Martin Pavlík
e-mail: mpavlik10@gmail.com

Grzegorz Michalski
Wrocław University of Economics
e-mail: grzegorz.michalski@gmail.com

Martin Lukáčik
University of Economics, Bratislava
e-mail: martin.lukacic@euba.sk

PORTFOLIO SELECTION: METHOD OF THE STEP BY STEP ASSIGNED WEIGHTS

WYBÓR PORTFELA: METODA WAG DOBIERANYCH KROK PO KROKU

Summary: The authors conceived a new simple method for creating the approximation of the *border of investment opportunities*. The method enumerates all the possibilities of assigning weights to the investment portfolio. It does not enable short sales. The software which the authors coded is written in VBA and also enables active management. The method is simple, accurate but demanding. The authors also created a simple methodology for testing the quality of the approximation of the *border of investment opportunities*.

Keywords: modern portfolio theory, VBA in Excel, enumeration, portfolio choice, VaR, Value at Risk.

Streszczenie: Autorzy zaprojektowali i wykonali program w Visual Basic dla Excel 2010. Oprogramowanie to było testowane kontrolnie na Excel 2016. Z założenia program był utworzony z myślą o pomocy małym inwestorom, którzy chcą stworzyć własny portfel inwestycyjny. Analogicznie może być pomocny w zarządzaniu zapasami i należnościami przez analizę portfela należności i portfela zapasów w małych i średnich przedsiębiorstwach o pełnym cyklu operacyjnym. Oprogramowanie działa dość powoli, lecz w efekcie daje unikalne rezultaty, których nie oferuje żadne inne oprogramowanie. Jest proste i nadaje się zarówno do naukowych, jak i do dydaktycznych celów w obszarze zastosowań komercyjnych i akademickich, z założenia może również być użyte na potrzeby studiów przypadków z zakresu modelowania finansowego. Pozwala na prezentację powodów, dla których podmiot dokonujący analizy używa ekonometrycznych i numerycznych metod w celu rozwiązania zagadnień i problemów o charakterze nieliniowym. Autorzy wykorzystali podstawy teorii finansów oraz teorii doboru i konstrukcji portfeli inwestycyjnych wraz z podstawami teorii zarządzania portfelami niestandardowymi. Poprzedni artykuł z tej serii zawierał podejście wykorzystujące biblioteki Solvera oraz algebrę macierzy. W niniejszym artykule dokonano

obliczeń wykorzystujących podejście *brute force* (przeszukiwanie wszystkich modeli). Pozostawiono kody źródłowe oprogramowania otwarte, w związku z tym analitycy, naukowcy i studenci mogą bez ograniczeń weryfikować i ewentualnie wykorzystać kodowanie. Artykuł zawiera także przykład ilustracyjnego portfela. Portfel jest sztuczny i został stworzony z wykorzystaniem oprogramowania środowiska R. Artykuł publikuje też źródłowy kod niniejszego sztucznego portfela.

Słowa kluczowe: wybór portfela, metoda wag dobieranych krok po kroku, VBA, Excel, Visual Basic dla aplikacji, ryzyko, zwrot, portfel inwestycyjny, wartość narażona na ryzyko, odchylenie standardowe.

1. Introduction

Portfolio investments in the financial liquidity of a firm are made through investing money in current assets [Michalski 2012]. The result are the financial consequences of managerial decisions on the current assets investments level. There are two kinds of current assets investment influence: positive, linked with cash revenues stimulation and the decreasing level of risk, and negative, linked with increasing cash expenses [Keynes 1936; Michalski 2008]. Current assets maintained in the firm are linked with portfolio effects. Financial liquidity decisions could be considered from three perspectives: the intrinsic value of financial liquidity compared with the market value of liquidity point of view, the risk sensitivity on current assets investments' consequences point of view, and the behavioral finance approach to financial liquidity investments point of view. The portfolio choice problems are concerned with all of them, but here we concentrate on the point of view of risk sensitivity on current assets investments consequences. Financial liquidity investments in enterprises are made for safety purposes. Without them there is no possibility to safely manage the operational cycle of the enterprise. The operational cycle is realized as the main process in enterprise value creation, during which there is added value through collecting raw materials, and next thanks to using the enterprise's fixed assets, with the energy and intellectual capital of the enterprise, raw materials change into final production. The finished goods are offered to the clients of an enterprise, and the way they are offered is also an opportunity to create additional value. Current assets are an important part of all steps in the operational cycle. The lack of current assets at any one time of the operational cycle realization could stop value creation and is linked with possible long term consequences.

The following article is a result of a Slovak-Polish cooperation. We have created a set of three programs in MS Excel which calculate the approximation of the border of the investment opportunities. The applications are continuously being developed. All the programs are written in VBA for Excel.

The aim of the article is to introduce the first program in which we tried to develop a new method for creating the approximation of the border of the investment op-

portunities. The method enumerates all the possibilities of the assigned portfolio weights with the given step. The method is simple, accurate, but demanding.

Our achievement is that we have created a simple method which can draw the approximation of the *border of the investment opportunities.* Our method is suitable for those who want to create portfolios which no other software offers. It contains an *Active Management* option, so the user can back or penalize particular shares.

2. Expected return and risk measures of the shares

The Markowitz diversification is found to be the start of the modern theory of an investment portfolio. Markowitz suggested measures for measuring the risk of a portfolio and its expected return. Risk was measured as a standard deviation and expected return as an arithmetical mean.

Expected return of a single share: $E(R) = \sum_{k=1}^{n} p_k R_k$,

where R_k is a random variable, P_k is the probability of observing R_k.

Risk of a single share: $\sigma^2(R) = E\left[\left(R - E(R)\right)^2\right] = \sum_{k=1}^{n} p_k \left(R_k - E(R)\right)^2$,

$\sigma(R) = \sqrt{\sigma^2(R)}$.

Arithmetical mean and standard deviation are the measures which Markowitz started with. Other measures were developed afterwards such as Lower Semi-Variance.

Lower semi-variance for risk: $\sigma_- = E\left[\left(R - E(R)\right)_-^2\right]$ is a measure of the dispersion of all observations that fall below the mean. Semi-variance is an average of the squared deviations of values that are less than the mean, which means that the formula for semivariance includes just negative deviations. Positive deviations are set to 0. Lower semi-variance is based on the idea that the investor does not pay attention to the positive deviations from the mean. The investor does not penalise them, the investor needs them.

Another risk measure is Value at Risk (VaR). VaR is 0,01;0,05 or 0,1 percentile of the loss calculated from the random distribution of a share or investment portfolio. The formula for VaR_q is:

$VaR_\alpha = V_0(z_\alpha \sigma(R) + E(R))$ where z_α is α quantile of the normal distribution and V_0 is initial investment.

We suppose that both shares and investment portfolio have normal distribution. Our software calculates all three levels of the VaR and both the variance and semi-variance for the individual shares. The user's choice is which risk calculation option for the investment portfolio will be chosen. Setting initial investment value is not necessary. In that case V_0 is 1 EUR and VaR_α is a loss from 1 EUR.

3. Expected return and risk measures of the investment portfolio

The expected return of an investment portfolio is calculated as:

$$E\left(R_p\right) = E\left(\sum_{k=1}^{n} w_k . R_k\right) = \sum_{k=1}^{n} w_k E\left(R_k\right) \qquad \sum_{k=1}^{n} w_k = 1$$

Where w_k are the weights of individual shares. We do not allow short sales in our software, so w_k is not negative. The risk of an investment portfolio in case the risk measure was chosen as the standard deviation is calculated as:

$$\sigma^2\left(R_p\right) = \sum_{k=1}^{n} w_k^2 . \sigma_k^2 + 2\sum_{k=1}^{n}\sum_{l>k} w_k w_l \sigma_{kl}$$

or

$$\sigma_p^2 = \mathbf{w}^T . \mathbf{C} . \mathbf{w}$$

Where $\mathbf{C}$ is a variance-covariance matrix. We have experimented with both formulas for standard deviation in our software and we have discovered that matrix calculations are much slower than "brute force" risk calculation, that is why we do not use matrix calculation for the risk.

Lower semi-variance risk calculation is very similar to variance. The difference is that we have not included positive deviation of:

$$(R_{ik} - \overline{R_k})^2, \text{ where } (R_{ik} - \overline{R_k}) \text{ is greater than zero.}$$

$$(R_{ik} - \overline{R_k})(R_{il} - \overline{R_l}), \text{ where both } (R_{ik} - \overline{R_k}) \text{ and } (R_{il} - \overline{R_l}) \text{ are greater than zero.}$$

VaR calculation is different:

$$VaR_\alpha = V_0\left(z_\alpha \sigma(R_p) + E(R_p)\right) \text{ where } z_\alpha \text{ is } \alpha\text{-quantile of the normal distribu-}$$
tion and V_0 is initial investment.

Our software calculates VaR_α for $\alpha = 0.01, 0.05, 0.1$. VaR is a negative number.

Creating the border of the investment opportunities is a nonlinear programming problem which is different for variance and for VaR. The difference isobvious:

$$\min \sigma^2(R_p) \qquad (1) \qquad \max \quad VaR_\alpha \qquad (2)$$
$$E(R_p) = E_p \qquad\qquad\qquad E(R_p) = E_p$$
$$\sum w_i = 1 \qquad\qquad\qquad \sum w_i = 1$$
$$w_i \geq 0 \qquad\qquad\qquad\quad w_i \geq 0$$

4. Expected return and its distribution

One of the assumptions of the modern portfolio theory is that the distributions of the returns are jointly normally distributed random variables. We test every share and every portfolio with the Jarque-Berra test, which tests whether the tested distribution is normal or not, although normal distribution for portfolios is not a fundamental assumption of the modern theory of portfolio. The Jarque-Berra test is a test which is based on testing skewness and curtosis. Normal distribution has skewness equal to zero and kurtosis is equal to 3. That is why the JB test proves that distribution is normal when Jarque-Bera statistics is close to zero. Jarque Bera statistics has χ^2 distribution with two degrees of freedom. $q_{0.95}^{\chi^2(2)} \approx 6$. If Jarque-Bera statistics is smaller than 6, we accept H$_0$that the observed random variable has an independent and identically distributed normal distribution.

$$JB = \frac{T}{6}\left(\widehat{skew}^2 + \frac{(\widehat{kurt} - 3)^2}{4}\right)$$

$$\widehat{skew} = \frac{1}{T-1}\sum_{t=1}^{T}\frac{(x_t - \bar{x})^3}{s_x^3}$$

$$\widehat{kurt} = \frac{1}{T-1}\sum_{t=1}^{T}\frac{(x_t - \bar{x})^4}{s_x^4}$$

Our software fills every cell with Jarque-Bera statistics greater than 5.99 with the color red. The software is not solving non-normal distribution problem neither for portfolios nor for individual shares.

5. Method of the step by step assigned weights

Tasks (1) and (2) can be solved in Excel using Solver. Solver can solve nonlinear programming tasks. We tried to develop a new method. We were inspired with the method of the explicit enumeration in the bivalent programming. The bivalent programming tasks are tasks with discrete variables, whereas the portfolio selection tasks are tasks with continuous variables. We had to change continuous variables to discrete variables. We did this by rounding the expected return to the specific digits. This is what the user sets with the *Scale* in our software. *Scale* sets the number of digits after the comma. Another tool which makes variables discrete is step. The user chooses the step. Our method's name is *the method of the step by step assigned weights,* because the user chooses the step in which the weights will be assigned to the particular shares. The following table (Table 1) shows the weights assigned to three shares with the step of 25 percent. The method does not allow short sales since it would increase computation demand rapidly.

Table 1. Weights assigned for three shares and step 25

S1	100	75	50	25	0	75	50	25	0
S2	0	25	50	75	100	0	25	50	75
S3	0	0	0	0	0	25	25	25	25

S1	50	25	0	25	0	0
S2	0	25	50	0	25	0
S3	50	50	50	75	75	100

Source: authors' calculations.

The procedure which assigns the weights correctly is the key procedure of the whole application. An assignment has to be done correctly which means:
- the sum of the weights has to be 100%,
- all the possibilities of assigning weights have to be fulfilled,
- the assignment has to be made quickly,
- the assignment has to be done without redundancy.

Step 25 leaves many solutions unexplored. The user can decrease the step for exploring more solutions. The only disadvantage of the method is that if the user decreases the step, the number of explored solutions will increase rapidly. This is the only disadvantage of the method. Tables 2 and 3 show how rapidly this happens.

Table 2. The increase of computation demand according to the step and shares count

	Step						
Shares	50	33.3	25	20	10	5	3.33
2	3	4	5	6	11	21	31
3	6	10	15	21	66	231	496
4	10	20	35	56	286	1771	5456
5	15	35	70	126	1001	10626	46376
6	21	56	126	252	3003	53130	324632
7	28	84	210	462	8008	230230	1947792
8	36	120	330	792	19448	888030	10295472
9	45	165	495	1287	43758	3108105	48903492
10	55	220	715	2002	92378	10015005	211915132
11	66	286	1001	3003	184756	30045015	847660528
12	78	364	1365	4368	352716	84672315	3159461968
13	91	455	1820	6188	646646	225792840	11058116888
14	105	560	2380	8568	1144066	573166440	36576848168
15	120	680	3060	11628	1961256	1391975640	114955808528

Source: authors' calculation.

Table 3. The increase of computation demand according to the step and shares count

Shares	Step				
	2.5	2	1	0.5	0.2
2	41	51	101	201	501
3	861	1326	5151	20301	125751
4	12341	23426	176851	1373701	21084251
5	135751	316251	4598126	700587751	2656615626
6	1221759	3478761	96560646	2872408791	268318178226
7	9366819	32468436	1705904746	98619368491	x

Source: authors' calculation.

The source code of the procedure is:

```
Sub weights ()
Dim counter As Single
Dim i As Integer
counter = 0
i = 1
w(1) = 1000
Call calculation
Do Until w(pcp) = 1000
w(i) = 0
i = i + 1
w(i) = w(i) + step
counter = counter + step
If counter = 1000 Then
counter = counter - w(i)
        Else
w(1) = 1000 - counter
         i = 1
        End If
Call calculation
Loop
End Sub
```
w(i) is a field of weights

Variable *counter* is a variable which pays attention to the condition $\sum w_i = 1$
Variable *pcp* is a shares count

The procedure looks very simple. We have spent many hours to achieve this. In fact, we have spent most of the time debugging the procedure *weights*. The procedure allows the user to set weights for such steps as: 1,2,5,10,25.These are the steps whose sum is exactly 100 (or 1000, if we allow also step 2.5). Our ambition was higher. We have also developed a procedure which enables to use such steps as: 3.33, 5.55, 6.25.

We have modified the procedure so we can also use steps with a sum which is not equal exactly to 100. That procedure is shown here:

```
Sub weights()
Dim counter As Double
Dim i As Integer
poc = 0
i = 1
w(1) = 10000
Call calculation
Do Until w(pcp) = 10000
w(i) = 0
i = i + 1
w(i) = w(i) + step
counter = counter + step
If poc = sucet Then
poc = poc - w(i)
        w(i) = w(i) + rest
Else
w(1) = 10000 - counter
            i = 1
        End If
Call calculation
Loop
End Sub
```

The difference is that we have added the *rest* variable. This is the variable which contains the remaining number after the summation to 10,000. Another variable which was added is the *sucet* variable. It contains the real sum which is lower than 10,000. (For example for step 3.33 (333) there is *sucet* = 9990 and *rest* = 10). We used 10,000 as a summation target, because we also used steps with two digits after the point. Our software contains these steps: 0.5, 1, 1.51, 2, 2.5, 3.33, 3.7, 4, 4.76, 5, 5.55, 6.25, 7.14, 8.33, 9.09, 10, 11.1, 12.5, 14.28, 16.66, 20, 25, 33.3, 50.

Another key procedure is the procedure *calculation*. Procedure *weights* calls the procedure *calculation*. Procedure weights have two parts. The first part calculates the expected return and risk. The second part is a gatekeeper part. The gatekeeper part decides whether a new solution will be kept or thrown away. The user sets the *Scale* in the MENU. The *Scale* splits the expected return axe into "boxes" and turns the continuous expected value variable into the discrete variable. We have achieved it with rounding. According to calculated expected return, the procedure *calculation* finds the appropriate "box" for the calculated risk and decides whether the calculated risk is better than the stored one. This is what the gatekeeper's part does. The following procedure calculates variance as a risk measure:

```
Sub calculation()
vynportpom = 0
rizportpom = 0
For a = 1 To pcp
vynportpom = vynportpom + (w(a) / 1000) * return(a)
rizportpom = rizportpom + (w(a) / 1000) ^ 2 * varko(a, a)
For b = a + 1 To pcp
rizportpom = rizportpom + 2 * (w(a) / 1000) * (w(b) / 1000) * varko(a, b)
Next b
Next a
vynportpom = Int(vynportpom * des + 0.5)
If rizportpom >rizport(vynportpom - ui + 1) Then Exit Sub
rizport(vynportpom - ui + 1) = rizportpom
For b = 1 To pcp
vahyport(vynportpom - ui + 1, b) = w(b)
Next b
```

Varko(i,j) is a field of the variance-covariance matrix
Return(i) is a field of the individual shares expected return
Des is a constant which sets the digits count after the point in expected return value
Variable *vynportport* contains calculated expected return.
Variable *rizportpom* contains calculated risk.
Field *vahyport* stores weights of the stored portfolios

Following procedure *calculation* contains the calculation of the VaR as a risk measure.

```
Sub calculation()
vynportpom = 0
rizportpom = 0
For a = 1 To pcp
vynportpom = vynportpom + (w(a) / 10000) * vynosy(a)
rizportpom = rizportpom + (w(a) / 10000) ^ 2 * varko(a, a)
For b = a + 1 To pcp
rizportpom = rizportpom + 2 * (w(a) / 10000) * (w(b) / 10000) * varko(a, b)
Next b
Next a
rizportpom = rizportpom ^ 0.5
rizportpom = qq * rizportpom + vynportpom
vynportpom = Int(vynportpom * des + 0.5)
If rizportpom <rizport(vynportpom - ui + 1) Then Exit Sub
rizport(vynportpom - ui + 1) = rizportpom
For b = 1 To pcp
vahyport(vynportpom - ui + 1, b) = w(b)
Next b
End Sub
```
Variable *qq*is a quantile of the normal distribution

Thus, procedure *weights* assigns appropriate weights to the shares and the procedure *calculation* calculates the expected return and risk. These two procedures are the core procedures of the whole application. After the calculation, the software draws a graph. The points which are drawn in the figure are written in the *Results* sheet. Figure 1 shows the approximation of the border of the investment opportunities for ten shares and step 25.
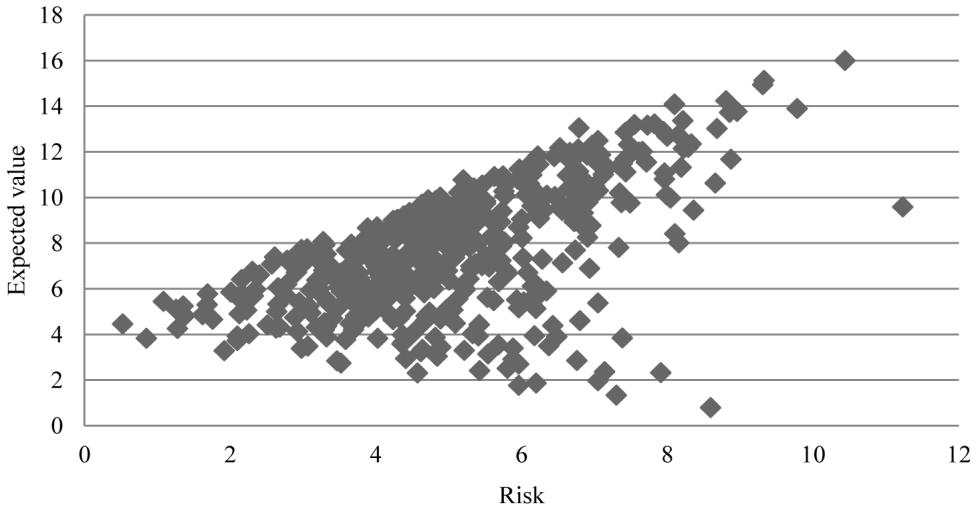
Figure 1. Ten shares and step 25

Source: authors' calculation.

As can be seen, the figure does not look like the border of the investment opportunities. This is because we set the step too high. We have to decrease the step. Figure 2 shows the same border of investment opportunities, but we decreased step to 5.

Figure 2 shows many more points which look like the approximation of *the border of the investment opportunities.*It contains many redundant points, which are drawn inside *the border of the investment opportunities.* We have solved this problem with implementing *Filter*. The user can decide whether or not the *Filter* will be used. The filter has two levels. The first level of the filter is weak, the second much stronger. Figure 3 shows the same border of the investment opportunities as Figure 2, but we have applied the first level of the filter. Figure 4 shows the same border of the investment opportunities as Figure3 and Figure 2, but we have applied the second level of the filter. We used standard deviation as a risk measure from Figure 1 to Figure 4, but the filter works also for VaR risk measure and semi-variance risk measure.
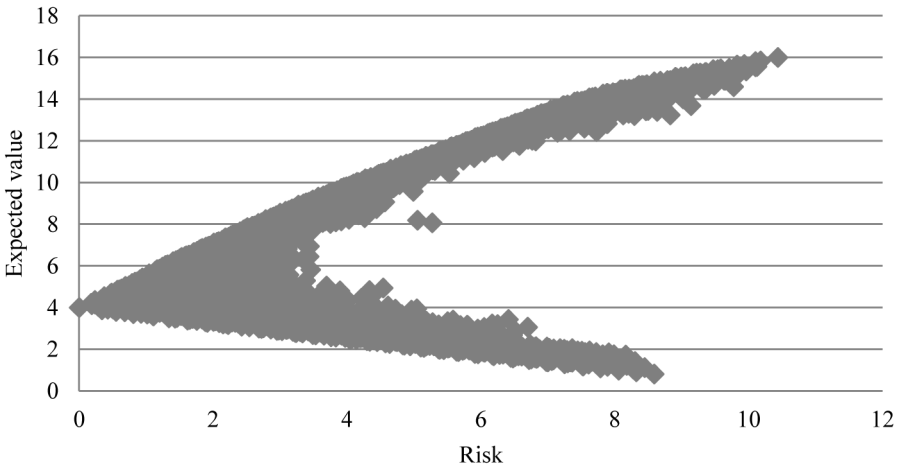
Figure 2. Ten shares and step 5

Source: authors' calculation.
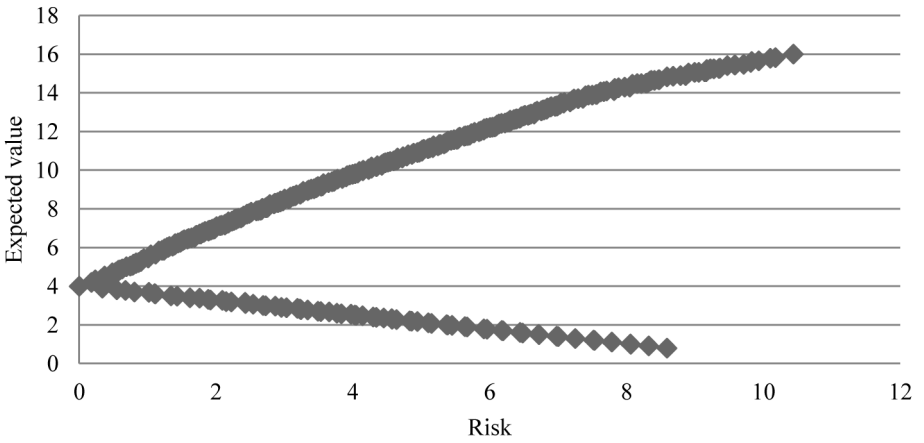
Figure 3. Application of the 1st level of the filter for ten shares and step 5

Source: authors' calculation.

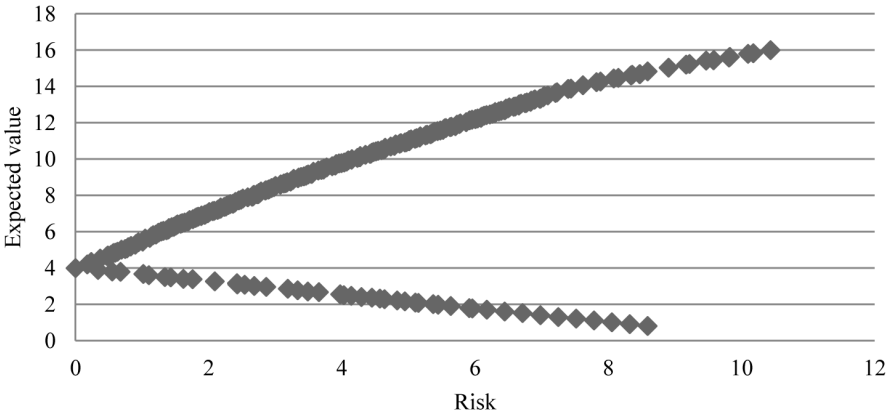Graph No. 4 Application of the 2nd level filter for ten shares and step 5.

Figure 4. Application of the 2nd levelof the filter for ten shares and step 5

Source: authors' calculation.

5.1. Algorithmisation of the filter

The first filter is very simple. We have to look at the border of the investment oppor-tunities as it is a convex function. This can be done rotating the figure by about 90 degrees. We will swap the axis. Figure 5 shows how the points of the convex func-tion should be located. Our method does not produce "decent" points. Figure 6 shows the function with the point which causes that the function is not convex. That point is a local extreme.
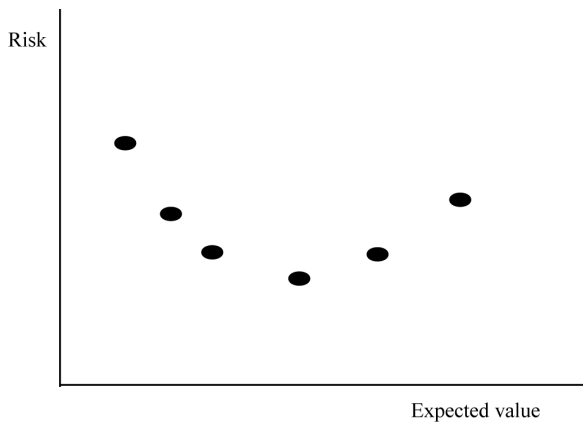
Figure 5. Convex function

Source: authors' drawing.
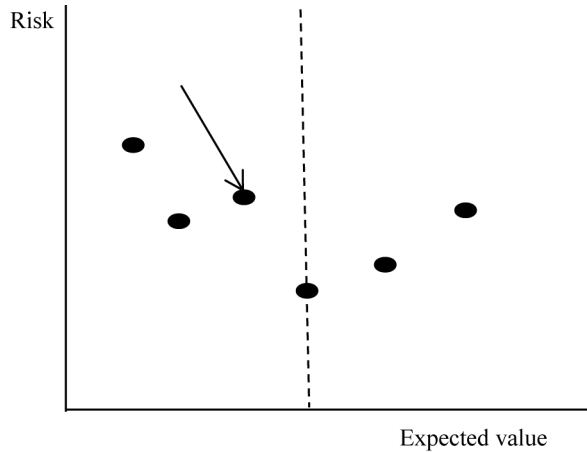
Figure 6. Convex function with local maximum extreme

Source: authors' drawing the first filter finds local extremes and excludes those points which disturb the convexity. The algorithm of the first filter contains these three steps:

1. To find global risk minimum $\sigma(Rp)_{min}$ with expected return $E(Rp)_{min}$, which splits all *the border of the investment opportunities* into two groups.

2. We have to search the group where all expected returns $>E(Rp)_{min}$ and test all the points of the group for the following condition: for every $E(Rp)_k$ and $E(Rp)_l$ where k<l, all risks must fulfil that $\sigma(Rp)_k < \sigma(Rp)_l$. If any $[E(Rp)_k, \sigma(Rp)_l]$ does not fulfil the condition, we have to exclude that point.

3. We have to search the group where all expected returns $<E(Rp)_{min}$ and test all the points of the group for the following condition: for every $E(Rp)_k$ and $E(Rp)_l$ where k>l, all risks must fulfil that $\sigma(Rp)_k < \sigma(Rp)_l$. If any $[E(Rp)_k, \sigma(Rp)_l]$ does not fulfil the condition, we have to exclude that point.

Indexes k, l are indexes of the whole set of data from the smallest expected value to the biggest expected value.

The second filter is sophisticated. It is applied after the first filter has been finished. It finds and excludes those points which disturb the convexity. Figure 7 shows that case.

We developed a test which works with vectors. The test works in the following way:

1. To find global risk minimum $\sigma(Rp)_{min}$ with expected return $E(Rp)_{min}$, which splits all the border of the investment opportunities into two groups. This step software is recalled from the application of the first filter.

2. To search the group where all expected returns $>E(Rp)_{min}$. For every three points $[E(Rp)_i, \sigma(Rp)_i]$, $[E(Rp)_j, \sigma(Rp)_j]$, $[E(Rp)_k, \sigma(Rp)_k]$ where i<j<k two vectors have to be set. Vector **u** (u_1, u_2) and vector **v**(v_1, v_2) where $u_1 = E(Rp)_j - E(Rp)_i$,

$v_1 = E(Rp)_k - E(Rp)_i$, $u_2 = \sigma(Rp)_j - \sigma(Rp)_i$, $v_2 = \sigma(Rp)_k - \sigma(Rp)_i$. Both vectors start from $[E(Rp)_i, \sigma(Rp)_i]$ point. If $u_2/u_1 > v_2/v_1$, then point $[E(Rp)_j, \sigma(Rp)_j]$ has to be excluded.
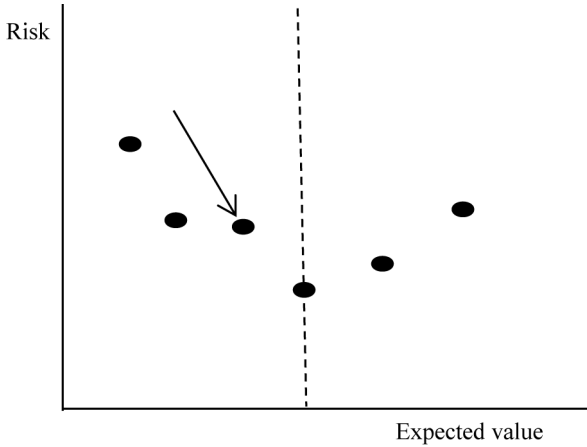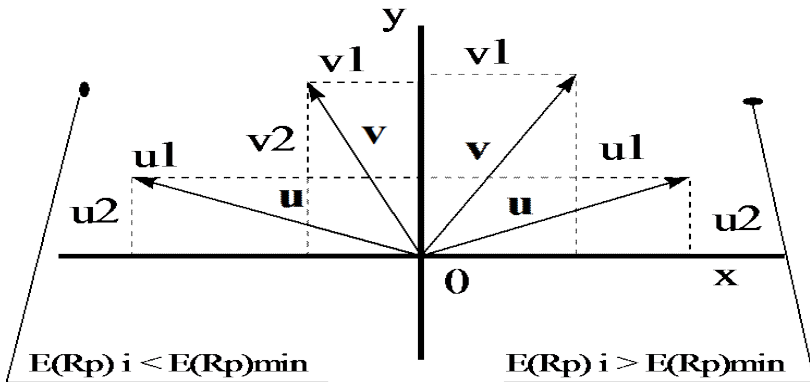

Figure 7. Quasi convex function

Source: authors' drawing.

3. To search the group where all expected returns $<E(Rp)_{min}$. For every three points $[E(Rp)_i, \sigma(Rp)_i]$, $[E(Rp)_j, \sigma(Rp)_j]$, $[E(Rp)_k, \sigma(Rp)_k]$ where $i>j>k$ two vectors have to be set. Vector **u** (u_1, u_2) and vector **v** (v_1, v_2) where $u_1 = E(Rp)_j - E(Rp)_i$, $v1 = E(Rp)_k - E(Rp)_i$, $u_2 = \sigma(Rp)_j - \sigma(Rp)_i$, $v_2 = \sigma(Rp)_k - \sigma(Rp)_i$. Both vectors start from $[E(Rp)_i, \sigma(Rp)_i]$ point. If $u_2/u_1 > v_2/v_1$, then point $[E(Rp)_j, \sigma(Rp)_j]$ has to be excluded.

Figure 8. Vectors of the second filter

Source: authors' calculation.

The test is sophisticated but it is about finding the tangent alpha. The tangent alpha of vector **u** has to be smaller than the tangent alpha of vector **v**. Figure 8 shows the principle of the second filter. The test is very rigorous and we slightly modified it. The modification is that we changed the condition from $u_2/u_1 > v_2/v_1$ to the condition $int(u_2/u_1) > int(v_2/v_1 + 0.5)$. "Int" is a VBA function which ignores the digits after the point.

6. Software – the manual

We have created software where the described algorithms and procedures were implemented. We left the source codes unlocked, so the user can read the code through and discover all the details of coding. The software is written in VBA for Excel and has a very simple menu. Figure 9 shows the menu.
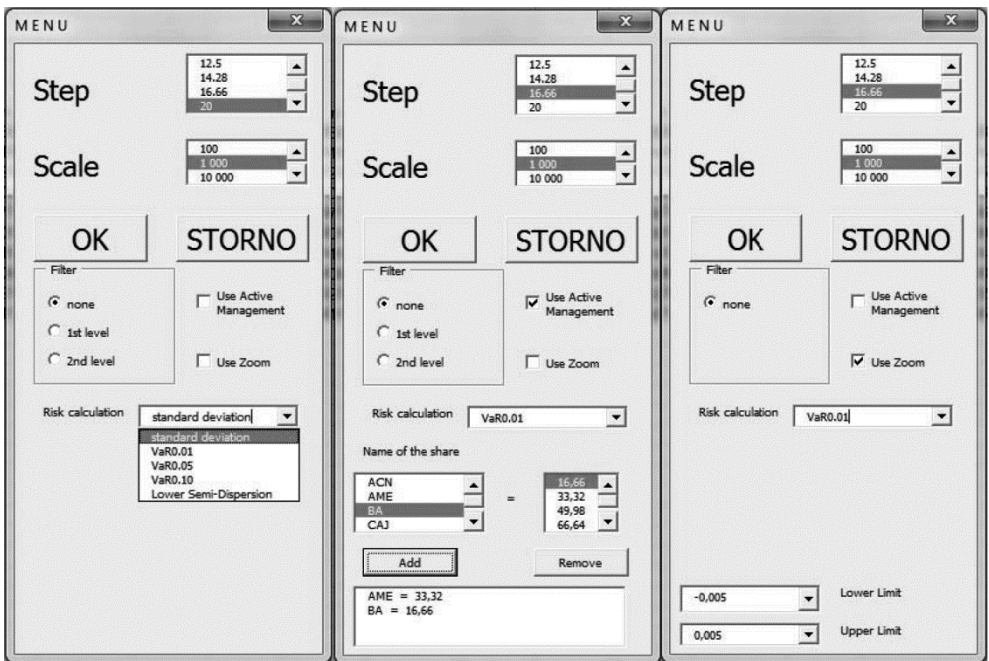
Figure 9. Menu

Source: authors' software.

Figure 9 shows all three possibilities of how MENU can look like. The first possibility is without clicking *Active Management* or *Zoom*. The user can set *Step*, *Scale* and *Filter*. *Step* and *Filters* were already described. *Scale* can be set as: 100, 1000, 10,000 and 10,000. *Scale* sets how the change from the continuous expected return to discrete expected return will be made. 100 means that the expected value

will be rounded to 2 digits after the point. 1000, 10,000 and 10,000 means that expected value will be rounded to 3, 4 and 5 digits after the point.

If the user clicks *Zoom*, they can zoom into a particular part of the border of investment opportunities. *Zoom* disables *Filter*. *Zoom* was developed as a tool for saving time, but it does not save much time. The thinner the zoomed area is, the more time the user should save.

If the user clicks *Active Management* option, they can back or penalise particular shares. This is what portfolio managers often do. *Active Management* also saves a lot of time.

The user writes inputs to the *Input* sheet. *Input* sheet also contains the start, end and duration times. *Input* also contains initial investment which is needed for the VaR calculation. *Input sheet* keeps details about the last calculation (Step, Scale, Filter and enumeration count).

Menu appears after clicking *MENU* banner in the *Input* sheet.

The *Results* sheet contains the portfolios. You can see the JB test values with the red background for those portfolios which are not normally distributed. The program is not solving non-normal distribution problem at this stage of the development neither for portfolios nor for individual shares.

Figure 10. The *Input* sheet

Source: authors' software.

Figure 11. The *Results* sheet

Source: authors' software.

7. Testing

We have compared the results for the particular step with the results achieved with the Solver. We have used the default settings for the Solver in the testing phase. We have not used any specific shares. We have generated the testing portfolio in R. We used a very simple code, which is described here:

```
set.seed(100)
j<-0
x<-0
for (i in seq(-0.08,0.1,0.02)) {
j<-j+1
x[j]<-list(rnorm(60,i,abs(i)*2+0.05))
}
a<-data.frame(x[1:10])
names(a)<-(1:10)
write.table(a,file="shares.txt")
```

We had to find the appropriate measure for testing and comparing the quality of approximation of the border of investment opportunities. We realised that the appropriate measure was:

$$\int_{E(Rp)_{min}}^{E(Rp)_{max}} border\ of\ investment\ opportunities$$

The smaller the integral is, the better the approximation of the *border of the investment opportunities* is. We solved the integral numerically. We used the lower and upper integration sum and also the trapeze integration sum. Table 4 shows the results. We used *Scale* 1000 and standard deviation as a risk calculation. We did not use Filter. The duration is observed in seconds. NOP means the number of calculated points in the *Results* sheet. LIS means lower integration sum. UIS means upper integration sum and TS means trapeze integration sum. The trapeze integration sum is also drawn in Figure 12. Figure 12 and Table 5 also show the integration sum for the calculations made in Solver Excel. As anyone can see, Solver "beats" our method, because its integral sums are smaller than those made in our method. Solver's efficiency can be clearly seen, which is reflected in the duration column in Table 5. We used Solver which has the default settings. We also ran tests in which we used the second level filter. The results are in Table 6.

Table 4. Integral sums for the particular steps

| Step	Duration	Enumerations	NOP	LIS	UIS	TIS
50	0	55	52	0.026011	0.034891	0.030451
33.33	0	220	132	0.024413	0.031087	0.02775
25	0	715	178	0.022761	0.027168	0.024964
20	0	2002	197	0.021778	0.024886	0.023332
16.66	0	5005	207	0.021242	0.023691	0.022466
14.28	1	11440	213	0.020825	0.022803	0.021814
12.5	1	24310	218	0.020524	0.021946	0.021235
11.1	1	48620	221	0.020304	0.021519	0.020912
10	1	92378	225	0.020241	0.021186	0.020713
9.09	1	167960	226	0.020126	0.020916	0.020521
8.33	2	293930	228	0.019989	0.020736	0.020363
7.14	5	817190	232	0.019939	0.020538	0.020239
6.25	13	$2×10^6$	232	0.019835	0.020345	0.02009
5.55	29	$4.7×10^6$	233	0.019778	0.02028	0.020029
5	62	$1×10^7$	234	0.01972	0.020211	0.019965
4.76	89	$1.4×10^7$	234	0.019721	0.020192	0.019956
4	324	$5.3×10^7$	236	0.019697	0.020137	0.019917
3.7	570	$9.4×10^7$	236	0.01968	0.020116	0.019898
3.33	1286	$2.1×10^8$	237	0.01969	0.020103	0.019896
2.5	12596	$2×10^9$	237	0.01965	0.02006	0.019855

Source: authors' calculation.
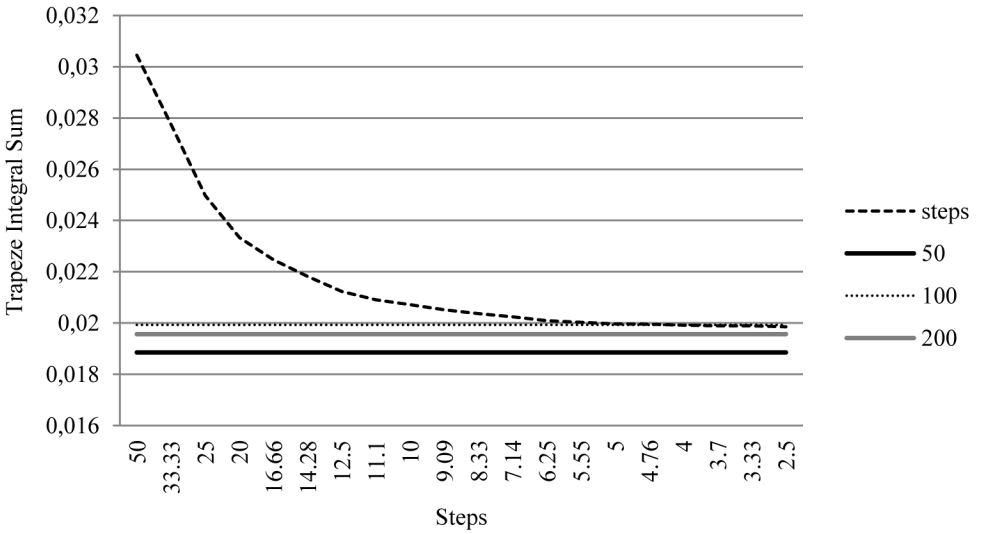
Figure 12. Trapeze integral sums for particular steps and specific points count for Solver

Source: authors' calculation.

Table 5. Integral sums for calculations made in Solver

NOP	Duration	LIS	UIS	TIS
50	6	0.018858	0.018858	0.018858
100	11	0.019332	0.019332	0.019332
200	21	0.019563	0.019563	0.019563
300	31	0.019641	0.019854	0.019854
400	47	0.019681	0.019681	0.019681

Source: authors' calculation.

Table 6. Integral sums for the particular steps and Filter of the second level

Step	Duration	Enumerations	NOP	LIS	UIS	TIS
5	60	1×10^7	191	0.019644	0.020234	0.019939
4.76	92	1.4×10^7	196	0.019652	0.020207	0.01993
4	314	5.3×10^7	215	0.019644	0.020164	0.019904
3.7	574	9.4×10^7	221	0.019622	0.020149	0.019886
3.33	1268	2.1×10^8	238	0.019668	0.020106	0.019887
2.5	11424	2×10^9	232	0.019632	0.020075	0.019853

Source: authors' calculation.

8. Similar solutions

We did not find a fully free solution with a similar functionality. We found some solutions at www.supershareware.com with prices around $200. We also found different solutions but they were also not free. Our solution is 100% free with the unlocked source codes.

9. Conclusions

The software is programmed in VBA Excel, which is very wide-spread software, which is an advantage. We left the source codes unlocked so everyone can read the code through and make any changes, which is another advantage. The results are comparable with the Solver results but the computation time is much longer. Our method shows how efficient the Excel's Solver is. The software illustrates the fundaments of the theory of investment portfolio and is also suitable for teaching purposes. The aim of the research was carried out successfully.

References

Brigham E.F., Ehrhardt M.C., 2005, *Financial Management:Theory and Practice*, South-Western Cengage Learning, Mason.

Elton E.J., Gruber M.J., 2007, *Modern Portfolio Theory and Investment Analysis*, Wiley, New Jersey.

Harrington D.R., 1987, *Modern Portfolio Theory, The Capital Asset Pricing Model and Arbitrage Theory: A User's Guide*, Prentice-Hall.

Keynes J.M., 1936, *The General Theory of Employment, Interest, and Money*, Macmillan, London.

Markowitz H., 1997, *Portfolio Selection – Efficient Diversification of Investment*, Wiley, New Jersey.

McCutcheon J.J., Scott W.F., 1986, *An Introduction to the Mathematics of Finance, Heinemann*, London.

Michalski G., 2008, *Value-based inventory management*, Romanian Journal of Economic Forecasting, No. 9, pp. 82-90.

Michalski G., 2012, *Financial Liquidity Management in Relation to Risk Sensitivity: Polish Enterprises Case, Quantitative Methods in Economics*, Vydavatelstvo EKONOM, Bratislava, pp. 141-160.

Mlynarovič V.,2001, *Finančné Investovanie: Teórie a aplikácie*, Iura Edition, Bratislava.

Pavlík M., Lukáčik M., Michalski G., 2014, *Software for the demonstration of the fundaments of portfolio selection*, Ekonometria 3(45), Wydawnictwo Uniwersytetu Ekonomicznego, Wrocław.

www.quantitativeresearch.eu.