International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)
ISSN (P): 2249–6890; ISSN (E): 2249–8001
Vol. 10, Issue 3, Jun 2020, 4167–4176


# UTILIZATION AND COST BASED PERFORMANCE EVALUATION OF CONVENTIONAL META-HEURISTIC SCHEDULING ALGORITHMS FOR HYBRID CLOUD ENVIRONMENT

**ASHUTOSH BHATT[1] & DR. PRITI DIMRI[2]**

[1]*Assistant Professor, Shivalik College of Engineering, Dehradun, India*

[2]*Associate Professor, GBPEC Ghurdauri Pauri, India*

***ABSTRACT***

***Cloud computing has gained a lot of popularity in recent times for the execution of data intensive or compute intensive tasks by scheduling them on cloud resources. Cloud architecture offers a huge pool of sharable resources that is easily accessible over internet through anywhere and also supports the pay per use model for its resources. However, scheduling tasks to cloud resources is a risky and challenging task as there are lot of QoS constraints associated with it. Also, the dynamic nature of the cloud makes the problem even more challenging. This article has tried to explore the four conventional meta-heuristic task scheduling algorithms, BSO, PSO, GA and DE; and, presented a statistical evaluation of the performance of these algorithms based on two parameters, utilization and cost. The results obtained are tabularly listed and graphically explained for better understanding.***

***KEYWORDS: Cloud Computing, Meta-heuristic Algorithms & Task Scheduling***

**Received:** Jun 09, 2020; **Accepted:** Jun 29, 2020; **Published:** Jul 20, 2020; **Paper Id.:** IJMPERDJUN2020396

**Original Article**

## 1. INTRODUCTION

Cloud computing is being rapidly adopted by the world as the new age computing paradigm. Cloud computing is an easy replacement of our age-old high performance computing systems, which consume a huge amount of electricity [1]. Another reason for the popularity of cloud computing is that a large number of idle resources can be pooled together and provisioned on demand to multiple users over the internet [2]. The good thing about these pooled resources is that they can be dynamically resized to suit the varied demands of application users based on their requirements [3][4].The roots of cloud computing lies in the traditional grid computing and distributed computing [5]. The difference is that the resources pooled in a cloud can be heterogeneous in nature [6].

One of the most significant activities in grid and cloud computing is the task scheduling problem. The aim of task scheduling is to provide the existing computing resources to various cloud users through the internet. The major challenge in providing these resources is that the resources may be heterogeneous in nature and geographically distributed [7]. Also, the resource allocation has to be done dynamically depending on the need and requirement of the user [8]. Task scheduling problem is a complex issue and involves a significant amount of risk as the appropriate resources have to be scheduled in order to perform the specific task also keeping in mind the various constraints involved such as energy consumption, deadline, CPU cost, etc [9] [10]. The fundamentals of task scheduling in cloud computing have been derived from the basic concept of scheduling in traditional operating systems. In cloud environment, the operating systems are simulated, they are able to utilize the shared resources in order to process the tasks allocated to them while enhancing the overall performance of the system [11].

A lot of research has been done in the field of task scheduling in cloud environment and a lot of scheduling algorithms have been proposed [12]. However, the problem of task scheduling in cloud with enhanced throughput still remains an NP-hard problem and thus attracts a lot of researchers [13]. The task scheduling algorithms are broadly categorized as heuristic and meta-heuristic scheduling techniques. Meta-heuristic scheduling algorithms have proved to provide more optimized solutions and hence, all research in this field is based on meta-heuristic techniques. Several hybrid versions of the classic meta-heuristic techniques such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Brain-Storming Algorithm (BSO) and Differential Evolution (DE) have been developed.

In this paper, the above mentioned conventional scheduling techniques will be statistically evaluated based on two constraints utilization and cost. The results obtained will be compared with each other to find out the two most optimized algorithms among the four mentioned above.

The rest of the paper is organized as follows. Section 2 discusses the literature available and the work done in this field. Section 3 talks about the scheduling methodology adopted in a hybrid cloud environment. Section 4 describes the scheduling constraints to be considered. Results are discussed in section 5. And finally, section 6 concludes the paper and discusses the future scope.

## 2. LITREATURE REIVIEW

In this article [14], the authors have proposed a new, enhanced version of the PSO algorithm by defining a new fitness function. The enhanced PSO can be used for identifying parameters of dynamic nonlinear hysteric models. The authors claim that the proposed algorithm improves the global searching capacity and can also escape the premature convergence of the conventional PSO algorithm. The proposed algorithm was able to identify nine parameters when applied to LB dynamic stall model of rotor blade and was able to identify six parameters when applied to ADF model of elastomeric damper. The accuracy of the parameters identified by the proposed algorithm is largely influenced by PSO's randomness. For increasing accuracy by reducing randomness, the authors increased the number of repeated identifications. The number of repeated identifications will be directly proportional to the complicacy of the hysteric model.

In this article [15], the authors have proposed a new map-matching technique which is based on multi-objective genetic algorithm. The authors have also proposed some innovative techniques like dynamic programming and dynamic time wrapping. Dynamic programming was adopted to fast-track the calculation of genetic algorithm's fitness function while dynamic time wrapping was incorporated to assess geometric similarities between the observed route and the recorded trajectory. The results were generated by implementing the proposed algorithm on the data gathered from the street networks, which demonstrated good accuracy and optimized running time. The proposed algorithm was also able to process the GPS data collected from dense street networks with almost zero errors.

This article [16] introduced a new optimization technique based on the actual human brainstorming process and named it brain storm optimization technique. The authors were inspired by the fact that human behaviour is superior that any other living being like insects, birds, animals etc. Thus, they believed that the optimization algorithm based on human behaviour will give better results as compared to any insect or bird. The proposed algorithm was based on two functions. One was the unimodal function called Sphere function and the second was a multimodal function called Rastrigin function. The BSO algorithm was tested based on these two functions, and the results show the usefulness of the proposed algorithm.

In this article [17], the authors have focussed on minimizing energy consumption in cloud environment while proposing a game theory based task scheduling algorithm. The authors have considered a system which manages big data and propose that energy management and task scheduling can be better coordinated by applying game theory. The authors have a proposed a mathematical model for scheduling task to computing resources while considering reliability of scheduling algorithm. The experimental validation was carried out in two steps. In the first step, objective function value was evaluated in equilibrium state. And in the second step, multiple nodes system was considered while calculating the balancing capability. The experiments proved that the proposed algorithm was able to minimize energy consumption in cloud computing environment.

In this article [18], the authors have introduced an algorithm for workflow scheduling in hybrid cloud environment which is intended to enhance the throughput of the system by dividing workflows in several sub-workflows which are independent of each other. The authors have considered load balancing and the proximity of resources to distributively schedule these sub-workflows on volunteer resources. Each sub-workflow has its own deadline and if execution time of any sub-workflow missed the deadline then it is rescheduled to another public cloud resource. This rescheduling of sub-workflows increases the percentage of workflows meeting deadline thus, improving the system performance by minimizing the waiting time. The experimental results show that the proposed algorithm improves system performance as an average of 75% workflows meet deadlines while being scheduled on volunteer resources.

In this article [19], the authors have introduced a new scheduling algorithm known as Grouped Task Scheduling (GTS) algorithm, which schedules multiple tasks to cloud resources while taking into consideration the QoS requirements of each user. The said algorithm works by categorising tasks into various groups depending on some common attributes of the tasks such as task size, task type, task latency and user type. Once the tasks are grouped to their appropriate categories, they are then scheduled to the available cloud resources. The scheduling process is further divided into two steps, wherein, the first step includes deciding which group of tasks will be scheduled first which depends on their attributes. And, the second step involves deciding which task from that group will be scheduled first which is based on the minimum execution time. The proposed GTS algorithm was compared with the traditional min-min algorithm and TS algorithm and the results show that GTS was able to achieve minimum execution time and minimum latency. The research gap in the above mentioned work is that it was applied only to a set of independent tasks and the scheduling was not done in real time.

In this article [20], the authors have presented an energy-aware scheduling technique for task-based applications which aims at minimizing the energy consumption and the total execution time of the system. The algorithm presents a bi-objective function which makes a choice between the two parameters deciding which of them is more important for the current application. The authors have a proposed a model in this article for estimating the energy consumption based on the execution time of each application given a set of resources. The energy consumption estimation is done by aggregating the individual task execution time, VM requirements, data transfers and other background services. The authors have additionally proposed a technique which automatically extracts the power profile of each resource which is required during the calculations of energy estimations.

## 3. HYBRID CLOUD TASK SCHEDULING

### 3.1 Scheduling Problem

This article considers a cloud computing environment which is a combination of private and public clouds for the execution of independent tasks which could be data intensive or compute intensive. All tasks are required to be completed within a set deadline while processing on similar kind of VMs.

Assume a set of cloud providers CLP= $CLP_1$, $CLP_2 \ldots CLP_n$ where, private clouds are represented by $CLP_1$ and the external public clouds are represented as $CLP_2 \ldots CLP_n$. Set of virtual machines is represented as VM = $VM_1$, $VM_2$, …, $VM_i$ and application groups are represented as A = $a_1$, $a_2 \ldots a_w$. The deadline for each application $a_j$ ($j \in \{1, 2, \ldots w\}$) is denoted by $d_j$ and the runtime is represented as $r_j$. Also, the set of tasks is represented as $task_{lj} = \{tj_1, tj_2 \ldots t_{jLj}\}$, where $l = 1, 2 \ldots L_j$. In Equation (1) *I* indicates the number of VM types, *w* represents the number of applications, *Lj* denotes the number of tasks in the $j^{th}$ application. Also, if $qjv = 1$, the *j*th application uses VM type VM*v*; otherwise, it does not use this type. *Pv* represents the price of the $v^{th}$ VM type in *CP*1, *rj* symbolizes the runtime of each task in the $j^{th}$ application and also *s* indicates the number of cloud providers. *Yjlk* as defined in Equation (2) refers to binary decision variable, such that $yjlk = 1$ if *rjl* is allocated to CLP*k*; otherwise $yjlk = 0$.

$$profit = \sum_{j=1}^{w}\sum_{V=1}^{I} L_j\, q_{jv} P_v r_j - \sum_{j=1}^{w}\sum_{l=1}^{L_j}\sum_{v=1}^{I}\sum_{k=1}^{s} y_{jlk}\, q_{jv} c_{kv} r_j \tag{1}$$

$$\sum_{k=1}^{n} y_{jlk} = 1, \forall j \in \{1, 2, \ldots w\}, \qquad l \in \{1, 2, \ldots L_j\} \tag{2}$$

### 3.2 Scheduling Model

In a hybrid cloud architecture, the own resources such as memory, processor and storage can be provided through the private cloud while other resources can be taken on lease or rent from external public cloud providers as depicted in figure 1. The user requirements are sent to the request manager through the user interface, which then forwards it to the task scheduler. The task scheduler, in coordination with the resource manager allocates the resources for each task. The own available resources are provided by the private cloud, while the resources not privately available are obtained through the external public clouds. The resource manager keeps track of all the allocated resources and updates its list as soon as any resource is deallocated.

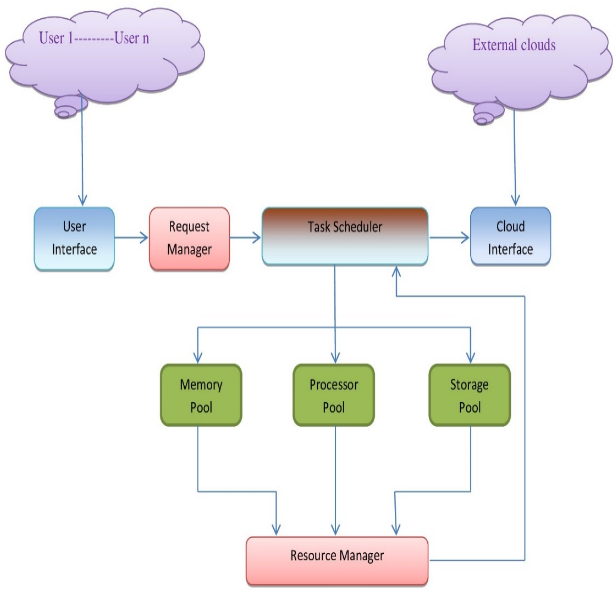


**Figure 1: Hybrid cloud scheduling model**

### 3.3 Scheduling Constraints

In this article, two scheduling constraints are taken into consideration, utilization and cost. The objective of the various meta-heuristic scheduling techniques considered in this article is to maximize the utilization and minimize the cost.

Utilization means the time period for which a VM is busy executing tasks. The system's utilization can be calculated by taking the average of all VMs utilization. It will represented as $U_t$.

The goal of cost function (C) will be to complete execution of as many tasks as possible with minimal time and cost.

Hence, the objective function considered in this article can be defined as in Equation (3).

$$obj = max\left(U_t + \frac{1}{C}\right) \tag{3}$$

## 4. RESULTS AND DISCUSSIONS

### 4.1 Experimental Setup

For the experimental analysis, four different meta-heuristic scheduling techniques namely, BSO, PSO, GA and DE were implemented in JAVA and simulated in CloudSim. All four algorithms were run different datasets consisting of different numbers of tasks and VMs.The description of the dataset in given in table 1. The given algorithms were compared based on the two scheduling constraints utilization and cost by executing them for similar datasets and providing them with similar resources.

**Table 1: Dataset Description**

| Data | Application | No. of Tasks | No. of VMs |
|---|---|---|---|
| Data 1 | Application 1 | 15 | 5 |
| Data 2 | | 15 | 10 |
| Data 3 | | 15 | 15 |
| Data 4 | Application 5 | 35 | 20 |
| Data 5 | | 35 | 25 |
| Data 6 | | 35 | 30 |
| Data 7 | Application 7 | 45 | 35 |
| Data 8 | | 45 | 40 |
| Data 9 | | 45 | 45 |

### 4.2 Utilization Analysis

The statistics presented in table 2 describe the utilization of each algorithm BSO, PSO, GA and DE respectively for different datasets as mentioned from Data 1 to Data 9. The values mentioned in table 2 are graphically represented in figure 2 which shows that BSO displays the maximum utilization while PSO stood second best in terms of utilization.

**Table 2: Utilization analysis of conventional meta-heuristic scheduling techniques**

| Algo | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Dat a 7 | Data 8 | Data 9 |
|---|---|---|---|---|---|---|---|---|---|
| BSO | 0.606 | 0.645 | 0.645 | 0.63 3 | 0.60 7 | 0.608 | 0.63 1 | 0.607 | 0.60 7 |
| PSO | 0.523 | 0.45 | 0.509 | 0.63 3 | 0.58 1 | 0.549 | 0.54 2 | 0.55 | 0.53 4 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| GA | 0.572 | 0.523 | 0.509 | 0.41 | 0.31 6 | 0.316 | 0.21 2 | 0.107 | 0.18 9 |
| DE | 0.606 | 0.523 | 0.509 | 0.41 | 0.24 9 | 0.282 | 0.22 | 0.155 | 0.20 4 |

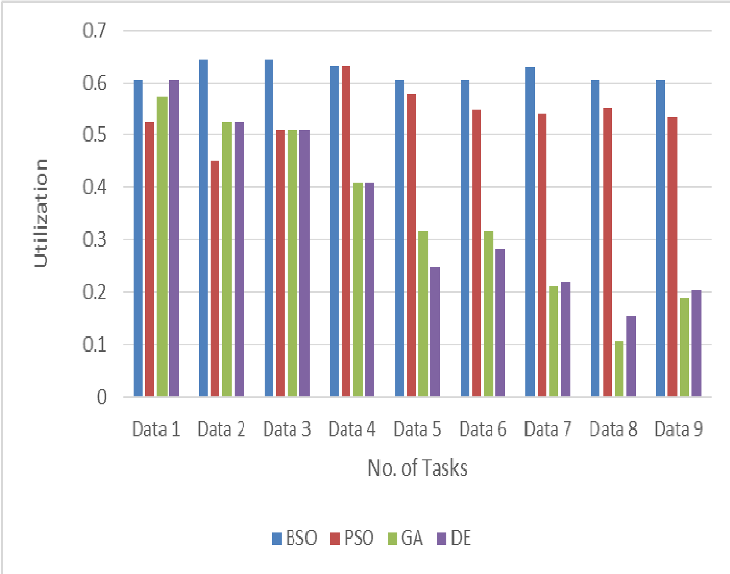


**Figure 2: Graphical representation of utilization analysis**

### 4.3 Cost Analysis

The statistics presented in table 3 describe the cost of each algorithm BSO, PSO, GA and DE respectively for different datasets as mentioned from Data 1 to Data 9. The values mentioned in table 3 are graphically represented in figure 3 which shows that BSO incurs minimum cost for most of the datasets while PSO stood second best in terms of cost for maximum datasets.

**Table 3: Cost analysis of conventional meta-heuristic scheduling techniques**

| Alg o | Dat a 1 | Data 2 | Dat a 3 | Dat a 4 | Dat a 5 | Dat a 6 | Data 7 | Dat a 8 | Data 9 |
|---|---|---|---|---|---|---|---|---|---|
| BS O | 0.1 6 | 0.16 | 0.1 6 | 0.2 1 | 4.4 | 2.0 3 | 1.05 | 7.7 1 | 11.5 |
| PS O | 0.4 2 | 0.42 | 0.4 8 | 1.4 | 2.3 5 | 2.0 8 | 4.77 | 13. 2 | 10.6 |
| GA | 0.1 6 | 0.88 | 1.0 4 | 3.5 6 | 1.3 4 | 5.3 4 | 19.5 | 19. 4 | 22.8 |
| DE | 0.1 6 | 0.16 | 1.0 4 | 3.2 7 | 1.6 3 | 4.1 4 | 22.4 | 19. 4 | 27.1 |

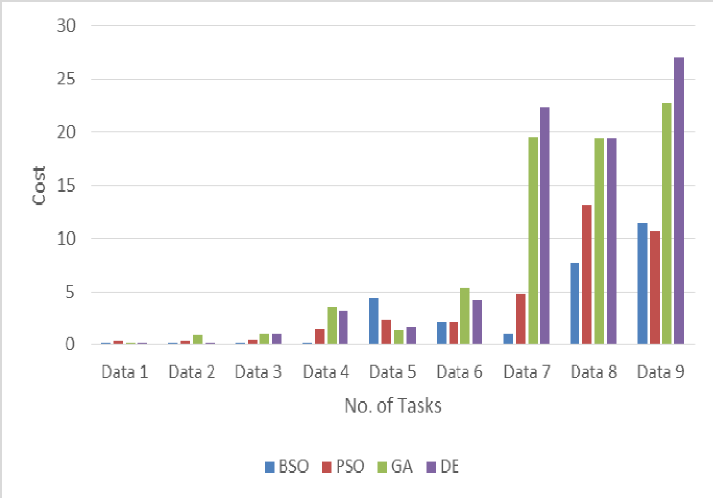


**Figure 3: Graphical representation of cost analysis**

## 5. CONCLUSION AND FUTURE SCOPE

In this paper, four different meta-heuristic scheduling techniques have been implemented and compared based on two parameters utilization and cost. BSO, PSO, GA and DE were implemented in JAVA and simulated in CloudSim. The experimental results show that BSO outperforms the other three algorithms in terms of both the scheduling constraints utilization and cost for most of the datasets considered for the experiment. Whereas, PSO depicted the second best performance for both the parameters while being implemented on same datasets.

For future work, BSO algorithm can be modified by merging it with either PSO or GA or any other meta-heuristic scheduling algorithm found suitable so that even more optimized algorithm can be obtained. Also, the optimized algorithm must be implemented for bigger datasets than considered in this paper and another scheduling parameter like makespan can also be considered.

***REFERENCES***

1. *Gregor von Laszewski, Lizhe Wang, Andrew J. Younge, Xi He, Power-aware scheduling of virtual machines in dvfs-enabled clusters, in: CLUSTER, 2009, pp. 1–10.*
2. *Ge M, Choo KKR,Wu H, Yu Y. Survey on key revocation mechanisms in wireless sensor networks. J Netw Comput Appl. 2016;63:24-38.*
3. *Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst. 2009;25(6):599-616.*
4. *JiangW,Wang G. Generating trusted graphs for trust evaluation in online social networks. Future Gener Comput Syst. 2014;31(1):48-58.*
5. *Kim S-S, Byeon J-H, Yu H, Liu H. Biogeography-based optimization for optimal job scheduling in cloud computing. Appl Math Comput.2014;247:266-280.*
6. *[6] Wang X, Wang Y, Cui Y. A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing. Future Gener Comput Syst. 2014;36:91-101.*
7. *Hao Y, Liu G. Evaluation of nine heuristic algorithms with data-intensive jobs and computing-intensive jobs in a dynamic environment. IET Softw. 2015;9(1):7-16.*

8. Wan C,Wang C, Pei J. A QoS-awared scientific workflow scheduling schema in cloud computing. Paper presented at: Information Science and Technology (ICIST), 2012 International Conference; March 2012:634-639.

9. Bhoi U, Ramanuj PN. Enhanced max-min task scheduling algorithm in cloud computing. Int J Appl Innov Eng Manag. 2013;2(4):259-264.

10. Tsai CW, Huang WC, Chiang MH, Chiang MC, Yang CS. A hyper-heuristic scheduling algorithm for cloud. IEEE Trans Cloud Comput. 2014;2(2):236-250.

11. Bahnasawy NA, Omara F, Koutb MA, Mosa M. Optimization procedure for algorithms of task scheduling in high performance heterogeneous distributed computing systems. Egyptian Inform J. 2011;12(3):219-229.

12. BHATTA, GOUTAM. "Cloud computing." International Journal of Computer Science and Engineering (IJCSE) 5.3 (2016): 7-16.

13. Nadeem F, Fahringer T. Optimizing execution time predictions of scientific workflow applications in the grid through evolutionary programming. Future Gener Comput Syst. 2013;29(4):926-935.

14. Liu C, Zhang X, Yang C, Chen J. CCBKE-session key negotiation for fast and secure scheduling of scientific applications in cloud computing. Future Gener Comput Syst. 2013;29(5):1300-1308.

15. Patel, Jay, Chirag S. Thaker, and Hardik Chaudhari. "Task Execution Efficiency Enrichment in Cloud Based Load Balancing Approaches." Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies. 2014.

16. J.Zhang, & P.Xia, An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models, Journal ofSoundandVibration (2016)

17. Marko Nikolic, Jadranka Jovic, Implementation of generic algorithm in map-matching model, Expert Systems With Applications (2016)

18. Bepery, Chinmay, et al. "Advanced home automation system using Raspberry-Pi and Arduino." International Journal of Computer Science and Engineering 8.8 (2019): 1-10.

19. Y. Tan et al. (Eds.): ICSI 2011, Part I, LNCS 6728, pp. 303–309, 2011. © Springer-Verlag Berlin Heidelberg 2011

20. J. Yang, B. Jiang, Z. Lv, K.-K.R. Choo, A task scheduling algorithm considering game theory designed for energy management in cloud computing, Future Generation Computer Systems (2017)

21. Bangar, Ashwini, and Swapnil Shinde. "Study and comparison of cryptographic methods for cloud security." Int J Comput Sci Eng Inf Technol Res 4.2 (2014): 205-213.

22. T. Ghafarian, B. Javadi, Cloud-aware data intensive workflow scheduling on volunteer computing systems, Future Generation Computer Systems (2014)

23. Gamal El Din Hassan Ali H et al., Grouped tasks scheduling algorithm based on QoS in cloud computing network, Egyptian Informatics J (2016), http://dx.doi.org/10.1016/j.eij.2016.07.002

24. F. Juarez, J. Ejarque, R. M.Badia, Dynamic energy-aware scheduling for parallel task-based application in cloud computing, Future Generation Computer Systems (2016), http://dx.doi.org/10.1016/j.future.2016.06.029

25. J. Williams, "Narrow-Band Analyzer," PhD dissertation, Dept. of Electrical Eng., Harvard Univ., Cambridge, Mass., 1993. (Thesis or dissertation)

26. E.E. Reber, R.L. Michell, and C.J. Carter, "Oxygen Absorption in the Earth's Atmosphere," Technical Report TR-0200 (420-

*46)-3, Aerospace Corp., Los Angeles, Calif., Nov. 1988. (Technical report with report number)*

27. *L. Hubert and P. Arabie, "Comparing Partitions," J. Classification, vol. 2, no. 4, pp. 193-218, Apr. 1985. (Journal or magazine citation)*
28. *R.J. Vidmar, "On the Use of Atmospheric Plasmas as Electromagnetic Reflectors," IEEE Trans. Plasma Science, vol. 21, no. 3, pp. 876-880, available at http://www.halcyon.com/pub/journals/21ps03-vidmar, Aug. 1992. (URL for Transaction, journal, or magzine)*
29. *J.M.P. Martinez, R.B. Llavori, M.J.A. Cabo, and T.B. Pedersen, "Integrating Data Warehouses with Web Data: A Survey," IEEE Trans. Knowledge and Data Eng., preprint, 21 Dec. 2007, doi:10.1109/TKDE.2007.190746.(PrePrint)*