International Journal of Hybrid Information Technology
Vol.8, No.8 (2015), pp.35-46
http://dx.doi.org/10.14257/ijhit.2015.8.8.03

Task Scheduling Model of Cloud Computing based on Firefly Algorithm

Jichao Hu[1] and Yue Fu[1]

[1]*Department of Information Engineering, Shijiazhuang University of Economics, Shijiazhuang Hebei, P. R. China 050031*
hujichao001@163.com

Abstract

We proposed a task scheduling in cloud computing based on intelligence firefly algorithm aimed at the disadvantages of cloud computing task scheduling. Firstly, on the basis of cloud model, used intelligence firefly algorithm with strong ability of global searching to find the better solution of cloud computing task scheduling then turned the better solution into the initial pheromone of improved firefly algorithm, and found out the cloud computing task scheduling and the algorithm's global optimal solution through improved firefly information communications and feedbacks. Finally, made comparison test of the three benchmark function on the basis of MATLAB, the results showed, compared with traditional intelligence firefly algorithms, the improved algorithm can preferably allocate the resources in cloud computing model, the effect of prediction model time is more close to actual time, can efficiently limit the possibility of falling into local convergence, the optimal solution's time of objective function value is shorten which meet the user's needs more.

Keywords: *Cloud Computing, Network Computing, Firefly Algorithm, Task Scheduling*

1. Introduction

Cloud computing is a now wildly used architecture hot, it's product of the development of grid computing, distributed computing, network storage and parallel processing [1]. It shows that the user's applications can operate without personal computer but the server cluster in the Internet. There are three basic forms of cloud computing services including: Infrastructure as a Service (IAAS), Platform as a Service (PAAS) and Software as a Service (SAAS) [2]. In cloud computing, the allocation of resources is a very important issue, the unsatisfactory allocation of resources can easily led the cloud's servers crashed and other servers in idle. So in cloud environment, the problem mostly need to solve is the ways to control any server's resources allocation and use condition by the information communication of local and in the Internet to make better use of the resources. Literature [3] made researches of the resources allocation conditions in different environment. Literature [4] proposed the resources allocation mechanism of self-management, self-adjustment and self-protection. Literature [5, 6] proposed a resources allocation system applies to extensive distributed system, which efficiently increased the system's service quality under cloud computing.

Cloud computing is a combination of parallel computing, distributed computing and virtual technology, a hot technology of nowadays computer industry. The cloud system firstly combined computer, storage device and so on and formed resources pool, then the users could choose the corresponding resources by their needs, this dynamically offers users a computing service environment with reliable and ensure quality of service(QOS). Task scheduling is one of the core technologies of cloud computing which has big effect on the whole performance of cloud computing [7].

ISSN: 1738-9968 IJHIT
Copyright ⓒ 2015 SERSC

Aimed at the task scheduling of cloud computing, a scholar came up with a scheduling algorithm of HADOOP, this algorithm scheduled by task priority and submission time, which was easy to realize but it ignored the difference in tasks and made a long response time of the tasks [8]. Later some scholars proposed the cloud computing task scheduling algorithm with optimal efficiency; the task scheduling algorithm based on cost-driven; the task scheduling algorithm based on trust-driven, those algorithms is flexible and better meet the needs of the users, but they all aimed at a certain object so it's limited in application. A big amount of researches showed, cloud computing used in multiple task scheduling is not only reliable, quality of service (QOS) ensured but also be fair in task allocation, cloud computing multiple task scheduling is a world-recognized NP problem [9]. It's very complex to solve with exhaustive search method, with swarm intelligence algorithm became mature, in recent years, some scholars brought particle swarm optimal algorithm, genetic algorithm and ant colony optimization into the task scheduling of cloud computing, which has a good result [10-14]. But as the isomerism, dynamics and the differences of users of cloud computing, the cloud task scheduling is a complex problem, single genetic algorithm or ant colony algorithm both has its shortage, and the combinational algorithm based on combination-optimization theory have complementary advantages then find the optimal solution of the problem [15-17].

In this text, in order to improve the task scheduling quality of cloud computing, proposed a improved firefly algorithm cloud computing task scheduling research. Firstly on the basis of cloud model, uses the intelligence firefly algorithm with strong ability of global searching to find the better solution of cloud task scheduling fast and turns it into initial pheromone of improved firefly algorithm. Finally, made comparison test of the three benchmark function on the basis of MATLAB, the results showed, compared with traditional intelligence firefly algorithms, the improved algorithm can preferably allocate the resources in cloud computing model, the effect of prediction model time is more close to actual time, can efficiently limit the possibility of falling into local convergence, the optimal solution's time of objective function value is shorten which meet the user's needs more.

2. Problem Description

2.1. Basic Knowledge of Cloud Model

The cloud model is a transformation model uses linguistic values to express the uncertainty between a certain conception and its quantification expression, it fully combines fuzziness and randomness and forms the mapping between qualitative and quantification, shows in Figure 1.

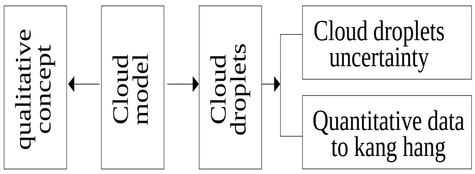


Figure 1. Transformation Schematic of Cloud Mode's Qualitative Concept and Quantitative Data

Sets U is discourse domain expressed by accurate numerical value, A is corresponding qualitative concept in U. If quantitative value $x \in u$ and x is a random implementation with likely normal distribution of qualitative concept A in discourse domain U, the certainty degree $A(x) \in [0,1]$ of x to A is also a random number with likely normal distribution, then data array $(x, A(x_i))$ is called as cloud drop, the whole

 Copyright ⓒ 2015 SERSC

element $x_i\ (i = 1, 2 \ldots n)$ in discourse domain U and its certainty degree $A(x_i)$ for A, i, e, n data array $(x, A(x_i))$, forms the cloud model with n cloud drop, calls x distribution in discourse domain U as cloud distribution. The number characteristics of cloud model are expressed as expectation (Ex), entropy (En) and excess entropy (He). Among them, expectation (Ex) refers to the central value of discourse domain U, is the center of qualitative concept, reflects the cloud focus of the whole cloud drop swarm; entropy (En) refers to the range which can be received by fuzzy concept, $En \succ 0$; excess entropy (He) is a uncertain measurement of entropy, i, e, the excess entropy is the entropy's entropy, He >0. The excess entropy reflects the degree of reach an agreement of cloud drop of representation qualitative concept or the concentration degree of cloud drop's representation qualitative concept; the bigger excess entropy is, the qualitative concept has worse common sense or the qualitative concept is worse decentralization.

2.2. Intelligence Firefly Algorithm

Intelligence firefly algorithm, proposed by KRISHNANAD, *etc.*, in 2005, it's a new intelligence swarm optimal algorithm, this algorithm is wildly used in producing and scheduling, its simulated the search and optimize process to the firefly's attraction and migration, measured the advantages and disadvantages of the individual's position by solving the objective function. In this algorithm, each intelligence firefly distributes in the declaration space of objective function, this intelligence firefly has its field of view and carries fluorescent powder, the brightness of fireflies is related to its position and the fitness value of objective function, the brighter position shows the firefly there has pretty objective value and it can attract more fireflies to move towards this direction, as each firefly has its own range of view, the range can be affected by the neighbor fireflies, when the number of firefly becomes fewer, the range of view is larger and attracts more fireflies. When the fireflies are more, the range of view becomes smaller. But at last the position which most of the fireflies in is the optimal solution position.

Suppose the firefly swarm id N, the i firefly's position (x_i, y_i) matches the objective function $f((x_i, y_i))$ and the firefly's fluoresce in value is T_i, the updating formula of each firefly's range of view is:

$$f_k^i(u+1) = \min\left\{ f_t, \max\left\{0, f_k^i(u) + \beta\left(t_u - \left|t_u(u)\right|\right)\right\}\right\} \tag{1}$$

Among them, $f_k^i(u+1)$ is the COOLEYE of i firefly in $u+1$ range of view, t_u is the threshold value of the neighbor firefly's number. β is control constant , $t_u(u)$ is the number of firefly with high fluoresce in range of view. There, the formula of $t_u(u)$ is as following:

$$t_i(u) = \left\{ j : \left\| y_j(u) - y_i(u) \right\| \prec f_k^i l_i(t) \prec l_j(u) \right\} \tag{2}$$

There, $y_j(u)$ is the position of j firefly in t generation, $l_j(u)$ is the j firefly's fluoresce in value in t generation, the view of range between neighbor firefly is in f_k^i.

The firefly neighbor's selecting probability is:

$$f_{ij}(u) = \frac{l_i(u) - l_j(u)}{\sum_{k \in n_i(u)} l_k(u) - l_i(u)} \tag{3}$$

The position updating formula of firefly:

$$f_i(u) = f_i(u-1) + s\frac{f_j(u-1) - f_i(u-1)}{\left\| f_j(u-1) - f_i(u-1) \right\|} \tag{4}$$

Fluoresce in value's formula:

$$f_i(u+1) = (1-t)l_i(u) + \gamma k(c_i(u+1)) \tag{5}$$

In formula (5), γ is a parameter to measure the function value, $k(c_i(u+1))$ is the fitness value of the function.

In nature, supposes firefly i enters into the view range of j, and the fluorescence in value is bigger than itself. Firefly i chooses firefly j according to probability $t_{ij}(u)$, after choosing, the position of firefly i and the fluorescence in is updating, calculates the objective function value of this position.

3. Measurement of Cloud Model'S Spray Characteristic

The cloud model's spray characteristic refers to the character of cloud drop distributes around cloud expectation curve's discrete degree. Professor Liuyu, *etc.*, made researches on excess entropy measures cloud drop's discrete degree with fixed entropy. But these works did not show the essence factors of determine cloud model's spray characteristic, *i. e*, the standard deviation Y's distribution of cloud drop quantitative data X determines the cloud model's spray characteristic. The same as the cloud distribution probability density of cloud model algorithm identified is the theoretical basis of uncertainty reverse cloud model algorithm, this chapter revised the cloud distribution probability density and gave a strict proof according to spray characteristic Y >0.

The positive direction cloud model algorithm steps in one-dimension theory's domain are as following:

Step 1: Generates normal random number y_i whose expectation is E_n, standard deviation is He;

Step 2: Generates normal random number x_i whose expectation is Ex, standard deviation is y_i, x_i is a concrete and quantitative realize of qualitative concept A operates in its corresponding quantitative theory of the domain U, called cloud drop qualitative data;

Step 3: Calculates $r_i = \exp\left(-\frac{(x_i - E_x)^2}{2y_i^2}\right)$, r_i is the certainty degree or subjection degree of x_i belongs to qualitative concept A;

Step 4: Repeats step one to three until generates n cloud.

Prove: because $y - r(E_n, H_e^2)$, En refers to the discourse domain must be greater than zero, as $x \sim N(E_x, y^2)$, y, as the standard deviation of x, must be greater than zero, so according to normal distribution random variable meets 3σ rule, gets $E_n / H_e \geq 3$. Besides, the probability density of Y is

$$x_i(y) = \frac{1}{\sqrt{2\pi H_e}} \exp\left[-\frac{(t-E_n)}{2He^2}\right]$$

When $x_i = y$, the conditional probability density is

$$x_{i,j}(x|y) = \frac{1}{\sqrt{2\pi y}} \exp\left[-\frac{(x-E_x)^2}{2y^2}\right]$$

Gets joint probability density through conditional probability density formula:

$$x(i,j) = \frac{1}{2\pi H_e j} - \exp\left[-\frac{(j-E_n)^2}{2He^2} - \frac{(i-Ei)^2}{2j^2}\right]$$

Gets probability density which marginal probability density is cloud distribution through joint probability density formula:

$$x_i(x) = \int_x^y \frac{1}{2\pi Hey} \exp\left[-\frac{(y-En)}{2He} - \frac{(x-E_x)}{2y^2}\right]$$

 Copyright ⓒ 2015 SERSC

This formula has no analytic form Quod x_i demonstrandum.

From step 2, 3, y is the standard deviation of cloud drop qualitative data X, its distribution character directly determines the cloud drop's distribution character, the bigger distribution scale of Y, the more cloud drop distributes discrete. Because

$$Y \sim N(En, He^2)$$

This text takes $a = En / He$ as the measurement of cloud drop's discrete degree, called spray factor, because qualitative data's standard deviation Y, En and He must be greater than zero at the same time so a ≥3. Spray factor a integrative considers the nature that standard deviation Y of cloud drop's qualitative data X must be greater than zero, the distribution of Y directly affects cloud drop discrete degree and a determines the distribution character of Y, so 0.0 can be the significant digital characteristic of cloud model to presents the discrete condition of cloud drop's distribution. The spray characteristic of cloud model has the following characters:

Character 1: The distribution characteristics of cloud drop's qualitative data standard deviation determines the cloud drop's distribution characteristics, a refers to the cloud drop's discrete degree and $a \geq 3$. The smaller a be, the bigger discrete degree of cloud drop's distribution; when $\alpha = 3$, the discrete degree of cloud drop's distribution reaches the biggest; the bigger α is, the smaller discrete degree of cloud drop's distribution, finally tends to normal distribution. Now the cloud drop all approximate distributes on cloud expectation curve.

Character 2: cloud distribution's corresponding range of spray factor: $3 \leq a \leq 18$.

The spray factor determines the distribution characters of cloud drop qualitative data, and the kurtosis describe the figure of data distribution at the same time, the kurtosis of normal distribution is 3, if the kurtosis of cloud distribution values around 3, the cloud distribution turns to normal distribution[8]. The kurtosis of cloud distribution defines as following:

Definition 1: the kurtosis of cloud distribution

$$K(X) = 9 - \frac{6}{\left(1 + \frac{He^2}{En^2}\right)^2} = 9 - \frac{6}{\left(1 + \frac{1}{\alpha^2}\right)^2}$$

From this formula, spray factor determines the transformation between cloud distribution and normal distribution. When a =18, the kurtosis of cloud distribution id 3.036, draws its cloud drop distribution as Figure 2. From the figure, when cloud distribution approximate degrades into normal distribution. i, e , essentially, when spray factor meets 3≤a ≤18, the distribution of cloud drop's qualitative data can be called as cloud distribution. So in later discussion, can only considers the condition when spray factor meets 3≤a ≤18.

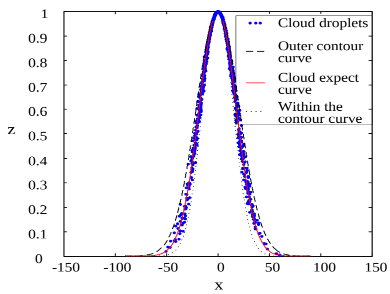
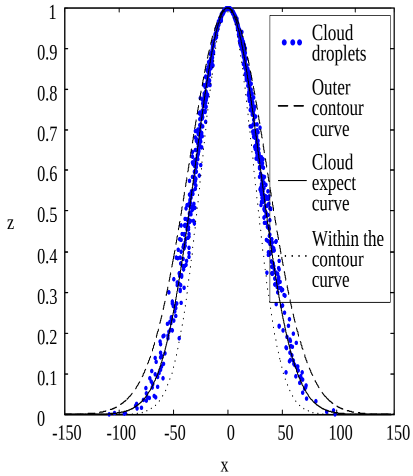


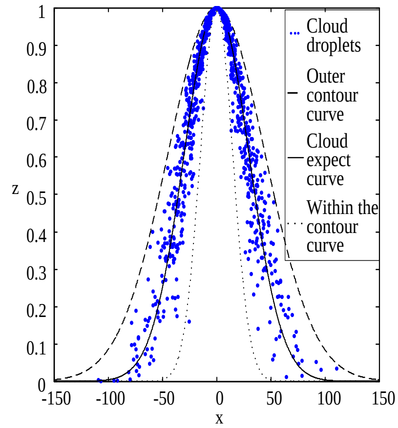
Figure 2. The Cloud Drop's Distribution Map when a =18

Character 3: From the distribution of Y , gets 99.7% cloud drop distributes between curve $f_1 = \exp\left[\frac{-(x-Ex)^2}{2(En+3He)^2}\right]$ and $f_2 = \exp\left[\frac{-(x-Ex)^2}{2(En-3He)^2}\right]$. F 1 is cloud mode's outer contour curve, f2 is inner contour curve.

The experiment result of cloud model's spray characteristic is showed as Figure 3. From this figure, the spray factor of (a) is greater than (b), the cloud drop discrete degree of (a) is obviously smaller than (b). The spray factor of (b) and (c) are being the same, their cloud drop discrete degrees are obviously being the same. The experiment showed, spray factor can reflects the discrete degree of cloud drop's distribution, no matter how cloud model entropy and excess entropy values, if only their spray factors are the same, the cloud model's cloud drop discrete degrees are the same. Besides, expectation (Ex) only affects the whole position of cloud drop's distribution without affects cloud drop's discrete degree.

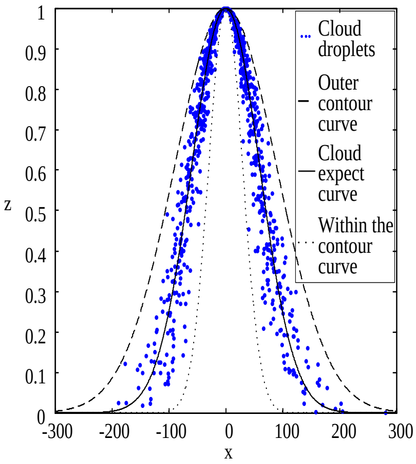


(a) $Ex = 0\ En = 30\ He = 2\ \alpha = 15$



(b) $Ex = 0\ En = 30\ He = 5\ \alpha = 6$

 Copyright ⓒ 2015 SERSC



(c) $Ex = 0$ $En = 60$ $He = 10$ $\alpha = 6$

Figure 3. Cloud Drop's Discrete Degree is determined by Spray Factor

In cloud computing environment, the mostly used model is Map/Reduce, this model operates well in large-scale parallel task. Especially in cloud computing environment, it needs to processes each cloud user's resource number, time, network channel fee, etc. in time. The currently related task scheduling algorithm focuses on the needs of overall task, considers less about the cloud user's complementing time, which led to unreasonable in time and resources distribution for the users when multiple tasks operates. Supposes cloud client's tasks of cloud computing as Table 1:

a) Divides large-scaled task into relatively small tasks, divides in average, the sub-tasks' operating time are similar.

b) The number of resource distribution offers enough for sub-tasks.

c) Reasonable defines sub-task occupies resources time.

Table 1. Sub-tasks and Resources Table

The sub tasks	resources	Running time	Running costs	Total resources
n_1	m_1	$t(n_1,m_1)$	$\cos t(n_1,m_1)$	$Sm=\sum_{i=1}^{n} m_i$
n_2	m_2	$t(n_2,m_2)$	$\cos t(n_2,m_2)$	
… … ….	… … ….	……….	……….	
n_n	m_n	$t(n_i,m_j)$	$\cos t(n_i,m_j)$	

N_i refers to the number of sub-tasks, m_i refers to the number of resources, $t(n_i,m_i)$ refers to the time in task i, resources j, $\cos(n_i,m_j)$ refers to the costs in task i, resources j. In these above models, supposes the resources in cloud computing reasonable can be distributes into the computing resources of sub-tasks and ensures the shortest time and the lower costs for complementing the sub-tasks.

4. Proposed Algorithm

4.1. Mathematical Model

The resources distribution under cloud computing related some specification for cloud users, they have this restrictions as following in general:

a) The required order is given for each cloud user.

b) Can only receives one requirement of one user in one period of time, each user can only be occupied by one server in cloud server, one started it cannot be disrupted.

c) Each task can only operate once in one cloud server in the whole machining process.

d) Without considering the superiority of each task.

The mathematic model of this problem can be expressed as following:

$$FinishTime_{ij} - MachiningTime_{ij} + M\left(1-\alpha_{ijk}\right) \geq FinishTime_{ij}\, i = 1,2,\ldots n,\ j = 1,2,\ldots n \tag{6}$$

$$FinishTime_{jk} - FinishTime_{ij} + M\left(1-\beta_{ij}\right) \geq MachiningTime_{oj}\, i,k = 1,2,\ldots n,\ j = 1,\ldots,n \tag{7}$$

$$FinishTime_{ij} \geq 0, i = 1,2,\ldots n, k = 1,2,\ldots m \tag{8}$$

$$Min\,\max\left\{\max FinishTime_{ik}\right\} \tag{9}$$

Formula (6) refers to the operating order of each sub-task determined by each task. Formula (7) refers to the order of each sub-task, formula (8) refers to each sub-task's time variable restriction. Formula (9) refers to objective function. $FinishTime_{ij}$ refers to the complementing time of task i in server k, $MachiningTime_{oj}$ refers to the processing time of task i in server k, M is a coefficient defined values, α_{ijk} and β_{ijk} are expressed as following: α_{ijk} is 1 refers to server j operating task i in server k, α_{ijk} is 0 refers to other conditions. β_{ijk} is 1 refers to task i processing task i in server k, β_{ijk} is 0 refers to other conditions.

4.2. The Applications of Improved Firefly Algorithm in Cloud Computing Tasks

In basic firefly algorithm, fluoresce in value judges position the current position of firefly is the best position, when it's in the best position, it can attracts more fireflies to move towards so as to find out the objective value of function, but in the obtaining of fluoresce in, current algorithm is easily to fell into local optimum. Aimed at this, improves formula (5), the fitness value of firefly is between $[l_{\min}, l_{\max}]$ then gets formula (10)

$$l_j(u) = l_i(u-1) + r\frac{l_{\min}}{l_{\max}}\gamma p\left(x_i(u)\right) \tag{10}$$

The improved fluoresce in value can better avoid felling into local convergence and find best position, this improvement fits the reasonable use of multiple tasks in cloud computing, from formula (6) and formula (7), it ensures the complementing time among tasks in cloud server can reasonable close to processing time and confirms the objective function in formula (9) reaches minimum. Applies improved firefly algorithm into cloud computing, all the processed task uses string encoding, randomly chooses the cloud user's task which needs to be processed, all the tasks are numbered. Each firefly refers is a solution, the firefly's position length refers to all the working procedure, the number of firefly refers to the searching number and space of the solution. The objective function in fireflies turns to the minimum value in cloud computing complementing time.

Copyright ⓒ 2015 SERSC

4.3. Steps of Algorithm

a) Initializes each parameter in the algorithm and defines the initial firefly's position, defines the minimum l_{min} and maximum l_{max} of fitness degree.

b) Calculates new fluoresce in value by formula (10) and the objective function of firefly is fluoresce in value.

c) Judges the new fluoresce in value by formula (10), controls by formula $r\frac{l_{min}}{l_{max}}$.

d) Changes the position of firefly, chooses fireflies which meets the standards by formula (2).

e) Randomly chooses directional firefly i, updating by formula (4).

f) After one iteration, judges whether the iteration meets the finishing condition when enters into next iteration. If not meets, turns to step 2. If satisfied, directly output the optimal solution.

5. Experimental Results

In order to prove the performance of this algorithm, tests in two aspects, one is the performance of algorithm, the other is the task scheduling in cloud computing. It makes comparison test using three benchmark function in literature [8] and to test the algorithm's efficiency and performance. Using of MATLAB in Windows.

(a) Sphere function

$$f(x)=\sum_{i=1}^{m} x_i^2 \quad -100 \le x_i \le 100$$

This is a continuous, un-modal convex function, the minimum point of the global function is zero, $i, e, x_i = 0 \, (i = 1, 2, \dots n)$, there are on interaction among variables.

(b) Goldstein-Price function

$$f(x)=\left[1+(x_1+x_2+1)^2+\left(18-13x_1+2x_1^2+5x_1x_2+2x_2^2\right)\right]$$

$$\left[30+(2x_1-3x_2)^2\left(19-33x_1+12x_1^2+27x_2^2\right)\right]-3\le 2, i=\{1,2\}$$

This is a multiple hump function, the minimum point of the whole function is 3, $i, e, x_i = 3\,(i = 1, 2, \dots n)$, there are on interaction among variables.

(c) Ackely function

$$f(x)=-21\exp\left(-0.2\sqrt{(\frac{1}{n}\sum_{i=1}^{m}}\cos 2\pi x_i)+21+r-34\le x_i\right)$$

This is a multiple hump function with many local minimum points, the minimum point of the whole function is 0, $i, e, x_i = 0\,(i = 1, 2, \dots n)$, there are on interaction among variables.

In the setting process of algorithm parameter, the scale of initial firefly is 500, iteration time is 200, parameter of fluoresce in is $p = 0.6$, parameter of function is $\gamma = 0.4$, initial fluoresce in is $l_0 = 10$.

Respectively tests 10 times of three functions and gets the best solution, worst solution and the average value. According to make compare with basic firefly algorithm, the results is as Table 1, Table 2 is the convergence curve comparison between the algorithm in this text and intelligence firefly algorithm in three functions.

Table 1. Comparison of Testing Function

function	algorithm	the best solution	the worst solution	average
Sphere function	intelligence firefly algorithm	0.0212501	0.0298742	0.0255600
	algorithm in this text	0.0019610	0.0105412	0.0062476
Goldstein-Price function	intelligence firefly algorithm	3.0005126	3.0014521	3.0009822
	algorithm in this text	3.0000003	3.0000301	3.0000112
ACKELY function	intelligence firefly algorithm	3.4589131	3.6521469	3.5565298
	algorithm in this text	3.2456321	3.3562153	3.3109179

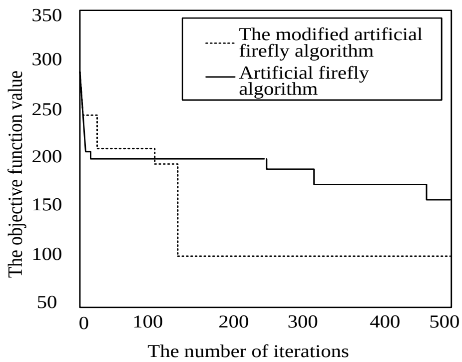


Figure 4. Comparison of Sphere Function's Convergence Curves

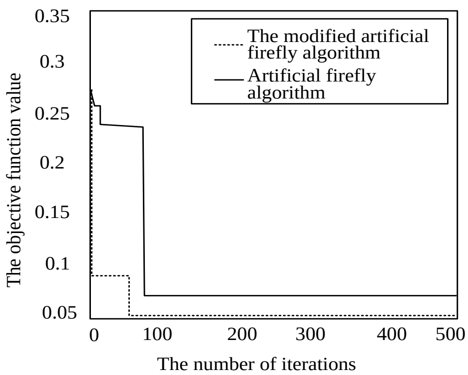


Figure 5. Comparison of Goldstein-Price Function's Convergence Curves

 Copyright ⓒ 2015 SERSC

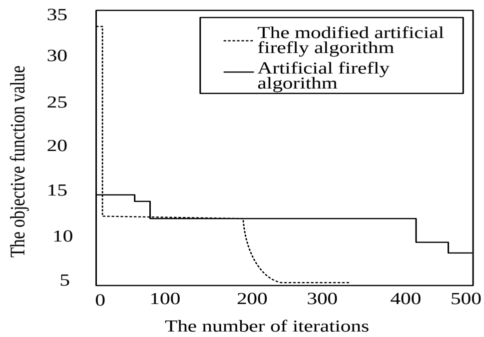


Figure 6. Comparison of ACKELY Function's Convergence Curves

From Table 1, the improved firefly algorithm is obviously better than intelligence firefly algorithm no matter in the best solution, the worst solution or the average. When basis intelligence firefly algorithm has certain iterations, it's easily fell into local optimum. This text efficiently limits the probability of felling into local optimum according to improve the fluoresce in of firefly, therefore reaches the optimum time of objective function.

6. Conclusion

How to make full use of the resources in cloud computing environment is a current-focusing problem. The method in this text improved the intelligence firefly algorithm in nature, combined the number of sub-task, resources with algorithm. In intelligence firefly algorithm, improved the method of firefly's fluoresce in position so as to make the firefly find the better object faster. On this improvement, the method reasonable solved the problem of balancing the network load and extending network, enhanced the global convergence of algorithm, it's valuable for increasing network operation. But there are still many practical problems in cloud computing to solve, resources distribution in cloud computing needs to be further researched.

Acknowledgements

The research is supported by the Science and Technology Department of Hebei Province Foundation of China under Grant No. 13210130.

References

[1] Y. Geng, J. He, H. Deng and K. Pahlavan, "Modeling the Effect of Human Body on TOA Ranging for Indoor Human Tracking with Wrist Mounted Sensor", 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), Atlantic City, NJ, **(2013)** June.

[2] J. Fan, "The modified Levenberg-Marquardt method for nonlinear equations with cubic convergence", Math. Comput., http://dx.doi.org/10.1090/S0025-5718-2011-02496-8, vol. 81, **(2012)** June, pp. 447-466.

[3] S. Li, Y. Geng, J. He and K. Pahlavan, "Analysis of Three-dimensional Maximum Likelihood Algorithm for Capsule Endoscopy Localization", 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), Chongqing, China, **(2012)** October, pp. 721-725.

[4] E. Provenzi, C. Gatta and M. Fierro, "A spatially variant white-patch and gray-world method for color image enhancement driven by local contrast", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 10, **(2008)**, pp. 1757-1770.

[5] S. Zhou, G. Aggarwal and R. Chellapa, "Appearance characterization of linear lamebrain object", Generalized photometric stereo and illumination- Invariant face recognition, IEEE Trans on PAMI, vol. 29, no. 2, **(2007)**, pp. 230-245.

[6] A. Ali-Eldin, J. Tordsson and E. Elmroth, "An Adaptive hybrid elasicity controller for cloud infrastructures", IEEE Network Operations and Management Symposium, **(2012)**, pp. 204-212.

[7] D. Breitgand and A. Epstein, "SLA-aware Placement of Multi-Virtual Machine Elastic Services in Compute Clouds", IFIP/IEEE International Symposium on Integrated Network Management, Dublin, Ireland, **(2011)**.

[8] W.-L. Zhou, J. Xie and L.-B. Deng, "Sequencing Optimization Method for Drawing Train Operation Diagram of High-Speed Railway", Journal of Multimedia, vol. 8, no. 1, **(2013)**, pp. 16-23.
[9] H. Chikara, K. Taketoshi, T. Norio, K. Shigeki and A. T. Motoshi, "Train stop deployment planning algorithm using a petri-net-based odeling approach quarterly", Report of RTRI, http://dx.doi.org/10.2219/rtriqr.50.8, vol. 50, no. 1, **(2009)**, pp. 8-13.
[10] K. P. Young, P. Seokl and K. K. Joong, "Retinex method based on adaptive smoothing for illumination base on adaptive smoothing for illumination invariant face recognition", Signal Processing, vol. 88, **(2008)**, pp. 1929-1945.
[11] S. Sudevalayam and P. Kulkami, "Affinity-Aware Modeling of CPU Usage for Provisioning Virtualized Applications", 2011 IEEE 4th Intemational Conference on Cloud Computing, **(2011)**, pp. 139-146.
[12] R. Berangi, S. Saleem and M. Faulkner, "TDD cognitive radio femtocell network (CRFN) operation in FDD downlink spectrum", IEEE, 22nd International Symposium on Personal, Indoor and Mobile Radio Communications, **(2011)**, pp. 482-486.
[13] A. D'Ariano and M. Pranzo, "An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances", Networks and Spatial Economics, http://dx.doi.org/10.1007/s11067-008-9088-1, vol. 9, no. 1, **(2009)**, pp. 63-84.
[14] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition", Proceeding(s) of Conference on Computer Vision and Pattern Recognition (CVPR), **(2010)**, pp. 2691-2698.
[15] M. X. Nguyen, Q. M. Le, V. Pham, T. Tran and B. H. Le, "Multi-scale Sparse Representation for Robust Face Recognition", Proceeding(s) of Conference on Knowledge and Systems Engineering (KSE), **(2011)**, pp. 195-199.
[16] Y. Zhai, J. Gan and J. Li, "Study of occluded robust face recognition approach based on homotopy algorithm and color information fusion", Signal Processing, vol. 21, no. 11, **(2011)**, pp. 1762-1768.
[17] S. Li, Y. Geng, J. He and K. Pahlavan, "Analysis of Three-dimensional Maximum Likelihood Algorithm for Capsule Endoscopy Localization", 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), Chongqing, China, **(2012)** October, pp. 721-725.
[18] Y. Geng, J. He, H. Deng and K. Pahlavan, "Modeling the Effect of Human Body on TOA Ranging for Indoor Human Tracking with Wrist Mounted Sensor", 16th International Symposium on Wireless Personal Multimedia Communications (WPMC), Atlantic City, NJ, **(2013)** June.
[19] Y. Geng, J. He and K. Pahlavan, "Modeling the Effect of Human Body on TOA Based Indoor Human Tracking", International Journal of Wireless Information Networks, vol. 20, no. 4, pp. 306-317.
[20] L. Cordero-Grande, G. Vegas-Sánchez-Ferrero, P. Casaseca-de-la-Higuera, J. A. San-Román-Calvar, A. Revilla-Orodea, M. Martín-Fernández and C. Alberola-López, "Unsupervised 4D myocardium segmentation with a Markov Random Field based deformable model", Medical Image Analysis, http://dx.doi.org/10.1016/j.media.2011.01.002, vol. 15, **(2011)** June, pp. 283-301.

Author

Hu Jichao, male, was born in 1979, in Hebei province. Now, he is lecture. His Main research area is Sports and Simulation.

 Copyright ⓒ 2015 SERSC