(IJNCAA)

ISSN 2220-9085(ONLINE)
ISSN 2412-3587(PRINT)

# INTERNATIONAL JOURNAL OF NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Volume 5, Issue 2
2015

SDIWC

www.sdiwc.net

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS



**Overview**

The SDIWC International Journal of New Computer Architectures and Their Applications (IJNCAA) is a refereed online journal designed to address the following topics: new computer architectures, digital resources, and mobile devices, including cell phones. In our opinion, cell phones in their current state are really computers, and the gap between these devices and the capabilities of the computers will soon disappear. Original unpublished manuscripts are solicited in the areas such as computer architectures, parallel and distributed systems, microprocessors and microsystems, storage management, communications management, reliability, and VLSI.

One of the most important aims of this journal is to increase the usage and impact of knowledge as well as increasing the visibility and ease of use of scientific materials, IJNCAA does NOT CHARGE authors for any publication fee for online publishing of their materials in the journal and does NOT CHARGE readers or their institutions for accessing the published materials.

**Publisher**

The Society of Digital Information and Wireless Communications
Miramar Tower, 132 Nathan Road, Tsim Sha Tsui, Kowloon, Hong Kong

**Further Information**

Website: http://sdiwc.net/ijncaa, Email: ijncaa@sdiwc.net,
Tel.: (202)-657-4603 - Inside USA; 001(202)-657-4603 - Outside USA.



The issue date is April 2015.

IJNCAA

ISSN 2220-9085 (Online)
ISSN 2412-3587 (Print)

Volume 5, Issue No. 2 2015

CONTENTS

## ORIGINAL ARTICLES

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 48-52
The Society of Digital Information and Wireless Communications, 2015 (ISSN: 2220-9085)

# Application of Machine Learning Techniques for Yield Prediction on Delineated Zones in Precision Agriculture

Anshal Savla[1], Himtanaya Bhadada[1], Parul Dhawan[1], Vatsa Joshi[1],

[1]Department of Information Technology, NMIMS' MPSTME,
Mumbai, India

anshalsavla@gmail.com, himtanaya@gmail.com, paruldpsr@yahoo.com, vatsajoshi94@gmail.com

**Abstract.** Precision agriculture is the implementation of the recent technology in agriculture. Huge amount of data is collected in agriculture and various techniques of data mining are used to make efficient use of it. In this paper, we have discussed how with the help of both, clustering and classification algorithms, the crop suitable for a particular piece of land can be determined. Management zone delineation is a key task in this. From a data-mining point of view this comes down to variant of spatial clustering which has a constraint of keeping the resulting clusters spatially mostly contiguous. We analyze the need to discretize and normalize the data set and the various techniques that are used for the same. Further, a comparative analysis of the algorithm is shown where it can be seen which algorithm is best suited. We also talk about the future scope of the same and how these could actually be implemented in the real life scenarios.

## KEYWORDS

Discretization, Normalization, Clustering, Classification, Precision agriculture, Zone delineation.

## 1 INTRODUCTION

In recent years precision agriculture [1] has gained a lot of attention due to enormous possibilities it can open up in the field of agriculture. In one of our previous paper titled "Survey of Zone Tessellation Techniques for Defined Parameters in Precision agriculture", we had applied various clustering algorithms on a dataset and used zone delineation for predicting the yield of crops in a particular area. DBSCAN and ICEAGE algorithms had the best time complexity. In another paper titled "Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture" we have obtained the graphs and results presented in the paper. These are the result of the application of various classification algorithms on the dataset for predicting the yield of the crop. Further it was seen that Bagging algorithm gives the least error in predicting the seeds for the crop for a particular year.

The data set that we have used comprises of the following attributes: Area harvested, Seed, Yield and Production. The data is collected for the soybean crop. Yield is the actual generation of seed from the soybean crop, area harvested is the amount of the crop collected in a season and production is the quantity produced and actually harvested for the soybean crop. The data range over 53 years, starting from 1961 to 2013.

In the paper, we have provided a combined approach of both the techniques of zone delineation and prediction. First we divide the entire farm in zones using clustering technique and then we apply classifying technique for yield prediction of each zone. In section 2 of the paper we discuss zone delineation with respect to clustering techniques, in section 3 data preprocessing methods (normalization, discretization) are explained.

## 2 LITERATURE SURVEY

### 2.1 Normalization

The technique in which data is organized in a database is Normalization [2]. The process involves number of steps. It refactors the tables a table into number of smaller tables that are less redundant. No information is lost during the process and foreign keys are defined in old tables that refer to the primary keys in the new ones. Normalization is done in order to isolate the data so that the changes done in the attributes of one table can be sent to the entire database easily. Creation of tables and defining the relationships between them is done in order to protect the data.

Data normalization is a useful concept in organizing the data set. Without it the data system can become slow, inefficient and inaccurate. Normalization aims to organize the data into logical groups where every group effectively describes the small part of data. Also it becomes easy to modify the data in large database. This is because the change that is done is only at one place. Ease of access and quick manipulation of data is done by normalizing the date set.

$$Z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad \text{……….(1)}$$

where,

$x_i$= current value
$z_i$=normalized value
x=column vector

**2.2 Discretization**

Before data-mining, often data preprocessing is required. (ex: normalization, discretization)
When using data mining we usually work on large sets of data wherein an attribute value can vary over an extremely large range. It becomes necessary to reduce the number of values of a continuous attribute and divide the range into discrete intervals.

**Equal distance discretizer:**
The equal distance discretizer [3] is a simple, static and unsupervised method that divides the entire range of an attribute that has to be discretized into N equal intervals where the user specifies the value of N. The algorithm computes the minimum (Vmin) and maximum (Vmax) values for an attribute and divides it into k parts: intervals=(Vmax+Vmin)/K where 'K' is provided by the user and boundaries = Vmin+(
i * interval) for the i = 1...k-1 boundaries.

The limitations of this method are:

1. It is a parameterized method that requires input from the user.
2. The data values are not equally distributed over the intervals.

**Equal frequency discretizer:**
Equal frequency discretizer is a static, unsupervised and parametric method. It finds the minimum and maximum value attributes and then arranges all the values in increasing order. It then divides the entire range (n values) of an attribute in such a way that each interval has almost the same number of samples (interval= n/k) [4]. Thus, it overcomes the shortcomings of Equal distance discretizer.

**Chi2**
The chi2 algorithm is comprised of two parts. In the first part, the algorithm starts with computing a high significance level (sigLevel) for each numeric attribute that has to be discretized and then all the attributes are sorted according to its sigLevel [5].
It then calculates the Chi^2 for every pair of adjoining intervals. In the second part of the algorithm adjacent intervals with the lowest Chi^2[6] values are merged. This process continues until Chi^2 values of all interval pairs exceed the parameter determined by the sigLevel. The entire process is repeated for decreasing values of sigLevel until an inconsistency rate is exceeded in the discretized data.

$$\mathbf{x}^2 = \sum_{i=1}^{2}\sum_{j=1}^{k}\frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad \text{………(2)}$$

where[15],
K = number of classes,
$A_{ij}$ = number patterns in the ith interval, jth class,
$R_i$ = number patterns in the ith interval
$C_j$ = number of patterns in the jth class
N = total number patterns
$E_{ij}$ = expected frequency of $A_{ij}$ = $R_i$ * $C_j$ / N

Class attribute contingency coefficient discretization
This discretization method is proposed by Lee et al [7] which is based on the concept of class attribute contingency coefficient. This supervised and top down method is very efficient as different attributes are considered. Discretization of continuous data is done by measuring the value of CACC. The minimum and the maximum value of each attribute is determined and then all the values of each attribute is sorted in the ascending order. The attributes are partitioned according to the maximum CACC value into intervals.

### 2.3 Zone Delineation with respect to Clustering Techniques

In clustering techniques, zone delineation is a vital aspect. It helps us understand which part of the land is best suited for a particular type of crop. As opposed to homogenous crop selection methods like in traditional agriculture, we can have a heterogeneous crop selection using zone delineation in precision agriculture.
In DBSCAN and ICEAGE algorithms, the agricultural data attribute (k) is responsible for the density of the redundant cluster.
In our data set, there are fifty-three data entities and if we label each entity as 'k', then the density of the cluster increases since every individual cluster will be one single value. However if we have only three clusters of approximately seventeen data values each, then the density of the cluster will be very low.
Ideally the value of 'k' should be between 3 to 10.

### 2.4 Application of Classification Over Individual Zone

In previous papers, various classification algorithms are discussed. After analyzing all the algorithms from previous papers we came to a conclusion that bagging is the best algorithm for predicting the yield for data set used in previous papers. Therefore, bagging algorithm is used to predict the yield of crop over individual zones.

## 3 PROPOSED MODEL

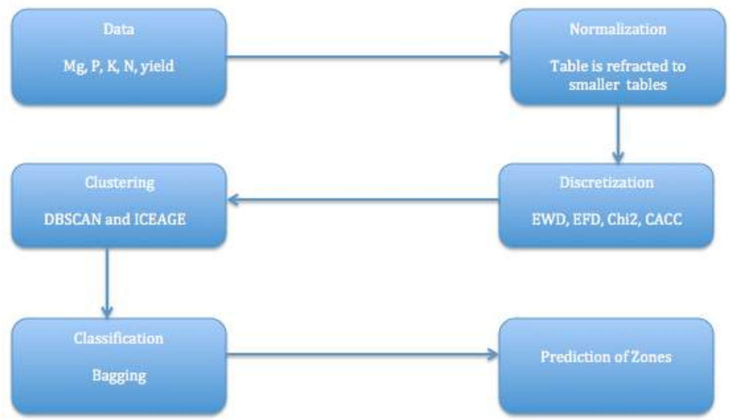



Figure 1: Block diagram

Step 1: Import data set
Step 2: The variables used in the data set are transformed into a specific range. This brings the data set into a consistent state and so anomalies are avoided. Once the data set is normalized redundancy is reduced and also managing of data becomes easier.
Further after refactoring is done old table are assigned the foreign keys that refer to the primary keys of the new ones.
Step 3: Continuous data set is then discretized. We have used Equal Distance Discretizer, Equal Frequency Discretizer, Class Attribute Contingency Coefficient and Chi2 discretization techniques.

Table 2: Comparison of discretization techniques

| Characteristics | Equal Distance Discretizer | Equal Frequency Discretizer | Class Attribute Contingency Coefficient | Chi2 |
|---|---|---|---|---|
| **Supervised/ Unsupervised** | Unsupervised | Unsupervised | Supervised | Supervised |
| **Top Down/ Bottom Up** | Top Down | Top Down | Top Down | Bottom Up |
| **Parametric** | Yes | Yes | No | Yes |
| **Incremental** | No | No | Yes | Yes |
| **Static/ Dynamic** | Static | Static | Static | Static |

Step 4: Clustering is then applied to the existing data set. The clustering technique used is DBSCAN, as the analysis done in previous papers showed that DBSCAN has the best complexity.
Step 5: Classification method is applied on the clustered data set. The classification technique used is bagging. Bagging is used since it was best suited for yield prediction as shown in previous papers.

## 4 EXISTING COMPARATIVE ANALYSIS

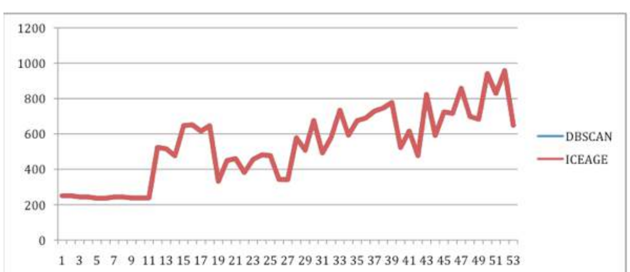



Figure 3: Comparison of DBSCAN and ICEAGE complexity

From previous papers we conclude that DBSCAN [8] and ICEAGE [9] have the best time complexity.

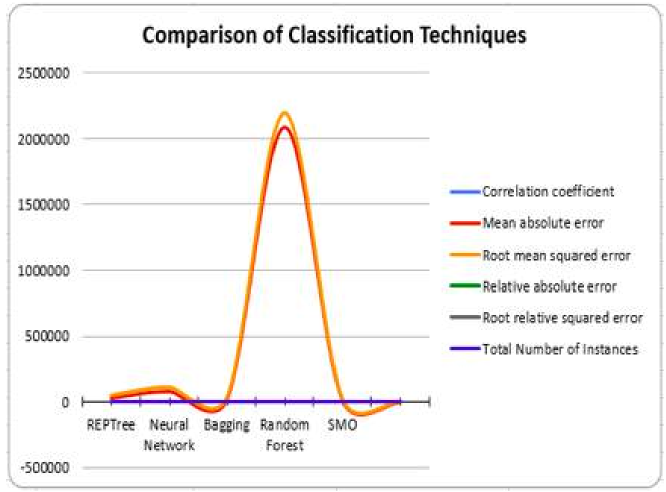



Figure 4 : Graphical view of accuracy parameters

From previous papers we conclude that bagging [10] is the most efficient algorithm among REPTree [11], Neural Network [12], Random Forest [13] and SVM [14] since it has the least error deviation.

## 5 FUTURE WORK

In the future we plan to work on doing comparative analysis of various different discretization techniques and choose the best fit for the proposed model.

## 6 CONCLUSION

In this paper we studied and described a framework that can help us analyze and understand the yield of the crop for a designated zone based on the density of attributes. In future work we would fully develop a DSS that could provide decisions on the type of crop for each zone based on NPK parameters.

## 7 REFERENCES

1. Bongiovanni, Rodolfo, and Jess Lowenberg-DeBoer.: Precision agriculture and sustainability. Precision Agriculture 5.4 (2004): 359-387.
2. Stalikas, Constantine D., George A. Pilidis, and Stella M. Tzouwara-Karayanni.: Use of a sequential extraction scheme with data normalisation to assess the metal distribution in agricultural soils irrigated by lake water. Science of the total environment 236.1 (1999): 7-18.

3. Lim, Meng-Hui, and Andrew Beng Jin Teoh.: An analytic performance estimation framework for multibit biometric discretization based on equal-probable quantization and linearly separable subcode encoding. Information Forensics and Security, IEEE Transactions on 7.4 (2012): 1242-1254.
4. R. Chaves, J. Ramirez and J.M. Gorriz,: Integrating discretization and association rule-based classification for Alzheimer’s disease diagnosis. Expert Systems with Applications, pp. 1571–1578, 2013
5. Garcia, Salvador.: A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning." Knowledge and Data Engineering, IEEE Transactions on 25.4 (2013): 734-750.
6. Su, Chao-Ton, and Jyh-Hwa Hsu.: An extended chi2 algorithm for discretization of real value attributes. Knowledge and Data Engineering, IEEE Transactions on 17.3 (2005): 437-441.
7. MacQueen, James.: Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.
8. Zhou, Aoying,: Approaches for scaling DBSCAN algorithm to large spatial databases. Journal of computer science and technology 15.6 (2000): 509-526.
9. Guo, Diansheng, Donna J. Peuquet, and Mark Gahegan.: ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata. GeoInformatica 7.3 (2003): 229-253.
10. Bauer, Eric, and Ron Kohavi.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning 36.1-2 (1999): 105-139.
11. Mohamed, W. Nor Haizan W., Mohd Najib Mohd Salleh, and Abdul Halim Omar.: A comparative study of reduced error pruning method in decision tree algorithms. Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on. IEEE, 2012.
12. Gorman, R. Paul, and Terrence J. Sejnowski.: Analysis of hidden units in a layered network trained to classify sonar targets. Neural networks 1.1 (1988): 75-89.
13. Gislason, Pall Oskar, Jon Atli Benediktsson, and Johannes R. Sveinsson.: Random forests for land cover classification. Pattern Recognition Letters 27.4 (2006): 294-300.
14. Vishwanathan, S. V. M., and M. Narasimha Murty.: SSVM: a simple SVM algorithm. Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on. Vol. 3. IEEE, 2002.
15. Liu, Huan, and Rudy Setiono.: Chi2: Feature selection and discretization of numeric attributes. 2012 IEEE 24th International Conference on Tools with Artificial Intelligence. IEEE Computer Society, 1995.

# Modelling the Performance of Class-Based Weighted Fair Queue Using OPNET

Idris Zakariyya[1], M Nordin A Rahman[2], Mohd Nazri Ismail[3]

[1, 2] Faculty of Informatics and Computing, University Sultan Zainal Abidin, Besut, Terengganu, Malaysia.
[3]National Defence University of Malaysia
[1] idris_zakariyya@yahoo.com [2] mohdnabd@unisza.edu.my [3] m.nazri@upnm.edu.my

## ABSTRACT

Congestion control and internet network resources management are complex and critical issues in a high-speed packet switch data network, due to the emergence growth of modern multimedia streaming services. The increasing number of computer users in various organizations and institutions of higher learning have spurred a great deal of research on network traffic control. Network administrators are facing the challenges of providing efficient services that can satisfy user requirements. This research study proposed queuing algorithm based on class-based weighted fair queue scheme to complement congestion. Network simulation environment are designed and modeled using OPNET simulation software in-order to overcome the limitation of the traditional queuing approach. Various simulations scenarios are conducted. Analysis comparison with first-in-first-out and priority queue is recorded. And also, various network traffics such as: HTTP, video conferencing and voice applications among others are considered. From the graphical results obtained clearly shows that the overall applications services performances optimize significantly. In terms of the throughput, packet loss and queuing delay, the algorithms performs excellently compared with the FIFO and priority queue. This paper examines the implication of queuing scheduling algorithms on an IP router. It also outlines the effectiveness of the proposed algorithm in managing network resources during the period of congestion.

## KEYWORDS

Congestion Control, Scheduling Algorithm, Simulation Model, Traffic Control, OPNET Simulator.

## 1 INTRODUCTION

Internet network traffic growth rapidly, due to the introduction and development of modern multimedia application and streaming services. Efficient services performances are highly demanded for both industries and universities to satisfy internet user requirements. Memory management implemented on a buffer tried to avoid packets congestion. The, TCP congestion avoidance mechanism is not sufficient enough to provide better services in all circumstances. It uses the best effort model that only attempts to provide the required service without reliability. Admittedly, packets scheduling algorithms managed per flow bandwidth allocation for a fair and an efficient sharing of the network resources among various traffic classes and support quality of service. During packets transmission, queuing scheduling algorithm provides the minimal applications bandwidth guaranteed.

Network resources are highly demanded among various traffic classes. Advertently, real time applications consume much of the link bandwidth and buffer and starved non real time counterpart. Saturation of bandwidth and buffer degrades network performance. There is a need of a queuing scheduling algorithm in-order to complement congestion and enforce fairness in sharing of network resources. However, routers and switches use the traditional scheduling discipline to route traffic to the destination. In this study, class-based weighted fair queue algorithm is proposed and modeled to provide significant internet performance improvement.

Simulation-based study required a powerful network simulator for modeling the network environments and verification. According to [2] research study stated that simulation prevent over-utilization of resources and optimizes system performance. Among the simulation software available in the markets, OPNET modeler is the powerful application software oriented simulator that modeled all aspects of real time and distributed networks. It can serve as teaching tool in schools and has been widely used in modeling the abstracting behavior of real time network. It also involves in the design and development of various aspect of wireless network and data communications.

Various researches using simulation modeling has been done with OPNET [8], [9] [10]. The research study done by [8], constructed real enterprises network prototype and virtual OPNET simulation model. Also, explore the practical application of the models in university education. In [9] paper, proposed a bandwidth guarantee scheme using a class-based weighted fair queue scheduling algorithm implemented on a router. The study presented clearly that traffic classes met their minimum bandwidth requirements. Similarly, according to [10], study on modeling the performance of SOHO network. An investigation was carried out, and analysis results revealed that links data rate upgrades is not economically feasible. Consequently, there is a need of examining the system under study.

Rapid growth in the computer applications development added the complexity in network configuration and quality of service support. Limited network resources leads to the saturations of the buffer and bandwidth scarcity among traffic class. Dynamic internet access by different users within organizations make congestion controls an outstanding issue of consideration. Resources management and congestion control still remain an important area of study in data communication and distributed system.

According to [1], congestion control in packet switching networks, become a high priority in network design and researches due to ever-growing networks bandwidth and intensive applications. Thus, there is growing literature in the area, [3], [4], [5], [6], [7]. Admittedly, [6] of these have shown that congestion can be controlled at gateway through routing and queuing scheduling algorithm. The research study by [3], mentioned that priority of packets scheduling are based on the conditions that the delay sensitive real time packets as the highest priority and packets waiting for a specified time as the lowest priority. Also [5] research study, proposed a link-based fair aggregation technique for queuing scheduling at both the ingress and egress routers. In the [4] paper, simulated an analytical call admission control algorithm in heterogeneous wireless network. In the research study by [7], detailed the inaccuracy of the traditional traffic model like the Poisson’s model, under the bursty real network traffic condition. Hence, performance analysis based on these models can lead to a severe underestimation of packet delay or loss that can affect performance. As a result of that, the study introduces a new traffic model using OPNET Modeler based on hierarchical scheme of Bernoulli sources

Dynamic changes in networks and continuous growth in internet applications development are making congestion control problems a critical issue. However, these potential effects have to be observed and address. The main causes of congestion control in packet-switching network are saturation of network resources like: communication links, buffers and processors cycles. Adverse effects of congestion degrade performance.

In-order to overcome the immediate effects of congestion, we proposed a queue scheduling algorithm based on class-based weighted fair queue scheme. Discrete event simulation model environment are designed and developed. The proposed class-based weighted fair queue algorithm is implemented in a router.

Simulation scenarios results are recorded and analysis comparison is presented. Considering queuing delay, packet loss, traffic sent and received. Our proposed class-based weighted fair queuing algorithm shows performance improvements in comparison with the other traditional approach. In terms of the quality of service measures, the algorithm attained the highest level of expected throughput with low queuing delay. Consequently, the information presented in this paper examined the impacts of queuing scheduling algorithm in router. Also address the critical challenges faces by network administrator in sharing of network resources.

## 2 QUEUING ALGORITHMS

Emergence growth in high-speed packet switch network initiates the need of packet scheduling algorithms. Network is said to be congested when a link output-buffer cannot accommodate incoming packet. This effects, generates a long delay in a queue during transmission. Memory management aims to avoid network congestion. Meanwhile, packet scheduling algorithms monitors and control congestion. According to the research by [11], proposed packet queuing algorithm and also examines its implication in wireless network. Simulation results revealed significant performance improvement.

This section discusses some of the traditional queuing scheduling algorithm in-order to show the effectiveness of our proposed algorithm. Also, the details explanation of the algorithm is presented.

### 2.1 First-In-First-Out

The concept behind first-in-first-out is based on first come first serve mechanism. Packets are served in the queue buffer based on their arrival time. Due to the priority given to the first packet on arrival, other traffic classes waiting for service gets dropped instantly. In addition, a flow of traffic can be starved by other flows that have a higher traffic, as clearly illustrated in figure 1 below.

It is easy to implement and no added delay from the queues. Also, does not always make the best use of bandwidth. It is the simplest queuing algorithm implemented in most of the network routers and switches.

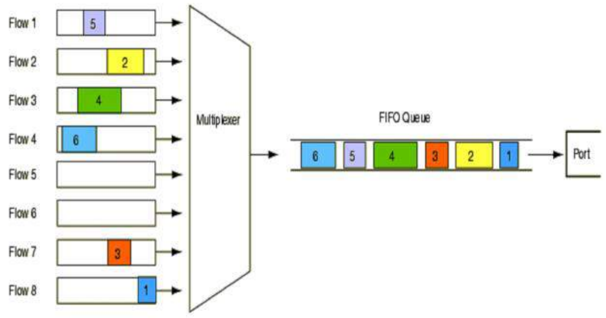



**Figure 1.** First-in-First-Out Queue

### 2.2 Class-Based Weighted Fair Queue

This packet scheduling discipline guarantees a minimal bandwidth to each class of service in a network. It uses the scheduling techniques of weighted fair queuing (WFQ), for assigning weight to different class of service. Priority is given to certain classes of traffic in the router. This scheme enforces fairness at which specific bandwidth is reserved in case if some classes have utilized their share. Further transmission can be done using the available bandwidth. In [9] study, details of bandwidth allocation, flow weight assignation and calculation of packets virtual finish time are presented. As shown in figure 2 below, is the classification and scheduling of an incoming packet to an output ports.

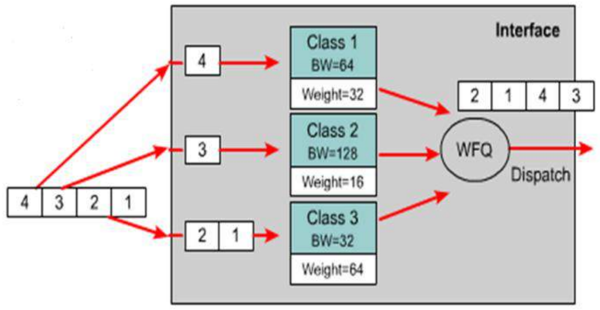



**Figure 2**. Class-Based Weighted Fair Queue [9]

### 2.3 General Processor Sharing

In this scheduling technique, packets in a queue are served using a flow timestamp approach of a processor cycle [9]. Early arrival packets have higher priority in service than late arrival ones. This queuing scheme works best when combined with the priority queue algorithm to allocate internet bandwidth. It is the basis of which the class-based weighted fair queue operates.

### 2.4 Priority Queuing

In priority queuing scheme, flows are queued using the following categories: high, medium, normal, and low. And within each priority, packets are managed in a FIFO manner. During transmission, higher priority packets are allows to cut to the front of the queue. It is the basis for a class of queue scheduling algorithms used to provide the quality of service support for real time application. As shown in figure 3 below is the illustration diagram indicating an incoming flow of traffic from the ingress port interface to the different classes of queue for transmission.

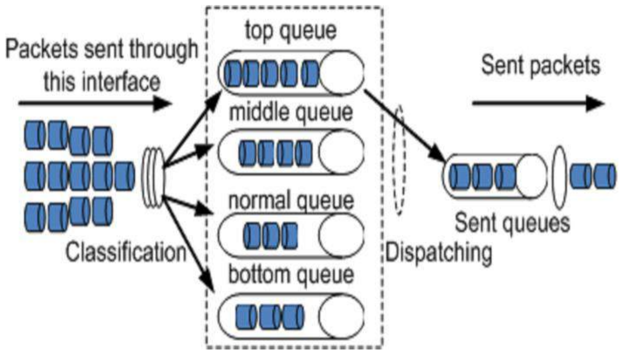



**Figure 3**. Priority Queue

## 3 Experimental Descriptions and Simulation

The simple Ethernet network is designed and constructed during the modeling of the class-based weighted fair queue algorithm, first-in-first-out and priority queue router. The network topological model illustrated in figure 4 below consists of four clients sending traffics to their servers via switches and routers.

In addition, all the nodes in the network are connected using 10 Mbps data link speed, with the exception of the potential bottleneck link between the routers A and B. Clients transmits HTTP, FTP, video and voice traffic to the HTTP & FTP server, voice and video server clients respectively.

Each client is properly configured to provide the required services for each traffic classes using the application configuration object. The application profile attribute is configured to supports the services of the applications. In the model, servers are configured to supports various applications services. The potential bottleneck links between routers has been configured globally using the IP QoS configuration to supports the routing of traffic classes in the network. During simulation discrete event statistic is collected and various scenario graphs are generated for analysis.

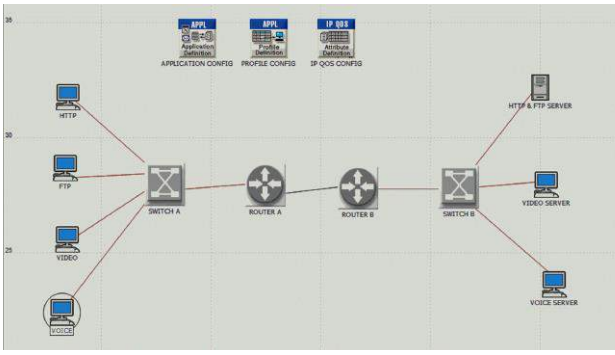



**Figure 4.** Simulation network model

### 3.1 Network Configuration

#### 3.1.1 Configuring Network and Applications

The HTTP application is configured to support the excellent effort web services model using the application configuration attribute object. It page inter-arrival time is set to exponential 60 and constant 10 as the start time offset. Figure 5 below gives the details of the configuration process.

**(Http) Table**

| Attribute | Value |
|---|---|
| HTTP Specification | HTTP 1.1 |
| Page Interarrival Time (seconds) | exponential (60) |
| Page Properties | (...) |
| Server Selection | (...) |
| RSVP Parameters | None |
| Type of Service | Excellent Effort (3) |

Details | Promote | OK | Cancel

**Figure 5.** HTTP Application Configured

The file transfer protocol (FTP) is configured to supports the heavy load file application up-to 5MB of size at an inter-request time of constant 10. It also supports the best effort file services delivery for meeting the expected quality of service. As shown in the figures 6 and 7 below is the outline of the configuration.

**(Ftp) Table**

| Attribute | Value |
|---|---|
| Command Mix (Get/Total) | 50% |
| Inter-Request Time (seconds) | constant (10) |
| File Size (bytes) | constant (5000000) |
| Symbolic Server Name | FTP Server |
| Type of Service | Best Effort (0) |
| RSVP Parameters | None |
| Back-End Custom Application | Not Used |

Details | Promote | OK | Cancel

**Figure 6.** FTP Application Configured

**(Applications) Table**

| | Name | Start Time Offset (seconds) | Duration (seconds) | Repeatability |
|---|---|---|---|---|
| File | File | constant (10) | End of Profile | Once at Start Time |

**Figure 7.** FTP Configuration Table

Additionally, video conferencing application is configured as low resolution quality video, with 10 frames interval per seconds and frame size of 128X120 pixels as shown in figures 8 and 9 below. It provides support for streaming multimedia application services.

**(Video Conferencing) Table**

| Attribute | Value |
|---|---|
| Frame Interarrival Time Information | 10 frames/sec |
| Frame Size Information (bytes) | 128X120 pixels |
| Symbolic Destination Name | Video Destination |
| Type of Service | Streaming Multimedia (4) |
| RSVP Parameters | None |
| Traffic Mix (%) | All Discrete |

Details | Promote | OK | Cancel

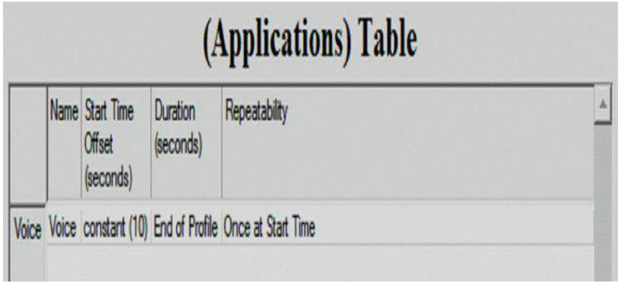
**Figure 8.** Video Conferencing Application Configured

**(Applications) Table**

| | Name | Start Time Offset (seconds) | Duration (seconds) | Repeatability |
|---|---|---|---|---|
| Video | Video | constant (10) | End of Profile | Once at Start Time |

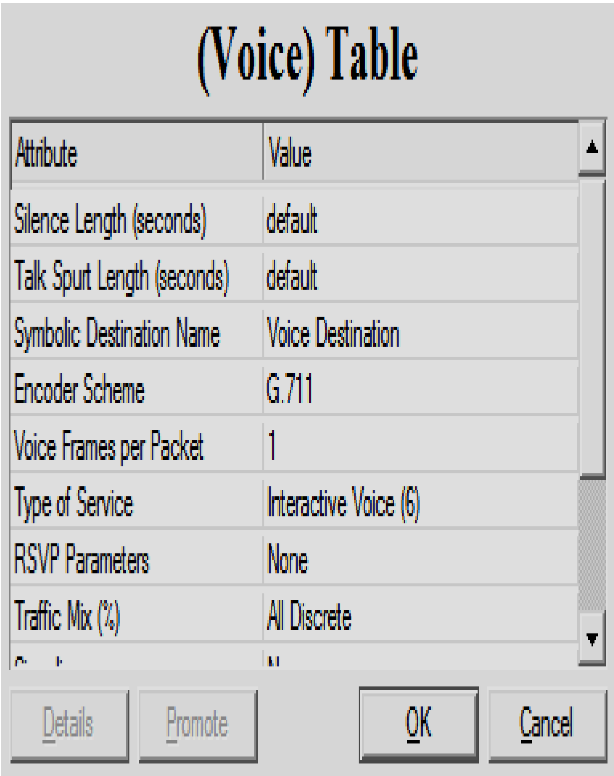
**Figure 9.** Video Conferencing Application Table

The profile configuration for voice application is shown clearly in the figures 10 and 11 below. In the configuration, voice have assigned pulse code modulation (PCM) quality speech and supported the interactive voice services.


(Applications) Table

| | Name | Start Time Offset (seconds) | Duration (seconds) | Repeatability |
|---|---|---|---|---|
| Voice | Voice | constant (10) | End of Profile | Once at Start Time |

**Figure 10**. Voice Configuration Table


(Voice) Table

| Attribute | Value |
|---|---|
| Silence Length (seconds) | default |
| Talk Spurt Length (seconds) | default |
| Symbolic Destination Name | Voice Destination |
| Encoder Scheme | G.711 |
| Voice Frames per Packet | 1 |
| Type of Service | Interactive Voice (6) |
| RSVP Parameters | None |
| Traffic Mix (%) | All Discrete |



**Figure 11.** Voice Application Configured

## 4. SIMULATION RESULTS

Various simulations scenarios are analyzed under the constructed discrete event simulation environment. The graphical scenarios results have been taken and also examined. Moreover, significant performance improvements are recorded.

In terms of throughput, delay and packets loss, our proposed algorithm performs efficiently over the selected traditional scheduling algorithm. And also the available network resources are utilized effectively. In the packet dropped graph shown in the figure 12 below, indicated clearly that the algorithm has the least probability of dropping packets in the network. As represented by the blue line, the number of packets drop in a second is uniform throughout the simulation in comparison with red and green line for the first-in-first-out and priority queue. This gives an outstanding detail about the efficient performance of our proposed scheme during successful packet transmission in a network.

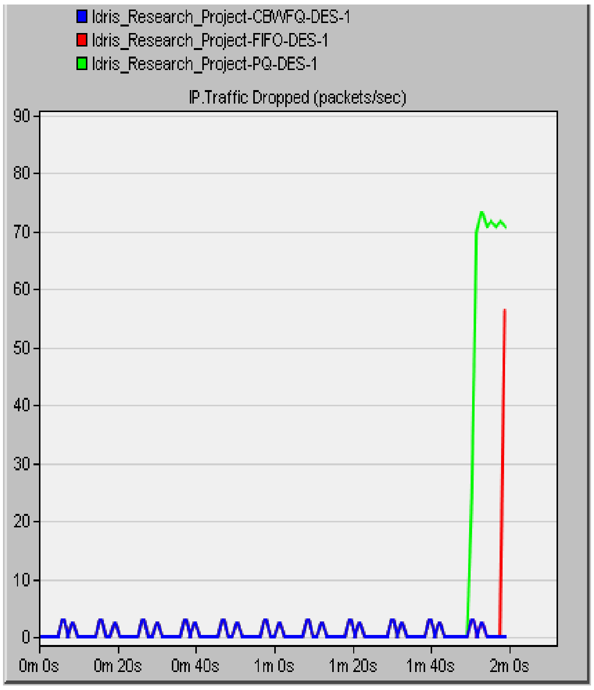



**Figure 12.** IP Traffic dropped in packet/second

As depicted in the figures 13 and 14 below is the packets end to end delay in second for video and voice traffic. On arrival packet have experienced the maximum queuing delay when encounter a packet receiving service at the head of its queue. From the graphs, CBWFQ packet represented by the blue dotted line has minimal delay compared to PQ and FIFO respectively. Consequently, the algorithm work best in transmitting a packet from source to destination in a network.

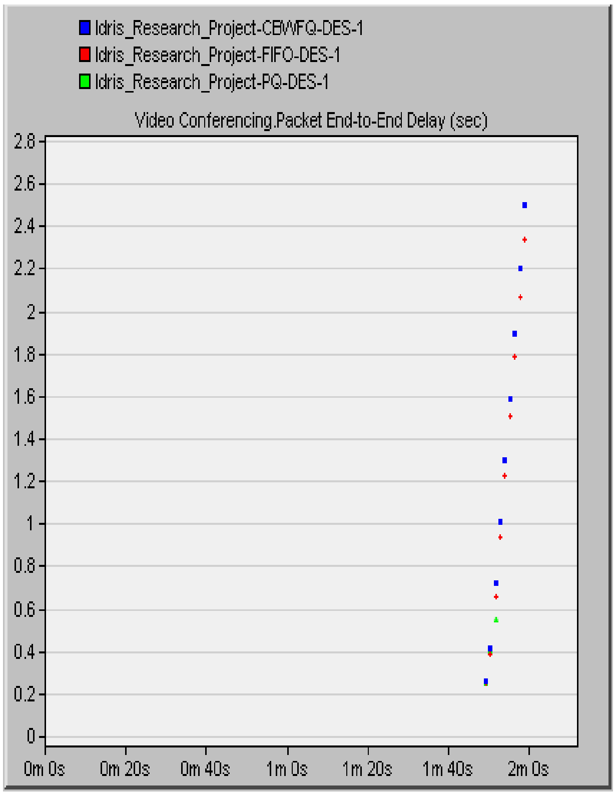



**Figure 13.** Video Conferencing Packets ETE

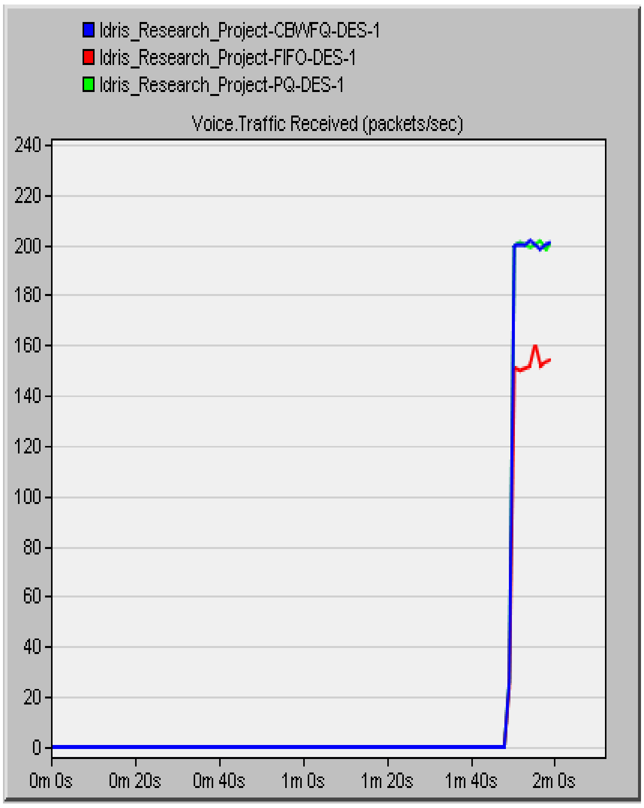



**Figure 15.** Voice Traffic Received

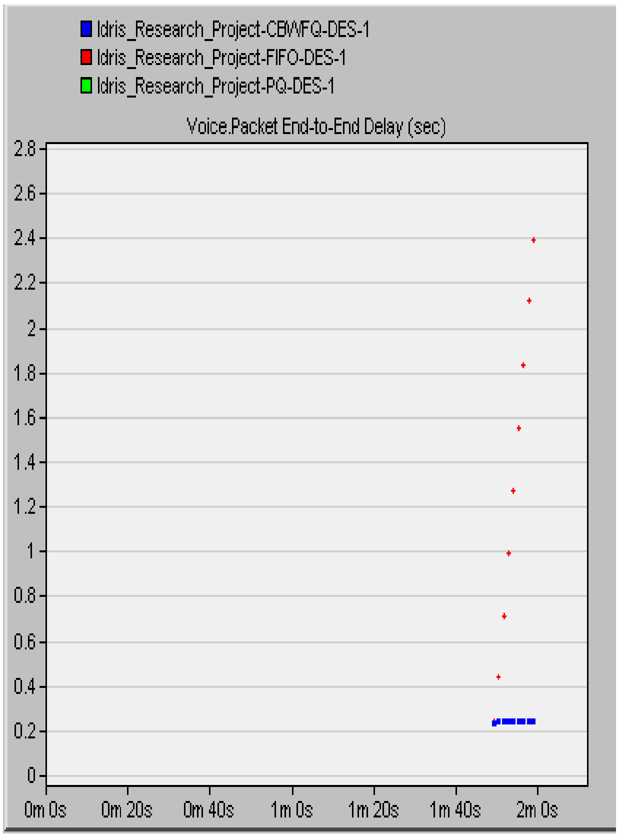



**Figure 14.** Voice Packets ETE Delay (sec)

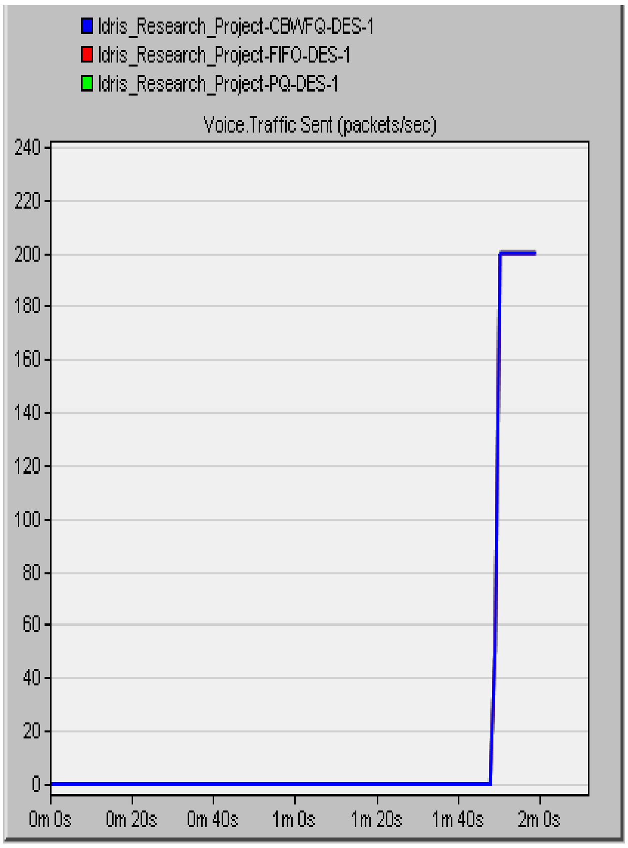



**Figure 16.** Voice Traffic Sent

## 4 CONCLUSION

The detailed implementation of the class-based weighted fair queue with the first-in-first-out and priority queue using OPNET simulator is fully outlines. The fundamental use of the packet scheduling algorithm in the transmission of packet and managing of network resources is examines. Moreover, simulations experiments are conducted and also scenarios are generated. Simulations results are studied and analyzed. The effectiveness of the proposed algorithm in terms of packet loss, throughput and bandwidth utilization is clearly illustrated. And also, the optimal performance improvement attained is recorded, that shows the efficiency of the proposed algorithm over the selected traditional scheduling mechanism. The study contributed immensely in benefitting an organization and industries to manage their internet network resources and provides the efficient services required by the users.

## 5 ACKNOWLEDGEMENTS

The authors thanks the faculty of informatics and computing (UniSZA) for their duly support during the study. Thanks are also extended to the anonymous reviewers and editors who have provided valuable criticism and suggestion for the simulation environment model design.

## 6 REFERENCES

1. Cui-Qing Yang and A.V.S Reddy, "A Taxonomy for Congestion Control algorithms in Packet Switching Networks.pdf,"1995, pp. 34-35.
2. A. Maria, "Introduction to Modeling and Simulation", Proceedings of the 1997 Winter Simulation Conference, 1997.
3. L. Karim, N. Nasser, T. Taleb, and A. Alqallaf, "An efficient priority packet scheduling algorithm for Wireless Sensor Network," 2012 IEEE International Conference on Communications (ICC), pp. 334–338, Jun. 2012.
4. H.S Ramesh Babu, Gowrishankar, P.S Satyanarayana, "Call Admission Control for Next Generation Wireless Networks Using Higher Order Markov Model", Computer and Information Science, www.ccsenet.org/cis, Vol. 3, No. 1, February, 2010.
5. Y. Jiang, "Link-based Fair Aggregation: a Simple Approach to Scalable Support of Per-Flow Service Guarantees", in Tech. Report, Norwegian Inst. of Technology, 2004.
6. A. Damers, S.Keshav and S.Shenker, "Analysis and Simulation of a Fair Queuing Algorithm" Journal of Internetworking Research and Experience, Vol.1, No.1, pp. 3-26, September, 1990.
7. J. Potemans, B. Van den Broeck, Y.Guan, J. Theunis, E. Van Lil, A. Van de Capelle, "Implementation of an Advanced Traffic Model in OPNET Modeler", OPNETWORK 2003, 2003
8. D. Akbas, H. Gumuskaya, "Real and OPNET modeling and analysis of an enterprise network and its security structure", Procedia Computer Science, 3, pp. 1038 – 1042, 2011.
9. I. Zakariyya, M.N.A. Rahman, "Bandwidth Guarantee using Class Based Weighted Fair Queue (CBWFQ) Scheduling Algorithm", International Journal of Digital Information and Wireless Communication (IJDIWC), Vol.5, No.3, pp. 152-157, SDIWC, 2015.
10. G. Hai and J. Bo, "SOHO Network Modeling and Simulation Using OPNET", Journal of Theoretical and Applied Information Technology", www.jaitit.org, Vol. 48, No. 3, 28th February, 2013.
11. A. Mahiddin, R. Hasyifah, W.N.W. Nik, "Performance Comparison of Packet Queuing Algorithm in Wireless Networks" ICECNCCT, Vol. 13, No.2, pp. 18-24. 2014.

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 61-78
The Society of Digital Information and Wireless Communications, 2015 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

# SLA-based Resource Allocation within Cloud Networking Environment

Mohamad T. Hamze, Nader Mbarek and Olivier Togni
Le2i Laboratory UMR 6306 CNRS, University of Burgundy
9 avenue Alain Savary, 21000 Dijon, France
{Mohamad.Hamze, Nader.Mbarek, Olivier.Togni}@u-bourgogne.fr

## ABSTRACT

Today, cloud networking which is the ability to connect users with their cloud services and to interconnect these services in an inter-cloud approach, is one of the recent research areas within the cloud computing research communities. The main drawback of cloud networking consists in the lack of Quality of Service (QoS) guarantee and management in conformance with a corresponding Service Level Agreement. In this paper, a framework is proposed for resource allocation according to an end-to-end service level agreement established between a Cloud Service User and several Cloud Service Providers within a cloud networking environment. The paper focuses on QoS parameters for NaaS and IaaS services. In addition, several algorithms are proposed for the best Cloud Service Providers selection to allocate VMs and network resources within a Broker and a Federation based inter-cloud scenarios. The objective is to minimize the total cost while satisfying NaaS and IaaS QoS constraints. Moreover, the proposed cloud networking architecture is used to provide videoconferencing and intensive computing applications with QoS guarantee. The corresponding scenarios are simulated using an extended CloudSim toolkit. The results reveal a high scalability and good performances where the Broker architecture is the most interesting while ensuring QoS requirements.

## KEYWORDS

Cloud Networking, Inter-cloud, Service Level Agreement, Quality of Service, Optimization, Broker/Federation.

## 1 INTRODUCTION

Cloud computing is a promising technology for the realization of large, scalable and on-demand computing infrastructures. Many enterprises are adopting this technology to achieve high performance and scalability for their applications while maintaining low cost. In addition, a Cloud Service User (CSU) that can be an end-user, an organization, a Software as a Service (SaaS) provider, or a Platform as a Service (PaaS) provider, requires for its services an end-to-end QoS assurance with a high level reliability and a continued availability. Moreover, cloud computing success requires that CSUs and Cloud Service Providers (CSPs) can be confident that established SLAs are supporting their respective business activities to their best extent.

However, a Service Level Agreement (SLA) may be violated when using a single cloud provider model due to an unpredictable workload, resource failure in the data centre [1] and security attack on cloud resources [2]. Thus, inter-cloud systems are unavoidable as it is very difficult for a single CSP to satisfy its customer requirements. Therefore, geographically distributed Data Centers offer better end-to-end performance between CSU and CSP and improve reliability when failure occurs.

Moreover, the inter-cloud should be designed as a multi-vendor environment with the ability to migrate services from one provider to another and to locate the best resources not only in terms of computing capacity and storage, but also connectivity, bandwidth and delay. Thus, the networking aspect of cloud computing is a critical factor for adopting this approach.

In this context, cloud networking is defined as the ability to connect the user to his cloud services and interconnect services within an inter-cloud. It is one of the recent research areas in the cloud computing research communities. However, to achieve the cloud networking model, there are a number of open issues, such as the formality of a language for SLA description between CSUs and CSPs, the interoperability of data formats, and communication using standard interfaces [3]. In addition, there are major challenges on how to establish an SLA in cloud networking and how to select the best CSPs for resource allocation based on the SLA and QoS requirements [4].

Moreover, in order to allow a cost-effective scaling and best performance of videoconference and intensive computing systems, the use of cloud computing resources appears as a natural approach, since they provide pay for use and on demand resources [5]. Therefore, CSPs must provide these services with QoS guarantee and lower cost according to an established SLA.

In this paper, a framework is proposed for resource allocation in conformance with an end-to-end SLA in a cloud networking environment. Two cloud networking architectures are specified, a Broker and a Federation based inter-cloud architectures. Then, the paper describes the SLA establishment and the best CSPs selection using optimization algorithms under QoS constraints while minimizing the cost objective. In addition, the paper describes how videoconferencing and intensive computing applications could take full advantage of the proposed framework.

The remainder of this paper is organized as follows: Section 2 presents a brief overview on the state of the art for cloud computing and networking. Section 3 highlights the most relevant research works and trends in this area. Section 4 describes the proposed architectures and algorithms for resource allocation and establishing an end-to-end SLA in a cloud networking environment. Section 5 presents usage cases and the framework evaluation. Lastly, Section 6 concludes the paper.

## 2 STATE OF THE ART

### 2.1 Cloud Computing

In 2011, the NIST [6] defined cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This model consists of five characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. It contains four deployment models: private cloud, community cloud, public cloud, and hybrid cloud. It consists of three service models: SaaS, PaaS, and Infrastructure as a Service (IaaS).

IaaS consists of a large number of computing and storage resources that are connected with each other through the Data Center network. It is completely abstracted from the hardware below and allows users to use the infrastructure as a service without worrying about the underlying complexities. In addition, most IaaS cloud providers offer a set of general-purpose Virtual Machine (VM) classes with generic resource configurations. For example, Amazon EC2 [7] supports 11 types of VMs, each one with different characteristics.

### 2.2 Cloud Networking

The term cloud networking is introduced in a multi-administrative domain scenario, where network and Data Center domains must interact through defined interfaces to provide a service to CSU [8]. Cloud networking extends network virtualization beyond the Data Center to bring two new aspects to cloud computing, connecting the user to services in the cloud and interconnecting services that are geographically distributed across cloud infrastructures. Hence, there are three kinds of networks in cloud environment [9]:

1) Intra-cloud network: this network connects local cloud infrastructures. It concerns network resources between servers and storage systems.

2) Core transport network (WAN/MAN): this kind of network can be used by customers to access and consume cloud services deployed within the cloud provider's Data Centers.

3) Inter-cloud network: it can be used to interconnect cloud infrastructures together. These cloud infrastructures may be owned by the same cloud provider or by different ones.

Cloud networking services that provide network connectivity in the three kinds of networks described above are named Network as a Service (NaaS) [33]. It is difficult to guarantee the QoS of data transfer in cloud networking. The most convenient solution is to differentiate the treatment of different kinds of information while using enough bandwidth in such environment. This bandwidth is generally provided by a network operator. In addition, it is offered on demand by NaaS services. Bandwidth on Demand (BoD) is the ability to perform on-demand changes (increase, decrease) and instant provisioning of bandwidth on particular links via standardized interfaces. The NaaS BoD QoS guarantee could be provided thanks to the establishment of an end-to-end SLA between CSUs and CSPs [9]. It provides a cost effective solution for QoS guarantee.

### 2.3 Inter-Cloud

Inter-cloud computing allows on-demand assignment of cloud resources, including computing, storage and network, and the transfer of workload through interworking of cloud systems [9]. The communication between different clouds is established through the cloud networking that can provide a NaaS (BoD) service. CSPs can interwork in different manners [9]:
1) Inter-cloud peering (P2P): it is a direct inter-connection between two CSPs using already-established interfaces.
2) Inter-cloud Service Broker (ISB): it provides an indirect interconnection between two or more CSPs and Brokering services to CSUs or CSPs. It manages the performance, security and delivery of cloud services.
3) Inter-cloud Federation: it provides an alliance among several CSPs in which mutually trusted clouds join together logically through agreed-upon interfaces. It allows CSPs to search and reserve available resources in other CSPs, based on different service levels, in order to avoid SLA violations.

## 3 RELATED WORKS

From standardization perspective, IEEE Cloud Computing [10] formed the Inter-Cloud Working Group (ICWG). It announced the launch of two new standards development projects in April 2011: P2301 [11], a guide for Cloud Portability and Interoperability Profiles (CPIP) and P2302 [12], a Standard for Intercloud Interoperability and Federation (SIIF). Open Grid Forum (OGF) is active in the definition of the Open Cloud Computing Interface (OCCI) [13] for the interoperability between clouds. Global Inter-Cloud Technology Forum (GICTF) [14] studies the standard Inter-Cloud interfaces to improve the reliability of the Clouds, and presents SLA metrics for Inter-Cloud environments. IBM presented in 2011 CloudNaaS [15], a cloud networking platform for enterprise applications. However, the proposed research work develops a cloud networking framework and aims to enable communications between, not only CSPs Data Centers (DC), but also CSU, CSPs (DC) and CSPs (BoD). In addition, this communication is through agreed-upon interfaces. For that purpose, Web Services standard technologies will be used.
Furthermore, there are many related research works concerning QoS within cloud environments [16–18], but the QoS mentioned in these works concerns SaaS, PaaS or IaaS and do not consider NaaS services. Moreover, most of the current CSPs are limited to the resources availability guarantee and do not take into account many other important QoS parameters such as latency and bandwidth [19]. Yanzhi et al. [20] consider the problem of resource allocation and the corresponding cost under an established SLA. However, they do not consider either the inter-cloud approach or the NaaS QoS parameters.
In addition, several research projects present SLA for cloud computing without considering the cloud networking aspect: the project Mycloud [3] proposes Cloud Service Level Agreement (CSLA) and Patel et al. [21] propose to use Web Service Level Agreement (WSLA) [22] in a cloud computing. The research project Scalable & Adaptive Internet Solutions (SAIL) [23] describes a cloud networking architecture and focuses on security, but it does not consider the QoS guarantee and the corresponding SLA. Faniyi et al. [24] present the design of a cloud architecture enabling the coordination of cloud Federation entities to meet cloud users' SLA terms. However, they do not consider either the Broker case or the NaaS QoS parameters. Finally, several recent research works on cloud multimedia streaming services [25–29] consider only QoS parameters without establishing the corresponding SLA.

Therefore, this paper considers IaaS and NaaS services, QoS attributes and an SLA for cloud networking environment based on inter-Cloud Broker and Federation architecture.

## 4 PROPOSED CLOUD NETWORKING ARCHITECTURE

In order to deliver the service level expected by CSU, the proper design and construction of a cloud networking architecture is very critical. Therefore, two architectures are designed, the first one is based on inter-Cloud Broker (Fig. 1) and the second is based on inter-cloud Federation (Fig. 6). An environment with multiple CSPs interconnected together is assumed. Within these cloud networking architectures, consistency must be ensured between the QoS requirements requested by the CSU, and the SLA proposed by CSPs to allow multiple CSPs collaboration to meet the CSU requirements. These architectures will be validated for two different services: cloud videoconference and intensive computing as described in Section 5.

### 4.1 Cloud Networking Broker Architecture

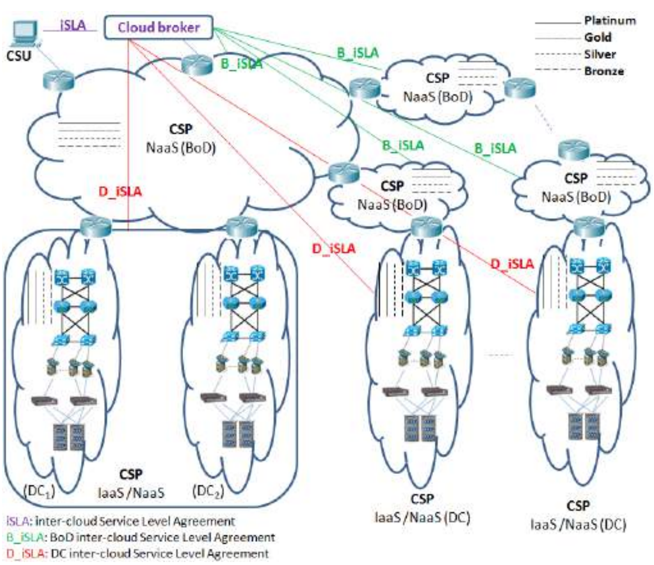



**Figure 1.** Cloud Networking Broker Architecture.

In the first proposed architecture (Fig. 1), the Cloud Broker is emerged as an intermediate entity between a CSU and CSPs to help the establishing of a service level that meets the CSU requirements and to simplify secure integration and lower prices for CSU. In addition, it allows organizations to consume a portfolio of cloud services which are well integrated and offered under a single global SLA and unified billing. Two kinds of CSPs are specified, the CSP (BoD) providing NaaS (BoD) service and the CSP (DC) providing IaaS (Virtual Machine and storage resources) and NaaS (network Data Center resources) services. The CSP (DC) can offer resources from one or several DCs. Moreover, CSPs can offer different service levels (Platinum, Gold, Silver, and Bronze) with different QoS parameters guarantee and cost.

#### 4.1.1 SLA Specification and Description

In the proposed cloud networking Broker architecture, a Service Level Agreement (SLA) is a contract between a CSU and a Cloud Broker or between a Cloud Broker and a CSP. It contains different QoS guarantees offered by CSP and required by CSU. The SLA can be structured by: SLA Identifier, Validity Period, Obliged Parties, Service Identification, Guarantees offered on the Services, and Business Attributes (cost, violation and penalties).

Furthermore, based on “Pay as you go” cloud characteristic, the proposed SLA must identify service level guarantees for all QoS requirements in a cloud networking environment and the CSU pays as it uses. Therefore, in the cloud networking Broker architecture three types of SLA are proposed and constructed using the XML language for interoperability and portability:

a) inter-cloud Service Level Agreement (iSLA): it is a contract between a CSU and a Cloud Broker (ISB: Inter-cloud Service Broker). It guarantees QoS for NaaS (BoD and/or DC) services and/or QoS for IaaS (VMs and/or storage) services. The most important attributes of an iSLA are shown in Figure 2. Business attributes define the unit cost of VMs, storage and bandwidth resources then the CSU is charged based on its use. The monitoring interval time is used to calculate the penalties in case of violation. In addition, Figure 3 specifies the IaaS and NaaS QoS parameters corresponding to the Service Performance Guarantees attribute within the iSLA. Each parameter can be quantitative or qualitative.

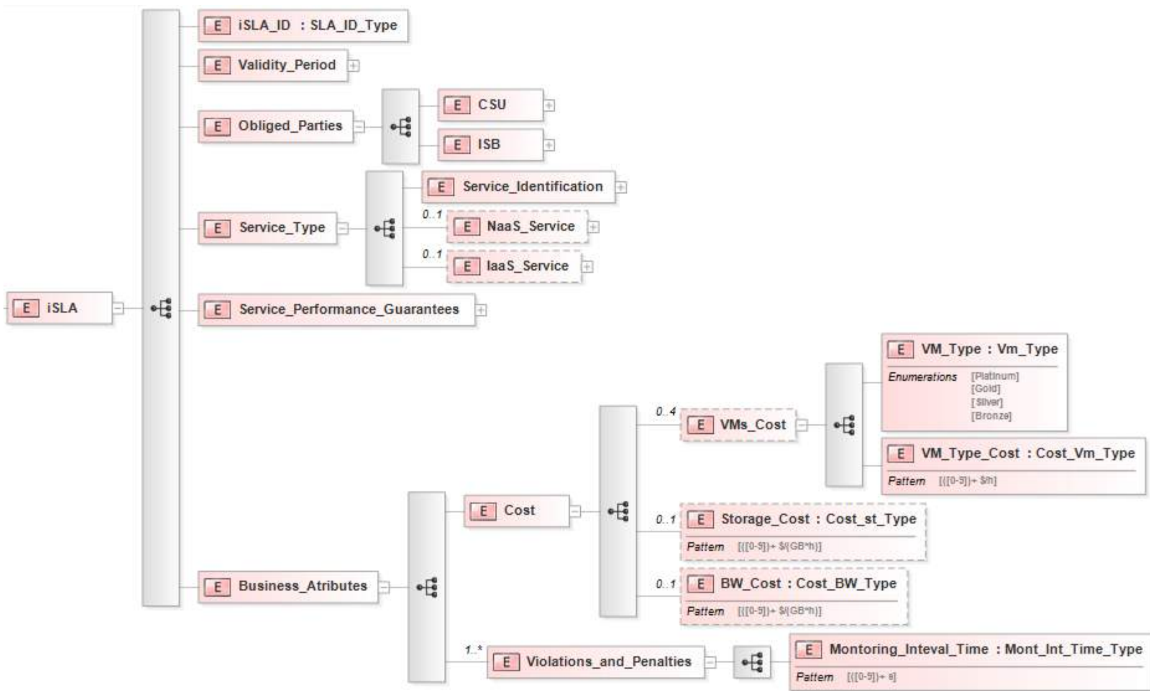



**Figure 2.** iSLA XML Schema (XSD representation).

```
<xs:element name="Service_Performance_Guarantees">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="NaaS_QoS_parameters" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Latency" minOccurs="0" maxOccurs="1" >
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Qualitative_Value" minOccurs="0" maxOccurs="1" type="QoS_Qualitative_Type"/>
                                    <xs:element name="Quantitative_Value" minOccurs="0" maxOccurs="1" type="Lat_Type"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Jitter" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                        <xs:element name="Packet_Loss_Ratio" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                        <xs:element name="Bandwidth" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                        <xs:element name="Availability" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="IaaS_QoS_parameters" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Respnse_Time" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                        <xs:element name="Availability" minOccurs="0" maxOccurs="1">
                            ...
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

**Figure 3.** IaaS and NaaS performance guarantees attribute XML Schema.

b) BoD inter-cloud Service Level Agreement (B_iSLA): it is a contract between a Cloud Broker and CSP (BoD) interconnecting CSU sites or connecting CSU sites to CSP (DC). Its structure looks like the iSLA structure. However, it guarantees only QoS for NaaS (BoD) services. Furthermore, service performance guarantees attribute contains only quantitative NaaS QoS parameters and business attributes contain only the bandwidth unit cost.

c) Datacenter inter-cloud Service Level Agreement (D_iSLA): it is a contract between a Cloud Broker (ISB) and a CSP (DC) for NaaS and/or IaaS services. Its structure looks like the iSLA structure. However, IaaS and NaaS QoS parameters in service performance guarantees are quantitative.

4.1.2 Interaction between Entities

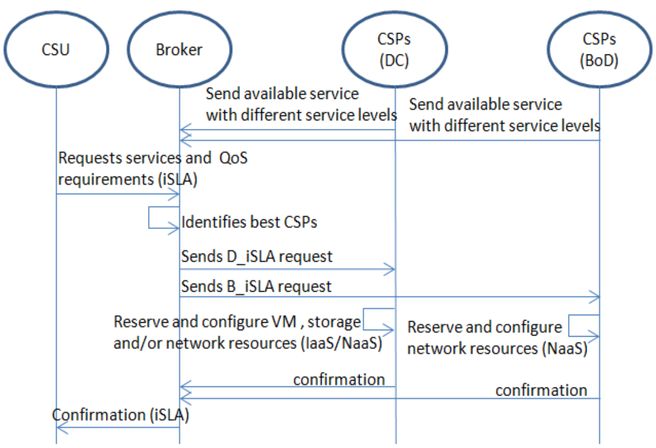



**Figure 4.** MSC for Inter-Cloud Broker.

To achieve an end-to-end SLA, several interactions exist between CSU, Cloud Broker and CSPs (Fig. 4). At first, CSPs (DC/BoD) describe IaaS and/or NaaS services for their available resources with different service levels. Then, CSPs send periodically this information to the Cloud Broker using an XML based Web Service interface for interoperability and portability. However, if important changes occur within these CSPs, they send these changes immediately to the Cloud Broker. In addition, each entity (Broker or CSP) has a repository that contains information about resources and the corresponding service levels. IaaS services consist on VMs and storage capacities with QoS guarantees. Each VM has a specific type (Platinum, Gold, Silver, or Bronze) that defines its characteristics. These characteristics concern hypervisor type, CPU computing capacity, CPU number, RAM capacity, and memory capacity. However, NaaS services consist on the BoD with QoS guarantee.

Another interaction concerns the CSU that uses a proposed Graphical User Interface (GUI) to specify requested services and QoS requirements (Fig. 5). The CSU specifies sites number, their distributions at different areas using IP addresses and their destination sites. Moreover, it provides the Cloud Broker with its application and service type (NaaS and/or IaaS services) as well as the corresponding QoS parameters (quantitative or qualitative) and weights that can be normalized afterward. Furthermore, for IaaS services, it can specify its different job lengths, VMs numbers and/or storage capacity.

Then, the CSU sends these service requirements to the Cloud Broker to construct an iSLA. The Cloud Broker consults its repository and compares the CSU requirements with the different services and corresponding service levels offered by CSPs to select the appropriate CSPs that meet the CSU QoS requirements (see Sec. 4.3). Afterward, the Cloud Broker sends a request to establish a D_iSLA and a B_iSLA with respectively selected CSPs (DC) and CSPs (BoD). The concerned CSPs (DC) reserve and configure VM, storage capacity and network resources to deliver IaaS and NaaS services, and the concerned CSPs (BoD) reserve and configure network resources to deliver NaaS (BoD) service. Then, they update their repository with these changes and send confirmation to Cloud Broker. Finally, the Cloud Broker updates its repository with these changes and establishes the iSLA with the CSU.

### 4.2 Cloud Networking Federation Architecture

In the second proposed architecture (Fig. 6), the federation provides an alliance among several CSPs (DC/BoD) collaborating together to help the establishing of a service level that meets the CSU requirements. In addition, CSPs provide IaaS and/or NaaS services with different service levels.

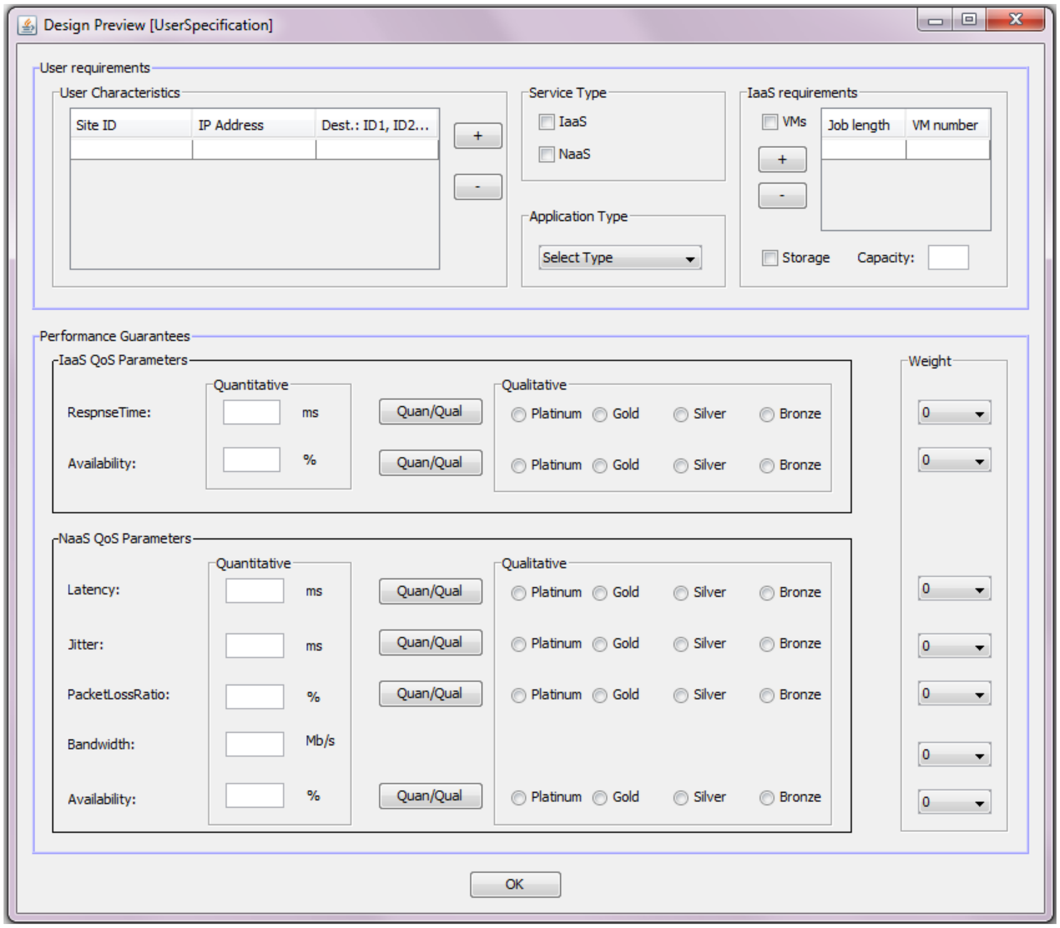



**Figure 5.** GUI for CSU Preferences.

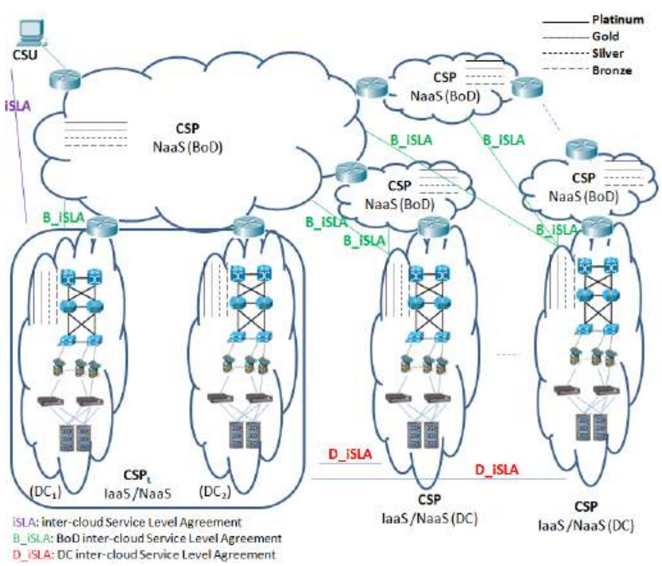



**Figure 6.** Cloud Networking Federation Architecture.

NaaS services concern network Data Center resources that connect local IaaS resources or concern BoD resources. IaaS services concern VMs and storage resources. Note that the CSU is associated to a Lead CSP (DC) ($CSP_L$).

4.2.1 SLA specification and description

In the cloud networking Federation architecture, the same three types of SLA are proposed and the corresponding XML schema representation. However, the iSLA is a contract between CSU and $CSP_L$, the D_iSLA is a contract between $CSP_L$ and CSP (DC), and the B_iSLA is a contract between CSP (DC) and CSP (BoD) that enable CSU sites to reach the considered CSP (DC).

4.2.2 Interaction between entities

The interaction between CSU, $CSP_L$ and CSPs is described in Figure 7. At first, the CSU sends service requirements to the $CSP_L$ to construct an iSLA while specifying a service level that contains QoS parameters for IaaS with/without NaaS services using the proposed GUI (Fig. 5). The $CSP_L$ contacts CSPs (BoD) that enable CSU sites to reach it, in order to receive their available resources for NaaS (BoD) services with different service levels. Then, it selects the best CSPs (BoD) that enable CSU sites to reach it (see Sec. 4.3.2). In addition, it consults its repository to identify its available resources with their different service levels. As a consequence the following scenarios are identified:

a) Scenario 1: in this scenario, the $CSP_L$ has available resources that meet the CSU requirements. Therefore, the $CSP_L$ selects its best IaaS resources (see Sec. 4.3.3) and sends a request to establish a B_iSLA with selected CSPs (BoD). Next, the $CSP_L$ reserves and configures VMs, storage and network resources to deliver IaaS with/without NaaS services.

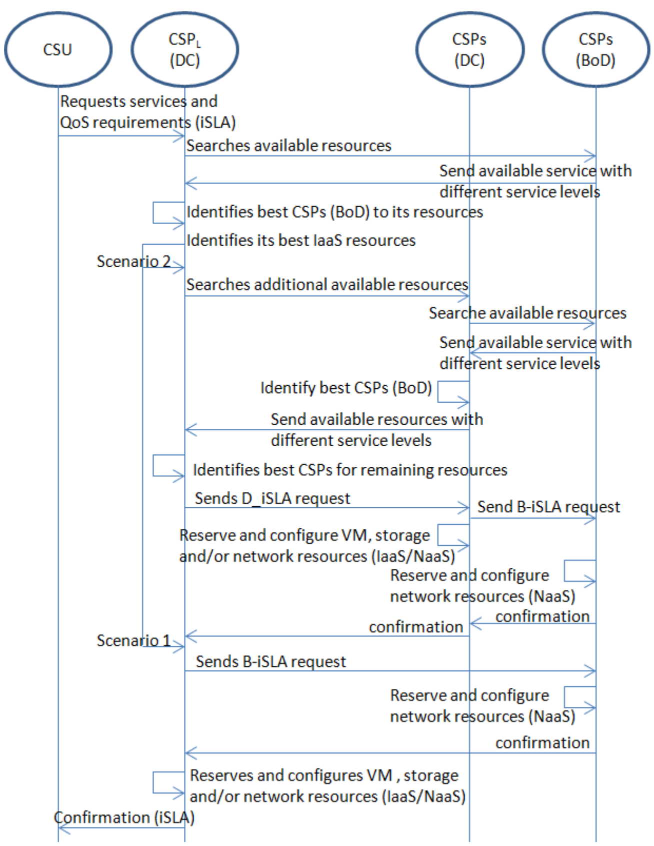



**Figure 7.** MSC for Inter-Cloud Federation Scenarios.

b) Scenario 2: in this scenario, the $CSP_L$ has not sufficient resources. Therefore, it allocates its available resources, calculates the remaining resources and contacts other CSPs (DC) in the federation to get their available resources. In addition, CSPs (DC) contact CSPs (BoD) that enable CSU sites to reach them, in order to request available NaaS (BoD) resources with different service levels. Then, they select the best CSPs (BoD) (see Sec. 4.3.2) and send the corresponding IaaS and NaaS services for available resources with different service levels to the $CSP_L$. The latter selects the best CSPs that meet CSU QoS requirements for IaaS with/without NaaS services (see Sec. 4.3.3). Next, the $CSP_L$ sends a request to establish a D_iSLA with every selected CSP (DC). The $CSP_L$ and every selected CSP (DC) establish a B_iSLA with the best CSPs (BoD) that enable the CSU to reach them. Thus, the $CSP_L$ and the concerned CSPs (DC) reserve and configure VM, storage and/or network resources to deliver IaaS with/without NaaS services. Finally, after these scenarios, the concerned CSPs (BoD) reserve and configure network resources to deliver NaaS (BoD) service with QoS guarantee if requested by the CSU, and the $CSP_L$ establishes the iSLA with the CSU.

### 4.3 CSPs Selection and QoS Splitting

In general, the CSU requests IaaS and/or NaaS services to run specific applications with QoS guarantee for its different sites using Brokerage or Federation services. However, this paper considers that the CSU requests IaaS services with or without NaaS services. Therefore, the following sections present the problem statement and proposed algorithms that ensure the best CSPs (DC) selection offering IaaS resources with the best path selection (CSPs (BoD) and CSP (DC) network) between CSU sites and selected CSPs (DC). Notations are described in Table 1.

**Table 1.** Notations.

| Symbol | Meaning |
|---|---|
| $Q$, $Q_1$, $Q_2$, $Q_3$, $T$, $T_1$, $T_2$ | Temporary sets |
| $T_3[]$ | Table of temporary sets |
| $min_c$, $flag$ | Local variables |
| $L$, $J$, $P$, $BW$, $A$, $RT$ | Latency (ms), Jitter (ms), Packet Loss Ratio (%), Bandwidth (Mb/s), Availability (%), Response Time (ms) |

| Symbol | Meaning |
|---|---|
| $S$ | $= \{s_1,\dots,s_m\}$, Set of CSU sites |
| $m$ | Number of CSU sites that consume IaaS/NaaS services from CSPs |
| $s_i$ | i-th CSU site in $S$ |
| $s_{sr}$, $s_d$ | *Source site, destination site* |
| $C$ | $= \{c_1,\dots,c_n\}$, Set of CSPs Data Centers |
| $n$ | Number of CSPs Data Centers |
| $c_j$ | j-th CSP Data Center in $C$ |
| $R_{ij}$ | $= \{r_1,\dots,r_K\}$, Set of CSPs network routes between $s_i$ and $s_j$ |
| $K$ | Number of CSPs network routes between $s_i$ and $s_j$ |
| $r_k$ | $= [b_1,\dots,b_{zk}]$, k-th route in $R_{ij}$, *List of CSPs (BoD)* |
| $z_k$ | Number of particular CSPs (BoD) forming the route $r_k$ |
| $b_i$ | i-th CSP (BoD) offering NaaS services with different service levels |
| $SL_k$ | $= \{sl_{1k},\dots,sl_{Xk}\}$, Set of service levels combination of traffic offered by $r_k$ |
| $X_k$ | Number of service level combination in $r_k$ |
| $sl_{xk}$ | x-th service level combination in $SL_k$ |
| $cost_{xk}$ | Cost unit of $sl_{xk}$ (*$* per *GB*) |
| $Cost_t$ | Total cost of selected routes for all sites for a specific CSP (DC) |
| $IVM$ | $=\{[Jlu_1, nbVu_1],\dots,[Jlu_D, nbVu_D]\}$, User VM Requirement |
| $Jlu_d$ | d-th User job length *(Instructions)* and key of *IVM* Hash table |
| $nbVu_d$ | Required User VM number, value of IVM Hash table, $=IVM[Jlu_d]$ |
| $D$ | Number of different *Jlu* required by the CSU |
| $VT_j$ | $=\{vt_1,\dots,vt_N\}$, Set of VM Types in $c_j$ |
| $N$ | Number of VM Types |
| $vt_{ij}$ | i-th VM type in $VT_j$ |
| $VCa_{ij}$ | VM Computing Capacity of $vt_{ij}$ *(GHz)* |
| $nbVc_{ij}$ | Available number of $vt_{ij}$ |
| $RT_{ij}$ | Response Time of $vt_{ij}$ corresponding to a particular $Jlu_d$ |
| $A_{ij}(VM)$ | Availability of $vt_{ij}$ |
| $VM\_cost_{ij}$ | Cost unit of $vt_{ij}$ (*$* per *hour*) |

### 4.3.1 Problem Statement: Constraint Optimization Problem

The selection of the best CSPs according to QoS splitting for a given iSLA is a multi-objective constraint optimization problem (QoS offered, total cost, etc.). This problem becomes important and challenging as the number of functionally equivalent services offered by CSPs at different QoS levels increases.

The goal is to minimize the cost subject to QoS Constraints (2) to (13). A feasible and an optimal CSPs selection must be ensured. A feasible selection means that aggregated QoS values from selected CSPs satisfy the CSU QoS requirements (Constraints (2) to (13)). Then, an optimal selection is considered as the feasible selection that minimizes the overall cost value (Alg. 1 and Alg. 2). However, if there is the same minimum cost for different offers, the goal becomes to maximize a utility function (1) for these offers to select the optimal one using different normalized weights $w_i$ assigned by the CSU for each *i-th* QoS parameter based on its importance and the type of application.

$$f(sl) = \sum_{i=1\to q}(w_i \times |Ur_i - Uo_i| / Ur_i), \text{ where } \sum_i w_i = 1 \quad (1)$$

$Ur_i$ is the value of *i-th* QoS parameter requested by the CSU and $Uo_i$ is the *i-th* QoS parameter aggregated value for a CSP service level offer (*sl*) whereas $q$ is the number of QoS parameters. Therefore, due to the fact that all selections are feasible and to the usage of the absolute value, the highest utility functions value indicates the best offer. Indeed, an estimated end-to-end value of a QoS attribute can be computed by aggregating the corresponding values of QoS attributes offered by CSPs. In this model, three types of QoS aggregation functions are considered: summation (Σ), product (Π) and minimum (min) relation.

### 4.3.2 Optimization and Path Selection Algorithm

The CSU requests services to run applications in two manners. In the first one, the application communicates Site-to-DC then DC-to-Site (Site/Site); for example, videoconferencing communication between sites using IaaS resources or intensive computing workflows executed by IaaS resources within a DC for a specific site. The second kind of communication can be Site-to-DC or DC to-Site (Site/DC); for example, storing data in the cloud (DC) or using a Video on Demand

(VoD) service stored in the cloud (DC). The Cloud Broker in Broker architecture or the $CSP_L$ (scenario 1) with other CSPs (DC) (scenario 2) in Federation architecture select the best CSPs (BoD) in terms of minimal cost and QoS guarantee using a proposed optimization and path selection algorithm (Alg. 1), subject to NaaS QoS parameters Constraints (2) to (6) for Site/Site scenario and Constraints (7) to (11) for Site/DC scenario when the CSU requests IaaS with NaaS QoS guarantee, or subject to minimal cost in IaaS case only. In addition, in case of IaaS with NaaS QoS guarantee, the algorithm uses the utility function (see Sec. 4.3.1) with only NaaS QoS parameters.

At first, according to Algorithm 1, the Cloud Broker or the $CSP_L$ with/ without other CSP (DC) calculate and sort in ascending order the cost of different service levels combinations offered by CSPs (BoD) for each route between a site $s$ and a specific CSP (DC) $c_j$. Then, if the CSU requests IaaS without NaaS services, the Cloud Broker or the $CSP_L$ with/without other CSP (DC) select a route with a minimal cost between a site $s$ and $c_j$ (Site/DC) or between a source site ($s_{sr}$) and a destination site ($s_d$) traversing $c_j$ (Site/Site). However, if the CSU requests IaaS with NaaS services, the Cloud Broker or the $CSP_L$ with/without other CSP (DC) calculate and sort in descending order the utility function value of different offers that have the minimal route cost between $s$ and $c_j$ (Site/DC) or between $s_{sr}$ and $s_d$ traversing $c_j$ (Site/Site), while guaranteeing the CSU QoS requirements (Constraints (2) to (6) for Site/Site and Constraints (7) to (11) for Site/DC). Then, they select the route corresponding to the minimal cost with maximal utility function value. Finally, when the Cloud Broker or the $CSP_L$ with/without other CSP (DC) do not find a route that meets the CSU requirements, they return null. However, if all sites have routes, they propose the selected routes with the total cost.

$$L(iSLA) \geq \sum_{i1=1\rightarrow zk1} L(b_{i1}) + 2 \times L(c_j) + \sum_{i2=1\rightarrow zk2} L(b_{i2}) \quad (2)$$

$$J(iSLA) \geq \sum_{i1=1\rightarrow zk1} J(b_{i1}) + 2 \times J(c_j) + \sum_{i2=1\rightarrow zk2} J(b_{i2}) \quad (3)$$

$$P(iSLA) \geq 1 - \prod_{i1=1\rightarrow zk1}(1 - P(b_{i1})) \times (1 - P(c_j))^2 \times \prod_{i2=1\rightarrow zk2}(1 - P(b_{i2})) \quad (4)$$

$$BW(iSLA) \leq \min(\min_{i1=1\rightarrow zk1}(BW(b_{i1})),\ BW(c_j),\ \min_{i2=1\rightarrow zk2}(BW(b_{i2}))) \quad (5)$$

$$A_{NaaS}(iSLA) \leq \min(\min_{i1=1\rightarrow zk1}(A(b_{i1})),\ A(c_j),\ \min_{i2=1\rightarrow zk2}(A(b_{i2}))) \quad (6)$$

where $b_{i1} \in r_{k1}$, $b_{i2} \in r_{k2}$ and $r_{k1}$ is the route between $s_{sr}$ and $c_j$ whereas $r_{k2}$ is the route between $c_j$ and $s_d$.

$$L(iSLA) \geq \sum_{i=1\rightarrow zk} L(b_i) + L(c_j) \quad (7)$$

$$J(iSLA) \geq \sum_{i=1\rightarrow zk} J(b_i) + J(c_j) \quad (8)$$

$$P(iSLA) \geq 1 - \prod_{i=1\rightarrow zk}(1 - P(b_i)) \times (1 - P(c_j)) \quad (9)$$

$$BW(iSLA) \leq \min(\min_{i=1\rightarrow zk}(BW(b_i)), BW(c_j)) \quad (10)$$

$$A_{NaaS}(iSLA) \leq \min(\min_{i=1\rightarrow zk}(A(b_i)), A(c_j)) \quad (11)$$

where $b_i \in r_k$ and $r_k$ is the route between $s$ and $c_j$.

**Algorithm 1** Best path selection between CSU Sites and a specific CSP (DC) $c_j$ [Get_suitable_Sites_routes(S, $c_j$)]

```
for each s ∈ S do
    R_sc ← Get_routes_between(s, c_j)
    for each r_k ∈ R_sc do
        for each sl_xk ∈ SL_k do
        T_3[s].Enqueue(r_k, sl_xk, cost_xk)
        end for
    end for
  T_3[s].sorting_ascending_order(cost)
  end for
  for each s_sr and/or s_d ∈ S do
    if IaaS without NaaS then
        if Site/DC then
        T.Enqueue(T_3[s_sr].Dequeue(),0)
      //0 means No NaaS QoS guarantee
        else if Site/Site then
  {r_k1,sl_xk1,cost_xk1}← T_3[s_sr].Dequeue()
  {r_k2,sl_xk2,cost_xk2}← T_3[s_d].Dequeue()
      //offers with the minimal cost
          T.Enqueue(r_k1.r_k2,sl_xk1.sl_xk2,
                         cost_xk1 + cost_xk2, 0)
// the symbol . is used as concatenation
        end if
```

```
      else if IaaS with NaaS then
         if Site/DC then
         while T_3[s_sr]≠∅ && flag_1==0 do
   {r_k,sl_xk,cost_xk}← T_3[s_sr].Dequeue()
            if Constraints(7)to(11)are
            met with r_k, sl_xk then
               flag_1=1
       T.Enqueue(r_k,sl_xk,cost_xk,f(sl_xk))
         while T_3[s_sr]≠∅ && flag_2==0 do
   {r_j,sl_xj,cost_xj}← T_3[s_sr].Dequeue()
            if cost_xk == cost_xj then
              if Constraints(7)to(11)
              are met with r_j,sl_xj then
       T.Enqueue(r_j,sl_xj,cost_xj,f(sl_xj))
               end if
            else
               flag_2=1
            end if
          end while
     T.sorting_descending_order(f(sl))
         end if
        end while
       else if Site/Site then
         min_c = T_3[s_sr].get_max_cost()+
T_3[s_d].get_max_cost()//initialization
         while T_3[s_sr] ≠ ∅ do
           T_1← T_3[s_d]
  {r_k1,sl_xk1,cost_xk1}← T_3[s_sr].Dequeue()
            while T_1 ≠∅ && flag_1==0 do
       {r_k2,sl_xk2,cost_xk2}← T_1.Dequeue()
               if Constraints(2)to(6)
          are met with r_k1 and r_k2 then
                flag_1=1
       if min_c == cost_xk1 + cost_xk2 then
           T.Enqueue(r_k1.r_k2,sl_xk1.sl_xk2,
                    min_c,f(sl_xk1.sl_xk2))
           while T_1 ≠ ∅ && flag_2==0 do
          {r_j,sl_xj,cost_xj}← T_1.Dequeue()
            if cost_xk1 == cost_xj then
              if constraints(2)to(6)
           are met with r_k1 and r_j then
            T.Enqueue(r_k1.r_j,sl_xk1.sl_xj,
                    min_c,f(sl_xk1.sl_xj))
              end if
            else
               flag_2=1
            end if
           end while
  else if min_c > cost_xk1 + cost_xk2 then
           T ← ∅
           min_c =  cost_xk1 + cost_xk2
           T.Enqueue(r_k1.r_k2,sl_xk1.sl_xk2,
                    min_c,f(sl_x1.sl_xk2))
             while T_1≠∅ && flag_2==0 do
          {r_j,sl_xj,cost_xj}← T_1.Dequeue()
                  if cost_xk1 == cost_xj then
                    if constraints(2)to(6)
                 are met with r_k1 and r_k then
                  T.Enqueue(r_k1.r_j,sl_xk1.sl_xj,
                           min_c,f(sl_xk1.sl_xj))
                            end if
                         else
                            flag_2=1
                         end if
                      end while
                   end if
                end if
             end while
          end while
     T.sorting_descending_order(f(sl))
         end if
      end if
      if T ≠ ∅ then
    {r_k,sl_x,cost_xk,f(sl)}← T.Dequeue()
         Cost_t = Cost_t + cost_xk
        Q.Enqueue(s_sr,s_d,r_k,sl_xk,cost_xk)
      else
        return ∅
    end if
end for
Q.Enqueue(Cost_t)//add the total cost
return Q
```

#### 4.3.3 Optimization and Resource Selection Algorithm

When the CSU requests IaaS with/without NaaS services, it interacts with the Cloud Broker or the $CSP_L$ specifying its requirements using the proposed GUI. Then, the Cloud Broker or the $CSP_L$ selects the best CSP resources in terms of minimal cost and QoS guarantee using a proposed optimization and resource selection algorithm (Alg. 2) subject to IaaS QoS parameters splitting Constraints (12) and (13). In this algorithm, the utility function uses IaaS with/without NaaS QoS parameters according to the CSU requirements. Note that Algorithm 1 is reused within Algorithm 2.

At first, using Algorithm 1, the Cloud Broker in Broker architecture gets the best routes between CSU sites and each CSP (DC) $c_j$, or only the $CSP_L$ (scenario 1) with other CSPs (DC) (scenario 2) in Federation scenario get the best routes between CSU sites and themselves. Furthermore, the Cloud Broker or the $CSP_L$ calculates and sorts in ascending order the cost of VM types that meet the CSU requirements (Constraint (12) and (13)) while taking into account the best routes cost (Alg. 1). Then, it calculates and sorts in descending order the utility function values of different offers that have the same global cost. Note that, in Federation architecture, when the $CSP_L$ calculates its resources, *C* (set of CSPs Data Centers) used in Algorithm 1 contains only the $CSP_L$ Data Centers. However, when the $CSP_L$ calculates resources in other CSPs, *C* contains all CSPs Data Centers without $CSP_L$ Data Centers. Finally, if the Cloud Broker or the $CSP_L$ does not find CSPs that meet the CSU requirements, it releases allocated resources and it rejects the user request. However, if all CSU requirements are met, it allocates the best VM resources and routes, accepts user request, and establishes iSLA with the CSU, and B_iSLA and D_iSLA with selected CSPs.

$$RT(iSLA) \geq RT_{ij} \quad (12)$$

$$A_{IaaS}(iSLA) \leq A_{ij}(VM) \quad (13)$$

**Algorithm 2** CSPs Selection and VMs Resource Allocation

```
for each c_j ∈ C do
T←Get_suitable_Sites_routes(S,c_j)//Alg.1
   if T ≠ ∅ then
     for each Jlu_d ∈ IVM do
       for each vt_ij ∈ VT_j do
  RT_ij= Get_appropriate_RT(JLu_d, VCa_ij)
         if Constraints(12)and(13)
                          are met then
     Q.Enqueue(c_j,T,Jlu_d,vt_ij,VM_cost_ij)
         end if
       end for
     end for
   end if
 end for
Q.sorting_ascending_order(VM_cost+Cost_t)
{c_j,T,Jlu_d,vt_ij,VM_cost_ij}← Q.Dequeue()
       // offer with the minimal cost
T_1.Enqueue(c_j,T,Jlu_d,vt_ij,VM_cost_ij,f(sl))
 while Q ≠ ∅ do
   flag = 0
   while Q ≠ ∅ && flag==0 do
     {c'_j,T',Jlu'_d,vt'_ij,VM_cost'_ij}
       ← Q.Dequeue() //the next offer
if VM_cost_ij+ Cost_t==VM_cost'_ij+ Cost'_t
         T_1.Enqueue(c'_j,T',Jlu'_d,vt'_ij,
                     VM_cost'_ij,f(sl))
      else
        flag=1
         end if
      end while
    T_1.sorting_descending_order(f(sl))
      Q_1.Enqueue(T_1)
      if Q ≠ ∅  then
         T_1← ∅
         T_1.Enqueue(c'_j,T',Jlu'_d,vt'_ij,
                     VM_cost'_ij,f(sl))
         VM_cost_ij = VM_cost'_ij
         Cost_t = Cost'_t
      end if
   end while
while Q_1≠∅ and ∃IVM[Jlu_x]≠0,∀1≤x≤D do
    {c_j,T,Jlu_d,vt_ij,VM_cost_ij,f(sl)}
                           ← Q_1.Dequeue()
    while nbVc_ij≠0 and IVM[Jlu_d]≠0 do
     allocate vt_ij in c_j with VM_cost_ij
       nbVc_ij − − , IVM[Jlu_d] − −
       if c_j ∉ T_2 then
           T_2.Enqueue(c_j)
           while T ≠ ∅ do
    {s_sr,s_d,r_k,sl_xk,cost_xk}← T.Dequeue()
          allocate r_k with sl_xk, cost_xk
                between s_sr, c_j and s_d
         end while
       end if
    end while
end while
if ∃  IVM[Jlu_x] ≠ 0, ∀ 1 ≤ x ≤ D then
release resources, reject CSU request
else
  accept CSU request
end if
```

### 4.4 Cost Calculation

Based on the cloud computing characteristic "Pay as you go", the CSU replaces its upfront capital expense with low variable cost and pays only for what it uses. For computing resources, it pays on an hourly basis from the time it launches a resource until the time it terminates it. For data transfer, it pays on a per gigabyte basis. This section introduces a general methodology to calculate costs. It is similar to the way Amazon is charging its clients. Although almost all other CSPs (like RackSpace, GoGrid, Azureus, etc.) offer their own payment methods, they basically follow Amazon's model [28]. However, AWS does not consider separately CSP (BoD) cost and does not calculate violation cases of all QoS parameters because they are not included in its

SLAs. Therefore, the cost is calculated based on (14):

$$Cost_{total} = Cost_{VM} + Cost_{BW} \quad (14)$$

where, $Cost_{total}$ is the total cost of the different CSU resource consumption. For instance, the proposed architectures offer IaaS and NaaS services. Therefore, $Cost_{VM}$ is the total cost of VMs resources calculated based on (15) and $Cost_{BW}$ is the total cost of bandwidth resources calculated based on (16). Note that, each CSP has different costs for different resources types, and these costs are sent at first to the Cloud Broker or the $CSP_L$.

$$Cost_{VM} = \Sigma_j \Sigma_i (VM\_cost_{ij} \times t_{ij}) \quad (15)$$

$$Cost_{BW} = \Sigma_k (BW\_cost_k \times BW_k) \quad (16)$$

where, $VM\_cost_{ij}$ is the cost unit ($ per hour) of a selected VM type $vt_{ij}$ in a selected $CSP_j$ (DC), and $t_{ij}$ is the number of $vt_{ij}$ consumption hours. Note that the cost of memory capacity for each VM is included in $VM\_cost_{ij}$. $BW\_cost_k$ is the cost unit ($ per GB) of the traffic traversing a selected $CSP_k$ (DC and/or BoD) with a guaranteed QoS level and $BW_k$ (GB) is the bandwidth consumed by the CSU traffic traversing $CSP_k$.

## 5 USAGE CASES AND EVALUATION

To take full advantage of the proposed cloud networking framework, this section presents two usage cases for the efficient establishing of large-scale cloud videoconferencing and intensive computing applications. The corresponding scenarios are tested as a proof of concept and its performances are evaluated by conducting a set of different simulations using the CloudSim toolkit [30]. CloudSim toolkit supports both system and behavior modeling of cloud system components such as Data Centers, VMs and resource provisioning policies that can be extended. Therefore, CloudSim is extended to support three new entities. The first one is a CSU entity and the second is a Cloud Broker entity, instead of CloudSim DatacenterBroker entity, where the CSU entity sends its requirements to the Cloud Broker entity to allocate the best resources and establish SLAs as described in Section 4. In addition, a CSP (BoD) entity is proposed to interconnect Broker, CSPs (DC) and CSUs using BRITE topology [31] for modeling link bandwidth and associated latencies. The simulated model of the proposed architecture is composed of one Broker, four CSPs (BoD) and four CSPs (DC) containing each one 10 hosts. Each host has a quad-core processors (4 × 1,2GHz) and 16GB of RAM.

All entities are initiated at the beginning of the simulation. In this simulation model, for a Gold service level for example, a CSP (DC) can have the IaaS characteristics presented in Table 2 and NaaS characteristics presented in Table 3. In addition, for a Gold service level, a CSP (BoD) can have the NaaS characteristics presented in Table 3. Other CSPs (DC/BoD) have different values for each service level (Platinum, Gold, Silver, Bronze).

**Table 2.** IaaS Gold Service Level.

| | VM Capacity | $A_{IaaS}$ | CPU Cost |
|---|---|---|---|
| $CSP_1$ (DC) | 750 MHz | 99.999% | 0.3 $ per hour |

**Table 3.** NaaS Gold Service Level.

| | L | J | P | BW | $A_{NaaS}$ | BW Cost |
|---|---|---|---|---|---|---|
| $CSP_1$ (DC) | 7 ms | 1 ms | $2.5x 10^{-3}$ | 10 Mb/s | 99.999 % | 0.05 $ per GB |
| $CSP_1$ (BoD) | 12 ms | 2 ms | $2.5x 10^{-3}$ | 10 Mb/s | 99.999 % | 0.3 $ per GB |

The CSU specifies its requested service type and corresponding QoS requirements to the Cloud Broker or the $CSP_L$ using the proposed GUI (Fig. 5). Then, after the best CSPs selection, the Cloud Broker or the $CSP_L$ allocates required resources to CSU sites and establishes the different SLAs. The participating entities start the application and when it is finished, the Cloud Broker or the $CSP_L$ releases allocated resources and the CSU pays as it uses.

### 5.1 Usage Case 1: Cloud videoconferencing scenario

From the perspective of bandwidth requirements, end-to-end delay and jitter constraints, multi-party videoconferencing may be one of the most demanding multimedia applications [27]. A videoconference system is a real-time exchange of media data among several parties, which breaks the limit of human communications due to the geographical location of participants. Therefore, to assess the performance of the proposed framework while considering the videoconferencing service, CSPs selection algorithms (Alg. 1 and Alg. 2) are used to validate the videoconferencing scenario within the cloud networking architectures (Broker and Federation). These algorithms determine the best delivery paths of videoconferencing streams among allocated VMs for videoconferencing servers. It is possible that VMs are distributed in different Data Centers. The objective of the Cloud Broker or the $CSP_L$ is to minimize the total service cost without violating end-to-end QoS constraints.

Three videoconferencing simulation scenarios are evaluated. The first one corresponds to a static selection of resources without QoS guarantee. The second and third scenarios are respectively a Broker and a Federation based selection with QoS guarantee. In each scenario, multiple CSU sites are connected to different CSPs (BoD) with possible multiple destinations. The CSU chooses IaaS (without storage) and NaaS services in both Broker and Federation scenarios.

The global average streaming delay and jitter from source to destination for these scenarios are evaluated. Formally, the delay is defined as a period of time necessary for a bit of data to travel from one endpoint to another. However, the jitter is defined as the delay variation, i.e., the absolute value of the difference between the arrival times of two adjacent packets minus their departure times [32]. In our case, for a specific source ($s_{sr}$) and destination ($s_d$) sites, VM type ($vt_i$) and CSP (DC) ($c_j$), the delay ($D$) is calculated using the equation (17). For each source site the CSU requests a VM for transcoding. In addition, at each simulation the $CSP_L$ in Federation scenario and destination sites in all scenarios are selected randomly.

$$D = \sum_{i1=1\rightarrow zk1} L(b_{i1}) + L(c_j) + RT_{ij} + L(c_j) + \sum_{i2=1\rightarrow zk2} L(b_{i2}) \qquad (17)$$

The delay and the jitter are very important QoS parameters for real time interactive applications such as videoconferencing. An end-to-end delay greater than 200 ms and a jitter greater than 30 ms could cause a degraded videoconferencing service. Therefore, the CSU specifies a network latency less than 180 ms, a network jitter less than 30 ms and a transcoding response time less than 20 ms in the Broker and Federation scenarios. Other QoS parameters are Gold and weights equal to 1.

Four CSU video types are simulated with one hour length, a size of 1, 2, 3 and 4 GB, respectively and requested bandwidth equal to 2.2Mb/s, 4.5Mb/s, 6.8Mb/s and 9.1Mb/s, respectively. According to several experiments for each video type with different CSU sites location, The global average delay including Response Time and network Latency, the global average jitter and the global Bandwidth and VMs costs are calculated.

The results illustrated in Figure 8 and Figure 9, reveal a good streaming delay achieved by the Broker and Federation proposal, as compared to a static selection (Fig. 10). Therefore, in all Broker and Federation video types, the global end-to-end delays are well controlled to keep queuing delays to the minimum while packets traverse through intermediate Data Centers on their paths. Moreover, the global jitter (Fig. 11) is also well controlled (less than 30 ms) and does not impact the global videoconferencing QoS.

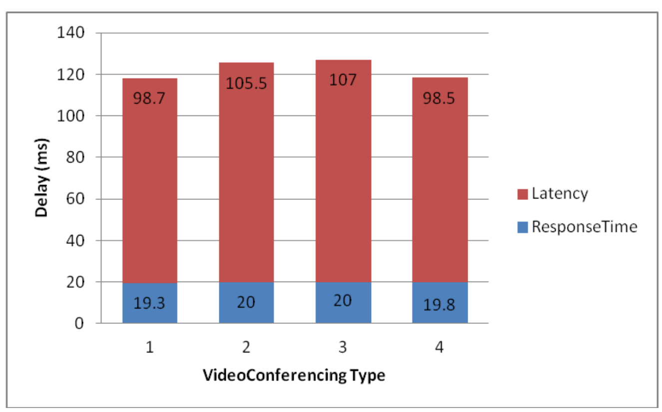



**Figure 8.** End-to-end Global Delays in Broker Selection.

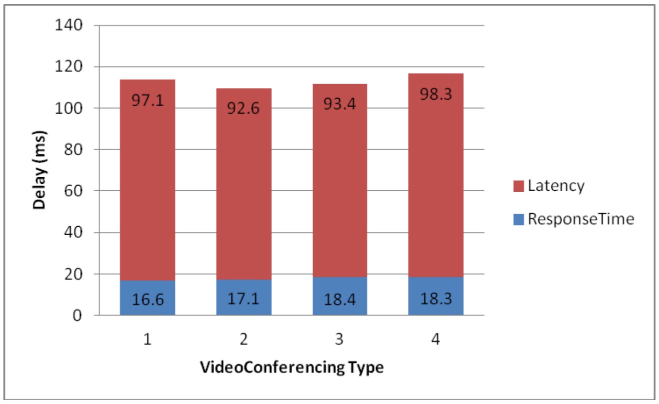



**Figure 9.** End-to-end Global Delays in Federation Selection.

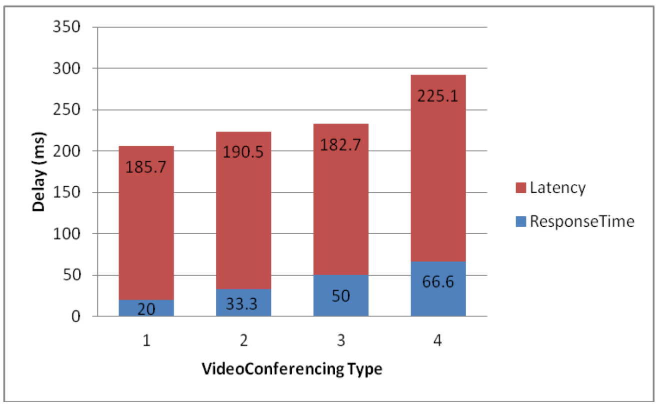



**Figure 10.** End-to-end Global Delays in Static Selection.

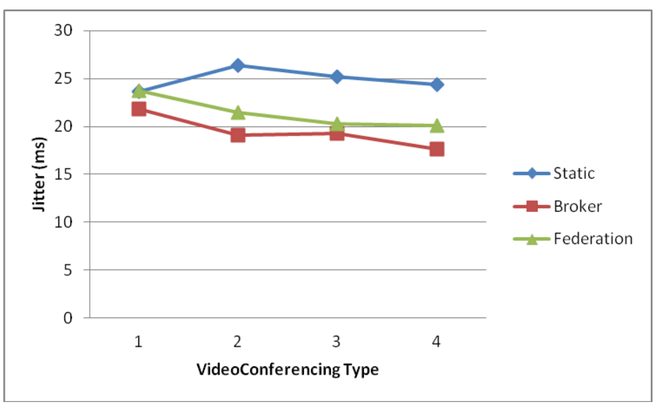



**Figure 11.** Jitter comparison between Static, Broker and Federation Selection.

In addition, as shown in Figure 12 and Figure 13, the global bandwidth and VMs costs increase when the video size and bandwidth corresponding to a video type increase due to the NaaS and IaaS QoS constraints. Moreover, in a static selection, the global bandwidth and VMs costs are less than those of the Broker and Federation selection due to the additional IaaS and NaaS QoS guarantee by the Broker or the $CSP_L$. Furthermore, the global bandwidth and VMs costs in the Broker scenario are less important than the Federation scenario costs due to the selection of resources within the $CSP_L$ firstly. Indeed, $CSP_L$ resources are not usually the best and that's why the Broker architecture is the most economical, while ensuring QoS requirements.

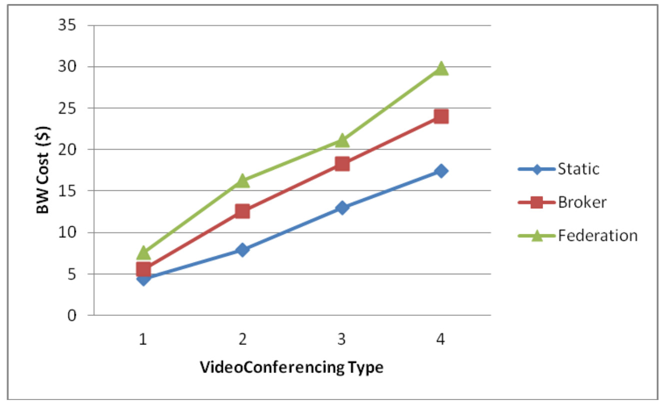



**Figure 12.** Global Bandwidth cost comparison.

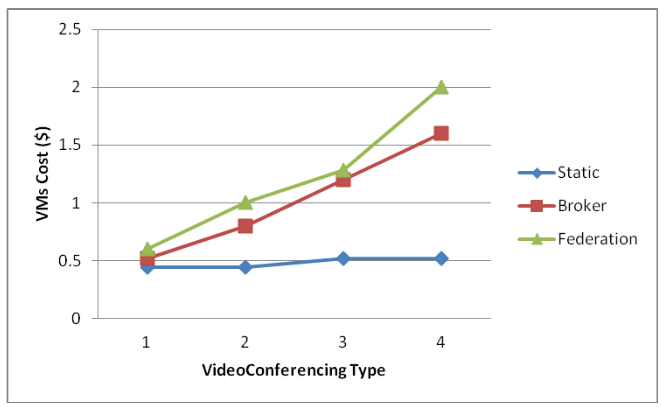



**Figure 13.** Global VMs cost comparison.

### 5.2 Usage Case 2: Cloud Intensive Computing

The cloud computing services are mainly used for the treatment of huge amount of information [34] such as intensive computing applications which are based on requests for the execution of computationally intensive tasks, for example, scientific workflows. CSPs selection algorithms are used to validate this scenario within Broker and Federation based cloud networking architectures and to determine the best VMs allocated for requested jobs while minimizing the total application cost without violating end-to-end QoS constraints. The CSU service requirements are simulated as a series of job requests executed by allocated VMs.

Three intensive computing simulation scenarios are evaluated. In the first scenario, the CSU chooses IaaS without NaaS service and a response time less than 300 ms. In the second scenario, the CSU chooses IaaS with NaaS service, a response time less than 300 ms, a latency less than 150 ms, a bandwidth of 1Mb/s, other QoS parameters are equal to Silver and all weights are equal to 1. In the third scenario, the CSU chooses a static selection without QoS guarantee.

According to several experiments with four CSU sites where each one sends 100 jobs to four VMs, and for different job lengths (200, 250, 300, 350 instructions), the global average delay (average Response Time and Latency) as well as the global Bandwidth and VMs costs are calculated.

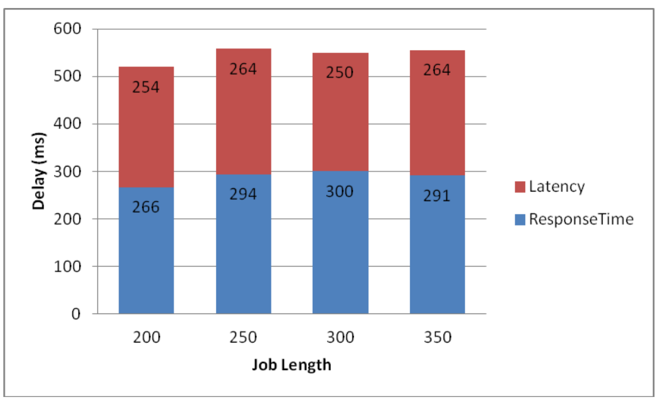



**Figure 14.** End-to-end Global Delays for IaaS without NaaS in Broker Scenario.

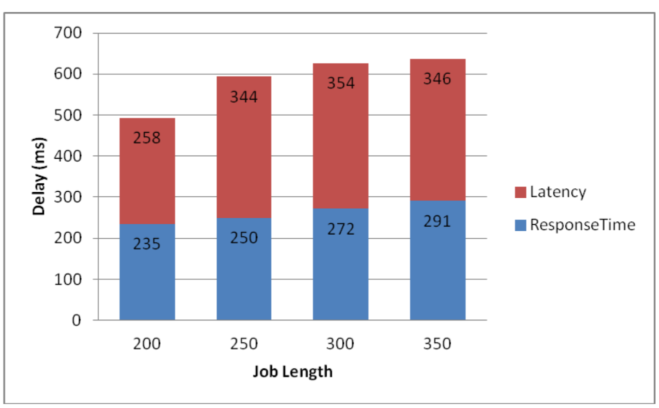



**Figure 15.** End-to-end Global Delays for IaaS without NaaS in Federation Scenario.

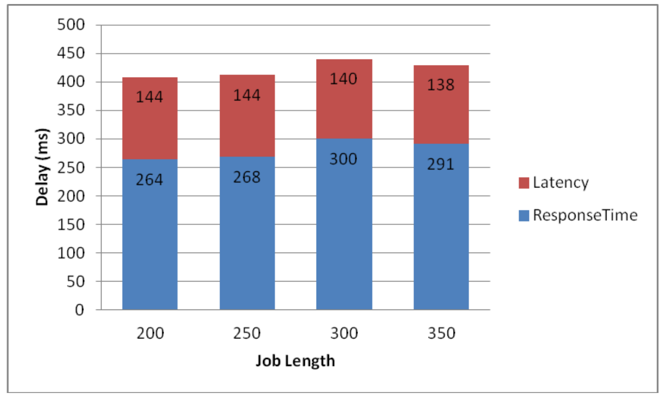



**Figure 16.** End-to-end Global Delays for IaaS with NaaS in Broker Scenario.

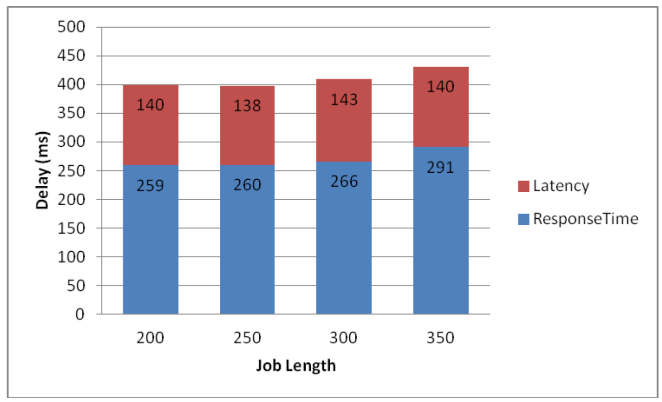



**Figure 17.** End-to-end Global Delays for IaaS with NaaS in Federation Scenario.

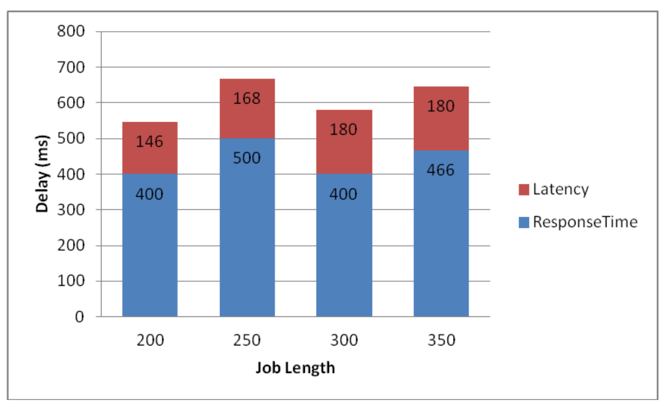



**Figure 18.** End-to-end Global Delays for Static Selection without QoS guarantee.

As illustrated in Figure 14 and Figure 15, by providing only IaaS QoS guarantee, the global Response Time is well controlled but the Latency is not controlled in both Broker and Federation selection. However, as shown in Figure 16 and Figure 17, by providing NaaS and IaaS QoS guarantee, the Response Time and the Latency are well controlled as compared to a static selection (Fig. 18).

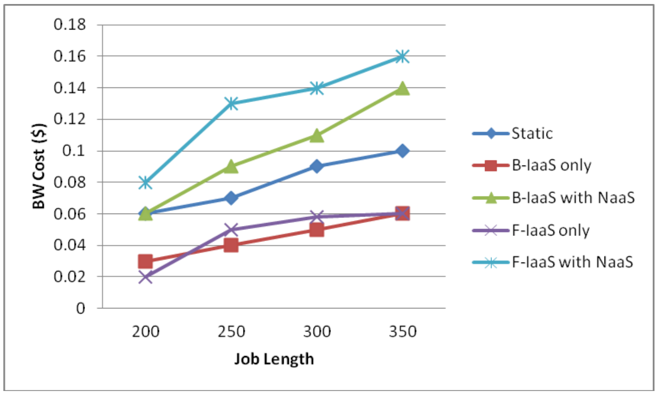



**Figure 19.** Global Bandwidth cost comparison.

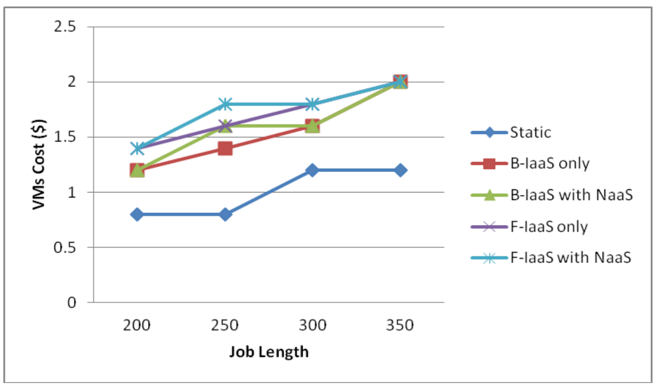



**Figure 20.** Global VMs cost comparison.

In addition, the results obtained in Figure 19 show that the global bandwidth cost increases when the job length increases. Moreover, the global bandwidth cost without NaaS services (static and IaaS without NaaS for Broker and Federation) is less than the global bandwidth cost of Broker and Federation with NaaS selection due to the NaaS QoS guarantee by the Cloud Broker or the $CSP_L$. Furthermore, due to the selection of minimal route cost by Algorithm 1 in IaaS without NaaS selection for Broker and Federation, their global bandwidth cost is less than the global bandwidth cost of static selection.

Furthermore, Figure 20 demonstrates that the global VMs cost increases when the job length increases due to the CSU Response Time constraint. In addition, the global VMs cost in static selection is less than the VMs cost for IaaS with/without NaaS (Broker and Federation selection) due to the IaaS QoS guarantee by the Broker and the $CSP_L$. Moreover, the global VMs cost in the Broker scenario are less important than the Federation scenario due to the selection of resources in the $CSP_L$ at first. Therefore, according to the same reasons described in Usage Case 1, the Broker architecture is the most economical cloud networking proposed architecture while ensuring QoS requirements.

## 6 CONCLUSION

In this paper, a framework was presented for resource allocation in conformance with three type of SLA established within cloud networking environment. At first, two architectures of cloud networking (Broker and Federation) were proposed. The paper has focused on QoS parameters for IaaS and NaaS services. In addition, the calculation of these QoS parameters was presented and two algorithms were specified for the best CSPs selection using optimization under QoS constraints to achieve a minimizing cost objective. Moreover, the paper has proposed the establishment of an iSLA with the CSU, a D_iSLA with CSPs (DC) and a B_iSLA with CSPs (BoD). Finally, the proposed architectures were evaluated for two different applications (cloud videoconferencing and intensive computing), and good performance results were obtained showing that the Broker architecture is the most economical as compared to the Federation one while ensuring QoS requirements.

## 7 REFERENCES

1. Gunawi, H., Do, T., Hellerstein, J., Stoica, I., Borthakur, D., Robbins, J.: Failure as a service (faas): A cloud service for large-scale, online failure drills. Technical Report UCB/EECS-2011-87 (2011).
2. Paquette, S., Jaeger, P., Wilson, S.: Identifying the security risks associated with governmental use of cloud computing. Government Information Quaterly, 27(3):245--253 (2010). DOI: 10.1016/j.giq.2010.01.002
3. Kouki, Y., Ledoux, T.: CSLA: a Language for improving Cloud SLA Management. In: Proc. International Conference on Cloud Computing and Services Science, pp. 586--591 (2012).
4. Yanzhi, W., Shuang, C., Pedram, M.: Service Level Agreement-Based Joint Application Environment Assignment and Resource Allocation in Cloud

Computing Systems. In: Proc. IEEE Green Technologies Conference, pp. 167--174 (2013).
5. Armbrust, M., Fox, A., Griffith, R.: Above the clouds: a berkeley view of cloud computing. Technical Report, UCB/EECS-2009-28 (2009).
6. The NIST Definition of Cloud Computing. Special Publication, USA, 800--145 (2011).
7. Amazon EC2. http://aws.amazon.com/ec2/ [retrieved: 06, 2015]
8. Murray, P., Sefidcon, A., Steinert, R., Fusenig, V., Carapinha, J.: Cloud Networking: An Infrastructure Service Architecture for the Wide Area. Future Network, pp. 1--8 (2012).
9. ITU-T, Focus Group on Cloud Computing, Technical Report (2012).
10. IEEE cloud computing. http://cloudcomputing.ieee.org [retrieved: 06, 2015]
11. Cloud Portability and Interoperability Profiles (CPIP) Standard. http://standards.ieee.org/develop/project/2301.html [retrieved: 06, 2015]
12. Standard for Intercloud Interoperability and Federation (SIIF) Standard. http://standards.ieee.org/develop/project/2302.html [retrieved: 06, 2015]
13. Open Cloud Computing Interface (OCCI). http://occi-wg.org/ [retrieved: 06, 2015]
14. Global Inter-Cloud Technology Forum (GICTF). http://cloud-standards.org [retrieved: 06, 2015]
15. Benson, T., Akella, A., Shaikh, A., Sahu, S.: CloudNaaS: a cloud networking platform for enterprise applications. In: Proc. of 2nd ACM Symposium on Cloud Computing, pp. 1--13 (2011). DOI: 10.1145/2038916.2038924
16. Goiri, I., Julia, F., Fito, J., Macıas, M., Guitart, J.: Resource-Level QoS Metric for CPU-Based Guarantees in Cloud Providers. 7th International Workshop, GECON 2010, 6296:34--47 (2010). DOI: 10.1007/978-3-642-15681-6_3
17. Cicotti, G., Coppolino, L., Cristaldi, R., Antonio, S., Romano, L.: QoS Monitoring in a Cloud Services Environment: the SRT-15 Approach. Euro-Par: Parallel Processing Workshops, 7155:15--24 (2012). DOI: 10.1007/978-3-642-29737-3_3
18. Karim, R., Chen, D., Miri, A.: An End-to-End QoS Mapping Approach for Cloud Service Selection. IEEE Ninth World Congress on Services, pp. 341--348 (2013).
19. Brandic, I., Feller, E.: Energy and QoS-aware workload management in clouds. Research report, COST group (2011).
20. Yanzhi, W., Xue, L., Pedram, M.: A Game Theoretic Framework of SLA-based Resource Allocation for Competitive Cloud Service Providers. In Proc of Sixth Annual IEEE Green Technologies Conference, pp. 37--43 (2014).
21. Patel, P., Ranabahu, A., Sheth, A.: Service Level Agreement in Cloud Computing. OOPSLA Cloud Computing workshop, pp. 1--10 (2009).
22. Ludwig, H., Keller, A., Dan, A., King, R., Franck, R.: Web service level agreement (WSLA) language specification, version 1.0. IBM Corporation pp. 815--824 (2003).
23. Scalable & Adaptive Internet Solutions (SAIL) Project. http://www.sail-project.eu/ [retrieved: 06, 2015]
24. Faniyi, F., Bahsoona, R., Theodoropoulos, G.: A Dynamic Data-Driven Simulation Approach for Preventing Service Level Agreement Violations in Cloud Federation. In Proc of International Conference on Computational Science, 9:1167--1176 (2012). DOI: 10.1016/j.procs.2012.04.126
25. Feng, H., Wu, W.: Framework and User Migration Strategy of Cloud-Based Video Conference Multi-Gateway System. 19th International Conference on High Performance Computing (HiPC), pp. 1--8 (2012). DOI: 10.1109/HiPC.2012.6507521
26. Wu, Y., Wu, C., Li, B., Lau, F.: vSkyConf: Cloud-assisted Multi-party Mobile Video Conferencing. In Proc of Second ACM SIGCOMM workshop on Mobile cloud computing. pp. 33--38 (2013). DOI: 10.1145/2491266.2491273
27. Feng, Y., Li, Ba., Li, Bo.: Airlift: Video Conferencing as a Cloud Service using Inter-Datacenter Networks. In Proc of 20th IEEE International Conference on Network Protocols. pp. 1--11 (2012). DOI: 10.1109/ICNP.2012.6459966
28. Cervino, J., Rodriguez, P., Trajkovska, I., Escribano, F., Salvachua, J.: A Cost-Effective Methodology Applied to Videoconference Services over Hybrid Clouds. Mobile Netw Appl, 18(1):103--109 (2013).
29. Kesavan, S., Jayakumar, J. : Network Performance Analysis of Cloud Based Multimedia Streaming Service. International Journal of New Computer Architectures and their Applications (IJNCAA), Vol. 4, No. 3, pp. 156--166 (2014). DOI : http://dx.doi.org/10.17781/p0015
30. Calheiros, R., Ranjan, R., Beloglazov, A., Rose, C., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environment and evaluation of resource provisioning algorithms. Software: Practice and Experience. 41(1):23--50 (2011). DOI: 10.1002/spe.995
31. Medina, A., Lakhina, A., Matta, I., Byer, J.: BRITE: An approach to universal topology generation. In Proc of the Ninth International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, U.S.A., pp. 346--353 (2001).
32. Jacobson, V., Nichols, K., Poduri, K.: An Expedited Forwarding PHB. RFC 2598 (1990).
33. Costa, P., et al.: "NaaS: Network-as-a-Service in the Cloud." In Proc of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, Hot-ICE. Vol. 12 (2012).
34. Alwesabi, A., Okba, K.: Security Method: Cloud Computing Approach Based on Mobile Agents. International Journal of New Computer Architectures and their Applications (IJNCAA), Vol. 4, No. 1, pp. 17--29 (2014). DOI : http://dx.doi.org/10.17781/p002

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 79-93
The Society of Digital Information and Wireless Communications, 2015 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

# Future Motion Decisions using State-action Pair Predictions

Masashi Sugimoto Kentarou Kurashige
Muroran Institute of Technology
Email: sm04322@gmail.com Email: kentarou@csse.muroran-it.ac.jp

## ABSTRACT

Robots that works in a dynamic environment must possess, the ability to autonomously cope with the changes in the environment. This paper proposes an approach to predict changes in the state and actions of robots. Further, this approach attempts to apply predicted future actions to current actions. This method predicts the robot's state and action for the distant future using the states that the robot adopts repeatedly. Using this method, the actions that the robot will take in the future can be predicted. The method proposed in this paper predicts the state and action of a robot each time it decides to perform an action. In particular, this paper focuses on defining weight coefficients, using the characteristics of the future prediction results. Using this method, the compensatory current action will be obtained. This paper presents the results of our study and discusses methods that allow the robot to quickly determine its most desirable action, using state prediction and optimal control methods.

## KEYWORDS

Online State Prediction, Prediction using Combination of State Space and Action, Online SVR, Control using State-action Prediction

## 1 INTRODUCTION

Over the years, many studies have been conducted with the objective of developing working robots that can facilitate or are suitable for dynamic environments [1]-[9]. Further, various robots have been developed to assist humans in workspaces such as homes and factories [10]. However, because robots are required to act differently in different situations, it is virtually impossible to predict all possible situations or to pre-program a robot with all suitable reaction patterns for each possible situation [11, 12]. Further, it is difficult to calculate the inverse problem of the robot for generating a suitable reaction in real time. Considering these problems, it can be argued that robots working in a dynamic environment must possess the ability to cope with changes in the environment. For mobile robots, which are the focus of this study, it is clear that behaviour and posture have crucially important meaning during task execution.

Some of the conventional studies related to robot control focus on the states of the target robot or the parameters of the robot. Based on these viewpoints, these studies have presented certain methods to control robots [13, 15]. Thus, it is important to know the states of the robot and to know that the robot will take an action in the next scene when robots in dynamic environments must be controlled. Moreover, the model used to obtain the states of the robot and to predict its actions must be known. Now let us consider controlling a robot in a dynamic environment. Actions determined by ordinary control methods may be suitable in most static environments, in which the robot performs the same action each time. Unfortunately, these actions may be unsuitable in dynamic environments, owing to problems such as computation delay (1 or some steps) and hardware overhead.

To address these problems, many studies have used machine learning techniques such as reinforcement learning (RL) that acquire the optimal action to learn the environment by trial and error [14, 16, 17]. Alternatively, model predictive control (MPC) has been used to sequentially provide data to the control system; using this technique, suitable input for forming predictions is gained each time. As a result, MPC is better suited for managing typical control rules [18, 19]. However, these techniques suffer from problems such as computation delay, hardware overhead, and whether the robot can respond flexibly to changes in the dynamic environment [20]-[24]. Some techniques for generating robot motion have also been presented. In these cases, ordinary control rules were combined with an extended Kalman filter (EKF) [1] or unscented Kalman filter (UKF) [1], to avoid linearizing the robot model [25]-[27]. However, these techniques still present various problems; applied filters will often become unstable, using a non-linear model is often difficult, and cases in which parameters must be defined are increased [28].

In contrast, some research efforts have utilized Support Vector Regression (SVR). In general, these works provides appropriate results [18, 19, 29]. However, these works avoided analyzing disturbances. Moreover, these works did not consider the relationship between a robot's state transition and the action taken by the robot. For example, in dynamic environments, there are unforeseen loads and backlash, complex friction, and stiction. In this case, it is essential to predict state and action. Furthermore, the model must be allowed to adapt to changes in the robot's dynamics.

Thus, applying state-action pair predictions to the control of robots has already been proposed [30, 31], on the basis of Online SVR [32]. This method predicts the robot's state and action, using them as a "pair" for determining future actions by applying the state that the robot adopts repeatedly. Using this method, the suitable actions that the robot can take in the future can be obtained [33]. Most of the previously mentioned conventional approaches are based on parametric techniques. In contrast, this study is based on non-parametric techniques. In this respect, the proposed method does not require strict parameters or a model of the target robot. Based on these characteristics, it is clear that the proposed method is different from other related methods.

This paper describes the results of these studies and discusses methods that allow the robot to quickly decide its desirable behavior, using the state prediction and conventional optimal control methods. First, we will attempt to apply the action to be taken in the future to the current action, by extending the former approach. In particular, in this study, we aim to determine the future action and apply it to the current action; further, the we describe the calculation of weight coefficients for future actions that were obtained through the predictions. Fixed-value weight coefficients were already proposed in [33]. Hence, in this paper, weight coefficients will be proposed that focus on the "variation of the predicted results." Using this method, compensatory current actions can be obtained more flexibly.

In this study, posture stabilization will be realized using a system that combines the optimal control and the proposed method. In particular, to verify and implement the proposed method, this study focused on an inverted pendulum that can be linearized around these operating points. From this example, an ordinary linear-quadratic regulator (LQR) can be applied. Using this method, this study aims to converge the posture to a stable state immediately, using future prediction results. Therefore, in this study, the proposed method was combined with the conventional LQR method. In particular, as an application example, the two-wheeled mobile inverted pendulum robot "NXTway-GS" was used to execute a stabilization control task. Moreover, the move instruction was sent from the command input within the stabilization control task. The control response results produced by the proposed method were compared with those of the conventional control method, using a computer simulation. The computer simulation showed that, in contrast with the ordinary control method, the proposed method, the proposed method is well adapted to the disturbance input; it obtained the action that decreased the pitch angle, and achieved a desirable state for the NXTway-GS.

This paper is organized as follows: In Section 2 explains the process of using future prediction techniques to obtain or decide the optimal action for the robot. In addition, details describing the proposed method are provided. In Section 3, the experimental setting is explained. Finally, Section 4 presents the conclusions of this study.

## 2 AN APPROACH FOR DECIDING THE OPTIMAL ACTION FOR A ROBOT

### 2.1 Basic Idea

We mentioned in Section 1 that for controlling a robot in a dynamic environment, an action can be chosen for (and adopted by) the current result, by predicting a future state using previous actions and states. In this study, we will aim to obtain the optimal action to minimize the body pitch angle of the inverted pendulum, in cases with continual input of predictable disturbances. To achieve this, we will aim to use predictions from state-action pairs, a technique we proposed in our previous studies [30], [31]. Therefore, in this study, a system that decides the optimal action using the proposed method illustrated in fig. 1 (based on previous studies, including [33]) is considered.

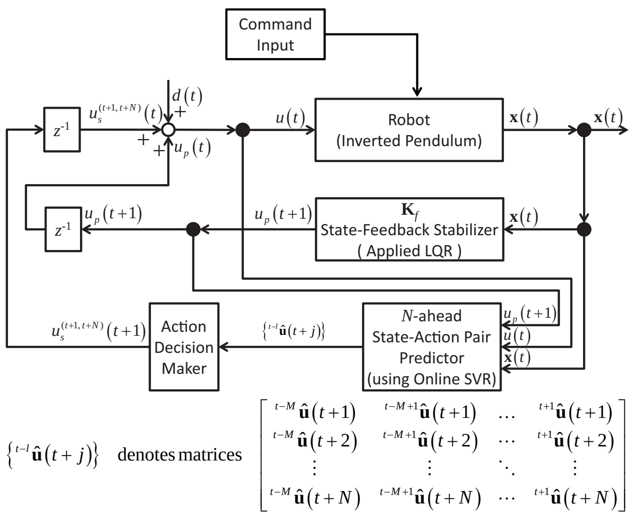



**Figure 1.** Outline for Deciding the Optimal Action for the Robot using the Prediction of State-Action Pair

#### 2.1.1 $N$-ahead state-action pair predictor

As shown in fig. 1, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ describes the prediction result of the control input $\mathbf{u}(t+j)$, when this input is predicted in time $(t-l)$. Hence, the proposed method is attempting to revise the current action, using a combination of the optimal control and the prediction result of the state-action pair. The structure of the prediction of a state-action pair is named "$N$-ahead state-action pair predictor," the internal structure is described in fig. 2 [31].

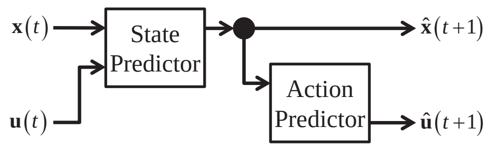



**Figure 2.** Outline of the Prediction System of State and Action [31]

The equations for the state prediction are as follows.

$$\hat{x}_i(t+1) = \begin{cases} 0 & \text{if} \quad t=0 \\ \Delta\theta & \text{if} \quad t=1 \\ \mathbf{k}_{sv}({}^{x}\mathbf{z}(t))^\top (\mathbf{K}_{sv} \\ \quad +\lambda \mathbf{I}_l)^{-1}\, {}^{x}\mathbf{z}_{sv} + b'_i & \text{otherwise} \end{cases} \tag{1}$$

when $i \in \dim \mathbf{x}(t)$

In this paper, the notation is defined as

$$\mathbf{z}(t) = \left[\mathbf{x}(t) \quad | \quad \mathbf{u}(t-1)\right] \tag{2}$$

Here, $b'_i$ is a bias term for $x_i(t)$ (the $i$-th element of $\mathbf{x}(t)$ in time $t$), $\Delta\theta$ represents the Lagrange multiplier, $l$ represents the number of the former support vector $\mathbf{z}_{s_k}(t)(k \in l)$, $\lambda$ represents the regularization parameter, $\mathbf{I}_l$ represents the $l \times l$ identity matrix, $\mathbf{K}_{sv}$ represents the Gramian matrix, and $\mathbf{k}_{sv}$ is the mapping matrix.

Moreover, ${}^{x}\mathbf{z}(t)$ is defined by state $\mathbf{x}(t)$ and the pair $\mathbf{z}(t)$: ${}^{x}\mathbf{z}(t) = \left[\mathbf{z}(t) \quad | \quad \mathbf{x}(t)\right]$.

Next, an action predictor will be generated using the linear-quadratic regulator (LQR). The future action $\hat{\mathbf{u}}(t+1)$ can be predicted using state feedback gain $\mathbf{k}_f$ if it is possible to correctly describe the model of a prediction target as a nonlinear discrete state space model:

$$\hat{\mathbf{u}}(t+1) = \mathbf{k}_f \hat{\mathbf{x}}(t+1) \tag{3}$$

Here, the LQR calculates the feedback gain $\mathbf{k}_f$ in order to minimize the cost function $J\left[\mathbf{x}(t), \mathbf{u}(t)\right] \equiv J$; this is given as

$$J = \int_0^\infty \left(\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right) dt \tag{4}$$

In this equation, $\mathbf{x}^\top$ indicates the transpose of $\mathbf{x}$. The tuning parameter is the weight matrix for state $\mathbf{Q}$ and input $\mathbf{R}$. Thus, $\mathbf{k}_f$ represents a state feedback gain that is given by

$$\mathbf{k}_f = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P} \tag{5}$$

In this equation, $\mathbf{R}$, $\mathbf{B}$, and $\mathbf{P}$ are the parameters of the Riccati differential equation.

$$\mathbf{PA} + \mathbf{A}^\top\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{B}^\top\mathbf{P} + \mathbf{Q} = 0 \tag{6}$$

Thus, the state and action of the robot can be predicted at any given time. The proposed system can be used to predict the next state and action of a robot, and to repeatedly predict its future states and actions.

#### 2.1.2 State-Feedback Stabilizer

As shown in fig. 1, this system will be applied to an optimal control system by using a feedback gain $\mathbf{K}_f$ based on the LQR; in parallel, it will decide the actions that the robot will perform in the future, using the predictions of a state-action pair.

In this system, the optimal feedback gain derived by the LQR is applied. Therefore, the controller was designed based on modern control theory. This LQR calculates the feedback gain $\mathbf{K}_f$ so as to minimize the cost function $J$ given as eq. (4).

### 2.2 How to Obtain and Use the Optimal Action

However, the prediction error must be considered, because the proposed method uses the action obtained from the $N$-ahead state-action pair predictor. If $N$ is larger or more distant than current time $t$, then the prediction error will be proportional to $N$; thus, the piling prediction error must be considered in the prediction (fig. 3).

To address this prediction problem, in [33], fixed-value weighting coefficients for the prediction series of an action were defined. A measure of the importance of each prediction series was also defined, and attempts were made to decrease the influence of the prediction errors of these action prediction series. This paper focuses on variations in the predicted results at any time $(t + j)$, and on determining the "*dependability*" of the predicted value at that time. In other words, variable weight coefficients based on this dependability are defined.

The predictor will be learning and predicting continuously, at each point in time. Therefore, the prediction results will be obtained at each time. A state-action pair prediction that can predict future values at each sampling time is illustrated in fig. 3. Note the "focus" area

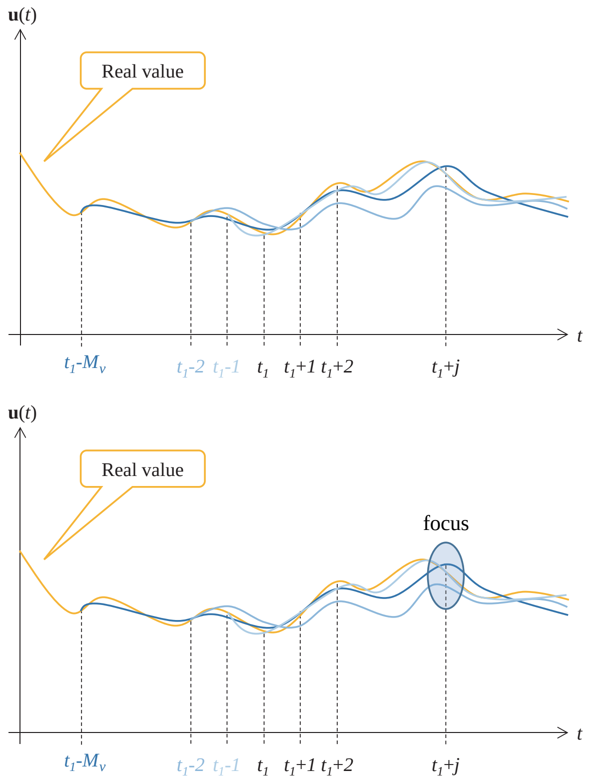



**Figure 3.** Prediction Using a State-Action Pair

in fig. 3. Here, the prediction values at time $(t_1 + j)$ can be obtained, from each previous time $(t_1 - i)$ (the range of $i$ is $(0 \ll M_v \le i \le N)$) as illustrated in fig. 4. In this area, the predicted values are distributed. To quantitatively measure these variations, the standard deviation is focused on. In brief, the standard deviation $\sigma_j$ of the control input in time $(t_1+j)$, using the prediction values that are predicted in

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 79-93
The Society of Digital Information and Wireless Communications, 2015 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

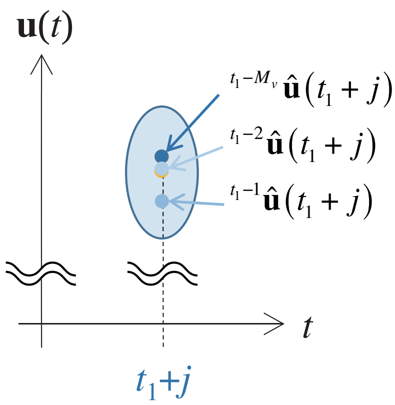



**Figure 4.** Focus on the Variations of Predicted Values from Previous Time Points (From Fig. 3)

each past time $(t_1 - i)$, is derived as follows:

$$\sigma_j = \sigma\left[{}^{t_1-M_v}\hat{\mathbf{u}}(t_1+j), {}^{t_1-M_v+1}\hat{\mathbf{u}}(t_1+j), \ldots, {}^{t_1-i}\hat{\mathbf{u}}(t_1+j), \ldots, {}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right] \quad (7)$$

Based on the above $\sigma_j$, we aim to consider the weight coefficients. In eq. (7), $\sigma[\cdot]$ denotes the standard deviation of $(\cdot)$. Using the characteristics of the standard deviation, a weight coefficient according to the variations of the predicted results is defined, flexibly. Therefore, weight coefficients $\alpha_j$ can be expressed as follows:

$$\alpha_j = C_\sigma \cdot \sigma_j \quad (8)$$

Here, $C_\sigma$ is a coefficient with a small positive real value.

Now, we consider the revising of a current action using predicted future actions. Here, the fact that the influence increases for an action in the near future and inversely decreases for an action in the distant future can be considered. That is, the "decide the optimal action" section from [33] can be referenced as follows:

$$\mathbf{u}_s^{(t+1,t+N)}(t) = \sum_{j=1}^{N} \alpha_j \hat{\mathbf{u}}(t+j) \quad (9)$$

Here, $\alpha_j$ describes the weight coefficients for each $j$ in each $t$. This $\mathbf{u}_s^{(t+1,t+N)}(t)$ is generated from an "Action Decision Maker" and is the revised action used to consider a future action. Now, the prediction results $\mathbf{u}(t)$ and $\hat{\mathbf{u}}(t+j)$ from the $N$-ahead state-action pair predictor's outputs explicitly include the disturbance input $\mathbf{d}(t)$ [33]. Here, if an action can correspond to the disturbance input before the future action is created, the optimal action that is considered in the future can be obtained. In other words, the future action $\mathbf{u}_s^{(t+1,t+N)}(t)$ and the optimal control action $\mathbf{u}_p(t)$ are used as follows:

$$\mathbf{u}(t) = \mathbf{u}_p(t) + \mathbf{u}_s^{(t+1,t+N)}(t) + \mathbf{d}(t) \quad (10)$$

Here, the compensate control input $\mathbf{u}(t)$ is given. The proposed method can obtain a series of actions in time $(t+N)$ in the distant future from current time $t$ using the $N$-ahead state-action pair predictor. Now, on the basis of this prediction series, the current action, which combines "*the action that will be taken in the future*" and the prediction series of the action, can be revised. Namely, the compensate control input combines the current action and the action that takes place in the future. As a result, the compensate control input can compensate for future disturbances by using the $N$-ahead state-action pair predictor. Moreover, the $N$-ahead state-action pair predictor can work for any given sampling time. Therefore, the compensate control input can effectively manage changing disturbances.

This equation will produce accurate results at any given sampling time. The Online SVR used in the state-action pair prediction is the corresponding online learning algorithm. According to this rule, the proposed method will predict and revise the action at each point in time. At these times, the LQR controller will control the robot if the tendency of a disturbance changes. In parallel, the predictors will be learning the control response as a training set. Then, the proposed method can decide an action, again.

# 3 EXPERIMENT – SIMULATION USING THE PROPOSED METHOD FOR CONTROL OF TWO-WHEELED INVERTED PENDULUM

## 3.1 Outline of Experiment

In this study, as an application, the posture of a two-wheeled self-propelled inverted pendulum was stabilized using a computer simulation. Moreover, the command input was sent to the inverted pendulum. In this experiment, it will be confirmed that the proposed method enables a robot to rapidly adapt to a changing environment. Specifically, it will be confirmed that by using learning and predicting continuously, the posture will be continuously stabilized on either a flat floor or an undulating floor (fig. 5). Furthermore, the environment will be changed based on the command input. From this simulation, states and actions were obtained as training samples while postural control was stabilized. To evaluate the proposed system, this study focused on stabilizing the inverted pendulum by using a future prediction based on the prediction of a state-action pair. In this verification experiment, an inverted pendulum "NXTway-GS" (fig. 6) is used as an application example. The control response of the proposed method was compared with the control response of the conventional method. The experiment included more than 400 steps (actual predictive control range was 3.00 [s] to 20.00 [s]). Furthermore, in the proposed method, the predictor only used the proximate predicted result repeatedly with for postural control, 0.05 [s] for each sampling time.

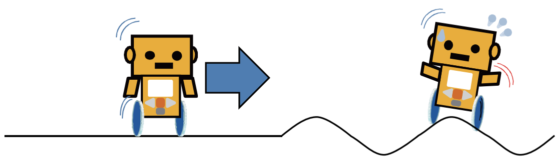
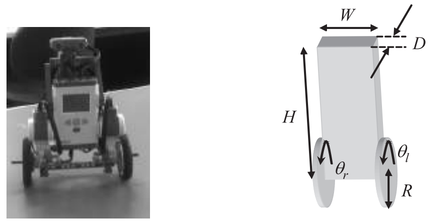


**Figure 5.** Image of the Simulation Environment

## 3.2 Simulation Setup - NXTway-GS Model

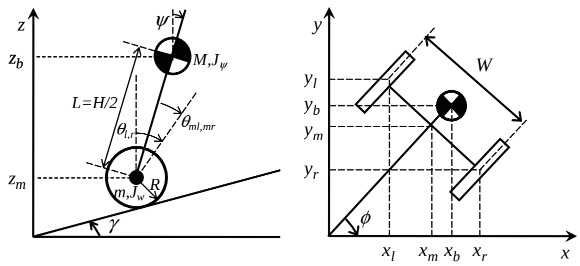
NXTway-GS (fig. 6) can be considered a two-wheeled inverted pendulum model, as shown in fig. 7. Figure 7 shows the side view and the plane view of the model. The coordinate system used in 3.3 is described in fig. 7. Further, in fig. 7, $\psi$ denotes the body pitch angle and $\theta_{l,r}$denotes the wheel angle ($l$ and $r$ indicate *left* and *right*, moreover $\theta = \frac{1}{2}(\theta_l + \theta_r)$), and $\theta_{ml,mr}$ denotes the DC motor angle ($l$ and $r$ indicate *left* and *right*). The physical parameters of the NXTway-GS are listed in table 1.




(a) NXTway-GS

(b) Model of NXTway-GS

**Figure 6.** Two-Wheeled Inverted Pendulum "NXTway-GS"




(a) Side view

(b) Plane view

**Figure 7.** Side View and Plane View of NXTway-GS [34]-[36]

## 3.3 Simulation Setup - Modeling of NXTway-GS

The motion equations of the two-wheeled inverted pendulum model can be derived using the Lagrange equation based on the coordinate system shown in fig. 7. If the direction of the model is the $x$-axis positive direction at $t = 0$, the equations of motion for each coordinate can be given as follows ([34]-[36]):

$$\left[(2m + M) R^2 + 2J_w + 2n^2 J_m\right] \ddot{\theta} + \left(MLR - 2n^2 J_m\right) \ddot{\psi} - Rg\,(M + 2m) \sin\gamma = F_\theta \quad (11)$$

$$\left(MLR - 2n^2 J_m\right) \ddot{\theta} + \left(ML^2 + J_\psi + 2n^2 J_m\right) \ddot{\psi} - MgL\psi = F_\psi \quad (12)$$

$$\left[\frac{1}{2} m W^2 + J_\phi + \frac{W^2}{2R^2}\left(J_w + n^2 J_m\right)\right] \ddot{\phi} = F_\phi \quad (13)$$

Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the state variables and $\mathbf{u}$ is the input variable.

$$\mathbf{x}_1 = \begin{bmatrix} \theta & \psi & \dot{\theta} & \dot{\psi} \end{bmatrix}^\top \quad (14)$$

$$\mathbf{x}_2 = \begin{bmatrix} \phi & \dot{\phi} \end{bmatrix}^\top \quad (15)$$

$$\mathbf{u} = \begin{bmatrix} v_l & v_r \end{bmatrix}^\top \quad (16)$$

Consequently, the state equations of the inverted pendulum model can be derived using eqs. (11), (12), and (13).

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \quad (17)$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \quad (18)$$

In this study, only the state variable $\mathbf{x}_1$ is used. Because $\mathbf{x}_1$ includes the body pitch angles as variables $\psi$ and $\dot{\psi}$, which are important for self-balancing control, the plane motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$) will not be considered.

### 3.4 Simulation Setup - Applying the Online SVR to the State Predictor

In this method, online SVR [32] is used as a learner for state prediction, as shown in fig. 2. Moreover, the RBF kernel [38] is applied as the kernel function to the online SVR of the learner. The RBF kernel of samples $\mathbf{x}$ and $\mathbf{x}'$, which represent feature vectors in some input space, is defined as

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\beta \left|\left|\mathbf{x} - \mathbf{x}'\right|\right|^2\right) \quad (19)$$

Further, the learning parameters of Online SVR are listed in table 2. In table 2, $i \in \{1, 2, 3, 4\}$.

### 3.5 Simulation Setup - Applying the Linear-Quadratic Regulator to the Action Predictor

In this experiment, the LQR is applied as an action predictor, as shown in fig. 2. Therefore, the controller is designed as an action predictor based on modern control theory. This LQR calculates the feedback gain $\mathbf{k}_f$ so as to minimize the cost function $J$ given as eq. (4).

In this study, the following weight matrix $\mathbf{Q}$ and $\mathbf{R}$ was chosen:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 6\times10^5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4\times10^2 \end{bmatrix} \quad (20)$$

$$\mathbf{R} = 1\times10^3 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (21)$$

Then, feedback gain $\mathbf{k}_f$ is obtained by minimizing $J$. Therefore, $\mathbf{k}_f$ is applied as an action predictor [31]. Furthermore, the feedback gain $\mathbf{K}_f$ of the state-feedback stabilizer is applied. Hence, in this experiment, the plane move of the two-wheeled inverted pendulum is not considered. In other words, $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$ were considered.

### 3.6 Conditions of Simulation - Acquiring Training Sets

In this experiment, the command input will be sent to the inverted pendulum. Furthermore, the environment will be changed based on the command input. The shape of the floor will be changed from flat to undulating. Figure 8

**Table 1.** Physical Parameters of NXTway-GS

| Symbol | Value | Unit | Physical property |
|---|---|---|---|
| $g$ | 9.81 | [m/s$^2$] | Gravity acceleration |
| $m$ | 0.03 | [kg] | Wheel weight [34, 35] |
| $R$ | 0.04 | [m] | Wheel radius |
| $J_w$ | $\frac{mR^2}{2}$ | [kgm$^2$] | Wheel inertia moment |
| $M$ | 0.635 | [kg] | Body weight [34, 35] |
| $W$ | 0.14 | [m] | Body width |
| $D$ | 0.04 | [m] | Body depth |
| $H$ | 0.144 | [m] | Body height |
| $L$ | $\frac{H}{2}$ | [m] | Distance of center of mass from wheel axle |
| $J_\psi$ | $\frac{ML^2}{3}$ | [kgm$^2$] | Body pitch inertia moment |
| $J_\phi$ | $\frac{M(W^2+D^2)}{12}$ | [kgm$^2$] | Body yaw inertia moment |
| $J_m$ | $1 \times 10^{-5}$ | [kgm$^2$] | DC motor inertia moment [36] |
| $R_m$ | 6.69 | [Ω] | DC motor resistance [37] |
| $K_b$ | 0.468 | [V·s/rad.] | DC motor back EMF constant [37] |
| $K_t$ | 0.317 | [N·m/A] | DC motor torque constant [37] |
| $n$ | 1 | [1] | Gear ratio [36] |
| $f_m$ | 0.0022 | [1] | Friction coefficient between body and DC motor [36] |
| $f_W$ | 0 | [1] | Friction coefficient between wheel and floor [36] |

**Table 2.** Learning Parameters of Online SVR

| Symbol | Value | Property |
|---|---|---|
| $C_i$ | 300 | Regularization parameter or predictor of $x_i$ |
| $\epsilon_i$ | 0.02 | Error tolerance for predictor of $x_i$ |
| $\beta_i$ | 30 | Kernel parameter for predictor of $x_i$ |

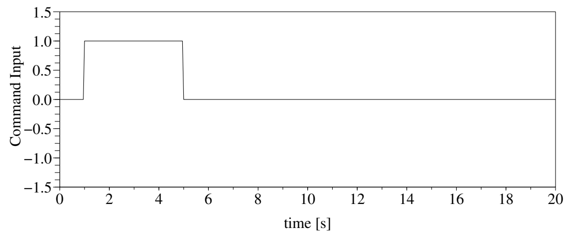



**Figure 8.** Command Input Signals (+1 Indicates Forward Run, 0 Indicates Stationary Balancing)

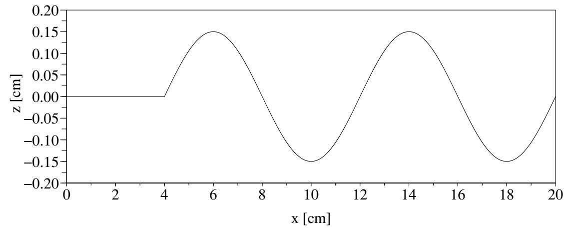



**Figure 9.** Simulation Environment (the Shape of the Floor)

shows the command input, and fig. 9 shows the shape of the floor. The shape of the floor is given by the following equation.

$$z = 0.15U(x_f)\sin[2 \times 0.125 \cdot \pi \cdot (x_f)] \quad \text{[cm]} \quad (22)$$
$$x_f = x - 4 \quad \text{[cm]}$$

In this equation, $U(x)$ indicates the unit step function, $x$ [cm] indicates the length of the floor, and $z$ [cm] indicates the undulating height of the floor.

Using the settings mentioned above, we attempted to drive the inverted pendulum model on the floor by using the command input (fig. 10). Thus, the training sets from the two-

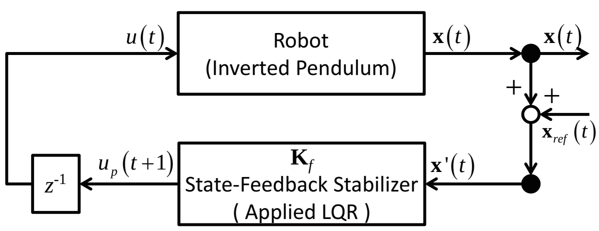



**Figure 10.** Control Input Obtained by Mixing the Action and Command Inputs

wheeled inverted pendulum can be acquired. Figures 11 through 15 show training sets that were obtained from the computer simulation of the stabilizing control of the two-wheeled inverted pendulum, and the move forward instructions sent from the command input within the control (as shown in fig.8).

Moreover, figs. 16 and 17 show the position of the mass of the inverted pendulum. Here, movement distance for displaying the position is given by the following equation.

$$x = 100 \cdot R \cdot \int \dot{\theta}(t)dt \quad \text{[cm]} \tag{23}$$

The position of the mass of inverted pendulum $z_m$ is given in the following equation.

$$\begin{aligned} z_m &= 100 \cdot \left[ R + R\sin\gamma \cdot \int \dot{\theta}(t)dt \right. \\ &\quad \left. + L\cos\left\{ \int \dot{\psi}(t)dt \right\} \right] \quad \text{[cm]} \end{aligned} \tag{24}$$

In this experiment, the two-wheeled inverted pendulum model performs stationary balancing or moves forward based on the command input. The disturbance properties that we provide as input, and other conditions of the simulation, are listed in table 3.

**Table 3.** Simulation Parameters

| Symbol | Value | Unit | Physical property |
|---|---|---|---|
| $\psi_0$ | 0.0262 | [rad.] | Initial value of body pitch angle |
| $\gamma_0$ | 0.0 | [rad.] | Slope angle of movement direction |
| $t_s$ | 0.05 | [s] | Sampling rate |
| $N_s$ | 60 | — | Initial dataset length |
| $N_{\max}$ | 401 | — | Maximum dataset length for the prediction |
| $N$ | 20 | — | Step size of outputs for $N$-ahead state-action pair predictor's outputs |
| $C_\sigma$ | 0.05 | — | Coefficient for the standard deviation of the predicted values |
| $N_\sigma$ | 10 | — | Calculation range of the standard deviation of the predicted values |

## 3.7 Simulation Results

In this simulation, the input data provided to the NXTway-GS is based on predicted results

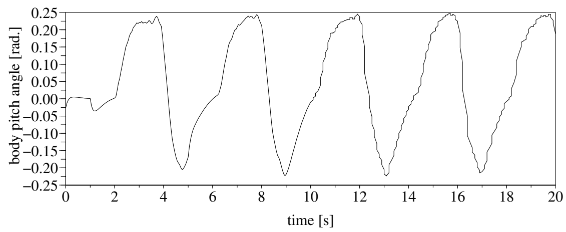



**Figure 11.** Training Set of Control Response of Body Pitch Angle $\psi$ using Only LQR

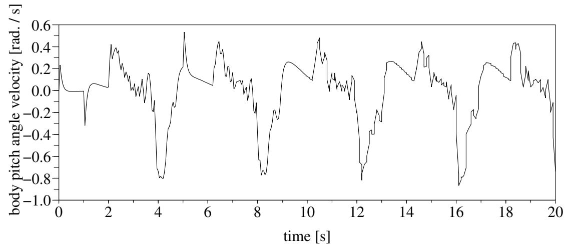



**Figure 12.** Training Set of Control Response of Body Pitch Angle Velocity $\dot{\psi}$ using Only LQR

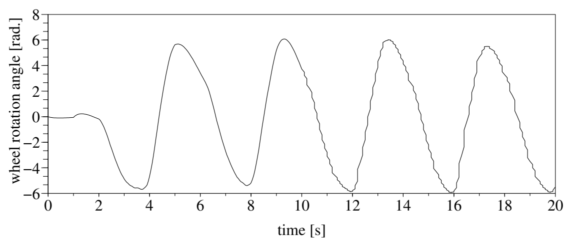



**Figure 13.** Training Set of Control Response of Wheel Rotation Angle $\theta$ using Only LQR

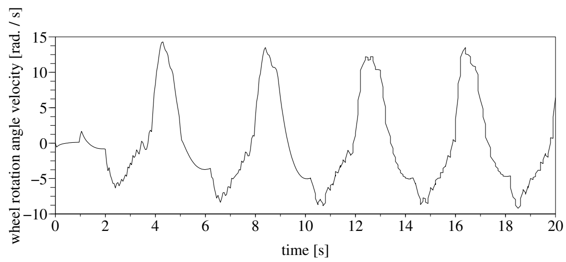



**Figure 14.** Training Set of the Control Response of Wheel Rotation Angle Velocity $\dot{\theta}$ using Only LQR

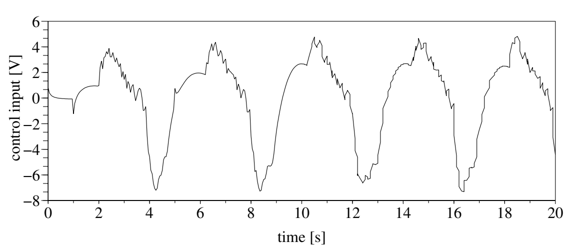



**Figure 15.** Training Set of the Control Response of Control Input $u$ using Only LQR

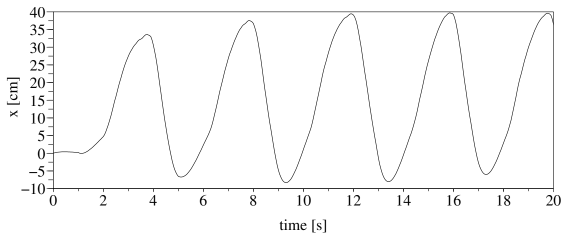



**Figure 16.** Position of the Inverted Pendulum on the $x$-axis using Only LQR

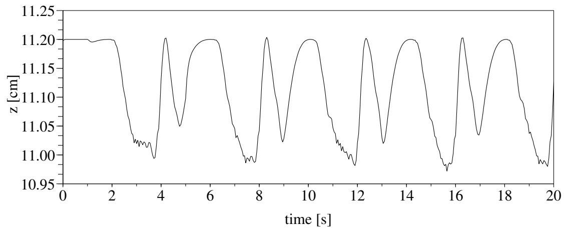



**Figure 17.** Position of the Mass of the Inverted Pendulum on the $z$-axis using Only LQR

generated by the proposed method. In addition, the NXTway-GS model performs stationary balancing based on that input, and thereby moves backward or forward.

Figures 18 through 23 show the compensation results of state $\mathbf{x}_1$, and fig. 25 shows the compensation results of control input $\mathbf{u}$. In this section, the part of the graph that was obtained from actual training sets are not considered. Thus, only those parts of the graph pertaining to the state prediction part shown in $T$ (at $t = 3.00$ [s]) of figs. 18 to 23 will be analyzed. Moreover, figs. 26 and 27 show the position of the mass of the inverted pendulum. The graph in fig. 28 shows enlargement around the $z$ position of the mass of the inverted pendulum in fig. 27. Here, the movement distance used for displaying the trajectory was obtained by eqs. (23) and (24).

### 3.8 Discussion on Simulated Results of the Proposed Method

Here, the starting and predicting of the state prediction point are shown at $t = 3.00$ [s]. Therefore, we will only focus on the portion of the graph pertaining to the state prediction portion shown in $T$.

According to these results (figs. 18, 20, and 23), compensation results obtained using

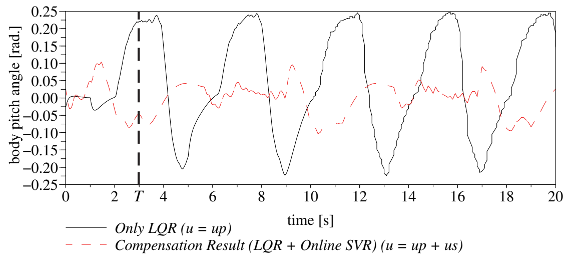



**Figure 18.** Control Response of Body Pitch Angle $\psi$ using the Proposed Method

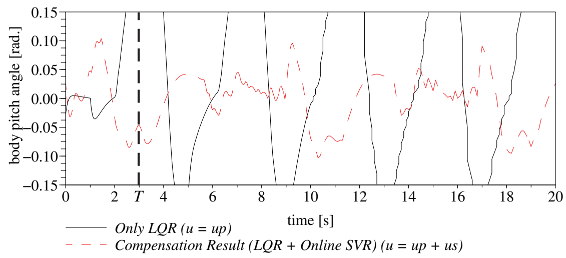



**Figure 19.** Control Response of Body Pitch Angle $\psi$ using the Proposed Method focused around $\psi(t) = 0$

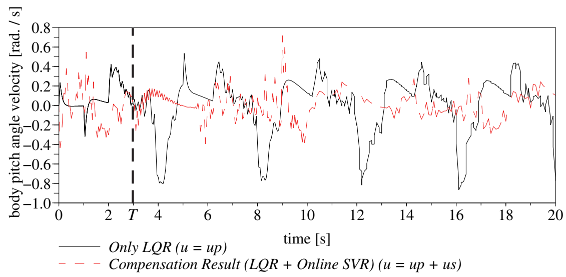



**Figure 20.** Control Response of Wheel Rotation Angle Velocity $\dot{\psi}$ using the Proposed Method

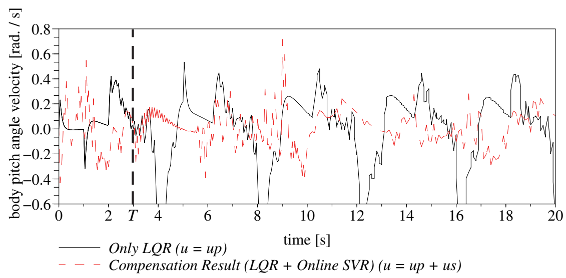



**Figure 21.** Control Response of Wheel Rotation Angle Velocity $\dot{\psi}$ using the Proposed Method focused around $\dot{\psi}(t) = 0$

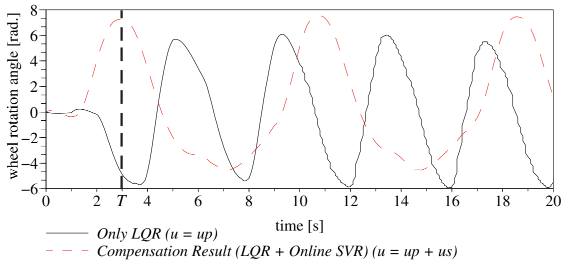



**Figure 22.** Control Response of Wheel Rotation Angle $\theta$ using the Proposed Method

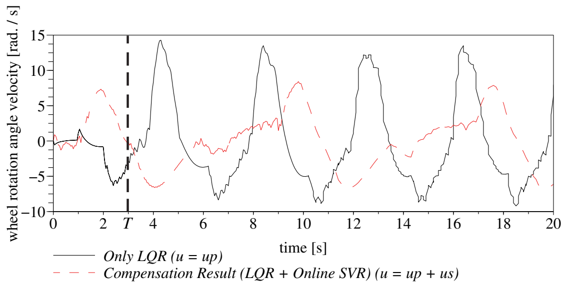



**Figure 23.** Control Response of Wheel Rotation Angle Velocity $\dot{\theta}$ using the Proposed Method

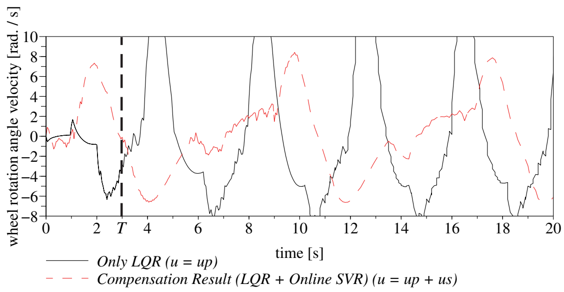



**Figure 24.** Control Response of Wheel Rotation Angle Velocity $\dot{\theta}$ using the Proposed Method focused around $\dot{\theta}(t) = 0$

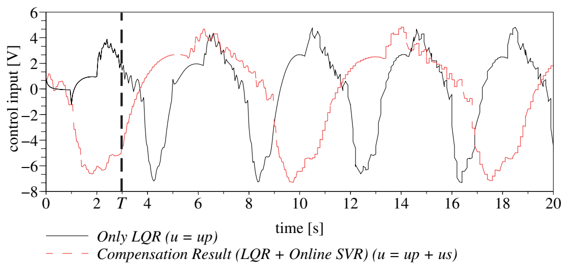



**Figure 25.** Control Response of the Control Input $u$ using the Proposed Method

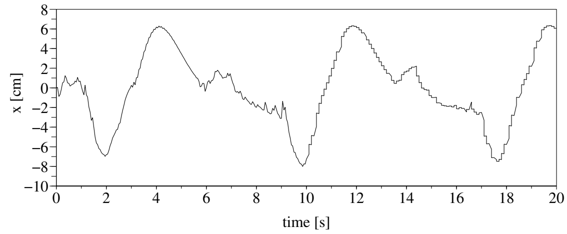



**Figure 26.** Position of the Inverted Pendulum on the $x$-axis using the Proposed Method

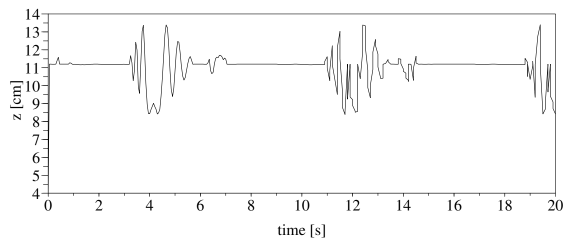



**Figure 27.** Position of the Mass of the Inverted Pendulum on the $z$-axis using the Proposed Method

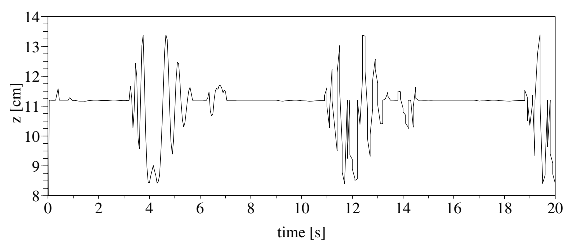



**Figure 28.** Position of the Mass of the Inverted Pendulum on the $z$-axis using the Proposed Method focused around $z = R + L$ [cm] in Figure 27

the proposed method (shown as the red solid line) approach or converge to near zero with time. Moreover, according to wheel rotation angle $\theta(t)$ in fig. 22, the compensation results obtained using the proposed method indicate continuous backward and forward driving. In addition, according to control input $u(t)$ in fig. 25, the compensation results obtained using the proposed method are continuously generating a control input. Next, in fig. 18, the result of the proposed method almost converges to zero. Next, each result will be analyzed. From fig. 22, $\theta(t)$ swings considerably and can be confirmed. Therefore, it can be said that the wheel is moving while attempting to decrease the body pitch angle $\psi(t)$. In figs. 20 and 23, these results are almost becoming smaller than those of the other state in advance of the conventional method that uses only LQR. Finally, in fig. 25, this result is also observed in more advanced states in the proposed method, compared to the conventional method that uses only LQR. This is occurring because the compensate control input is combining the current action and future actions in advance. In this case, for future actions, the compensation control input will be using state-action pair prediction results directly. Therefore, the compensate control input generates actions that consider future disturbances. As a result, it was confirmed that the proposed method is robust against the undulated floor. Furthermore, the effect of the disturbance will be reduced and the system will be converge to the desirable state.

Next, the movement distance will be focused on. In figure 26, it can be confirmed that the inverted pendulum is moving forward because it received the forward run from the command input. In parallel, the inverted pendulum is also moving backward. From this command input, the inverted pendulum approaches the undulating floor. From this result, it can be said that the inverted pendulum moved autonomously based on prediction results, and was able to balance itself. Therefore, body pitch angle was near zero. Subsequently, the trajectory of the mass of the inverted pendulum will be analyzed. In figure 27, it cannot be confirmed that the body is pitching around the balance point. Accordingly, figure 28 will be focused on. In figure 28, it can be confirmed that the mass of the inverted pendulum is only slightly pitched around the 0 [deg.] balance point, compared to the method that uses only LQR (fig. 17). These results also show that the inverted pendulum moved autonomously based on prediction results, and was able to balance itself. Based on these results, it can be confirmed that the body pitch angle is near zero.

Therefore, along with the action predictor, the LQR feedback controller was applied, and this controller maintained the “desired” stable state. In other words, this system stabilizes the inverted pendulum using the current outside data, previous states, and an action. As a result, it can be said that the robot’s states will converge to a stable state, according to the time course. Furthermore, this system acquires data at each of the sampling times. Using these results, the proposed system derives an action that multiplies states with the optimal feedback gain for obtaining the future state. Thus, the predictors will predict in the direction of stable states, even if there are some disturbances in the environment such as a flat floor or undulated floor. In other words, this proposed method is robust against disturbances that directly affect the application. From these viewpoints, it can be concluded that the experimental results are reasonable.

## 4 CONCLUSION

In this study, we focused on the relationship between the state and the action of robots. Therefore, on the basis of our former study, we proposed a method that decides an action that the robot will take using the recent tendency of the prediction results for any given time. Moreover, we applied the LQR to derive an action using the optimal feedback gain, and defined the weight coefficient using the standard deviation as in the proposed method. Using the proposed method, we obtained the compensated current action for greater convergence in the desirable state. Further, the command input was sent to the inverted pendulum. From this command input, the shape of the floor was changed from flat to undulating. As a result, we confirmed that the inverted pendulum moved autonomously based on prediction results, and could balance itself.

On the basis of the experimental results, the proposed method could be converged to a desirable state as the optimal solution of this problem using prediction and selection. In other words, the proposed methods can adjust the transition of a robot's state or outside environment. Accordingly, we can conclude that the proposed method can predict using online SVR and LQR, and decide an action for the future.

In future work, the calculation range of the standard deviations of the predicted values $N_\sigma$ must be investigated. In this study, $N_\sigma$ was used as a fixed value. However, at first, $N_\sigma$ must be defined by taking the results of the state-action pair prediction, combinations of the hyper parameters of online SVR, and prediction errors into consideration. To confirm effectiveness, we will aim to introduce various floor shapes, and reconfirm the behavior of the proposed method.

## REFERENCES

[1] Sebastian Thrun, Wolfram Burgard, and Dieter Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*, The MIT Press, 2005.

[2] Shun'ichi Asaka and Shigeki Ishikawa, "Behavior Control of an Autonomous Mobile Robot in Dynamically Changing Environment," Journal of the Robotics Society of Japan, vol.12, no.4, pp.583-589, 1994.

[3] Takayuki Kanda, Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, and Ryohei Nakatsu, "Development of "Robovie" as Platform of Everyday-Robot Research," IEICE Transactions on Information and Systems, Pt.1 (Japanese Edition), vol.J85-D-1, no.4, pp.380-389, 2002.

[4] Denis F. Wolf and Gaurav S. Sukhatme, "Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments," Autonomous Robots vol.19, pp.53-65, Springer, Netherlands, 2005.

[5] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," Journal of Artificial Intelligence Research vol.11, pp.391-427, 1999.

[6] Mohammad Abdel Kareem Jaradata, Mohammad Al-Rousanb, and Lara Quadanb, "Reinforcement based Mobile Robot Navigation in Dynamic Environment," Robotics and Computer-Integrated Manufacturing vol.27, pp.135-149, 2011.

[7] Ellips Masehian and Yalda Katebi, "Sensor-Based Motion Planning of Wheeled Mobile Robots in Unknown Dynamic Environments," Journal of Intelligent & Robotic Systems, DOI:10.1007/s10846-013-9837-3, 2013.

[8] Mohammed Faisal, Ramdane Hedjar, Mansour Al Sulaiman, and Khalid Al-Mutib, "Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment," International Journal of Advanced Robotic Systems, DOI: 10.5772/54427, 2012.

[9] Fabrizio Abrate, Basilio Bona, Marina Indri, Stefano Rosa, and Federico Tibaldi, "Multi-robot Map Updating in Dynamic Environments," Distributed Autonomous Robotic Systems Springer Tracts in Advanced Robotics vol.83, pp.147-160, 2013.

[10] International Federation of Robotics, *All-Time-High for Industrial Robots: Substantial Increase of Industrial Robot Installations is Continuing*, 2011.

[11] Takushi Sogo, Katsumi Kimoto, Hiroshi Ishiguro, and Toru Ishida, "Mobile Robot Navigation by a Distributed Vision System," Journal of the Robotics Society of Japan, vol.17, no.7, pp.1-7, 1999.

[12] Jong Jin Park, Collin Johnson, and Benjamin Kuipers, "Robot Navigation with MPEPC in Dynamic and Uncertain Environments: From Theory to Practice," IROS 2012 Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs, 2012.

[13] Edvard Naerum, H. Hawkeye King, and Blake Hannaford, "Robustness of the Unscented Kalman Filter for State and Parameter Estimation in an Elastic Transmission," In Proceedings of the Robotics: Science and Systems, 2009.

[14] R. S. Sutton, A. G. Barto, *Reinforcement Learning*, The MIT Press, 1998.

[15] Minoru Asada, Shoichi Noda, Sukoya Tawaratsumida, and Koh Hosoda, "Purposive Behavior Acquisition for a Robot by Vision-Based Reinforcement Learning, " Journal of the Robotics Society of Japan, vol.13, no.1, pp.68-74, 1995.

[16] Norikazu Sugimoto, Kazuyuki Samejima, Kenji Doya, and Mitsuo Kawato, "Reinforcement Learning and Goal Estimation by Multiple Forward and Reward Models," IEICE Transactions on Information and Systems, Pt.2 (Japanese Edition), vol.J87-D-2, no.2, pp.683-694, 2004.

[17] Yasutake Takahashi and Minoru Asada, "Incremental State Space Segmentation for Behavior Learning by Real Robot," Journal of the Robotics Society of Japan, vol.17, no.1, pp.118-124, 1999.

[18] Jongho Shin, H. Jin Kim, Sewook Park, and Youdan Kim, "Model Predictive Flight Control using Adaptive Support Vector Regression," Neurocomputing, vol.73, no.4-6, pp.1031-1037, 2010.

[19] Younggeun Choi, Shin-Young Cheong, and Nicolas Schweighofer, "Local Online Support Vector Regression for Learning Control," In Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation Jacksonville, FL, USA, pp.13-18, 2007.

[20] Erik Schuitema, Lucian Busoniu, Robert Babuska, and Pieter Jonker, "Control Delay in Reinforcement Learning for Real-Time Dynamic Systems: A Memoryless Approach," In Proceedings of Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference, pp.3226-3231, IEEE, 2010.

[21] Thomas J. Walsh, Ali Nouri, Lihong Li, and Micheal L. Littman, "Planning and Learning in Environments with Delayed Feedback," Machine Learning: ECML 2007, pp.442-453, 2007.

[22] Yang Su, Kok Kiong Tan, and Tong Heng Lee, "Computation Delay Compensation for Real Time Implementation of Robust Model Predictive Control," In Proceedings of Industrial Informatics (INDIN), 2012 10th IEEE International Conference, pp.242-247, IEEE, 2012.

[23] Cunjia Liu, Wen-Hua Chen, and John Andrews, "Model Predictive Control for Autonomous Helicopters with Computational Delay," In Proceedings of Control 2010, UKACC International Conference, pp.1-6, IET, 2010.

[24] Giancarlo Marafioti, Sorin Olaru, and Morten Hovd, "State Estimation in Nonlinear Model Predictive Control, Unscented Kalman Filter Advantages," Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences vol.384, pp.305-313, 2009.

[25] Niko Sünderhauf, Sven Lange, and Peter Protzel, "Using the Unscented Kalman Filter in Mono-SLAM with Inverse Depth Parametrization for Autonomous Airship Control," In Proceedings of IEEE International Workshop on SSRR 2007, pp.1-6, 2007.

[26] Mohammad Ali Badamchizadeh, Iraj Hassanzadeh, and Mehdi Abedinpour Fallah, "Extended and Unscented Kalman Filtering Applied to a Flexible-Joint Robot with Jerk Estimation," Discrete Dynamics in Nature and Society, vol. 2010, Article ID 482972, 2010.

[27] J. G. Iossaqui, J. F. Camino, and D. E. Zampieri, "Slip Estimation Using The Unscented Kalman Filter for The Tracking Control of Mobile Robots," In Proceeding of the International Congress of Mechanical Engineering - COBEM, pp.1-10, 2011.

[28] Ramazan Havangi, Mohammad Ali Nekoui, and Mohammad Teshnehlab, "Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, pp.15-23, 2010.

[29] Foudil Abdessemed, "SVM-Based Control System for a Robot Manipulator, " Int J Adv Robotic Sy, 2012, vol. 9, 247, DOI: 10.5772/51192, 2012.

[30] Masashi Sugimoto and Kentarou Kurashige, "The Proposal for Prediction of Internal Robot State Based on Internal State and Action," In Proceedings of IWACIII2013 CD-ROM, SS1-2, Oct.18-21, Shanghai, China, 2013.

[31] Masashi Sugimoto and Kentarou Kurashige, "The Proposal for Deciding Effective Action using Prediction of Internal Robot State Based on Internal State and Action," In Proceedings of 2013 International Symposium on Micro-NanoMechatronics and Human Science, pp.221-226, Nov.10-13, Nagoya, Japan, 2013.

[32] Francesco Parrella, Online Support Vector Regression. PhD thesis, Department of Information Science, University of Genoa, Italy, 2007.

[33] Masashi Sugimoto and Kentarou Kurashige, "Real-time Sequentially Decision for Optimal Action using Prediction of the State-Action Pair," In Proceedings of 2014 International Symposium

on Micro-NanoMechatronics and Human Science, pp.199-204, Nov.9-12, Nagoya, Japan, 2014.

[34] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao Ohmura, "A Study on Model-Based Development of Embedded System using Scilab/Scicos," In Proceedings of the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp.343-344, 2010.

[35] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao Ohmura, "A Study on Model-Based Development of Embedded System using Scilab/Scicos – Development of Auto-Code Generator –," In Proceedings of the 2010 JSME Conference on Robotics and Mechatronics (ROBOMEC'10), Asahikawa, vol.10, no.4, 2A2-C10, 2010.

[36] Yorihisa Yamamoto, NXTway-GS Model-Based Design –Control of Self-Balancing Two-Wheeled Robot Built with LEGO Mindstorms NXT–. Cybernet Systems Co., Ltd., 2009.

[37] Ryo Watanabe, Ryo's Holiday LEGO Mindstorms NXT, 2008.

[38] Yin-Wen Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin, "Training and Testing Low-Degree Polynomial Data Mappings Via Linear SVM," J. Machine Learning Research, vol.11, pp.1471-1490, 2010.

[39] John Devcic, "Weighted Moving Averages: The Basics," 2006. [Online]. Available: http://www.investopedia.com/articles/technical/060401.asp

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 94-101
The Society of Digital Information and Wireless Communications, 2015 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

# A Parallel Load Balancing Based on Pseudo-Cliques

Liviu Octavian Mafteiu-Scai
West University of Timisoara,
Timisoara, Romania
lscai@info.uvt.ro

## ABSTRACT

In solving systems of equations on parallel computers, one very important problem is load balancing, because this affects the process efficiency. One way of achieving this goal is a good partitioning of the equations system. In this work, a connection between partitioning and pseudo-cliques from graph theory is shown. A preconditioning that uses pseudo-cliques balancing is also proposed.

## KEYWORDS

system equations, parallel, partitioning, preconditioning, pseudo-cliques, iterative methods

## 1 INTRODUCTION

A lot of real applications involves solving large sparse linear systems of equations and, because in the current real problems these systems have dimensions of millions [2], parallel computing is one way to increase performance.

In the case of small systems, direct methods like Gauss are good choices but these are prohibitive for medium and large systems. A good choice in last case are iterative methods like Newton, Conjugate Gradient, Broyden, etc. But, these methods are more dependent on the properties of the systems. Furthermore, the iterative methods require systems preconditioning to obtain a good approximation of the solution in a reasonable time. From a global perspective ie preconditioning and effective solving, we consider that a preconditioner is good only if it improves the convergence of the iterative method sufficiently to overcome the extra cost of applying it. The importance of preconditioning becomes even more important in the case of parallel solving and there are numerous studies that show the importance of preconditioning for solving linear equations systems on parallel computers [1, 2].

One of the most used iterative methods is the Conjugate Gradient (CG) [3], being seen as a special case of Gaussian elimination. CG method is very effective when the associated matrix of equations system is symmetric and positive definite, but there are versions of CG for nonsymetric case. The CG method involves small errors and exact solutions are generally obtained after at most $n$ steps in the case of a well conditioned system, where $n$ is the size of the equations system. But, if there is a rapid convergence for well conditioned systems, this can be arbitrary if the matrix is ill conditioned. Moreover, in Krylov type iterative methods, the convergence decreases or is lost after parallelization of these methods -compared with serial variants of these-, the least affected seems to be the CG, as shown in [4].

An important factor in parallel solving of equations systems is the load balancing of processors. In general, there is a good load balancing of the processors if each of them has roughly the same amount of work to perform. If for each partition in part coresponds an equations subsystem, the load balancing can be seen from three perspectives: the size of the subsystems, the number of nonzero elements in each subsystem in part and the difficulty of solving these subsystems (condition number of the associated matrix). The first can be easily controlled and does not involve a considerable computational effort. The second can be controlled in most cases but sometimes a computational effort is required: preconditioning techniques like bandwidth or average bandwidth reduction [5]. The third -and most important in our opinion- unfortunately it can not be practical considered due to large computational effort required ie computing the condition number for each subsystem in part. It is well known that the condition number has a major influence in terms of convergence speed, therefore the execution time [4,5,6]: in the case of parallel solving, the condition number of each diagonal submatrix in part, influence the local convergence for each subprocess in part and implicitly the global convergence. In paper [5] other factors that influence the performances of parallel solving equations systems with iterative methods were experimentally identified: a distribution of non-zero elements along and around the main diagonal of the associated matrix, the ratio between the number of nonzero elements inside and outside of Jacobian sub-matrix for each partition in part and of course the number of partitions.

In CG method the parallelism is derived mainly from parallel matrix-vector product and this represents the main computational effort. Because this product is directly proportional to the number of nonzero values, an implementation of a load balanced distribution can be viewed as an uniformly non-zero elements distribution between processors. That is, the nonzero elements must be uniform distributed along and around the main diagonal: this should to be as uniform as to ensure a more balanced load of processors. At the same time, the syncronization between processors can be viewed as a result of a load balancing. For example, in parallel conjugate gradient the updating of the residual vector and the vector solution do not depend on each other and can be performed at the same time, but these operations cannot be performed before performing the matrix-vector product. But the matrix-vector product in a new iteration can't be performed until the residual vector is updated. Thus, there are two moments in which processors must synchronize before they can move on to the next iteration. It is very important that the work to be balanced between processors such synchronization moments so the processors do not have any periods of inactivity between these two moments (ideally) or these periods to be the smallest possible. Thus, minimizing this waiting/idle time is an important goal in parallel solving systems of equations. In conclusion, the problems to be solved in parallel solving systems of equations are related to finding a suitable division of the processes to be performed, as well as finding the most appropriate mechanisms for synchronization and communication between different processes.

Concepts from graph theory are often used in theoretical formalization of parallelization issues [7,8,9,10,11,12]. In our approach the basic concept used from graph theory is *clique*. So, there are known definitions for a *graph* G and a *subgraph* S:

$$G(X,U) = \{X = \{x_1, x_2, \dots . x_n\}, U = \{(x_i, x_j), x_{i,j} \in X\}\}$$

$$S(X', U') = \{ X' \subset X, U' \subset U\}$$

where $x_i$ are <u>*vertices*</u>, and the pairs $(x_i,x_j)$ are *edges* that connect two vertices. If $(x_i,x_j)=(x_j,x_i)$ we have an undirected graph.

*Clique* is an old term [13] and is one of the basic concepts of graph theory. A clique is a *complete subgraph* of an undirected graph, ie every two distinct vertices in the clique are adjacent:

$$C(X', U') = \{X' = \{x_1, x_2, \dots . x_k\}, X' \subset X, U' = \{\forall i, j \; \exists (x_i, x_j), x_{i,j} \in X'\}, U' \subset U\}$$

The finding cliques problem is NP-complete [14]. Because cliques are used in a great number of theoretical and real problems, many algorithms have been developed. In particular, some issues present a high interest:

- *maximal clique:* a clique that cannot be extended by including one more adjacent vertex;
- *maximum clique:* a clique, such that there is no others cliques with more vertices;
- *clique number:* represents the number of vertices in a maximum clique;
- *intersection number:* the smallest number of cliques that together cover all the graph's edges.

Cliques are considered as a part of dense graphs and can be considered that they do not have much practical effect in the case of sparse structures. For these cases, the *pseudo-cliques* or *quasi-cliques* concept was proposed. Mainly, there are two models to explain pseudo-cliques [15]. First, the pseudo-clique is a subgraph that is obtained from a clique by removing a constant number of clique's edges. From a second point of view a pseudo-clique is a subgraph that has at least a constant ratio of edges compared to a clique of the same size. Note that this second perspective is considered in our paper. Also, the pseudo-clique problem is NP-complete.

## 2 THE PROPOSED APPROACH

The partitioning in terms of graph theory is an old problem in parallel computing, a relevant paper that shows partitioning models being [16]. In this paper we propose a partioning model of parallel solving equations system based on pseudo-cliques concept.

This paper will show the connection between cliques/pseudo-cliques and partitioning in parallel solving of equations systems. In our approach it is considered the case of parallel solving of an equations system using Kyrlov iterative methods that using the Jacobian, a synchronous model of parallelization and equal (or about equal) row-blocks partitioning. Also we take into consideration the undirected and unweighted graph corresponding for associated matrix of equations system.

From matriceal point of view, in an ideal case all nonzero elements should be distributed inside of diagonal blocks /submatrices. In the case of equal row-blocks partitioning, partitions with same number of nonzeros represent independent and equal subsystems of equations. In this case we have a minimum of communication processes ie a better load balancing is achieved. From the graph point of view, this situation corresponds to a graph that has cliques with same sizes and all edges of associated graph are covered by these. Of course, another ideal case in terms of communication processes is that in which the graph can be divided completely in cliques, but not necessarily of equal sizes.

Because such ideal situations are difficult or impossible to obtain in practice, we consider that a reasonable solution is when inside of diagonal blocks $Ji(x)$, $i = 1, ..., p$ ($p$ is the number of partitions) of the Jacobian $J(x)$ to be as many nonzero values to ensure a smaller number of communication processes. In addition, it is desirable that blocks $Ji(x)$ to contain close values of the number of nonzeros to ensure a good processors balancing. It is obvious that such a situation corresponds to a division of the associated graph in pseudo-cliques with about equal sizes. From associated matrix point of view, a relabeling in graph according to pseudo-cliques can conduct to an uniform distribution of nonzeros along the main diagonal. That is, that situation corresponds to close values of nonzeros in diagonal blocks of Jacobian, a situation that is reasonable and can be obtained in practice.

Further, we will introduce some new terms which are useful in pseudo-cliques evaluation in our approach.

**Definition 1.** An edge is <u>internal</u> if their adjacent vertices are in same pseudo-clique.

**Definition 2**. An edge is <u>external</u> if their corresponding vertices are in different pseudo-cliques.

Something similar to external edges are the c*ritical edges* or *bridges*, edges that interconnect two partitions into a graph.

**Definition 3.** It is said that a set of pseudo-cliques <u>cover the graph</u> if all the vertices of the graph are contained in these pseudo-cliques.

It is obvious that a larger ratio between internal and external edges for all pseudo-cliques of an covered graph involves a smaller number of communication processes -relative to equations system's size- and that is better for parallelization. More, the values for internal and external edges for each pseudo-clique in part must be close, a fact that ensures a good balancing between processors. So, we can appreciate that the internal edges represent the equations subsystem processing and external edges represents the communication processes.

**Definition 4.** Degree of filling (DF) is the ratio between number of internal edges of a pseudo-clique and number of edges of a clique containing the same vertices.

The computing relation is:

$$DF = \frac{k}{\frac{v(v-1)}{2}}100 \qquad (1)$$

for a pseudo-clique with $v$ vertices and $k$ internal edges. This value is expressed as a percentage and a value of this equal with 100% shows a clique. It is obvious that this particular case is an ideal situation, but, in practice we are satisfied with values above 60-80%.

Further, a simplified example for a better understanding of our approach is shown. Consider G($X$,$U$) the associated graph of a 10x10 equations system where X={1,2,3,4,5,6,7,8,9,10} and U={(1,2),(1,3),(1,4),(2,5),(5,6),(5,7),(6,7),(8,9),(9,10),(8,10)}. In Figure 1 are represented three ways of partitioning.

As it can be seen in Figure 1, it is obvious that a good partitioning implies that the total number of external edges of the associated graph to be as small. More, it implies that the pseudo-cliques obtained by partitioning the associated graph to be approximately equal in terms of the number of vertices (in fact unknowns of the equations system). In conclusion, a bad balancing in terms of the size of equations subsystems assigned to each processor (pseudo-cliques) and a large number of communication processes (external edges) may lead to large downtime for some of the processors involved in computing ie a low efficiency of the entire parallel process.

The example from Figure 1 is relevant for the proposed approach. Thus, in a) we have a large number of processors involved in the computation (4) but there are three external edges. In b) we have the ideal case in terms of the number of external edges (0) but we have a disequilibrium in terms of equations subsystems sizes. The case from c) represents a compromise between case a) and b), with an acceptable balancing and a small number of communications.

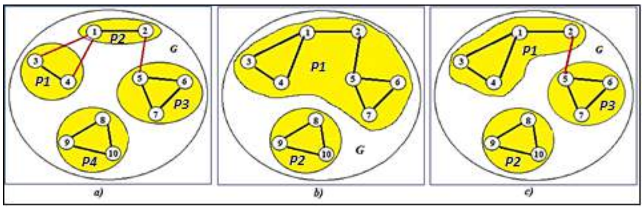


Figure 1 Examples of possible partitioning

**Notes:**

a) it should be noted that in our approach were not taking into account other factors that influence the computational effort per processor such as the condition number of each equations subsystem in part;

b) the partitioning used was a standard approach [16], ie each external edge causes one external communication process. In fact, the processor *p1* that has allocated partition *P1*, does not communicate twice to the processor *p2* (with the partition allocated *P2*) the value from vertex 1. So we have a single process of communication, not two. Consequently, from the communication processes point of view, we have a single external edge between *P1* and *P2*, thus the new graph obtained is a hipergraph, a much more realistic representation as shown in paper [17]. Such an approach will be the subject of our future research;

c) the partitioning must be treated differently, depending on the parallelization model: synchronous or asynchronous;

So, how can we practically use the pseudo-clique concept to optimize the parallelization process? Before showing this, we must mention that there are two perspectives in terms of partitioning. First, for a given number of processors/partitions the equations system is ”arranged” so as to ensure an enhanced efficiency of parallelization process. The second approach consists of determining the best partitioning or one as close/satisfactory for a given equations system.

The main idea in the first approach consists of determining a number of cliques (ideal case) or pseudo-cliques (frequent case) equal to the number of partitions desired and rearranging the equations system using a relabelling of associated graph’s vertices. A load balancing of processors and a small number of communication between them will constitute the main goals of algorithms proposed for this issue.

In the second approach the proposed algorithm searches the best division of the associated graph in pseudo-cliques that ensure an efficient parallelization. This means pseudo-cliques with *sizes degree of filling* and *number of external edges* close as values. More, it will be chosen the partitioning where the pseudo-cliques have a larger value for *degree of filling* and a smaller value of *external edges.*

## 3 EXPERIMENTS

In the following, the experiments performed that validate the proposed approach are presented.

Finding all cliques of an undirected graph is a hard task in terms of memory and runtime because the number of cliques can grow exponentially with every node added. There are many algorithms proposed for this problem but that proposed by Bron & Kerbosch [18] seems to be one of the fastest. An adapted form of this was used in our experiments, ie an algorithm that can find a given number of cliques with appropriate sizes, without common vertices between them. More, these cliques must cover the graph in sense of Definition 3. A small example is presented in Figure 2. In Figure 2a) are represented all cliques found by an algorithm inspired from that proposed by Bron & Kerbosch. In Figure 2b) are represented new cliques found after making a balancing between those initially determined. It can be seen in Figure 2b) that the intersection of these new cliques in terms of vertices is an empty set.

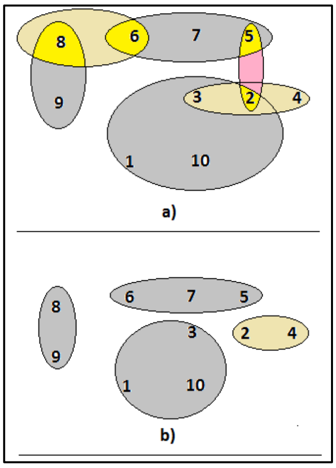


Figure 2 Balancing example

After experiments with cliques we observe that in the case of big and sparse graph the sizes of cliques after balancing are very small in relation to the size of the graph, most of them consisting only of two vertices. This situation in terms of solving large and sparse equations systems in parallel, would require a relatively large number of processors, that is not an option in most situations.

In such a situation -more and small cliques- the research was oriented mainly towards the pseudo-cliques problem. An efficient algorithm for enumerating all pseudo-cliques of a graph was proposed in paper [19]. This was a good starting point for an incipient algorithm that could find pseudo-cliques closer in size, independent in relation to the contained vertices and with a closer number of external vertices. The situation is similar to that one illustrated in Figure 2, but instead of cliques here are pseudo-cliques.

The case of a 20x20 linear system of equations exemplifies the proposed approach. The number of possible partitions/pseudo-cliques analyzed was 4, 5 and 10. The associated graph with 5 balanced pseudo-cliques has the lowest

International Journal of New Computer Architectures and their Applications (IJNCAA) 5(2): 94-101
The Society of Digital Information and Wireless Communications, 2015 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

number of external edges -equal to 32- as can be seen in Table 1.

After relabelling the associated graph, we obtain an equivalent equations system that has a better convergence in the case of the conjugate gradient method.

Thus, for the following linear equations system:

$7x_4 + 10x_7 + 13x_{10} + 16x_{13} + x_{18} + 3x_{19} = 511$
$8x_4 + 11x_7 + 14x_{10} + 17x_{13} + x_{16} + 4x_{19} = 562$
$10x_5 + 13x_8 + 16x_{11} + 3x_{17} + 6x_{20} = 501$
$7x_1 + 8x_2 + 13x_7 + 16x_{10} + 3x_{16} + 6x_{19} = 436$
$10x_3 + 14x_7 + 17x_{10} + x_{13} + 4x_{16} + 7x_{19} = 508$
$16x_8 + 3x_{14} + 6x_{17} + 9x_{20} = 452$
$10x_1 + 11x_2 + 13x_4 + 14x_5 + 3x_{13} + 6x_{16} + 9x_{19} = 460$
$13x_3 + 16x_6 + x_{10} + 4x_{13} + 7x_{16} + 10x_{19} = 499$
$3x_{11} + 6x_{14} + 9x_{17} + 12x_{20} = 510$
$13x_1 + 14x_2 + 16x_4 + 17x_5 + x_8 + 6x_{13} + 9x_{16} + 12x_{19} = 648$
$16x_3 + 3x_9 + 7x_{13} + 10x_{16} + 13x_{19} = 573$
$9x_{14} + 12x_{17} + 15x_{20} = 630$
$16x_1 + 17x_2 + x_5 + 3x_7 + 4x_8 + 6x_{10} + 7x_{11} + 12x_{16} + 15x_{19} = 722$
$3x_6 + 6x_9 + 9x_{12} + 13x_{16} + 16x_{19} = 692$
$15x_{17} + 18x_{20} = 615$
$x_2 + 3x_4 + 4x_5 + 6x_7 + 7x_8 + 9x_{10} + 10x_{11} + 12x_{13} + 13x_{14} + 18x_{19} = 1012$
$3x_3 + 6x_6 + 9x_9 + 12x_{12} + 15x_{15} = 495$
$x_1 + 2x_{20} = 41$
$3x_1 + 4x_2 + 6x_4 + 7x_5 + 9x_7 + 10x_8 + 12x_{10} + 13x_{11} + 15x_{13} 16x_{14} + 18x_{16} = 1183$
$6x_3 + 9x_6 + 12x_9 + 15x_{12} + 18x_{15} + 2x_{18} = 666$

the equivalent system after pseudo-cliques balancing and associated graph relabelling is:

$8x_2 + 11x_3 + 14x_5 + 17x_6 + 4x_7 + x_8 = 562$
$8x_1 + 13x_3 + 7x_4 + 16x_5 + 6x_7 + 3x_8 = 436$
$11x_1 + 13x_2 + 10x_4 + 3x_6 + 9x_7 + 6x_8 + 14x_9 = 460$
$7x_2 + 10x_3 + 13x_5 + 16x_6 + 3x_7 + x_{20} = 511$
$14x_1 + 16x_2 + 13x_4 + 6x_6 + 12x_7 + 9x_8 + 17x_9 + x_{10} = 648$
$17x_1 + 3x_3 + 16x_4 + 6x_5 + 15x_7 + 12x_8 + x_9 + 4x_{10} + 7x_{11} = 722$
$4x_1 + 6x_2 + 9x_3 + 3x_4 + 12x_5 + 15x_6 + 18x_8 + 7x_9 + 10x_{10} + 13x_{11} + 16x_{13} = 1183$
$x_1 + 3x_2 + 6x_3 + 9x_5 + 12x_6 + 18x_7 + 4x_9 + 7x_{10} + 10x_{11} + 13x_{13} = 1012$
$14x_3 + 17x_5 + x_6 + 7x_7 + 4x_8 + 10x_{12} = 508$
$x_5 + 4x_6 + 10x_7 + 7x_8 + 13x_{12} + 16x_{15} = 499$
$7x_6 + 13x_7 + 10x_8 + 16x_{12} + 3x_{14} = 573$
$10x_9 + 13x_{10} + 16x_{11} + 3x_{16} + 6x_{18} = 501$
$16x_7 + 13x_8 + 6x_{14} + 3x_{15} + 9x_{17} = 692$
$3x_{11} + 6x_{13} + 9x_{16} + 12x_{18} = 510$
$16x_{10} + 3x_{13} + 6x_{16} + 9x_{18} = 452$
$3x_{12} + 9x_{14} + 6x_{15} + 12x_{17} + 15x_{19} = 495$
$9x_{13} + 12x_{16} + 15x_{18} = 630$
$6x_{12} + 12x_{14} + 9x_{15} + 15x_{17} + 18x_{19} + 2x_{20} = 666$
$15x_{16} + 18x_{18} = 615$
$x_4 + 2x_{18} = 41$

In Figure 3 it can be seen a partitioning with 5 partitions/pseudo-cliques, before and after preconditioning the given equations system by pseudo-cliques balancing. The experimental results obtained using a parallelized implementation of conjugate gradient on IBM Blue Gene/P supercomputer are shown in Table 1.

A set of experiments was done using systems of equations with sizes between 10 and 300. The goal was to improve the convergence of parallelized conjugate gradient method after preconditioning by pseudo-cliques balancing. The average results of these experiments can be seen in Figure 4.

It is to be mentioned that there isn't a general rule to increase the speed of convergence with system size, as one might believe of Figure 4. In some cases, we obtained better results and in others worse. The graphic represents the average of results obtained for about a hundred systems for each size in part, ie 10, 50, 100, 200 and 300.

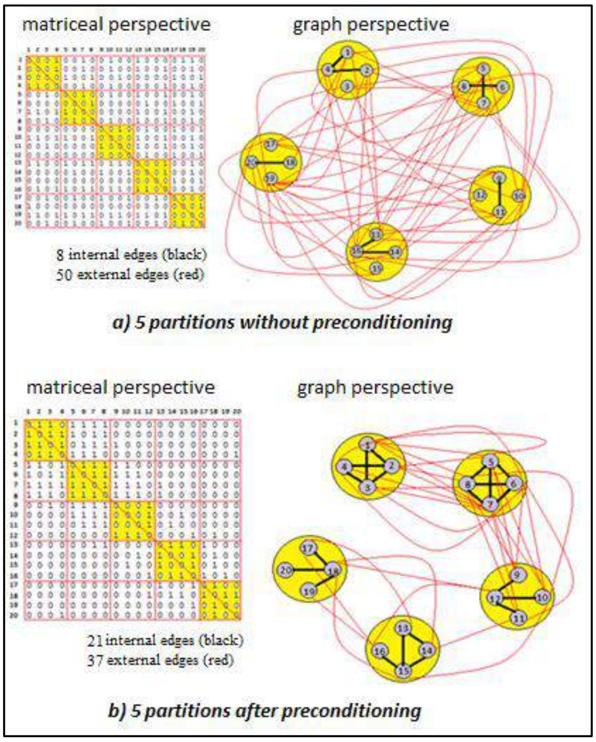



Figure 3 An example of partitioning

Table 1. Experimental results

| Partitions (pseudo-cliques) | Iterations necessary for convergence | | Internal/external edges | |
|---|---|---|---|---|
| | Initial | After balancing | Initial | After balancing |
| 4 | 47 | 47 | 9/49 | 9/49 |
| 5 | 59 | 47 | 8/50 | 21/37 |
| 10 | 55 | 55 | 8/50 | 8/50 |

After experiments, it has been observed:

- only pseudo-cliques balancing is not sufficient for improving the parallel computing process, but in most cases the pseudo-cliques balancing assure a minimal number of external edges for some partitioning, that can lead to a better convergence.

- the partitioning scheme is very important, sometimes even determinant;
- other factors, such as condition numbers of equations subsystems, are also very important in convergence;
- a lower number of external edges in pseudo-cliques lead to efficiencies of up to 30% in terms of convergence (or more in some cases).

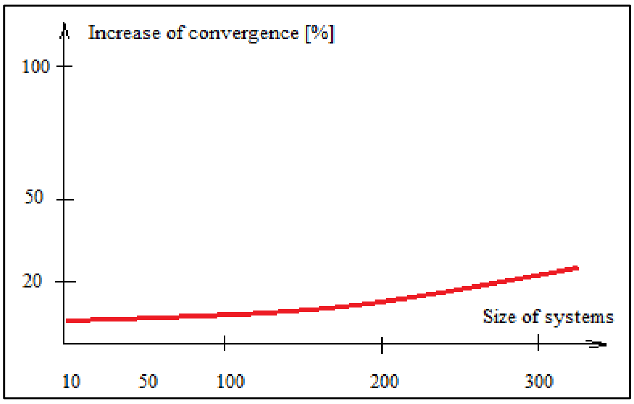



Figure 4. Improving the CG convergence

## 4 CONCLUSIONS

The proposed analogy *graph-equations* in this paper represents a new approach in terms of partitioning equations systems to solve these on a parallel computer and opens new research opportunities. This approach was validated through experiments.

An efficient algorithm for determining an optimal division of the associated graph in pseudo-cliques will be one of our research concerns.

At the same time, a study in terms of convergence depending by degree of filling must be taken into consideration.

## REFERENCES

[1] O. Axelsson, ”A survey of preconditioned iterative methods for linear systems of algebraic equations”, Springer, BIT Numerical Mathematics 1985, 25(1), 165-187, 1985

[2] M. Benzi, ”Preconditioning Techniques for Large Linear Systems: A Survey”, Journal of Computational Physics 182**,** 418–477, 2002

[3] M.R. Hestenes, E. Stiefel, ”Methods of Conjugate Gradients for Solving Liniar Systems”, Journal of Research of the National Bureau of Standards, 49(6), 1952

[4] S. Maruster, V. Negru, L.O. Mafteiu-Scai, ”Experimental study on parallel methods for solving systems of equations”, SYNACS Timisoara, 2012, IEEE Xplore CPS ISBN: 978-1-4673-5026-6, DOI: 10.1109/SYNASC.2012.7, 2013

[5] L.O. Mafteiu-Scai, ”Experiments and Recommendations for Partitioning Systems of Equations”, in West University of Timisoara Annals, 52(1), ISSN 1841-3307, DOI: 10.2478/awutm-2014-0009, 141-156, 2014

[6] Y. Saad, ”Iterative methods for sparse liniar systems”, Chapter 6: Krylov Subspace Methods, Part I”, SIAM, ISBN 978-0-89871-534-7, 2003

[7] U.V. Catalyurek C. Aykanat, ”Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication”, Parallel and Distributed Systems, IEEE Transactions, 10/7, 1999

[8] U.V. Catalyurek C. Aykanat, ”A Hypergraph-Partitioning Approach for CoarseGrain Decomposition”, Supercomputing, ACM/IEEE 2001 Conference, ISBN:1-58113- 293-X, IEEE, 2001

[9] U.V. Catalyurek E.G. Boman K.D. Devine D. Bozdag, ”Hypergraph-based Dynamic Load Balancing for Adaptive Scientific Computations”, Parallel MESH Distributed Processing Symposium, IEEE, ISBN:1- 4244-0910-1, 2007

[10] A. Cevahir, A. Nukada, S. Matsuoka, ”High performance conjugate gradient solver on multi-GPU clusters using hypergraph partitioning”, Computer Science - Research and Development, 25/1-2, 83-91, 2010

[11] K.D. Devine et. all, ”Parallel hypergraph partitioning for scientific computing”, Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, ISBN: 1-4244-0054-6, 2006

[12] B.A. Hendrickson, ”Graph partitioning and parallel solvers: Has the emperor no clothes?”, Proceedings of the 5th Solving Irregularly Structured Problems in Parallel, 218–225, 1998

[13] R.D. Luce, A.D. Perry, ”A method of matrix analysis of group structure", Psychometrika 14 (2): 95–116, doi:10.1007/BF02289146, PMID 18152948, 1949

[14] R.M. Karp, ”Reducibility among combinatorial problems”, in Miller, R. E.; Thatcher, J. W., Complexity of Computer Computations, New York: Plenum, 85–103, 1972

[15] T. Uno, ”An Efficient Algorithm for Solving Pseudo Clique Enumeration”, Problem, Algorithmica, 56 (1), 3-16, Springer, 2010

[16] B. Hendrickson, T. Kolda, ”Graph Partitioning Models for Parallel Computing”, Elsevier, Journal Parallel Computing - Special issue on graph partioning and parallel computing”, Volume 26 Issue 12, Nov. 2000, 1519 – 1534, 2000

[17] B. Hendrickson, ”Graph partitioning and parallel solvers: Has the emperor no clothes?”, în A. Ferreira et al. editors, Solving Irregularly Structured Problems în Parallel: 5th Intl. Symposium, Berkeley, California,USA, August 9-11, 1998, LNCS 1457, 218-225. Springer, 1998.
[18] C. Bron, J. Kerbosch, ”Algorithm 457: Finding all cliques of an undirected graph”, Communication of ACM 16 (9), 575–577, 1973
[19] T. Uno, ”An Efficient Algorithm for Enumerating Pseudo Cliques”, Algorithms and Computation, Lecture Notes in Computer Science, Springer, vol. 4835, 2007, 402-414, 2007

International Journal of
# NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

**The *International Journal of New Computer Architectures and Their Applications* aims to provide a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the field of computer architectures, information technology, and mobile technologies. The IJNCAA is published four times a year and accepts three types of papers as follows:**

1. **Research papers**: that are presenting and discussing the latest, and the most profound research results in the scope of IJNCAA. Papers should describe new contributions in the scope of IJNCAA and support claims of novelty with citations to the relevant literature.
2. **Technical papers**: that are establishing meaningful forum between practitioners and researchers with useful solutions in various fields of digital security and forensics. It includes all kinds of practical applications, which covers principles, projects, missions, techniques, tools, methods, processes etc.
3. **Review papers**: that are critically analyzing past and current research trends in the field.

Manuscripts submitted to IJNCAA **should not be previously published or be under review** by any other publication. Plagiarism is a serious academic offense and will not be tolerated in any sort! Any case of plagiarism would lead to life-time abundance of all authors for publishing in any of our journals or conferences.

Original unpublished manuscripts are solicited in the following areas including but not limited to:

- Computer Architectures
- Parallel and Distributed Systems
- Storage Management
- Microprocessors and Microsystems
- Communications Management
- Reliability
- VLSI