Maximize the total projected gain

maximize Total_Gain: sum{s in STOCKS} (projected_price[s] - current_price[s]) * invest[s] /
current_price[s];


# Constraints

s.t. Investment_Limit: sum{s in STOCKS} invest[s] <= total_investment;


s.t. Max_Investment: forall{s in STOCKS} invest[s] <= max_investment_per_stock;


s.t. Min_Investment: forall{s in STOCKS} invest[s] >= min_investment_per_stock;
```

## References

[1] Thorsten Koch,
*Rapid Mathematical Programming*,
Berlin 2004.