Constrained portfolio selection using artificial bee colony (ABC) algorithm

Nebojsa Bacanin, Milan Tuba, Branislav Pelevic

***Abstract*—**Well-known and important problem of portfolio selection (optimization) in economics, finance and management was partially solved using traditional methods and techniques. However, since the problem is intractable, nondeterministic optimization metaheuristics are better suited tools for such problems. This paper presents artificial bee colony (ABC) swarm intelligence metaheuristic for solving constrained portfolio optimization problem. To prove the algorithm's robustness and efficiency for this application, series of tests on standard benchmark portfolio data used in the literature were performed. The algorithm was compared to the genetic algorithm (GA) approach and firefly algorithm (FA) swarm intelligence metaheuristic, which were tested on the same data set. Optimization results showed that the ABC algorithm obtains satisfying results.

***Keywords*—**Portfolio optimization problem, metaheuristic optimization, swarm intelligence, artificial bee colony (ABC), nature inspired algorithms.

I. INTRODUCTION

PORTFOLIO optimization problem, which is sometimes referred to as portfolio selection problem, is a well-known problem in economy, management and finance. Portfolio includes various financial securities, such as bonds and stocks owned by an organization or by individual [1].

One of the main issues when dealing with portfolio optimization is risk. Investors are always trying to balance between portfolio's gains and risk. Thus, the goal is to select a portfolio with minimum risk at defined minimal expected returns. This further means reducing non-systematic risks to zero.

Portfolio optimization problem is a multi-criteria optimization problem where the goal is to minimize risks, while maximizing returns. Unfortunately, this problem approach has several shortcomings [2]. First, it can be quite difficult to gather enough data for risk and returns evaluation. Second, this model is too simple for modeling real-world problem features. It does not capture all properties such as transaction costs, cost of management, etc. Third, the estimation of return and covariance from historical data is very prone to measurement errors. Covariance matrix is used for defining the risk.

Portfolio optimization problem is being solved using variety of methods and techniques. Parametric quadratic programming technique [3], linear programming method [4] and integer programming [5] were successfully applied to solving fuzzy portfolio selection problem.

With the application of additional real-world constraints on the basic portfolio optimization formulation, the problem becomes harder for solving. In this case, traditional techniques and methods cannot generate satisfying results, and the use of heuristic and metaheuristic approaches is more promising. Those algorithms have relatively low computational complexity (typical polynomial) and do not guarantee that the optimal solution will be retrieved. Heuristics improve algorithm`s performance by shortening execution time at the cost of accuracy [6]. There are two basic types of heuristic methods: constructive and local search heuristics. Constructive approaches build solution "from the scratch" in a step-by-step manner. At the other side, local search heuristics selects random complete solution from the population of potential solutions. Then, the chosen solution is being incrementally improved during algorithm`s execution.

A metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space. Thus, metaheuristics search for a good heuristics of a particular problem. Learning strategies are used to structure information in order to find efficiently near optimal solutions. Key point in metaheuristics is that they do not guarantee to find the optimal solution, but the satisfying solution in a reasonable amount of execution time.

Some of the nature-inspired metaheuristics were used for solving portfolio optimization problem. Bio-inspired algorithms mimic the behavior of natural systems and can be roughly divided into two groups: evolutionary algorithms (EA) and swarm intelligence. Well-known representative of EA, genetic algorithm (GA) employs selection, crossover and mutation operators while performing the exploration and exploitation of the search space. GA was applied on portfolio selection problem [7]. In [8], a combination encoding scheme and genetic operators are designed for solving combination

Manuscript received December 29, 2013, revised April 28, 2014, accepted May 03, 2014.

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

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

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

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

optimization problems. The authors applied this combination GA to the portfolio optimization problem which can be reformulated approximately as a combination optimization problem [8]. Besides GA, the representatives of EA include evolutionary programming (EP), evolution strategies (ES) and genetic programming (GP).

Swarm intelligence algorithms use the effects of the fact that behavior of many individuals show extraordinary collective intelligence without employing any centralized supervision component. They belong to the group of population-based optimizations and start with the initial (usually random) population of candidate problem solutions and improve them in iterations.

Particle swarm optimization (PSO) is a swarm intelligence algorithm which emulates social behavior of school of fish or flock of birds. It was successfully applied to variations of constrained portfolio optimization problems [9], [10], [11]. Ant colony optimization (ACO) simulates the foraging behavior of ants. Properties of ants system that ant deposit a substance called pheromone between food source and its nests was captured and incorporated into the algorithm. ACO implementation for solving portfolio problem was not found in the literature, but ACO was successfully applied on many hard optimization problems [12], [13], [14], [15], [16].

Seeker optimization algorithm (SOA) is based on human search process which uses human reasoning, memory, past experience and human interactions. Seeker operates in the larger environment of candidate solutions called search population. The total population is divided into three equally-sized subpopulations according to the sequence of the seekers. All the agents in the same population form a social unit called neighborhood, and each population performs search in its domain of the search space [17]. Although SOA was not applied on portfolio selection problem, it was applied on a variety of other optimization problems such is global optimization [17], [18].

Cuckoo search (CS) algorithm models search process by using the Levy flights (series of straight flight paths with sudden 90 degrees turn with short and long steps). This algorithm was developed by Yang and Deb [19] and was tested on different optimization problems [20]. Firefly algorithm (FA) is one of the latest swarm intelligence metaheuristics. It models the flashing behavior of fireflies and the basic algorithm's principle is that each firefly moves toward the brighter one. Using this philosophy, FA performs the search process. FA was first proposed for unconstrained numerical optimization [21], applied to image processing [22] with entropy function [23], but it was also tested on portfolio selection problem [24], [25]. Bat algorithm is the latest SI algorithm [26].

In this paper, we present implementation of artificial bee colony (ABC) algorithm for solving constrained portfolio optimization problem. The constrained ABC implementation was adopted and the tests were performed on a portfolio of a five assets. This swarm intelligence metaheuristics have not yet been applied in solving this kind of problem, but obtained satisfying results in various numerical optimization problems [27], [28], [29], [30], [31]. Side-by-side comparison was made with one GA approach and FA which were tested on a same data set. Empirical tests showed that the ABC is a promising method for solving this kind of problem.

This paper is organized as follows. After Introduction, overviews of portfolio optimization problem models are given in Section 2. Section 3 describes constrained version of the ABC metaheuristic. In Section 4, data sets, experimental results and comparisons are presented, while Section 5 gives final conclusions and remarks.

II. Models and Formulations for Portfolio Optimization Problem

Portfolio optimization is the process of choosing the proportions (weights) of various assets to be held in a portfolio, in such a way as to make the portfolio better than any other according to some criterion (constraint). The criterion will mix together, directly or indirectly, considerations of the expected value of the portfolio's rate of return as well as of the return's dispersion and possibly other measures of financial risk.

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.

There are two basic methods for solving portfolio optimization problem. The first one named standard mean-variance mode was defined by Markowitz [32], and it is funded on a assumptions of a rational investor with either multivariate normally distributed asset returns, or, in the case of arbitrary returns, a quadratic utility function [33]. 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]. This model is 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

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

$$\sum_{i=1}^{N}\omega_i = 1 \qquad (3)$$

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

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 Eq. (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.

A second method is applied by construction of only one evaluation function which models the whole problem formulation. This model consists of two submodels: efficient frontier and Sharpe ratio [9].

When employing efficient frontier model, the aim is to find different objective function values by varying the desired mean return R. Many practical approaches include risk aversion parameter $\lambda \in [0,1]$ and use the following equation in model formulation:

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

Subject to

$$\sum_{i=1}^{N}\omega_i = 1 \quad (6)$$

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

λ parameter controls the relative importance of the mean return to the risk. 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. he 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 [32].

Sharpe ration (SR) model uses information from mean and variance of an asset [34]:

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

where *p* denotes portfolio, *Rp* 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.

Besides basic portfolio optimization problem formulations, there are also other definitions which take into account other factors which make model more realistic. This refers to [35]:

- 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;
- the finite divisibility of the assets to select.

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:

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

where

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

$$\sum_{i=1}^{N} z_i = M \leq N, M, N \in N, \forall i = 1,...N, z_i \in \{0,1\} \quad (11)$$

Subject to

$$\sum_{i=1}^{N} x_i c_i z_i \overline{R_i} \geq BR \quad (12)$$

$$\sum_{i=1}^{N} x_i c_i z_i \leq B \quad (13)$$

$$0 \leq B_{low_i} \leq x_i c_i \leq B_{up_i} \leq B, i = 1,...N \quad (14)$$

$$\sum_{i_S} W_{i_S} \geq \sum_{i_{S'}} W_{i_{S'}}, \forall y_s y_{s'} \neq 0, s, s' \in \{1,...S\}, s < s' \quad (15)$$

where

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

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 used for cardinality constraint. If an asset i is present in the portfolio, the value of z_i is 1, otherwise it is equal to 0. Equation (11) models cardinality constraint.

Budget constraint is shown in Equation (13), and it is converted to inequality. Lower and upper bounds of budget constraint are given in Equation (14).

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 [24]. Equation (15) establishes 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 (16), 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.

III. Constrained ABC Metaheuristic

ABC is well-known population based swarm intelligence metaheuristic. It is inspired by the foraging behavior of honey bee swarms in nature. This approach firstly proposed by Karaboga [36], and lately developed by the Karaboga and Basturk [37], [38].

An important difference between the ABC and other swarm intelligence algorithms is that in the ABC algorithm the possible solutions represent as food sources (flowers), not individuals (honeybees). In other algorithms, like PSO, each possible solution represents an individual of the swarm. In the ABC algorithm the quality of solution is represented as fitness of a food source. Fitness is calculated by using objective function of the problem.

In ABC metaheuristic, there are three types of artificial bees (agents): employed, onlookers and scouts. Half of the colony is employed bees. The relation between employed bee and the food source is one-to-one, and that means that there is only one employed bee per each food source. If a food source becomes abandoned, employed bee that is mapped to that food source becomes a scout, and as soon as scout finds a new food source, it again becomes employed bee. In the ABC algorithm onlookers and employed bees carry out the exploitation process in the search space, while the scouts control the exploration process.

In the case of honey bees, the basic properties on which self-organization relies are as follows:

- positive feedback: As the nectar amount of food sources increases, the number of onlookers visiting them increases, too;
- negative feedback: The exploration process of a food source abandoned by bees is stopped.
- fluctuations: The scouts carry out a random search process for discovering new food sources.
- multiple interactions: Bees share their information about food source positions with their nest mates on the dance area.

ABC algorithm, as an iterative algorithm, starts by associating each employed bee with randomly generated food source (solution). Each solution x_i $(i = 1, 2, \ldots, SN)$ is a D-dimensional vector, where SN denotes the size of the population. Initial population of randomly generated solution is created using:

$$x_{i,j} = lb_j + rand(0,1) \cdot (ub_j - lb_j) \tag{17}$$

In each iteration, each employed bee discovers a food source in its neighborhood, and evaluates its nectar amount (fitness). Discovery of a new, neighborhood solution is modeled with the following expression:

$$v_{i,j} = \begin{Bmatrix} x_{i,j} + \phi \cdot (x_{i,j} - x_{k,j}), R_j < MR \\ x_{i,j}, otherwise \end{Bmatrix} \tag{18}$$

where $x_{i,j}$ is j-th parameter of the old solution i, $x_{k,j}$ is j-th parameter of a neighbor solution k, ϕ is a random number between 0 and 1, and MR is modification rate. MR is ABC control parameter.

Pseudo-code of the ABC algorithm for constrained optimization problems [39] is:

```
1. Initialize the population of solutions

2. Evaluate the population

3. cycle=1

4. repeat

5. Produce new solutions for the employed bees
   by using Equation (18) and evaluate them

6. Apply selection process based on Deb's
   method [40].

7. Calculate the probability values pi for the
   solutions xi, using fitness of the solutions
   and the constraint violations (CV) by
```

$$CV = \sum_{g_j > 0} g_j(x) + \sum_{q+1}^{m} h_j(x) \tag{19}$$

$$pi = \begin{cases} \left(0.5 + \dfrac{fitness_i}{\sum_{i=1}^{SN} fitness_i}\right) * 0.5 \ \ if \ \ solution \ is \ \ feasible \\ \left(1 - \dfrac{CV}{\sum_{i=1}^{SN} CV}\right) * 0.5 \ \ if \ \ solution \ is \ infeasible \end{cases} \quad (20)$$

```
8. For each onlooker bee, produce a new
   solution vij by Equation (18) in the
   neighborhood of the solution selected
   depending on pi and evaluate it

9. Apply selection process between υi and xi
   based on Deb's method [40].

10. Determine the abandoned solutions by using
    "limit" parameter for the scout; if they
    exist, replace them with new randomly
    produced solutions by (17).

11. Memorize the best solution achieved so far

12. cycle = cycle+1

13. until cycle = MCN
```

We also note that the fitness is in the case of minimization calculated using:

$$fitness_i = \begin{cases} \dfrac{1}{objFun_i}, & if \ objFun_i > 0 \\ 1 + |\, objFun_i \,| \ , \ otherwise \end{cases}, \quad (21)$$

where $objFun_i$ is value of objective function which is the subject of optimization.

IV. PROBLEM FORMULATION, DATA AND RESULTS

In this section, we present data used in the experiments, portfolio optimization problem formulation used in testing ABC approach, and experimental results. We used the same problem formulation and data set like in [24] and [41]. We compared the ABC approach to the GA [41] and FA [24]. MATLAB software was used for GA implementation, while FA has its own framework developed in C# using .NET 4.5 Framework and Visual Studio 2012 working environment.

A. Data set for the experiments

As mentioned above, for testing purposes, we used simple historical data set like in [24] and [41]. The data encompasses historical return of a five stocks portfolio of a period of five years (2007-2011). Data set is shown in Table 1.

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

The mean return on each asset and covariance matrix is given in Tables 2 and 3 respectively.

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

B. Problem formulation

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:

$$E(\omega_i) = w_i r_i \,, \quad (22)$$

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:

$$E(P) = \sum_{i=1}^{n} E(\omega_i), \quad (23)$$

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 (23) 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:

$$\sigma^2(P) = \sigma^2(\omega_i) = \sum_{i=1}^{n} (\omega_i^2 \sigma^2(r_i)) + \sum_{i=1}^{n} \sum_{j=i+1}^{n} 2\omega_i \omega_j Cov(r_i, r_j), \quad (24)$$

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

$$H(P) = E(P) - \sigma^2(P) \quad (25)$$

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

$$H(\omega_i) = E(\omega_i) - \sigma^2(\omega_i) \quad (26)$$

Problem constraints are:

$$\sum_{i=1}^{n} \omega_i = 1 \quad (27)$$

$$\omega_i^{\min} \le \omega_i \le \omega_i^{\max} \quad (28)$$

To satisfy condition that the positive portfolio's return should be reached, we used:

$$\sum_{i=1}^{n} r_i \omega_i \ge 0, \quad (29)$$

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

C. ABC parameters setup

In this subsection, we present experimental results for testing ABC metaheuristics for portfolio optimization problem. (see Subsection A for problem formulation). All tests were performed on Intel Core 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, and maximum cycle number number *MCN* was set to 6000, yielding total of 240,000 objective function evaluations (40*6000). The same number of objective function evaluations was used in ABC's approach for constrained optimization presented in [27], and in the FA for constrained portfolio optimization in [24].

Limit parameter is calculated using:

$$limit = \frac{MCN}{SN} \quad (30)$$

Thus, in this case, limit is set to 150 (6000/40). According to ABC experimental studies, limit calculated with (30) established optimal balance between explotation and exploration [39].

The algorithm was tested on 30 independent runs, each starting with a different random number seed.

Since the portfolio of a five stocks is used, dimension *D* of a problem is 5. Each food source in the population is a 5-dimensional vector. In initialization phase, food source *x* is generated using the following expression:

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

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

Constraint handling techniques were used 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 [42]:

$$|h(x)| - \varepsilon \le 0, \quad (32)$$

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

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

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, expression (33) is no longer applied.

Summary of all ABC's parameters is given in Table 4.

TABLE IV
FA PARAMETER SET

Parameter	Value
Number of food sources (SN)	40
Number of cycles (MCN)	6000
Modification rate (MR)	0.8
Limit	150
Initial violation tolerance (ε)	1.0
Decrement (*dec*)	1.002
ω^{min}	0
ω^{max}	1

We also ran additional test where we wanted to see whether our algorithm could perform better if it used more function evaluations and to make better comparison with the FA [24]. For this additional test we set maximum cycle number *MCN* to 8000 while keeping solution number *SN* on the previous value. In this way, we employed 320,000 function evaluations (40*8000). In this test, limit is set to 200 (8000/40).

D. Experimental results and comparisons

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	4.529	4.656	4.608
Variance	0.027	0.059	0.037
Return	0.234	0.209	0.218

TABLE VI
PORTFOLIO WEIGHTS FOR BEST AND WORST RESULTS

	ω_1	ω_2	ω_3	ω_4	ω_5
Best	0.021	0.505	0.310	0.065	0.099
Worst	0.072	0.210	0.320	0.195	0.203

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 cycles (*MCN*) to 8,000, while the number of food sources (*SN*) 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 VII
EXPERIMENTAL RESULTS WITH 320.000 EVALUATIONS

	Best	Worst	Mean
Objective function	4.513	4.612	4.587
Variance	0.019	0.049	0.029
Return	0.245	0.217	0.231

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

	ω_1	ω_2	ω_3	ω_4	ω_5
Best	0.032	0.569	0.355	0.031	0.013
Worst	0.052	0.241	0.205	0.221	0.281

If we compare ABC results obtained with 240,000 and 320,000 evaluations, only slight improvement can be noticed. Bests are improved by 0.3% (4.529/4.513), worsts by 0.9% (4.656/4.612) and means by only 0.4% (4.608/4.587). If we compare those figures with the increase of 33.3% in function evaluations, we conclude that this is a bad trade-off.

According to the experiment results presented in Tables 5 and 6, ABC for constraint portfolio optimization performs similar like GA approach. Three variants of GA were shown [41]: single-point, two-point and arithmetic. Arithmetic variant performed significantly better than other two variants, but not better than the ABC approach presented in this paper, even with 240,000 function evaluations. At the other hand, ABC completely outperformed single-point and two-point variants of the GA. 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 9.

TABLE IX
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

As can be seen from Table 7, with higher number of function evaluations, our ABC algorithm performs far better than all three variants of the GA presented in [41].

FA metaheuristic was also tested on 240,000 and 320,000 function evaluations [24]. Results for 240,000 and 320,000 evaluations are given in Tables 10 and 11 respectively.

TABLE X
FA EXPERIMENTAL RESULTS WITH 240.000 EVALUATIONS

	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 XI
FA 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

Also, in FA approach, a bad trade-off when more evaluations are used can be noticed [24]. 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) [24].We conclude that similar trade-off exist in the ABC and FA approach when considering results enhancements and the increase of number of evaluations.

If we compare ABC and FA metaheuristcs for constrained portfolio optimization problem, we conclude that the ABC is noticeably better. If we compare test with 240,000 function evaluations, ABC's superiority is evident. All three, best, mean and worst results are better. In 320,000 function evaluation tests, ABC also performs much better than the FA.

Even if we compare ABC's results with 240,000 and FA's results with 320,000 evaluations tests, worst results that ABC obtains are better than worsts in FA implementation (4.656:4.662). Best and mean results are better in the FA, but with 33% more function evaluations.

As a final conclusion, we state that the ABC algorithm has significantly better performance than GA nature-inspired algorithm [41] and the FA swarm intelligence metaheuristic when tackling the constrained portfolio optimization problem.

V. CONCLUSION

In this paper, ABC algorithm for constrained portfolio optimization problem was presented. The implementation of the ABC for this problem was not found in the literature. The algorithm was tested on a set of five stocks portfolio.

The results of the investigation reported in this paper show that the ABC swarm intelligence metaheuristic has potential for solving this problem.

Two experiments were conducted with different number of function evaluations. In the first experiment (240,000 evaluations), ABC performed better than all three variants of GA: single-point crossover, two-point crossover, and arithmetic version. Also, in this test, ABC completely outscored FA.

In the second experiment (320,000 evaluations), the performance difference between ABC and GA is more expressed. Also ABC outperformed FA with the same number of evaluations.

It is interesting to point out that the ABC with 240,000 evaluations obtained better worst results than the FA with 320,000 evaluations.

ABC 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 ABC 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.

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

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

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

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

[6] Z. Michalewicz and B. D. Fogel, “How to solve it: Modern Heuristics 2nd edition”, *Springer-Verglag*, p. 561, 2004.

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

[8] J. Chen and J. Hou, “A Combination Genetic Algorithm with Applications on Portfolio Optimization”, *Advances in Applied Artificial Intelligence, LNCS*, vol. 4031, pp. 197-206, 2006.

[9] 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, pp. 10161–10169, 2011.

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

[11] 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, pp. 6–11, 1993.

[12] 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, no. 8, pp. 5360–5366, 2011.

[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, pp. 133-149, 2013.

[14] M. Tuba and R. Jovanovic, “Improved ACO algorithm with pheromone correction strategy for the traveling salesman problem”, *International Journal of Computers, Communications & Control,* vol. 8, no. 3, pp. 477–485, 2013.

[15] 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, pp. 83–92, 2010.

[16] M. Tuba and R. Jovanovic, “An Analysis of Different Variations of Ant Colony Optimization to the Minimum Weight Vertex Cover Problem”, *WSEAS Transactions on Information Science and Applications*, vol. 6, no. 6, pp. 936-945, June 2009.

[17] C. Dai, W. Chen, Y. Song, and Y. Zhu, “Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization“, *Journal of Systems Engineering and Electronics*, vol. 21, no. 2, pp. 300–311, 2010.

[18] M. Tuba, I. Brajevic, and R. Jovanovic, “Hybrid seeker optimization algorithm for global optimization”, *Applied Mathematics and Information Sciences*, vol. 7, no. 3, pp. 867–875, 2013.

[19] X.-S. Yang and S. Deb, “Cuckoo search via levy flights”, World Congress on Nature & Biologically Inspired Computing, pp. 210–214, 2009.

[20] 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, No. 2, pp. 185–193, 2010.

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

[22] I. Brajevic and M. Tuba, “Cuckoo search and firefly algorithm applied to multilevel image thresholding”, In Yang, X.-S. (ed.) Cuckoo Search and Firefly Algorithm: Theory and Applications, vol. 516 of Studies in Computational Intelligence, Springer International Publishing, pp. 115–139, 2014.

[23] M. Tuba, “Asymptotic behavior of the maximum entropy routing in computer networks”, Entropy, vol. 15, No. 1, pp. 361–371, 2013.

[24] M. Tuba, N. Bacanin and B. Pelevic, “Framework for constrained portfolio selection by the firefly algorithm”, *International Journal Of Mathematical Models and Methods in Applied Sciences*, vol. 7, no. 10, pp. 888-896, 2013.

[25] N. Bacanin, M. Tuba, “Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Optimization Problem with Entropy Diversity Constraint”, The Scientific World Journal, special issue Computational Intelligence and Metaheuristic Algorithms with Applications, Guest Editors: Xin-She Yang, Su-Fong Chien, and T. O. Ting, ISSN: 2356-6140, Volume 2014 (2014), Article ID 721521 , 16 pages, doi:10.1155/2014/721521

[26] A. Alihodzic, M. Tuba, “Improved Bat Algorithm Applied to Multilevel Image Thresholding”, The Scientific World Journal, special issue Computational Intelligence and Metaheuristic Algorithms with Applications, Guest Editors: Xin-She Yang, Su-Fong Chien, and T. O. Ting, ISSN: 2356-6140, Volume 2014 (2014), Article ID 176718 , 18 pages, doi:10.1155/2014/176718

[27] N. Bacanin and M. Tuba, “Artificial bee colony (ABC) algorithm for constrained optimization improved with genetic operators,” *Studies in Informatics and Control*, vol. 21, no. 2, pp. 137–146, 2012.

[28] I. Brajevic and M. Tuba, “An upgraded artificial bee colony (ABC) for constrained optimization problems”, *Journal of Intelligent Manufacturing*, vol. 24, no. 4, pp. 729-740, 2013.

[29] M. Subotic and M. Tuba, “Parallelized Multiple Swarm Artificial Bee Colony Algorithm (MS-ABC) for Global Optimization”, *Studies in Informatics and Control*, vol. 23, no. 1, pp. 117-126.

[30] M. Tuba, N. Bacanin, and N. Stanarevic, “Adjusted artificial bee colony (ABC) algorithm for engineering problems”, *WSEAS Transaction on Computers*, vol. 11, no. 4, pp. 111-120, , April 2012.

[31] M. Tuba and N. Bacanin, “Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio problem”, *Applied Mathematics & Information Sciences*, vol. 8, no. 6, pp. 2934-2947.

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

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

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

[35] D. F. Marco Corazza, “On the existence of solutions to the quadratic mixed-integer meanvariance portfolio selection problem”, *European Journal of Operational Research*, vol. 176, pp. 1947-1960, 2007.

[36] D. Karaboga, “An idea based on honey bee swarm for numerical optimization, Technical Report TR06”, Computer Engineering, Department, Erciyes University, Turkey, 2005.

[37] D. Karaboga and B. Basturk, “Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems”, Lecture Notes in Artificial Intelligence 4529, Springer - Verlag, Berlin, pp. 789–798, 2007.

[38] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, *Springer Science and Business Media*, pp. 459-471, 2007.

[39] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm”, *Applied Soft Computing*, vol. 8, no. 1, pp. 687-697, 2007.

[40] K. Deb, “An Efficient Constraint-handling Method for Genetic algorithms”, *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2-4, pp. 311-338, 2000.

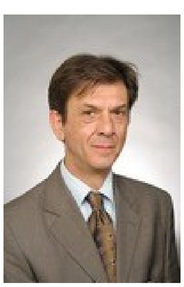
[41] S. Sefiane and M. Benbouziane, “Portfolio selection using genetic algorithm”, *Journal of Applied Finance & Banking*, vol. 4, no. 2, pp. 143–154, 2012.

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

Nebojsa Bacanin was born in Belgrade in 1983. He 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 twenty scientific 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 went to primary and grammar school. He graduated from Faculty of Economics, Belgrade University, and received his master’s and PhD degrees from the same faculty. 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 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. He is fluent in English and a member of Scientific Society of Economists.

Milan Tuba was born in Yugoslavia in 1952. He 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 supervisor of many theses. 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, Dean of the College of Computer Science and Provost for Mathematical and Technical Sciences, 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. He participated in many WSEAS Conferences with plenary lectures and articles in Proceedings and Transactions.