Review Paper on PSO in workflow scheduling and Cloud Model enhancing Search mechanism in Cloud Computing

Kavita Bhatt
Sudent-MTech, Department of Computer Science and Engineering
Poornima College of Engineering, Jaipur, Rajasthan, India

Dr. Mahesh Bundele
Department of Computer Science and Engineering,
Poornima Group of Colleges Jaipur, Rajasthan, India

Abstract-This is a review paper. Review is carried out in three parts. First part covered up the introduction to PSO algorithm, load balancing based on PSO and multi objective job shop scheduling by improved PSO. Second part of this paper focused on workflow scheduling in cloud based on various versions of PSO. Third part explained PSO with cloud model with a little result analysis. This survey is done to know the processing of PSO in workflow scheduling with cloud model and further demand more work in this direction.

Keywords- PSO, Work-flow scheduling, Cloud Model.

I. INTRODUCTION

Life before cloud was very tough as user had to install lots of stuff on their machines which might rarely use again. The main idea to develop cloud was to decentralize the application from hardware and OS. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth.

When working with cloud some challenges occurs. These are safeguarding data security, managing contractual relationship, dealing with lock-ins and managing cloud. In this survey we focused on challenge of managing cloud. When resource are available for services then it has to be managed and for this it is required to be properly scheduled for the processing in the cloud environment. Resource management for scheduling the work has the major concerned and hence this paper based on the review of PSO as optimization algorithm for better output and workflow scheduling based on PSO and cloud model used by PSO.

II. PSO

[James Kennedy and Russell Eberhart,1995] introduced method for optimization of continuous nonlinear function. Authors presented their work details using concepts from different areas. They covered social as well as animal behavior and hence studied bird flocking by Heppner, Grenander (1990[11]), Reynolds (1987)[12] and fish schooling by Wilson(1975)[13].Their paper described human's behavior and preferred optimization in terms of multidimensional space and collision free[10].

Paper described PSO concept in terms of precursors and revised the various stages of its progression from social simulation to optimizer. The precursors were nearest neighbor velocity matching & craziness, cornfield vector, elimination of ancillary variable like craziness and considered flock behaved as swarm, acceleration by Distance. Swarm has basis in literature with a paper by Millonas (1994) who developed his model and articulated five basic principles of swarm intelligence [15]. Authors selected the label to define the optimization concept was Particle Swarm. For optimizing continuous nonlinear function, authors presented Particle Swarm Algorithm in simple manner with its optimizer. They performed experiments by using Schaffer's f6 function [14]. Over a series of 10 training sessions, the particle swarm optimizer required an average of 284 epochs. The particle swarm optimizer was able to train the network so as to achieve 92 percent correct [10].

[Zhao Yongyi, et-at, 2010] presented a new algorithm PSOB for Load balancing on Multidimensional Network Services which provided effective & efficient Load Balancing of Network. Authors presented the related work done by them on high performance management of network [22] and load balancing as well as service composition [23][24] from 2002 to 2008. They also presented their study on service overlay network [24], multiple service instances [26] and multidimensional services [27] from 2003 to 2010.Clustering Algorithms considered as effective topology control which were used for balancing load on average nodes [25]. But as the load increased it avoided optimizing the objective and structure.

In Mathematical modeling for network values of service load, costs has been considered for the network and used for the calculation of an objective function. Inertia weight has been adjusted after then it used in load balancing process as significant parameter. The inertia weight considered critical for the PSOB quick convergence and hence it must be adjusted appropriately by maintaining fitness values of the particles using fitness function. This used in the load balancing algorithm for further processing [2]. Depends on fitness value and iteration, results determined by the algorithm, showed that PSOB achieved fast convergence which had inertia weight that played a major role in the optimization of load balancing process.

Based on the loading conditions, PSOB tried to balance load of network communication flows by auto adaptive operation through which adjusted services load on the network nodes. Mathematical modeling has been presented for the network which calculated an objective function. Inertia weight was adjusted which then used in load balancing process as significant parameter. PSOB algorithm was executed in standard C++ program using VC & GCC compiler. The data was produced by random function of normal distribution in MATLAB. Iteration number has lager value than fitness and the fitness began to wobble within 160iteration and converged at last, through which PSO's fast convergence proved. The iteration number was larger than the value which has been estimated. When relation between the value of εη and Iterations plotted into a graph, it showed that during iteration 10to150 fitness was changed accordingly but it began to wobble within 160 iterations and converge at last through which better performance of PSOB achieved [2].

Concrete congestion dispatching method and its technical study for PSO and the model experiment proved that the algorithm performed with better converging action and overarching adjustment [2].

[Li Li, et-at, 2010] woked on Multiobjective job shop scheduling problem and developed algorithm in which Traditional ACA combined with PSO used .This provide Feasible & effective approach for flexible job shop scheduling problem [28][31][32]. Authors presented related work from 1990 to 1995 based on few research papers. They presented totally new approach as PSO had fast convergence speed but later, the search and feedback information were not sufficient [10] and ACA could provide this [29][30]. Hence these gaps reduced by applying both algorithm in the same problem domain in two parts.

Their algorithm made full use the parameters of fast convergence of PSO and positive feedback of ACA determined by the same parameters, which were repeatedly updated by performing the comparison between old and new fitness values so that the solution would be improved and strengthen the search capability for optimal solution and quick convergence of algorithm. The finalized algorithm was capable to solve multi objective FJSSP in very effective and efficient manner [3][28][31][32].

First scheduling had been performed on the basis of transition possibilities and state transition rule, object function could be calculated. Local search was carried out and using that global updating had been carried out according to best scheduling. Optimum scheduling could be obtained for each subset and this procedure had been repeated until maximal subsets could be met. Experiments carried out tests with problem 4×5 with 12 operations and problem 8×8 with 27 operations. The value of object function was 14.8 for problem 4×5 and 26.0 for problem 8×8 which was lesser than the values calculated by applying other algorithms. Comparisons made in different five algorithms- temporal decomposition, approach by localization, Controlled GA, PSO+SA, AL+CGA, and (PSO+ACA) respectively. This comparison carried out between object function (F(C)), makespan (F1(C)) [1], total workload (F2(C)) and critical machine workload (F3(C)).This comparison study showed that the algorithm used in this paper provide effective results than other algorithms. presented algorithm (PSO + ACA) proved to be feasible and effective for multi-objective flexible job shop scheduling problem [3].

III. WORKFLOW SCHEDULING BASED ON PSO

[Suraj Pandey, et-at, 2010] developed a model for Computation & Data Transmission cost minimization on scheduling of workflow application. The specific question addressed in their paper was scheduling tasks of

workflow application and obtaining minimized computation cost. Authors presented the related work based on workflow systems and mapping of data termed as BRS from 2002 to 2006; work based on GA from 2002 to 2008 and also studied some papers based on PSO from 2002 to 2007.BRS solved only NP complete problems [19]and performance of GA was not that much good in case of workflow systems[17].
PSO could be able to find near optimal solution for mapping all tasks in the workflow for the given set of resources [21]. The algorithm selected the best resources to solve the workflow problem occurs in cloud environment while keeping the track of position value and fitness value which could be computed by fitness function [18]. On the basis of these values best position of particle was computed. During Experiments and implementations, performance computed using PSO was better than BRS as per the data and results [20][16]. For storing the input, authors used performance matrices. The values for PP-matrix taken from Amazon CloudFront.PC1 to be in US, PC2 in Hong Kong (HK) and PC3 in Japan (JP), respectively. The values for TP matrix taken from Amazon EC2's5 pricing policy for different classes of virtual machine instances. For DS-matrix, the sum of all the values in the matrix varied according to the size of data. Authors used JSwarm Package for their simulation [1].
Experiments performed on workflow application model with 30 independent execution, for 25 particles and for 45 number of iterations .In every execution, the x-axis parameters such as total data size (e.g. 1024MB), range of computation cost (e.g. 1.1-1.3 $/hour) remain unchanged, while the particle's velocity and position changed.PSO achieved at least three times lower cost for 1024MB of total data processed than the BRS algorithm. Also, the value of CI in cost given by PSO algorithm was +/- 8.24, which was much lower as compared to the BRS algorithm (+/- 253.04), for 1024 MB of data processed by the workflow. When range of compute resources increased from 0.1-0.3 to 1.1-1.3, BRS needed more total cost of computation i.e. 100-900 cents. But for the same range PSO required very less i.e. 10-30 cents. Fast convergence rate of PSO obtained by the graph between number of iterations and total cost of computation is, that proved the performance of PSO was far better than BRS.[1]

[Zhangjun Wu, et-at, 2010] proposed a model for Cloud Workflow [33] scheduling using a revised discrete Particle Swarm Optimization method which actually works on Execution Time & cost minimization. RDPSO algorithm could achieve better performance on makespan [1] and cost optimization. Authors presented the related work details from 1995 to 2010 using concepts of so many papers. They studied cloud workflow from 2008 to 2010 and then came to the point that Set based scheme [34][35] could applied to Revised Discrete PSO and that was able to fill the gap of expensive cost and more execution time.
Set Based concept introduced in PSO solved combinatorial problems [10][36] and when this concept used in RDPSO, then it was possible to minimize total computation cost of cloud workflow. RDPSO used GRASP to state that each particle in the initial swarm had a feasible and efficient solution. On the basis of position and fitness value better results were obtained which used less computation as well as transmission cost in the cloud network. In RDPSO candidate solution was obtained by the set of task-service pairs, in which particle learns from different exemplars and from other feasible pairs of different dimensions.[4]
RDPSO used GRASP(greedy randomized adaptive search procedure) to state that each particle in the initial swarm had a feasible and efficient solution. The particle's new position could be determined by selecting promising set of gbest and pbest which was generally learnt from its previous position. After then fitness value could be calculated. All data took from Amazon cloudFront. Population size was 50 for this experiment. RDPSO achieved relatively large optimization ratio of value 10 to17% on makespan. The total costs for different data size were compared between RDPSO,PSO and BRS. For data size 64MB RDPSO gave cost 56.7, PSO gave 80.41 and BRS gave 259.5 respectively. For other data sizes also RDPSO achieved lower cost. This proved that scheduling the large workflow could be more promising when applying RDPSO as compared to standard PSO or BRS. Set based theory applied to PSO yields outstanding performance on scheduling workflow applications in cloud environment.[4]

[Sheng-Jun Xue, et-at, 2012] presented an algorithm for time cost optimization on scheduling workflow applications in cloud environment. The specific question addressed in their paper was scheduling tasks of workflow application in discrete problem with improved local search ability while solving cost minimization problem. Authors presented the related work details from 2004 to 2011 using concepts from different papers. They studied cloud computing [37], its comparison with distributed as well as grid computing [38] [39]. They also studied resources used in cloud, NP hard problems solved in cloud [41][42], directed acyclic graph for solving network problem and PSO algorithm [40]. PSO had some disadvantages, such as poor local search ability and not suitable for problems in discrete areas[43]. For solving these shortages new workflow scheduling strategy for cloud computing proposed which was based on hybrid particle swarm algorithm (GHPSO).[5]
DAG used as mathematical model for their problem domain which stored input as set of tasks, data dependencies, execution time and computation costs.PSO selected for continuous area, crossover and mutation strategies for

discrete problem and further reducing premature convergence, hill climbing used. Cross-over strategy gave location exchange whereas mutation was responsible for maintaining population diversity. At every iteration crossover operation performed first on gbest then on pbest and after then mutation operation performed on resultant position, which decided the fitness value. Hill climbing performed on global extreme position vector to update individual vector pbest of particles. GHPSO used randomized strategy, multipoint crossover and uneven variation. Java programming language used for simulating the algorithm [5]. For experiments-compute sites ranged in interval [5,9];prices in [50,99];mission length in [100,199];iteration I=100 and population size N=30.For iteration 100,PSO run for 100 times, CPSO for 94.56 times and GHPSO for 80.05 times. GHPSO had faster convergence ability and search efficiency as compared to average number of iterations with PSO or CPSO. when the scale of 100 tasks used, then deadline at 100 sec took 1.2Lakh $ as computation cost in case of GHPSO which was 1.25Lakh in case of CPSO and 1.3Lakh $ in case of standard PSO respectively. The computation cost decreased as the deadline increased which showed the better performance of GHPSO compared than other two solution approach. Crossover and mutation strategy of GA with hill climbing algorithm introduced in PSO worked well with great solution quality including the consideration for deadline constraints. When deadline increases, solution quality of GHPSO got significantly higher with minimum cost as compared to PSO and CPSO.[5]

[Shaobin Zhan, et-at, 2012] presented an algorithm which solved large scale combination optimization for task scheduling. The specific question addressed was Scheduling of jobs and providing maximum benefit to cloud service provider and cloud computing user while solving large scale combinational optimization problem. Authors presented their related work details from 2007 to 2010 based on various research papers. They studied Job scheduling in Grid computing and found that this concept could implemented in cloud also [44][46]. They also studied QoS of job scheduling and its performance analysis. Most research papers rarely mentioned the differential service-oriented QoS guaranteed job scheduling system in a cloud without any good profit [45][47]. To overcome this gap, PSO with simulated annealing used.

The algorithm IPSO (Improved Particle Swarm Optimization) used simulated annealing in each iteration of PSO. The resultant algorithm improved the convergence rate and guaranteed the solution for combinational optimization problem. Meanwhile reversal variation strategy introduced which enhanced the population diversity. CloudSim used for implementation of the algorithm. IPSO shortened the average operation time of tasks, supplied proper resources to user task efficiently in the environment, increased utilization ratio of resources.[6]

When the scale of 250 tasks used, then average execution time taken by IPSO was 800T, which was 825T when ACO (Ant Colony Optimization) used, 850T when SA (Simulated annealing) used and 900T when GA (Genetic Algorithm) used. The execution time and efficiency of IPSO algorithm was better than the other algorithms. For solving precocious slow convergence speed of PSO at higher iterations and also discard sinking into local optima, IPSO used. Simulated annealing with PSO worked well for job scheduling problem and experiment proved algorithm's better convergence rate with high utilization ratio for resources. When number of tasks increases, average execution time of IPSO was comparatively lower than other algorithms.[6]

IV. PSO WITH CLOUD MODEL

[Changming Zhu, et-at, 2012] developed a model which worked on problem of premature or slow convergence rate for cloud model. The specific question addressed was DE related evolutionary algorithm applied in cloud based model replaced the roulette-wheel-selection model and used to improve artificial bee colony algorithm. Authors presented their related work details from 1997 to 2009 using concepts from various research papers. They studied Differential evolution and applied area [48][49], GA, PSO, vector generation strategies [51] and optimization performance and cloud based mode[50][52]l. DE suffered from premature convergence [49]. The cloud based model produced cloud drops of random and stable characteristics of orientation [52] but for pheromone and sensitivity model of free search algorithm authors presented CMDE(Cloud Model based Differential Evolution) algorithm.[7]

The mathematical cloud model had three factors Expectation (Ex), Entropy (En) and Hyper Entropy(He) respectively. Ex repres-ented the quality concept, En represented qualitative concept and He represented coagulation degree of cloud drop.

Cloud generator had two parts-forward generator and backward generator. Forward generator used Ex, En and He to generate cloud drop (x, μ) where x denoted quantity value and μ denoted membership degree of x. Backward generator used to convert quantity concept into quality concept i.e. (x, μ) into Ex, En, He. This cloud model was used in crossover operation of DE algorithm to avoid getting into local optimum. The resultant algorithm was CMDE in which first fitness value calculated, after then mutation and crossover with cloud model performed. [7] Then selection made between original individual and excellent individual. Algorithm run repeatedly until

termination condition gave optimal output in result. For solving slow convergence speed of DE algorithm, cloud model applied in the crossover operation of algorithm which gave better convergence rate and reliable result. For 9 benchmark functions, algorithm made 50 run and the result was compared to other algorithms. Without prior knowledge of user interaction, presented algorithm gave better performance.[7]
Cross over strategy implied into cloud based model with self adaptive rule for parameter settings, showed better convergence performance. For high conditioned elliptic function; the mean best value calculated by CMDE was 1.2341e-300(3.074e-300), for ODE 3.2240e-136(2.170e-136),for JADE 5.2620e-28(8.534e-27) and for jDE 3.4255e-28(6.321e-27) respectively. For other functions, we could get almost similar performances. So it could state that mean best value for CMDE was better than others. CMDE gave better performance in terms of convergence rate and reliability on both uni-modal and multimodal benchmark functions.[7]

[Ying Gao, et-at, 2010] presented an algorithm for performance optimization on cloud model. The specific question answered was global information about search space and local information of solutions provided by a single methodology which had PSO and CMBOA approaches together[10][36][53]. Authors presented their related work details from 2001 to 2010 using concepts from various research papers. They studied PSO in detail, its convergence nature, and search space. They also studied Cloud model, their qualitative and quantitative concepts and cloud model based optimization algorithm (CMBOA).PSO had less mechanism to extract and use global information about search space whereas cloud model used the concept of qualitative and quantitative concepts[10][52]. CMBOA directly extracted the global information from search space[53]. An evolutionary algorithm required which can used both local and global information which was provided in this paper.[8]
Cognitive population was used to estimate good solution regions and generate new particles in the search space. Three characteristics expected value, entropy and super entropy for the cognitive population was made via backward cloud generator. After then cloud drop used to generate same characteristics using forward cloud generator. New generation of Population generated through PSO particles and cloud particles. For improving search ability, cloud model applied with PSO which gave better result with weight factor. For 9 benchmark functions, algorithm made 50 run and the result was compared to original PSO. For schwefel's function the proposed algorithm had 0 as standard deviation where as in case of PSO it was 0.0111.[8]
The proposed algorithm was better with weight factor and remarkable standard deviation for all test functions. Presented algorithm was effective and had stronger global search ability than original version of PSO. local information from PSO particles and global information from cloud particles were used together to guide the overall search.[8]

[Yan Gao-wei, et-at, 2012] presented an algorithm based on cloud model for solving numerical optimization problem. The novel numerical stochastic optimization was done by algorithm which used the generation behavior, move behavior and spread behavior of cloud in a simple way. Authors presented their related work details from 1998 to 2006 using concepts from some research papers. They studied optimization problem, genetic algorithms and PSO[10][17][53]. They also studied cloud and its behavior. For resolving premature convergence they presented a new algorithm which was based on cloud model.[9]
Search space was disjoint into several separate regions. Cloud generation, Cloud move and cloud spread were the steps for the algorithm. Cloud generated for only those regions whose humidity value was higher than certain threshold and moved from higher air pressure value to lower air pressure value. The algorithm updated the humidity values and air pressure values for all regions every time after each step. The experiments were performed on windows 7 in MATLAB 7.1 with 1.0GB TRAM and 2GHz intel processor. Algorithm run for search space N=500 for four different benchmark functions. Results of the algorithm were compared with PSO and GA's results.[9]
Four test functions schaffer, needle in haystack, yang and rastingin applied for PSO, GA and ACMO respectively. In case of yang function's graph at iteration 20, fitness value was 1 for ACMO, but for PSO fitness value was 1 at iteration 60 and in case of GA fitness value was 0.8 at iteration 85.So it proved that ACMO worked better in convergence process when it compared with PSO and GA. ACMO algorithm had a good evolutionary ability by preventing premature convergence rate and it could escape from local optimum as compared to GA or PSO. It found that the presented ACMO algorithm could find optimum 100% for all functions except needle in haystack function.[9]

V. CONCLUSION

This paper presented an analytical review based on PSO. Particle Swarm Optimization algorithm has lots of features like good convergence rate, less expensive, easy to apply in different scenario and simple to implement. Job shop scheduling and load balancing are some of area where its feature makes the application easy to run. Further workflow scheduling can also performed with the help of PSO. Various versions of PSO exists and they can also implement in the application area. When cloud model applied with PSO then, the efficiency of result enhances and sometime it gave 100% results also. We can propose a new version of PSO with cloud model as our future scope of this review work which can be used for workflow scheduling.

REFERENCES

[1] Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar Buyya, A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments, 24th IEEE International Conference on Advanced Information Networking and Applications, 2010, ISBN: 978-0-7695-4018-4,Pg 400.
[2] Zhao Yongyi, Xia Shengxian, Research on Load Balancing for Multidimensional Network Services Based on Particle Swarm Optimization Algorithm, Third International Conference on Intelligent Networks and Intelligent Systems,2010, ISBN: 978-0-7695-4249-2, Pg 411.
[3] Li Li, Wang Keqi, Zhou Chunnan, An Improved Ant Colony Algorithm Combined with Particle Swarm Optimization Algorithm for Multi-objective Flexible Job Shop Scheduling Problem, International Conference on Machine Vision and Human-machine Interface,2010, ISBN:978-1-4244-6595-8, Pg 88.
[4] Zhangjun Wu, Zhiwei Ni, Lichuan Gu, Xiao Liu, A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling, International Conference on Computational Intelligence and Security, 2010,ISBN:978-1-4244-9114-8,Pg 184.
[5] Sheng-Jun Xue, Wu Wu, Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm, TELKOMNIKA Indonesian Journal of Electrical Engineering. Vol.10, No.7, November 2012.
[6] Shaobin Zhan, Hongying Huo, Improved PSO-based Task Scheduling Algorithm in Cloud Computing, Journal of Information & Computational Science 9: 13 (2012) 3821–3829.
[7] Changming Zhu, Jun Ni, Cloud Model-Based Differential Evolution Algorithm for Optimization Problems, Sixth International Conference on Internet Computing for Science and Engineering,2012, ISBN:978-1-4673-1683-5,Pg 55.
[8] Ying Gao, Xiao Hu, Huiliang Liu, Fufang Li, Cloud Estimation of Distribution Particle Swarm Optimizer, Fourth International Conference on Genetic and Evolutionary Computing, 2010,ISBN: 978-1-4244-8891-9,Pg 14.
[9] Yan Gao-wei,Hao Zhanju, A Novel Atmosphere Clouds Model Optimization Algorithm, International Conference on Computing, Measurement, Control and Sensor Network,2012,ISBN: 978-1-4673-2033-7,Pg 217.
[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1942-1948, IEEE Press, Piscataway, NJ, 1995.
[11] Heppner, F. and U. Grenander (1990). A stochastic nonlinear model for coordinated bird flocks.In *S* . Krasner, Ed., The *Ubiquity of Chaos.* AAAS Publications, Washington, DC.
[12] Reynolds, C. W. (1987). Flocks, herds and schools: a distributed behavioral model. *Computer Graphics,* 2 1 (4):25-34.
[13] Wilson, E.O. (1975). *Sociobiology: The new synthesis.* Belknap Press, Cambridge, hlA.
[14] Davis, L., Ed. (1991). *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, New York,*NY.*
[15] Millonas, M. M. (1994). Swarms, phase transitions, and collective intelligence. In C. G. Langton,Ed., *Artijicial* Life *III.* Addison Wesley, Reading, MA.
[16] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil,C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good,A. Laity, J. C. J
[17] A. Salman. Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems*, 26(8):363–371, November 2002.
[18] M. F. Tasgetiren, Y.-C. Liang, M. Sevkli, and G. Gencyilmaz. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, 177(3):1930–1947, March 2007.
[19] J. D. Ullman. Np-complete scheduling problems. *J. Comput. Syst. Sci.*, 10(3), 1975.
[20] J. Yu, R. Buyya, and K. Ramamohanarao. *Workflow Scheduling Algorithms for Grid Computing*, volume 146, pages 173–214. Springer Heidelberg, 2008.
[21] L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang. A task scheduling algorithm based on pso for grid computing. *International Journal of Computational Intelligence Research*,4(1), 2008.407
[22] Lu Jun, Qi Bing, " Multiple Ant Colony Algorithm Load Balancing for Network Sessions," Computer Applications, 2008, Vol. 28,pp.572-574.
[23] G. Dittmann, A. Herkersdorf, "Network processor load balancing for high-speed links," In 2002 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS2002), San Diego, USA, July,2002.
[24] Li Wenzhong, Guo Sheng, Xu Ping, "An Adaptive Load Balancing Algorithm for Service Composition," Journal of Software, 2006,Vol.17, pp. 1068-1077.
[25] Hu Rong, YANG Chun, HE Jun. "Clustering scheme based on simulated annealing algorithm in wireless sensor network," Journal of Computer Application, 30th ed,vol.2, pp.299-302
[26] Ou Rui, " The Research and Implementation of Multidimensional Service Composition Meachnism for Mobile Environment," MS thesis, Beijing University of Posts & Telecommunications, Department of Computer Science and Technology, 2009.
[27] Chen Wei, "The Study and Implement of the Load Balance Technique for Network Service," MS thesis, National University of Defense Technology Graduate College, 2004,pp. 15-16.
[28] P. Bruker, R.Schlie, "Job-shop scheduling with multi-purpose machines", Computing, 1990, pp.45: 369–375
[29] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies in European Conference on Artificial Life, Elsevier Publishing, Amsterdam ,1991, pp.134-142.
[30] M. Dorigo, T.Stutzle, Ant colony optimization, MIT Press, Cambridge, MA, 2004.
[31] I. Kacem, S. Hammadi and P. Borne, "Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems", IEEE Trans. Syst. Man Cybern, 2002, pp.(1):1–13.

[32] W. J. Xia, Z. M. Wu, "An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems",Comput. Ind. Eng, 2005,.pp. 48: 409–425..

[33] I. Foster, Zhao Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Proc. Grid Computing Environments workshop, 2008. GCE '08, pp. 1-10, 2008.

[34] M. Neethling and A. P. Engelbrecht, "Determining RNA Secondary Structure using Set-based Particle Swarm Optimization," IEEE Congress on Evolutionary Computation, BC, Canada,pp. 1670-1677, 2006.

[35] C. Wei-Neng, et al., "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems," IEEE Transactions on Evolutionary Computation, , vol. 14, pp. 278-300, 2010.

[36] X. Liu, J. Chen, Z. Wu, Z. Ni, D. Yuan, Y. Yang, Handling Recoverable Temporal Violations in Scientific Workflow Systems: A Workflow Rescheduling Based Strategy. Proc. of 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid2010), pages 534-537, Melbourne, Australia, May 2010. 1889

[37] Vaquero L, Rodero- Merino L, Caceres J, Lindner M. A break in the clouds: towards a cloud dentition. *ACM SIG-COMM computer communications review*.2009.

[38] Armbrust, M., Fox, A., et al.Above the Clouds: A Berkeley View of Cloud Computing. *Technical Report* No. UCB /EECS-2009-28, University of California at Berkley, USA .2009; February 10.

[39] M.A. Vouk. Cloud computing - issues, research and implementations.*Journal of Computing and Information Technology.*2008; 16 (4):235–246.

[40] Yu, J., Buyya, R.A taxonomy of scientific workflow systems for Grid computing.SIGMOD Record,Special Section on Scientific Workflows.2005; 34(3):44-49.

[41] B. Yu, X. Yuan, and J. Wang. Short-term hydro-thermal scheduling using particle swarm optimization method. *Energy Conversion and Management*.2007; 48(7):1902–1908.

[42] Veeramachaneni, K., Osadciw, L.A.*Optimal scheduling in sensor networks using swarm intelligence*.In: Proceedings of 38th Annual Conference on Information Systems and Sciences, 2004:17-19

[43] Pandey, S.; Linlin Wu; Guru, S.M.; Buyya, R. *Advanced Information Networking and Applications*(AINA), 2010 24th IEEE International Conference on Digital Object Identifier: 10.1109/AINA. 2010;400-407.

[44] Z. Shan, C. Lin, Modeling and performance evaluation of hierarchical job scheduling on the grids, in Sixth International Conference on Grid and Cooperative Computing, GCC 2007, 2007

[45] Y. Patel, J. Darlington, A novel stochastic algorithm for scheduling QoS-constrained workflows in a Web service-oriented grid, in 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, WI-IAT 2006 Workshops, 2006

[46] E. Carona, V. Garonneb, A. Tsaregorodtsevc, Definition, modelling and simulation of a grid computing scheduling system for high throughput computing, Future Generation Computer Systems,Vol. 23, No. 8, 2007, 968-976

[47] Weidong H., Y. Yang, L. Chuang, Qos performance analysis for grid services dynamic scheduling system, in International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, 2007

[48] R. Storn, K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, 1997, 11(4): 341–359.

[49] K. V. Price, "An introduction to differential evolution," Maidenhead:McGraw-Hill, 1999, 79–108.

[50] G. Jeyakumar, C. S. Velayutham, "A comparative performance analysisof differential evolution and dynamic differential evolution variants," *In:Proceedings of 2009 World Congress on Nature & Biologically Inspired Computing*, 2009, 463–468

[51] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Berlin:Springer, 2003.

[52] A. E. Eiben, R. Hinterding, Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 124–141.

[53] Ying Gao, An Optimization Algorithm Based on Cloud Model IEEE International Conference on Computational Intelligence and Security (CIS09) Vol.2,84-87.