SOLVING CARDINALITY CONSTRAINED PORTFOLIO OPTIMIZATION PROBLEM BY BINARY PARTICLE SWARM OPTIMIZATION ALGORITHM

Aleš Kresta

Klíčová slova:
optimalizace portfolia, binární algoritmus rojení částic

Key words:
portfolio optimization, binary particle swarm optimization

Abstrakt
Při řešení úloh optimalizace portfolia se většinou využívají metody matematického programování. Tyto metody však nemohou být použity, pokud zavedeme omezení počtu držených aktiv. K řešení takto definovaného problému je nutno použít jednu z mnoha heuristických metod (genetické algoritmy, neuronové sítě nebo algrotimus rojení částic). V tomto příspěvku je využito binárního algoritmu rojení částic a metody kvadratického programování při hledání efektivní množiny řešení při optimalizaci portfolia. V článku jsou použity dvě množiny vstupních dat. První množinu tvoří akcie zahrnuté do indexu Dow Jones Industrial Average, druhou pak akcie zahrnuté do indexu Standard & Poor's 500. V závěru příspěvku jsou graficky srovnány nalezené efektivní množiny pro různá omezení počtu držených akcií.

Abstract
Mathematical programming methods dominate in the portfolio optimization problems, but they cannot be used if we introduce a constraint limiting the number of different assets included in the portfolio. To solve this model some of the heuristics methods (such as genetic algorithm, neural networks and particle swarm optimization algorithm) must be used. In this paper we utilize binary particle swarm optimization algorithm and quadratic programming method to find an efficient frontier in portfolio optimization problem. Two datasets are utilized. First dataset consists of the stocks incorporated in the Dow Jones Industrial Average, second dataset contains stocks from the Standard & Poor's 500. The comparison of found efficient frontiers for different limitation on the number of stock held is made at the close of the paper.

Introduction

Proper allocation of the funds is nowadays getting more and more important. With the increasing amount of the money fund managers administer, their responsibility is increasing and quantitative approaches get more attention than qualitative. In the field of the portfolio optimization the pioneer work was Markowitz mean-variance model [1]. Assuming that assets returns follow a multivariate normal distribution, we are concerned only in the portfolio expected return and variance. We are thus looking for the portfolios with maximum expected return and minimal variance. This Pareto efficient set of portfolios is called an efficient frontier (EF). Selection of one optimal portfolio from the efficient frontier then depends only on the risk attitude.

This relatively simple model is easily solvable by the quadratic programming methods. However for a practical purpose, there are two major weaknesses: (i) there is an evidence that assumed multivariate normal distribution of assets returns does not hold [2]; (ii) in the real world, we are interested in integer constraint of a number of assets desired to hold, or to limit the proportion of the assets into a certain interval. Limiting the number of different assets included in the portfolio decreases transaction cost, which can be very high utilized result of the unconstrained model and buying a small number of each asset. Also the monitoring of news and firm's results is easier and so less costly with few assets in the portfolio. One approach to deal with this could be the use of mean-absolute deviation [3] instead of variance as a measure of risk. Problem is then solvable by linear programming methods. However, this approach has been showed to be insensitive to some extremes. Second approach is to use some heuristic method such as the genetic algorithms, neural networks, simulated annealing or particle swarm optimization [4-10]. These methods have showed good results.

The goal of the paper is to exemplify the use of binary particle swarm optimization algorithm in finding the efficient frontier in non-linear portfolio optimization problem. The paper is organized as follows. Section 1 is focused on the portfolio optimization problem, both general and constrained version are described. In section 2 the attention to particle swarm optimization algorithm as a used optimization method is given. In section 3 application and its results on the data from large-cap common stocks actively traded in the United States are presented.

1. Description of portfolio optimization problems

Portfolio optimization consists of a portfolio selection problem in which we want to find the optimum way of investing a particular amount of the money in a given set of securities or assets [7]. While without any information about the investor's risk attitude we are not able to find one optimal portfolio, we are looking for the Pareto efficient frontier. This frontier contains all possible portfolios for which we are not able to lower the risk with the same level of the expected return or to heighten the expected return without increasing the risk. The model in which we consider simple portfolio optimization without any limitation on the maximum number of the different assets we want to hold will be referred as general portfolio optimization problem (Problem P1). The model in which we limit the number of different assets held will be referred as cardinality constrained portfolio optimization problem (Problem P2).

***Problem P1** - General portfolio optimization problem*

Objective function:

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N} x_i \sigma_{ij} x_j ,$$

subject to:

$$\sum_{i=1}^{N} x_i = 1, \tag{C1}$$

$$\sum_{i=1}^{N} x_i E(R_i) = E(R^*), \tag{C2}$$

$$x_i \in [0,1],\ i=1,...,\mathrm{N}. \tag{C3}$$

In this problem N is the number of the available assets, x_i is the proportion of available amount invested in asset i, σ_{ij} is the covariance between assets i and j, $E(R_i)$ is the expected return of i-th asset and $E(R^*)$ is a desired expected portfolio return. The above described minimizing problem is easily solvable by the nonlinear (quadratic) programming methods. The efficient frontier is constructed when solving this model for different values of $E(R^*)$.

As any other model even the unconstrained portfolio optimization model has many premises, which make it very simplified. On the one hand it allows us to find the solution easily; on the other hand the model could not be utilized for a real world application. For example, suppose we are looking for the minimal risk portfolio, which is compounded from the assets incorporated in the Standard & Poor's 500 index. In fact it is not a big problem to find such a portfolio but it would be nearly impossible to invest money into it. It is due to high number of assets we must invest to, which stands for a high transaction costs. These transaction costs are generally dependent on the fact how often we rebalance the portfolio. For simplicity we will abstract from these costs in the paper. So in the real world we want to limit the number of assets we invest in. Introducing K as the desired number of different assets in the portfolio, we can extend the previous model to the cardinality constrained case (Problem P2).

Problem P2 *- Cardinality constrained portfolio optimization problem*

Objective function:

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N} z_i x_i \sigma_{ij} x_j z_j ,$$

subject to:

$$\sum_{i=1}^{N} z_i x_i = 1 , \tag{C1}$$

$$\sum_{i=1}^{N} z_i = K , \tag{C2}$$

$$\sum_{i=1}^{N} x_i z_i E(R_i) = E(R^*) , \tag{C3}$$

$$x_i \in [0,1] ,\ i=1,...,\text{N}, \tag{C4}$$

$$z_i \in \{0,1\} ,\ i=1,\dots,\text{N}. \tag{C5}$$

All variables have the same meaning as in the Problem P1 and z_i is a binary variable taking the value 1 or 0 depending on if i-th asset is or is not included in the portfolio.

The constrained portfolio optimization problem is a mixed-integer nonlinear (quadratic) programming problem for which the computationally effective algorithm does not exist [8]. To solve this problem we can utilize: (i) algorithms for solving mixed-integer nonlinear programs (see [11-15]); (ii) some heuristic method such as particle swarm optimization [4], simulated annealing [5, 6, 8], neural networks [7], genetic algorithms [8-10] and others [16-18]. In this paper a binary particle swarm optimization is used to find the optimal z_i and the weights are determined by the more standard quadratic programming approach.

2. Description of binary particle swarm optimization algorithm

The original particle swarm algorithm was introduced and discussed in [19, 20]. It imitates birds flocking and fish schooling as it is searching in D-dimensional real numbers space for the best position. In this algorithm the certain number of particles is utilized, each particle's position representing solution of the problem. Particles move across the search space partially randomly and partially in the dependence of the personal and global best position discovered so far. Objective function imitates the space richness for food. So particles are clearly determined by their velocities and positions:

$$v_{ji}(t+1) = w(t)v_{ji}(t) + C_1\varphi_1\big(p_{ji} - x_{ji}(t)\big) + C_2\varphi_2\big(g_i - x_{ji}(t)\big), \tag{1}$$

$$x_{ji}(t+1) = x_{ji}(t) + v_{ji}(t+1)\,, \tag{2}$$

where $v_{ji}(t+1)$ is the velocity of the *j*-th particle in the *i*-th dimension in the iteration $t+1$, $w(t)$ stands for momentum, C_1 and C_2 are constants, φ_1 and φ_2 are random numbers from interval <-1,1>, p_{ji} is the personal best position found so far, g_i is the global best position found so far, $x_{ji}(t)$ is the position of the *j*-th particle in the *i*-th dimension in the iteration t. The position of the *j*-th particle can be found as the sum of its previous position $x_j(t)$ and the current velocity $v_{ji}(t+1)$. The whole algorithm is shown in Fig. 1. After a particles initialization, there is the loop, in which the velocity and the position are repeatedly updated according to (1) and (2).

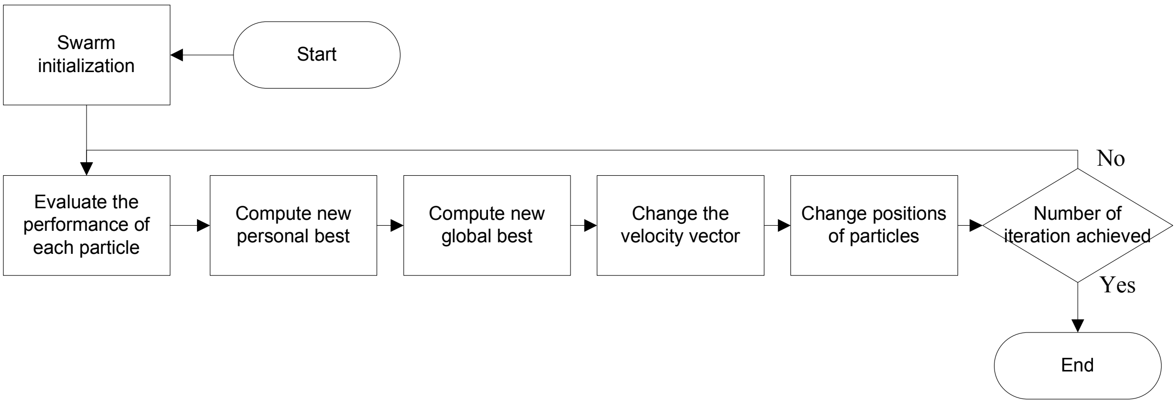
Swarm initialization
Start
Evaluate the performance of each particle
Compute new personal best
Compute new global best
Change the velocity vector
Change positions of particles
Number of iteration achieved
No
Yes
End

Fig. 1: Particle swarm optimization algorithm diagram

In [21] the particle swarm algorithm was modified to operate on a binary variables and discrete binary version of the particle swarm algorithm was introduced (BPSO). There is in fact just one modification – particle's velocity v_{ji} now represents the probability of x_{ji} taking the value 1. Since the probability must be in the interval [0,1], a logistic transformation of v_{ji} is used. The position equation (2) is then changed as follows:

$$if\,(rand() < S(v_{ji}))\quad then\;\; x_{ji} = 1 \quad else\; x_{ji} = 0\,, \tag{3}$$

where $rand()$ is a random number from interval [0,1], $S(\cdot)$ stands for the logistic transformation function:

$$S(x) = \frac{1}{1+e^{-x}}\,. \tag{4}$$

Equation (1) for the velocity remains unchanged, but $x_{ji}(t)$, p_{ji} and g_i are now not real numbers but take values only either 0 or 1.

In this algorithm it is simple to implement the constraint C2 in Problem P2 – we repeat generating random numbers and if they are smaller than $S(v_{ji})$ then we change the corresponding x_{ji} to 1 until $\sum_{i=1}^{N} x_{ji}$ is equal to the desired number of assets held in the portfolio *K*.

3. Application part – efficient frontier estimation

In this section the efficient frontier of constrained portfolio optimization problem defined by Problem P2 is searched. Used data consists of two datasets, which correspond to weekly prices between January 2002 and December 2007 for the assets included in indices Dow Jones Industrial Average (DJI, yahoo finance ticker ^DJI) and Standard & Poor's 500 (S&P, yahoo finance ticker ^GSPC). Both datasets have been acquired from web server http://yahoo.finance.com, [22]. For both datasets stocks with the missing values were eliminated - datasets thus consist of 30 and 465 stocks respectively. Expected returns $E(R_i)$ and covariance matrix σ_{ij} were estimated on weekly basis from downloaded time series (January 2002 to December 2007), thus we assume that the returns and covariances of assets will remain unchanged. For each dataset we are looking for the 500 different optimal portfolios in dependence of the desired expected return $E(R^*)$ and the desired number of assets *K*. The problem of finding optimal portfolio is solved using the BPSO and the quadratic programming. The BPSO is used to find the optimal vector of logical variables z_i while the quadratic programming is utilized in the objective function to find the optimal weights of the selected assets. The implementation of BPSO algorithm is written in MATLAB, utilizing MATLAB's pre-defined functions.

3.1. Dow Jones Industrial Average

Since the DJI dataset contains only 30 assets, it is possible to compare the results of BPSO with the results obtained by trying every possible combination of the assets held (brutal force method). This comparison is made on Fig. 2. We can see that the results are visually the same. As showed in [8], efficient frontier of the cardinality constrained model is a discontinuous curve with jumps. This means that some values of $E(R^*)$ are not rational (since there exist portfolios with the same or lesser risk and higher return).

The time needed for the optimization is showed in Tab. 1.

Tab. 1: Time complexity of the brutal force method and BPSO algorithm

The desired number of assets held (*K*)	Brutal force	BPSO
2	0.220 seconds	1.178 seconds
3	3.336 seconds	2.080 seconds
4	29.842 seconds	2.712 seconds

As we can see for *K*=2 the BPSO is slower than trying every possible combination of chosen assets. This is evoked by the small number of possible combination in the brutal force method. For *K*=3 the BPSO was quicker. For *K*=4 the BPSO was much quicker, using only 10% of the time needed by the brutal force method.

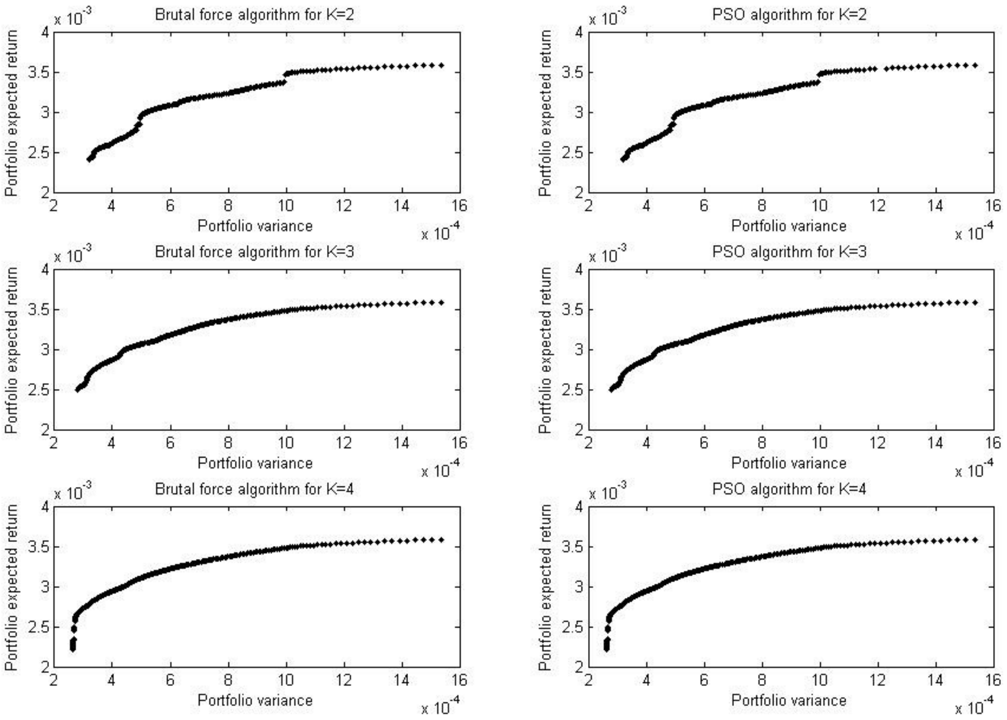

Fig. 2: Cardinality constrained efficient frontier (CCEF) found by brutal force (left) and BPSO (right) for *K*=2 (top), *K*=3 (middle), *K*=4 (bottom)

3.2. Standard & Poor's 500

Because of the high number of the stocks in the S&P dataset the brutal force method is not utilized for its time complexity. We consider only the BPSO algorithm, results of which are shown in Fig. 3. For better clarity the figure is zoomed and thus not the whole efficient frontiers are shown. We can see that the efficient frontier for *K*=3 is not delineated very well. From 500 portfolios found we utilize only 146 to draw efficient frontier, we exclude every portfolio which is not rational (which has the same or bigger risk and lesser return compared to other portfolios). For *K*=2 only 160 and for *K*=4 about 316 portfolios are used to draw the proper efficient frontier. This signifies that searching for efficient frontier in Problem P2 when dealing with vector of 465 stocks is hard to solve for BPSO algorithm. On the other hand the results are given in the reasonable time.

Visually CCEF for *K*=2 lays below CCEF for *K*=3, CCEF for *K*=3 lays below CCEF *K*=4. This is caused by the diversification. With the increasing number of different assets we invest in, the risk is decreasing while the expected return stays the same. We can see that efficient frontiers are very close to each other. Difference between CCEF for *K*=2 and *K*=4 is around 0.1% for variances below 0.0025. This means 0.1% extra expected return if we allow investing into 4 assets instead of 2.

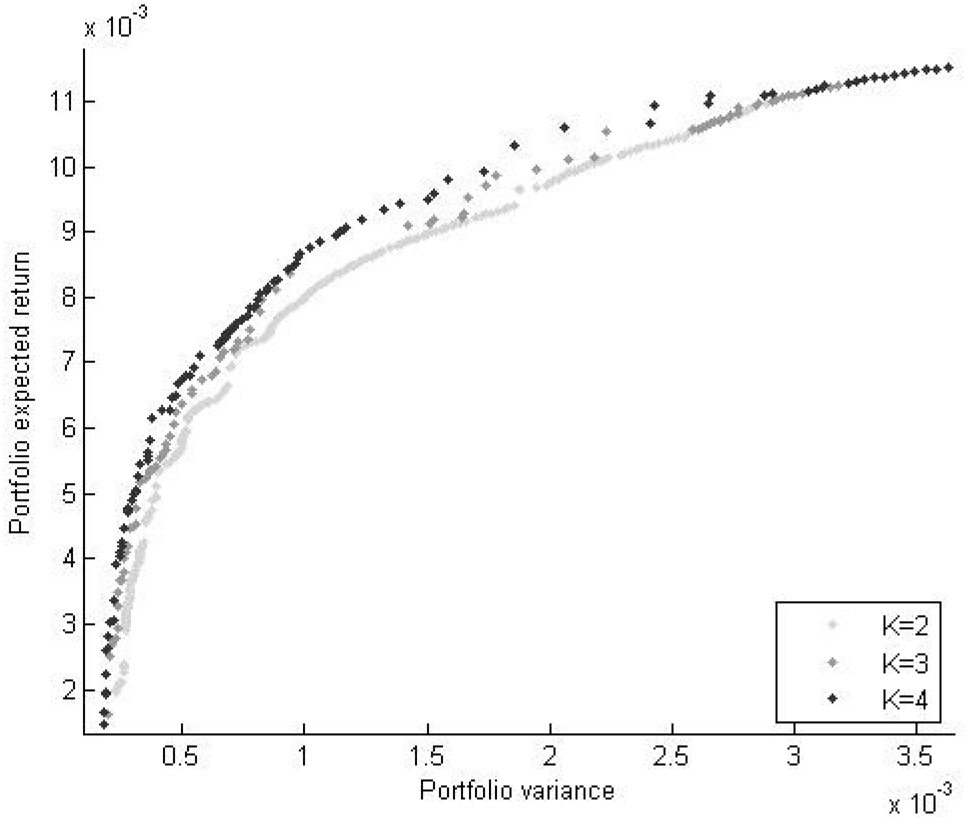

Fig. 3: Cardinality constrained efficient frontiers (CCEF) for S&P with limiting the number of different stocks included in the portfolio

4. Conclusion

This paper was focused on the problem of portfolio optimization. The cardinality constrained mean-variance model was utilized and solved using the binary particle swarm optimization algorithm and the quadratic programming. Two datasets were used. These datasets correspond to the weekly prices between January 2002 and December 2007 for the assets included in the indices Dow Jones Industrial Average for the first dataset and Standard & Poor's 500 for the second dataset. Datasets consist of 30 and 465 assets respectively. For each dataset 500 different optimal portfolios were found and the cardinality constrained efficient frontier was drawn. The results were presented in section 3. We can conclude that for smaller dataset DJI the binary particle swarm optimization outperform the brutal force method in terms of the time complexity. The S&P dataset was a hard nut to crack for the proposed algorithm. But we can say that for this dataset the algorithm gives reasonable result in reasonable time.

Acknowledgements

The research has been elaborated in the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic. The research was also supported by VŠB-TU Ostrava under the SGS project SP2011/7.

References:

[1] MARKOWITZ, H. M. *Portfolio selection.* Journal of Finance, 1952, vol. 7, n. 1, p. 77-91.

[2] MILLS, T. C. *Stylized facts on the temporal and distributional properties of daily FTSE returns.* Applied Financial Economics, 1997, vol. 7, n. 6, p. 599-604.

[3] KONNO, H.; YAMAZAKI, H. *Mean-absolute deviation portfolio optimization model and its application to Tokyo stock market.* Management Science, 1991, vol. 37, n. 5, p. 519-531.

[4] CURA, T. *Particle swarm optimization approach to portfolio optimization.* Nonlinear Analysis: Real World Applications, 2009, vol. 10, n. 4, p. 2396-2406.

[5] CRAMA, Y.; SCHYNS, M. *Simulated annealing for complex portfolio selection problems.* European Journal of Operational Research, 2003, vol. 150, n. 3, p. 546-571.

[6] DERIGS, U.; NICKEL, N.H. *On a local-search heuristic for a class of tracking error minimization problems in portfolio management.* Annals of Operations Research, 2004, vol. 131, n. 1-4, p. 45-77.

[7] FERNANDEZ, A.; GOMEZ, S. *Portfolio selection using neural networks.* Computers and Operations Research, 2007, vol. 34, n. 4, p. 1177-1191.

[8] CHANG, T. J., et al. *Heuristics for cardinality constrained portfolio optimisation.* Computers and Operations Research, 2000, vol. 27, n. 13, p. 1271-1302.

[9] OH, K. J.; KIM, T. Y.; MIN, S. *Using genetic algorithm to support portfolio optimization for index fund management.* Expert Systems with Applications, 2005, vol. 28, n. 2, p. 371-379.

[10] YANG, X. *Improving portfolio efficiency: A Genetic Algorithm approach.* Computational Economics, 2006, vol. 28, n. 1, p. 1-14.

[11] FLOUDAS, C. A. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications.* Oxford University Press: New York, 1995.

[12] BORCHERS, B.; MITCHELL, J. E. *A computational comparison of branch and bound and outer approximation algorithms for 0-1 mixed integer nonlinear programs.* Computers and Operations Research, 1997, vol. 24, n. 8, p. 699-701.

[13] BIENSTOCK, D. *Computational study of a family of mixed-integer quadratic programming problems.* Mathematical Programming, Series B, 1996, vol. 74, n. 2, p. 121-140.

[14] HANSEN, P.; JAUMARD, B.; MATHON, V. *Constrained nonlinear 0-1 programming.* ORSA Journal on Computing, 1993, vol. 5, n. 2, p. 97-119.

[15] BORCHERS, B.; MITCHELL, J.E. *An improved branch and bound algorithm for mixed integer nonlinear programs.* Computers and Operations Research, 1994, vol. 21, n. 4, p. 359-367.

[16] DERIGS, U.; NICKEL, N. H. *Meta-heuristic based decision support for portfolio optimization with a case study on tracking error minimization in passive portfolio management.* OR Spectrum, 2003, vol. 25, n. 3, p. 345-378.

[17] MANSINI, R.; SPERANZA, M. G. *Heuristic algorithms for the portfolio selection problem with minimum transaction lots.* European Journal of Operational Research, 1999, vol. 114, n. 2, p. 219-233.

[18] SCHLOTTMANN, F.; SEESE, D. *A hybrid heuristic approach to discrete multi-objective optimization of credit portfolios.* Computational Statistics and Data Analysis, 2004, vol. 47, n. 2, p. 373-399.

[19] EBERHART, R.; KENNEDY, J. New optimizer using particle swarm theory. In *Proceedings of the International Symposium on Micro Machine and Human Science.* 1995.

[20] KENNEDY, J.; EBERHART, R. Particle swarm optimization. In *IEEE International Conference on Neural Networks - Conference Proceedings*. 1995.

[21] KENNEDY, J.; EBERHART, R. A discrete binary version of the particle swarm algorithm. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*. 1997. p. 4104-4109.

[22] *Yahoo! Finance - Business Finance, Stock Market, Quotes, News* [online]. 2009 [cit. 2009-09-01]. Dostupné z WWW: <http://finance.yahoo.com/>.

[23] ZMEŠKAL, Z.; TICHÝ, T.; DLUHOŠOVÁ, D. *Finanční modely*. Praha: Ekopress, 2004. ISBN 80-86119-87-4.

Classification JEL: C61 , G11

Ing. Aleš Kresta
Katedra financí
Ekonomická fakulta
VŠB-TU Ostrava
Sokolská tř. 33, 701 21 Ostrava
ales.kresta@vsb.cz

Appendix

Pseudo code for BPSO algorithm

```
for 500 different $R^*$
   randomly initialize all particles
   evaluate all particles
   find global best $g_i$ and for each particle personal best $p_{ji}$
   repeat
         for each particle j
             compute new velocity according to Eq. 1
             for each dimension i if $v_{ji} < v_{\min}$ then $v_{ji} = v_{\min}$
             for each dimension i if $v_{ji} > v_{\max}$ then $v_{ji} = v_{\max}$
             $x_{ji} = 0$ for each dimension i
             repeat
                     i= $rand(1,$ number of dimension)
                     $if\ (rand(0,1) < S(v_{ji}))\quad then\ \ x_{ji} = 1$
             until $\sum_{i=1}^{N} x_{ji} = K$
             evaluate particle j
         end
         find new global best $g_i$ and for each particle new personal best $p_{ji}$
   until number of iterations reaches 100
end
```