Scientific Publications

Journal of Artificial Intelligence and Big Data, 2016, 1, 1291
www.scipublications.com/journal/index.php/jaibd
DOI: 10.31586/jaibd.2016.1291

Review Article

Scalable Task Scheduling in Cloud Computing Environments Using Swarm Intelligence-Based Optimization Algorithms

Srinivas Chippagiri [1*], Savan Kumar [1], Sumit Kumar [1]

[1] University of Utah, Salt Lake City, Utah, USA

*Correspondence: Srinivas Chippagiri (u1014717@utah.edu)

Abstract: Effective task scheduling in cloud computing is crucial for optimizing system performance and resource utilization. Traditional scheduling methods often struggle to adapt to the dynamic and complex nature of cloud environments, where workloads, resource availability, and task requirements constantly change. Swarm intelligence-based optimization algorithms, such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Artificial Bee Colony (ABC), offer a promising solution by mimicking natural processes to explore large search spaces efficiently. These algorithms are effective in balancing multiple objectives, including minimizing execution time, reducing energy consumption, and ensuring fairness in resource allocation. They also enhance system scalability, which is vital for modern cloud infrastructures. However, challenges remain, including slow convergence speeds, complex parameter tuning, and integration with existing cloud frameworks. Addressing these issues will be essential for the practical implementation of swarm intelligence in cloud task scheduling, helping to improve resource management and overall system performance.

Keywords: Cloud Computing, Task Scheduling, Swarm Intelligence, Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC)

How to cite this paper:
Chippagiri, S., Kumar, S., & Kumar, S. (2016). Scalable Task Scheduling in Cloud Computing Environments Using Swarm Intelligence-Based Optimization Algorithms. *Journal of Artificial Intelligence and Big Data, 1*(1), 1291. Retrieved from https://www.scipublications.com/journal/index.php/jaibd/article/view/1291

Received: March 6, 2016
Revised: April 19, 2016
Accepted: May 11, 2016
Published: June 28, 2016

1. Introduction

Cloud computing has great potential for a future of service computing, and it is being more and more used for "on-demand" and "utility computing" by both large and medium-sized enterprises. Cloud computing relies heavily on virtualization, a technology that allows for the simultaneous execution of numerous operating systems and applications on a single piece of hardware in order to deliver services through a virtual unit [1]. Automated host monitoring is only one of many benefits of virtualization technology, which also improves hardware utilization and reduces disaster recovery costs [2]. The problem with distributed computing is that it's hard to spread a lot of work to variable resources [3]. A number of variables, such as unequal distribution of resources, changing user needs, freshly joined nodes, and a high probability of failure in the overload nodes, might cause certain nodes to enter the overload state while others stay in the underload state.

A growing capacity of modern infrastructures, a ubiquity of network resources, and the decreasing cost of storage have given rise to cloud computing, where computational tasks are distributed across a network of physical and virtual resources [4]. The main objective is to leverage underlying infrastructure services to meet performance requirements and Service Level Agreements (SLAs) for users. Cloud environments consist of numerous physical hosts, each capable of running multiple virtual machines (VMs) [5]. The challenge lies in effectively allocating VMs to these hosts, taking into account varying resource demands and availability. Traditional approaches, such as static or random allocation, often lead to suboptimal resource utilization, increased response times, and inefficiencies in handling workload spikes [6].

Complex optimization issues have been found to be amenable to swarm intelligence optimization algorithms, which draw inspiration from the cooperative behavior of social creatures such as ants and bees. One such algorithm, Salp Swarm Optimization (SSO), simulates the movement of salps to find optimal solutions. Cloud computing task scheduling and resource allocation can be enhanced with SSO by utilizing the collective intelligence of salp swarms [7]. The method provides a flexible, scalable answer to the ever-changing needs of cloud computing, both in terms of activities and resources.

Task scheduling in cloud computing is inherently complex, requiring the balancing of multiple objectives such as resource utilization, workload distribution, and quality of service. Traditional scheduling techniques often fail to meet the scalability requirements of large-scale cloud systems [8]. Swarm intelligence-based optimization algorithms, particularly SSO, provide an effective way to address these challenges. These algorithms not only improve scheduling efficiency but also ensure that resources are allocated dynamically in response to fluctuating workloads, thus enhancing the overall performance of cloud environments.

1.1. Motivation and Contributions of the paper

The dynamic and complex nature of cloud computing environments necessitates efficient task scheduling to ensure optimal resource utilization, reduced execution time, and cost-effectiveness. Traditional scheduling approaches often fail to address the challenges posed by the heterogeneity and scalability of cloud infrastructures. This motivates the exploration of advanced optimization techniques, particularly swarm intelligence, which mimics natural behaviors to provide robust and adaptable solutions for task scheduling in cloud environments. A study aims to leverage swarm intelligence algorithms to address these challenges, offering innovative approaches for enhancing system performance, energy efficiency, and scalability. The key contributions include:

- Emphasizes optimization methods based on swarm intelligence and gives a thorough overview of cloud computing job scheduling approaches.
- Analyses the role of optimization in addressing critical challenges like resource utilization, makespan reduction, and energy efficiency.
- Explores various SI algorithms, including PSO, ACO, and ABC, highlighting their application in cloud task scheduling.
- Identifies key challenges and limitations of SI algorithms, such as convergence speed and parameter sensitivity, in cloud environments.
- Proposes insights for integrating SI algorithms with real-world cloud frameworks to improve scheduling efficiency.

1.2. Organization of the paper

In this paper, they lay out the layout: Cloud computing task scheduling basics are covered in Section II. Section III delves into optimization algorithms that are based on swarm intelligence. Cloud Task Scheduling using Swarm Intelligence is the topic of Section IV. Previous studies literature reviews are examined in Section V. Directions for Further Study Finish Section VI.

2. Fundamentals of Task Scheduling in Cloud Computing

In cloud computing, task scheduling is crucial for getting the best solution. The term "task scheduling" refers to the process of allocating specific days and times to complete various activities within predetermined parameters. Time is one kind of constraint, while resources are another. Cloud computing relies heavily on task scheduling. Scheduling tasks allows for the distribution of work across available processors, which in turn reduces execution time and makes better use of available resources. Both static and dynamic scheduling are available. The scheduling mechanism has multiple proposed algorithms [9]. Scheduling mechanisms play a crucial role in enhancing resource utilization.

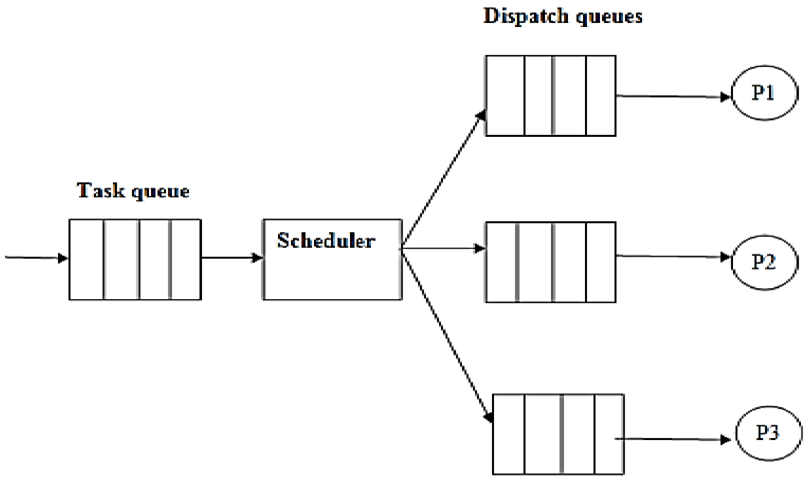

Figure 1. Task scheduling.

Figure 1 depicts the fundamental steps of the work scheduling process. In cloud computing, various tasks are queued up with varying priorities. After determining the relative importance of each task, the scheduler distributes them to available processors [10].

2.1. Role of Optimization in Task Scheduling

Efficiently allocating tasks to available resources in order to accomplish performance targets is what task scheduling is all about in cloud computing environments. Performance, efficiency, and user satisfaction are all affected by optimization, making it an essential part of the process [11]. Below is an exploration of the role of optimization in task scheduling:

1. **Enhancing Resource Utilization:** Optimization helps to guarantee the efficient employ of a cloud resources including a CPU, a memory and storage space. Through reduction of idle time and limiting resource excess, optimal scheduling increases the rates of using resources. This is mainly true in constantly shifting cloud environments, as the demand for resources is never constant [12].
2. **Reducing Makespan**: The total amount of time required to complete all of the jobs is known as Makespan. Minimization of makespan is possible with proper distribution of tasks across the available resources to reduce waiting time leading to efficient operation of workflows [13].
3. Ensuring Scalability: It is a known fact that in large-scale cloud environments, the number of tasks and available resources increases exponentially. Optimization is instrumental in scalability as it constantly changes to accommodate growing workloads and keep schedule proper even when the base is expanded.
4. **Balancing Workloads:** Skewed workload distribution that occurs can oppress certain resources whilst degrading the performance of others. Scheduling algorithms assist in distributing loads across resources while possibly none of the resources is over-consumed and others are underutilized [14].
5. **Minimizing Energy Consumption:** Use of energy in cloud data centers is significant thus leads to operation costs' incline with impacts the environment. In terms of energy efficiency in task scheduling, more tasks are grouped on the working resources, and unoccupied servers are powered off. Efficient technologies for energy management are important for green computing programs [15].

6. **Cost Efficiency:** Consumers employ the available services of the cloud by paying for the resources used a model referred to as the pay as you go model. Optimization is cost effective since it eliminates wastage by minimizing resource utilization and task accomplishment cost of cloud services.
7. **Meeting Quality of Service (QoS) Requirements:** Optimization helps to observe the set QoS parameters by offering the response time, throughput and reliability. Optimization is useful in making certain that service schedule agreements involving both the providers and the users are continually met in light of the existing QoS need [16].
8. **Addressing Dynamic and Heterogeneous Environments:** Users and workloads, as well as resources, are variable by nature, which necessitates dynamic cloud infrastructures. To deal with such changes, optimization techniques are needed to keep scheduling tasks more efficient even in the face of those heterogeneities and uncertainties.
9. **Supporting Multi-Objective Scheduling:** Often, in scheduling tasks, they come across conflicting criteria, which are cost, performance and power. Multi-objective optimization algorithms assist in identifying the compromises and obtaining the proper balance [17].
10. **Enabling Real-Time Scheduling:** Optimization plays a crucial role in applications where the response time has to be met, e.g., video streaming or IoT systems, to perform tasks on time [18].

2.2. Task Scheduling Challenges in Cloud Environments

Cloud computing's dynamic and complicated architecture presents a number of difficulties for task scheduling. These issues need to be resolved after maximize an employ of resources, effectiveness of a systems and economical cost. Below are some of the key challenges:

1. **Dynamic Resource Allocation:** Resource availability in cloud environments is a dynamic one because of the workloads and user demands. The most difficult task is to distribute resources in real-time based on these variations. Tasking scheduling algorithms need to be able to quickly reschedule task execution based on variations in resource availability. However, variables like task importance, runtime and resource availability make the allocation process more complex, which then needs complex optimization to balance between performance and response time [19].
2. **Scalability and Load Balancing:** Availability and ability to address the growing number of tasks and users are the key aspects of scalability in cloud systems. For the scale of the system, it becomes relatively difficult to distribute workload evenly across resources. This situation could result in resource bottlenecks, underutilization, or overloading, all of which are very damaging to system performance [20]. Thus, proper load balancing methodologies are essential to achieve fair distribution across the load, eliminate contention issues, or guarantee optimal performance in a continually expanding cloud environment [21].
3. **Energy Efficiency and Cost Optimization:** The use of energy in cloud data centers is also a big issue because of its consequences on company costs and the influence on the environment. If there is a problem with scheduling, then energy is wasted by having inactive resources work while other gears are overworking. Furthermore, the specificity of cloud services is that they are based on the pay-per-use scheme; thus, cost control is the most critical factor for introducing cloud services [22]. Task scheduling algorithms have to coordinate energy consumption with cost control and satisfy performance criteria, which in turn involves features like energy-aware scheduling techniques alongside consolidation of workloads [23].

3. Overview of Swarm Intelligence-Based Optimization Algorithms

An up-and-coming branch of AI, swarm intelligence (SI) was initially proposed in 1989 by Gerardo Beni and JingWang for use in creating cellular robotic systems. Machine learning algorithms such as SI are being widely adopted because of the high degrees of flexibility, portability, self-learning and incorporation of outside changes [24]. The growing number of NP-hard issues, in which obtaining a global optimum in real-time is extremely difficult, has piqued people's interest in them. SI is useful for addressing nonlinear design issues in many domains, including as optimization, data mining, bioinformatics, computational intelligence, corporate planning, and industrial applications. Malignant tumor identification, planetary motion sensing, micro-robot control, interferometry, navigation control, and image processing technologies are examples of high-end applications. Despite being a relatively new field of study, swarm intelligence has not been extensively published, with the exception of a handful overused prominent techniques.

3.1. Types of Swarm-Based Optimization Algorithms

This section explores the various types of swarm-based optimization algorithms.

3.1.1. Particle Swarm Optimization (PSO)

The 1995 introduction of PSO drew inspiration from the cooperative behavior of swarming organisms like fish and birds. Particles, representing possible solutions, navigate the search space in this method by mimicking the actions of the swarm's top performers and modifying their own pathways to get to more promising areas. Initially, particles are assigned random positions and velocities [25]. Their movement is influenced by two key factors: the global best-known position (gBest) and each particle's personal best-known position (pBest).

The algorithm tracks the gBest and a stopping criterion to determine when the optimization process should end. Each particle maintains data representing a solution, a pBest value indicating its best-known position, and a velocity that determines the direction and magnitude of its movement. Velocity is updated based on how far a particle's current position is from its target, with particles farther from the optimum requiring greater velocity adjustments [26]. This behavior is analogous to birds in a flock, where those farthest from the food source strive to catch up by flying faster. Over time, the particles converge around one or more optima.

3.1.2. Ant Colony Optimization (ACO)

The ant's pheromone communication and foraging habits served as inspiration for ACO. Ants initially investigate their surroundings at random. An ant marks a path for other ants to follow by leaving pheromones behind when it finds a food source [27]. Over time, more ants are attracted to paths with higher pheromone concentrations, leading to the identification of the shortest path to the food source [24].

The ACO algorithm leverages a positive feedback mechanism, where the shortest paths accumulate more pheromones and inherent parallelism as multiple ants work simultaneously. However, the algorithm faces challenges such as overhead and stagnation, where all ants converge prematurely on a suboptimal solution. In spite of these limitations, ACO produces good results when applied to optimization problems involving the search for optimal or nearly optimal solutions [28].

3.1.3. Artificial Bee Colony (ABC)

In 2005, the ABC algorithm was proposed to simulate honey bees' sophisticated foraging behavior [29]. A hive typically contains thousands of bees, each with distinct roles that evolve over time. Foraging bees are categorized into three groups: active foragers, scout foragers, and inactive foragers.

Bees forage nectar from identified and known food sources, and then determine the quality of the sources and then return to the colony. While foragers are idle at the door of the hive, waiting for data to be brought to them by other bees, the scout bees go searching for new areas containing food [30]. A waggle dance is a method through which active or scout bees inform the rest of the sleeping bees about the quality and location of food sources. By this exchange of information the colony is in a position to fine-tune its foraging strategy.

Whenever food is scarce in a certain area, the active will change position and may become inactive. Thus, the ABC algorithm effectively imitates this cooperative behavior, so it can be considered a rather effective means to address optimization issues [31].

4. Application of Swarm Intelligence in Cloud Task Scheduling

The way companies handle their computer resources has been revolutionized by cloud computing, which offers solutions that are scalable, adaptive, and cost-effective. Nevertheless, task scheduling, the process of efficiently assigning jobs to resources, is a significant challenge in cloud computing. Efficiently scheduling tasks leads to better system performance, shorter task completion times, and efficient use of resources. Due to cloud systems' complexity and dynamic nature, conventional scheduling approaches are insufficient [32]. Swarm intelligence (SI) has been investigated as a potential solution to cloud computing work scheduling issues as a result of this.

4.1. Application in Cloud Task Scheduling

The difficulties of scheduling tasks in cloud systems have been successfully tackled by using swarm intelligence algorithms. The following sections highlight their key contributions:

1. Load Balancing: SI algorithms distribute tasks among available resources to prevent overloading. For instance, ACO ensures balanced resource allocation by dynamically adjusting pheromone levels and guiding tasks to underutilized resources [33].
2. Minimizing Makespan: The entire time needed to finish all tasks is called makespan. Making real-time adjustments to job assignments in response to changes in particle velocity and position, PSO has found extensive use in reducing makespan [34].
3. Energy Efficiency: Cloud data centers consume significant energy. SI algorithms like ABC optimize task scheduling to reduce energy consumption by considering energy-efficient resources.
4. Scalability and Adaptability: SI algorithms are inherently scalable and adaptable, making them suitable for dynamic cloud environments. For example, FA adapts to changes in task requirements by adjusting the attraction mechanism [35].

4.2. Challenges and Limitations

While SI algorithms offer numerous advantages, they also face certain challenges:

- **Convergence Speed:** Some SI algorithms, such as ACO, may suffer from slow convergence in large-scale environments.
- **Parameter Sensitivity:** The performance of SI algorithms heavily depends on parameter tuning.
- **Resource Heterogeneity:** Handling the diverse nature of cloud resources remains a complex task [36].
- **Integration with Cloud Frameworks:** Implementing SI algorithms in real-world cloud platforms requires seamless integration with existing frameworks.

5. Literature Review

This section provides an overview of prior studies focusing on task scheduling in cloud computing environments utilizing swarm intelligence. A summary of the literature review is presented in Table 1.

Awad, El-Hefnawy and Abdel-Kader (2015) study algorithm that has been suggested takes availability and reliability into account. The difficulty in attaining these characteristics means that most scheduling methods do not take the cloud computing environment's availability and reliability into account. Suggest a computational model for cloud computing scheduling and allocation that incorporates a LBMPSO-based approach and accounts for dependability, execution time, transmission time, makespan, round trip time, transmission cost, and load balancing between virtual machines and tasks. Due to its ability to reschedule tasks in the event of resource allocation failure and take existing resources into account, LBMPSO can contribute to a reliable cloud computing environment [37].

Table 1. Summary of Literature Review based on Task Scheduling in Cloud Computing Environments Using Swarm Intelligence.

Reference	Focus Area	Technique Used	Key Findings	Limitations
[37]	Reliability and availability in cloud computing	Load Balancing Mutation Particle Swarm Optimization (LBMPSO)	Achieved reliability by rescheduling tasks and considered execution time, transmission time, makespan, round trip time, and cost.	High complexity due to multiple parameters and dynamic task rescheduling.
[38]	Task scheduling optimization	Multi-objective nested Particle Swarm Optimization (TSPSO)	Improved energy efficiency and processing time; balanced multiple objectives effectively.	Limited evaluation metrics; focus restricted to energy and processing time optimization.
[39]	System reliability and energy consumption	Analytical Model and Imperialist Competition Algorithm	Balanced system reliability, energy consumption, and QoS requirements using a hybrid meta-heuristic approach.	Potential scalability issues and high computational overhead in large-scale environments.
[40]	Performance of PSO in cloud workflows	PSO with parameter-wise hill-climbing heuristic (PSO-HC)	Improved scalability and cost-efficiency for CPU-intensive and I/O-intensive workflows on Amazon EC2.	Limited generalizability to non-Amazon EC2 platforms; focus on specific workflow types.
[41]	Task mapping and resource cost optimization	Cost-based scheduling algorithm	Enhanced communication/computation ratio and efficiency by grouping tasks based on resource capability.	May not handle dynamic workloads effectively; focus on coarse-grained tasks limits adaptability.

Jena (2015) implemented a TSPSO work scheduling system to optimise processing time and energy usage. The open-source cloud platform Cloud Sim has confirmed the TSPSO result. Lastly, after comparing the findings to other scheduling methods, it was decided that a suggested algorithm (TSPSO) achieves a best possible balance for a number of objectives. One way to boost cloud computing's overall performance is with task scheduling. Reduced power consumption and increased service provider profit due to faster processing are two more critical benefits of task scheduling [38].

Faragardi et al. (2014) provided an analytical model that may be used to assess not just energy usage and QoS needs, but also system reliability. In order to meet QoS objectives while keeping energy consumption and system reliability in check, the recommended model informs a novel online resource allocation method. The method is a novel imperialist-competition-based swarm intelligence approach that effectively merges the advantages of several popular meta-heuristic algorithms with a lightning-fast local search. The suggested algorithm's great efficiency is shown by a variety of simulation results derived from real data [39].

Ludwig (2013) study found that taking PSO and adding a local search heuristic improved its performance. Now that PSO with a parameter-wise hill-climbing heuristic, or PSO-HC, is available, it is feasible to execute processes that are both computationally and I/O heavy. Amazon Elastic Compute Cloud is used to conduct studies that measure the runtime scalability and cost-effectiveness of CPU-intensive and I/O-intensive operations [40].

Selvarani and Sadhasivam (2010) cloud computing platforms with an effective task mapping method that is cost-based. An algorithm measures resource costs and computation performance, allowing for better communication between coarse-grained jobs and resources. Additionally, it maximizes the computation/communication ratio by assigning grouped jobs to a cloud resource according to the processing capability of that resource, which groups user tasks together. Improved cloud computing efficiency is a result of this method's increased computation/communication ratio [41].

6. Conclusion and Future Scope

Cloud computing still remains one of the most effective paradigms to enable and/ or deliver highly scalable, flexible, efficient and cost-effective services. Resource management in dynamic cloud environments is extremely important for efficient scheduling of activities, optimization of resources and minimize time taken for their completion. Therefore, the utilization of task scheduling for cloud computing services is crucial to maximize resource utilization, shorten its execution time and achieve lower costs. Current approaches to scheduling do not meet the demands in real-time in large-scale clouds due to inefficiencies. Algorithms based on swarm intelligence, for instance, PSO, ACO, and ABC present optimistic solutions because utilize the features of workload interception and resources availability. These algorithms improve scalability, load balancing, energy utilization and general performance of the system. However, there are some questions that remain like time for convergence; dependence of parameters; integration with cloud platforms. Therefore, future research should be directed toward eliminating such shortcomings and enhancing the practical applicability of swarm intelligence algorithms for cloud task scheduling.

Further work in this area should include enhancing the sub-models by integrating them with cloud architectures in order to enhance scalability and performance. Further, to apply these techniques in a large-scale cloud environment, the rate of convergence needs to be increased, and the issue of parameter sensitivity has to be resolved as well. Further studies examining more comprehensive integration of swarm intelligence with other forms of optimization may be more helpful for task scheduling in cloud computing.

References

[1] L. Wang *et al.*, "Cloud computing: A perspective study," in *New Generation Computing*, 2010. doi: 10.1007/s00354-008-0081-5.
[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Serv. Appl.*, 2010, doi: 10.1007/s13174-010-0007-6.
[3] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, 2009. doi: 10.1109/NCM.2009.218.
[4] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Futur. Gener. Comput. Syst.*, 2009, doi: 10.1016/j.future.2008.12.001.
[5] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities," in *Proceedings - 10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008*, 2008. doi: 10.1109/HPCC.2008.172.
[6] M. Armbrust, A. Fox, and R. Griffith, "Above the clouds: A Berkeley view of cloud computing," *Univ. California, Berkeley, Tech. Rep. UCB*, 2009, doi: 10.1145/1721654.1721672.
[7] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, 2015, doi: 10.1109/ACCESS.2015.2508940.
[8] T. Mathew, K. C. Sekaran, and J. Jose, "Study and analysis of various task scheduling algorithms in the cloud computing environment," in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, 2014. doi: 10.1109/ICACCI.2014.6968517.
[9] V. Kumar and F. T. S. Chan, "A superiority search and optimization algorithm to solve RFID and an environmental factor embedded closed loop logistics model," *Int. J. Prod. Res.*, vol. 49, no. 16, 2011, doi: 10.1080/00207543.2010.503201.

[10] Aujla Sumandeep and Amandeep Ummat, "Task scheduling in Cloud Using Hybrid Cuckoo Algorithm," *Int. J. Comput. Networks Appl. (IJCNA*, 2015.
[11] V. V. Kumar, F. W. Liou, S. N. Balakrishnan, and V. Kumar, "Economical impact of RFID implementation in remanufacturing: a Chaos-based Interactive Artificial Bee Colony approach," *J. Intell. Manuf.*, 2015, doi: 10.1007/s10845-013-0836-9.
[12] I. Harjunkoski *et al.*, "Scope for industrial applications of production scheduling models and solution methods," *Computers and Chemical Engineering*. 2014. doi: 10.1016/j.compchemeng.2013.12.001.
[13] V. V. Kumar, M. K. Pandey, M. K. Tiwari, and D. Ben-Arieh, "Simultaneous optimization of parts and operations sequences in SSMS: A chaos embedded Taguchi particle swarm optimization approach," *J. Intell. Manuf.*, 2010, doi: 10.1007/s10845-008-0175-4.
[14] M. Akhtar, B. Hamid, and M. Humayun, "An Optimized Shortest job first Scheduling Algorithm for CPU Scheduling An Optimized Shortest job first Scheduling Algorithm for CPU Scheduling," no. January, 2015.
[15] V. V. Kumar, S. R. Yadav, F. W. Liou, and S. N. Balakrishnan, "A digital interface for the part designers and the fixture designers for a reconfigurable assembly system," *Math. Probl. Eng.*, 2013, doi: 10.1155/2013/943702.
[16] V. V Kumar, M. Tripathi, M. K. Pandey, and M. K. Tiwari, "Physical programming and conjoint analysis-based redundancy allocation in multistate systems: A Taguchi embedded algorithm selection and control (TAS&C) approach," *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, vol. 223, no. 3, pp. 215–232, Sep. 2009, doi: 10.1243/1748006XJRR210.
[17] J. Zhou, P. E. D. Love, X. Wang, K. L. Teo, and Z. Irani, "A review of methods and algorithms for optimizing construction scheduling," *J. Oper. Res. Soc.*, vol. 64, no. 8, pp. 1091–1105, Aug. 2013, doi: 10.1057/jors.2012.174.
[18] V. Kumar, V. V. Kumar, N. Mishra, F. T. S. Chan, and B. Gnanasekar, "Warranty failure analysis in service supply Chain a multi-agent framework," in *SCMIS 2010 - Proceedings of 2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering*, 2010.
[19] V. V. Kumar, F. T. S. Chan, N. Mishra, and V. Kumar, "Environmental integrated closed loop logistics model: An artificial bee colony approach," in *SCMIS 2010 - Proceedings of 2010 8th International Conference on Supply Chain Management and Information Systems: Logistics Systems and Engineering*, 2010.
[20] M. Sajid and Z. Raza, "Cloud Computing: Issues & Challenges," 2013.
[21] V. V Kumar, M. Tripathi, S. K. Tyagi, S. K. Shukla, and M. K. Tiwari, "An integrated real time optimization approach (IRTO) for physical programming based redundancy allocation problem," *3rd Int. Conf. Reliab. Saf. Eng.*, pp. 692–704, 2007.
[22] X. Zhu, C. Chen, L. T. Yang, and Y. Xiang, "ANGEL: Agent-Based Scheduling for Real-Time Tasks in Virtualized Clouds," *IEEE Trans. Comput.*, 2015, doi: 10.1109/TC.2015.2409864.
[23] V. V Kumar, "An interactive product development model in remanufacturing environment : a chaos-based artificial bee colony approach," Missouri University of Science and Technology, 2014. [Online]. Available: https://scholarsmine.mst.edu/cgi/viewcontent.cgi?article=8243&context=masters_theses
[24] M. Z. Hasan, R. Fink, M. R. Suyambu, and M. K. Baskaran, "Assessment and improvement of intelligent controllers for elevator energy efficiency," in *IEEE International Conference on Electro Information Technology*, 2012. doi: 10.1109/EIT.2012.6220727.
[25] M. P. Mathiyalagan, U. R. Dhepthie, and D. S. N. Sivanandam, "Grid Scheduling Using Enhanced Pso Algorithm," *Int. J. Comput. Sci. Eng.*, vol. 02, no. 02, pp. 1–6, 2010.
[26] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2010. doi: 10.1109/AINA.2010.31.
[27] R. F. Tavares Neto and M. Godinho Filho, "Literature review regarding Ant Colony Optimization applied to scheduling problems: Guidelines for implementation and directions for future research," *Eng. Appl. Artif. Intell.*, 2013, doi: 10.1016/j.engappai.2012.03.011.
[28] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," *Proc. - 2011 6th Annu. ChinaGrid Conf. ChinaGrid 2011*, 2011, doi: 10.1109/ChinaGrid.2011.17.
[29] N. Karaboğa and M. B. Çetinkaya, "A novel and efficient algorithm for adaptive filtering: Artificial bee colony algorithm," *Turkish J. Electr. Eng. Comput. Sci.*, 2011, doi: 10.3906/elk-0912-344.
[30] D. Karaboga, "An idea based on Honey Bee Swarm for Numerical Optimization," *Tech. Rep. TR06, Erciyes Univ.*, 2005.
[31] L. P. Wong, M. Y. H. Low, and C. S. Chong, "Bee colony optimization with local search for traveling salesman problem," in *International Journal on Artificial Intelligence Tools*, 2010. doi: 10.1142/S0218213010000200.
[32] Z. H. Zhan, X. F. Liu, Y. J. Gong, J. Zhang, H. S. H. Chung, and Y. Li, "Cloud computing resource scheduling and a survey of its evolutionary approaches," *ACM Comput. Surv.*, 2015, doi: 10.1145/2788397.
[33] F. Ramezani, J. Lu, and F. Hussain, "Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013. doi: 10.1007/978-3-642-45005-1_17.
[34] M. Z. Hasan, R. Fink, M. R. Suyambu, M. K. Baskaran, D. James, and J. Gamboa, "Performance evaluation of energy efficient intelligent elevator controllers," in *IEEE International Conference on Electro Information Technology*, 2015. doi: 10.1109/EIT.2015.7293320.
[35] L. Guo, S. Zhao, S. Shen, and C. Jiang, "Task scheduling optimization in cloud computing based on heuristic Algorithm," *J. Networks*, vol. 7, no. 3, pp. 547–553, 2012, doi: 10.4304/jnw.7.3.547-553.

[36] C. Y. Liu, C. M. Zou, and P. Wu, "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing," *Proc. - 13th Int. Symp. Distrib. Comput. Appl. to Business, Eng. Sci. DCABES 2014*, pp. 68–72, 2014, doi: 10.1109/DCABES.2014.18.

[37] A. I. Awad, N. A. El-Hefnawy, and H. M. Abdel-Kader, "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments," in *Procedia Computer Science*, 2015. doi: 10.1016/j.procs.2015.09.064.

[38] R. K. Jena, "Multi Objective Task Scheduling in Cloud Environment Using Nested PSO Framework," in *Procedia Computer Science*, 2015. doi: 10.1016/j.procs.2015.07.419.

[39] H. R. Faragardi, A. Rajabi, R. Shojaee, and T. Nolte, "Towards energy-aware resource scheduling to maximize reliability in cloud computing systems," in *Proceedings - 2013 IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2013*, 2014. doi: 10.1109/HPCC.and.EUC.2013.208.

[40] S. A. Ludwig, "Particle swarm optimization approach with parameter-wise hill-climbing heuristic for task allocation of workflow applications on the cloud," in *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2013. doi: 10.1109/ICTAI.2013.39.

[41] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," in *2010 IEEE International Conference on Computational Intelligence and Computing Research, ICCIC 2010*, 2010. doi: 10.1109/ICCIC.2010.5705847