The Design and Analysis of an Improved Parallel Genetic Algorithm Based on Distributed System

Yan Chen[1,a], Zhimei Li[1,b]

[1]Department of Information Engineering Guilin University of Aerospace Technology, Guilin, China

[a]chen1982yan@163.com, [b] zhimeili2006@163.com

Keywords: genetic algorithm; parallel genetic algorithm; distributed system; migration; adaptive migration strategy;

Abstract. Genetic Algorithm (GA) is a powerful global optimization search algorithm imitating natural selection and genetic mechanism, but it has low search efficiency in the late evolving period. Parallel genetic algorithm (PGA) can improve computational efficiency and accuracy greatly, so it has become one of the main research fields of GA. This paper introduces the procedure of PGA in detail, analyses the migration limitations of traditional PGA, and puts forward an improved coarse-grained PGA based on distributed system, which adopts adaptive migration strategy to evolve. This implementation can fully tap the computing capability of distributed system to improve the convergence speed and ameliorate the population diversity in order to restrain premature convergence. The experiments show that this algorithm not only has faster convergent speed but also has more accurate calculation precision as well as higher parallel speedup.

Introduction

Genetic Algorithm (GA) is a global optimization search algorithm imitating natural selection and genetic mechanism, present first by Holland, a professor of Michigan College, U.S. 1960s, and developed by him and his students later [1]. Compared with the other conventional methods, GA has unique superiority in solving complex problems with the large space, nonlinear and global optimization, so it is widely used in more and more fields at present. Although GA has a powerful quality of global search, it has low search efficiency and the time required to find adequate solutions increases considerably when the scale of optimization problem and the complexity of space searching become larger and larger. As a consequence of this, there have been multiple efforts to make the GA faster, and one of the most promising choices is to use parallel implementations [2]. This paper analyzes the procedure and disadvantage of Parallel Genetic Algorithm (PGA), and then puts forward an improved Parallel Genetic Algorithm (IPGA) based on distributed system, which adopts adaptive migration strategy to evolve. With the development of high speed network technology, multiple computers could collaborate together to attain higher calculating speed and reduce time distinctly [3]. The distributed parallel computing environment in this paper is constructed with several computers (PCs) by a switch. This paper introduces the dynamic migration solution of IPGA in detail, and the experimental results demonstrate the validity of the IPGA to improve parallel speedup and computing efficiency in the distributed system.

Genetic Algorithm

The basic thought of genetic algorithm is to represent the problem domain by a population of individuals which are candidate solutions. The major procedure of GA is as follow:

Step 1. Randomly initialize a population of individuals as the first generation. The individuals are represented by a chromosome that encodes the variables of the problem.

Step 2. Evaluate each individual in the population. Each individual has a numeric fitness value that measures how well the individual fits in the current generation. Thus the fitness value is calculated by an objective function for the particular problem.

Published by Atlantis Press, Paris, France.

Step 3. Select parents of the next generation according to some selection methods. Usually individuals are selected in such a way that fittest individuals have more chances of being chosen.

Step 4. Create new population of individuals, and apply crossover and mutation operators. The crossover operator is responsible for the genotypic information exchange between individuals while mutation can be considered as a secondary search operator related to diversity maintenance.

Step 5. Evaluate the new individuals and perform the above operations until stopping criteria is reached, and the last surviving individual is considered as the optimal solution.

Improved Parallel Genetic Algorithm

PGA in this paper is an algorithm that combines high speed parallelism of computers with the inherent parallelism of GA in order to enhance the solution speed of the population. PGA may be categorized into three different approaches: Master-Slave, fine-grained and coarse-grained. The coarse-grained PGA is also called island PGA or distributed PGA, which is most widely used for its strong adaptability [4]. In coarse-grained PGA each processor runs the GA process on its own deme in parallel most of time and requires less communication between demes, so it is the most indicated model to run on distributed systems. The PGA presented in this paper is coarse-grained PGA based on distributed system.

Traditional parallel genetic algorithm. The procedure of the traditional PGA is as follows:

Step 1. Set parameters for PGA (the max of generation, the crossover rate, the mutation rate, size of population and so on) and design the fitness function according to the practical problem.

Step 2. Generate randomly a population of individuals and divide it into N subpopulations (demes) averagely if the number of PCs in distributed system is N. Each sub-population is located into a processor (a PC in distributed system) which has its own independent evolution process.

Step 3. Each deme uses the same fitness function to evaluate individuals in it and execute the above genetic operators independently, such as selection, crossover, mutation, and recombination and so on.

Step 4. Migrate the best individuals in each sub-population to the neighbors in regular generation intervals. Every several generations migration once with a given migration rate, the best individuals emigrate and the worst individuals are replaced. As migration occurs, information about different areas of the search space is exchanged between demes, which can provide more diversity in the search and thus overcome premature convergence.

Step 5. Terminate the computation if the current generation reached the given maximum of generation; otherwise go to Step 2.

The traditional coarse-grained PGA model is as shown in Fig.1, which uses the ring migration topology generally.

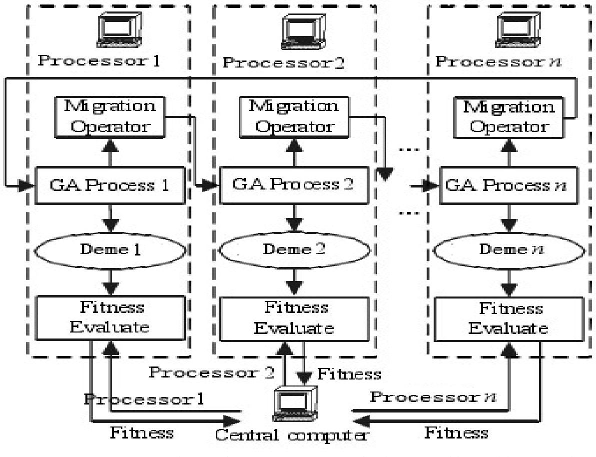

Fig. 1 Coarse-grained PGA model on distributed system

Analysis of the tradional PGA. How to select the parameters to improve the efficiency and accuracy of PGA is a difficult problem for research, especially the migration parameters which play an important role on the performance of PGA. The migration of individuals between demes is controlled by the follow parameters mainly [5]: The topology that defines the connections between the demes; A migration rate that controls how many individuals migrate; A migration scheme that decides which individuals from one deme migrate to another one, and which individuals are replaced; A migration interval that determines the frequency of migrations. The most common migration in traditional PGA uses the fixed topology, such as the above ring topology to migrate the best individuals to replace the worst ones, and defines the fixed migration interval which may be not suitable for too short or too long. Reducing the migration interval between demes in PGA will increase the communication and synchronous cost, which can reduce the speedup and global convergence of PGA to cause get local minimization easily; On the other hand, increasing the migration interval will cause the best individuals can not migrate in time, which generate a negative effect on the solution precision and convergence speed. Although we can get a better regular interval by repetitious experimental tests, that is difficult to keep balance between them.

Improved PGA Based on Adaptive Migration Strategy. In improved PGA (IPGA) of this paper, we choose an adaptive migration strategy, which means the migration can be adjusted dynamically according to the current evolution situation. For a population of individuals P, the individual similarity of P is defined as follow:

$$PIS(P) = \sum_{i=0}^{PopSize} \left(\sum_{j-1}^{PopSize} Fitness(X_j) \Big/ PopSize - Fitness(X_i) \right)^2 \qquad (1)$$

Let $PopSize$ is the size of the population, $Fitness(X)$ is the fitness function, the $PIS(P)$ represents the difference degree of individuals at current generation. When $PIS(P)$ is bigger, the difference of individuals is larger and the individuals remain the population diversity; Otherwise, the difference is smaller and the individuals are about to get convergence.

The migration strategy of IPGA is: calculate the PIS of the current population in each deme at first, when $PIS < Min\_PIS$ the deme will accept the individuals from another deme, which individuals can maximize the PIS to ameliorate the population diversity; when $PIS > Max\_PIS$ the deme will accept the best individuals from another deme in order to speed up the convergence. In order to complete the migration efficiently, each deme sets a migration buffer to temporarily store the individuals for migration and uses critical area to realize mutual exclusion between accepting the individuals to migration buffer and removing the individuals in migration buffer.

The parallel strategy of IPGA is same to the traditional PGA, and the difference between them is the migration operator. The migration procedure of IPGA is as follows:

Step 1. Calculate the $PIS(P)$ of the current generation according to the Eq. 1 in parallel on each deme.

Step 2. Define the Min_PIS and Max_PIS, such as let Min_PIS =0.1, Max_PIS=10.

Step 3. Travel the all subpopulations and store the best individuals in its migration buffer.

Step 4. Judge whether immigration operation is necessary. For the improved PGA in this paper, that is to say, compare the $PIS(P)$ with Min_PIS and Max_PIS, when $PIS(P)$ on any one of demes meet the one of the two conditions, i.e. $PIS < Min\_PIS$ or $PIS > Max\_PIS$, migrate will occur : move the best individuals of a fixed rate in the deme to the migration buffer of other demes, and accept the individuals of same number from other demes to replace the worst individuals to generate next generation. This can keep the total number of individuals in each subpopulation unchanged. The migration of best individuals between the demes is help to develop the guidance of best individuals to the full and improve calculation precision as well as convergence speed.

Fig.2 is the flow chart of the improve PGA of our research:

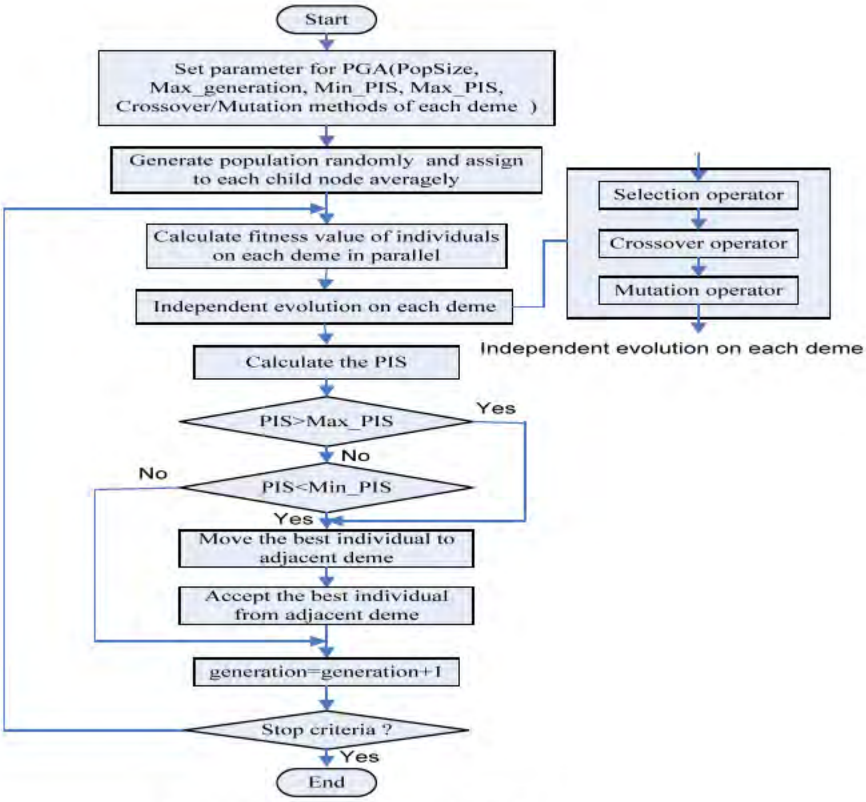

Fig. 2 The flow chart of IPGA

The selection of IPGA in this paper is based on the mechanism of nature elimination whereby superior individuals will prosper and inferior ones is eliminated. Firstly calculate the fitness values of each individuals of subpopulation P, set $P(t)=\{X_1,X_{2,}X_{PopSize}\}$; Sort the fitness values according to the ascending order of them; Determine the best and worst individuals, choose K best individuals, meanwhile choose M individuals from the remaining $PopSize-K$ individuals randomly. The M individuals organize the individuals for crossover together.

Simulation Experiments

In this paper we use the IPGA to solve the problem described in reference [6,7], which is thermal properties identification of material samples subject on known boundary conditions and several temperature measurements inside the domain. In our experiments, let $PopSize$=1000, the crossover rate=0.8, the mutation rate=0.2, Min_PIS =0.1, Max_PIS =10, $Max\_Generation$ =1000, the aimed heat conduction parameter $\bar{\theta}$ =(A,B,C,D,E,F,G) = (1.0,0.1,0.01, 0.001,1.0,0.1, 0.01), the fitness function Fitness $Fitness(X)=(Tm_x-Tc_x)^2$, where Tm_x is the actual temperature value and Tc_x is the calculate value separately. We execute the IPGA steps based on distributed system of 8 PCs installed with LAM/MPI and Red Hat Linux in local area network. Table 1 shows the numerical datas for the conduction parameter $\bar{\theta}$ calculated by the IPGA and traditional PGA.

Table 1 simulated values of experiments

	A	B	C	D	E	F	G
Real value	1.0	0.1	0.01	0.001	1.0	0.1	0.01
Simulated value(PGA)	1.0194	0.0982	0.00098	0.00083	0.9865	0.10320.	00949
Simulated value(IPGA)	1.0065	0.0981	0.01019	0.00094	0.9886	0.10252	0.00968
Relative error(PGA)	1.94%	1.8%	2.0%	6.0%	1.35%	3.2%	5.1%
Relative error (IPGA)	0.65%	1.87%	1.9%	4.0%	1.14%	2.52%	3.20%

Form table 1, we can see that the IPGA can get better precision than traditional PGA. The performance comparison about speedup is showed in Fig.3. Although the distinction is not very obvious in figure 3, the speedup of IPGA is still better than the traditional PGA. This indicates that

Published by Atlantis Press, Paris, France.

the improved PGA base on adaptive migration strategy in this paper can help to improve computational efficiency and accuracy.

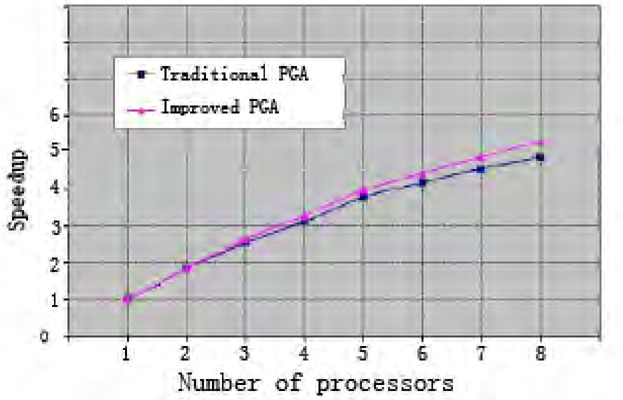

Fig.3 The Speedup of IPGA with number of processors

Conclusion

The paper presents an improved Parallel Genetic Algorithm based on distributed system, which adopts adaptive strategy to execute migration among the demes. Firstly, it introduces the procedure of traditional PGA, analyzed the disadvantage of the migration topology of it, designed an adaptive migration strategy based on traditional PGA, and then described the migration steps in detail. At last, it is proved to be efficient by the experiment results.

Acknowledgment

This work is supported by a research project of Guilin University of Aerospace Technology (No.X10Z006).

References

[1] Chen Guolian, Genetic algorithm and it’s application. The people’s mail press (1996).

[2] E.Cantu-Paz, A survey of parallel genetic algorithms, Illinois Genetic Algorithms Lab. IlliGAL Report, No.97003 (1997).

[3] Wei Yang, Qingming Wu, et al, Nesting System for Cutting Stock Problem Based on Distributed Parallel Genetic Algorithm. DCABES Proceedings (2007), p.63-66

[4] Iorio A, X Li, Parameter control within a co-operative co-evolutionary genetic algorithm. In Proceedings of the Seventh Conference on Parallel Problem Solving From Nature (2002), p.247-256

[5] Nowostawski, M. Poli, R, Parallel Genetic Algorithm Taxonomy, Knowledge-Based Intelligent Information Engineering Systems, 1999. Third International Conference(1999), p.88-92

[6]Chen Yan, Ma Li, Guo Qingping, A Parallel GA- BP Neural Newtwork Algorithm for Solving Inverse Heat Conduction Problem .2011 International Conference on Electronic & Mechanical Engineering and Information Technology. Vol.9(2011), p. 4622-4625.

[7] Guo Qingping，Y. Paker et al, Parallel Multi-grid Algorithm with Virtual Boundary Forecast Domain Decomposition Method for Solving Non-linear Heat Transfer Equation，Lecture Notes in Computer Science, High Performance Computing and Networking, Spreinge Press, vol.1823(2000), p.568-571.