Framework for constrained portfolio selection by the firefly algorithm

Milan Tuba, Nebojsa Bacanin, and Branislav Pelevic

***Abstract* —** Well-known economics and finance problem of portfolio selection (optimization) has received a lot of attention in recent decades and many methods and techniques exist for tackling this problem. Classical mean-variance problem model is directed towards simultaneously maximizing the expected return of the portfolio and minimizing portfolio variance. Additional constraints are being added to the basic problem definition to make it more accordant with the real world, but the problem than becomes computationally very expensive or intractable. Standard deterministic techniques become insufficient and optimization metaheuristics emerge as a better approach. In this paper, firefly algorithm (FA), as one of the latest swarm intelligence metaheuristics, was applied to the portfolio optimization problem. The developed framework was tested on a set of five assets with promising results.

***Keywords* —** portfolio optimization problem, metaheuristic optimization, swarm intelligence, firefly algorithm (FA), nature inspired algorithms.

I. Introduction

PORTFOLIO optimization problem, also known as portfolio selection problem, is one of the most studied research topics in the field of finance and economics. Financial portfolio is collection of financial instruments (investments), all owned by the same organization or by an individual. It usually includes bonds (investments in debts), stocks (investments in individual businesses), and mutual funds (pools of money from many professional investors). Portfolio structure is generally designed according to the investor's risk sensitivity, objectives of an investment and a time frame.

In its basic definition, portfolio optimization problem is dealing with the selection of portfolio's assets (or securities) that minimizes the risk subject to the constraint that guarantees a given level of returns. Individual and institutional investors prefer to invest in portfolios rather than in a single asset because by doing this, the risk is mitigated with no negative impact on the expected returns [1]. In other words, portfolio optimization problem seeks for an optimal way to distribute a given budget on a set of available assets [2]. The goal to select a portfolio with minimum risk at defined minimal expected returns means reducing non-systematic risks to zero. Alternatively, portfolio optimization problem can be defined as multicriteria optimization in which risks have to be minimized, while, on the other hand, return has to be maximized. Unfortunately, this approach to the problem has several drawbacks [3]. First, it can be difficult to collect enough data for precise estimation of the risk and returns. Second, the estimation of return and covariance (used for defining the risk) from historical data is very prone to measurement errors [4]. Finally, this model is considered to be too simplistic for practical purposes because it does not capture many properties of the real-world trading, such as maximum size of portfolio, transaction costs, preferences over assets, cost management, etc. These properties can be modeled by adding additional constraints to the basic problem formulation leading to the constrained portfolio optimization problem. Constrained problem is more complex than the unconstrained one, and belong to the class of NP-Complete problems [5].

Portfolio optimization problem can be solved using various methods and techniques. Fuzzy portfolio selection problem was successfully solved using parametric quadratic programming technique [6], linear programming method [7] or application of integer programming [8].

As mentioned above, constrained portfolio optimization problem adds additional real-world requirements to the basic problem formulation. Moreover, in some cases, portfolio characteristics, such is its size (number of assets in portfolio), makes the problem intractable in a reasonable amount of computational time. In these cases exact methods cannot obtain results and the use of approximate algorithms, and in particular metaheuristics, is necessary. Modern metaheuristic algorithms are typically high level strategies which guide an underlying subordinate heuristic to the desired objective. Metaheuristic methods can find satisfying feasible solution in a reasonable amount of computational time.

One of the most interesting groups of metaheuristics are nature-inspired algorithms. These algorithms mimic the behavior of natural systems and present an important subset of metaheuristic methods. They can roughly be divided into two groups: evolutionary algorithms (EA) and swarm intelligence.

Manuscript received June 26, 2013.

The research was supported by the Ministry of Science of Republic of Serbia, Grant No. III-44006

M. Tuba is with the Graduate School of Computer Science, Megatrend University, Belgrade, Serbia, e-mail: tuba@ieee.org

N. Bacanin is with the Graduate School of Computer Science, Megatrend University, Belgrade, Serbia, e-mail: nbacanin@megatrend.edu.rs

B. Pelevic is with the Graduate School for International Economics, Megatrend University Belgrade, Serbia, e-mail: bpelevic@megatrend.edu.rs

Well-known representative of EA is genetic algorithms (GA). GA is an iterative approach that employs natural operators: selection, crossover and mutation. GA was successfully applied on portfolio optimization problem [9].

Swarm intelligence is using principles of the collective behavior of social insect colonies and other animal groups in the search process. It belongs to the group of population based algorithms. They start with initial (usually random) population of candidate problem solutions and iteratively improve them. The key concept of swarm intelligence lies in the effect of emergent behavior of many individuals that exhibit extraordinary collective intelligence without any centralized supervision mechanism. Entire swarm intelligence system is fully adaptive to internal and external changes and it is established on four basic properties on which self-organization rely: positive feedback, negative feedback, multiple interactions and fluctuations. Positive feedback refers to a situation when one individual directs behavior of the others by some directive. Negative feedback discourages individuals to pursuit bad solution to the problem. Multiple interactions are the basis of the tasks to be carried out by certain rules, while fluctuations refer to the random behaviors of individuals by which the new regions are being explored. Swarm intelligence approach has obvious advantages over other methods: scalability, adaptation, fault tolerance, parallelism and speed.

Particle swarm optimization (PSO) is a swarm intelligence algorithm which mimics social behavior of fish schooling or bird flocking. PSO was tested on portfolio optimization problem [10]. Ant colony optimization (ACO) showed great success in solving many hard optimization problems [11], [12], [13], [14]. ACO was inspired by the foraging behavior of ants who deposit pheromone trails which help them in finding the shortest path between food sources and their nests. The philosophy of ACO algorithm involves the movement of an ant colony which is directed by two local decision policies: pheromone trails and its attractiveness.

Artificial bee colony (ABC) metaheristic is one of the latest simulations of the honey bee swarm. In this implementation, three group of bees: employed bees, onlookers and scouts work together and carry exploitation and exploration processes. Software implementation of ABC algorithm can be found in the literature [15]. ABC showed outstanding results in constrained optimization problems [16], and satisfying results on engineering problems in its modified form [17].

Seeker optimization algorithm (SOA) mimics the human search process established on the human reasoning, memory, past experience, and human interactions. In the SOA, artificial agents (seekers) are grouped into smaller social units called neighborhoods. At each generation of algorithm's execution, subpopulations (neighborhoods) learn from each other in the process called inter-subpopulation learning by exchanging individual problem solutions. SOA was applied to different kind of problems such as tuning neural networks [18], global optimization [19], or image thresholding [20] based on entropy [21], but not to the portfolio optimization problem.

Cuckoo search (CS) algorithm mimics the behavior of cuckoo birds in the nature. It was recently developed by Yang and Deb [22]. Search process is modeled by the Levy flights (series of straight flight paths with sudden 90 degrees turn with short and long steps). CS algorithm proved to be a robust technique for solving various numerical optimization problems, but it was not yet been applied to the portfolio optimization problem. Also, and object-oriented framework for CS algorithm was developed and tested on unconstrained benchmark problems [23]. There are also many other swarm intelligence algorithms which can be found in the literature [24].

In this paper, we present the firefly algorithm (FA) for portfolio optimization problem. The implementation of the FA for this problem was not found in the literature.

This paper begins with illustration of mathematical formulation of the portfolio optimization problem in Section 2. In this Section, we present different problem formulations that can be found in the literature. Section 3 introduces FA metaheuristic. Experimental data, problem setup and experimental results are presented in Section 4, while Section 5 concludes the paper.

II. Portfolio optimization problem

The fundamental guideline in making financial investments decisions is diversification where investors invest into different types of assets. Portfolio diversification minimizes investors' exposure to the risks while maximizing returns on portfolios.

Many methods can be applied to solving multi-objective optimization problems such is portfolio optimization. One essential method is to transform the multi-objective optimization problem into a single-objective optimization problem. This method can be further divided into two sub-types. In the first approach, one important objective function is selected for optimization, while the rest of objective functions are defined as constrained conditions. Alternatively, only one evaluation function is created by weighting the multiple objective functions [25].

The first method is defined by Markowitz and is called the standard mean-variance model [26]. It was first introduced more than 50 years ago and its basic assumptions are a rational investor with either multivariate normally distributed asset returns, or, in the case of arbitrary returns, a quadratic utility function [2]. If these assumptions hold, then the optimal portfolio for the investor lies on the mean-variance efficient frontier.

In this model, the selection of risky portfolio is considered as one objective function and the mean return on an asset is considered to be one of the constraints [10]. It can be formulated as follows:

$$min\,\sigma_{R_P}^2 = \sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\omega_i\omega_j Cov(\overline{R_i}\,\overline{R_j}) \qquad (1)$$

subject to

$$\overline{R_p} = E(R_p) = \sum_{i=1} \omega_i \overline{R_i} \geq R \tag{2}$$

$$\sum_{i=1}^{N} \omega_i = 1 \tag{3}$$

$$\omega_i \geq 0,\ \forall i \in (1,2,...,N) \tag{4}$$

where N is the number of available assets, $\overline{R_i}$ is the mean return on an asset i and $Cov(\overline{R_i}\overline{R_j})$ is covariance of returns of assets i and j respectively. Weight variable ω_i controls the proportion of the capital that is invested in asset i, and constraint in (3) ensures that the whole available capital is invested. In this model, the goal is to minimize the portfolio risk σ_p^2, for a given value of portfolio expected return $\overline{R_p}$.

In the presented standard mean-variance model, variables are real and they range between zero and one, as they represent the fraction of available money to invest in assets. This choice is quite straightforward, and has the advantage of being independent of the actual budget.

The second method refers to the construction of only one evaluation function that models portfolio selection problem. This method comprises two distinct models: efficient frontier and Sharpe ratio model [25].

In efficient frontier model, the goal is to find the different objective function values by varying desired mean return R. The best practice is to introduce new parameter $\lambda \in [0,1]$ which is called risk aversion indicator [25]. In this case, the model is approximated to only one objective function:

$$min\, \lambda[\sum_{i=1}^{N}\sum_{j=1}^{N} \omega_i \omega_j Cov(\overline{R_i}\overline{R_j})] - (1-\lambda)[\sum_{i=1}^{N} \omega_i \overline{R_i}] \tag{5}$$

subject to

$$\sum_{i=1}^{N} \omega_i = 1 \tag{6}$$

$$\omega_i \geq 0, \forall i \in (1,2,...N) \tag{7}$$

λ controls the relative importance of the mean return to the risk of the investor. When λ is zero, mean return of the portfolio is maximized regardless of the risk. Contrary, when λ equals 1, risk of the portfolio is being minimized regardless of the mean return. Thus, with the increase of λ, the relative importance of the risk to the investor increases, and importance of the mean return decreases, and vice-versa.

With the change of the value of λ, objective function value changes also. The reason for this change is that the objective function is composed of the mean return value and the variance (risk). The dependencies between changes of λ and the mean return and variance intersections are shown on a continuous curve which is called efficient frontier in the Markowitz theory [26]. Since each point on this curve indicates an optimum, portfolio optimization problem is considered as multi-objective, but λ transforms it into single-objective optimization task.

Sharpe ratio (SR) model combines the information from the mean and variance of an asset [27]. This simple model is risk-adjusted measure of mean return and can be described with the following equation [27]:

$$SR = \frac{R_p - R_f}{StdDev(p)}, \tag{8}$$

where p denotes portfolio, R_p is the mean return of the portfolio p, and R_f is a test available rate of return on a risk-free asset. *StdDev(p)* is a measure of the risk in portfolio (standard deviation of R_p). By adjusting the portfolio weights ω_i, portfolio's sharpe ratio can be maximized.

So far, we presented only the basic problem definitions. These definitions do not seem realistic because they do not consider several aspects, such as [28]:

- the existence of frictional aspects like the transaction costs, sectors with high capitalization and taxation;
- the existence of specific impositions arising from the legal, economic, etc. environment;
- finite divisibility of the assets to select.

As mentioned in the previous section, additional constraints can be introduced to make the problem more realistic. For example, budget, cardinality, transaction lots and sector capitalization constraints were successfully applied in solving portfolio optimization problem using particle swarm optimization (PSO) method [10]. The minimum transaction lots constraint assures that each asset can only be purchased in certain numbers of units. With transaction lots constraint applied, classical portfolio optimization problem becomes a combinatorial optimization problem whose feasible region is not continuous. Sector capitalization constraint refers to the fact that the investors tend to invest in the assets that belong to the sectors where higher value of market capitalization can be obtained. Investing in such way, risk is reduced. The significance of this constraint is discussed in [29].

Taking into account all above mentioned additional portfolio optimization constraints, new portfolio optimization problem can be established [10]. This model is called extended mean-variance model and it is classified as a quadratic mixed-integer programming model necessitating the use of efficient heuristics to find the solution. It can be formulated as follows:

$$min\sigma_{R_P}^2 = \sigma_p^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} \omega_i \omega_j Cov(\overline{R_i}\overline{R_j}) \tag{9}$$

where

$$\omega_i = \frac{x_i c_i z_i}{\sum_{j=1}^{N} x_j c_j z_j}, \; i = 1,...,N \tag{10}$$

$$\sum_{i=1}^{N} z_i = M \le N, M, N \in N, \forall i = 1,...N, z_i \in \{0,1\} \tag{11}$$

Subject to

$$\sum_{i=1}^{N} x_i c_i z_i \overline{R_i} \ge BR \tag{12}$$

$$\sum_{i=1}^{N} x_i c_i z_i \le B \tag{13}$$

$$0 \le B_{low_i} \le x_i c_i \le B_{up_i} \le B, i = 1,...N \tag{14}$$

$$\sum_{i_S} W_{i_S} \ge \sum_{i_{S'}} W_{i_{S'}}, \forall y_s y_{s'} \ne 0, s, s' \in \{1,...S\}, s < s' \tag{15}$$

where

$$y_s = \begin{cases} 1, \; if \sum_{i_S} z_i > 0 \\ 0, \; if \sum_{i_{S'}} z_i = 0 \end{cases} \tag{16}$$

where M represents the number of selected assets among possible N assets. B is the total available budget, while B_{low_i} and B_{up_i} are lower and upper limits respectively of the budget that can be invested in asset i. S is the total number of sectors. c_i represents the minimum transaction lot for asset i, and x_i denotes the number of c_i that is purchased. According to this, $x_i c_i$ are integer values that show the units of asset i in the portfolio.

Decision variable z_i is defined for modeling cardinality constraint. It is equal to 1 if an asset i is present in the portfolio, otherwise it is equal to 0. Equation (11) represents the cardinality constraint and inequality (12) is the same as (2). In order to make the search process easier, budget constraint in (13)

is converted to inequality. Equation (14) shows lower and upper bounds of budget constraint.

Sector capitalization constraint improves portfolio's structure decisions by preferring investments in assets that belong to the sector with higher capitalization value. The assets which belong to the sector with more capitalization should have more shares in the final portfolio. This constraint is held only if securities from the corresponding sectors are selected [10]. Equation (15) introduces sector capitalization constraint into extended mean-variance model. Despite the fact that a certain sector has high capitalization, security from this sector that has low return and/or high risk must be excluded from final portfolio's structure. To make such exclusion, variable y_s is defined and it has a value of 1 if the corresponding sector has at least one selected asset, and 0 otherwise. In (15) i_s is a set of assets which can be found in sector S. Sectors are sorted in descending order by their capitalization value. Sector 1 has the highest capitalization value, while sector S has the lowest value.

There are other constraints that can be found in the literature. One of them is 5-10-40 constraint which is based on the 60(1) of the German investment law [30]. This constraint defines upper limit of each individual asset and for the sum of all "heavyweights" in the portfolio. It actually means that the securities of the same issuer are allowed to the amount of 5% of the net asset value of the mutual fund [30]. They are allowed to amount to 10%, however, if the total share of all assets with a share between 5% and 10% is less than 40% of the net asset value [30].

III. IMPLEMENTATION OF THE FA

FA is one of the latest swarm intelligence metaheuristics. It is inspired by the flashing behavior of fireflies. The main algorithm's principle is that each firefly moves towards the brighter firefly. Firefly's flash is used as a signaling system for attracting other fireflies for mating, and also for attracting the potential prey. In addition, flashing may also be used as a protective warning mechanism. Female fireflies respond to a male's unique pattern of flashing in the same species, while in some species, female fireflies are able to mimic the mating flashing patterns of other species in order to lure and eat males who are deceived. FA was first proposed for unconstrained optimization problems [31].

Three simplification rules guide the construction of the FA:

- each firefly attracts all other fireflies with weaker flashes (firefly's sex is neglected);
- attractiveness of fireflies is proportional to their brightness, while, at the other side, the brightness is reverse proportional to its distance from the light source. If there is no brighter one than a particular firefly, it move in a random manner;
- the brightness of a firefly is determined, or at least affected, by the distribution of the objective function.

With the increase of the distance from the lighting source, the light intensity decreases. So, light intensity follows the inverse square law:

$$I(r) = \frac{I_0}{r^2} \tag{17}$$

where $I(r)$ is the light intensity, r is distance, and I_0 is the light intensity at the source. Besides that, the air also absorbs part of the light, and the light becomes weaker. Thus, the light absorption coefficient γ must be included in (17):

$$I(r) = \frac{I_0}{1+\gamma r^2} \tag{18}$$

As mentioned above, attractiveness β of a firefly is proportional to its brightness (light intensity), and this can be shown in the following expression:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{19}$$

where β_0 is the attractiveness at *r = 0*. Since the exponential function is hard to be calculated, and if there is a need to accelerate computation, the above expression can be replaced with the following:

$$\beta(r) = \frac{\beta_0}{1+\gamma r^2} \tag{20}$$

Objective function *f(x)* is used to encode the brightness of a given firefly. It represents the light intensity at location *x*, as *I(x) = f(x)*. For a maximization problem, the brightness can simply be proportional to the value of the objective function. For other problems, brightness can be defined similar to fitness function in evolutionary algorithm's approach.

Movement of a firefly (process of exploitation) is based upon attractiveness, and when firefly *j* is more attractive (brighter) than firefly *i*, firefly i is moving towards *j*:

$$x_i(t) = x_i(t) + \beta_0 r^{-\gamma r_{i,j}^2}(x_j - x_i) + \alpha(rand - 0.5) \tag{21}$$

where β_0 is attractiveness at *r = 0*, α is randomization parameter, *rand* is random number uniformly distributed between 0 and 1, and $r_{i,j}$ is distance between fireflies *i* and *j*. This distance is calculated using Cartesian distance form:

$$r_{i,j} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})}, \tag{22}$$

where *D* is the number of problem parameters. For most problems, $\beta_0 = 0$, and $\alpha \in [0,1]$ are adequate settings. The parameter γ has crucial impact on determining the convergence speed of the algorithm. This parameter shows the variation of attractiveness and in theory it has a value of $[0,+\infty)$. But, in most applications, parameter varies between 0.01 and 100.

FA pseudo-code is shown below. Some details are omitted for simplicity.

```
Generate initial population of fireflies xi,
(i = 1, 2, 3, ..., FN)
Light intesity Ii at point xi is defined by
f(x)
Define light absorption coefficient γ
Define number of iterations IN
while (t < IN) do
   for (i = 1 to FN) do
      for (j = 1 to i) do
         if (Ij < Ii) then
            Move firefly j towards firefly i
in d dimension
            Evaluate new solution, replace
worst with better solution, and update light
intensity
         end if
      end for
   end for
   Rank all fireflies, find the current best,
and move them randomly
end while
```

In the pseudo-code above, *FN* is total number of fireflies in the population, *IN* is total number of algorithm's iterations, and *t* is the current iteration.

IV. PROBLEM FORMULATION, DATA AND RESULTS

In this section, we present portfolio optimization problem formulation used in testing FA approach, data used in the experiments and experimental results. We used the same problem formulation and data set as in [32].

A. Problem definition

The goal is to select weights of the each asset in the portfolio in order to maximize the portfolio's return and to minimize the portfolio's risk. We transformed multi-objective problem into single one with constraints.

The expected return of each individual security I is presented as follows:

$$E(\omega_i) = w_i r_i, \tag{23}$$

where ω_i denotes the weight of individual asset *i*, and r_i is the expected return of *i*. Total expected return of the portfolio *P* can be formulated as follows:

$$E(P) = \sum_{i=1}^{n} E(\omega_i), \tag{24}$$

where *n* is the number of securities in the portfolio *P*.

In our problem formulation, first goal is to maximize portfolio's expected return, and thus, the expression shown in (24) is objective function for the portfolio's return and it should be maximized.

The objective function of the portfolio variance (risk) is presented as a polynomial of second degree:

$$\sigma^2(P) = \sigma^2(\omega_i) = \sum_{i=1}^{n}(\omega_i^2 \sigma^2(r_i)) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} 2\omega_i \omega_j Cov(r_i, r_j), \tag{25}$$

where $\sigma^2(\omega_i)$ is variance of asset i, and $Cov(r_i, r_j)$ is covariance between securities i and j.

According to (24) and (25), the multiobjective function to be minimized is illustrated as:

$$H(P) = E(P) - \sigma^2(P) \tag{26}$$

Alternatively, considering individual asset *i*, not the whole portfolio *P*, it can be formulated as:

$$H(\omega_i) = E(\omega_i) - \sigma^2(\omega_i) \tag{27}$$

Problem constraints are:

$$\sum_{i=1}^{n} \omega_i = 1 \tag{28}$$

$$\omega_i^{\min} \le \omega_i \le \omega_i^{\max} \tag{29}$$

and to reach the positive portfolio return, we used:

$$\sum_{i=1}^{n} r_i \omega_i \ge 0, \tag{30}$$

where $\omega_i^{\min}$ and $\omega_i^{\max}$ are minimum and maximum weights of asset *i* respectively.

B. Experimental data

For testing purposes, we used simple historical data set from [32]. Data set is shown in Table 1. The mean return on each asset and covariance matrix are given in Tables 2 and 3 respectively.

TABLE I
DATA SET FOR THE EXPERIMENTS

Year	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5
2007	-0.15	0.29	0.38	0.18	-0.10
2008	0.05	0.18	0.63	-0.12	0.15
2009	-0.43	0.24	0.46	0.42	0.15
2010	0.79	0.25	0.36	0.24	0.10
2011	0.32	0.17	-0.57	0.30	0.25

TABLE II
MEAN RETURNS FOR EACH ASSET

Stock 1	0.116
Stock 2	0.226
Stock 3	0.252
Stock 4	0.204
Stock 5	0.11

TABLE III
COVARIANCE MATRIX

	Stock 1	Stock 2	Stock 3	Stock 4	Stock 5
Stock 1	0.21728	-0.003376	-0.053492	-0.009264	0.01064
Stock 2	-0.003376	0.00253	0.008468	0.002376	-0.00456
Stock 3	-0.053492	0.008468	0.22247	-0.31128	-0.02392
Stock 4	-0.009264	0.002376	-0.031128	0.04068	0.00276
Stock 5	0.01064	-0.00456	-0.02392	0.00276	0.01675

C. Algorithm settings and experimental results

In this subsection, we present experimental results for testing FA for portfolio optimization problem. See subsection A for problem formulation. Tests were performed on Intel Core i7 3770K processor @4.2GHz with 8GB of RAM memory, Windows 8 x64 operating system and Visual Studio 2012 with .NET 4.5 Framework. Solution number *SN* was set to 40, while maximum iteration number *IN* was set to 6000, yielding total of 240,000 objective function evaluations (40*6000). The same number of objective function evaluations was used in [33]. The algorithm was tested on 30 independent runs, each starting with a different random number seed.

We also ran additional test where we wanted to see whether our algorithm could perform better if it used more function evaluations. For this additional test we set maximum iteration number *IN* to 8000 while keeping solution number *SN* on the previous value. In this way, we employed 320.000 function evaluations (40*8000).

Since we used a set of five portfolio's assets, dimension *D* of a problem is 5. Each firefly in the population is a 5-dimensional vector. In initialization phase, firefly *x* is created using the following:

$$x_i = \omega_i^{\min} + rand(0,1)(\omega_i^{\max} - \omega_i^{\min}), \tag{31}$$

where *rand(0, 1)* is a random number uniformly distributed between 0 and 1.

We also used constraint handling techniques to direct the search process towards the feasible region of the search space. Equality constraints decrease efficiency of the search process by making the feasible space very small compared to the entire search space. For improving the search process, the equality constraints can be replaced by inequality constraints using the following expression [34]:

$$|h(x)| - \varepsilon \le 0, \tag{32}$$

where $\varepsilon > 0$ is very small violation tolerance. The ε was dynamically adjusted according to the current algorithm's iteration:

$$\varepsilon(t+1) = \frac{\varepsilon(t)}{dec}, \tag{33}$$

where *t* is the current iteration, and *dec* is a value slightly larger than 1. When the value of ε reaches the predetermined threshold value, (33) is no longer applied.

Also, it should be noted that the search parameter α is being gradually decreased from its initial value during the algorithm's execution according to the following equation:

$$\alpha(t) = (1-(1-((10^{-4}/9)^{1/IN})))\alpha(t-1), \tag{34}$$

where t is the current iteration, and IN is overall iterations'number. Summary of FA parameter set is given in Table 4.

TABLE IV
FA PARAMETER SET

Parameter	Value
Number of fireflies (FN)	40
Number of iterations (IN)	6000
Initial value for randomization parameter α	0.5
Attractiveness at $r=0$ β_0	0.2
Absorption coefficient γ	1.0
Initial violation tolerance (ε)	1.0
Decrement (*dec*)	1.002
ω^{min}	0
ω^{max}	1

In experimental results, we show best, mean and worst results for objective function value, variance (risk) and average return of portfolios (Table 5). In Table 6, we show portfolio weights for the best and worst results.

TABLE V
EXPERIMENTAL RESULTS

	Best	Worst	Mean
Objective function	4.542	4.698	4.615
Variance	0.036	0.072	0.059
Return	0.218	0.198	0.205

TABLE VI
PORTFOLIO WEIGHTS FOR BEST AND WORST RESULTS

	ω_1	ω_2	ω_3	ω_4	ω_5
Best	0.056	0.432	0.361	0.072	0.079
Worst	0.042	0.198	0.319	0.262	0.179

According to the experiment results presented in Tables 5 and 6, FA for portfolio optimization performs similar like GA approach in [32]. In [32], three variants of GA were shown: single-point, two-point and arithmetic. Arithmetic variant performed significantly better than other two variants, and also better than the FA presented in this paper. But, on the other hand, FA showed better performance than single-point and two-point variants of the GA presented in [32]. We should note that the objective function values, which should be minimized, were compared. GA experimental results for all three variants are shown in Table 7.

TABLE VII
GA EXPERIMENTAL RESULTS

Objective function	Variance	Return
Single-point variant		
4.900	0.019	0.204
Two-point variant		
4.598	0.080	0.221
Arithmetic variant		
4.532	0.0325	0.222

We also wanted to see how our algorithm performs when the number of function evaluations is slightly greater. So, we ran additional test, but this time, we set the number of iterations (*IN*) to 8000, while the number of firefly agents (*FN*) remained the same as in the first experiment. This parameter set gives 320.000 (40*8000) function evaluations which is 33.3 % higher than in the first experiment. The results are shown in the tables below.

TABLE VIII
EXPERIMENTAL RESULTS WITH 320.000 EVALUATIONS

	Best	Worst	Mean
Objective function	4.528	4.662	4.593
Variance	0.032	0.064	0.051
Return	0.231	0.208	0.217

TABLE IX
PORTFOLIO WEIGHTS FOR BEST AND WORST RESULTS IN 320.000 EVALUATIONS TEST

	ω_1	ω_2	ω_3	ω_4	ω_5
Best	0.069	0.453	0.352	0.042	0.084
Worst	0.054	0.211	0.303	0.215	0.217

As can be seen from Table 8, with higher number of function evaluations, our FA algorithm performs better than arithmetic variant of GA presented in [32] (see Table 7).

But, on the other hand, if we compare FA results obtained with 240.000 and 320.000 evaluations, only slight improvement can be noticed. Bests are improved by 0.3% (4.542/4.528), worsts by 0.7% (4.698/4.662) and means by only 0.4% (4.615/4.593). If we compare those figures with the increase of 33.3% in function evaluations, we conclude that this is definitely bad trade-off.

V. CONCLUSION

In this paper, FA for portfolio optimization problem was presented. The algorithm was tested on a set of five assets, like GA in [32]. The results of the investigation reported in this paper show that the FA metaheuristics has potential for solving this problem.

Two experiments were conducted with different number of function evaluations. In the first experiment (240.000 evaluations), FA performed better than single-point and two-point variant of GA, while the arithmetic variant of GA outperformed our FA approach. In the second experiment (320.000 evaluations), FA outscored all three GA variants. But, final conclusion is that the increase of evaluation numbers only slightly improves FA behavior.

FA was applied only to the basic portfolio optimization problem definition. There is a large potential for applying metaheuristic techniques to this class of problems, because they appear not to be investigated enough. In the subsequent work, original, as well as the modified version of the FA will be applied to the extended-mean variance, and other portfolio optimization models. Also, other swarm intelligence metaheuristics will be applied to various portfolio optimization problem models and definitions.

Acknowledgment

This research is supported by Ministry of Education and Science of Republic of Serbia, Grant No. III-44006.

References

[1] F. J. Fabozzi, P. N. Kolm, D. Pachamanova and S. M. Focardi, "Robust Portfolio Optimization and Management Studies in Computational Intelligence", *Wiley Finance*, 2007.

[2] K. P. Anagnostopoulos and G. Mamanis, "Multiobjective evolutionary algorithms for complex portfolio optimization problems," *Computational Management Science*, vol. 8, 2011, pp. 259–279.

[3] G. di Tollo and A. Roli, "Metaheuristics for the portfolio selection problem," *International Journal of Operations Research*, vol. 15, no. 1, 2008, pp. 13–35.

[4] V. K. Chopra and W. T. Ziemba, "The effect of errors in means, variances and covariances on optimal portfolio choice," *Journal of Portfolio Management*, vol. 19, no. 2, 1993, pp. 6–11.

[5] R.Mansini and G. Speranza, "An exact approach for portfolio selection with transaction costs and rounds," *IEEE Transactions*, vol. 37, no. 10, 2005, pp. 919–929.

[6] X.-L. Wu and Y.-K. Liu, "Optimizing fuzzy portfolio selection problems by parametric quadratic programming," F*uzzy Optimization and Decision Making*, vol. 11, 2012, pp. 411–449.

[7] Y. Lan, X. Lv, andW. Zhang, "A linear programming model of fuzzy portfolio selection problem," *IEEE International Conference on Control and Automation*, 2007, pp. 3116–3118.

[8] M. W. Cooper and K. Farhangian, "An integer programming algorithm for portfolio selection with fixed charges," *Naval Research Logistics Quarterly*, vol. 29, 2006, pp. 147–150.

[9] P.-C. Lin, "Portfolio optimization and risk measurement based on non-dominated sorting genetic algorithm," J*ournal of Industrial and Management Optimization (JIMO)*, vol. 8, 2012, pp. 549–564.

[10] H. R. Golmakani and M. Fazel, "Constrained portfolio selection using particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 1, 2011, pp. 8327–8335.

[11] R. Jovanovic and M. Tuba, "An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem", Applied Soft Computing, Vol. 11, Issue 8, Dec 2011, pp. 5360–5366, DOI: 10.1016/j.asoc.2011.05.023

[12] R. Jovanovic, M. Tuba, and D. Simian, "Comparison of different topologies for island-based multi-colony ant algorithms for the minimum weight vertex cover problem," *WSEAS Transactions on Computers*, vol. 9, no. 1, 2010, pp. 83–92.

[13] R. Jovanovic and M. Tuba, "Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem", Computer Science and Information Systems (ComSIS), Vol. 10, No. 1, Jan 2013, pp. 133–149, DOI: 10.2298/CSIS110927038J

[14] M. Tuba and R. Jovanovic, "Improved ant colony optimization algorithm with pheromone correction strategy for the traveling salesman problem", International Journal of Computers, Communications & Control, Vol. 8, Issue 3, Jun 2013, pp. 477–485.

[15] N. Bacanin, M. Tuba, and I. Brajevic, "Performance of object-oriented software system for improved artificial bee colony optimization", *International Journal of Mathematics and Computers in Simulation*, vol. 5, issue 2, 2011, pp. 154-162

[16] I. Brajevic and M. Tuba, "An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems", Journal of Intelligent Manufacturing, Vol. 24, No. 4, Aug 2013, pp. 729-740, DOI: 10.1007/s10845-011-0621-6

[17] M. Tuba, N. Bacanin and N. Stanarevic, "Adjusted artificial bee colony (ABC) algorithm for engineering problems", *WSEAS Transaction on Computers*, volume 11, issue 4, 2012, pp. 111-120.

[18] C. Dai, W. Chen, Y. Zhu, Z. Jiang, and Z. You, "Seeker optimization algorithm for tuning the structure and parameters of neural networks," *Neurocomputing*, vol. 74, no. 6, 2011, pp. 876–883.

[19] M. Tuba, I. Brajevic, and R. Jovanovic, "Hybrid seeker optimization algorithm for global optimization", Applied Mathematics and Information Sciences, Vol. 7, No. 3, May 2013, pp. 867–875, DOI: 10.12785/amis/070304

[20] M. Tuba, I. Brajevic, "Modified seeker optimization algorithm for image segmentation by multilevel thresholding", International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 7, No. 4, 2013, pp. 370-378

[21] M. Tuba, "Asymptotic Behavior of the Maximum Entropy Routing in Computer Networks", Entropy, Vol. 15, Issue 1, 2013, pp. 361-371, DOI :10.3390/e15010361

[22] X.-S. Yang and S. Deb, "Cuckoo search via levy flights," *in Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, 2009, pp. 210–214.

[23] N. Bacanin, "Implementation and performance of an object-oriented software system for cuckoo search algorithm*," International Journal of Mathematics and Computers in Simulation*, vol. 6, 2010, pp. 185–193.

[24] M. Tuba, N. Bacanin and N. Stanarevic: Multithreaded implementation and performance of a modified artificial fish swarm algorithm for unconstrained optimization, *International Journal of Mathematics and Computers in Simulation*, vol. 7, no. 3, 2013, pp. 215-222

[25] H. Zhu, Y. Wang, K. Wang, and Y. Chen, "Particle swarm optimization (pso) for the constrained portfolio optimization problem," *Expert Systems with Applications*, vol. 38, no. 1, 2011, pp. 10161–10169.

[26] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, 1952, pp. 77–91.

[27] W. F. Sharpe, "Mutual fund performance," *The Journal of Business*, vol. 39, 1966, pp. 119–138.

[28] M. Corazza and D. Favaretto, "On the existence of solutions to the quadratic mixedinteger meanvariance portfolio selection problem," *European Journal of Operational Research*, vol. 176, 2007, pp. 1947-1960.

[29] H. Soleimani, H. R. Golmakani, and M. H. Salimi, "Markowitz-based portfolio selection with minimum transaction lots, cardinality constraints and regarding sector capitalization using genetic algorithm," *Expert Systems with Applications*, vol. 36, no. 3, 2009, pp. 5058–5063.

[30] J. Brankea, B. Scheckenbacha, M. Steina, K. Deb, and H. Schmecka, "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm," *European Journal of Operational Research*, vol. 199, 2009, pp. 684–693.

[31] X.-S. Yang, "Firefly algorithms for multimodal optimization," Stochastic Algorithms: Foundations and Applications, *LNCS*, vol. 5792, 2009, pp. 169–178.

[32] S. Sefiane and M. Benbouziane, "Portfolio selection using genetic algorithm," *Journal of Applied Finance & Banking*, vol. 4, no. 2, 2012, pp. 143–154.

[33] N. Bacanin and M. Tuba, "Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators", Studies in Informatics and Control, Vol. 21, Issue 2, Jun 2012, pp. 137–146.

[34] E. Mezura-Montes, C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, 2011, pp. 173–194.

Milan Tuba received B.S. in mathematics, M.S. in mathematics, M.S. in computer Science, M.Ph. in computer science, Ph.D. in computer science from University of Belgrade and New York University.

From 1983 to 1994 he was in the U.S.A. first as a graduate student and teaching and research assistant at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as an assistant professor of electrical engineering at Cooper Union Graduate School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and theses supervisor. From 1994 he was associate professor of computer science and Director of Computer Center at University of Belgrade, Faculty of Mathematics, and from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. He was teaching more than 20 graduate and undergraduate courses, from VLSI design and Computer architecture to Computer networks, Operating systems, Image processing, Calculus and Queuing theory. His research interest includes mathematical, queuing theory and heuristic optimizations applied to computer networks, image processing and combinatorial problems. He is the author of more than 100 scientific papers and a monograph. He is coeditor or member of the editorial board or scientific committee of number of scientific journals and conferences.

Prof. Tuba is member of the ACM since 1983, IEEE 1984, New York Academy of Sciences 1987, AMS 1995, SIAM 2009, IFNA. He participated in many WSEAS Conferences with plenary lectures and articles in Proceedings and Transactions.

Nebojsa Bacanin received B.S. and M.S. in economics and computer science in 2006 and 2008 from Megatrend University of Belgrade and also M.S. in computer science in 2008 from University of Belgrade

He is currently Ph.D. student at Faculty of Mathematics, Computer science department, University of Belgrade and works as teaching assistant at Faculty of Computer Science, Megatrend University of Belgrade. He is the coauthor of ten papers. His current research interest includes nature inspired metaheuristics.

Mr. Bacanin participated in WSEAS conferences**.**

Branislav Pelevic was born in Belgrade in 1955 where he received B.S. from Faculty of Economics, University of Belgrade, and received his master's and PhD degrees from the same university. While working on his PhD thesis, in 1991-1992 he was at Florida State University, and London School of Economics. At the Faculty of Economics he held all teaching positions from a teaching assistant to full-time professor. He was also editor-in-chief of the 'Economic Annals' journal, and held the position of vice Dean for Science and of the Dean. Since 2008 he has been working at Megatrend University where he is currently at the position of Dean of Faculty of International Economics. The fields of his scientific research are International economics, and theory and policy of international commerce in particular. He teaches Microeconomics, International commerce and International economic relations at undergraduate studies, and International economics at PhD studies. Prof. Pelevic is member of Scientific Society of Economists.