The application of swarm activity based bat algorithm in constrained optimization

Wu Tiebin [1, a], Long Wen [2,b*] LiuYunlian[1,c] Li Xinjun[1,d]

[1] Hunan University of Humanities, Science and Technology,Loudi,Hunan,417000, China

[2] Guizhou Key of Economics System Simulation, Guizhou University of Finance and Economics, Guiyang, Guizhou 550004;

[a]wutiebin81@163.com, [b*]lw770457@163.com, [c]liuyunlian85@163.com, [d]lixinjun80@163.com

* Long Wen: Corresponding Author

Keywords: Bat algorithm; Constrained optimization; Adaptive penalty function; Swarm activity

Abstract. A solving method for constrained optimization problems was designed by combining adaptive penalty function (APF) method and an improved bat algorithm (IBA) based on swarm activity. Firstly, the authors proposed a kind of APF method. And then considering the conditions of constraint violations and the characteristics of evolutionary process, the IBA based on swarm activity was put forward. On this basis, APF method and IBA were combined together to solve constrained optimization problems. Finally, it was demonstrated by simulation that the solving method for constrained optimization was feasible and effective.

Introduction

In recent years, intelligent optimization algorithm has been widely used to solve constrained optimization problems. But it is necessary to employ a suitable constraint-handling technique during solving constrained optimization problems by using intelligent optimization algorithm. Combined with the modified and augmented Lagrange multiplier method and differential evolution algorithm, Long et al. [1] proposed an improved hybrid algorithm to solve constraint numerical values and the engineering optimization problems. Le Riche et al. [2] proposed an isolation genetic algorithm. Besides, an intelligent algorithm based on multi-objective optimization was put forward to conduct constrained optimization [3]. Surry et al. proposed COMOGA [4] to regard the single-objective constrained optimization problems as the problems with single-objective unconstrained optimization or constraint satisfaction, and the method achieved preferable results.

Yang from University of Cambridge proposed bat algorithm [5] in 2010, which was characterized by simple model, potential parallelism, etc. Moreover, its convergence rate and stability are superior to the basis particle swarm optimization (PSO). Whereas, just like the other swarm intelligent algorithms, it also has some defects, for example, it is prone to converge prematurely and with weak local search capability, etc.

As for these problems, an APF method was proposed to handle constraint conditions. In order to improve the optimization performance of bat algorithm, Logistic mapping was applied to generate the initial population to improve the diversity. On this basis, a local search strategy with crossover operation was put forward to improve local optimization capability. Moreover, in order to prevent premature convergence, a mutation strategy based on swarm activity was established. And then the constraint-handling method was combined with the IBA to solve constrained optimization problems.

A novel APF method

It is assumed that a constraint-handling problem includes m inequality constraints, $g_i(x)$ ($i=1,2,\mathbf{L},m$) and p inequality constraints, $h_i(x)$ ($i=m+1,\mathbf{L},m+p$). On the basis of the constraint-handling method in intelligent algorithm, the equality constraints can be converted to inequality constraint according to Formula 1, namely,

$$\left|h_j(x)\right| - d \le 0, \quad j = m+1, m+2, \mathbf{L}, m+p \tag{1}$$

where d is small positive numbers.

Suppose that function $R_i(x)$ represents the violation degree of an individual, x in an intelligent algorithm population to each constraint condition, and it is expressed as

$$R_i(x) = \begin{cases} \max(0, g_i(x)) & i = 1, 2, \mathbf{L}, m \\ \max(0, (\left|h_i(x)\right| - d)) & i = m+1, m+2, \mathbf{L}, m+p \end{cases} \tag{2}$$

Then the new objective function $F(x)$ is

$$F(x) = \begin{cases} \dfrac{f(x)}{|f|_{\max}} & \textit{if } x \text{ is feasible solution} \\ g(x) + \dfrac{v(x)}{|v|_{\max}} & \textit{otherwise} \end{cases} \tag{3}$$

where $f(x)$ is the corresponding objective function of the individual x.

Wherein, $|f|_{\max} = \max(\left|f(x_1)\right|, \left|f(x_2)\right|, \mathbf{L}, \left|f(x_M)\right|)$, where M is the size of the population, and the formula refers to the maximum absolute value of the primitive objective function value $f(x)$ for all individuals in the population, excluding the penalty terms. $g(x)$ and $v(x)$ are as shown in Formulas 4 and 5 respectively:

$$g(x) = \begin{cases} m & \text{if } f(x) < f_{\text{mean}} \\ \dfrac{f(x)}{|f|_{\max}} & \textit{otherwise} \end{cases} \tag{4}$$

$$v(x) = \sum_{i=1}^{m+p} (b_i \times R_i(x)) \tag{5}$$

where m is the constant, in the range of $0.6 \le m \le 1$, and $|v|_{\max} = \max(\left|v(x_1)\right|, \left|v(x_2)\right|, \mathbf{L}, \left|v(x_M)\right|)$. b_i is penalty coefficient, as indicated in Formula 6.

$$b_i = x_1 \times (1 + \frac{s_i}{s_A}) + x_2 \times (1 + \frac{num}{\mathrm{M}}), \; i = 1, 2\,\mathbf{L}, m+p \tag{6}$$

where x_1 and x_2 are constants meeting the condition of $[0,1]$, and $x_1 = x_2 = 0.5$ here, s_i represents the total frequency for violating the i^{th} constraint in the present generation, s_A stands for the frequency for violating all constraints, and num is the number of feasible solutions in present population.

It can be known from Formulas 3, 5 and 6 that if the frequency for violating a constraint is higher, then the constraint is stronger and the weight endowed is greater. If there are more infeasible solutions in present population, the constraint should take a smaller value to keep the population diversity. In addition, the objective function values of the feasible solutions are not always greater than those of the infeasible solutions. In this way, some infeasible solutions can enter the next generation to improve the probability of searching the global optimal feasible solution at the common border of feasible region and infeasible region.

Improved bat algorithm

The basic bat algorithm [5] changes bats' behavior by frequency tuning and pulse loudness, which is superior to the essential genetic algorithm and PSO algorithm during solving unconstrained optimization problems. But the algorithm is similar with the other swarm intelligence algorithm that it has some disadvantages, for example, it is apt to being caught in local optimal solution owing to the lack of the mechanism escaping from the local optimum and its convergence rate is slow at the late stage of optimizing. Thereby, it is necessary to improve bat algorithm.

Generating the initial population by using chaos method

The initialization mode of population greatly influences the convergence performance of the evolutionary algorithms for all populations. Thus, it is needed to make the initial populations distribute as evenly as possible in the whole search space.

In this chapter, Logistic equation was employed to produce initial population. The Logistic mapping model is as follows:

$$X_{m+1,n} = mX_{m,n}(1 - X_{m,n}) \qquad (n = 1,2\cdots,\mathrm{N};\ m{=}1,2\cdots,\mathrm{M}) \qquad (7)$$

where when $X_{m,n} \in [0,1]$, and $m = 4$, the system is in a chaos state and its track is with favourable ergodicity.

$X_{m,n}$ was transformed according to Formula 8 to obtain the initial population with size of M and dimension of N.

$$x_{m,n} = \mathrm{L}_n + X_{m,n}\ (\mathrm{U}_n - \mathrm{L}_n) \qquad (8)$$

where U_n and L_n are respectively the upper bound and lower bound of variables or bats in the n^{th} dimension.

Producing uniform population by using Logistic chaos mapping can improve the searching efficiency of the algorithm.

Local search strategy with crossover operation

In the basic bat algorithm, if $rand(1) < r_i$, where r_i is the pulse frequency of the i^{th} bat and $rand(1)$ is a random number from 0 to 1, the local search formula is as follows:

$$x_{new}(i) = x_{old} + s \times AV^t \qquad (9)$$

where x_{old} is the selected bat (solution), AV^t is the average value of the pulse loudness of the bat in the t^{th} generation, s is the random number of d dimension from -1 to 1, and x_{new} stands for the new position (solution).

Owing to the global optimal bat fails to be fully used in the local search, the local search is modified as:

$$x_{new}(i) = l_1 \times x_{old} + l_2 \times x^{*'} + s \times AV^t \qquad (10)$$

where l_1 and l_2 are coefficients, and $x^{*'}$ is the global optimal bat. Considering that $l_1{+}d_2{=}1$, here d_1 is 0.6 and d_2 is 0.4. Then the convergence rate of algorithm can be accelerated by integrating crossover operation into local search, that is to say, by intersecting with the optimal bat.

The mutation strategy based on swarm activity

Owing to the mechanism escaping from the local optimum is absent for the basic bat algorithm, all bats gradually approach to the optimal bat, namely the optimal solution in the late algorithm evolution, which means that the positions of bats are extremely close. On the condition, once the position of the optimal bat is locally optimal, the algorithm is highly likely to be premature.

Hence, a mutation mechanism based on swarm activity was applied, that is to say, mutation was conducted in the process of algorithm evolution to obtain the mutation probability, as shown in Formula 11.

$$p_m = f(f) \times p_{m0} + (1 - f(f)) \times p_{m1} \qquad (11)$$

where p_{m0} is the small mutation probability, and $p_{m0} \in [0.01, 0.10]$, while p_{m1} is the large mutation probability, which $p_{m1} \in [0.2, 0.6]$; $f(f)$ is the correlation function on objective function value of bat (see Formula 12).

$$f(f)=\begin{cases}0 & if \quad \dfrac{f_{avg}}{f_{\max}}>k_f \\ 1 & if \quad \dfrac{f_{avg}}{f_{\max}}\le k_f\end{cases} \tag{12}$$

where f_{avg} is the average value of the objective functions of all bats under the assumption that all the objective function values are more than or equal to 0; $f_{\max}$ is the corresponding objective function value of the optimal bat, k_f is a coefficient, and $k_f \in [0.85,1.0)$. Obviously, if $f_{\max}$ is really close to f_{avg}, it shows that the population activity reduces. Thus, it is necessary to improve the population activity by more mutation probability.

In other words, when all objective function values are approximate, which means that the vast majority of the positions of bats is close, the bats should be dispersed by using great mutation probability to improve the global optimization capability of algorithm.

Numerical experiment and comparison

For the bat algorithm based on swarm activity, the mutation probability p_{m0} was set as 0.05 and p_{m1} was 0.2. Four complex test functions were utilized to test the hybrid algorithm composed by APF and IBA, namely APF-IBA, while four extremely complex constrained optimization functions were set as g07, g16, g18 and g19 respectively [6]. Then APF-IBA was compared with the hybrid algorithm including exponential penalty function [7] and basic bat algorithm, namely EPF-BA, where 2000 iterations was conducted for each algorithm. Each function was independently operated for 30 times. Besides, the optimal results, the average results, the worst results and the standard deviations were recorded and listed in Table 1.

Table 1 The optimization results for three functions by using APF-IBA and EPF-BA

Function	Optimal	Statistical results	Method	
			EPF-BA	APF-IBA
g07	24.30620	Best	24.30812	24.30620
		mean	24.34675	24.31221
		worst	24.44832	24.33242
		std	2.52E-02	4.20E-03
g16	-1.90515	Best	-1.90133	-1.90514
		mean	-1.81323	-1.8762
		worst	-1.69617	-1.8521
		std	6.81E-02	1.19E-02
g18	-0.86602	Best	-0.86492	-0.86599
		mean	-0.86243	-0.86543
		worst	-0.85997	-0.86211
		std	1.91E-03	9.57E-04

It can be known from Table 1 that the optimization results of APF-BA were superior to those of EPF-BA. It was proved the validity of the APF method and bat algorithm proposed in this research.

Conclusions

The authors proposed an APF method to handle the constraint conditions and an improved bat optimization algorithm at first. Then they combined the two methods together to solve constrained optimization problems. Finally, it was demonstrated that the algorithm was effective through a simulation test.

Acknowledgements

This work was partially supported by the project supported by National Natural Science Foundation of China (NO. 61463009), and science and Technology Department of Hunan Province(NO.2014GK3033), and science and Technology Department of Loudi city, and Scientific Research Fund of Hunan Provincial Education Department(NO.14B097)

References

[1] LONG W, LIANG X M, HUANG Y F, et al. A hybrid differential evolution augmented Lagrangian method for constrained numerical and engineering optimization[J]. Computer-Aided Design, 2013, 45(12): 1562-1574.

[2] Le Riche R.G., Knopf-Lenoir C., Haftka R.T. A segregated genetic the 6th International Conference on Genetic Algorithms, Morgan Kaufman, San Francisco, CA, 1995, 558-565.

[3] CAI Z X,WANG Y.A Multiobjective Optimization-based Evo-lutionary Algorithm for Constrained Optimization [J]. IEEE Trans on Evolutionary Computation, 2006, 10(3): 658-675.

[4] Surry P.D., Radcliffe N.J. The COMOGA method: constrained optimization by multiobjective genetic algorithm. Control and Cybernetics, 1997, 26(3): 391-412.

[5] Yang X S. Nature Inspired Meta-heuristic Algorithms (2nd Edition)[M]. Frome, UK: Luniver Press, 2010: 97-104.

[6] Wang Y, Cai Z, Zhou Y. Accelerating adaptive trade‐off model using shrinking space technique for constrained evolutionary optimization[J]. International journal for numerical methods in engineering, 2009, 77(11): 1501-1534.

[7] Han Y L. Research on the penalty function of constrained optimization problem [D]. Zhengzhou: Henan Polytechnic University, 2011