Acta Technica 62 No. 1B/2017, 551–560

Improvement research of genetic algorithm and particle swarm optimization algorithm based on analytical mathematics

Shuai Man[1]

Abstract. Through its own evolution, the optimization method makes many problems that seem highly complex to be solved more perfectly, so a new intelligent calculation method that is different from the classical optimization method is generated. In this paper, the algorithm mechanism, algorithm improvement and application of two kinds of biomimetic intelligent calculation methods of genetic algorithm and particle swarm optimization algorithm were studied deeply. Besides, in view of the constrained optimization problems, two different improvement strategies were adopted, two different improved evolutionary algorithms were proposed respectively and their time and spatial complexity were analyzed. The final experimental results proved that the evolutionary algorithm that integrates the improvement strategies is feasible and effective, and the uniformity and diversity of the solution set are ideal.

Key words. Genetic algorithm, particle swarm optimization, optimization problem.

1. Introduction

The essence of all human activities is nothing more than "knowing the world and building the world". Understanding the world depends on establishing the model, constructing the world relies on the optimal decision-making, and the purpose of optimization is to find a set of parameter values under meeting certain constraint conditions, so as to make some of the performance indicators of the model reach maximum or minimum. The application of the optimization problem can be said everywhere, which always runs through the process of all human activities. In a sense, all human knowledge is nothing more than the phenomenon and process understanding model of human beings to a field. The purpose of knowing the world is to build the world better, similarly, and the purpose of modeling is to optimize. Assuming that the world must first understand the world, similarly, all optimization

[1]Langfang Health Vocational College, Langfang, Hebei, 065000, China

cannot be separated from the model. However, with the continuous improvement of social productivity, the ability of humans to understand and build the world is also growing, followed by optimization problems, which also show the features of high-dimensional, strong nonlinear, strong constraints, difficult to model.

2. State of the art

Bionics was founded in the middle 1950s. People seek a new method for solving all kinds of complex problems in real world from the biological evolution mechanism, and then, the biological simulation becomes an important part of computer science. For example: the early theory was to assume that the machine was composed of basic elements in which they were similar to the neuron [1]. Under the influence of this biological simulation thought, in the early 1970s, a professor in the University of Michigan first proposed the mathematical framework of the genetic algorithm [2]. The idea of genetic algorithm came from Darwin's evolutionism, Weismann's theory of species selection and Mendel's theory of population genetics in biological sciences. In terms of GA, it was a probabilistic search algorithm method that used natural selection and evolutionary mechanisms to find the optimal point in N-dimensional space [3]. From the evolutionary thought, "the survival of the fittest" makes the individual quality of the population has been improved. And the random exchange theory uses the existing information in the original solution to speed up the search process to the optimization. Since more than a decade of the proposition of the particle swarm optimization algorithm, it has attracted many researchers and research institutions at home and abroad to conduct various aspects of exploration to its theory and application. In addition, the research results of the PSO algorithm are increasingly published in high-level publications [4]. Then, the famous conference IEEE CEC in the field of evolutionary computing has set up a special discussion of PSO algorithms. And the important international conference PPSN and GECCO related to computational intelligence have made PSO algorithms as one of the key themes of the conference [5]. In 2001, in view of the PSO theory research and the emergence of applied monograph Group Intelligence, in 2003, the first swarm intelligence symposium IEEE Swarm Intelligence Symposium was held in the United States. Then, PSO algorithm was regarded as one of the main bodies in each year of Symposium. In 2004, the top academic journals in the field of evolutionary computing IEEE Transaction on Evolutionary Computation published PSO algorithm special issue, till now, PSO algorithm has become an important research topic in computational intelligence field [6].

3. Methodology

Genetic algorithm is a kind of self-organizing and adaptive probabilistic search algorithm which simulates the natural evolution process and mechanism to solve the optimization problem. It does not depend on the specific model of the problem, and has strong robustness to all kinds of complex optimization problems. The basic idea of genetic algorithm constructs a fitness function according to the objective function of problems that wait to be solved. Then, according to certain rules, the initial population after the gene encoding is generated, and the evaluation, genetic operations (crossover and mutation), selection and other operations are carried out to the group [7]. After several generations of evolution, one or several optimal individuals with the best fitness are obtained as the optimal solution of the problem. Figure 1 shows the basic GA flow chart. It can be seen from Fig. 1 that the steps in the genetic algorithm implementation include coding strategy, initial population generation, fitness function design, selection strategy, genetic operation and stop criterion.

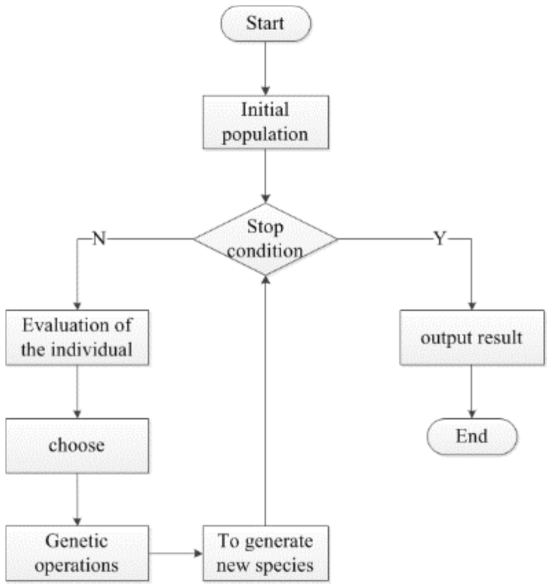

Fig. 1. Genetic algorithm flow chart

It is found that under the situation of the absence of centralized control, the bird group composed of a large number of individuals can make collective acts in flight, such as changing direction, spreading or reorganizing the formation. According to the further observation to the predatory behavior of birds, people feel that there must be some potential capacity or rules to ensure these intelligent behaviors. As a kind of bionic evolutionary algorithm, PSO is similar to genetic algorithm, and it is a kind of optimization technology based on iteration. However, there is no cross mutation operation in the algorithm implementation process [8]. At present, many improved algorithms have been proposed, such as adaptive PSO, hybrid PSO, cooperative PSO [9]. These improved algorithms are mostly based on the standard particle swarm optimization algorithm, and Fig. 2 describes the implementation flow chart of the particle optimization algorithm. The basic principles of the particle swarm optimization algorithm are described as follows.

A group composed of m particles (Particles) flies at a certain speed in D-dimensional

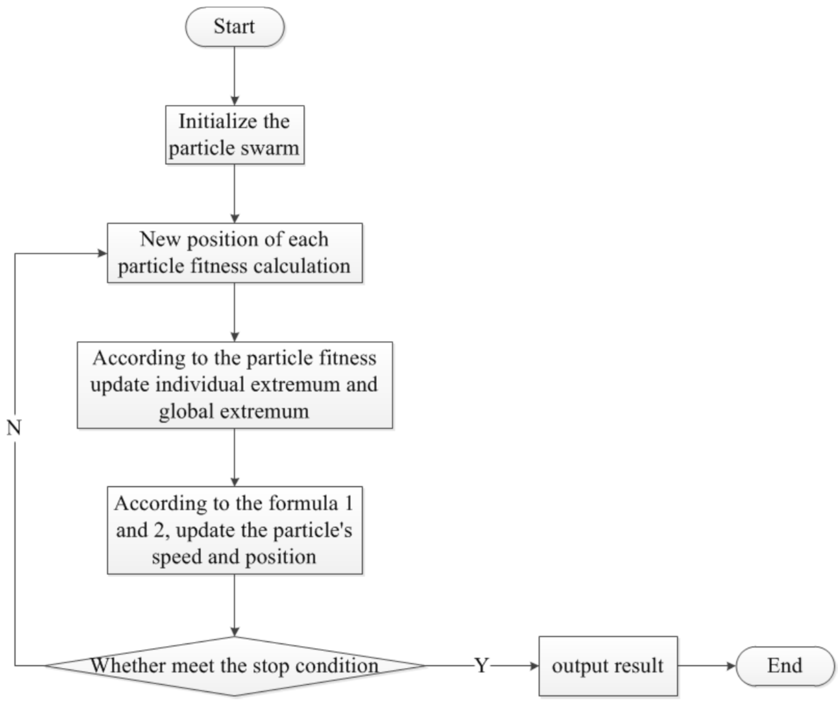

Fig. 2. Particle swarm optimization algorithm flow chart

search space, each particle representing a bird in the search space. For an optimization problem that waits to be solved, a particle is a potential solution [10]. Each particle also has a speed to determine the distance and direction of its flight. All particles have a fitness valve that is determined by the optimized function. In the course of the flight, the particles will use their own flight experience and group flight experience to dynamically adjust themselves, after several iterations of the search, and ultimately, the optimal solution is obtained [11]. PSO is initialized as a group of random particles, and then, the optimal solution is found by iteration. The particles update themselves by tracking two "extremes" in each time of iteration. One is that the optimal solution found by the particle itself is called the individual extremum pbest, and the other is the optimal solution currently found by the whole population. This extreme is the global extremum gbest [12]. Figure 3 depicts the trajectory of the particle flight, and each particle updates its speed and new position by formulas 1 and 2 when the two optimal values are found:

$$v_{k+1} = c_0 v_k + c_1(\mathrm{pbest}_k - x_k) + c_2(\mathrm{qbest}_k - x_k)\,, \tag{1}$$

$$x_{k+1} = x_k + v_{k+1}\,. \tag{2}$$

Here, v_k is particle's velocity vector, x_k is the position of the current particle, pbest_k represents the position of the optimal solution found by the particle itself,

qbest_k represents the position of the optimal solution currently found by the whole population, c_0 is the random number that is generally between (0, 1), which is called the inertia coefficient or contraction factor. Symbols c_1 and c_2 are called as the "self-cognition factor" and "social cognition factor" of the particle, which are respectively used to adjust the effect intension of pbest_k and qbest_k to the particle attraction. Then, the values of c_1 and c_2 are the random number between (0, 2). Finally, v_{k+1} is the sum of vectors v_k, $\text{pbest}_k - x_k$ and $\text{qbest}_k - x_k$. The velocity of each dimension of the particle will be limited by a maximum speed $v_{\max}$.

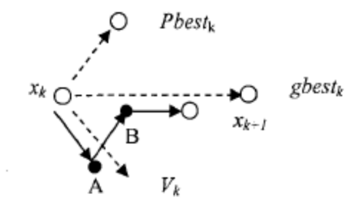

Fig. 3. Particle flight path map

In the PSO algorithm, if all the particles in the group are used as the neighborhood members, at this time, it is known as the global version of the PSO algorithm; if part of members in the group constitute the neighborhood, then, this is called the local version of the PSO algorithm [13]. In the local version, there are two ways to form the neighborhood, one is that the particles in which the index numbers are adjacent constitute the neighborhood, and the other is that the particles that are adjacent according to the spatial distance constitute the neighborhood. The neighborhood definition strategy of the particle swarm optimization algorithm is also called the neighborhood topology structure of the particle swarm.

Multi-objective optimization is a common problem in the field of engineering. Its main characteristic is that there is a conflict between the targets, furthermore, all the targets cannot obtain the optimal value at the same time, only to find a set of compromised Pareto non-inferior solution [14]. The traditional method of solving MOP is to convert MOP into a number of different single objective optimization problems, and then solve them. For the case where the front of the Pareto is non-convex, all Pareto optimal solutions cannot be obtained and the computational complexity is large. Evolutionary algorithm is an adaptive global optimization probabilistic algorithm of the simulated creatures formed in the genetic and evolutionary processes in the natural environment. It is characterized by the multi-directional and global nature of the search, which can process large-scale search space in parallel, and moreover, it has a good adaptability to complex MOP.

Based on the Pareto principle's evolutionary algorithm elitist mechanism, preserving the non-inferior solution obtained by constructing the Pareto candidate set and maintaining the solution diversity in this solution set is an effective means for the multi-objective evolutionary algorithm to obtain non-inferior solution [15]. Based on the criterion of neighborhood function, this paper proposes the construction and maintenance mechanism of Pareto candidate set. The process is described as follows:

Firstly, if the scale of the Pareto candidate solution set does not reach the specified size, the obtained non-inferior solution is added directly to the Pareto candidate solution set.

Secondly, if the new non-inferior solution dictates the individual of the Pareto candidate solution set, the new individual is added to the Pareto candidate solution set and the individuals dominated are placed into another independent external set (IES). Otherwise, the new non-inferior solution is directly added to the Pareto candidate solution set, then, the neighbor function criterion is used to maintain the diversity and population size of the Pareto candidate solution set.

Thirdly, the neighborhood local search is carried out to the individuals of IES, and the newly obtained individual is compared with the Pareto candidate solution set again. If the new individual dominates an individual of the Pareto candidate solution set, this individual is replaced directly. Then, empty the IES.

The algorithm flow is as follows:

Step 1: Setting algorithm parameters: evolutionary algebra is N_{p}, the length of the population is I_{s}, Pareto candidate solution set length is E_{s}, local search length is L_{s}, crossover probability is P_{c}, probability of mutation is P_{m}, and evolutionary algebraic indexer is t.

Step 2: Initializing the population p_{init}, assuming $t = 1$.

Step 3: A set of non-inferior solution is obtained from the group p_{init}, and the number of the non-inferior solution is u, then, the elite retention mechanism is used to add non-inferior solutions into the Pareto candidate solution set.

Step 4: In Pareto candidate solution set, the discrete crossover operator and Gaussian variation operator are used to generate $u_{\mathrm{f}} \times v$ individual, u_{f} is the individual number in Pareto candidate solution set, and v is the proportion coefficient of the sum of descendants and offspring.

Step 5: The population obtained by the cross and variation in the above steps is combined with the Pareto candidate solution set, and then, they are combined into a new population p_{new}, next is to execute $p_{\mathrm{init}} \leftarrow p_{\mathrm{new}}$.

Step 6: Symbol I_{s} is the number of individuals which are selected in the population p_{init}. Here, NSGA-II's non-inferior solution scheme is adopted, and one selection is carried out from low to high (the lower the level is, the higher the rank is), thus, the limited different "grade" groups $\{f_1, f_2, \cdots, f_n\}$ can be obtained, similarly, in the last grade of population f_i, $(1 \leq i \leq n)$ of the individual, the neighbor function criterion is used to select the remaining individuals, and if the termination condition is satisfied, the algorithm stops, otherwise, it moves to Step 3.

4. Result analysis and discussion

Experimental environment: Intel Pentium 4, 2.26 GHz, 512 MB memory, Windows Xp Professional, Matlab 7.0.

The experiment was carried out in two groups G1 and G2, and the algorithm used the real number coding. Then, two groups of typical multi-objective optimization function were selected, the first group of optimization problem was a single target minimum optimization function with high-dimensional constraint conditions, the variable constraints included the test questions of inequality constraints and equality constraints, and the objective function contained up to 10 decision variables. In the second group, the performance comparison of the algorithm in this paper and two commonly used multi-objective evolutionary algorithms NSGA-II and SPEA were conducted by using graphing method.

In the first group of tests, the initial parameters of the algorithm were set as: population individuals were $I_{\mathrm{s}} = 200$, $N_{\mathrm{p}} = 300$, Parteo candidate solution set length was $E_{\mathrm{s}} = 30$, local search length was $L_{\mathrm{s}} = 5$, crossover rate was $P_{\mathrm{c}} = 0.9$, variation rate was $P_{\mathrm{m}} = 0.5$ and each issue ran 20 times independently under the same conditions. Then, all of the implementation of the algorithm was completed on the same computer, and when the algorithm was running, the calculation accuracy was set as 10^{-4}. In order to compare the solution performance of the proposed algorithm on the high-dimensional single objective optimization problems, the comparison was carried out with the other three algorithms: random sort method, homomorphic mapping method and Pareto intensity value evolution algorithm. And the algorithms were respectively denoted as RY, KM and ZW. After each operation, the performance test of the algorithm used the best result (Best), the worst result (Worst) and the average result (Mean) respectively, and it was compared with the optimal solution experimental data. Table 1 shows the correlation comparison result between new algorithm MP and algorithms KM, RY and ZW. The results are listed in Table 1.

It can be seen from Table 1 that for the question g_1, the optimal solution comparison obtained by MP algorithm and ZW algorithm was relatively close, which was close to the real optimal solution, furthermore, it was better than the optimal solution RY. On the mean value and worst solution such two tests, the optimal solution was inferior to ZW, but better than RY. Then, in each performance test, question g_2 and g_4 were better than the other three algorithms, moreover, the proposed MP algorithm found the optimal solution for 13 times in 20 times of the independent operation. While for the question g_3, the MP algorithm was inferior to the ZW algorithm in the three performance tests, but better than other two algorithms KM and RY. In question g_5, although the optimal solution obtained by MP was inferior to ZW, the gap between the solutions was smaller, which was close to the true solution. In addition, MP was superior to the other three algorithms in mean value and worst solution performance. And the reason was that in 20 independent operations, MP found the optimal solution for 16 times.

In the second group of tests, in order to verify whether the proposed algorithm MP can handle two-dimensional multi-objective optimization problems, it was often necessary to design some test functions to evaluate the algorithm. Then, the performance measure standard was used, and two kinds of performance evaluation standards between different algorithms were given, which was respectively the convergence: the convergence of the algorithm could be measured through the actually obtained non-inferior optimal target domain and the minimum distance average value between the theoretically non-inferior optimal target domain; diversity: diversity was used to describe the spread coverage between non-inferior solutions in a population. The performance measure M_1 was used to evaluate the performance of the non-inferior solution comparison in the Pareto candidate solution set of use algorithm in M_1 under the same conditions. The performance measure data of the algorithm are given in Tables 2–4.

Table 1. Correlation comparison between new algorithm MP and algorithm KM, RY and ZW

Problems		g_1	g_2	g_3	g_4	g_5
Optimal solution		5126.49811	0.0539498	680.6300573	7049.3307	23.3062091
Best solution	MP	5126.498162	0.053949821	680.631127	7049.33122	24.30630327
	KM	-	0.054	680.91	7147.9	24.620
	RY	5126.4965	0.053957	680.630	7054.316	24.307
	ZW	5126.49811	0.053949831	680.6300573	7049.2480205	24.306209068
Mean solution	MP	5126.52765	0.053940235	680.6312233	7050.153672	24.315356306
	KM	-	0.064	681.16	8163.6	24.826
	RY	512.881	0.057006	680.656	7559.192	24.374
	ZW	5126.52654	0.053950257	680.6300573	7051.2874292	24.325487652
Worst solution	MP	5139.2522	0.0539677331	680.63123678	7055.233735	24.350563206
	KM	-	0.557	683.18	8659.3	25.069
	RY	5124.472	0.216915	680.763	8835.665	24.642
	ZW	5127.15641	0.053972292	680.6300573	7058.2353585	24.362999860

Table 2. Convergence performance measure γ

Algorithm	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MP	0.003401	0.001079	0.000856	0.001189	0.001137	0.320625
	0	0.000101	0.000056	0.000069	0.000153	0.042030
NSGA-II	0.003389	0.033480	0.072389	0.114499	0.513053	0.296566
	0	0.004751	0.031688	0.007938	0.118460	0.013138
SPEA	0.003403	0.001799	0.001339	0.047520	7.340299	0.221138
	0	0.000001	0	0.000049	6.572416	0.000448

Table 3. Diversity performance measure Δ

Algorithm	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MP	0.566835	0.325136	0.328062	0.408365	0.358806	0.750632
	0.0236635	0.001460	0.001406	0.002352	0.016002	0.005562
NSGA-II	0.0477899	0.0390307	0.430776	0.738540	0.702612	0.668025
	0.003471	0.001876	0.004721	0.019706	0.064648	0.009923
SPEA	1.021110	0.784525	0.755148	0.672938	0.798463	0.849389
	0.004372	0.004440	0.004521	0.003587	0.014616	0.003916

Table 4. Comparison of the performance measures M_1 and SP

Algorithm	SCH	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MP	0.002635	0.002312	0.002640	0.001355	0.039862	0.040362
	0.032651	0.002682	0.003919	0.001561	0.402513	0.056230
NSGA-II	0.006321	0.002906	0.003368	0.0015626	0.043115	0.036250
	0.010265	0.005613	0.007227	0.0089013	0.336522	0.033621
SPEA	0.009632	0.008653	0.140322	0.0026354	0.044752	0.040356
	0.010263	0.006739	0.010654	0.02956665	0.35032	0.0625103

As can be seen from Table 2, on the SCH, the convergence of the proposed algorithm MP was closer to the other two algorithms. In ZDT6, it was worse than the other two algorithms. The convergence of ZDT1 to ZDT4 indicated that the algorithm in this paper was superior to the other two algorithms. The data of diversity performance measures Δ in Table 3 showed that the diversity of MP in the SCH solution was inferior to the other two algorithms, but superior to the other two algorithms on ZDT1 to ZDT4. Then, the diversity of the three algorithms' solution on ZDT6 was closer. Besides, in the performance measurements on M_1 and SP, Table 4 showed that for the non-inferior performance difference generated under the same conditions and the algorithm performed the same number of target calculation each time, the performance measure of MP in SCH, ZDT1 to ZDT3 was better than other two algorithms, on ZDT4 and ZDT6, the performance measures of the three algorithms were similar.

In this paper, from above experimental results and performance analysis, it can be seen that under the single-dimensional and two-dimensional situation, the algorithm in this paper performs better, and the Pareto front obtained by the algorithm is relatively close to the real Pareto front. On the basis of the proposed algorithm MP, the improvement is carried out by combining with the characteristics of practical engineering applications, which will have a good engineering application prospect.

5. Conclusion

Although the intelligent optimization method has achieved many remarkable results, the combination of it with the specific practice areas still has many problems to be solved. In this paper, the popular hot spot methods in the field of intelligent optimization, the genetic algorithm and particle swarm optimization algorithm, were studied from the perspective of bionics, then combined with the biology basis of the two methods, some in-depth researched were carried out and some achievements were obtained: for the multi-objective optimization problem, the method of maintaining the population diversity by traditional evolutionary algorithm mainly depends on the shared function, however, its niche radius is difficult to set effectively. Then, the neighborhood function criterion can be introduced to the selection process, so as to select good individuals from the population and ensure the diversity of the population. In addition, a candidate set maintenance method based on neighborhood function criterion can be integrated into the new algorithm, and the use of this method can effectively maintain the diversity of individuals in candidate solution sets. Then, the proposed algorithm is analyzed theoretically from time and space complexity. The test of a group of typical optimization problem shows that the proposed algorithm has relatively high search performance, and the diversity and convergence of solution set distribution are ideal. Of course, there are many places in this paper that need further study, such as an improved algorithm for nonlinear programming problems.

References

[1] Y. Wang, Z. Cai, G. Guo, Y. Zhou: *Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) *37* (2007), No. 3, 560–575.

[2] M. Kanakubo, M. Haqiwara: *Parameter-free genetic algorithm using pseudo-simplex method.* Systems and Computers in Japan *36* (2005), No. 13, 35–44.

[3] T. Li, D. W. Cui: *Genetic algorithm selection and parameter setting based on rule induction.* Computer Engineering *36* (2010), No. 3, 218–220.

[4] M. S. Gibbs, H. R. Maier, G. C. Dandy: *Comparison of genentic algorithm parameter setting methods for chlorine injection optimization.* Journal of Water Resources Planning and Management *136* (2010), No. 2, 288–291.

[5] C. C. Y. Dorea, J. A. Guerra Jr., R. Morgado, A. G. C. Pereira: *Multistage Markov chain modeling of the genetic algorithm and convergence results.* Numerical Functional Analysis and Optimization *31* (2010), No. 2, 164–171.

[6] J. F. Wang, B. Q. Du, H. M. Ding: *A genetic algorithm for the flexible job-shop scheduling problem.* Advanced Research on Computer Science and Information Engineering. Proc. International Conference, CSIE 2011, May 21-22, 2011, Zhengzhou, China, Part I, 332–339.

[7] Y. Peng, X. Luo, W. Wei: *A new fuzzy adaptive simulated annealing genetic algorithm and its convergence analysis and convergence rate estimation.* International Journal of Control, Automation and Systems *12* (2014), No. 3, 670–679.

[8] F. Herrera, M. Lozano, J. L. Verdegay: *Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis.* Artificial Intelligence Review *12* (1998), No. 4, 265–319.

[9] F. Y. Xiao, P. K. Li: *An exact schema theorem for adaptive genetic algorithm and its application to machine cell formation.* Expert Systems with Applications *38* (2011), No. 7, 8538–8552.

[10] H. Min, H. J. Ko, C. S. Ko: *A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns.* Omega *34*, (2006), No. 1, 56–69.

[11] C. S. Zhang, D. T. Ouyang, N. Yue, Y. G. Zang: *A hybrid algorithm based on genetic algorithm and Levenberg-Marquardt.* Journal of Jilin University(Science Edition) *46* (2008), No. 4, 675–680.

[12] S. Zhuo, C. Liu, K. Q He: *A software pattern of the genetic algorithm.* Wuhan University Journal of Natural Sciences *6* (2001), Nos. 1–2, 209–217.

[13] K. Deep, K. N. Das: *Quadratic approximation based hybrid genetic algorithm for function optimization.* Applied Mathematics and Computation *203* (2008), No. 1, 86 to 98.

[14] D. H. Kim, A. Abraham: *Hybrid genetic algorithm and bacterial foraging approach for global optimization and robust tuning of PID controller with disturbance rejection.* Hybrid Evolutionary Algorithms, Part: Studies in Computational Intelligence, Springer-Verlag Berlin Heidelberg *75* (2007).

[15] K. Vaisakh, L. R. Srinivas: *Evolving ant colony optimization based unit commitment.* Applied Soft Computing *11* (2011), No. 2, 2863–2870.

Received July 12, 2017