ATLANTIS PRESS
Advances in Intelligent Systems Research, volume 166
7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019)

The Decision Support of the Securities Portfolio Composition Based on the Particle Swarm Optimization

Efim Bronshtein
Ufa State Aviation Technical University
Ufa, Russia
bro-efim@yandex.ru

Olga Kondrateva
Ufa State Aviation Technical University
Ufa, Russia
kondr_o@mail.ru

***Abstract* — Stochastic behavioral methods are becoming increasingly widespread for optimization tasks solving. One of these methods is the particle swarm optimization (PSO). The particle swarm is an algorithm for finding optimal regions of complex search spaces through the interaction of individuals in a population of particles. Its effectiveness and efficiency has rendered it a valuable metaheuristic approach in various scientific fields where complex optimization problems appear. Investment activity in the stock market is an area where the use of optimization techniques is required. The volatility and unpredictability of future stock prices create a situation of uncertainty for decision making in the financial market. Investors are forced to solve a multi-criteria optimization problem when forming an effective securities portfolio. This article proposes to use PSO to compile an optimal securities portfolio based on combined entropy risk index indicators. The portfolio consisted of 8 shares of Russian companies, the shares of each stock in the portfolio were found using particle swarm optimization. The efficiency of the modified algorithm is investigated, coefficients are proposed, and the results of a computational experiment are obtained.**

Keywords—swarm intelligence, particle swarm optimization, multi-objective optimization task, securities portfolio, index-entropic risk measures.

I. Security Portfolio

The problem of securities portfolio formation is the actual issue in the modern economy. Depositors are investing assets in securities at the world's stock markets every day. The securities portfolio structure optimization is one of the main decision-making tasks in investment activity in the stock market. It is necessary to form such a portfolio of securities, which will bring the best result within a certain period. It is very difficult to generate an optimal portfolio in the face of uncertainty. In this paper a methodology for forming a securities portfolio based on combined index-entropy risk measures using the particle swarm optimization algorithm is outlined.

A securities portfolio is a set of individual types of securities selected by an investor to achieve certain goals. The variety of securities in the market allow to form many securities portfolios different in composition. Investment portfolios may be different in terms of a return and a risk, depending on the current and strategic goals of investors.

Let the portfolio structure be represented as the vector $X=(x_1,x_2,\dots,x_n)$, where x_i is the share of the i – th stock in the portfolio $(i=\overline{1,n})$, i.e. $x_i \geq 0, \sum_{i=1}^{n} x_i = 1$, where n is the quantity of stocks.

The value of the portfolio $P_j(X)$ at the time moment j is equal to the sum of the product of the price of the i – th stock and its share in the portfolio:

$$P_j(X) = \sum_{i=1}^{n} c_{ij} x_i, \quad (1)$$

where c_{ij} is the price of the i-th stock on the j-th day $(j=\overline{1,T})$,

T is the considered time horizon.

The yield of the securities portfolio $V_j(X)$ is calculated by using following formula:

$$V_j(X) = \frac{\sum_{i=1}^{n} c_{ij} x_i}{\sum_{i=1}^{n} c_{i1} x_i} = \frac{P_j(X)}{P_1(X)} \quad (2)$$

All financial transactions, including the formation of a securities portfolio, are associated with risk. The volatility and unpredictability of future stock prices (and therefore the portfolio) creates certain risks for portfolio investor, who needs to manage risks reasonably.

The investor have to form a portfolio of financial instruments in such a way as to protect against various types of risk. Risk refers to the possibility of non-receipt of the expected income or loss (full or partial) of the funds invested in this security. Risk is a reflection of the uncertainty in the receipt of income by the investor, so each investor has a subjective attitude to the investment process – a measure of risk aversion.

The use of optimization techniques in the investment theory began with the solution of the problem of constructing an optimal portfolio based on two criteria – profitability and risk. As a rule, securities with a low risk have a small expected return, and securities that can generate high returns have significant risk indicators. Since a portfolio is a set of various securities, the investor will always face the problem of choosing the currently effective investment portfolio. The choose of investors determined by the presence of a certain free amount of money for investment and time for their alienation, as well as the natural desire to receive as much income as possible while minimizing risk.

II. PARTICLE SWARM OPTIMIZATION

Metaheuristic methods are relatively new and rapidly developing methods. Among these methods are evolutionary and behavioral methods. Behavioral methods are based on modeling the collective behavior of self-organizing living or non-living systems. The interacting elements of these systems are called agents. The key ideas of behavioral methods are decentralization, agent interaction, and agent simplicity.

The particle swarm optimization is one of the evolutionary computation techniques and the most powerful methods for solving global optimization problems. The particle swarm algorithm is a heuristic algorithm and belongs to the behavioral method, which is based on the socio-psychological behavioral model of the crowd. It isn't need to know the exact gradient of the function being optimized, so the function does not have to be differentiable, but may be discontinuous, noisy, etc.

The prototype of the method under consideration was the behavior of a flock of birds, the observation of which prompted Craig Reynolds to create a computer model of Boids in 1986 [1]. Birds fly in large groups, almost never collide in the air, move smoothly and in a coordinated way, as if someone controls them – a flock has swarm intelligence.

In 1995 James Kennedy and Russell Eberhart proposed a method for optimizing continuous functions, which they called, the particle swarm optimization [2, 3] based on the Reynolds model and the work of Heppner and Grenadier [4]. This algorithm describes the decision making of particles in a flock and is a simple and effective way to optimize, and the position of each particle in the swarm determines the possible solution of the optimization problem. Thus, the algorithm models a multi-agent system, where particle agents move to optimal solutions, sharing information with their neighbors, since the task of the birds flock is to find the best conditions for existence, which are defined by the task functional.

The algorithm works by initializing a flock of birds randomly over the searching space, where every bird is called as a particle. Particle swarms explore the search space through a population of particles, which adapt by returning to previously successful regions. These particles fly with a certain velocity and find the global best position after some iteration. At every initialize position, each particle can adjust its velocity vector, based on its momentum and the influence of its best position as well as the best position of its neighbors, and then compute a new position that the particle is to fly to. The movement of the particles is stochastic, however, it is influenced by the particle's own memories as well as the memories of its peers. Each particle keeps track of its coordinates in the problem space.

At each moment of the time, agents-particles have a certain position and velocity vector in the space. Both of these parameters are randomly selected at the initialization stage. For each position of the particle, the corresponding value of the objective function is calculated, and on this basis, according to certain rules, the particle changes its position and speed in the search space. In addition, each particle stores the coordinates of the best solution it has found, as well as the best solution, from the swarm passed by all particles. These values can be updated at the each iteration depending on the optimization criteria. They are also used to calculate the new position of the particle. This operation simulates the exchange of the information between birds in the flock. The information is the found optimum of the agent-particle and the found optimum of the entire swarm at a specific point in time.

Thus, with each iteration, the particles adjust the velocity vector and their directions, trying to get closer to the best point of the swarm, taking into account their individual optimum. The desired function is calculated at the each iteration and the best solutions of the particle and the entire swarm are updated. The potential solutions fly through the research space by following the current optimum solution. Every particle finds its personal best position and the group best position at the each iteration, and then modifies their progressing direction and speed to reach the optimized position quickly.

Particle swarm optimization comprises a very simple concept and paradigms can be implemented in a few lines of computer code. A great deal of spreading of the PSO was obtained when solving problems of optimization of nonlinear-multidimensional functions. PSO algorithm has been successfully applied in many areas, in various exact and experimental sciences, such as bioengineering, etc.

A. *Traditional PSO and its Modifications*

Given the objective function that must be minimized: $\min_{x \in U} F(x)$. Here $F(x): R^D \to R$ is the functional based on which the quality of the position of the swarm particle is determined. Let x_i , v_i be vectors of parameters which characterize the position of the swarm particle and the velocity of the *ith* particle $(i = \overline{1, S})$, where S is the number of particles in the swarm.

The components of this vector define the position of the particle in the search space U, and the objective function F characterizes the state of the particle on the basis of this position. The parameters are subject to the restrictions $x \in U \subseteq R^D$, which specify the maximum and minimum values for each of them, $b_{low} \leq x \leq b_{up}$.

The task is considered on a discrete time interval, i.e. each moment of time corresponds to a separate iteration of the algorithm. The speed of each particle in the swarm is updated by using the following equation:

$$v_{i,t+1} = v_{i,t} + c_1 \cdot Rand_1() \cdot (p_{i,t} - x_{i,t}) + c_2 \cdot Rand_2() \cdot (g_t - x_{i,t}) \quad (3)$$

where $v_{i,t}$ is the current velocity of the *ith* particle at time step *t*, $Rand_1()$ and $Rand_2()$ are two independent random numbers in the range of (0,1], $p_{i,t}$ is the personal best solution found by the agent-particle at time step (*t* + *1*), g_t is the best solution of the whole swarm, c_1 and c_2 are weighting factors that reflect the strength of the influence of the personal best solution and the global best value.

The first addendum is the inertia of the particle, the second addendum implements the principle of simple nostalgia and it is a cognitive component reflecting the particle's memory of its own best position $p_{i,t}$ at the time (t + 1).

Since the vector g_t is the best point of the entire swarm, the third term is a collective memory or social component, which is responsible for the tendency of the particle-agent to the best position found.

Thus, a change in the speed of each particle or acceleration is defined as a weighted sum of two vectors, the first of which is directed to the best solution found by this agent, and the second to the global optimum detected by the whole swarm.

The position of each particle is defined as the adjustment of the particle coordinates at the previous time point to its current velosity. Each particle is updated by using the following equation:

$$x_{i,t+1} = v_{i,t+1} + x_{i,t} \quad (4)$$

where $x_{i,t}$ are the position of the *ith* particle at the time moment *t*.

The number of iterations and particles in the swarm determines the duration of the algorithm, and therefore affects the quality of the results.

The PSO algorithm works as follows:

Step 1 – creating a swarm of particles. The initial positions and velocities for *S* swarm particles are generated using random vectors $v_{i,1}, x_{i,1} \sim U(b_{low}, b_{up})$ with a multidimensional uniform distribution.

Step 2 – finding the best value for each particle. It is needs to calculate the value of the function being optimized at the current point. If the value of the function is less than the personal best value of the particle, then replace it with a new solution corresponding new value of objective function: $p_{i,t} \leftarrow F(x_{i,t})$.

Step 3 – finding the best solution among all particles. The values of the best solutions of all particles are compared with the best value of the entire swarm. If the personal best value of the particle is less than global optimum $p_{i,t} < g_t$ then replace it with current personal best value: $g_{t+1} \leftarrow p_{i,t}$. During the first iteration of the algorithm, the global optimum is assigned the minimum value from the personal best particles values: $g_1 \leftarrow \min_{x \in U} F(x_{i,1})$.

Step 4 – correction of the speed and position of each particle. The velocities and coordinates of the particles are calculated according to (3) and (4). At this step, it is necessary to check whether the particle with the updated position is in the admissible region *U*. If the *jth* component of the vector $x^j_{i,t+1}$ is out of the permissible frame, its value is taken equal to the boundary one, and the velocity component $v^j_{i,t+1}$ is reset. Similarly, when calculating the speed, it is checked that any of its *jth* component does not exceed the value equal to $b^j_{up} - b^j_{low}$. In this case, the position of the particle will certainly go beyond the limits of *U* and the speed is taken as the maximum allowable, equal to $b^j_{up} - b^j_{low}$.

It should be noted that there are several ways out of the situation when the particle velocity does not meet the criterion of the permissible values limits. There is a possibility to reflect the particle back into the area of possible solutions while maintaining speed, or “extinguish” the speed of the particle when it comes into contact with the boundary of the area, preventing the particle from leaving it. There is a possibility also just ignore these particles, leaving the area of possible solutions, and wait for her to return to this area.

Step 5 – checking the criterion for stopping the operation of the algorithm. In the general case, the algorithm stops working when the particle has moved through all iterations, that is, $t = k$, where k is the number of iterations. When during several iterations g does not change, you can stop the algorithm and take the current value of g as the optimal solution to this problem. If the criterion is not met, then go to Step 2.

A several modifications of the traditional algorithm were proposed. They relate to combining the algorithm with other optimization algorithms, reducing the probability of premature convergence by changing the characteristics of particle motion, and dynamically changing the parameters of the algorithm during optimization.

For example, Kennedy and Eberhart suggested using information about the solutions of neighboring particles l_t, replacing the global optimum g_t in the following formula:

$$v_{i,t+1} = v_{i,t} + c_1 \cdot Rand_1() \cdot (p_{i,t} - x_{i,t}) + c_2 \cdot Rand_2() \cdot (l_t - x_{i,t}) \quad (5)$$

where l_t is the best result among the particle and its neighbors. Neighboring are considered either particles that differ from this index by a step of a certain size, or particles whose distance does not exceed a predetermined threshold. This algorithm explores the search space more thoroughly, but is slower than the original one. And the smaller the number of neighbors taken into account when forming the velocity vector, the lower the rate of convergence of the algorithm.

In 1998, Shi and Eberhart [5] noticed that one of the main problems in solving optimization problems is the balance between the thoroughness of research of the search space and the rate of convergence of the algorithm. This balance may be different depending on the task itself and the characteristics of the search space, so they proposed to change the rule for updating the particle velocity vectors:

$$v_{i,t+1} = c_0 \cdot v_{i,t} + c_1 \cdot Rand_1() \cdot (p_{i,t} - x_{i,t}) + c_2 \cdot Rand_2() \cdot (g_t - x_{i,t}) \quad (6)$$

c_0 is the coefficient of the inertia, which determines the balance between the breadth of the search and attention to the found best personal solutions. The inertia weight controls particles momentum so that they can avoid continuing to explore the wide search space and switch to fine tuning when a good area is found. When $c_0 >$ 1, the particles speed increases, they fly apart and cover a larger search space. When c_0 <1, the particles velocity decreases with time and the rate of convergence of the algorithm depends on the choice of the parameters c_1 and c_2.

In 2002, Claire and Kennedy proposed a variant, in which there is no need to "guess", to adjust the values of the adjustable parameters of the algorithm in order to control the convergence of particles. This model is called the canonical algorithm of a particle swarm, in which the method of calculating the particle velocity vectors is changed by introducing an additional factorχ:

$$v_{i,t+1} = \chi \cdot [v_{i,t} + c_1 \cdot Rand_1() \cdot (p_{i,t} - x_{i,t}) + c_2 \cdot Rand_2() \cdot (g_t - x_{i,t})], \quad (7)$$

where χ is the constriction coefficient which calculated as:

$$\chi = \sqrt{\frac{2k}{\left|2-c-\sqrt{c^2-4c}\right|}} \quad (8)$$

where $c = c_1 + c_2 > 4$.

This algorithm requires no explicit limit. Such an approach makes it possible to have the convergence of the algorithm without the need to control the velocity of the particles, because the constriction coefficient makes it unnecessary [6].

III. The Task of Forming Optimal Investment Portfolio by the Particle Swarm Method

The stock portfolio is optimal when the return and risk are balanced in the proportion acceptable for the investor, therefore, it is necessary to achieve the required level of return with a minimum losses for the investor, when forming the composition of the portfolio.

Thus, it is necessary to solve a two-stage optimization problem, where at the first stage portfolios are selected that are characterized by the minimum risk measure, and at the second stage, the one with the highest yield is selected from the choosed portfolios at the first stage. In order to solve the issue of the security portfolio forming, we estimate the risk of portfolios on the interval T and compare the yield of these portfolios on the interval τ, i.e. by delimiting the time intervals at which the risk and profitability of portfolios are analyzed.

At the same time, when forming of a securities portfolio, it is necessary to apply portfolio diversification, which will eliminate the variation in the return levels of various financial assets and reduce risks. It is possible to use the brute force algorithm for the search of the effective portfolio with a small number of stocks in the portfolio. But if the diversification condition is met, the number of shares in the portfolio increases, and it becomes necessary to select and apply a numerical optimization method.

Thus, when forming a securities portfolio, it becomes important to use such advantages of the particle swarm method as: fewer calculations of the objective function, clarity and simplicity of the software implementation of the algorithm, greater reliability in the search for a global optimum.

A. *Conditions of the Problem*

The search space is a simplex $U = [0, 1]^N$, and the position of each particle of the swarm $x_{i,t}$ corresponds to a certain portfolio of securities, which is characterized by the quantitative composition of stocks in it, i.e. investment portfolios differ depending on shares of financial instruments. Thus, the number of components of the particle $x_{i,t}^j$ corresponds to the number of stocks included in the securities portfolio.

The following restrictions apply to the position of the particle and its velocity: $\sum_{j=1}^{N} x_{i,t}^j = 1$, $\sum_{j=1}^{N} v_{i,t}^j = 0$, where N is the number of shares in the portfolio, $x_{i,t}^j \in [0,1)$, thereby ensuring the portfolio diversification.

In contrast to the traditional particle swarm algorithm, different functions are used as a criterion: the functions of risk measures considered in [7] are used to select the best personal values $p_{i,t}$ of the particles and the return function of the investment portfolio (2) corresponding to the selected best agents, are used to determine g_t which the best solution for the whole swarm.

Due to the fact that several combined index-entropic measures (*E-M*$_1$, *E-M*$_2$, *E-M*$_3$, *E-CVaR*) are considered for portfolio risk assessment, a multi-swarm algorithm is used. The each swarm corresponds to a certain method of calculating the risk of the investment portfolio. There is a need to minimize the risk measure function: $F_l(x_{i,t}) \to min$, where l is the number of the swarm.

$$F_1 = \lambda\left(\gamma \ln E\left[exp\left(-\frac{V(X_{i,t})}{\gamma}\right)\right]\right) + (1-\lambda)M_1, \quad (9)$$

$$F_2 = \lambda\left(\gamma \ln E\left[exp\left(-\frac{V(X_{i,t})}{\gamma}\right)\right]\right) + (1-\lambda)M_2, \quad (10)$$

$$F_3 = \lambda\left(\gamma \ln E\left[exp\left(-\frac{V(X_{i,t})}{\gamma}\right)\right]\right) + (1-\lambda)M_3 \quad (11)$$

$$F_4 = \lambda\left(\gamma \ln E\left[exp\left(-\frac{V(X_{i,t})}{\gamma}\right)\right]\right) + (1-\lambda)CVaR \quad (12)$$

The search for the best solution g_t is carried out among all the personal best values of the particles $p_{l,i,t}$ of all the swarms under consideration, which will ensure the interaction of particles between the swarms.

The criterion for updating g_t (2) is the function of portfolio profitability, which must be maximized: $F^*(p_{l,i,t}) \to max$. Thus, in contrast to the traditional PSO, a minimax approach is used for the search personal and global optimums.

The dynamic version of the task is that for each position of the agent- particle in the solution space, the profitability function of the portfolio $F^*(p_{l,i,t})$, as well as the functions of risk measures $F_l(x_{i,t})$, are calculated over a long time interval T at time moments 0, τ, ..., n τ, where n is the number of periods considered, τ is the interval for which the portfolio returns are calculated, according to the methodology described in [8]. Therefore, the average value of the risk measure $E_{k=0}^{n}[F_{k,l}(x_{i,t})]$ is used to search for the best personal particle value. The cumulative yield $\prod_{k=0}^{n} F_k^*(p_{l,i,t})$ is used to search the best decision among all swarms g_t. Like the investor made a choice in favor of the current portfolio again and again n times in a row when making a purchase decision of the security portfolio.

B. Computational Experiment

Historical data for the period from January 05, 2015 on October 30, 2018 was used as the initial data of the computational experiment. Daily quotes for ordinary shares of 8 companies belong to the first echelon such as PJSC "Oil Company Lukoil", PJSC "Aeroflot", PJSC "Sberbank", PJSC "Gazprom", PJSC "Bank VTB", PJSC "MTS", JSC "Tander" ", PJSC" MMC Norilsk Nickel "[9].

At the first stage in each swarm, corresponding to the combined index-entropic risk measure, 10 particle-portfolios were generated. These portfolios were revised at each iteration according to the particle swarm optimization algorithm. The number of iterations ranged from 50 to 300 in increments of 50, so the maximum number of analyzed portfolios was 3000.

To identify the best personal value of a particle, risk measures were calculated on long time intervals *T*, according to [8]. These intervals equal to 5 months, 9 months and 14 months. Then among all the best personal values of particles of all swarms was selected the particle-portfolio with the maximum yield to determine the global optimum. The results of these calculations are presented in Table 1.

TABLE I. THE EFFECTIVE PORTFOLIOS RETURN

Time Interval, *T*	**Number of Iterations, *k***					
	50	***100***	***150***	***200***	***250***	***300***
5 month	1.160	1.192	1.140	1.120	1.092	1.104
9 month	1.224	1.278	1.251	1.183	1.207	1.172
14 month	1.327	**1.332**	1.324	1.321	1.328	1.326

The data on the optimal portfolios returns, presented in Table 1, show that it is enough to use no more than 100 iterations for form effective portfolios.

Since the sum of shares in the portfolio should not exceed one, it is necessary to check the condition, when the speed value calculated at the each iteration: $\sum_{j=1}^{N} v_{i,t}^{j} = 0$. The velocity vector was updated according to formula (6), where the inertia coefficient c_0 ranged from 0.529 to 0.929, and the cognitive and social coefficients, c_1 and c_2 respectively, varied from 1.49 to 2.09 in increments of 0.1.

The results of the calculations are reflected in Table 2 and are the basis for the following conclusions: in most cases, with using c_1 and c_2 equal to 1.79, optimal portfolios are formed, with the condition that the inertia coefficient is fixed, and the portfolio with record yield is obtained with c_0 = 0.729.

TABLE II. PARAMETER SELECTION

Cognitive and Social Coefficients, c_1 and c_2	**Inertia Coefficient, c_0**				
	0,529	***0,629***	***0,729***	***0,829***	***0,929***
2,09	1,318	1,329	1,331	1,323	1,323
1,99	1,167	1,284	**1,332**	1,326	1,329
1,89	1,21	1,329	1,331	1,33	1,33
1,79	1,226	1,195	1,33	1,331	1,329
1,69	1,219	1,33	1,229	1,328	1,331
1,59	1,129	1,068	1,308	1,325	1,327
1,49	1,134	1,323	1,326	1,327	1,314

So the new positions of the particles were calculated according to the formula (4), and a personal and global optimum was checked at each iteration. As shown in Fig. 1, one of the particles reached its best value at the 21st iteration.

The particle swarm optimization algorithm was launched more than 300 times to obtain results. It was analyzed the impact of the chosen risk measures on the ability to form an effective portfolio with a record value of profitability. Four swarms used according to the number of selected risk measures in the PSO algorithm. The global optimum was found among all the best personal solutions of all swarms for different values of parameters. The swarms were tracked on the basis of the particle of which became the optimal portfolio, i.e. the risk measures were tracked at which the effective portfolio was formed.

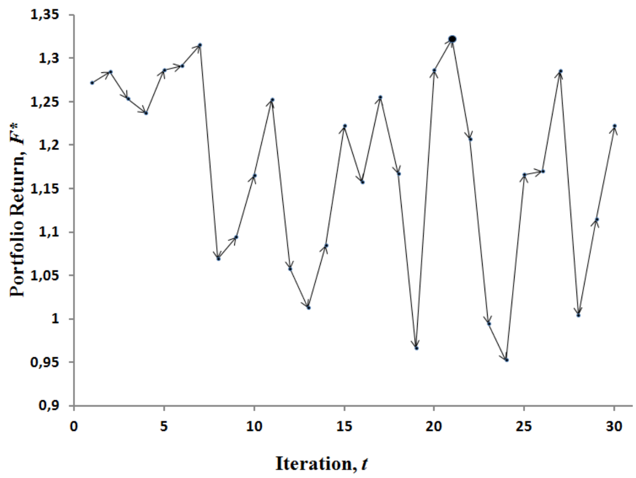

Fig. 1 Fragment of the particle trajectory

Fig. 2 shows the composition of the found effective portfolio. Pie charts correspond to shares in the security portfolio.

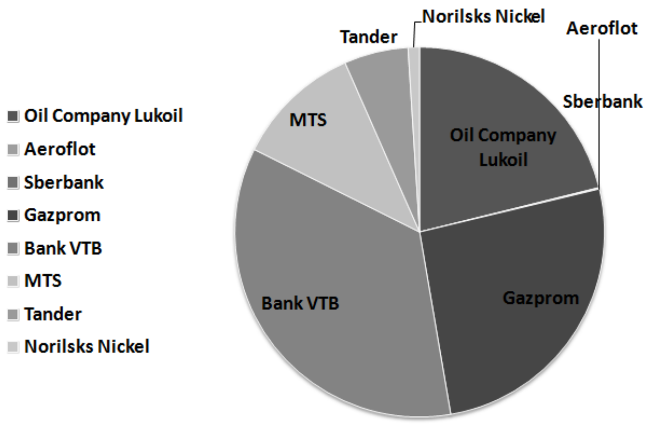

Fig. 2 The security portfolio composition

IV. Conclusion

For the first time, it was proposed to use the particle swarm optimization algorithm for solving the problem of forming an optimal securities portfolio and it was investigated the effectiveness of the proposed modified algorithm. The results of the analysis of the computational experiment showed that:

- the using of combined index-entropic risk measures $E\text{-}M_2$ and $E\text{-}M_3$ make possible to form more efficient portfolios than the using other measures;
- more half of all obtained optimal portfolios were formed on the basis of measure $E\text{-}M_2$, and the fourth part were formed by using measure $E\text{-}M_3$;
- it is advisable to launch the particle swarm method in a mode of 100 iterations;
- algorithm coefficients are configured to use PSO algorithm in the formation of an investment portfolio. It is recommended to use the parameter of the inertia c_0 = 0.829; and cognitive and social coefficients $c_1 = c_2$ = 1.729.

Acknowledgment

This work was supported by the Russian Foundation for Basic Research, project no. 19-08-00895.

References

[1] Reynolds C. Flocks, Herds, and Schools: A Distributed Behavioral Model // Computer Graphics. 1987. №21(4). P. 25–34.

[2] Kennedy J., Eberhart R. C. Particle swarm optimization // Proc. of IEEE International Conference on Neural Networks, 1995. P. 1942–1948.

[3] Eberhart R. C., Kennedy J. A new optimizer using particle swarm theory // Proc. of the Sixth International Symposium on Micro Machine and Human Science, MHS'95, 1995. P.39–43.

[4] Heppner F., Grenander U. A stochastic nonlinear model for coordinated bird flocks // The Ubiquity of Chaos. 1990. P. 233–238.

[5] Shi Y., Eberhart R. A modified particle swarm optimizer // Proc. of the IEEE International Conference on Evolutionary Computation Proceedings, 1998. P. 69–73.

[6] Clerc M., Kennedy J. The particle swarm – explosion, stability, and convergence in a multidimensional complex space" // Proc. of the IEEE Transactions on Evolutionary Computation, №6(1). 2002. P. 58–73.

[7] Kondrateva O. V. Application of indexed-entropic risk measures in decision support systems for security portfolio management // Proc. of the 3rd international conference on intelligent technologies for information processing and management (ITIPM'2015), P.159-162.

[8] Bronshtein E. M., Kondrateva O. V. Security Portfolio Management Based on Combined Entropic Risk Measures // Journal of Computer and Systems Sciences International. 2013. Vol. 52. №. 5. P. 837–841.

[9] Public Joint-Stock Company "Moscow Exchange MICEX-RTS". URL: http://www.moex.com