# A Discussion on AMPL Models For Portfolio Management

Hongyu Li

Dr. Vardges Melkonian

Math Department

Ohio University

Athens, OH 45701

hl133609@ohio.edu

www.ohio.edu/people/hl133609

# Abstract

This is a final report for independent study with Dr. Vardges Melkonian on Integer Programming, Discrete Optimizations and AMPL. The goal of this paper is to demonstrate the models created with AMPL, which optimize the returns on portfolio in a certain time horizon and to compare them within same data in order to show the advantages and disadvantages among each other. The data collected for this paper was recorded from the Shanghai Stock Exchange. The three models we were using were created based on different perspectives, such as risk-averse, profit maximizing and risk-return balanced. Since there is always risks among all level of returns, the models are aimed to optimize the return with minimized risk level.

# Contents

# 1. Introduction

It is human nature that chasing on profits without exposing to the corresponding risk. But it is too ideal to be true. There is always risk among opportunities. In finance, an investor always considers how much risk he or she is able to take when making a decision on investment. As a result, portfolio management was created to balance the risk and return. Although the output of a portfolio is simple, it requires rounds of assumptions and modifications before the finalized “report” is ready to be displayed. Based on the rules of portfolio making and optimization procedures, we established three AMPL models under some assumptions and special cases.

In our models, we chose stocks from different sectors, for instance, Banks, Real Estate and Insurance Company. And for running the optimization, we established a T times horizons excluding the initial stage. In each stage, there were different economic scenarios which might happen to individual stock with respect to pre-assigned probabilities. In addition, the stock prices on a stage were dependent on the decisions made on the previous stages.

## 1.1. Acronyms and Abbreviations

In this report and the models we created, Abbreviations are applied to simplify the notation and expressions, especially in our data sets. We used Chinese alphabets to show the stock names.

# 2. Description of the Models

## 2.1 Components in AMPL models

Before we discuss our three models, we need to show the components in AMPL models: sets, parameters, decision of variables, objective function, and constraints.

### 2.1.1. Sets

We have defined four types of sets.

- Scenarios

  The set scenarios is the set of all possible scenarios is terms of economy. For instance, there are three types that we can consider: Increasing, Decreasing and Constant.
- Sectors

  The set sectors is the set of all possible investment areas. It is a way to spread our portfolio onto a larger scale of stocks with respect to the types of companies.
- Stocks in Sector

  The set $\text{stocks\_sectors}_{i2sectors}$ is the set of all possible stocks in the investment sector I that we have created.
- Stocks

  This set is a union of all stocks in all sectors that we have include in the above sets.

### 2.1.2 Parameters

Parameters are pre-assigned by the users

- Probability

  It carries the probability of occurring of a scenario at each stage. Thus, this is a column vector whose sum of probabilities equals one
- Principle Amount

Parameter P stands for the initial amount of money that one investor is going to spend. The unit is in dollars.

- Time Horizon

  We use T to denote the total number of stages after the initial stage. Thus, T= 0 stands for the initial stage.
- Portfolio Volatility

  This parameter denotes the upper limit of risk forced by an individual investor. This is given in percentage. This parameter occurs in the profit-oriented model
- Weight

  This parameter represents the proportion of the risk inside the objective function along with the return. It is introduced in the balanced model.
- Stock Prices

  There are three parameters defined here:
  - Initial stock price_j It is the current market price of stock j
  - Stock price_j,t,s is the market price of stock j at stage t in scenario s. this is a preassigned value by assumption and prediction.
  - Estimate stock price_j,t is the estimated price of stock j at stage t
- Standard Deviation

  The parameter std_dev_j,t is for denoting the standard deviation of the price of the stock j at stage t
- Coefficient of Variation

  To examine the significance of variability with respect mean, we need to compute the coefficient of variation. CV_j,t denotes the ratio of the standard deviation of stock j at stage t and the expected stock price of j at stage t
- Dividend

  D_j,t,s is the expected Dividend of stock j at stage t.
- Limits for Diversification

  The parameter prop_ub denotes the upper limit of the proportion of the total money that can be invested in sector i at each stage.

  The parameter prop_lb denotes the lower limit of the proportion of the total money that should be invested in sector I at each stage.

### 2.1.3 Decision Variables

Decision Variables are the variables that need to be computer by the AMPL model. There are two decision variables in our models:

- The number of stocks at the end of each stage

  Variable num_stocks is used to record the number of shares bought for stock j at the end of stage t. We should here note that if the number of shares for stock j is increased compared with its previous stage, we are buying more shares of stock j. vice versa for decreasing values.
- Cash left in hand at the end of each stage

  This variable cash_balance is introduced to record the Cash left in hand at the end of each stage after purchasing and selling stocks.

### 2.1.4 Objective Function

- Profit-oriented model: maximize final return

We are seeking the maximized total expected return that can be collected at the end of stage T. we can obtain this by adding the cash left at the end of stage T and the total price of all stocks possessed at that stage, since the parameter volatility is pre-assigned.

- Risk-averse Model: minimize risk

  In this model, we are interested in minimizing the sum of products between the Coefficient of variables and the number of corresponding stocks at all stages.
- Balanced model: maximize the final return and minimize the risk with weights assigned.

#### 2.1.5 Constraints

- Constraints for Cash Balance

  When t = 0, Cash Balance available at the end of initial stage is the principal investment subtracted by the amount invested for buying stocks at the beginning.

  When t > 0, cash left at the end of stage t consists of the cash left in hand at the previous stage, the cash obtained as dividend paid by the company and the cash obtained by selling some stocks minus the cash spent for buying some stocks.
- Constraints for Diversification

  This may be common that we buy stocks from only one sector or we do not buy any stocks from one sector. This is a problem for reducing risks in finance. To overcome this problem, we restrict the minimum and maximum amount of money to be invested in a particular sector. For simplicity, we have given the upper and lower limits of the proportions of total money at any stage t. these can be addressed by introducing the following constraints.
- Constraint for Portfolio Volatility (for profit-oriented model)

  There is always a certain type of risk in investment. In modern portfolio theory, standard deviation is regarded as the measurement of risk. We seek to maximize portfolio return given certain risk. In our risk-averse model, we introduce this constraint to restrict our product of CV of stocks and shares of stocks.

### 2.2 Types of Models

The ideas of making AMPL models are based on the needs of investors: some investors are risk-averse while others may not be as much sensitive as them.

#### 2.2.1 Description of Profit-oriented Model

The value of parameter Portfolio Volatility can be adjusted to meet different levels of risk requirements. The goal is to show the maximized return with a fixed level of risk.

#### 2.2.2 Description of Risk-averse Model

The objective function in this model is to minimize the risk, with pre-assigned percentage of returns. This model is the “inverse-thinking” of the first model. As the level of risk is always minimized.

#### 2.2.3 Description of Balanced Model

This model is a combination of the previous two models regarding the objective function. With pre-assigned weight (i.e. the percentage of risk in the objective function), we can observe the relation between the risk and return.

# 3. Solution and Comparison

## 3.1. Solution Chart

| MODEL 1 | | | | MODEL 2 | | | MODEL 3 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| risk lvl % | return | return % | # iterations | risk lvl % | return % | # iterations | risk lvl % | return % | # iterations | weight |
| 11.00% | infeasible | | | 9.23% | 10.00% | 106 | 8.66% | 11.55% | 124 | 1 |
| 12.00% | 23722.355 | 18.61% | 308 | 9.18% | 11.00% | 111 | 8.66% | 11.92% | 1.24 | 0.99 |
| 12.20% | 24049.341 | 20.25% | 336 | 9.33% | 12.00% | 103 | 8.64% | 13.16% | 125 | 0.95 |
| 12.40% | 24421.039 | 22.11% | 268 | 9.47% | 13.00% | 110 | 9.03% | 17.15% | 114 | 0.9 |
| 12.60% | 24534.183 | 22.67% | 308 | 9.61% | 14.00% | 96 | 10.71% | 26.20% | 107 | 0.8 |
| 12.80% | 24700.77 | 23.50% | 268 | 9.81% | 15.00% | 123 | 11.35% | 28.55% | 116 | 0.73 |
| 12.90% | 24815.726 | 24.08% | 221 | 10.00% | 16.00% | 110 | 12.05% | 30.40% | 108 | 0.7 |
| 12.92% | 24833.553 | 24.17% | 220 | 10.20% | 17.00% | 123 | 13.33% | 33.08% | 102 | 0.65 |
| 12.94% | 24851.92 | 24.26% | 248 | 10.63% | 18.61% | 125 | 14.37% | 34.96% | 117 | 0.618 |
| 12.95% | 24670.892 | 23.35% | 212 | 11.00% | 19.98% | 132 | 14.54% | 35.22% | 98 | 0.6 |
| 12.96% | 24636.887 | 23.18% | 210 | 11.07% | 20.25% | 103 | 18.47% | 40.31% | 86 | 0.5 |
| 12.97% | 24646.215 | 23.23% | 211 | 11.63% | 22.11% | 102 | 18.53% | 40.33% | 96 | 0.4 |
| 12.98% | 24886.519 | 24.43% | 238 | 11.82% | 22.67% | 110 | 18.68% | 40.42% | 93 | 0.382 |
| 13.00% | 24673.457 | 23.37% | 211 | 12.42% | 23.50% | 112 | 19.10% | 40.68% | 99 | 0.35 |
| 14.00% | 25718.758 | 28.59% | 188 | 12.89% | 24.08% | 111 | 19.27% | 40.79% | 102 | 0.3 |
| 15.00% | 26236.735 | 31.18% | 190 | 12.96% | 24.17% | 116 | 20.53% | 41.21% | 115 | 0.2 |
| 16.00% | 26551.224 | 32.76% | 210 | 13.04% | 24.26% | 107 | 25.19% | 41.86% | 103 | 0.1 |
| 17.00% | 26860.203 | 34.30% | 156 | 12.22% | 23.25% | 121 | 28.86% | 42.01% | 84 | 0.05 |
| 18.00% | 27100.375 | 35.50% | 199 | 12.16% | 23.18% | 119 | 29.03% | 42.02% | 164 | 0 |
| 19.00% | 27298.375 | 36.49% | 177 | 12.20% | 23.23% | 118 | | | | |
| 20.00% | 27485.217 | 37.43% | 127 | 13.18% | 24.43% | 105 | | | | |
| 30.00% | 28382.45 | 41.91% | 107 | 12.31% | 23.37% | 127 | | | | |
| 35.00% | 28404.038 | 42.02% | 96 | 21.54% | 28.59% | 89 | | | | |
| 40.00% | 28404.038 | 42.02% | 89 | infeasible | 31.18% | 64 | | | | |
| 45.00% | 28404.038 | 42.02% | 86 | | | | | | | |

## 3.2. Comparison and Analysis

- Allowance of Range

For Model 1, we are not able to find data when the volatility is below 12%. In other words, as a profit-oriented model, it is likely to expect more risk to compensate with the maximized return. Also, there is an upper limit for the return, which is 42.02%.

For Model 2, there exists a ceiling for risk where the return should be assigned below 30%. It is understandable that based on minimizing the risk a portfolio has limited ability to gain profits.

For Model 3, the range of return is limited from both above and below by the value of parameter weight (form 0 to 1).

- Risk-return comparison

  If we take a look at the blue shaded values on the above table. It is obvious to compare the first two models: if the required return is less than 24.08%, portfolio based on model 2 will bring less risk. On the other hand, if the required return from the investor is greater than 24.08%, model 1 will show less risk.

  When we bring Model 3 into account as well, with the same level of volatility, model 3 generates the largest return.
- Monotonicity

  In Model 1, there is a problem, as shown in light grey part of the table: as the level of volatility is increasing, the return is not strictly increasing among 12.95% and 13%. My guess is there might be a pre-assigned approximation method in online AMPL, which sometime lose the accuracy.
- Iterations

  For Model 1 and 2, in general, the higher the return goes, the less number of iterations required for optimal solution. Among all three models, profit-oriented model requires the most number of iterations.

# 4. Conclusions

According to the table above, the balanced model generates the highest return-risk ratio portfolio, but it is hard to show the desired percentage of return. Due to the indirect establishment of the objective function. With AMPL, it requires repetitive adjustment on parameter weight to find the requested level of return.

In general, the balanced model is good enough when compared with the previous two for generating high return portfolio with less volatility.

# 5. Code Directory

## 5.1. AMPL Code

**Model 1:**

```
############### SETS #######################################

set scenarios; # all possible scenarios (in terms of economy)

set sectors; # investment sectors

set stocks_sectors{i in sectors}; # possible stocks in sector i

set stocks := union{i in sectors} stocks_sectors[i];

# all stocks to be considered
```

```
############### PARAMETERS #################################

param prob{s in scenarios}; # probability of occuring of scenario s

param P; # principle amount for investment

param T; # no. of stages

param risk; # maximum risk to be allowed

##############

param initial_stock_price{j in stocks};

# price of stock j at the beginning

param stock_price{j in stocks, t in 1..T, s in scenarios};

# User given actual price of stock j at stage t in scenario s

param est_stock_price{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] * stock_price[j,t,s];

# estimated price of stock j at stage t

param est_stock_price_squared{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] *
(stock_price[j,t,s])^2;

# expectation of stock price squared

################

param std_dev{j in stocks, t in 1..T} := sqrt(est_stock_price_squared[j,t] -
(est_stock_price[j,t])^2);

# standard deviation of the price of stock j at stage t

param CV{j in stocks, t in 1..T} := std_dev[j,t]/est_stock_price[j,t];

# Coefficient of variation of stock price of stock j at stage t

##################

param D{j in stocks, t in 1..T, s in scenarios};

# Dividend of stock j at stage t in scenario s

#################

param prop_ub{i in sectors};

param prop_lb{i in sectors};

# upper and lower bounds of the proportion that we can invest in sector i

############### DECISION VARIABLES ##############################

var num_stocks{j in stocks, t in 0..T} integer >= 0;

# no. of stock j at the end of stage t
```

```
var cash_balance{t in 0..T} >=0;

# Cash left in hand at the end of stage t

############### OBJECTIVE FUNCTION ###################################

maximize final_return:

sum{j in stocks} num_stocks[j,T] * est_stock_price[j,T] + cash_balance[T];

############### CONSTRAINTS ###########################################

subject to initial_cash:

cash_balance[0] = P - sum{j in stocks}num_stocks[j,0] * initial_stock_price[j];


subject to cash{t in 1..T, s in scenarios} :

cash_balance[t] = cash_balance[t-1] + sum{j in stocks}D[j,t,s] * num_stocks[j,t-1]

+ sum{j in stocks}(num_stocks[j,t-1]-num_stocks[j,t]) * stock_price[j,t,s];

# Cash available in hand at the end of stage t

###############

subject to initial_proportion_ub{i in sectors}:

sum{j in stocks_sectors[i]} (initial_stock_price[j] * num_stocks[j,0]) <= prop_ub[i] * P;

subject to proportion_ub{i in sectors, t in 1..T, s in scenarios}:

sum{j in stocks_sectors[i]} (stock_price[j,t,s] * num_stocks[j,t])<= prop_ub[i] * (sum{j in
stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);

# upper limit of the investment in sector i

###############

subject to initial_proportion_lb{i in sectors}:

sum{j in stocks_sectors[i]} initial_stock_price[j] * num_stocks[j,0] >= prop_lb[i] * P;

subject to proportion_lb{i in sectors, t in 1..T, s in scenarios}:

sum{j in stocks_sectors[i]} stock_price[j,t,s] * num_stocks[j,t]

>= prop_lb[i] * (sum{j in stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);

# lower limit of the investment in sector i

################

subject to volatility{t in 1..T}:

(sum{j in stocks} (CV[j,t] * num_stocks[j,t]))/(sum{j in stocks} num_stocks[j,t]) <= risk;
```

# portfolio volatility (risk) is the weightage standard deviation

**Model 2:**

```
# Risk-Averse Model

#################### SETS ####################################

set scenarios; # all possible scenarios (in terms of economy)

set sectors; # investment sectors

set stocks_sectors{i in sectors}; # possible stocks in sector i

set stocks := union{i in sectors} stocks_sectors[i];

# all stocks to be considered

############### PARAMETERS ##################################

param prob{s in scenarios}; # probability of occuring of scenario s

param P; # principle amount for investment

param T; # no. of stages

param preturn; #the minimum return ratio

##############

param initial_stock_price{j in stocks};

# price of stock j at the beginning

param stock_price{j in stocks, t in 1..T, s in scenarios};

# User given actual price of stock j at stage t in scenario s

param est_stock_price{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] * stock_price[j,t,s];

# estimated price of stock j at stage t

param est_stock_price_squared{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] *

(stock_price[j,t,s])^2;

# expectation of stock price squared

################

param std_dev{j in stocks, t in 1..T} := sqrt(est_stock_price_squared[j,t] -
(est_stock_price[j,t])^2);

# standard deviation of the price of stock j at stage t

param CV{j in stocks, t in 1..T} := std_dev[j,t]/est_stock_price[j,t];

# Coefficient of variation of stock price of stock j at stage t
```

```
##################
param D{j in stocks, t in 1..T, s in scenarios};
# Dividend of stock j at stage t in scenario s
#################
param prop_ub{i in sectors};
param prop_lb{i in sectors};
# upper and lower bounds of the proportion that we can invest in sector i
############### DECISION VARIABLES ##############################
var num_stocks{j in stocks, t in 0..T} integer >= 0;
# no. of stock j at the end of stage t
var cash_balance{t in 0..T} >=0;
# Cash left in hand at the end of stage t
############### OBJECTIVE FUNCTION ##################################
minimize risk:
(sum{j in stocks,t in 1..T} (CV[j,t]*num_stocks[j,t]))/(sum{j in stocks, t in 1..T} num_stocks[j,t]);
############### CONSTRAINTS ###########################################
subject to initial_cash:
cash_balance[0] = P - sum{j in stocks}num_stocks[j,0] * initial_stock_price[j];
subject to cash{t in 1..T, s in scenarios} :
cash_balance[t] = cash_balance[t-1] + sum{j in stocks}D[j,t,s] * num_stocks[j,t-1]
+ sum{j in stocks}(num_stocks[j,t-1]-num_stocks[j,t]) * stock_price[j,t,s];
# Cash available in hand at the end of stage t
###############
subject to initial_proportion_ub{i in sectors}:
sum{j in stocks_sectors[i]} (initial_stock_price[j] * num_stocks[j,0]) <= prop_ub[i] * P;
subject to proportion_ub{i in sectors, t in 1..T, s in scenarios}:
sum{j in stocks_sectors[i]} (stock_price[j,t,s] * num_stocks[j,t])
<= prop_ub[i] * (sum{j in stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);
# upper limit of the investment in sector i
###############
```

subject to initial_proportion_lb{i in sectors}:

sum{j in stocks_sectors[i]} initial_stock_price[j] * num_stocks[j,0] >= prop_lb[i] * P;

subject to proportion_lb{i in sectors, t in 1..T, s in scenarios}:

sum{j in stocks_sectors[i]} stock_price[j,t,s] * num_stocks[j,t]

>= prop_lb[i] * (sum{j in stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);

# lower limit of the investment in sector i

################

subject to mini_return_stage{t in 1..T}:

sum{j in stocks} num_stocks[j,t]*est_stock_price[j,t]+cash_balance[t]>= P* (1+preturn);

**Model 3:**

#Weighted Objective Function on Risk&Return

#################### SETS ####################################

set scenarios; # all possible scenarios (in terms of economy)

set sectors; # investment sectors

set stocks_sectors{i in sectors}; # possible stocks in sector i

set stocks := union{i in sectors} stocks_sectors[i];

# all stocks to be considered

############### PARAMETERS ##################################

param prob{s in scenarios}; # probability of occuring of scenario s

param P; # principle amount for investment

param T; # no. of stages

param wgt; #the weight of risk in objective function

##############

param initial_stock_price{j in stocks};

# price of stock j at the beginning

param stock_price{j in stocks, t in 1..T, s in scenarios};

# User given actual price of stock j at stage t in scenario s

param est_stock_price{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] * stock_price[j,t,s];

# estimated price of stock j at stage t

```
param est_stock_price_squared{j in stocks, t in 1..T} := sum{s in scenarios} prob[s] *
(stock_price[j,t,s])^2;

# expectation of stock price squared

################

param std_dev{j in stocks, t in 1..T} := sqrt(est_stock_price_squared[j,t] -
(est_stock_price[j,t])^2);

# standard deviation of the price of stock j at stage t

param CV{j in stocks, t in 1..T} := std_dev[j,t]/est_stock_price[j,t];

# Coefficient of variation of stock price of stock j at stage t

##################

param D{j in stocks, t in 1..T, s in scenarios};

# Dividend of stock j at stage t in scenario s

#################

param prop_ub{i in sectors};

param prop_lb{i in sectors};

# upper and lower bounds of the proportion that we can invest in sector i

############### DECISION VARIABLES ##############################

var num_stocks{j in stocks, t in 0..T} integer >= 0;

# no. of stock j at the end of stage t

var cash_balance{t in 0..T} >=0;

# Cash left in hand at the end of stage t

############### OBJECTIVE FUNCTION ####################################

maximize risk_return:

-wgt*((sum{j in stocks,t in 1..T} (CV[j,t]*num_stocks[j,t]))/(sum{j in stocks, t in 1..T}
num_stocks[j,t]))+(1-wgt)*(sum{j in stocks}
num_stocks[j,T]*est_stock_price[j,T]+cash_balance[T])/P;

############### CONSTRAINTS ############################################

subject to initial_cash:

cash_balance[0] = P - sum{j in stocks}num_stocks[j,0] * initial_stock_price[j];


subject to cash{t in 1..T, s in scenarios} :

cash_balance[t] = cash_balance[t-1] + sum{j in stocks}D[j,t,s] * num_stocks[j,t-1]
```

```
+ sum{j in stocks}(num_stocks[j,t-1]-num_stocks[j,t]) * stock_price[j,t,s];
# Cash available in hand at the end of stage t
###############
subject to initial_proportion_ub{i in sectors}:
sum{j in stocks_sectors[i]} (initial_stock_price[j] * num_stocks[j,0]) <= prop_ub[i] * P;
subject to proportion_ub{i in sectors, t in 1..T, s in scenarios}:
sum{j in stocks_sectors[i]} (stock_price[j,t,s] * num_stocks[j,t])
<= prop_ub[i] * (sum{j in stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);
# upper limit of the investment in sector i
###############
subject to initial_proportion_lb{i in sectors}:
sum{j in stocks_sectors[i]} initial_stock_price[j] * num_stocks[j,0] >= prop_lb[i] * P;
subject to proportion_lb{i in sectors, t in 1..T, s in scenarios}:
sum{j in stocks_sectors[i]} stock_price[j,t,s] * num_stocks[j,t]
>= prop_lb[i] * (sum{j in stocks}stock_price[j,t,s] * num_stocks[j,t] + cash_balance[t]);
# lower limit of the investment in sector i
```

## 5.2. Data Set

```
##################  Data Set ##############################
set scenarios := Increasing Decreasing Stationary;
set sectors := Bank Real_Estate AGR RES INT;

set stocks_sectors[Bank] := Minsheng Huaxia Pufa Xingye BOC Jianshe;
set stocks_sectors[Real_Estate] := Lujia Wanke Waigao Zhangjiang;
set stocks_sectors[AGR] := Beidahuang Longping Zhongmu;
set stocks_sectors[RES] := Jinshan Jiaozuo Hangtian Yunnan;
set stocks_sectors[INT] := Beixinyuan Landun Langchao;
################# PARAMETERS #############################
param prob :=
[Increasing] 0.2
```

```
[Decreasing] 0.25
[Stationary] 0.55;
param P := 20000;
param T := 4;
param risk := 0.01;
#param wgt := 0.382;
#param preturn := 0.19;

########################
param prop_ub :=
[Bank] 0.4
[Real_Estate] 0.25
[AGR] 0.2
[RES] 0.2
[INT] 0.2;
param prop_lb :=
[Bank] 0.1
[Real_Estate] 0.05
[AGR] 0.02
[RES] 0.04
[INT] 0.05;

###########################
param initial_stock_price :=
Minsheng 8.49

Huaxia 10.85

Pufa 12.58

Xingye 12.38

BOC 3.41

Jianshe 4.94

Lujia 31.49

Wanke 11.20

Waigao 32.60

Zhangjiang 8.42

Beidahuang 13.18

Longping  20.10

Zhongmu 12.56

Jinshan 7.69
```

Jiaozuo 5.60

Hangtian 15.37

Yunnan 8.56

Beixinyuan 23.40

Landun 16.00

Langchao 17.26 ;

###########################
param stock_price :=
[*, *, Increasing] :
1 2 3 4 :=
Minsheng 9.05 9.86 10.15 11.30

Huaxia 11.58 12.15 13.04 13.94

Pufa 13.27 14.89 16.05 17.91

Xingye 13.29 13.56 14.15 14.07

BOC 3.53 3.98 3.87 3.95

Jianshe 5.18 6.81 7.46 7.95

Lujia 35.97 38.85 39.62 41.57

Wanke 11.69 13.05 15.67 16.73

Waigao 37.08 39.59 41.68 43.41

Zhangjiang 12.00 13.19 15.06 17.71

Beidahuang 18.13 19.65 20.29 22.34

Longping 26.53 27.56 28.18 29.96

Zhongmu 15.31 16.24 17.65 19.06

Jinshan 11.50 13.15 14.87 15.31

Jiaozuo 11.16 12.05 13.86 14.56

Hangtian 34.98 36.89 39.54 43.15

Yunnan 14.68 16.89 19.12 23.56

Beixinyuan 54.50 58.50 58.35 59.14

Landun 30.98 31.59 33.56 38.97

Langchao 55.45 59.54 61.12 65.02

[*, *, Decreasing] :
1 2 3 4 :=
Minsheng 7.68 6.89 6.78 5.96

Huaxia 9.76 8.89 7.96 7.13

Pufa 12.05 11.53 10.96 10.14

Xingye 11.46 10.86 10.16 9.59

BOC 3.26 3.15 3.09 3.04

Jianshe 4.81 4.59 4.36 4.02

Lujia 30.60 29.75 28.54 26.89

Wanke 11.05 10.84 10.15 9.45

Waigao 28.07 27.62 25.81 24.15

Zhangjiang 5.73 5.14 4.89 4.13

Beidahuang 8.46 7.62 6.93 5.75

Longping 18.05 18.01 17.26 15.40

Zhongmu 10.52 9.89 9.56 7.99

Jinshan 3.95 3.59 3.17 2.98

Jiaozuo 4.21 4.05 3.57 3.19

Hangtian 11.05 10.89 10.16 9.89

Yunnan 6.89 5.98 5.41 4.14

Beixinyuan 16.74 15.45 14.41 13.21

Landun 12.12 11.54 10.37 9.87

Langchao 10.48 10.04 9.84 9.56

[*, *, Stationary] :
1 2 3 4 :=
Minsheng 8.5 8.5 8.7 8.4

Huaxia 10.65 10.69 10.68 10.64

Pufa 12.40 12.43 12.35 12.35

Xingye 12.50 12.48 12.53 12.60

BOC 3.40 3.42 3.40 3.35

Jianshe 5.00 4.97 5.05 5.01

Lujia 31.50 30.98 31.42 31.50

Wanke 11.20 11.18 11.23 11.18

Waigao 34.01 34.13 33.98 34.07

Zhangjiang 7.53 7.69 7.49 7.52

Beidahuang 13.89 13.84 13.92 13.87

Longping 25.31 25.15 25.24 25.16

Zhongmu 11.86 11.84 11.82 11.87

Jinshan 6.74 6.76 6.81 6.71

Jiaozuo 5.89 5.75 5.81 5.91

Hangtian 28.40 28.31 28.17 28.53

Yunnan 7.98 8.01 7.84 7.95

Beixinyuan 25.14 25.01 24.95 24.96

Landun 19.75 19.75 19.14 20.01

Langchao 26.81 26.45 27.01 27.51;

##############################

param D :=
[*, *, Increasing] :
1 2 3 4 :=
Minsheng 0.05 0.04 0.04 0.05

Huaxia 0.12 0.10 0.05 0.07

Pufa 0.05 0.06 0.08 0.10

Xingye 0.05 0.05 0.05 0.05

BOC 0.10 0.10 0.15 0.05

Jianshe 0.05 0.04 0.05 0.05

Lujia 0.12 0.05 0.05 0.10

Wanke 0.2 0.25 0.21 0.2

Waigao 0.05 0.07 0.15 0.16

Zhangjiang 0.20 0.20 0.20 0.20

Beidahuang 0.50 0.50 0.45 0.45

Longping 0.12 0.15 0.10 0.09

Zhongmu 0.15 0.15 0.15 0.15

Jinshan 0.05 0.05 0.05 0.05

Jiaozuo 0.10 0.10 0.10 0.10

Hangtian 0.05 0.07 0.05 0.05

Yunnan 0.20 0.20 0.15 0.25

Beixinyuan 0.50 0.25 0.25 0.15

Landun 0.05 0.10 0.10 0.15

Langchao 0.05 0.05 0.05 0.05

[*, *, Decreasing] :
1 2 3 4 :=
Minsheng 0.05 0.04 0.04 0.0

Huaxia 0.12 0.7 0.05 0.03

Pufa 0.05 0.05 0.03 0

Xingye 0.05 0.05 0.05 0.05

BOC 0.10 0.10 0.07 0.05

Jianshe 0.05 0.04 0.05 0.05

Lujia 0.12 0.05 0.05 0.10

Wanke 0.2 0.25 0.21 0.2

Waigao 0.05 0.07 0.05 0.06

Zhangjiang 0.20 0.20 0 0.15

Beidahuang 0.50 0.50 0.35 0.15

Longping 0.12 0.15 0.10 0.09

Zhongmu 0.15 0.15 0.15 0.15

Jinshan 0.05 0 0.05 0.05

Jiaozuo 0.10 0.10 0.10 0.10

Hangtian 0.05 0.05 0.05 0.05

Yunnan 0.20 0.20 0.15 0.05

Beixinyuan 0.50 0.25 0 0.15

Landun 0.05 0.10 0.10 0.15

Langchao 0.05 0.05 0.05 0.05

[*, *, Stationary] :
1 2 3 4 :=
Minsheng 0.04 0.04 0.04 0.04

Huaxia 0.05 0.07 0.05 0.05

Pufa 0.05 0.05 0.05 0.05

Xingye 0.05 0.05 0.05 0.05

BOC 0.10 0.10 0.10 0.10

Jianshe 0.05 0.05 0.05 0.05

Lujia 0.07 0.07 0.07 0.07

Wanke 0.2 0.2 0.2 0.2

Waigao 0.05 0.05 0.05 0.05

Zhangjiang 0.20 0.20 0 0.15

Beidahuang 0.50 0.50 0.35 0.15

Longping 0.12 0.11 0.10 0.09

Zhongmu 0.15 0.15 0.15 0.15

Jinshan 0.05 0.04 0.05 0.05

Jiaozuo 0.10 0.10 0.10 0.10

Hangtian 0.05 0.05 0.05 0.05

Yunnan 0.20 0.20 0.15 0.10

Beixinyuan 0.20 0.25 0 0.15

Landun 0.05 0.10 0.10 0.10

Langchao 0.05 0.05 0.05 0.05;

# References

[1] L. Hiller. Introduction to Operations Research. 9th Edition: McGraw Hill.

[2] R. Fourer, D.M. Gay, B.W.Kernighan. AMPL: A Modeling Language for Mathematical Programming. 2nd Edition: Thomson Books/Cole.

[3] Gopal Koirala. An AMPL Model for Portfolio Management.

[4] David Harper. The Uses and Limits of Volatility. Investopedia. http://www.investopedia.com/articles/04/021804.asp