Acta Technica 62 (2017), No. 7A, 1733–1746

Making technology for virtual reality-based course design of laboratory micro learning resource

Xinchao Song[1], Yongsheng Zhao[1], Lizhi Wang[1]

Abstract. An optimization algorithm for virtual reality-based (VR) course design problem of laboratory micro learning resource of MapReduce parallel quantum ant colony algorithm is presented for NP-hard imagination in course design problem of laboratory micro learning resource. Firstly, the course design problem of laboratory micro learning resource is modeled for study. An optimized objective function is recommended for the course design problem of laboratory micro learning resource. The optimized mathematical model is established. Moreover, the adaptive form of the multi-objective weight is given. Secondly, the ant colony algorithm is introduced. A MapReduce parallel quantum improvement algorithm is offered for the problem that the traditional ant colony algorithm does no good to the improvement of algorithm search efficiency. The quantum ant colony algorithm is parallelized by using MapReduce key/value programming model. The result is that the MapReduce-based quantum ant colony algorithm (MQACA) is proposed. Operation of MQACA on Hadoop cloud computing platform provides good speedup ratio and parallel efficiency. Lastly, an experiment is conducted to compare and validate the efficiency of the proposed algorithm. The system design of the algorithm is implemented for the optimization of course design problem of laboratory micro leaning resource by the algorithm.

Key words. Course design of laboratory micro learning resource, Algorithm evolution, Micro learning resource, Virtual reality, Ant colony algorithm

.

1. Introduction

The course design of laboratory micro learning resource in colleges and universities involves a variety of uncertain factors such as a large number of courses, students, classrooms, classrooms, etc., and this has made it extremely heavy work [1∼2]. In

[1]College of information and electrical engineering, Ludong University, Yantai Shandong, 264025, China

particular, given the enrollment expansion for course design of laboratory micro learning resource in recent years and the importance colleges and universities attach to teaching quality, how to implement course assignment in a more reasonable and efficient way remains a focus of them. Particularly, a proposal for the concept that course resources should be shared among colleges and universities makes the course design of laboratory micro learning resource increasingly important [3].

The course design problem of laboratory micro learning resource in colleges and universities is essentially a multi-objective and multi-constraint NP-hard combinatorial optimization problem [4]. There are There have already been many mature algorithms for such problem, such as branch and bound [5], grouping optimization strategy [6], association rule algorithm [7], etc. These algorithms may be effective in solving NP-hard combinatorial optimization problem; however, there are certain problems as follows: (1) the algorithm only tries to solve one problem during the course of solving. It results in the impossibility to establish a common way to the course design of laboratory micro learning resource; (2) with fewer discrimination criteria for the quality of course arrangement, algorithm solving attaches excess importance to the optimization over a certain direction and fails to achieve global optimization; and (3) for the association rule algorithm, it is not universal due to the fact that there is little access to the association rule in the course of solving. The solving result is not desirable. The evolution algorithm [8] is an intelligent bionic computation method emerging in recent years. The objective function and constraint condition are things only allowed in its optimization course. Neither gradient nor other supplementary information is required. Additionally, with the universal application to the fields, the algorithm is a relatively efficient solution to NP-hard combinatorial optimization problem. The ant colony algorithm was first proposed by an Italian scholar Dorigo M. in 1991. The algorithm featuring with better optimizing capacity and higher robustness has been successfully applied to TSP solving, workpiece sequencing, graph coloring, vehicle scheduling and other multi-objective combinatorial optimization problems [2-4]. The quantum evolution algorithm (QEA)[5] was brought to us by KuK-Hyuan Han et al. in 2002. This is a quantum theory-based evolution algorithm. It is filled with the ideas including superposition state, coherence and entanglement in quantum computing. This enables the quantum algorithm to break through the limit to traditional ones and provides it with better performance. Such algorithm turns to the focus of research with its unique computing property and has aroused the interests of domestic and domestic scholars. A number of research findings have been obtained. QACA algorithm combines quantum computing and ant colony algorithm. It introduces the stat vector and quantum rotation gate in quantum computing to the ant colony algorithm. The pheromone is updated by such quantum rotation gate and the optimal solution to accelerate the convergence speed and avoid premature convergence of the algorithm. QACA algorithm has successfully solved many NP-hard problems including 0/1 knapsack problem [7], TravelingSalesman Problem (TSP). However, QACA algorithm solves these problems in the serial circumstance, but have not yet capitalized on the study of parallelization of quantum ant colony algorithm by cloud computing.

For this purpose, this paper utilizes a robust MapReduce parallel quantum improvement algorithm to implement the adaptive updating of the region of search by ant colony algorithm. It is to ensure that the computing complexity of the algorithm is effectively reduced and the convergence speed is increased while maintaining the best point in the region of search, in an effort to improve the outcome of the course design of laboratory micro learning resource.

2. Course design problem model of laboratory micro learning resource

2.1. Description of course design problem model of laboratory micro learning resource

Given that the school where the course design of laboratory micro learning resource is implemented has G substitute teacher(s), C class(es) pending the course design of laboratory micro learning resource, L course(s), T period(s) of time for course design of laboratory micro learning resource and R classroom(s). The mathematical model will be described as follows:

The set form of the class(es) pending the course design of laboratory micro learning resource is $C = \{c_1, \cdots, c_C\}$. The set of the number of student in each class is $K = \{k_1, \cdots, k_C\}$. The set comprising the substitute teachers is $G = \{g_1, \cdots g_G\}$. The number of courses of the teachers is $Y = \{y_1, \cdots y_G\}$. The course set is described with $L = \{l_1, \cdots l_L\}$. The number of classes to which each course is available is $Z = \{z_1, \cdots z_L\}$. The set of classroom(s) where the courses are given is described with $R = \{r_1, \cdots, r_R\}$. The number of students to be supported by each teacher is $X = \{x_1, \cdots, x_R\}$. The set of period(s) of time is $T = \{t_1, \cdots, t_T\}$.

By computing the Cartesian product of the time and the classroom, the course design problem of laboratory micro learning resource is converted to the course and an appropriate model time-classroom, with the form as follows:

$$M = T \cdot R = \{(t_1, r_1), \cdots, (t_T, r_R)\} . \tag{1}$$

2.2. Optimization problem

The course design system of laboratory micro learning course is essentially a multi-objective optimization problem with the following optimization objectives:

Objective 1: Major courses are offered at a period of time during which good teaching results will be obtained. If $a_i\,(i = 1, 2, 3, 4, 5)$ represents 5 teaching periods provided every day, the instruction results of periods 1, 3 and 5 will be the best according to practical teaching experience. It is expressed in $a_i = 1\,(i = 1, 3, 5)$. Periods 2 and 4 have poorer results, and are expressed in $a_i = 0\,(i = 2, 4)$. Parameter $\beta_j = 1\,(j = 1, 2, 3, 4)$ is used to represent the importance of course, such as optional, foundation, specialized and degree courses with different weights assigned, the optimization objective will be:

$$\max(f_1) = \sum (a_i \beta_j) . \tag{2}$$

Objective 2: The time and location where a course is given by the teacher is allowed for. It is assumed that the coefficient of professional title is χ_i $(i = 1, 2, 3, 4)$ for assistant, lecture, associate professor and professor. The teacher gives a course at the given time. His/Her willingness may be expressed in $\delta_i = 0, 1, 2$ to represent Not Agreed, Agreed and Willingness, respectively. The optimization objective form will be:

$$\max(f_2) = \sum (\chi_i \delta_j) . \tag{3}$$

Objective 3: for courses with more periods in a week (such as $n \geq 4$), they should be provided once every one day as possibly as it could in order to ensure teaching effects. The definition of $\beta_j = 1$ $(j = 1, 2, 3, 4)$ is the same with the one in Objective 1. The definitionε_i $(i = 1, 2, 3, 4)$ represents the teaching results of the case under which the courses are provided every i day(s). The optimization objective form is:

$$\max(f_3) = \sum (\beta_i \varepsilon_j) . \tag{4}$$

Objective 4: for the resource utilization objective, a higher proportion of the number of students k_c taking the course in the classroom to the maximum classroom capacity means higher utilization. The optimization objective will be:

$$\max(f_4) = \sum (k_c / r_r) . \tag{5}$$

For these objective functions, application of multi-objective functions for optimization has problems including too many optimization objectives, uncertainty of the final optimization scheme and non-optimal scheme. The application of single-objective function has the problem of varying magnitudes of objective values. Traditional schemes implement combination with weights. However, weight selection requires prior knowledge. The magnitude of objectives is subject to real-time change in the practical optimization course. It is clearly improper to make the weight fixed. On this account, here presents the adaptive weight form as follows:

$$f = \max \left(\sum_{m=1}^{4} \frac{f_m - f_m^{\min}}{f_m^{\max} - f_m^{\min}} \right) . \tag{6}$$

Where, f_m is current individual fitness of m generation population. $f_m^{\max}$ is the maximum fitness of the population. $f_m^{\min}$ is the minimum adaptive value of the population. This method implements the weight self-adaptive method and improves search accuracy.

3. MapReduce-based parallel quantum ant colony algorithm

3.1. MapReduce computing model

Enlightened by map and reduce functions in the functional language, Google presents MapReduce (map-reduce algorithm) abstract model that enables users to easily develop large-scale distributed applications. In this model, each map function is independent and utilizes the fault-tolerant mechanism that is re-executed upon occurrence of fault. It may implement large-scale parallel computing with ease. Hadoop [10] project of Apache open source community implements the model with Java language and also provides an open course implementation platform for cloud computing.

The core of MapReduce computing model is Map and Reduce functions both of which are written by users. The former computes the key value pair (k/v) input by the user to generate a series of intermediate key value pair $(k1/v1)$. MapReduce framework aggregates the key value pair (k/v) whose keyword is $k1$ to generate the value set list $(v1)$ of kl key to be transmitted to the user-defined Reduce function. The latter function further processes and combines the value set of such intermediate key. It finally forms a relatively small key value pair set list $(k2, v2)$.

The complete course may be expressed as follows:

$$\mathrm{Map}\,(k, v) \rightarrow \mathrm{list}(k1, v1)\,. \tag{7}$$

$$\mathrm{Reduce}(k1, \mathrm{list}\,(v1)) \rightarrow \mathrm{list}(k2, v2)\,. \tag{8}$$

In MapReduce computing model, the computing process of such operation includes 5 stages:

a) Input stage: data input by users will be split into m data splits automatically and converted to (k/v) form to be assigned to m Map task(s). Each of Map tasks are assigned to a machine in the cluster where it is operated. These Map tasks are executed in parallel on different machines. Each Map task should be indicated with the input/output path and other operating parameters.

b) Map stage: the output is provided in $\mathrm{list}(k1, v1)$ key value pair form after (k/v) key value pair is processed with user-defined Map operation in Map function.

c) Shuffle stage: data processed by Map tasks will be split before Reduce function is called. Key value pairs with the same keyword are combined to form ($k1$, list($v1$)). Each ($k1$, list($v1$)) will be assigned with a Reduce task. Each of the Reduce tasks is also assigned to a machine in the cluster. In this way, there will be multiple Reduce tasks being executed in parallel throughout Hadoop cluster.

d) Reduce stage: this stage executes the user-defined Reduce function for each sole ki key value. The result $\mathrm{list}(k2, v2)$ is exported after Reduce function completes the execution.

e) Output stage: this stage writes Reduce output result in the output directory file.

3.2. Quantum ant colony algorithm (QACA)

The below describes QACA algorithm based on 0/1 knapsack problem that is described that: n item and 1 knapsack are given. Item i weighs wi (i=1, 2, ..., n) and values vi. The knapsack has a capacity of c. Some of these n items are selected to be put into the knapsack provided that the weight of items in the knapsack does not exceed c and the total value is the maximum. When the ant colony algorithm is used to solve 0/1 knapsack problem, more pheromone aggregating on a certain item means higher probability for the item to be selected. In QACA, the pheromone of the ant aggregating on the item is subject to qubit encoding. The qubit of the item carried by the ant is updated by using the quantum rotation gate. The updating of pheromone aggregating on the item turns to the updating of qubit probability amplitude.

Process of QACA algorithm [7]:

a) The quantum ant colony is initialized: The quantum ant colony is initialized: $A\left(t\right) = (\propto_1^t, \propto_2^t, \ldots, \propto_n^t)$, number of ant: n. The number of quantum bits is the number of item(s) m. $\propto_i^t$ $(i = 1, 2, \ldots, n)$ is quantum ant i for iteration t in the population.

To ensure simultaneous occurrence of all states in the same probability in the initial search of the algorithm, the values of all $\propto_i, \beta_i (i = 1, 2, 3, \ldots, m)$ in A(0) will be $1/\sqrt{2}$.

b) By setting the values of parameters α, β and ρ, the maximum number of iteration is $NMAX$, the number of current iteration is t=0, and the pheromone is $\tau_i\left(0\right) = 1$.

c) Each ant individually builds up a solution. Ant $k(k = 1, 2, \ldots, n)$ randomly selects an item i to be packed into the knapsack. Items are selected for entry into the knapsack at the probability p_i^k of the remaining items to be selected calculated with the probability, till the knapsack can no longer accommodate any items. The probability of item selection is shown in Formula (4).

$$o_i^k = \begin{cases} \dfrac{\left[\tau_i(t)\right]^\alpha \left[\eta_i(t)\right]^\beta}{\sum_{S\in J(k)} \left[\tau_s(t)\right]^\alpha \left[\eta_s(t)\right]^\beta} & S \in J(k) \\ 0 & other \end{cases} \tag{9}$$

In Formula (4), $\tau_i(t)$ represents the amount of pheromone contained in item i on iteration t. Heuristic function $\eta_i(t)$ represents the value of unit mass of item I, say $\eta_i\left(t\right) = v_i/w_i$. α and β represent the amount of pheromone contained in the item and the weight of unit mass value of the item, respectively. $J(k)$ is the set of items not selected by ant k. Pheromone updating equation:

$$\tau_i\left(t+1\right) = \left(1-\rho\right)\tau_i\left(t\right) + \Delta\tau_i(k)\,. \tag{10}$$

$$\Delta\tau_i\left(k\right) = Q\left|\beta_i^t\right|^2\,. \tag{11}$$

$\Delta\tau_i(k)$ represents the amount of pheromone left on item i by ant k. Q is a constant. ρ is the volatility of pheromone $(0 \leq \rho \leq 1)$.

d) If n ant(s) build(s) their own solutions, go to Step e); otherwise, go to Step c).

e) The optimal solution built by m ant(s) in this iteration is recorded.

f) A(t) is updated by using the quantum rotation gate rule [11].

g) If the end condition is met, say $t > NMAX$, the optimal solution will be exported; if $t = t{+}1$, go to Step c).

3.3. MapReduce-based quantum ant colony algorithm (MQACA)

For 0/1 knapsack problem, the time complexity of QACA is $O(NMAX \cdot m \cdot n)$. The computing work is primarily done in Step 3 – independent solving process of the ant. MQACA algorithm completes the process of evolution of each generation of the population with MapReduce: Map completes the independent solving process of the ant. The index number of the ant family serves as the key. The optimal solution and quantum information of the ant is taken as the value. This part may be done in parallel. Reduce expresses the process of obtaining a comparatively solution and updating quantum ant information. The output information is converted to the format of Map input as the input of the next-generation Map function for the next generation cycle. The steps of MQACA algorithm are:

a) The population is initialized to generate key value pair (k, v) to be stored in hadoop file system as a file. k represents the index of the ant family. v represents the solution and quantum information of the ant.

b) Map function receives (k, v). The fitness value of each quantum ant is computed to generate the intermediate result list$(k1, v1)$. $k1$ represents the index of the ant family. $v1$ represents the solution obtained by one ant in the family.

c) Reduce function receives the key value pair list$(k1, v1)$ generated by Map function. The quantum rotation gate rule is used to update quantum ant and global pheromone so as to determine whether the maximum number of evolution generation is achieved. If yes, the optimal value will be exported. If not, the optimal value will be stored while list$(k2, v2)$ will be exported. $k2$ represents the index of the ant family. $v2$ represents the solution and quantum information of the ant.

list$(k2, v2)$ is stored hadoop file system for the next cycle.

Map stage: Map function mainly functions in that members of the ant family individually generate a solution, and the solution of each ant in the family is exported to generate list$(k1, v1)$ intermediate result. Map function is shown in Function 1.

Function 1 MQACA's Map Function

```
Function Map(k, v)
{
int n; //size of ant population in this map
int m;//total number of items
inti=0;
int j=0;
Compute the probability of items to be selected with Formula (4);
  while(i<n)
  {
    Randomly select an item to be put in the knapsack;
    while(j<m-1)
      {
      if(total weight of items selected by the ant < knapsack capacity)
      {
        Continue to select items not in the knapsack with the probability;
        Compute total weight of items selected by the ant;
      }
      else
      break;
      j++;
    }
    i++;
    Update solution L of ant i;
  k1=k;
  v1=L;
  Emit(k1, v1);
  }
}
```

Reduce stage: Reduce function receives the key value pair exported by Map function. The main function of Reduce function is to decompose the solution and value of members of the ant family. This is to obtain the optimal solution and value. The quantum rotation gate rule is then used to update the quantum information of members in the ant family. Pheromone files are updated with Formula (5) to determine whether the end condition is met. If yes, the optimal solution and value will be exported. If not, the key value pair list$(k2, v2)$ will be stored in hadoop file system. $k2$ represents the index of the ant family. $v2$ represents the updated solution and quantum ant information. Reduce function is shown in Function 2.

Function 2 MQACA's Reduce Function

```
Function Reduce (k1, list(v1))
{
int n;//number of ant(s) in ant family k1
  for(inti=0;i<n;i++)
  {
Update quantum ant information A with quantum rotation gate rule;
      Update pheromone with Formula (5);
    k2=k1;
    v2=v1+A;
    Emit(k2, v2);
}
Write the updated pheromone in pheromone file;
  if(t<NMAX)
  {
    Store list(k2, v2) in hadoop file system
  }
  else
    Export optimal value and solution;
}
```

4. Experimental analysis

4.1. Performance test of optimization algorithm

The algorithms will be first put simulation comparison for their performance in standard test functions. The standard test functions used as follows:

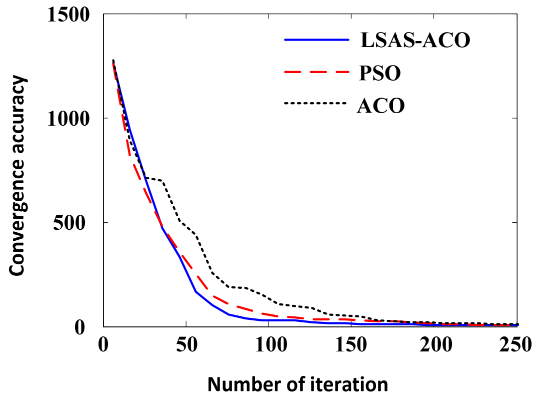
Griewank:

$$f2 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \ .$$

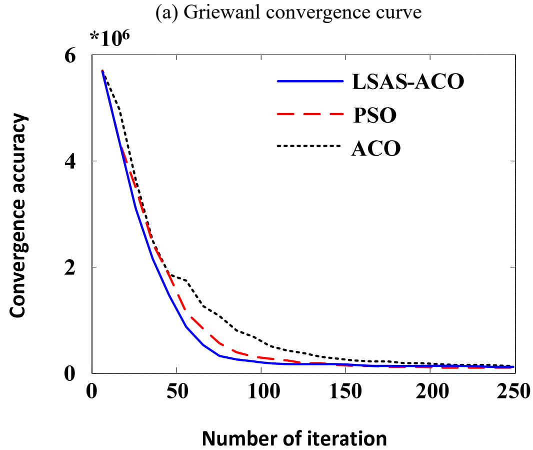
Rosenbrock:

$$f3 = \sum_{i=1}^{n} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right).$$

For the comparing algorithms, ACO, LSAC-ACO and PSO algorithms are selected. The stimulation comparison results between LSAS-AC algorithm and the comparing algorithm in convergence speed and accuracy (are shown in Fig. 1). Hardware equipment includes: CPU i3-2440, RAM 8G ddr3 1600. Stimulation software includes: matlab2012a. To ensure fairness of the results, the algorithms are operated for 20 times for the mean value. The number of iteration selected for ending the algorithm is $T_{\max} = 250$. Other parameter setting of water drop algorithm may be referred to in Literature [2].



(a) Griewanl convergence curve



(b) Rosenbrock convergence curve

Fig. 1. Convergence comparison

Fig. 1(a) and (b) give the convergence comparison curves of three comparing algorithms on the standard test function, respectively. It can be seen that PSO is slightly superior to LSAS-ACO in the early stage of the algorithm; however, LSAC-ACO has higher convergence speed than ACO and PSO in the whole. PSO has better convergence speed than LSAS-ACO. For convergence accuracy, LSAS-ACO does better than another two comparing algorithms.

4.2. Course design of laboratory micro learning resource

The course design level of laboratory micro learning resource may be evaluated with five levels below: exceptionally high [I], very high [II], fairly high [III], poor [IV] and worse [V]. The test data on the course design level of laboratory micro learning resource is the survey data for research assessment exercise of 20 domestic course designs of laboratory micro learning resource randomly selected by the College of Management Science in 2013, as shown in Table 1 [14].

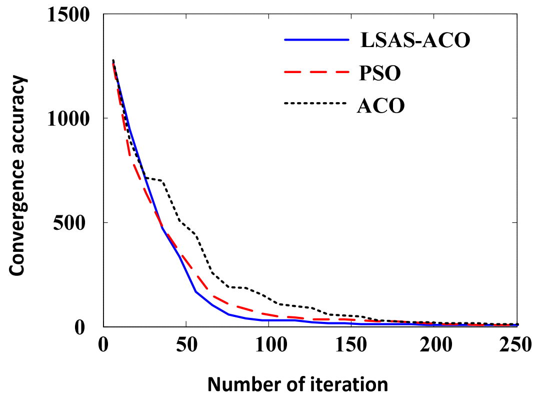
Table 1. Survey Data of course design level of laboratory micro learning resource

$x1$	$x2$	$x3$	$x4$	$x5$	$x6$	$x7$	$x8$	$x9$	$x10$	$x11$	Level
86	92	91	96	96	94	97	96	96	94	94	I
97	87	92	88	92	96	96	94	92	93	93	I
85	72	81	66	80	82	87	77	82	81	74	II
74	94	87	82	84	85	74	76	89	88	87	II
93	95	96	87	89	98	87	93	93	93	87	I
67	63	64	68	72	60	64	65	75	72	67	III
62	61	73	64	64	67	68	56	63	60	61	III
52	48	38	57	46	35	57	61	52	54	47	IV
45	56	45	45	58	53	44	57	55	61	44	IV
93	92	97	95	97	92	93	95	86	91	95	I
32	34	36	25	28	45	42	21	30	25	22	V
74	81	75	82	87	73	82	64	94	83	82	II
37	52	54	47	43	56	51	44	49	47	53	IV
15	33	21	36	44	34	32	37	34	25	39	V
25	44	32	12	37	28	37	26	22	17	31	V
66	67	64	71	67	57	63	60	64	66	65	III
57	64	63	63	65	68	73	66	70	68	64	III
77	75	72	79	71	87	75	92	63	72	83	II
36	17	28	27	23	23	23	30	38	37	27	V
91	93	87	93	96	92	91	87	93	86	98	I

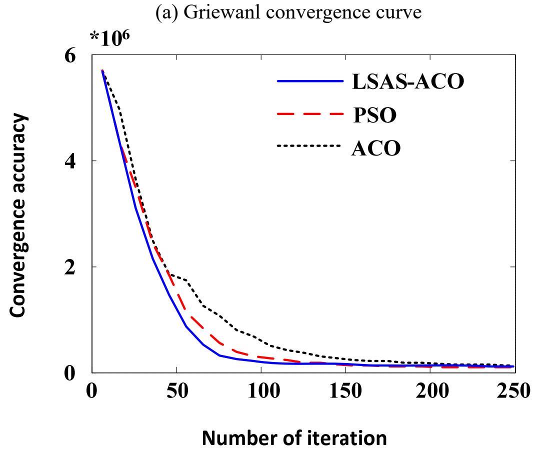
ACO, algorithm in this paper and in Literature [5] are selected as the comparing algorithms. 1-15 groups of data are selected from the experiment data in Table 1 the training of comparing network algorithms selected above. The test data is selected in two cases: (1) the test data is selected from the 15 groups of training data selected above. For example, data [3, 5, 8, 9, 11] is selected for the test; or (2) data outside the 15 groups of training data is selected for the network test. For example, the last 5 groups of data in the comparison experiment in this paper serve as the test data.

The training convergence processes of these two cases are shown in Fig. 2a – 2b. The evaluation results are shown in Fig. 9 and Table 4.

According to the training convergence comparison for the evaluation of course design of laboratory micro learning resource in Fig. 2, the method presented in this paper outweighs the comparing algorithms in both convergence speed and accuracy indexes compared to ACO and Literature [5]. It can be also seen that the convergence speed in scenario 1 of test data selection is faster than the one in scenario 2. Scenario 1 has 25 convergence generations, and the final convergence objective value is 0. For Scenario 2, it has 31 convergence generations, and the final convergence objective value is 0. In the comparing algorithms, the neural network optimized by ACO parameter does better than Literature [5] algorithm and ACO in terms of convergence performance. ACO algorithm has the poorest convergence performance.



(a) Griewanl convergence curve



(b) Rosenbrock convergence curve

Fig. 2. Model convergence course evaluation

5. Conclusion

This paper offers an optimization algorithm for virtual reality-based (VR) course design problem of laboratory micro learning resource of MapReduce parallel quantum ant colony algorithm. The course design problem of laboratory micro learning resource is modeled. The ant colony algorithm is introduced to optimize the model. In addition, the algorithm is improved for enhanced ant colony algorithm performance. The experimental results validate the effectiveness of the proposed algorithm. The system design of the algorithm is implemented for the optimization of course design problem of laboratory micro leaning resource by the algorithm. In this paper, QACA is parallelized by using MapReduce in Hadoop cloud environment. The proposed MapReduce-based QACA algorithm can be applied to the optimization design such as 0/1 knapsack problem, which provides it with a wide range of promising applications. The next step is to focus on studying the parameter optimization problem of the proposed algorithm.

Acknowledgement

Shandong Natural Science Foundation (ZR2013AQ018).

References

[1] Y. Du, Y. Z. Chen, Y. Y. Zhuang, C. Zhu, F. J. Tang, J. Huang: *Probing Nanostrain via a Mechanically Designed Optical Fiber Interferometer.* IEEE Photonics Technology Letters *29* (2017), 1348–1351.

[2] S. L. Fernandes, V. P. Gurupur, N. R. Sunder, N. Arunkumar, S. Kadry: *A novel nonintrusive decision support approach for heart rate measurement.* Pattern Recognition Letters. https: //doi.org/10.1016/j.patrec.2017.07.002 (2017).

[3] N. Arunkumar, K. Ramkumar, V. Venkatraman, E. Abdulhay, S. L. Fernandes, S. Kadry, S. Segal: *Classification of focal and non focal EEG using entropies.* Pattern Recognition Letters *94* (2017), 112–117.

[4] J. W. Chan, Y. Y. Zhang, and K. E. Uhrich: *Amphiphilic Macromolecule Self-Assembled Monolayers Suppress Smooth Muscle Cell Proliferation*, Bioconjugate Chemistry *26* (2015), No. 7, 1359–1369.

[5] M. P. Malarkodi, N. Arunkumar, V. Venkataraman: *Gabor wavelet based approach for face recognition.* International Journal of Applied Engineering Research *8* (2013), No. 15, 1831–1840.

[6] L. R. Stephygraph, N. Arunkumar: (2016) *Brain-actuated wireless mobile robot control through an adaptive human-machine interface.* Advances in Intelligent Systems and Computing *397* (2016), 537–549.

[7] Z. Lv, A. Halawani, S. Feng, H. Li, S. U. Réhman: *Multimodal hand and foot gesture interaction for handheld devices.* ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) *11* (2014) No. 1s, 10.

[8] Y. Z. Chen, F. J. Tang, Y. Bao, Y. Tang, G. Chen: *A Fe-C coated long period fiber grating sensor for corrosion induced mass loss measurement.* Optics letters *41* (2016), 2306–2309.

[9] D. S. Abdelhamid, Y. Y. Zhang, D. R. Lewis, P. V. Moghe, W. J. Welsh, and K. E. Uhrich: *Tartaric Acid-based Amphiphilic Macromolecules with Ether Linkages Exhibit Enhanced Repression of Oxidized Low Density Lipoprotein Uptake*, Biomaterials *53* (2015), 32–39.

[10] W. Pan, S. Z. Chen, Z. Y. Feng: *Automatic Clustering of Social Tag using Community Detection.* Applied Mathematics & Information Sciences *7* (2013), No. 2, 675–681.

[11] Y. Y. Zhang, E. Mintzer, and K. E. Uhrich: *Synthesis and Characterization of PEGylated Bolaamphiphiles with Enhanced Retention in Liposomes*, Journal of Colloid and Interface Science *482* (2016), 19–26.

[12] N. Arunkumar, K. M. Mohamed, Sirajudeen: *Approximate Entropy based ayurvedic pulse diagnosis for diabetics - A case study* TISC 2011 - Proceedings of the 3rd International Conference on Trendz in Information Sciences and Computing, art. No. 6169099 (2011), 133–135.

Received May 7, 2017