Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110228

A Bio-Inspired Hybrid Computation for Managing and Scheduling Virtual Resources using Cloud Concepts

N. C. Brintha[1], *Shajulin Benedict*[2] *and J. T. Winowlin Jappes*[3,*]

[1] Department of Computer Science and Engineering, Kalasalingam Institute of Technology, Krishnankoil, India.
[2] HPCC Cloud Research Laboratory, St. Xaviers Catholic College of Engineering, Chunkankadai, India.
[3] Department of Mechanical Engineering, Kalasalingam University, Krishnankoil, India.

Received: 2 Jan. 2017, Revised: 13 Feb. 2017, Accepted: 18 Feb. 2017
Published online: 1 Mar. 2017

Abstract: Resource allocation and scheduling is one of the major issues in manufacturing industries which are constrained to offer dynamic and virtualized resources to end users in-order to maximize the profit. Cloud manufacturing is a new paradigm that can satisfy the requirements of modern manufacturing industries. In this work, two variants of heuristic algorithm are used to solve resource scheduling issues in casting industries. Particle swarm optimization algorithm is used in this work, because it can solve large scale optimization problems with better search speed, and genetic algorithms can be used to provide solution for non-linear and highly intricate engineering problems. This work uses a hybrid approach which combines the advantages of genetic algorithm with particle swarm optimization in-order to provide global convergence at effective and optimal cost. Experimentation was carried out for casting of engine block in manufacturing industry and the simulation results shows that PSO with GA provides global optimal convergence and also produces effective results with respect to time, cost and resource utilization.

Keywords: Cloud manufacturing, PSO algorithm, engine block, tasks and resources, GA-PSO algorithm, makespan

1 Introduction

In recent years, manufacturing sectors are more concerned with achieving dynamic challenges of global market, gathering and sharing product based information including knowledge in-order to maximize the profit of production lines. Cloud computing has provided a new horizon towards product design and manufacturing sectors. Cloud manufacturing fosters faster product development by using virtualization and resource sharing to promote cost reduction [1,2,3]. In cloud manufacturing, the resources that are distributed geographically are encapsulated in a centralized way as manufacturing cloud services. This way of resource centralization enable the end user to utilize the manufacturing resources based on their demand and need. Bansal and Darbari [4] stated that in a manufacturing enterprise, the assignment of tasks to resources is a dynamic process, because the resources are dispersed across several geographical locations.

The algorithms used for provisioning resources may be either centralized, decentralized, dynamic, static or even hybrid. Static algorithms allocate resources to tasks by using prior knowledge of the resources were as a dynamic algorithm maps resources to tasks based on demand. Decentralized algorithms lacks global awareness for optimal placement decisions whereas a centralized algorithm has full knowledge about its placement decisions. Hybrid algorithms may combine the best efforts of two or more algorithm for effectiveness [5]. By considering the cloud characteristics, for optimally allocating resources various scheduling algorithms may be used. Among the several algorithms, Particle Swarm Optimization [6], Genetic Algorithm [7], Ant Colony Optimization [8], Differential Evolutionary Algorithms [9], Artificial Bee Colony optimization [10], heuristic Bat algorithm [11] are most commonly used. These algorithms use the Virtual Machine (VM) as the scheduling and management units for mapping the heterogeneous physical manufacturing resources to manufacturing tasks.

In this work, the problem of allocating and scheduling resources in manufacturing industry is considered as a main issue because, if this is not taken in to consideration

* Corresponding author e-mail: winowlin@yahoo.com

then it will have an effect in cost and time of product development [12,13]. A multi-objective resource allocation strategy is considered in this work, which includes minimization of time, cost and energy of computation. Particle Swarm optimization and Hybrid PSO is considered for mapping tasks with resources due to its intrinsic capabilities like flexibility, optimal convergence towards solution and robustness.

The rest of the contents in this paper is organized as follows: Section 2 represents an overall discussion about the related works in this concern. Section 3 represent the problem description which is then followed by the proposed resource provisioning algorithm PSO and Hybrid PSO in Section 4. Section 5 reviews a case study on casting industry and proposes the simulation results achieved. Section 6, finally concludes the work.

2 Related works

Manufacturing industries are intended to mainly focus on quality of the work product and automating the scheduling process on production line which has an impact on cost and efficiency [14,15]. Shen et al., [16] represented the purpose of collaborating manufacturing process with cloud based technology to promote distributed virtualized scheduling with better cost and time. Li [17] proposed the initial definition, concept and architecture of cloud manufacturing which has been later reviewed in detail by several other researchers. Cloud manufacturing is a service oriented concept that uses virtualization to distribute and schedule resources across several geographic locations [18]. Fu [19] proposed a new resource selection model based on manufacturing grid. The diversified manufacturing resources from small and medium scale industries can be integrated and accommodated within cloud platform. Kumar and Verma [20] proposed an improved mechanism for task scheduling using genetic algorithm. Genetic Algorithm can be used as an optimal tool for scheduling the resources in-order to reduce makespan and to balance workload. It uses the reproduction operators like cross over and mutation for producing the population. Zhang et al., [21] proposed assignment approach based dynamic resource allocation for scheduling tasks on cloud system. Zhang [22] proposed that PSO based approaches can be used to schedule workflow across distributed applications. Luo et al., [23] introduced Cloud Rank-D model to rank and benchmark cloud based systems and analysed the performance of the system using two metrics: the amount of data processed per second and per joule. Genetic algorithm can be combined with PSO algorithm to generate a hybrid algorithm which produces global optimal solution at reduced cost [24,25,26]. Various researches have been carried out using heuristic based scheduling techniques for multi-objective optimization, however issues related to scheduling which encompass small and medium scale industries has not been addressed much [27,28].

3 Resource allocation model used in cloud manufacturing

This section discusses about the algorithm used during experimentation process. Two algorithms were used during the evaluation process which include the classical Particle Swarm Optimization and another algorithm that combines the parameters from Genetic algorithm with PSO to have global convergence with reduced cost. The latter algorithm is a hybrid approach which utilises some operators of genetic algorithm with a combination of particle swarm optimization.

3.1 Problem representation

The allocation problem is represented as a graph 'G' which has a set of nodes 'N' and edges 'E' that can be depicted as,

$$G = (N, E)$$

The nodes are used to refer Virtual Machines (VM) and the edges are used for representing the communication between tasks and virtual machines. All particles are initialized randomly and these particles build solutions by moving between virtual machines during each iteration until the entire tour is completed. The maximum number of iteration can be indexed using time 't'as $1 < t < max_i$

Assume that the tasks are to be executed using several computing nodes that are distributed across several computing sites. This work proposes a multi-objective function that minimizes cost, time and provides better resource utilization. The total execution time of all jobs on the available resources also called as makespan can be computed by summating the computation time of all individual tasks

Makespan or Total time of execution

$$= \sum CT_{max}(i,j) \tag{1}$$

where 'CT_{max}' is the maximal time taken for computation of task 'T_i' on resource 'R_j'.

The total cost, '$Cost_{Tot}$' can be calculated by using length of task and the processing cost of resource for executing these tasks.

$$Cost_{Tot} = \sum \frac{Task\ length \times Cost\ of\ exe\ per\ sec}{Virtual\ machine\ (MIPS)} + Processing\ Cost \tag{2}$$

The resource utilization rate 'Res_{util}' can be calculated by taking the ratio of time the resource is available and the total scheduling time of tasks on virtual machine. Virtual Machine (MIPS) is an estimation about the processing power taken by virtual machines which is executing on the host.

$$Res_{util} = \frac{Avail_{time}(VM)}{No\,of\,VM \times Sched_{time}} \times 100 \qquad (3)$$

where 'Avail$_{time}$' is the time manufacturing resource available to service the tasks, 'Sched$_{time}$' is the total time taken for scheduling the resources.

3.2 Overall scheduling workflow

The cloud manufacturing workflow consists of tasks which are to be computed using computing nodes that are geographically scattered across several locations. The tasks are randomly mapped to the virtual machines and the solution is updated accordingly. When the resource is available at the mapped site, the task is executed with the available resource. Otherwise some other resource that can satisfy the job requirements needs to be located. This include transfer of job from one site to another which may additionally include the transfer cost. The overall process of scheduling in this work is depicted in Figure 1.

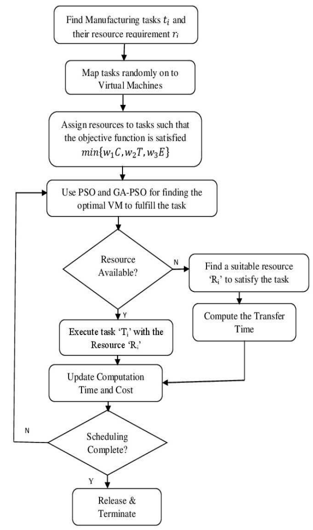

Fig. 1: Overall scheduling workflow of PSO and GAPSO algorithms

4 Task scheduling using PSO and GA-PSO

This section describes the algorithm used for scheduling 6 manufacturing tasks taken from casting of petrol engine block with 16 resources. Two algorithms were experimented for analysis and simulation purpose which includes the classical Particle Swarm Optimization and the other takes a hybrid procedure by combining the operations like cross over and mutation from genetic algorithm with PSO for optimal solution updation.

Particle Swarm Optimization (PSO) uses a number of particles that are initialized randomly in the solution space[29]. In order to mathematically describe the PSO concept, let the population size be 'n' and 'i' be the no of particles initialized in'm'dimensional search space as $X = (X_{i1}, X_{i2}, .., X_{ij}, .., X_{im})$.The speed of flight of particles '$V = (V_{j1}, V_{j2}, .., V_{jk}, .., V_{jm})$' and the individual position of particles, '$P = (P_{j1}, P_{j2}, .., P_{jk}, .., P_{jm})$'. The optimal position of particle'$P_{gbest} = (P_{gbest1}, P_{gbest2}, .., P_{gbestm})$'. The speed and individual position of particles among the population is updated using equations (4) and (5) as follows:

$$V_j(t) = wt.V_j(t-1) + c_1.random_1 \times (P_{gbest\,j} - p_j(t)) + c_2.random_2 \times (G_{gbest\,j} - p_j(t)) \qquad (4)$$

$$p_j(t) = p_j(t-1) + V_j(t) \qquad (5)$$

where:
V_j(t) Speed particle 'i' at time t
V_j(t-1) Speed particle 'i' at time t-1
wt Weight Inertia to control speed based on history
c_1,c_2 Acceleration coefficients
random$_j$ Random number which takes value between 0 and 1
p_j(t) Particle j's current position at time t
$P_{gbest\,j}$ Best position of particle 'j'
$G_{best\,j}$ Global/Optimal position of particle among the entire population
p_j(t-1) Particle j's position at time t-1

The general procedure behind particle swarm optimization algorithm is as follows:

* The dimension of the particle is initialized to size of ready tasks as $t_1, t_2, t_3, ...t_n \in T$
* The position 'p_j' and velocity 'V_j' of the particles are initialized randomly
* Calculate the fitness value of the particles initialized in the search space
* Analyse and compare the fitness value of the current solution with the previous best solution '$P_{gbest\,j}$'. If the new value for fitness is more effective, then assign that value as the new '$P_{gbest\,j}$'
* Repeat above steps for the entire particles and record the global best solution '$G_{best\,j}$' among the population
* Calculate and update velocity 'V_j' and position 'p_j' of the entire particles using equation (4) and (5)

∗ Continue with step 3 until termination criteria or the maximum number of iterations taken in to account is not reached

4.1 The hybrid PSO algorithm (GA-PSO)

Genetic Algorithm (GA) is a search based procedure that belongs to a class of evolutionary mechanism which uses the concept of genetics and natural selection. This algorithm starts up with a group of individuals which are randomly generated. After the initial population is produced at the end of each and every iteration i.e., generation, a new population is generated by applying a set of operators like mutation, selection and recombination.

The hybrid procedure that combines PSO with GA, executes both the systems simultaneously at the same time and identifies a set of individuals i from each of the systems after 'n' iterations [30]. The individual particle which has the largest fitness value has the best opportunity of being selected.

The procedure of hybrid GA-PSO is as follows:

- Initialize the population in a 'N' dimensional problem space.
- Find the value of fitness for the entire particles and rank them based on the evaluation.
- Apply the stochastic operators of genetic algorithm like crossover and mutation to create new particles.
- Use the concept of selection to select the particles according to their fitness.
- Apply crossover to update the particles position by using equation (6).

$$p_j = U_{rand}(0,1)p_j + (1 - U_{rand}(0,1))p_{(j+1)} \qquad (6)$$

where
p_j Position of the 'j^{th}' particle
U_{rand} Uniformly distributed random number that takes value between [0,1]
Apply 20% of mutation probability to update the particles position as indicated in the equation (7).

$$p_k = p_k + random \times N(0,1) \qquad (7)$$

where
p_k Updated particle position
N Gaussian distributed random number that takes value between [0,1]

- Use particle swarm optimization to revise the new value for velocity and position of the particles which has worst fitness value.
- Update velocity 'V_j' and position 'p_j' of the entire particles using equation (4) and (5).
- Evaluate and find the fitness value for the population and record this as the current and global best value.
- Repeat the steps 3 to 4 until the stopping criteria or the end if iteration is reached.

5 Experimental evaluation

This section presents the simulation and analysis results of a manufacturing problem in casting industry using cloud environment.

A case study on a casting industry was considered during the simulation process. According to the statistical report given by The Institute of Indian Foundryman, there are more than five thousand casting industries. So when considering countries other than India, the numbers are too high. Among these nearly 80% are Small and Medium scale Enterprises (SMEs). When casting a component, the tasks has to be completed by using different industries that may be located in different countries. When considering this factor, based on customer view point if the selection of task holder is inefficient, cost and time to complete the process may increase. In producer view point, raw materials and time will be wasted in a tremendous amount due to lack of design, simulation knowledge and facilities. To minimize these difficulties, a platform that combines these industries can be created using cloud technology which will guide the end users in selecting optimal resources across multiple sectors for task execution.

A four stroke petrol engine block casting is selected as a case study for this investigation. In order to improve performance at less weight and density, A319 and A356 aluminium alloys were selected. The material tensile strength ranges from 178 to 215MPa and its surface roughness is constrained to be not more than 1 to 1.5 m. The hardness of the die component should be 100 to 130 HB and the casting temperature is set around 660. By taking the above factors for consideration, the processing time required for experimentation has been simulated at different scales of production.

This investigation takes in to account, a set of manufacturing tasks like design of casting section, simulation, process and methods involved, secondary operations, finishing operation and quality check during the simulation process. The design process can be done through either Pro-E, Solid works or Catia, simulation process can be carried out using either Flow 3D, Magma, Solid flow or E-foundry, process and methods for casting can be either sand or die casting, secondary operations include several super finishing operations, finishing operations may include plating or painting and quality check may be carried out by either X-Ray or Ultra Sound. The resources used for the simulation process are geographically scattered and the tasks are allowed to select resources based on their resource requirement.

5.1 Parameter setting

Simulation was carried out using CloudSim toolkit which can effectively run on Linux and Windows systems [31, 32, 33]. The resource cost and the computation time of different jobs were taken from several input files for carrying out the simulation process.The same initial population is generated for both the algorithms in order to make a fair comparison. The control parameters used in PSO are the inertia weight 'wt' and the two constants c_1 and c_2. The parameters used for GA are, mutation and crossover probability which should be properly assigned to improve performance. The initial parameters depicted during experimentation process is represented in Table 1.

Table 1: Initial input parameters

Parameters	PSO	GA-PSO
Particle Dimension	2-10	2-10
Weight Inertia, wt	Decreases linearly from 0.9 to 0.4	0.5 + r/2.0, where r is a random number within [0,1] which is uniformly distributed
Acceleration constants, c_1 and c_2	2.0	2.0
Mutation probability	-	0.6
Crossover probability	-	0.5
No of iterations	Varying from 100-800	Varying from 100-800
No of VM_s	1020	1020
No of Data Centers	5	5
Physical hosts Dimension	Ranges between 2-6	Ranges between 2-6
Bandwidth Dimension	1000 kilobits per sec	1000 kilobits per sec
Transfer Cost per GB	7.33 to 29.33$	7.33 to 29.33$
Memory	204800mb	204800mb

Based on the task requirement, the input values taken for simulation process is shown in Table 1. This represent the approximation on time, cost, quality and bandwidth of executing the tasks on the resources.

By considering the above input values, the optimal sequence represented in Table 2 is formulated where CMSN represent the cloud manufacturing service nodes. This sequence provides better effort distribution, resource utilization and it also maximizes efficiency, productivity and will have effective machine utilization rate.

Table 2: Resource Allocation to Tasks

Optimal Resource Selection	No of Iterations
$CMSN_{12} \rightarrow CMSN_{23} \rightarrow CMSN_{33} \rightarrow CMSN_{41} \rightarrow CMSN_{53} \rightarrow CMSN_{62}$	600

5.2 Computation of execution time by varying workload

The workflow analysis and experimentation were carried out by varying the number of tasks which can also be called as cloudlets. For the simulation process, 16 resources related to casting of petrol engine block are taken for consideration. The casting workload is varied as 5, 10, 15, 20, 25, 30 and the virtual machines are set to 10 and 20 to obtain the computation result.

The comparison results of PSO and GA-PSO is shown in Figure 2.

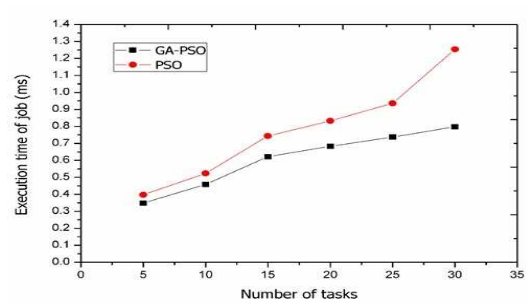

Fig. 2: Impact of Execution time when VM_s=10, runs= 200

When the number of virtual machines are increased, execution time of jobs have been minimized. This is due to the availability of more resources to service the task requirement. Also, GA-PSO algorithm outperforms the classical PSO approach in makespan computation. The variation in execution time when increasing the number of virtual resources is shown in Figure 3.

Thus the computational time of manufacturing task decreases due to the increase in virtual processing resources. The usage of improved GA-PSO algorithm is efficient when considering execution time than the classical PSO approach.

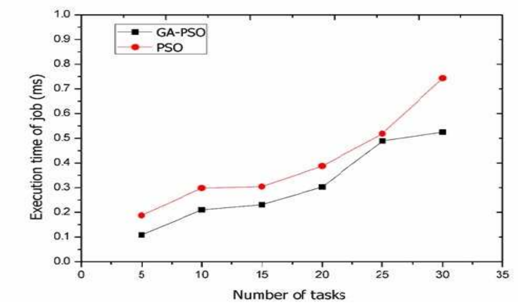

Fig. 3: Impact of Execution time when VM_s=20, runs= 200

5.3 Impact of load balance on resource utilization

When considering balancing of workload on the resources, the performance of workload distribution increases when considering resource without load balance. Hence when allocating resources to the casting tasks, if the data centers are balanced with their computation workload the performance of the overall system and the response time of jobs will be optimal. Figure 4 represents the improvement in performance when workload is evenly distributed across resources scattered across the computing nodes.

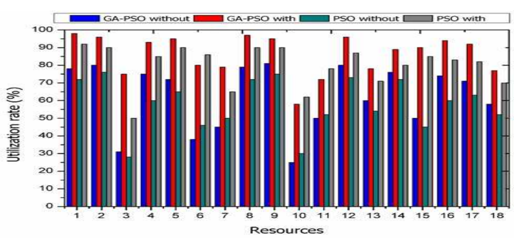

Fig. 4: Resource Utilization rate using PSO and GA-PSO with and without load balance consideration

When data centers are not heavily loaded and if the workload is evenly distributed between the resource providers performance of the overall system can be improved. In this work when considering this factor, GA-PSO algorithm performs well than PSO when workload is evenly distributed.

5.4 Impact of makespan on different runs

When the algorithm passes through several runs, the particle position is optimal during the solution update process. Due to this optimal solution, makespan is reduced when the number of iteration to reach the termination criteria for the algorithm increases. Thus the jobs are able to complete their processing within a minimum makespan which is depicted in Figure 5.

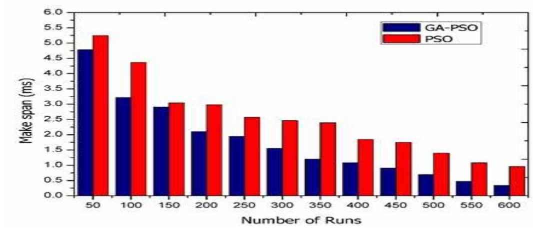

Fig. 5: Effect of makespan on the number of runs

5.5 Impact of cost by varying the workload size

When the number of jobs submitted to cloud is more, eventually cost of computation increases due to the fact that more work has to be performed on limited resources.

$$C_T = \sum_n (C_{exe}(i,j) + C_{trans}(j,K))\, for\ all\ i,\ j,\ k \in N \quad (8)$$

where
C_T Total cost of execution
C_{exe}(i,j) Cost of executing task 'T_i' on resource 'R_j'
C_{trans}(j,K) Migration or transfer cost if the requested resource is not available at the specified site.

When a resource is needed by a casting task, a search process is started to find a computing node that has the required resource. If the computing node that is nearest is overloaded, then rather than making the task to wait for a long time, the task can be transferred to the next available computing node. This in turn adds the migration or transfer cost on the total cost. The total computation cost is calculated for this investigation by varying the workload size among 16 resources. The virtual computing nodes are initialized for the experimental purpose and the cost of the work flow is simulated as shown in Figure 6.

6 Conclusion

This work is based on implementation of cloud based manufacturing model towards casting of cylinder blocks in small and medium scale enterprises by considering ten tasks and eighteen resources. Particle Swarm Optimization (PSO) and Genetic Algorithm combined with PSO (GA-PSO) was used for analysis and experimentation. Based on the experimentation on casting of petrol engine block, when stochastic genetic operators are used along with particle swarm optimization, due to global optimal convergence it provides effective results.

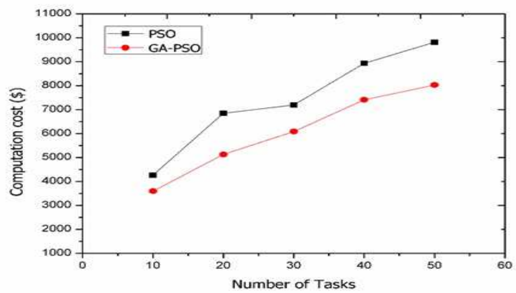

Fig. 6: Computation cost on casting workload

The experimental results show that, GA-PSO is better than the classical PSO algorithm in terms of makespan, computation cost and resource utilization rate. This investigation would enable the consumers on selecting resources in a fair way from different geographical locations with reduced execution time which in turn minimizes the cost of production.

References

[1] X. Xu, From cloud computing to cloud manufacturing, Robotics and computer-integrated manufacturing **28**, 75-86 (2012).

[2] B.H Li, L. Zhang, S.L Wang, F. Tao, J.W Cao, X.D Jiang, X. Songand and X.D Shai, Cloud manufacturing: a new service-oriented networked manufacturing model, Computer Integrated Manufacturing Systems **16**, 1-7 (2010).

[3] W. Dazhong, J.G Matthew, W. David, Rosen and Dirk Schaefer, Cloud manufacturing: Strategic vision and state-of-the-art, Journal of Manufacturing Systems **32**, 564-576 (2013).

[4] S. Bansal and M. Darbari, M. Multi objective intelligent manufacturing system for multi machine scheduling, International Journal of Advanced Computer Science and Applications **3**, 102-105 (2013).

[5] A. Al-maamari and F. Omara, Task Scheduling using Hybrid Algorithm in Cloud Computing Environments, IOSR Journal of Computer Engineering (IOSR-JCE) **17**, 96-106 (2015).

[6] G. Zhao, Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment, International Journal of Grid and Distributed Computing **7**, 33-42 (2014).

[7] X. Wang, C.S Yeo, R. Buyya and J. Su, Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm, Future Generation Computer Systems **27**, 1124-1134 (2011).

[8] X. Shengjun, L. Xiaolong and C. Jingyi, An ACO-LB Algorithm for Task Scheduling in the Cloud Environment, Journal of Software **9**, 466-473 (20114).

[9] J.T Tsong, C.F Jia and H.C Jyh, Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm, Computers & Operations Research **40**, 123045-3055(2013).

[10] J.Q Li and Q.K Pan, Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm, Journal of Information Sciences **316**, 487-502 (2015).

[11] Y.G Li and J.P Peng, An improved bat algorithm and its application in multiple $UCAV_s$, Applied Mechanics and Materials **442**, 282-286 (2014).

[12] J.F Yu, Y. Li, H.S Yuand and Q.A Shen, A Cloud Manufacturing Resource Allocation Model Based on Ant Colony Optimization Algorithm, Journal of Grid Distribution Computing **8**, 55-66 (2015).

[13] Moïse W. Convolbo and Jerry Chou, Cost-aware DAG scheduling algorithms for minimizing execution cost on cloud resources, The Journal of Supercomputing **72**, 985-1012 (2016)

[14] A. Ehab Nabiel, S. Lee and Rehman Khan, Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities, Future Generation Computer System **50**, 03-21 (2015).

[15] R. Zhang, A particle swarm optimization algorithm based on local perturbations for the job shop scheduling problem, International Journal of Advancements in Computing Technology **3**, 256-264 (2011).

[16] B. Shen, D.J Qui, L.Q Fanand and H. Meier, Collaborative engineering supporting technology for manufacturing in SOA, Computer Integrated Manufacturing Systems **17**, 876-881 (2011).

[17] B.H Li,Further discussion on cloud manufacturing, Computer Integrated Manufacturing Systems **17**, 449-457 (2011).

[18] D.C Zhan, X.B.Zhao and S.Q Wang, Cloud manufacturing service platform for group enterprises oriented to manufacturing and management, Computer Integrated Manufacturing Systems **17**, 487-494 (2011).

[19] J.Z Fu, An efficient resource-searching method in manufacturing grid, The International Journal of Advanced Manufacturing Technology **66**, 401-405 (2013).

[20] P.Kumar and A. Verma, Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm, International Journal of Advanced Research in Computer Science and Software Engineering **2**, 111-114 (2012).

[21] X. Xu,Dynamic resource provisioning in cloud computing: A randomized auction approach, IEEE Conference on Computer Communications (INFOCOM 2014) 433-441 (2014).

[22] L. Zhang, Y.L Luo, W.H Fan, F. Taoand and L. Ren, Analyses of cloud manufacturing and related advanced manufacturing models, Computer Integrated Manufacturing Systems **17**, 458-465 (2011).

[23] C. Luo, J. Zhan, Z. Jia, L. Wang, G. Lu, L. Zhang, Zhong Xu and N. Sun, Cloud Rank-D: benchmarking and ranking cloud computing systems for data processing applications, Frontiers of Computer Science **6**, 347-362 (2012).

[24] C. Song, X. Liu, W. Wang and X. Bai, A hybrid particle swarm optimization algorithm for job-shop scheduling problem, International Journal of Advancements in Computing Technology **3**, 79-88 (2011).

[25] K. Zhu, H. Song, L. Liu, J. Gao and G. Cheng, Hybrid Genetic Algorithm for Cloud Computing Applications, Proceedings of the IEEE International Conference on Asia-Pacific Services Computing Conference, Korea, 182-187 (2011).

[26] X. Shi, Y.H Lu, C.G Zhou, H.P Lee, W.Z Lin and Y.C Liang, Hybrid Evolutionary Algorithms Based on PSO and GA, Evolutionary Computing **4**, 2393-2399 (2003).
[27] H.B Li, Approach to multi-granularity resource composition based on workflow in cloud manufacturing, Computer Integrated Manufacturing Systems **19**, 210-216 (2013).
[28] P. Centobelli, G. Converso, T. Murino and L.C. Santillo, Flow shop scheduling algorithm to optimize warehouse activities, International Journal of Industrial Engineering Computations **7**, 49-66 (2016)
[29] E. Pacini, C. Mateos and C. Garcia Garino, Distributed job scheduling based on Swarm Intelligence: A survey, Computers and Electrical Engineering **40**, 252-269 (2014).
[30] C.F Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, Systems, Man and Cybernetics, Part B, IEEE Transactions **34**, 997-1006 (2004).
[31] R. Calheiros, R. Ranjan, A. Beloglazov, C. Rose and R. Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms., Software: Practice and Experience (SPE) **41**, 23-50 (2011).
[32] M.R Resmi and E. Arun, Foreground Segmentation for Live Videos by Texture Features, DJ Journal of Excellence in Computer Science and Engineering, **2**, 1-9 (2016).
[33] R. Sri Siva and B. Balachitra, Optimization of Machining Parameters using Scatter Search Algorithm, Journal of Advances in Mechanical Engineering and Science, **2**, 11-18 (2016).

N. C. Brintha was born in Nagercoil, India, in 1981. She received the B.E. degree in Computer Science and engineering from Manonmaniam Sunderanar University, India in 2002 and she received M.E Degree with distinction in Computer Science and Engineering from A.K.C.E, Anna University, Chennai. She is pursuing her P.hd degree in Anna University, Chennai under the guidance of Dr.Shajulin Benedict. Currently, she is working as Associate professor in Kalasalingam Institute of Technology, India. Her research interests include Cloud computing, Optimization, Scheduling, Cloud Manufacturing and so forth.

Shajulin Benedict graduated in 2001 from Manonmaniam Sunderanar University, India, with Distinction. In 2004, he received M.E Degree in Digital Communication and Computer Networking from A.K.C.E, Anna University, Chennai. He is the University second rank holder for his masters. He did his Ph.D degree in the area of Grid scheduling under Anna University, Chennai. He was affiliated towards the same group and published more papers in Int. Journals. After his Ph.D award, he joined a research team in Germany to pursue Post Doctorate under the guidance of Prof. Gerndt. Currently, he is a Professor at SXCCE Research Centre of Anna University-Chennai and he is heading the HPCCLoud Research Laboratory which was established by him at his workplace. He is also a Guest Scientist in TUM, Germany. His research interests include Grid scheduling, Performance Analysis of parallel applications (including exa-scale), Cloud Computing, and so forth.

J. T. Winowlin Jappes graduated in 1997 from Manonmaniam Sunderanar University, India in Mechanical Engineering. In 1999, he has completed his masters in Production Engineering from Annamalai University. He completed his Ph.D degree in 2004 in the area of Composite Deposition at Indian Institute of Technology Madras, Chennai. He has completed three DST funded research projects and published more than 125 research articles which includes 65 International Journal papers. Currently, he is working as Senior Professor in Kalasalingam University, India. His research interests includes High performance composite materials, machining of hard materials, optimization techniques, etc. At present, he is working as Professor and Head, Department of Mechanical Engineering, Kalasalingam University, Tamilnadu, INDIA.