components. Although the ant colony algorithm has good optimization ability, the initial pheromone is scarce and the convergence speed is slow. On the basis of the study, this paper puts forward a kind of particle swarm optimization and ant colony optimization based on cloud computing task scheduling algorithm. This algorithm absorbs the rapid convergence of particle swarm optimization algorithm and optimization ability of ant colony algorithm. The time of the system process scheduling problem and the total task execution time are reduced, which correspondingly improves the efficiency of the cloud computing task scheduling.

References

1. Shangguang Wang, Zhipiao Liu, Qibo Sun et al. Towards an accurate evaluation of quality of cloud service in service-oriented cloud computing. Journal of Intelligent Manufacturing, 2014, 25(2), pp. 283-291.
2. Muhammad Irfan, Zhu Hong, Tauseef Qamer et al. Requirement Analysis and Design of Service Level Integration Layer for Cloud Computing Services, to Meet Service Level Agreements and Quality of Service. Journal of computational and theoretical nanoscience, 2014, 11(3), pp. 629-636.
3. Yan, G., Wen, D., Olariu, S. et al. Security Challenges in Vehicular Cloud Computing. IEEE transactions on intelligent transportation systems, 2013, 14(1), pp. 284-294.
4. Murugesan, San. Cloud Computing: The New Normal?. Computer, 2013, 46(1), pp. 77-79.
5. Abdul Nasir Khan, M.L. Mat Kiah, Sajjad A. Madani et al. Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. Journal of supercomputing, 2013, 66(3), pp. 1687-1706.
6. Zou, D., Zhang, W., Qiang, W. et al. Design and implementation of a trusted monitoring framework for cloud platforms. Future generations computer systems: FGCS, 2013, 29(8), pp. 2092-2102.
7. Tao, F., Cheng, Y., Xu, L.D. et al. CCIoT-CMfg: Cloud Computing and Internet of Things-Based Cloud Manufacturing Service System. IEEE transactions on industrial informatics, 2014, 10(2), pp. 1435-1442.
8. Gutierrez-Garcia, J.O., Sim, K.M. Agent-based cloud service composition. Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, 2013, 38(3), pp. 436-464.

Research on the Optimization Problem of the Manage Resource Distribution of the Particle Swarm Neural Network

Ji Luo, Ying Wang

School of Economic & Management, Sichuan University of Science & Engineering, Sichuan, 643000, China

Corresponding author is Ji Luo

Abstract

In this paper, the author mainly discusses the optimization problem of the manage resource distribution of the particle swarm neural network. The particle swarm optimization (PSO) is a population-based algorithm that was invented by Kennedy and Eberhart, which was inspired by the social behavior of animals such as fish schooling and bird flocking in this paper. Similar to other population-based algorithms, such as evolutionary algorithms, PSO can solve a variety of difficult optimization problems but has shown a faster convergence rate than other evolutionary algorithms on some problems. The experiment result shows that particle swarm neural network can improve the performance of manage resource distribution.

Keywords: IOPTIMIZATION PROBLEM, MANAGEMENT RESOURCE, PARTICLE SWARM NEURAL NETWORK

1. Introduction

Optimization problems are of vital importance in fields of computer science, artificial intelligence, operational research and other relative fields. Many problems encountered in engineering technology, scientific research and economic management can be treated as variations of challengeable nonlinear optimization problem, such as: configuration designing need to minimize total weight of materials used while satisfying the intensity request; resource allotting need to maximize total benefit utilizing limited resources; transportation scheming need to minimize total expense on circumstance of appointed material and load capability; manufacture scheme arranging need to maximize total benefit by controlling the costs of manpower, devices, and raw and processed materials according to the flow of techniques and demand of client. Optimization theory and its techniques will surely take more and more important part in the information era of 21 century. Numerical optimization methods were proposed these years due to the universality of optimization problems. Many hard optimization problem emerged which cannot be solved in acceptable time, with the increasing complexity of real tasks in the field of industry and scientific research. More effective and practicable algorithms are needed for the traditional programming methods cannot meet complex problems nowadays. As a newly developed swarm intelligence paradigm, particle swarm algorithm is a very promising optimization tool, with many advantages in high-dimensional problems or tasks that lack prior knowledge. Its basic idea is originated from Social Psychology and Artificial Life as a simulation of socio-cognitive processes. Because of its high convergence rate and excellent generalization, particle swarm algorithm has attracted much attention since it was first proposed in 1995. In this literature, most researchers have focused their efforts on how to promote the convergence rate and avoid the premature convergence problem. Introducing new mechanisms to ensure the diversity of swarm population or escape from local minima may be useful on relieving premature convergence of the algorithm. As to improving convergence rate, much work focus on tuning strategy parameters, or modifying the original framework with ideas inspired from other metaheuristics. As most researchers of this field are with pure scientific computing or engineering applications background, they care more about the results than probe into the real cause, not to mention consider social psychology origins of the algorithm. In Shi's [1] research, we attach importance to both theoretical analysis and experimental demonstration. From the aspect of information spread efficiency, a dynamic topology of particle swarm algorithm is thoroughly investigated and a novel particle swarm algorithm based on small world network model is proposed. Liu [2] summarize the developing background of swarm intelligence, and introduce three methodologies of swarm intelligence: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Artificial Fish-Swarm Algorithm (AFSA). The intrinsic and general character of swarm intelligence is analyzed through relevancy of reductionism, artificial life, self-organization system, and other topics. The author investigates the influence of parameters by large experiments and verifies the parameter selecting of canonical PSO [3]. Zhu's paper [4] analyze convergence of traditional PSO algorithm from the view of linear constant system, and elicit that the track of any particle will converge to the position held by the optimal particle. Then thet theoretically analyze algorithm convergence from the view of random system, and present a sufficient condition for the system to be mean square stable. This enhances the effectiveness of conclusion under linear constant condition. Luo [5] proposes two novel PSO algorithm based on edge-reassigning and edge-addition small-world network model, thus implement dynamic topology of PSO algorithm. New parameters introduced are thoroughly investigated by large amount of experiments on Benchmark problems. The proposed algorithms are rigorously tested and compared on large amount of selected benchmark problems taken from the literature. These benchmark problems are designed to challenge optimization techniques with difficulty of high-dimensionality, multimodality, and deceptive gradient information. In this dissertation, we focus attention on comparing the performance of different algorithms on hard multimodal functions, and give the statistical results on measurements of convergence rate, success rate, function evaluations, and the quality of obtained solutions. The results of experiments demonstrate that the proposed small-world PSO algorithms with dynamic topology can improve performance of classical PSO algorithms apparently.

2. The Optimization Problem of the Manage Resource Distribution

At present, lots of industries are facing fierce market competition and program paralleling. So as the enterprises are more incline to program management, collaborative management of groups of program, a new development trend of program management, is becoming a new direction or hot point of program management research and practice. During collaborative management of groups of program, enterprise will face the problem of reasonable deploying of fund, machine, human resource, material, technology and so forth. Under the environment of groups of program, enterprise internal resource deploying among different program is becoming a key problem that confines the success of enterprise and different resources' deploying. Under the background of collaborative management of groups of program and around the question of resource optimization, this paper combines the existing resource optimization methods of collaborative management of groups of program with the collaborative theory, and raises the method by using key chain to deploy the resource of groups of program. At the beginning, this paper describes the background of the research and the meanings of this paper, giving short conclusion of the collaborative theory, the collaborative management of groups of program, and the resource optimization theory and further illustrating the research contains, angle, and innovation points. Then some basic knowledge of resource optimization of collaborative management of groups of program is putting as follow: the main contains of collaborative theory, the collaborative management of groups of program, and the resource optimization theory. The third part of this article analyzes the problem of resource optimization of the collaborative management of groups of program. The forth part of this article put forward the resource optimization of key chain among the collaborative management of groups of program. Firstly, the short introduction of the TOC theory and the key chain method is given. Secondly, the key chain method's conduct foundation and effect are giving. Lastly, practical design of the key chain method of resource optimization is given. The fifth part of this article is given a practical case of the key chain method, using key chain method to deploy the resource, making comparison with the traditional method and giving advice of the resource optimization of the collaborative management of groups of program. In the end, some conclusion is given and the further problem is also putting forward [6-7].

4. The Particle Swarm Neural Network Algorithm

Many problems possess a set of parameters to be optimized, especially in fields of engineering technology, scientific research and economic management. Optimization theory and its techniques will surely take more and more important part in the information era of 21 century. As a newly developed swarm intelligence paradigm, particle swarm algorithm is a very promising optimization tool, with many advantages in high-dimensional problems or tasks that lack prior knowledge. Its basic idea is originated from Social Psychology and Artificial Life as a simulation of socio-cognitive processes. Because of its high convergence rate and excellent generalization, particle swarm algorithm has attracted much attention since it was first proposed in 1995.In this literature, most researchers have focused their efforts on how to promote the convergence rate and avoid the premature convergence problem. Introducing new mechanisms to ensure the diversity of swarm population or escape from local minima may be useful on relieving premature convergence of the algorithm. As to improving convergence rate, much work focus on tuning strategy parameters, or modifying the original framework with ideas inspired from other meta-heuristics. As most researchers of this field are with pure scientific computing or engineering applications background, they care more about the results than probe into the real cause [8].

The algorithm based on fractal theory can be expressed as following:

$$\hat{f}_H^{\alpha}(x)=\frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{\infty}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha} = \frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{\infty} f(t)g(x-t)(dt)^{\alpha} = f(x)*g(x), \tag{1}$$

for $0<a\le 1$ where

$$\frac{1}{\Gamma(1+\alpha)}\oint_R\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha} = \lim_{\varepsilon\to 0}[\frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{x-\varepsilon}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha}+ \frac{1}{\Gamma(1+\alpha)}\int_{x+\varepsilon}^{\infty}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha}] \tag{2}$$

And local fractional integral of $f(x)$ defined by Eq.3.

$$ {}_aI_b^{(\alpha)}f(t)=\frac{1}{\Gamma(1+\alpha)}\int_a^b f(t)(dt)^{\alpha} \\ =\frac{1}{\Gamma(1+\alpha)}\lim_{\Delta t\to 0}\sum_{j=0}^{j=N-1} f(t_j)(\Delta t_j)^{\alpha} \quad (3)$$

If $f(x)$ is defined on the real line $-\infty<x<\infty$, its local fractional Hilbert transform, denoted by $f_x^{H,\alpha}(x)$ is defined by

$$H_\alpha\{f(t)\}=\hat{f}_H^{\alpha}(x)=\frac{1}{\Gamma(1+\alpha)}\oint_R\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha} \quad (4)$$

Where x is real and the integral is treated as a Canchy principal value, that is,

$$\frac{1}{\Gamma(1+\alpha)}\oint_R\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha} \\ =\lim_{\varepsilon\to 0}[\frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{x-\varepsilon}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha}+ \\ \frac{1}{\Gamma(1+\alpha)}\int_{x+\varepsilon}^{\infty}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha}] \quad (5)$$

To obtain the inverse local fractional Hilbert transform, write again Eq. (4) as

$$\hat{f}_H^{\alpha}(x)=\frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{\infty}\frac{f(t)}{(t-x)^{\alpha}}(dt)^{\alpha} \\ =\frac{1}{\Gamma(1+\alpha)}\int_{-\infty}^{\infty}f(t)g(x-t)(dt)^{\alpha}=f(x)*g(x), \quad (6)$$

The equation of motion is as follows:

$$\partial_j(C_{ijkl}\partial_k u_l+e_{kij}\partial_k\varphi)-\rho\ddot{u}_i=0 \quad (7)$$

Under the linear theory, that is:

$$\partial_j(e_{ijkl}\partial_k u_l-\eta_{kij}\partial_k\varphi)=0 \quad (8)$$

The linear equation can be expressed into the following simplified forms:

$$L(\nabla,\omega)f(x,\omega)=0,\ L(\nabla,\omega)=T(\nabla)+\omega^2\rho\mathrm{J} \quad (9)$$

In which,

$$T(\nabla)=\begin{Vmatrix}T_{ik}(\nabla) & t_i(\nabla)\\ t_k^T(\nabla) & -\tau(\nabla)\end{Vmatrix},\ \mathrm{J}=\begin{Vmatrix}\delta_{ik} & 0\\ 0 & 0\end{Vmatrix},$$

$$f(x,\omega)=\begin{Vmatrix}u_k(x,\omega)\\ \varphi(x,\omega)\end{Vmatrix} \quad (10)$$

Consider delay, the L can be expressed as:

$$\mathcal{L}^0=\begin{Vmatrix}C_{ijkl}^0 & e_{kij}^0\\ e_{ikl}^{0T} & -\eta_{ik}^0\end{Vmatrix} \quad (11)$$

These functions can be expressed in the following form:

$$C(\mathrm{x})=C^0+C^1(\mathrm{x}),\ e(\mathrm{x})=e^0+e^1(\mathrm{x}), \\ \eta(\mathrm{x})=\eta^0+\eta^1(\mathrm{x}),\ \rho(\mathrm{x})=\rho_0+\rho_1(\mathrm{x}) \quad (12)$$

The value with superscript of 1 represents the difference below:

$$C^1=C-C^0,e^1=e-e^0, \\ \eta^1=\eta-\eta^0,\rho_1=\rho-\rho_0 \quad (13)$$

The whole function can be simplified into the following integral equation set:

$$f(x,\omega)=f^0(x,\omega)+\int_V\mathcal{S}(x-x')(\mathrm{L}^1F(\mathrm{y}') \\ +\rho_1\omega^2\mathbf{g}(R)\mathrm{T}_1f(\mathrm{y}')]S(\mathrm{y}')\mathrm{dy}' \quad (14)$$

In addition, we can introduce the abbreviated formula:

$$\mathcal{g}(x,\omega)=\begin{Vmatrix}G_{ik}(x,\omega) & \gamma_i(x,\omega)\\ \gamma_k(x,\omega) & g(x,\omega)\end{Vmatrix},$$

$$\mathcal{s}(x,\omega)=\begin{Vmatrix}G_{ik,l}(x,\omega) & \gamma_{i,k}(x,\omega)\\ \gamma_{k,l}(x,\omega) & g_{,k}(x,\omega)\end{Vmatrix},$$

$$\mathcal{L}^1(x,\omega)=\begin{Vmatrix}C_{ijkl}^1 & e_{kij}^1\\ e_{kij}^{1T} & -\eta_{ik}^1\end{Vmatrix},$$

$$\mathcal{F}(x,\omega)=\begin{Vmatrix}u_{(i,j)}(x,\omega)\\ \varphi_{,i}(x,\omega)\end{Vmatrix} \quad (15)$$

In these expression, $G_{ik}(x,\omega)$, $\gamma_i(x,\omega)$, $g(x,\omega)$ can be represented as:

$$\mathcal{g}(x,\omega)=\frac{1}{(2\pi)^3}\int\mathcal{g}(k,\omega)\exp(-i\mathrm{k}\cdot\mathrm{x})d\mathrm{k},$$

$$\mathcal{g}(k,\omega)=\begin{Vmatrix}G_{ik}(k,\omega) & \gamma_i(k,\omega)\\ \gamma_k^T(k,\omega) & g(k,\omega)\end{Vmatrix}$$

$$G_{ik}=(\Lambda_{ik}+\frac{1}{\lambda}h_ih_k^T)^{-1},\ g=-(\lambda+h_i^T\Lambda_{ij}^{-1}h_j)^{-1},$$

$$\gamma_i=\frac{1}{\lambda}h_k^TG_{ki},$$

$$\Lambda_{ik}(k,\omega)=k_jC_{ijkl}^0k_k-\rho_0\omega^2\delta_{il},\ h_i(k)=e_{kil}^0k_kk_l,$$

$$h_l^T=e_{ikl}^{0T}k_ik_k,\ \lambda(k)=\eta_{ik}^0k_ik_k$$

$F(x,\omega)$ has nothing to do with coordinate x_3. In view of the following relationship

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ik_3x_3'}dx_3' = \delta(k_3) \tag{16}$$

The particle swarm optimization (PSO) is a population-based algorithm that was invented by Kennedy and Eberhart, which was inspired by the social behaviour of animals such as fish schooling and bird flocking. Similar to other population-based algorithms, such as evolutionary algorithms, PSO can solve a variety of difficult optimization problems but has shown a faster convergence rate than other evolutionary algorithms on some problems. Another advantage of PSO is that it has very few parameters to adjust, which makes it particularly easy to implement. It was pointed out that although PSO may outperform other evolutionary algorithms in the early iterations, its performance may not be competitive as the number generations are increased. Recently, several investigations have undertaken to improve the performance of standard PSO (SPSO). Lobjerg et al. presented a hybrid PSO model with breeding and subpopulations Kennedy and Mendes investigated the impacts of population Structures to the search performance of SPSO. Other investigations on improving the performance of SPSO were undertaken using cluster analysis and fuzzy adaptive inertia weightier.

Therefore, the procedure of the PSO algorithm can be described as follows.

Step 1. Randomly initialize positions and velocities of all particles.

Step 2. For each particle, set P_i to the current position X_i, and set P_g to the current best position of the swarm.

Step 3. For each particle in the swarm

Step 3.1. Update velocities Vi and positions Xi using Eq. (1) ~ (3);

Step 3.2. Calculate the fitness value of current particle: f (Xi);

Step 3.3. Compare the fitness value of P with f (Xi).If f (Xi) is better than the fitness value of P_i, then set P_i to the current position Xi.

Step 3.4. If f (Xi) is better than the fitness value of P_g, then g is set to the position of the current Xi.

Step 4. Calculate the variance of the population's fitness σ^2 using Eq. (4) and (5).

Step 5. Calculate the mutation probability prob according to Eq. (6).

Step 6. Randomly generate a number r $\in$ [0, 1], if r $\le$ prob, update P_g.

4. The Experiment Analysis

As to improving convergence rate, much work focus on tuning strategy parameters, or modifying the original framework with ideas inspired from other meta-heuristics. As most researchers of this field are with pure scientific computing or engineering applications background, they care more about the results than probe into the real cause, not to mention consider social psychology origins of the algorithm.

These benchmark problems are designed to challenge optimization techniques with difficulty of high-dimensionality, multimodality, and deceptive gradient information. In this dissertation, we focus attention on comparing the performance of different algorithms on hard multimodal functions, and give the statistical results on measurements of convergence rate, success rate, function evaluations, and the quality of obtained solutions. The results of experiments demonstrate that the proposed small-world PSO algorithms with dynamic topology can improve performance of classical PSO algorithms apparently.Table 1 shows the reference values of different samples in the 2-step PSO algorithm. Table 2 shows the reference values of different samples in the 2-step PSO algorithm and Table 3 shows the reference values of different samples in the 2-step PSO algorithm.

Table 1. Reference values of different samples in the 2-step PSO algorithm.

	High point (*st*)		Low point (*st*)		**Efficiency** (*st*)	
	mean	sd	mean	sd	mean	sd
Pw1	13.4	4.7	7.6	4.0	5.8	3.5
Pw 2	11.8	4.3	4.4	3.3	7.4	3.6
F(1, 691) = 22.4, p < 0.001			F(1, 691) = 124, p < 0.001		F(1, 691) = 31.6, p < 0.001	
Pw1 > Pw2			Pw1 > Pw2		Pw2 > Pw1	

Table 2. Reference values of different samples in the 3-step PSO algorithm.

	High point (*st*)		Low point (*st*)		**Efficiency** (*st*)	
	mean	sd	mean	sd	mean	sd
Pw1	13.9	4.2	8.2	3.8	5.7	3.2
Pw 2	12.7	4.8	6.7	3.5	6.0	3.9
Pw 3	11.1	4.2	4.1	3.2	7.0	3.4
$F(2, 835) = 29.8$, $p < 0.001$			$F(2, 835) = 102$, $p < 0.001$		$F(2, 835) = 10.5$, $p < 0.001$	
Pw1 > Pw2 > Pw3			Pw1 > Pw2 > Pw3		Pw3 > Pw1, Pw2	

Table 3. Reference values of different samples in the 4-step PSO algorithm.

	High point (*st*)		Low point (*st*)		**Efficiency** (*st*)	
	mean	sd	mean	sd	mean	sd
Pw1	15.0	4.5	8.8	4.4	6.2	2.7
Pw 2	14.3	4.5	8.1	3.9	6.2	2.9
Pw 3	12.9	5.1	7.4	3.7	5.5	4.3
Pw 4	11.8	4.5	4.6	3.7	7.1	3.7
$F(3, 513) = 12.1$, $p < 0.001$			$F(3, 513) = 28.1$, $p < 0.001$		$F(3, 513) = 5.1$, $p < 0.01$	
Pw1, Pw2 > Pw3, Pw4			Pw1, Pw2 > Pw2, Pw3 > Pw4		Pw1, Pw2, pw4 > Pw1, Pw2, Pw3	

Conclusions

In this paper, the author mainly discusses the optimization problem of the manage resource distribution of the particle swarm neural network. So as the enterprises are more incline to program management, collaborative management of groups of program, a new development trend of program management, is becoming a new direction or hot point of program management research and practice. During collaborative management of groups of program, enterprise will face the problem of reasonable deploying of fund, machine, human resource, material, technology and so forth. Under the environment of groups of program, enterprise internal resource deploying among different program is becoming a key problem that confines the success of enterprise and different resources' deploying. PSO can solve a variety of difficult optimization problems but has shown a faster convergence rate than other evolutionary algorithms on some problems. The experiment result shows that particle swarm neural network can improve the performance of manage resource distribution.

Acknowledgements

This work is supported by the Talents Project of Sichuan University of Science & Engineering (No. 2012RC27), the Science Research Project of the Education Department of Sichuan, China (No.13SB0177).

References

1. Shi Y, Eberhart R.C., Fuzzy Adaptive Particle Swarm Optimization , In: Proceedings of the IEEE International Conference on Evolutionary Computation., 2001, pp. 101-106
2. Peng Liu. Cloud computing. BeiJing: Electronic Industry Press, 2007, pp. 2-13
3. Rodrigo N. Calherios, Rajiv Ranjan, Anton Beloglazov. CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software Practice and Experience,2011,41(1), pp.23-50.
4. Chen Liang, James Z. Wang, Rajkumar Buyya. Bandwidth-aware divisible task scheduling for cloud computing, Software-practice and Experience, 2014, 44, pp. 163-174.
5. Zong-bin ZHU, Zhong-jun DU. Improved GA-based task scheduling algorithm in cloud computing. Computer Engineering and Applications, 2013, 49(5), pp. 77-80.
6. Jian-ping LUO, Xia LI, Min-rang CHEN. Guaranteed QoS resource scheduling scheme based on improved shuffled frog leaping algorithm in cloud environment. Computer Engineering and Applications, 2012, 48(29), pp. 67-72.
7. Ming-hai XU, Yuan ZLACO-based network selection algorithm. Computer Engineering and Applications, 2012, 48(5), pp. 84- 88.
8. Mingjiang Song, Youxing CUI, Fei XU. Investigation on the Current Status of Middle and Primary School Teachers' Career Development Impetus in the Context of Urban and Rural Planning: Based on Districts B and Y of Chongqing City, and County R. Canadian Social Science, 2015, pp. 113-129.