Application of Hybrid Genetic Algorithm Based on Simulated Annealing in Function Optimization

Panpan Xu, Shulin Sui, Zongjie Du

***Abstract*—**Genetic algorithm is widely used in optimization problems for its excellent global search capabilities and highly parallel processing capabilities; but, it converges prematurely and has a poor local optimization capability in actual operation. Simulated annealing algorithm can avoid the search process falling into local optimum. A hybrid genetic algorithm based on simulated annealing is designed by combining the advantages of genetic algorithm and simulated annealing algorithm. The numerical experiment represents the hybrid genetic algorithm can be applied to solve the function optimization problems efficiently.

***Keywords*—**Genetic algorithm, Simulated annealing, Hybrid genetic algorithm, Function optimization.

I. Introduction

As a random search algorithm, genetic algorithm has the characteristics of natural selection and genetic mechanism of the biological community [1]. The mode of genetic algorithm realizes a distributed collection and a group search in the solution space. The genetic algorithm breaks the limitation of neighborhood search compared with other general heuristic algorithms. In recent years, genetic algorithm has been widely applied into the optimization field because of its excellent global search capabilities and highly parallel processing capabilities [2], [3]. However, there are conflicts between convergence speed and global optimum in some function optimization problems, which causes premature convergence in genetic algorithm. In order to solve the problem, designing a hybrid genetic algorithm is one of the effective ways. Simulated annealing algorithm has an outstanding application in solving large-scale optimization problems. Simulated annealing algorithm plays a good role in the process of local optimization for its leaping search methods. In this paper, we propose a hybrid genetic algorithm by introducing the strong local search ability of the simulated annealing algorithm into the genetic algorithm, and apply it into function optimization problems.

II. Algorithm Introduction

A. Introduction of Genetic Algorithm

Genetic Algorithm (GA) is a kind of simulated evolutionary algorithm, which is proposed by Professor Holland of the University of Michigan. Genetic algorithm does not depend on the specific problem areas, Its robustness is strong and it has unique advantages in solving complex optimization systems.

GA is widely used in numerical optimization, computer science, artificial intelligence, technology, and engineering practices [4].

As a stochastic optimization and search method, Genetic algorithm is different with the enumeration method, heuristic algorithms, and other traditional optimization methods. It has the following features:

(1) GA starts the search from a group instead of a single initial solution to get the optimal solution [5]. Its wide coverage is conducive to global optimization, and GA searches multiple regions in the solution space with a certain implicit parallelism.

(2) GA is an adaptive search technology by random methods, the selection, crossover and mutation operations in GA are carried out probabilistically, thereby it increases the flexibility of the search process.

(3) The basic idea of GA is simple. The normative operation mode and achieve step in GA can be used to search a variety of optimization problems, and its parameters can be adjusted for different examples [6].

Genetic algorithm with the principle of "survival of the fittest", starts to search by encoding a set of randomly generated initial solutions. Genetic algorithm evolves constantly to get a new generation by going through with the rules of selection, reproduction, crossover and mutation operations, and makes the search process approach to optimal solution based on the fitness function. After several evolutions, genetic algorithm converges to the optimal solution or satisfactory solution. The genetic algorithm mainly includes the coding design, setting the initial group, designing the fitness function, setting control parameters and designing genetic operations (including population size, hereditary frequency, crossover probability and mutation probability, etc.).

In solving the optimization problems, genetic algorithm organizes to search by the method of population without other auxiliary information, and it can search in multiple regions of space at the same time, so the genetic algorithm has a rapid convergence and excellent global search abilities [7]. However, in practical application, genetic algorithm has some shortcomings, such as premature convergence and falling into local optimum easily.

B. Introduction of Simulated Annealing Algorithm

Simulated annealing algorithm was raised by Metropolis. He penetrated thermodynamic principles into the numerical calculation [8]. Simulated annealing algorithm is mainly used in a variety of optimization problems, and function optimization is a very important aspect. The starting point of

P. Xu , S. Sui and Z. Du are with the Department of Mathematics, Qingdao University of Science & Technology, Shandong 266061, China (e-mail: qustxupanpan@163.com).

simulated annealing algorithm is based on the similarity of the annealing process of solid substances in physics and general optimization problems [9]. The basic idea of the algorithm is starting from a given solution, and generating another solution randomly from the neighborhood. The acceptance criterion allows objective function to deteriorate within a limited range. Simulated annealing algorithm is determined by a control parameter T and algorithm continues iterative process by generating new solutions - determining - accepting or giving up for each value of the control parameter T , It corresponds to the process of thermal equilibrium at a constant temperature. We can obtain the optimal solution relative to a given control parameter T after transformations of a number of solutions. Then reduce the control value of the parameter T , and repeat the above iterative process. When the control parameter T decreases and tends to zero, the system has become more and more balanced [10]. Finally, the system state corresponds to the optimal solution of the optimization problem.

SA is a random search algorithm, which starts with an initial solution. SA decides whether to accept the new solution in accordance with the Metropolis Criterion, Finally it receives the optimal solution along with temperature reduction. For seeking the maximum of a function, the basic steps are as follows:

(1) Initialization: the initial temperature T , the initial solution S (starting point of iterative algorithm), iterations L of each T .

(2) Do step (3) to step (6) for $k = 1,......L$.

(3) Perturb the initial solution S , Generate a new solution S' .

(4) Calculate increment $\Delta f = f(S') - f(S)$, $f(S)$ is the evaluation function.

(5) Metropolis Criterion: if $\Delta f > 0$, then accept S' as the new solution, else accept S' as the new solution in accordance with the probability $\exp(\Delta f / \mathrm{T})$.

(6) If the termination condition is satisfied, output current solution as the optimal solution, then terminate the program.

(7) T decreases, $T \to 0$, then do step (2).

Simulated annealing algorithm is different from the local search algorithms. Simulated annealing algorithm accepts new solutions based on Metropolis criterion; thereby, in addition to receiving the optimal solution, it still accepts deteriorative solution within a limited range [11]. In the simulated annealing algorithm, we should pay attention to the following issues:

(1) Theoretically, the cooling process must be slow enough, so that it can reach thermal equilibrium at each temperature. However in computer implementations, if the cooling rate is too slow, the performance of the resulting solution will be more satisfactory, but it searches in a longer time, so it does not have a clear advantage compared with a simple search algorithm. If the cooling speed is too fast, it could not eventually get the global optimal solution. Thereby, we should take a compromise between the performance and speed of algorithm.

(2) Determine the end of the guidelines in state transition at each temperature [12]. The simulated annealing algorithm ends while the final temperature reaches a smaller value determined in advance, or the conversion process does not make the state change at several continuous temperatures .

In order to obtain the optimal solution, SA usually require a higher initial temperature and search in a longer time, but SA adopts Metropolis Criterion, So this algorithm can avoid falling into local optimum .

III. Hybrid Genetic Algorithm Based on Simulated Annealing

Genetic algorithm has good global search capabilities, and it can quickly search out the all solutions from the solution space. However, genetic algorithm has poor local search abilities, and leading to a lower search efficiency in the late stage of evolution, In practical applications, Genetic algorithm causes a premature convergence problem [13]. It has been difficult to select a method that not only makes the best individual preserved, but also maintains the diversity of the population in genetic algorithm.

Although simulated annealing algorithm has fast search capability to get the local optima, simulated annealing algorithm does not know much about status of the entire search space [14]. Therefore, it is not easy to make the search process into the most promising search area, and making the operation efficiency low in the early global search. Simulated annealing algorithm has a strong dependence on parameter (initial temperature), and the evolution is slow.

Based on the advantages and disadvantages both in genetic algorithm and simulated annealing algorithm, we combine genetic algorithm and simulated annealing algorithm and design a hybrid genetic algorithm based on simulated annealing (SAGA). The basic idea of hybrid genetic algorithm based on simulated annealing is: the operation of the process is similar to the basic genetic algorithm, and simulated annealing hybrid genetic algorithm starts searching for a global optimal solution by generating initial solution (initial population) randomly. It is the first to produce a new set of individuals through genetic operations of selection, crossover and mutation. Then we do simulated annealing process independently for each individual, and the results can become the individuals of next-generation group. This process is repeated iteratively until a termination condition is satisfied. Simulated annealing hybrid genetic algorithm greatly improves efficiency by fully integrating the advantages of genetic algorithm and simulated annealing algorithm. SAGA can overcome the defect of long search time and avoid falling into local optimum [15]-[16].

SAGA is divided into two stages: GA stage and SA stage. First of all, we evolve a group of solutions by performing genetic manipulation, and then we adjust and optimize the solution by SA, After repeating iterations, we gets the optimal solution finally. The specific steps are as follows:

(1) Parameters Initialization: population size N , crossover probability P_c , mutation probability P_m , hereditary frequency t , initial temperature T and so on.

(2) Code gene, generate initial population randomly C_t .

(3) Calculate the fitness value of each individual in population.

Determine whether the result of the program is satisfied with termination condition, if the result of the program is satisfied with termination condition, then do step (7).

(4) Do the selection, crossover, and mutation operations for individual in population C_t, then get population C_t'.

(5) After the individual variation, do the inner loop operation of simulated annealing algorithm until obtain stable population C_t''.

(6) Temperature reduction $T_{t+1}=\alpha T_t$, $C_{t+1}=C_t''$, $t=t+1$, go to step (3).

(7) Output results.

IV. Numerical Experiment

Many practical problems with a wide range of issues and complex factors can be abstracted as a function optimization problems when they are mathematical modeled, so it is very important to look for a way to handle a variety of complex functions with a good performance approach [17].

To verify the effectiveness of SAGA, We select a representative function to experiment compared with GA and SA. Seek maximum of the typical function:

$$f(x_1,x_2)=0.5-\frac{\sin^2\sqrt{x_1^2+x_2^2}-0.5}{\left[1+0.001(x_1^2+x_2^2)\right]^2},-5\le x_i\le 5,i=1,2\,,$$

Its geometry is shown in Fig. 1, it is a multi-peak function, There are an infinite number of local maxima and a global maximum point $f(0,0)=1$, Fig. 1 represents there is a peak around the maximum point, Thus the optimization process is most likely to stuck in these local optimum points.

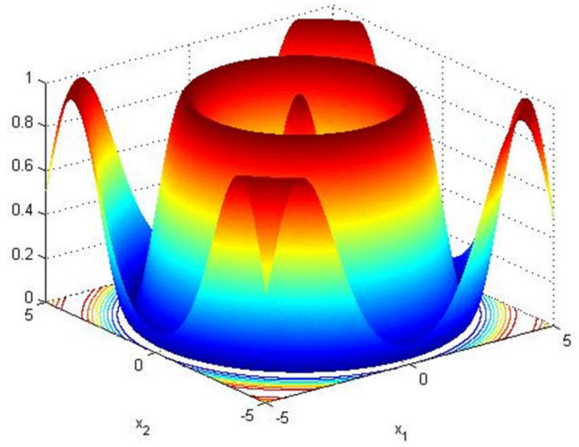

Fig. 1 Function Geometry

In this paper, we solve the maximum value of the above function separately by GA, SA, and SAGA. The parameters of GA are as follows: population size N is 60, crossover probability P_c is 0.6, mutation probability P_m is 0.02, hereditary frequency t is 500. The parameters of SA are as follows: the initial temperature T is 100, and the decay parameter of temperature α is 0.99. The parameters of SAGA are as the same as GA and SA. We program by software MATLAB, and the results are shown in Table I. Table I represents that we can get the maximum value of a function by SAGA, which is better than GA or SA.

TABLE I
The Results Obtained by 3 Algorithms

algorithm	maximum	x_1	x_2
SA	0.99028	0.5380	-3.0920
GA	0.9903	1.1251	2.9299
SAGA	1.0000	0.0000	-0.0003

The relationship between the best function value of each generation and the iteration in genetic algorithm is shown in Fig. 2 (because the function values do not change at the late stage of GA, so we select only the first 100 generations to map to make the effect obvious). Fig. 2 represents that GA has a fast rate of convergence, but later evolution cannot get rid of the local maxima.

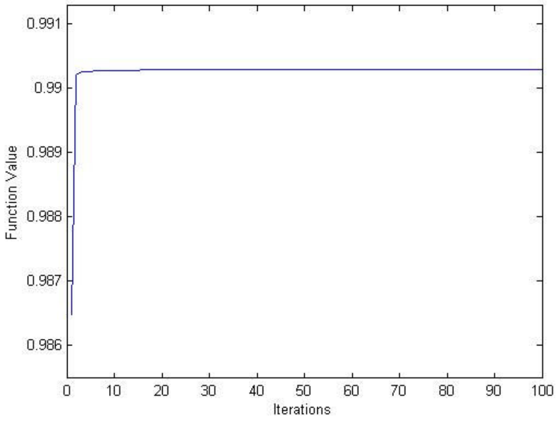

Fig. 2 Genetic Algorithm Iteration

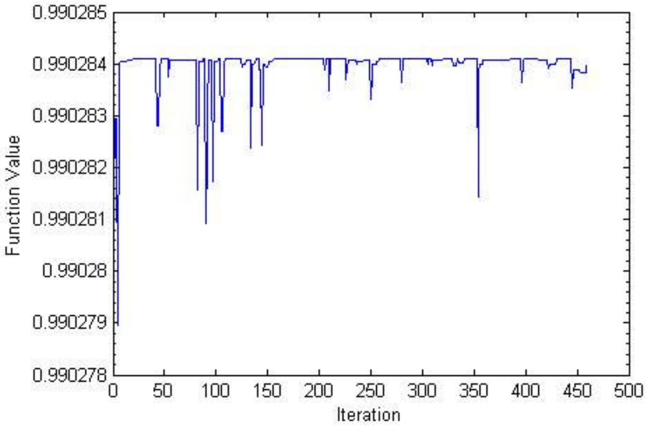

Fig. 3 Simulated Annealing Iteration

The relationship between the best function value of each generation and the iteration in simulated annealing algorithm is shown in Fig. 3. Fig. 3 represents that SA can accept the new solution by probability and SA has a sudden rebound. So SA avoids falling into local maxima. However, we can see in Fig. 3, SA searches for a long time.

The relationship between the best function value of each generation and the iteration in hybrid genetic algorithm based on simulated annealing is shown in Fig. 4. Fig. 4 represents that SAGA can get rid of the local optima and achieve the maximum optimization quickly [18]. The final result verifies the effectiveness of the hybrid genetic algorithm.

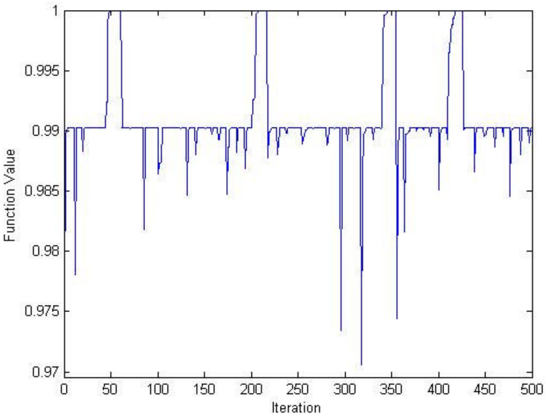

Fig. 4 Hybrid Genetic Algorithm Iteration

We continue to experiment the function above for 50 times separately with SA, GA, and SAGA. The success rate in that function optimization problem is shown in Table II. Table II represents hybrid genetic algorithm can get the global optimal solution every time and its success rate reaches 100% which is higher than the separate genetic algorithm and simulated annealing algorithm significantly [19]. The result proves that introducing simulated annealing algorithm into genetic algorithm can improve the efficiency and accuracy of genetic algorithm.

TABLE II
THE SUCCESS RATE OF THE FUNCTION OPTIMIZATION PROBLEM

Algorithm	Optimal solution	Worst solution	Number of successes	Success rate
SA	1.0000	0.9628	4	8%
GA	1.0000	0.9903	3	6%
SAGA	1.0000	1.0000	50	100%

V. CONCLUSION

In practical applications, Genetic algorithm converges to a local optimum at a rapid speed, but it is difficult to get rid of the local optima. Considering the advantages and disadvantages of genetic algorithm and simulated annealing algorithm, we design a hybrid genetic algorithm based on simulated annealing algorithm. The hybrid genetic algorithm with genetic and annealing mechanisms can give a full play to the advantages of group search capability in GA and strong local convergence in SA. Thereby the hybrid genetic algorithm has significant effects in function optimization applications.

REFERENCES

[1] Holland, J.H. Adaptation in Natural and Artificial Systems. Ann Arbor. University of Michigan press, 1975.

[2] De Jong K. A. Analysis of the Behavior of a Class of Genetic Adaptive Systems .Michigan: University of Michigan, 1975, pp.76-81.

[3] L.Y. Jia, X. Du. Study of Parallel Genetic Algorithm, Journal of Hunan City University, 2006, 15(3), pp.72-74.

[4] Y.J. Huang, Y.X. Wu, H.B. Liu. Application of Improved Genetic Algorithm in TSP, Computer Engineering and Design, 2007, 28(24), pp.9-11.

[5] S.D. Yang, S.P. Li. Study of Genetic Algorithm, Studies in Network and Information, 2008, 27(9), pp.60-62.

[6] J. Liu, Y. Wang. An Efficient Hybrid Genetic Algorithm, Journal of Henan University, 2002, 30(2), pp.49-53.

[7] Y. Zeng. Application of Improved Genetic Algorithm in Nonlinear Equations, Journal of East China Jiaotong University, 2004, 21(4), pp.39-41.

[8] H.G. Chen. Mechanism of Simulated Annealing Algorithm, Journal of Tongji University (Natural Science), 2004, 32(6), pp.802-804.

[9] X.Q. Du, J.X. Cheng. Study of Robot Strategy on Annealing Evolutionary Algorithm, Computer Technology and Development, 2008, 18(2), pp.101-103.

[10] Q. Wang. Improved Simulated Annealing Algorithm and its Application, Applied Mathematics, 1993, 4(3), pp. 392-397.

[11] J.L. Liu. Improved Simulated Annealing Algorithm of Reactive Power Optimization, Journal of Electric Power, 1998, 13(2), pp.86-89.

[12] W.X. Xing, J.X. Xie. Modern optimization methods, Beijing: Tsinghua University Press, 1999, pp.181-182.

[13] K.J. Zhou, D.B. Li. Product Assembly Sequence Planning Based on Genetic Simulated Annealing Algorithm, Computer Integrated Manufacturing Systems, 2006(7).

[14] Q. Yan, Y.L. Bao. New Genetic Simulated Annealing Algorithm in Logistics Distribution Routing Problem, Computer Application, 2004, 24(3), pp. 261-263.

[15] H.M. Jin, L. Ma. Application of Genetic Annealing Evolutionary Algorithm on Knapsack Problem, Journal of Shanghai University of Technology, 2004, 26(6), pp. 561-564.

[16] T.N. Liu, Y.B. Duan, S. Lei. Analysis of A Hybrid Algorithm and Convergence, Automation Techniques and Applications, 2003, 22(10), pp.4-6.

[17] W. Xie, K.L. Fang. Solving for Global Solution in Non-differentiable Nonlinear Function by Hybrid Genetic Algorithm, Control Theory & Applications, 2000, 17(2), pp.180-183.

[18] K. Chen, J.Y. Ma, H.B. Wen. An Improved Integrated Hybrid Algorithm, Statistics and Decision, 2008, 17, pp.41-43.

[19] L. He. Analysis of Hybrid Algorithm Convergence and Estimation of Convergence Rate, Systems Engineering, 1999, 6(17), pp.64-68.