ISSN : 0974 - 7435
Volume 10 Issue 12
2014
BioTechnology
An Indian Journal
FULL PAPER
BTAIJ, 10(12), 2014 [6756-6763]

The novel genetic algorithm to the grid resource assignment problem

Guo-Ping Zou
School of Mathematics and Computer Sciences, Xinyu University, Xinyu 338003, (P.R.CHINA)
E-mail: zougp2999@126.com

ABSTRACT

In the fields of science, engineering and commercial computer, there are many problems that we cannot use existing super computer to solve. For this reason, a novel genetic algorithm is proposed to the grid resource assignment problem. The improved genetic algorithm extracts the heuristic feedback from these obtained solutions, and applies the heuristic feedback to guide the subsequent optimization process. Experimental results suggest that this approach is efficient to the resource optimization problem.

KEYWORDS

Genetic algorithm; Grid resource; Assignment.

INTRODUCTION

In the fields of science, engineering and commercial computer, there are many problems that we cannot use existing super computer to solve, such as mass data resources that need be processed are distributed at different geographic areas and the special computing facilities as well as input and output equipment that are needed are not local etc. It is difficult for practical service condition of computer to satisfy people's demands for computing power. On the one hand, a number of computer resources are in an idle state and are not utilized effectively. On the other hand, many application problems cannot be solved because there are no sufficient computing recourses for use.

With people's endeavor in this aspect, techniques like parallel technique, cluster technique and distributed technology are generated. However, these techniques can only help people use computing resources within a certain scope which is the range of management domain. Nevertheless, resources that use these techniques and can be shared are limited and overall strength of aggregation is not powerful enough. As Internet and Web technique become mature as well as are popularized and applied, people have the idea that internet resources can be integrated for use and they want to use existing internet facilities to establish a kind of new infrastructure, integrate all kinds of computing resources and provide good interfaces when these resources are used for users in the whole world. This new infrastructure is Grid.

Grid utilizes existing internet framework, integrates all kinds of resources which are widely distributed geographically, including computing resources, storage resources, bandwidth resources, software resources, data resources, information resources and knowledge resources, into a logic whole (or called a 'virtual super computer'), provides users with application service like integrative information, computing, storage and visit, realizes resource sharing and cooperative work ultimately and eliminates resource 'islet' completely. The virtual super computer organized in this way has two advantages, i.e., super-strong data handling capacity and ability to make full use of idle resources on internet. Currently, grid technology has become a hot spot and frontier domain of domestic and foreign researches so that it is praised as the third information technology wave after internet and Web.

PROBLEM FORMULATION

Currently, researches on grid have been popularized to all continents of the world from America and Europe. As a result, all countries and regions have input a great deal of money to carry out researches on grid technology and infrastructure construction of grid. America is the womb of researches on grid technology as well as a country taking the lead in the field of researches on grid technology. Since the middle of 1990s, organizations and departments of America, such as National Science Foundation (NSF) of America, National Aeronautics and Space Administration (NASA), and American army have successively input a great deal of money into research project grid in their own field, such as TeraGrid project funded by NSF and IPG (Information Power Grid) supported by NASA.

Activity is the fundamental element in the practical engineering, suppose there are totally N activates and M persons. Suppose that T_{di} , T_{di}^{c} and T_{di}^{n} denote the period of activities, the time remainder of activities and the actual completion time of activities Respectively. $Role_{i,j}$ means the i^{th} activity can be done by the j^{th} person. $RoleNum_i$ is the completion person number of activities. T_D and T_N are the planned completion time and the actual completion time of projects. C_D and C_N denote the planned cost and the actual cost of projects. $f_T(EM)$ and $f_C(EM)$ denote the completion date function and the cost function of projects.

The cost of software process contains: management fee and development fee. The former is daily cost C_1 which maintaining software development and cost of administrators C_2 . The latter points at cost of device resources C_3 and development cost C_4 of different developers with various abilities and labor-hours.

$$\begin{cases} CN = C_1 + C_2 + C_3 + C_4 \\ C_4 = \sum_{i=1}^{M} c_i T_i \end{cases} \quad (1)$$

Here, c_i and T_i denote the development cost and labor-hour of developers respectively. Cost and construction period is the main attributes of software process. Assume the sum weight of cost and period as optimized objective function, the definition of which is as shown in the following equation. In which, C means cost, T signifies the weight value of construction period x_1, x_2 determined by demand of decision makers, EM is the executive matrix.

$$\max Fitness(C,T) = x_1 * C + x_2 * T$$
$$s.t. \begin{cases} C = f_C(EM) \\ T = f_T(EM) \\ 0 \le x_1 \le 1 \\ 0 \le x_2 \le 1 \\ x_1 + x_2 = 1 \end{cases}$$

IMPROVED GENETIC ALGORITHM

The improved genetic algorithm is characterized by the extraction and application of heuristic feedback in the whole evolution process. In this paper, the near-optimal solutions obtained throughout the search are analyzed to extract the heuristic feedback, and then the obtained heuristic feedback is used to guide the subsequent search. The basic framework of IGA is displayed as Figure 1. The computational flow of IGA is shown in Figure 2.

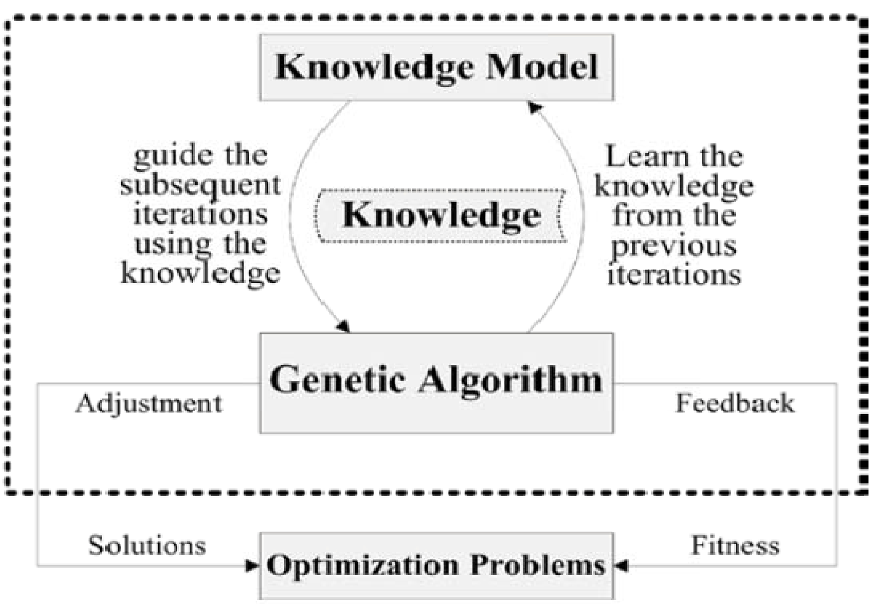

Figure 1: The basic framework of IGA

(1) Heuristic Feedback. The first kind of heuristic feedback is called the activity assignment position which is applied to establish a beneficial order for the given activity. A matrix HF_1 with size $N \times N$ is defined for the activity assignment position, $HF_1(i, j)$ denotes the total number of times of assigning the activity i to the j^{th} position among the near-optimal solutions obtained throughout the search. The second

kind of heuristic feedback is called the activity assignment person which is applied to establish the beneficial person for one given activity. A matrix HF_2 with size $N \times M$ is defined for the activity assignment person, $HF_2(i, j)$ denotes the total number of times of assigning the activity i to the j^{th} person among the near-optimal solutions obtained throughout the search.

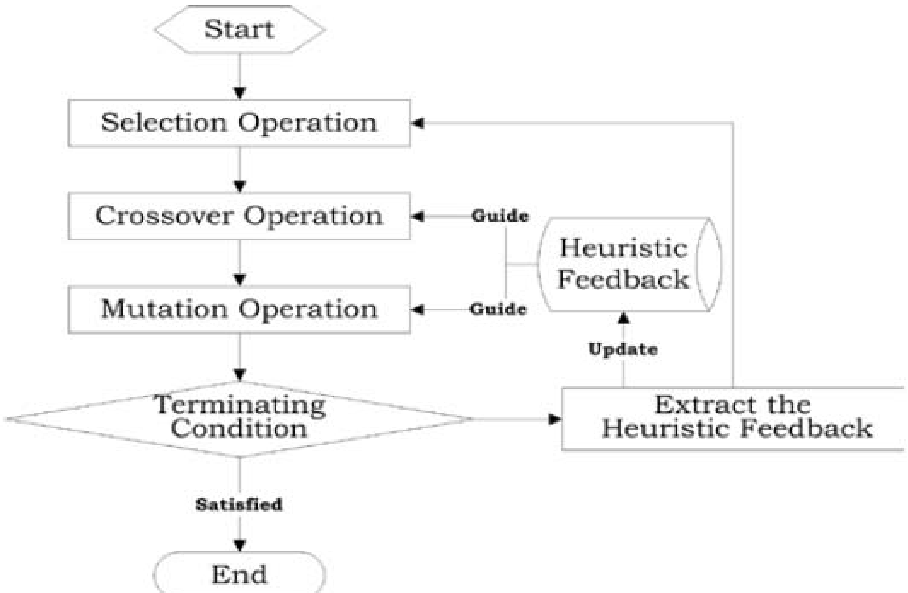

Figure 2 The computational flow of IGA

(2) Application of Heuristic Feedback. In IGA, the activity assignment position is applied to guide the crossover operation. The activity assignment position is employed to determine one beneficial position for the given activity. To the activity assignment position matrix displayed in TABLE 1, if we want to determine the beneficial position for activity 3, then we can obtain the following probabilities, and the beneficial position to activity 3 is decided by a random way with the following probability distribution.

TABLE 1: An example of activity assignemment position

3	5	4	2	1	0
1	5	6	2	1	0
0	1	2	8	2	2
0	2	2	6	5	1
0	0	4	5	6	0
0	0	2	3	4	6

$$\text{Position } 1: \frac{0}{1+2+2+8+2} = 0$$

$$\text{Position } 2: \frac{1}{1+2+2+8+2} = 0.07$$

$$\text{Position } 3: \frac{2}{1+2+2+8+2} = 0.13$$

$$\text{Position } 4: \frac{8}{1+2+2+8+2} = 0.53$$

$$\text{Position } 5: \frac{2}{1+2+2+8+2} = 0.13$$

$$\text{Position } 6: \frac{2}{1+2+2+8+2} = 0.13$$

In IGA, the activity assignment person is applied to guide the mutation operation. The activity assignment person is employed to determine one beneficial person for the given activity. To the activity assignment person matrix displayed in TABLE 2, if we want to determine the beneficial person for activity 6, then we can obtain the following probabilities, and the beneficial person to activity 6 is decided by a random way with the following probability distribution.

TABLE 2: An example of activity assignemment person

4	2	5	2	4	6
5	7	5	11	6	6
6	6	5	2	5	3

$$\text{Person } 1: \frac{6}{6+6+3} = 0.40$$
$$\text{Person } 2: \frac{6}{6+6+3} = 0.40$$
$$\text{Person } 3: \frac{3}{6+6+3} = 0.20$$

(3) Updating of Heuristic Feedback. After each generation, if the global optimal solution (the best solution from the start) was obtained at this iterative, then the knowledge level will be updated by the following rule, which is based on the optimal solution to accomplish knowledge updating. If the activity i to the j^{th} position among the best solution, then

$$HF_1(i,j) = HF_1(i,j) + 1 \tag{2}$$

If the activity i to the j^{th} person among the best solution, then

$$HF_2(i,j) = HF_2(i,j) + 1 \tag{3}$$

NUMERICAL EXPERIMENTAL RESULTS

The IGA was implemented using Visual C++ language, and executed on a personal computer with the 2 GHz processor and 2GB memory. In this paper, the final experimental results were averaged over 30 trials, and 10 testing instances were randomly produced to validate the performance of our approach. The optimal objectives obtained by the IGA are summarized in TABLE 3. From the experimental results of TABLE 3, we can see that, there exists the small gap among different trials. Experimental results suggest that it is efficient to the given problem.

TABLE 3: The final experimental results

SN	*N*	*M*	*Cost*		*Time*	
			Avg.	*Std.*	*Avg.*	*Std.*
1	20	5	55.6	0.28	22.9	0.15
2	20	5	86.9	0.15	25.6	0.19
3	20	5	75.3	0.31	26.7	0.21
4	50	10	128.6	1.26	42.5	0.59
5	50	10	135.6	2.14	43.8	0.61
6	50	10	129.4	2.09	46.9	0.63
7	200	30	658.1	12.58	59.8	1.25
8	200	30	682.7	15.49	61.2	1.38
9	200	30	648.2	10.87	62.7	1.09
10	200	30	684.1	11.68	59.4	1.53

ACTUAL EXPERIMENTAL RESULTS

Random sampling of user type means n units are directly selected as samples from N units in all according to random principles and requires that each unit should has the same chance to be selected in the process of sampling.

In the tests of this section, methods of random sampling are shown as follows. Firstly, 100 users are generated, i.e., N=100, and ρ each user's user-satisfaction tendency index is generated randomly within the interval [0, 1]. Then, n users are randomly selected from these 100 users (n≤100).

The random sampling method is used to select one, two, and three … and ten users to implement a simulation experiment on these ten situations, respectively. In each test, task submitted by each user is decomposed into 20 unequal-sized tasks randomly and communication traffic among tasks will be generated randomly. Figure 3-6 describe curves about changes in Makespan, cost, load balance and service quality of the three algorithms Min-Min, SGA and IGA under ten situations.

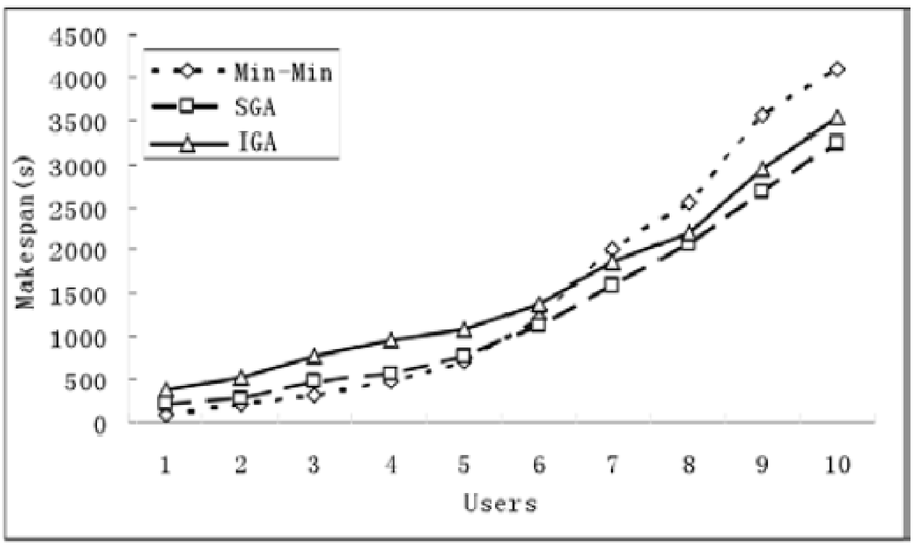

Figure 3: Curve of optimized makespan

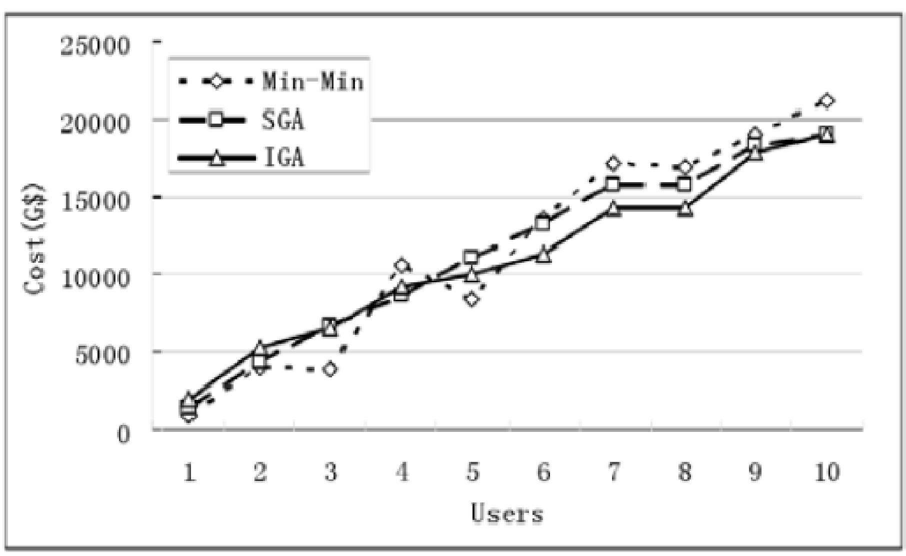

Figure 4: Curve of cost

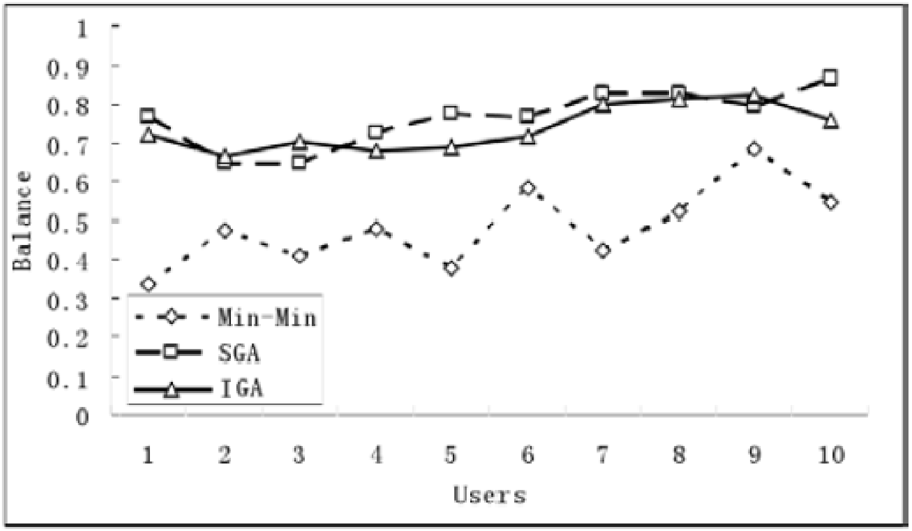

Figure 5: Curve of load balancing

According to experimental results, it is shown that:

In the aspect of Makespan the optimal span, optimal span performance of Min-Min algorithm is superior to that of SGA and IGA when the number of users is small because the number of dispatched tasks is small. However, as the number of uses increases and the number of dispatched tasks grows, the optimal span performance of Min-Min algorithm reduces sharply, while performance of SGA and IGA are much better than that of Min-Min algorithm when dispatching scale is large because their scheduling strategies can plan matters as a whole. In detail, since SGA merely regards time span as the only basis of heritable variation but individual fitness evaluation of IGA needs weigh two factors including time span and cost simultaneously, the optimal span of IGA is a little weaker than that of SGA.

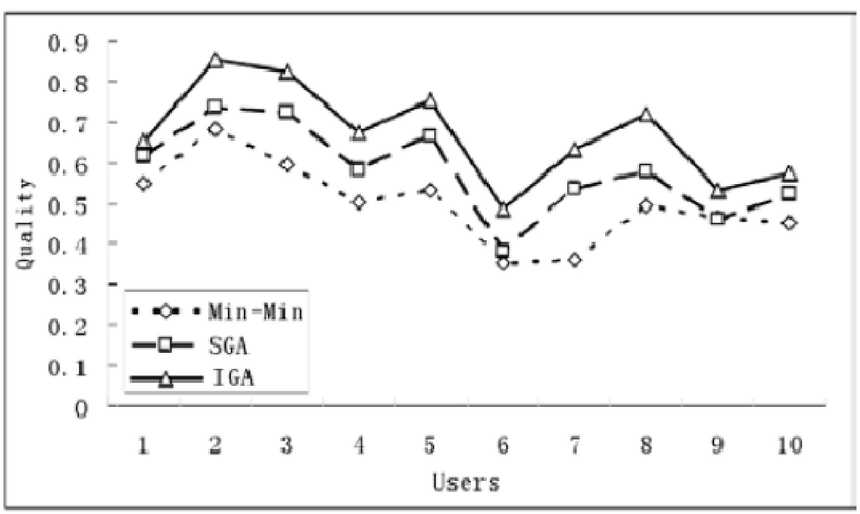

Figure 6: Curve of quality

With respect to cost, since Min-Min algorithm and SGA algorithm themselves do not consider this factor, their cost is higher than that of IGA algorithm.

Considering load balance, because of features, i.e., dispatching thoughts of Min-Min algorithm itself, its performance of load balance is bad. In addition, its dispatching process is significantly affected by the correlation between size of dispatching tasks and tasks. Thus, the load balance of dispatching fluctuates largely and is extremely unstable. Because of diversification of heritable variation population, balance of IGA and SGA in the aspect of resource allocation is obviously superior to that of Min-Min algorithm, which shows superiority of genetic algorithm (GA).

In the aspect of service quality, this thesis carries out measurement according to level of user satisfaction. It is found that its value is affected by time span and cost simultaneously. When random sampling is implemented, user satisfaction fluctuates largely because user type (i.e., user-satisfaction tendency index) has uncertainty.

CONCLUSION

The contribution of this paper can be summarized as follows: A novel genetic algorithm is proposed to the resource optimization problem. Experimental results suggest that it is efficient to the given problem.

CONFLICT OF INTERESTS

The authors declare that there is no conflict of interests regarding the publication of this article.

REFERENCES

[1] N.B.Ho, J.C.Tay, E.M.K.Lai; An Effective Architecture for Learning and Evolving Flexible Job-Shop Schedules, Eur.J.Oper.Res., **179(2)**, 316-333 **(2007)**.

[2] S.J.Louis, J.McDonnell; Learning with Case-Injected Genetic Algorithms, IEEE Trans.on Evo.Compu., **8(4)**, 316-328 **(2004)**.

[3] L.N.Xing, Y.W.Chen, H.P.Cai; An intelligent genetic algorithm designed for global optimization of multi-minima functions, Appl.Math.Comput., **178(2)**, 355-371 **(2006a)**.

[4] L.N.Xing, Y.W.Chen, X.S.Shen; A Constraint Satisfaction Adaptive Neural Network with Dynamic Model for Job-Shop Scheduling Problem, Lect.Notes Comput.Sci., **3973**, 927-932 **(2006b)**.

[5] L.N.Xing, Y.W.Chen, X.S.Shen; Multiprogramming Genetic Algorithm for Optimization Problems with Permutation Property, Appl.Math.Comput., **185(1)**, 473-483 **(2007)**.

[6] L.N.Xing, Y.W.Chen, K.W.Yang; A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problema, Eng.Appl.Artif.Intel., **21(8)**, 1370-1380 **(2008a)**.

[7] L.N.Xing, Y.W.Chen, K.W.Yang; Double Layer Ant Colony Optimization for Multi-objective Flexible Job Shop Scheduling Problems, New Generat.Comput., **26(4)**, 313-327 **(2008b)**.

[8] L.N.Xing, Y.W.Chen, K.W.Yang; An Efficient Search Method for Multi-objective Flexible Job Shop Scheduling Problems, J.Intell.Manuf., **20(3)**, 283-293 **(2009)**.

[9] L.N.Xing, R.Philipp, Y.W.Chen et al.; An Evolutionary Approach to the Multi-depot Capacitated Arc Routing Problem, IEEE Trans.on Evo.Compu., **14(3)**, 356-374 **(2010a)**.

[10] L.N.Xing, Y.W.Chen, P.Wang; A Knowledge-based Ant Colony Optimization for Flexible Job Shop Scheduling Problems, Appl.Soft Comput., **10(3)**, 888-896 **(2010b)**.