

APPLICATION OF PARTICLE SWARM OPTIMIZATION ALGORITHM IN  
ALLOCATING CLOUD RESOURCES FOR VIDEO ON DEMAND

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE  
OF  
THE MIDDLE EAST TECHNICAL UNIVERSITY BY



BETÜL AYGÜN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN  
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2018



Approval of the thesis:

**APPLICATION OF PARTICLE SWARM OPTIMIZATION ALGORITHM IN  
ALLOCATING CLOUD RESOURCES FOR VIDEO ON DEMAND**

submitted by **BETÜL AYGÜN** in partial fulfillment of the requirements for the degree of  
**Doctor of Philosophy in Information Systems Department, Middle East Technical  
University** by,

Prof. Dr. Deniz Zeyrek Bozşahin \_\_\_\_\_  
Dean, **Graduate School of Informatics**

Prof. Dr. Yasemin Yardımcı Çetin \_\_\_\_\_  
Head of Department, **Information Systems**

Assoc. Prof. Dr. Banu Günel Kılıç \_\_\_\_\_  
Supervisor, **Information Systems, METU**

Prof. Dr. Ahmet Coşar \_\_\_\_\_  
Co-supervisor, **Computer Engineering, THK**

**Examining Committee Members:**

Assoc. Prof. Dr. Altan KOÇYIĞIT \_\_\_\_\_  
Information Systems Dept., METU

Assoc. Prof. Dr. Banu GÜNEL KILIÇ \_\_\_\_\_  
Information Systems Dept., METU

Assoc. Prof. Dr. Nursal ARICI \_\_\_\_\_  
Computer Engineering Dept., GAZİ University

Assoc. Prof. Dr. Oumout CHOUSEINOGLU \_\_\_\_\_  
Industrial Engineering Department, HACETTEPE University

Assoc. Prof. Dr. Tuğba TAŞKAYA TEMİZEL \_\_\_\_\_  
Information Systems Dept., METU

Date:

03.09.2018





I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BETÜL AYGÜN

Signature :

# ABSTRACT

## APPLICATION OF PARTICLE SWARM OPTIMIZATION ALGORITHM IN ALLOCATING CLOUD RESOURCES FOR VIDEO ON DEMAND

Aygün, Betül

Ph.D., Department of Information Systems

Supervisor : Assoc. Prof. Dr. Banu Günel Kılıç

Co-Supervisor : Prof. Dr. Ahmet Coşar

September 2018, 108 pages

Video streaming services whether on demand or live has become one of the most popular services used recently. However, investments made for these type of applications cause a very serious financial problem just because video type of multimedia data needs more real time storage and high data transfer than other type of multimedia data. Furthermore, for the video streaming applications, significant amount of system resource in computing is required. To tackle this problem, cloud computing emerges as a preferred technology. Cloud services organizations are becoming more and more sophisticated as they enable the organizations to offer services without investing in hardware or software. A huge number of cloud service providers offer different pricing methods for various applications in various regions. For this reason, it is of great importance that incoming service requests are assigned to appropriate cloud services with minimum cost and maximum user satisfaction (QoS). Because of issues like multiple cloud providers, different quality of service requirements, different service level agreements (SLA) and uncertainties in demand, price and availability, optimization of resource allocation has some challenges. The objective of this study is to optimize the cost and performance of video on demand services using cloud CDNs, storage and transcoders based on QoS requirements of users. In this paper, Mixed Integer Quadratic Programming (MIQP) and different variants of Particle Swarm Optimization (PSO) algorithm are used to schedule video requests to cloud resources to achieve minimum cost of cloud services and maximum of

user satisfaction. Due to the nature of the problem, it is not possible to use the classic PSO, but the new algorithms which combine Binary PSO with heuristics algorithms are proposed. These algorithms are compared with LP algorithms which gives best result. The results show that proposed algorithms yield better results than the benchmarking algorithms.

Keywords: cost optimization, cloud services, QoS, PSO, resource allocation



# ÖZ

## BULUT KAYNAKLARINI TALEBE GÖRE VIDEO İÇİN TAHSİS ETME KONUSUNDA PSO TEKNİĞİNİN UYGULANMASI

Aygün, Betül

Doktora, Bilişim Sistemleri Bölümü

Tez Yöneticisi : Doç. Dr. Banu Günel Kılıç

Ortak Tez Yöneticisi : Prof. Dr. Ahmet Coşar

Eylül 2018 , 108 sayfa

İsteğe bağlı ya da gerçek zamanlı video akışı hizmetleri, son zamanlarda kullanılan en popüler hizmetlerden biri haline gelmiştir. Ancak, çoklu ortam verilerinin video tipinin, diğer multi-medya veri türlerinden daha fazla gerçek zamanlı depolama ve yüksek veri aktarımına ihtiyaç duyması nedeniyle bu tür uygulamalar için yapılan yatırımlar çok ciddi finansal soruna neden olmaktadır. Ayrıca, videoları servis edebilmek için önemli miktarda sistem kaynağına da ihtiyaç duyulmaktadır. Bu sorunu çözmek için bulut bilişim tercih edilen bir teknoloji olarak ortaya çıkmaktadır. Bulut hizmetleri organizasyonları, kuruluşların donanım veya yazılıma yatırım yapmadan hizmet sunmalarını sağladıkça gittikçe daha karmaşık hale gelmektedir. Çok sayıda bulut servis sağlayıcıları, çeşitli bölgelerde çeşitli uygulamalar için farklı fiyatlandırma sunmaktadır. Bu sebeple, gelen hizmet taleplerinin, minimum maliyet ve maksimum kullanıcı memnuniyeti (QoS) ile uygun bulut hizmetlerine tahsis edilmesi büyük önem taşımaktadır. Birden fazla bulut sağlayıcı, farklı hizmet kalitesi gereksinimleri, farklı hizmet seviyesi anlaşmaları (SLA) ve talep belirsizliği, fiyat ve kullanılabilirlik, kaynak tahsisinin optimizasyonu gibi konulardan dolayı bazı zorluklar vardır. Bu çalışmanın amacı, kullanıcıların QoS gereksinimlerine bağlı olarak bulut CDNleri, depolama ve kod dönüştürücülerini kullanarak talep üzerine video hizmetlerinin maliyetini ve performansını optimize etmektir. Bu makalede, Karma Tam-sayı Kuadratik Programlama ve Parçacık Sürü Optimizasyonu (PSO) algoritmasının farklı çeşitleri, bulut kaynaklarına video isteklerini atamak için kullanılmaktadır. Problemin doğası



geređi sürekli PSO kullanmak mümkün deđildir, ancak İkili PSO ile bulgusal algoritmaları birleřtiren yeni algoritmalar önerilmektedir. Bu algoritmaların sonuçları, en iyi sonucu veren MIQP ile karşılaştırılmaktadır. Sonuçlar, önerilen algoritmaların diđer algoritmalarından daha iyi sonuçlar verdiğini göstermektedir.

Anahtar Kelimeler: maliyet optimizasyonu, bulut servisleri, servis kalitesi, parçacık sürüsü optimizasyonu, kaynak tahsisi





*To my beloved husband Gürcan,*

*To my two little cute daughters, Ece Begüm & Arya Bengü*

*To my super parents, Fatma and Şerif Ali*

## ACKNOWLEDGMENTS

First of all, I would like to thank my previous supervisor Prof. Dr. Ahmet Coşar and supervisor Assoc. Prof. Dr. Banu Günel Kılıç for their constant support, guidance, trust and friendship. It was a great honour to work with them for the entire of the thesis. I also would like to thank to Assoc. Prof. Dr. Nursal Arıcı, who is member of my thesis committee, for her support at the critical phases of my academic study.

A lot of people influenced and supported this work scientifically and their contribution was most valuable for me. Therefore, I want to thank all my teachers in Middle East Technical University starting from Mathematics Department to the Informatics Institute.

I was working at the Public Procurement Agency at the thesis stage. I would like to thank the managers of the institution who understood the difficulty of working with writing PhD thesis in my tight times.

This work is also supported by TUBITAK-BİDEB PhD scholarship (2211) which made me more responsible and motivated to study and success.

My family also provided invaluable support for this work. I would like to thank and apologize specially to my husband Gurcan and my daughters Ece Begüm and Arya Bengü; since I needed extra time to learn and study, I unfortunately stole this extra time from ours. My mother Fatma and my father Şerif Ali helped me to reduce my shortcomings by taking care of my children. And my sister Yeliz and my brother Ahmet believed me, their support was so important for me.

And there are a lot of people that were with me in this tough doctoral road. They defined me, they made me who I am, they are true owners of this work. It is not possible to write down why each of them is important to me and this work, because it will take more space than the work itself. So I'll just give names of some of them; Duygu Fındık, Nurcan Alkış, Mualla Beyribey Yılmaz and Ayşegül Kılıç.

And lastly, I want to thank to my beloved husband, Gurcan. I am sincerely grateful of all that

you have done for me over the years. I could not have reached my goals without your help and support at all times. During these times, we have laughed, cried, grown up and brought up our little cute girls together. Thank you very much for being with and loving me endlessly.



# TABLE OF CONTENTS

ABSTRACT . . . . .	iv
ÖZ . . . . .	vi
ACKNOWLEDGMENTS . . . . .	ix
TABLE OF CONTENTS . . . . .	xi
LIST OF TABLES . . . . .	xvi
LIST OF FIGURES . . . . .	xviii
LIST OF ABBREVIATIONS . . . . .	xxi
CHAPTERS	
1 INTRODUCTION . . . . .	1
1.1 Background of the Problem and Problem Statement . . . . .	3
1.2 Significance and Purpose of the Thesis . . . . .	4
1.3 Research Questions . . . . .	5
1.4 Structure of the Thesis . . . . .	6
2 LITERATURE REVIEW . . . . .	7
2.1 Cloud Computing Resources Allocations . . . . .	7
2.2 Quality of Service Parameters for VoD . . . . .	14
2.3 Evolutionary Algorithms for Optimization Techniques . . . . .	15

2.3.1	Constrained PSO . . . . .	16
2.3.2	Binary PSO . . . . .	17
2.3.3	Multi-Swarm PSO . . . . .	17
2.3.4	Parallel PSO . . . . .	18
2.3.5	Drawbacks of PSO . . . . .	18
3	METHODOLOGY . . . . .	21
3.1	A Pre-Analysis for the Cloud Resources Selection and Algorithm . . . . .	22
3.1.1	Deciding whether encoded video will be saved to storage or not . . . . .	22
3.1.2	TTL Value Deciding . . . . .	24
3.2	Dynamic Public Cloud Resources (Virtual Machines and Storage) Allocation Problem for VoD Applications - VoDRAP <sub>VMS</sub> . . . . .	24
3.2.1	Parameters . . . . .	25
3.2.2	Assumptions . . . . .	25
3.2.3	Constraints . . . . .	26
3.2.4	Mathematical Modelling . . . . .	26
3.3	Dynamic Public Cloud Resources (Content Delivery Network, Transcoders and Storage) Allocation Problem for VoD Applications - VoDRAP <sub>CDNTS</sub> . . . . .	29
3.3.1	Parameters . . . . .	31
3.3.2	Assumptions and Constraints . . . . .	31
3.3.3	Mathematical Modelling . . . . .	33
3.3.3.1	Latency . . . . .	33
3.3.3.2	Storage Cost . . . . .	34
3.3.3.3	Content Delivery Network (CDN) Cost . . . . .	35

3.3.3.4	Transcoder Cost . . . . .	36
3.3.3.5	Security Options Cost . . . . .	37
3.3.3.6	Formulation . . . . .	38
3.3.3.7	Linear Algebra Representation . . . . .	40
3.3.4	Computational Complexity of the Problem . . . . .	41
3.4	Solution Approaches to the Proposed Mathematical Modelling of the Defined Problem . . . . .	43
3.4.1	Integer Linear Programming . . . . .	43
3.4.2	Mixed Integer Quadratic Programming . . . . .	44
3.4.3	Binary Particle Swarm Optimization . . . . .	44
3.4.4	Novel Binary Particle Swarm Optimization . . . . .	46
3.4.5	Constrained Particle Swarm Optimization . . . . .	48
3.4.6	Multi-Swarm Binary Particle Swarm Optimization with Greedy Heuristic Algorithm . . . . .	50
3.4.7	Parallel Multi-Swarm Binary PSO . . . . .	51
4	RESULTS . . . . .	53
4.1	Data and Implementation . . . . .	53
4.2	Validation Methods of the Proposed Model . . . . .	56
4.2.1	Forecast Accuracy Metrics . . . . .	58
4.2.1.1	Root Mean Squared Error (RMSE) . . . . .	58
4.2.1.2	Normalized Root Mean Squared Error (NRMSE) . . . . .	58
4.2.1.3	Weighted Mean Absolute Percentage Error (WMAPE) . . . . .	58
4.2.2	Implementation Details . . . . .	58

4.3	Experimental Results . . . . .	59
4.3.1	Case Study for the Model $CRAP\_VOD_{VMS}$ under Microsoft Azure Cloud Services . . . . .	59
4.3.2	Case Study for the Model $CRAP\_VOD_{CDNTS}$ under Amazon Web Services . . . . .	63
4.3.2.1	Experimental Results of Mixed Integer Quadratic Programming . . . . .	65
4.3.2.2	Experimental Results of Modified Versions of Binary PSO . . . . .	66
4.3.2.3	Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm . . . . .	69
4.3.2.4	Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm using Parallel Programming . . . . .	70
4.3.3	Case Study for the Model $CRAP\_VOD_{CDNTS}$ under Microsoft Azure Cloud Services . . . . .	72
4.3.3.1	Experimental Results of Mixed Integer Quadratic Programming . . . . .	74
4.3.3.2	Experimental Results of Modified Versions of Binary PSO . . . . .	75
4.3.3.3	Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm . . . . .	77
4.3.3.4	Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm using Parallel Programming . . . . .	79
5	CONCLUSION . . . . .	83
5.1	Summary and Contributions of the Thesis Study . . . . .	83
5.2	Future Work . . . . .	85



REFERENCES . . . . . 87

APPENDICES

A COST OF CLOUD WEB SERVICES . . . . . 99

A.1 The Cost of the Amazon Web Services . . . . . 99

A.2 The Cost of the Microsoft Azure Services . . . . . 104

CURRICULUM VITAE . . . . . 107



## LIST OF TABLES

Table 2.1	Literature Review of Resource Allocation . . . . .	9
Table 2.1	Literature Review of Resource Allocation . . . . .	10
Table 2.1	Literature Review of Resource Allocation . . . . .	11
Table 2.2	Literature Review of Service Allocation . . . . .	11
Table 3.1	Summary of the Problem . . . . .	21
Table 3.2	Symbols and Definitions . . . . .	27
Table 3.3	Symbols and Definitions . . . . .	32
Table 3.3	Symbols and Definitions . . . . .	33
Table 4.1	Samples for Video Size Evaluation . . . . .	54
Table 4.2	Aggregated Normal Distribution Parameters . . . . .	54
Table 4.3	Data Load Pattern Details . . . . .	54
Table 4.4	Cost of Azure Web Services . . . . .	56
Table 4.5	Pseudo-Code for the Benchmark Data Generation . . . . .	57
Table 4.6	PSO Parameters . . . . .	60
Table 4.7	Comparison of the Algorithms . . . . .	62
Table 4.8	PSO Parameters . . . . .	63
Table 4.9	Comparison of Algorithms Proposed in a Time Interval . . . . .	67

Table 4.10 Comparison of Different Variants of Binary PSO . . . . .	68
Table 4.11 Comparison of the Results of PSO Run Under Different Time Limits . . . . .	74
Table 4.12 PSO Parameters . . . . .	74
Table 4.13 Comparison of Algorithms Proposed in a Time Interval . . . . .	76
Table 4.14 Comparison of Different Variants of Binary PSO . . . . .	78
Table A.1 Cost and Type of Amazon Simple Storage Services (S3) . . . . .	99
Table A.1 Cost and Type of Amazon Simple Storage Services (S3) . . . . .	100
Table A.2 Amazon Simple Storage Service(S3) Transfer Pricing . . . . .	100
Table A.2 Amazon Simple Storage Service(S3) Transfer Pricing . . . . .	101
Table A.3 Amazon Elastic Transcoder Transcoding Pricing . . . . .	101
Table A.3 Amazon Elastic Transcoder Transcoding Pricing . . . . .	102
Table A.4 Streaming from CDN Transfer Pricing . . . . .	102
Table A.5 Streaming over Secured HTTP Transfer Pricing . . . . .	102
Table A.5 Streaming over Secured HTTP Transfer Pricing . . . . .	103
Table A.6 Amazon CDN(Cloud Front) Transfer Pricing . . . . .	103
Table A.7 Request Pricing for All HTTP- HTTPS Methods (per 10.000 requests) . . . . .	103
Table A.8 Cost and Region of Azure Media Services for transcoding HD and SD videos	104
Table A.9 Amazon CDN(Cloud Front) Transfer Pricing . . . . .	105
Table A.10 Cost and Region of Azure Media Services for transcoding HD and SD videos	105
Table A.10 Cost and Region of Azure Media Services for transcoding HD and SD videos	106

## LIST OF FIGURES

Figure 3.1 Architecture of Cloud Computing Systems . . . . .	30
Figure 3.2 Multi Swarm PSO with Greedy Algorithm . . . . .	51
Figure 4.1 Amazon Regions . . . . .	57
Figure 4.2 Comparison of bandwidth cost between LP and FIFO based algorithm when varying user video requests . . . . .	60
Figure 4.3 Comparison of the results of 1,000 iterative Proposed PSO optimization technique with LP results in varying user video requests . . . . .	61
Figure 4.4 Comparison of the results of 1,000 iterative Proposed BPSO and Novel BPSO optimization techniques with LP results in varying user video requests . . . . .	62
Figure 4.5 Comparison of the execution time of 1,000 iterative Proposed BPSO and Novel BPSO optimization techniques . . . . .	62
Figure 4.6 Amazon Web Service Latency Information . . . . .	64
Figure 4.7 Comparison between the results of FIFO resource allocation technique with MIQP with varying user video requests . . . . .	65
Figure 4.8 Comparison between the results of 1,000 iterative Binary BPSO, Novel Bi- nary PSO and MIQP in varying user video requests . . . . .	66
Figure 4.9 Comparison between the results of 1,000 iterative Novel BPSO, Neighbour- hood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	69

Figure 4.10 Comparison between the computing time of 1,000 iterative Novel Binary PSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	70
Figure 4.11 Comparison between the computing time of Multi-Swarm Neighbourhood Novel Binary PSO with Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	71
Figure 4.12 Comparison of the results of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests . . .	71
Figure 4.13 Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests . . . . .	72
Figure 4.14 Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests . . . . .	73
Figure 4.15 Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	73
Figure 4.16 Comparison between the results of FIFO resource allocation technique with MIQP with varying user video requests . . . . .	75
Figure 4.17 Comparison between the results of 1,000 iterative Binary BPSO, Novel Binary PSO and MIQP in varying user video requests . . . . .	77
Figure 4.18 Comparison between the results of 1,000 iterative Novel BPSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	79
Figure 4.19 Comparison between the computing time of 1,000 iterative Novel Binary PSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . .	80

Figure 4.20 Comparison between the computing time of Multi-Swarm Neighbourhood Novel Binary PSO with Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests . . . . . 80

Figure 4.21 Comparison of the results of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests . . . 81

Figure 4.22 Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests . . . . . 81



## LIST OF ABBREVIATIONS

<b>AMPL</b>	A Mathematical Programming Language
<b>ARIMA</b>	Autoregressive Integrated Moving Average
<b>ASFLA</b>	Augmented Shuffled Frog Leaping Algorithm
<b>AWS</b>	Amazon Web Services
<b>BPSO</b>	Binary Particle Swarm Optimization
<b>CBSP</b>	Cloud Based Service Provider
<b>CDN</b>	Content Delivery Network
<b>CPU</b>	Central Processing Unit
<b>CSP</b>	Cloud Service Provider
<b>DREAM</b>	Distributed Heuristics Algorithm
<b>EC2</b>	Elastic Compute Cloud
<b>EDF</b>	Earliest Deadline First
<b>FIFO</b>	First in First out
<b>fps</b>	Frame per second
<b>GA</b>	Genetic Algorithm
<b>GB</b>	Gigabyte
<b>HD</b>	High Definition
<b>HTTP</b>	Hypertext Transfer Protocol
<b>HULU HD</b>	Hrvatske Udruge Likovnih Umjetnika High Definition
<b>IaaS</b>	Infrastructure as a Service
<b>IDE</b>	Integrated Development Environment
<b>ILP</b>	Integer Linear Programming
<b>IP</b>	Internet Protocol
<b>IT</b>	Information Technology
<b>JPEG</b>	Joint Photographic Experts Group
<b>LFU</b>	Least Frequently Used
<b>LP</b>	Linear Programming
<b>LRU</b>	Least Recently Used

<b>MCDM</b>	Multi-Criteria Decision Making
<b>MILP</b>	Mixed Integer Linear Programming
<b>MIP</b>	Mixed Integer Programming
<b>MIQP</b>	Mixed Integer Quadratic Programming
<b>MNNBPSO</b>	Multi-Swarm Neighbourhood Novel Binary PSO
<b>MPC</b>	Model Predictive Control
<b>MPEG</b>	Moving Picture Expert Group
<b>NA</b>	Not Applicable
<b>NBPSO</b>	Novel Binary Particle Swarm Optimization
<b>NIST</b>	National Institute of Standards and Technology
<b>NNBPSO</b>	Neighbourhood Novel Binary PSO
<b>NP</b>	Non deterministic polynomial time
<b>NRMSE</b>	Normalized Root Mean Squared Error
<b>OVMP</b>	Optimal Virtual Machine Placement
<b>P2P</b>	Peer to Peer
<b>PaaS</b>	Platform as a Service
<b>PC</b>	Personal Computer
<b>PSO</b>	Particle Swarm Optimisation
<b>PMNNBPSO</b>	Parallel Multi-Swarm Neighbourhood Novel Binary PSO
<b>QoE</b>	Quality of Experience
<b>QoS</b>	Quality of Service
<b>RAM</b>	Random Access Memory
<b>RGB</b>	Red Green Blue
<b>RMSE</b>	Root Mean Square Error
<b>S3</b>	Amazon Simple Storage
<b>SaaS</b>	Software as a Service
<b>SCA</b>	Sine Cosine Algorithm
<b>SD</b>	Standard Definition
<b>SFLA</b>	Shuffle Frog Leaping Algorithm
<b>SIP</b>	Stochastic Integer Programming
<b>SLA</b>	Service Level Agreement
<b>SLP</b>	Stochastic Linear Programming
<b>SSL</b>	Secure Sockets Layer
<b>TB</b>	Terabyte



<b>TTL</b>	Time to live
<b>UHD</b>	Ultra High Definition
<b>URL</b>	Uniform Resource Locator
<b>VBR</b>	Video Bit Rate
<b>VLAN</b>	Virtual Local Area Network
<b>VPN</b>	Virtual Private Network
<b>VM</b>	Virtual Machine
<b>VoD</b>	Video on Demand
<b>VTR</b>	Video Tape Recorder
<b>WMAPE</b>	Weighted Mean Absolute Percentage Error
<b>WOA</b>	Whale Optimization Algorithm





# CHAPTER 1

## INTRODUCTION

The Cloud Computing is a new technology that grows exponentially in both academic and industrial institutions. Cloud systems become a more popular way for service providers to provide services to customers. Cloud computing is known as taking application which run on infrastructure of third parties. It is formally defined as "a model for enabling ubiquitous, convenient, ondemand 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" by National Institute of Standard and Technologies (NIST) [Mell and Grance, 2009]. As the definition implies, this new technology offers organizations many benefits which are classified into two categories as economical and technical [Botta et al., 2016]. Economical advantages can be explained as; cloud computing enables low cost public access to vast proprietary compute, storage and network resources and provides scalable, fault tolerant structured data management. Moreover, cloud computing reduces the cost of management of hardware and software resources and it provides improved resource optimization as well as no need for facilities, real-estate management and hiring employees. In addition to these economical benefits, it also offers technical benefits which are energy efficiency, needing less technical wisdom to implement hardware, software resources and security applications. Beside all of them, since the services provided by the cloud providers can be monitored, controlled and reported and paying only for services as you use instead of having them under high contracts, the idea of usage of cloud services become very appealing [Barba-Jimenez et al., 2016]. Since cloud resources are monitored, controlled and reported, this gives both providers and users clearness about the information of used resources [Espadas et al., 2013].

The survey conducted by IBM reveals that one of five organizations is adopting the cloud solutions and getting competitive benefit that accrued from both reduction in cost and increasing in efficiency [Armonk, 2013]. In these organizations, 170 percent prefer to use analytic tools to make decisions better and 136 percent more likely to use cloud to reinvent customer relationships. In summary, these organizations prefer to use cloud services to differentiate them from their opponents. The findings of the survey recommends to the leaders of the business and technology that they should give precedence to the investments on cloud area to gain advantage over competitors. This technology of usage of cloud computing, also attracts attention to industries like reputable social networking websites such as Facebook and also Google [Wang et al., 2014].

On the other hand, with the technology improvements, managing large amount of multimedia objects; such as audio, video, picture or a combination of them have become possible. Also,

video based services constitute the big part of the total download of the internet traffic [Ali-Eldin et al., 2015] and due to the heavy mobile content consumption, VoD (Video on Demand) traffic is continuing to increase [Passarella, 2012]. Processing and analyzing of multimedia data needs significant amount of computation processing capability, storing of them needs huge data storage capacity and streaming data needs high data transfer. Due to these reasons, cloud computing become a more popular way for service providers to provide services to customers. Owing to the elasticity of the cloud computing, VoD on cloud servers attain high performance and low cost than traditional methods [Zhao et al., 2014]. Although elasticity of the cloud computing ensures guaranteed QoS (Quality of Service), in real case, cloud resources can be inelastic especially for the start up delay. However, this new technology has some disadvantages; security and latency. These disadvantages are the main issues that need to be tackled and managed. Encryption and watermarking are two important issues for the security of multimedia data. Accessing multimedia data securely is another issue that needs to be considered. Most of cloud service providers provide organizations some security options. Streaming video over secure sockets layer (SSL) and storing encrypted data in cloud storages are the security examples provided by cloud providers. In this research, we entitle these security options with the network latency as QoS requirements as a whole. However, providing these options are raising the cost of cloud resources. So, the main of this study is to minimize cost while satisfying user QoS requirements.

Nonetheless, explosion of cloud computing services over the internet raises a new issue at determining and selecting a service [Sun et al., 2014]. For media service providers, adaptation to the Content Delivery Network (CDN) is an important point for both traditional methods and cloud systems; due to the fact that CDN usage provides reliable video services while reducing network congestion and service response time [Um et al., 2014]. CDN also provides services for on demand or live video. And one of the most crucial thing of it is that cloud CDN models are cheaper than the traditional CDNs [Barba-Jimenez et al., 2016]. Many big cloud services providers like Amazon, Microsoft and Google serves enormously wide range of cloud resources [Amazon Web Services, 2017, Microsoft, 2017, Google, 2017]. Of course, cloud providers pricing their resources in different ways. However, mostly they price their resources using 'as pay as you use' strategy or reservation strategy. The main purpose of the companies that use cloud resources is to minimize cost of usage of the cloud resources. At this point, cost is the main turning point of the organizations to adapt to cloud computing. Due to the different type of users, elastic payment policies of service providers and distinct QoS factors and heterogeneity in cloud services, selecting convenient service for cloud based service providers (CBSP) become extremely complicated case within a minimum cost and maximum satisfaction [Sun et al., 2014]. It is obvious that the most important reason why providers adopt and investigate cloud computing services is the cost. Thus, developing a mathematical model of the cloud resource allocation for VoD applications is a mandatory and keen interest by both academic and industry.

But, due to the huge number and distinct type of cloud services; divergence in video on demand applications and a wide range of quality of service features make resource allocation problem an NP (non-deterministic polynomial time) hard problem. Dealing with NP hard problem becomes another issue that must be fulfilled. The mathematical formulation of cloud services allocation becomes mixed integer quadratic non-convex problem. It is solved by using the Branch-cut method. Furthermore, for this kind of problems, prominent heuristic algorithms are used and it is extended according to the nature of our problem. All known and big companies have also some optimization techniques and in the literature there exist heaps of

studies. The performance success of these techniques alter in consonance with the nature of the problem. We apply the popular and recent proposed optimization techniques to our problem and compare them in terms of performance measure in accuracy and execution time.

Consequently, in the thesis, we take the perspective of video streaming providers which host their applications at an IaaS (Infrastructure as a Service) and SaaS (Software as a Service) provider. During this study, we define an NP hard problem for resource allocation of distributed cloud systems to optimize cost while considering quality of service requirements defined in service level agreements done between cloud service users for their services and customers. This study theoretically aims to be able to find an optimal resource allocation under a given workload in real time; and the corresponding SLAs from clients. It achieves minimum cost by using LP (Linear Programming) and other heuristic algorithms while meeting user requirements.

### 1.1 Background of the Problem and Problem Statement

Nowadays, resolution of the videos is increasing and we now see videos are created using HD and Ultra HD (UHD) techniques. As a result of these, video on demand applications constitute the huge part of the downstream of the internet [Ali-Eldin et al., 2015, Li et al., 2016, Juluri et al., 2016]. To process and store such high resolution videos, cloud computing provides an effective platform for computing, storage and transmission [Usman et al., 2016]. With the rise in quality and size of videos and development of technology, Cloud Services are started to become increasingly attractive for Information Technology (IT) and research industry; regarding computing and storage aspects [Armbrust et al., 2010]. Although geo-distributed clouds provides many advantages such as supporting large scales video on demand applications, it brings about new issues such as direction of requests to suitable cloud sites for timely responses at a lessen cost [Wu et al., 2012].

Cloud computing based on virtualization are divided into three levels: IaaS, PaaS and SaaS. The most commonly used virtualization is the server virtualization; however, many factors such as software, hardware, operating system, and network can be virtualized as well. For cloud service providers, supply services in a minimum effort and usage of resources are important keys to be successful in this battle on cloud servicing. For instance, Oracle buys big capacity server machines to win the battle on cloud systems. All known and big companies have some optimization techniques and also in the literature there exists heaps of studies. Cloud computing provides organization emphasizes on their own business values while freeing from tasks need to set up hardware and software infrastructures to deploy their applications. There exist different type of applications and services provided by cloud service providers and each of application has different composition, deployment and configuration requirements. Cloud environments of different applications and services under different conditions is deadly hard to do in terms of varying demand, supply patterns, system size and heterogeneous QoS requirements. For example, Amazon EC2 restricts their offering to the scale of infrastructure. In the literature, two main definitions for cloud resource allocation are defined; service allocation and resource allocation and there exist huge number of studies about these topics. However, greatest majority of these studies do not consider quality of service requirements and cloud resource types that are considered are varied according to study. Some of them only deal with CPU and memory, however considering network bandwidth is also important to provide good quality of services. In addition to this, security, response time, budget and other QoS

attributes should be taken into consideration. For the cost analysis, cost per memory and storage is decided while creating Virtual Machine (VM), while transferring data, bandwidth cost will be incurred. The aim of the scheduling algorithm is to seek a way to maximize performance while reducing cost and achieve QoS requirements defined in Service Level Agreements (SLAs). Most of the cloud service providers only deal with VM placement while minimizing cost. But, minimizing cost of VM resources especially for video streaming applications is not sensible. For these types of applications, CDN, storage and transcoder resources should be preferred. The study of Broberg et al. encourage end users to use the cloud CDN instead of using traditional CDNs in terms of charge for the reason that cloud CDNs are cheaper than traditional CDNs [Broberg et al., 2009]. Besides, the cost minimization is challenging because of the flexibility of video requests over time scales. To minimize the data access latency, cloud resources placement is decided according to data storage placement [Chen et al., 2015a] in addition to VoD application user location by using the latency information taken with in last 24 hours. Also, security is another topic that is discussed by the cloud service users is handled while streaming videos over cloud services. The aim of the scheduling algorithm is to seek a way to maximize performance while reducing cost (CDN, transcoder, storage and transfer) and achieve QoS features of latency, security and quality of video between customer and cloud user. Hence, for a given workload of videos in real time and the corresponding SLAs from clients, theoretically this study aims to be able to find an optimal resource allocation that provides minimum cost by using a mathematical model and solving it with machine learning techniques.

## 1.2 Significance and Purpose of the Thesis

In this study, we define an NP hard problem for resource allocation of distributed cloud systems to optimize cost while considering quality of service requirements defined in service level agreements done between cloud service providers and customers. The aim of this study is summarized as;

- To minimize the total cost of VoD service providers.
- To contribute and compare the existing heuristic algorithms in the literature
- To contribute to the QoS Requirements of VoD applications on Cloud Computing Services

First of all, parameters of QoS requirements of video sharing will be decided. The cloud products offered from different companies will be analyzed and handled during the optimum construction of cloud systems for application providers. While creating optimum scheduling resources, new or extended optimization methodologies will be created. In this study, we take the perspective of video streaming providers which host their applications at an IaaS and SaaS level. The cost minimization is challenging because of the flexibility of video requests over time scales. VM placement is decided according to data storage placement to minimize the data access latency. In this context, the following issues are researched and explored during this study;

- The architecture of Cloud Services offered by big Cloud Service Providers (CSP) for video streaming applications is designed.

- A mathematical model is provided for the designed architecture of cloud services for the cost analysis of offered services.
- Constraints are defined to ensure that application satisfies the user QoS features.
- Mathematical modelling is solved and optimized under QoS constraints.
- While solving the problem, various approaches are executed and compared with in various modifications and different number of iterations.
- New Binary Particle Swarm Optimization (PSO) and Whale Optimization Algorithm (WOA) are proposed and compared with the existing versions of them in the literature in terms of the performance of the accuracy and execution time.

In terms of QoS:

- To solve latency problem, CDN services and acceleration of transfer of storages are used in addition to taking the cloud services that have low latency value from the customer position. According to the latency, cloud services are grouped into three levels;
- Additional security options which are streamed over https and, encryption and decryption of the content are added.
- Different versions of videos by using video transcoder services are provided. We have handled fifteen different video versions.
- Analysis of deciding Time to Live (TTL) value for the video for caching and keeping different versions according to the popularity of it is attempted.

### 1.3 Research Questions

In this section, the research questions answered during the thesis are listed below;

1. What is the solution needed for VoD applications provided by the CSP?
2. How we design cloud systems (cloud servers and storages, CDNs and transcoders) to create effective systems for application providers?
3. QoS requirements for VoD applications are analyzed and what type of services (solutions) corresponds to satisfy QoS characterized for these kind of applications?
4. How to ensure video encoding service for different type of load conditions while satisfying a good trade of between QoS requirements and cost?
5. What kind of optimization techniques should be used to reach optimum solution while allocating resources or how we extend existing algorithms to overcome deficiencies of these algorithms and to yield better solutions?

## 1.4 Structure of the Thesis

The rest of the thesis is divided into four chapters. Chapter 2 reviews the literature in terms of the main topics that are studied during this thesis; Cloud Computing Resources (Services) allocation, QoS for video on demand applications, optimization algorithms that are used to solve the problem stated in this study. Chapter 3 offers the mathematical model constructed for cloud resources allocation for video on demand applications in two different perspectives in terms of services used and presents different solution methods for the mathematical model to ensure that proposed algorithms yield better results. Chapter 4 shows the results of the solutions of the problem and compares the techniques in terms of the performance of speediness and accuracy. Main contribution and conclusion about the thesis are explained and future works are listed in Chapter 5.





## CHAPTER 2

### LITERATURE REVIEW

This chapter outlines the related information about the topics researched in this study so far. As mentioned in introduction chapter, this study mainly contributes to the three main areas; resource allocation of cloud resources, QoS analysis for the type of video on demand applications on cloud and heuristic techniques. Although these topics are studied separately and especially heuristics algorithms are researched extensively, there is no study that deals with all of them simultaneously. Cloud computing technology is adopted by many organizations and it is considered to be futuristic technology due to the need for only limited technical knowledge and no need to invest in hardware and software. Thence, cloud computing is a matter of interest in both academic and industrial fields. In the event that the resources are not allocated well, it brings along astronomic costs. So, allocation of cloud resources properly for minimizing cost becomes very popular studied topic. During reviewing literature, we come across two different notions; cloud resources allocation and cloud services allocation. The cloud resources allocation concept can be classified into two groups. While first group minimizes the cost of CSP, second group minimizes the cost of service providers which uses cloud resources, i.e. cloud based service providers. First group allocate resources desired by the second group, most properly to the physical machines. The aim of the first one is to reduce cost, lower power consumption and increase customer satisfaction in terms of performance. Besides, second group, CBSP provides services to users with maximum performance and minimum cost by using resources proposed by cloud service providers. During this study, resource allocation by the view of the second group - CBSP are analyzed carefully in detail and solved effectively to increase the service performance and decrease the cost. Cloud service selection notion is apparently more self-evident than cloud resource allocation. Sun et al. classify the cloud service selection strategies into four groups; multi-criteria decision-making (MCDM), optimization based, logic based and others. In this study, while selecting appropriate services, a combination of MCDM and optimization-based approach is handled. In Section 2.1, studies that allocate cloud resources and/or cloud services are detailed. After detailed the closely related comprehensive cloud services allocation studies; Section 2.2. continues with the information about QoS for cloud based video service providers. Section 2.3 analyses the studies that are done on swarm intelligence algorithms and heuristic algorithms.

#### 2.1 Cloud Computing Resources Allocations

Cloud computing is an approach that allocate resources and connect to the user as desired when any application is run on from the pool where many servers and storages in different regions are connected [Nolle, 2009]. Network design for cloud computing can be done by using

public, private or hybrid cloud. For the public cloud, to bring down cost, the networking platform to connect cloud resources is the Internet [Nolle, 2009]. Designing cloud based services with private network and minimizing the cost is closed box that does not analyzed further. This study also deals with the network design of public cloud. The studies that allocate cloud resources and cloud services are given in Table 2.1 and Table 2.2 respectively. The study of Sun [Sun et al., 2014] gives detailed summary of the literature about cloud service allocation from many perspectives. The optimized-based resource allocation part of the study is summarized below with the additional studies that are researched after that time. While examining the studies, we characterize them according to five criterions;

- Cloud Resources: Which type of cloud solutions such as SaaS (transcoder, security options) or IaaS (VM, Storage) provided by CSP are used?
- Objective Function: What is the aim of the study?
- Search Area: In which areas or fields cloud resources are used?
- QoS : Which type of quality of service parameters are handled?
- Optimization or Resource Allocation Techniques: What kind of algorithms is used?

Table 2.1: Literature Review of Resource Allocation

Ref.	Cloud Resources	Aim and Area	QoS Attributes	Optimization Techniques
[Malawski et al., 2013]	VM, storage and Queuing Solution of Amazon	multiple heterogeneous compute and storage providers	cost and time between the computation machine and storages	Mixed Integer Non Linear Programming and AMPL modelling language
[Kaur and Mehta, 2017]	VM	cost minimization and scientific workflow applications	NA	an Augmented Shuffled Frog Leaping Algorithm (ASFLA)
[Mao and Humphrey, 2011a]	VM	effective while meeting deadlines of the jobs but not considers the efficiency of the cost comprehensively	NA	Earliest Deadline First (EDF) algorithm to schedule tasks on each VM type.
[Zhang et al., 2011]	distinct types of VM	maximize total revenue while minimizing the energy cost	NA	Constrained discrete-time optimal control problem and use Model Predictive Control (MPC)
[Zhao et al., 2014]	data center (bandwidth and storage)	satisfying SLA	VoD applications	Distributed Heuristic Algorithm (DREAM)
[Hu et al., 2016]	video sharing in online social network	satisfying SLA	Not Applicable	Lyapunov optimization theory
[Chen et al., 2015b]	VM	large scale streaming visual data		gradually scale up and prediction resource allocation
[Wei et al., 2010]	NA	sophisticated parallel computing problem	time and cost constraints	Game Theory
[He et al., 2014]	Amazon EC2 with on-demand, reserved and spot instances pricing	delivering video streams	desired playback rate	Lyapunov optimization technique

Table 2.1: Literature Review of Resource Allocation

Ref.	Cloud Resources	Aim and Area	QoS Attributes	Optimization Techniques
[Pandey et al., 2010]	Amazon EC2	work flow applications		PSO and Best Resource Allocation
[Mao et al., 2013]	Amazon EC2	work flow applications		PSO and Best Resource Allocation
[Nan et al., 2014]	server	multimedia sharing with in social network		The Stackelberg Game Approach
[Chaisiri et al., 2011a]	VM	general	NA	stochastic integer programming
[Aoun et al., 2010]	computing, storage, streaming and multicast node	Image processing for HD Video	NA	MILP
[Mao and Humphrey, 2011b]	VM	work flow	user defined deadline	Greedy and Gain
[Zhang-Jian, 2013]	on demand and reservation plan for VM	NA		Knapsack Problem to solve formulae and Kalman Filter to predict
[Legillon et al., 2013a]	VM	general processes	NA	New GA method with new defined operators
[Legillon et al., 2013b]	VM	general processes	NA	New GA method with new defined operators
[Bhise and Mali, 2013]	VM	workload having set of independent tasks with known computing/execution time	NA	naive algorithm and heuristic algorithm
[Zhang et al., 2014]	VM	cloud-based VoD system model	overload probability	ARIMA model to predict and integer programming for solving

Table 2.1: Literature Review of Resource Allocation

Ref.	Cloud Resources	Aim and Area	QoS Attributes	Optimization Techniques
[Vieira et al., 2014]	reserved and spot VM	general	3 QoS Types; FTRx, FTRt, VTR	ILP and heuristic algorithm
[Andrzejak et al., 2010]	EC2	workload	with in budget and time	probabilistic Optimization Method
[Adamuthe et al., 2013]	VM		within given time	GA and PSO

Table 2.2: Literature Review of Service Allocation

Ref.	Cloud Resources	Aim and Area	QoS Attributes	Optimization Techniques
[Ye et al., 2012]	I/SaaS	Aim and Area	QoS Attributes	Dynamic Programming
[Qu and Buyya, 2014]	VM	general	Performance, Elasticity, Security, Availability and Cost	Fuzzy Inference
[He et al., 2012]	SaaS	general	QoS Attributes	Greedy - integer programming
[Zheng et al., 2013a]	SaaS		Response time, Throughput	Greedy
[Esposito et al., 2016]	Storage		QoS Features	Fuzzy Logic, Theory of Evidence and Game Theory

The briefly summary of the studies are given in the both Table 2.1 and 2.2. Below, the studies in the literature discussed are given in further details.

Zheng et al. select the service by using the past data provided from other users and QoS collected by monitoring cloud services. They rank personalized QoS for the similar cloud services [Zheng et al., 2013b]. In the study of Qu et al., they provide a trust evaluation system that helps customers to select the appropriate VM service from CSPs by taking factors into consideration as transferring data from memory, secondary storage or to VM, availability, elasticity, scoring security mechanism and cost [Qu and Buyya, 2014]. Hu et al. clusters social users that have relationship between each other, close locations and similar interests to replicate videos based on cloud CDN [Hu et al., 2016]. They consider cost of three items: bandwidth cost for streaming, storage cost at CDN and replicating cost from storage to CDN. Although bandwidth cost is the main cost, second and third item is not applicable for the CSPs like Amazon. They do not consider the computation cost which is the second highest cost in cloud resources. Besides, they do not handle QoS attributes for the video like security. Chen et al. allocates VMs to stream for analyzing large-scale streaming visual data [Chen et al., 2015a]. In this study, only VM allocation is used, storage is not considered. They discover that VMs with fewer cores results better in terms of cost. Guiyi et al. also allocate cloud computing services by using game theoretic method [Wei et al., 2010]. They handle QoS constrained and computation intensive tasks performed on cloud computing services. In this study, real case for cloud computing services is not handled. Any VM or storage of any real cloud providers are not considered. Malawski et al. handle not only VM to process their data, they also consider the storages and queuing solution proposed by the Amazon. Their aim is to minimize the cost of the resources in addition to minimizing the time to transfer data between the computation machine and storages [Malawski et al., 2013]. Kaur et al. propose augmented shuffled frog leaping algorithm for the scientific work flows such as Montage, LIGO and CyberShake. They compare their proposed algorithms with the PSO and SFLA algorithms [Kaur and Mehta, 2017]. Zhang et al. use the VM resources of the cloud service providers. They provide discrete time optimal control problem. They use the spot instance of the VM machines [Zhang et al., 2011]. Chaisiri et al. propose provisioning algorithms to minimize the provisioning cost for long-term and short-term planning. They handle Amazon Elastic Compute Cloud (EC2) resources and they use the stochastic programming, robust optimization, and sample-average approximation optimization techniques to obtain optimal solutions [Chaisiri et al., 2011a]. They provision the on spot, reserved and on demand instance of EC2 for any type of applications that uses EC2. Furthermore, Chasiri et al. propose an optimal virtual machine placement algorithm (OVMP) with in reservation and on-demand payment systems. Aim is to minimize the user's budget. Optimal solution is achieved by using Stochastic Integer Programming (SIP). They use Bender decomposition and sampling average approximation approaches are used to optimize cost of reserved and on demand cloud resources. Trade-off between reservation plan and on-demand plan is to be adjusted optimal [Chaisiri et al., 2009, Chaisiri et al., 2011b]. Aoun et al. consider distributed data storage and multicast data transfer network services provided by CSP. They create Mixed Integer Linear Programming Algorithm for provisioning the aforementioned services. They use multicast node that are not yet deployed into cloud services. They decide the bandwidth between cloud resources. Objective of this study is to satisfy the highest number of end-users requests [Aoun et al., 2010]. Mao et al. propose an auto scaling mechanism under independent jobs of uniform performance requirements. The algorithm is effective while meeting deadlines of the jobs but not considers the efficiency of the cost comprehensively. They consider only virtual machines as cloud resources and uses Greedy and Gain scheduling algorithms. They use four workload

patterns; stable, growing, cycle/bursting and on-and-off [Mao and Humphrey, 2011b].

Hosain et al. consider multimedia service composition which are like streaming, transcoding, analysis and sharing [Hossain et al., 2012]. However, they allocate VMs to physical resources by considering memory and CPU of physical servers. Their approach is much more suitable for the first group researches. Hwang et al. propose two phase resource provision algorithms which first one includes long-term reservation subscription and on-demand subscription and second phase for prediction resource demand. They find upper and lower bound of the optimal amount of resources for long term reservation plan. They allocate VM and use Amazon EC2 as use case [Zhang-Jian, 2013]. Legillon et al. formulate problem with MIP and propose Genetic Algorithm (GA) with new defined operators to overcome deficiencies of MIP such as time consuming [Legillon et al., 2013a]. They compare their proposed heuristic algorithm with the GA. They also deal with only VM. They tried to minimize cost of renting VM which are served on hourly basis. Their solutions result in less time than MIP for the real life appliances [Legillon et al., 2013b]. Bhise et al. determine the right amount of resources to minimize cost and time from the user perspective. They handle EC2 of Amazon with on-demand and reserved and different instances prices. They first use naïve algorithm to find maximum number of VMs that needed, then apply heuristic algorithms to minimize number of VMs used [Bhise and Mali, 2013]. Zhenghuan et al. propose online cloud based VoD system model with the design of provisioning VM under QoS constraint. They use ARIMA model to predict the popularity of video. They use three different pricing strategy, on-demand, reserved and spot instance for Amazon EC2 [Zhang et al., 2014]. Ko et al. propose a method that deal with private cloud services [Ko et al., 2014]. They use time slot approach, and since private cloud has some finite resources, they are often constrained by the amount of investment. However, public cloud has infinite resources for the customers under some limitations. They also consider VMs for the resource of private cloud systems. Vieira et al. introduce a strategy to schedule VMs request on different public cloud providers with a minimum cost. They also implement QoS architecture and use ILP and heuristic methods to solve their problems. They classify QoS into three categories; FTRx, FTRt, VTR [Vieira et al., 2014, Vieira et al., 2015]. FTRx is used for fixed time request, i.e. request starts immediately and cannot be interrupted. FTRt is floating time request; request may not be started immediately but cannot be interrupted and VTR is variable time request i.e. may not start immediately and may be interrupted. Andrzejak et al. consider spot instances in Amazon EC2 to optimize monetary costs, performance and reliability under given user and application requirements. They offer probabilistic optimization method of cost, reliability and performance under user given conditions to satisfy the availability of spot instances in EC2 [Andrzejak et al., 2010]. Adamuthe et al. minimize the budget from customer perspective. They use PSO and GA algorithm and compare performance of these for tasks to virtual mapping. They also handle both reservation and on demand payment system [Adamuthe et al., 2013]. They minimize the used resources within given deadline. They design their jobs as task in workload. Gorde et al. deal with the reservation of the cloud resources to minimize cost. But over or under reservations result in higher cost, they use the prediction based resource allocation algorithms. They have two approaches: Bandwidth reservation and bandwidth pricing designed as a distribution optimization [Gorde et al., 2014]. On that account, when we go deep into these studies, we easily get that almost all studies discuss the allocation of VMs of CSPs. However, in contrast to traditional methods for video streaming, due to the extremely increases in demand on videos, VoD applications should rely on expensive CDNs. To satisfy the high quality in video streaming or satisfying QoS levels, using CDNs is crucial notwithstanding it brings about immense cost [Passarella, 2012]. Also, VoD applications

should use either CDN or P2P. But, CDN usage results in high availability [Zhao et al., 2014]. After all, in spite of the fact that it is clearly seen that the cloud resources like CDN, transcoder and/or storage are indispensable for the subject we focuses (VoD applications), there is not any study in the literature especially for these resources yet. Thus, this study provides video stream applications with minimum cost and maximum user satisfaction by using existing cloud services. On the other hand, in the literature, only VM services of cloud providers are dealt with in the studies and studies that deal with QoS attributes are quite poor. Additionally, we try to optimize cost and user satisfaction by satisfying QoS attributes of customers and use CDN, storage and transcoder services which are necessary services for VoD applications.

## 2.2 Quality of Service Parameters for VoD

Although most devices support many different encoding-decoding techniques, sometimes, between video capture and display, some decode and re-encode transaction causes reduction in fidelity and so reduction in video quality. In IP network, since best effort is applied, it is common for IP packets to be lost. Lost packets results in reduction in video quality. In our study, since public cloud is used, Internet, IP network is used. Also, best effort strategy of Internet brings about a substantial issue for public cloud service providers to satisfy QoS. So, QoS for video streaming applications should be carefully analyzed and integrated to the design of the cloud network.

Even if quality of service approach is applicable for many areas for a long while, especially for multimedia applications, it is indispensable. Service providers of such as video on demand applications have trouble in satisfying QoS by reason of diverseness of networks and customers, bandwidth and other challenges [Zhu et al., 2013]. Providing QoS of the cloud service providers is a challenging task for the end users [Sandhu and Sood, 2015]. There are many studies that analyze the quality of service and quality of experience modelling for video on demand applications. It is still an open question and no exact and clear relation between QoS and QoE [Juluri et al., 2016]. In some studies, especially start-up delay is tried to be minimized or guaranteed within a specified time in contrast to best effort of cloud service providers [Sujatha et al., 2007, Barba-Jimenez et al., 2016]. Although these studies focus on start-up delay which is an issue of CSP, in the study of Paudyal et al., the results show that initial delay does not actually affect the perceived quality of the video but jitters, throughput and packet loss rate has a significant effect on the perceived quality [Paudyal et al., 2014]. Moreover, some studies use the historic data or take the reservation strategy to satisfy the QoS for VoD applications [Niu et al., 2012, Zhang et al., 2004, Li et al., 2010].

Allocating resources for video sharing applications, choosing appropriate video quality type and needing for high transcoding are the main issues due to the heterogeneous video producers and broadcasters in terms of machine and location [He et al., 2016]. Well, then, we directly concern in QoS for video on demand applications under three main approaches. These are entitled as quality, latency and security and investigated under two perspectives; cloud providers and video streaming.

Latency for the cloud resources is a considerable issue that is discussed especially in industry [van der Zwet and Strom, 2018, Rouse, 2018]. There are many ways to overcome the latency problem for VoD applications. These are characterized as; defining new services at hosts or in nodes, new protocols which satisfy QoS attributes, new control algorithms for delay or error,



resource monitoring protocols, adaptive schemes for system changes and new architectures at host or switches [Nahrstedt and Steinmetz, 1994]. In terms of cloud providers, they provide several data centers and several cloud defined networking regions to choose. The regions of cloud resources are very important for data intensive services like streaming video. If the data center is far from user position, then QoS will decrease. When distance between provider and user increases and power of signal decreases, packet loses on the internet increases [Rouse, 2018]. Since distance is very important in terms of network latency, keeping location of user and location of cloud storage closest is also one of the constraints that must be satisfied in our problem formulation. Low latency should be the main goal for increasing the effectiveness of the video streaming applications [Cores and Subsystems, 2017]. Heretofore, while designing objective function, both distance between cloud resources and distance between user and cloud resources are considered for QoS by just assigning cloud resources that has low latency value to the user and holding cloud resources in the same regions to minimize the transfer cost and delay of different quality versions [He et al., 2016]. From the video perspective, there are two ways for the quality, network and user perspective in the literature. From the network perspective, packet loss and delay is handled. In the user perspective, perceived quality of video is considered. Video quality issues can be introduced while displaying video on a device. From the user perspective, the quality of video can be achieved by making compression more advanced. The best performing compression algorithms needs high computing resources. However, it results in higher latency. So selecting suitable compression standards and streaming by using suitable standard results in less latency while satisfying customer needs. In this context, according to the user needs in terms of QoS, different formats and compression standards will be applied to the video streamed. User's bandwidth limitation hedges high quality video signals, due to the VoD user's low speed connection [Microsoft, 2016]. So, according to the user constraints, selecting suitable video format and sending it is a good start point for the service providers to prevent users disappointed. We have dealt with different type of formats due to the influence of the video compression formats such as MPEG or H26x on the video quality [Chen et al., 2015c]. For the video streaming applications, although many studies take into considerations of jitters, delay, packet loss rate and throughput as the QoS attributes [Sandhu and Sood, 2015, Chen et al., 2015c, Klymash et al., 2014], small amount of studies in literature take security [Aurrecochea et al., 1998, Welch et al., 1998, Dhir et al., 2016]. But then, Irvine et al. take the security as a dimension of QoS. The security ranges in this context can be defined as binary, i.e. either it is satisfied or not [Irvine and Levin, 2000]. Besides, Manuel also takes the security as the one of the QoS factor in addition to cost in his study [Manuel, 2015]. Chen et al. are analyzing and examining the QoS for cloud gaming systems [Chen et al., 2014]. They concentrate on three main QoS attributes; traffic characteristics, latency and graphic quality. Although Li et al. thought that video streaming clients care with only latency [Li et al., 2016], we also regard highly security options which is another important matter in cloud computing systems.

### 2.3 Evolutionary Algorithms for Optimization Techniques

In the real world, there are many problems emerge and need to be solved. In line with the type of the problem, selecting suitable algorithm is very crucial point to come up with comprehensive solution. There are many algorithms proposed to solve real world problems in the literature. Some of these algorithms are heuristic algorithms to solve optimization and scheduling problems. But, these types of algorithms do not always give an explicit result. Due to this reason, using algorithm like LP is better starting point for the rest of the study

to compare performance of the proposed evolutionary algorithms. However, due to the time complexity of the LP, using heuristics algorithms while solving our problem provides better results in terms of the execution time. PSO and GA are two accepted popular evolutionary algorithms that are used mostly to solve problems in the literature. PSO is a type of swarm intelligence technique, developed by J. Kennedy and R. Eberhart in 1995, imitates the behavior of birds in their social life and finds the optimum solution through observing travel of particles in the population. PSO is used in many areas by using many different versions. The popularity of this technique comes from the easiness in its concept and coding, less sensitive to the nature of the objective function than the other heuristic methods, limited number of parameters and generating high quality solution within a short time [Lee and b. Park, 2006, Karpat and Özel, 2006, Bai, 2010]. Further, PSO has a faster convergence rate than GA and it can be applied to discrete problems easily [mei Yu et al., 2004]. Furthermore convergence speed of GA may become too slow when coding chromosomes with more genes to enhance the accuracy of the algorithm when a problem is complex and need many parameters than PSO [Karpat and Özel, 2006]. Additionally, PSO technique in nonlinear function is implemented successfully [Yang et al., 2007].

There are many variants of PSO is introduced by the academicians. Zang et al. reviews PSO algorithm comprehensively in many aspects like modifications, hybridization, and in many areas like automation, engineering, chemistry in their study [Zhang et al., 2015]. Advances on PSO are characterized as modifications, hybridization, extensions of PSO, theoretical analysis of PSO and paralleled PSO [Zhang et al., 2015]. In this thesis scope, we have been interested in extension of PSO to discrete data, adaptation to constrained problems and Parallel Multi-Swarm PSO. The studies that are done on these topics are detailed in the following sub chapters. From this groundwork, we can continue with discussing the searches that apply PSO with binary, parallel constrained muti-swarm PSO.

### 2.3.1 Constrained PSO

Although many of the algorithms are created for unconstrained problems originally, to solve real world problems, handling constraints is mandatory for the adaptation of algorithms to the problems. There are many ideas proposed to conduct constraints into algorithms. Koziel et al. definitely grouped constrained PSO into four categories: methods based on preserving feasibility of solutions; methods based on penalty functions; methods based on a search for feasible solutions; and hybrid methods [Koziel and Michalewicz, 1999]. When we look through the literature, broadly, there are two main attitudes to cover constraints; transforming constrained function into unconstrained one and direct search [Abd-El-Wahed et al., 2011]. As an example of the first approach, Kim et al. use augmented Lagrange multiplier to transform constrained problem into unconstrained problem [Kim et al., 2008]. In general approach of the direct search in the literature is comparing particles by pair according to fitness value [Cagnina et al., 2008, Wimalajeewa and Jayaweera, 2008, Worasuchep, 2008, Deb, 2000, Pulido and Coello, 2004]. If two of them are feasible then the particle that gets better value is taken. If one of them is feasible and the other is in-feasible, then surely feasible particle is added to population. If two of them are in-feasible solutions, then the particle the lowest sum of constraint violation is selected. Another approach is handled in the study of Hu et al. which is that PSO algorithm starts with the feasible solutions and feasibility function is used to check the updated solution satisfies all the constraints defined [Hu et al., 2003]. If the solution satisfies constraints, then continue with the updating particle best position, otherwise

even the solution has the best position, it is ignored. Also, penalty function is used in many researches to ignore in-feasible solutions [Parsopoulos et al., 2002, Deb, 2000]. Consequently, the approaches to cover constraints in the problem for PSO algorithm are almost similar to each other. Aggregation of these approaches is examined to handle constraint that is detailed in the methodology chapter.

### 2.3.2 Binary PSO

Since, nature of our problem domain is not suitable to the PSO algorithm; we have to propose new extensions of PSO which is binary PSO to solve our problem. PSO technique is firstly aroused for continuous problems. But it has evolved for constrained and multi objective problems. Kennedy and Eberhart who are the creator of continuous PSO also defined a discrete binary version of PSO [Kennedy and Eberhart, 1997]. However, they have two main problems which are parameters and memory for the classical binary PSO proposed [Khanesar et al., 2007]. Problem defined in the study of Khanesar et al. has discrete nature and classic PSO did not give proper results in discrete problems in comparison with continuous problems. They proposed binary PSO technique that also gives considerably better results in our binary problem.

Jeong et al. propose a Quantum-Inspired Binary PSO that is different from traditional binary PSO while updating velocity of the particle [Jeong et al., 2010]. The proposed algorithm does not depend on inertia and acceleration coefficients. Menhas et al. compare four proposed binary PSO in terms of performance. As we get, especially while updating position, interpretation of velocity changes in these studies while updating the position of the binary particle [Menhas et al., 2012]. To sum up, for the binary PSO, there are three main approaches exists. First approach, simplest one, does not change the general velocity and position update; they have only give insight to the interpretation of velocity. The second approach is to propose a new operator while updating the velocity and position of the particles but this approach is generally depend on to the problem domain and it is very hard to generalize them. Last approach is that changing velocity update entirely like quantum-inspired PSO approaches [Meng et al., 2010, Jeong et al., 2010].

### 2.3.3 Multi-Swarm PSO

Furthermore, by many researchers, multi-swarm PSO is proposed. Multi-swarm PSO approach has many advantages; increases the performance, decreases the execution time. Multi-swarm PSO converges effectively especially in huge and complex problems that requires more particles and iterations [Ostadrahimi et al., 2012]. In some studies, populations are divided into small swarms and information is exchanged between them that are grouped [Liang and Suganthan, 2005]. Also, Blackwell proposes Multi swarm PSO and proposes methods outperforms significantly than single population PSO. Solomon et al. propose parallel multi-swarm PSO that speeds up 37 times than single population and executes sequentially [Solomon et al., 2011]. Master-Slave multi swarm is proposed in research of Niu et al. [Niu et al., 2007]. In this approach, slave swarms run PSO separately are constructed identical and send information to master swarm. In the study of Van den Bergh et al., to cover n dimensional search space, swarm is divided into k sub-swarms [Van den Bergh and Engelbrecht, 2004].

Also, in many studies inertia weight is updated every iteration by using some equations, they

have showed that proposed methods yield better results [Shi and Eberhart, 1998, Eberhart and Shi, 2001]. Niu et al. also propose commensalism master slave multi swarm PSO [Niu et al., 2007]. Sub-swarms are classified into two groups; collaborative and concurrent [Brasileiro et al., 2017]. In concurrent approach, swarms compete to each other, they do not share information and they do not work in the same area. In collaborative approach, they communicate each other and exchange information and search in the same area. The approach you choose will depend entirely on the nature of the problem.

### 2.3.4 Parallel PSO

Although, PSO are more effective performance in terms of time than the other heuristics algorithms and has low execution time, if the size of data or particles increases, then automatically PSO runs in a larger time. To solve this issue, we propose a parallel multi-PSO by using parallel programming. Since for each step all particles are independent from each other, it is easy to make parallel the algorithm [McNabb et al., 2007]. We propose synchronous algorithm for the PSO. These are detailed according to the proposed algorithm. Due to the huge number of input characteristics, making parallel and sending huge data as a parameter may slow down the algorithm instead of speeding up. So, making parallel of points in the algorithm is very important to get pleasurable result. In the literature, researches propose synchronous and asynchronous parallel PSO according to the needs in their problem domain [Koh et al., 2006]. In the study of Kim et al., population is divided into sub-population and each population interacts with only two neighbourhoods (sub-populations) [Kim et al., 2011]. Also, Lou et al. apply parallel PSO by divide population into sub-population and after some generation, best individuals are shared among sub-populations [Su-hua et al., 2006]. Each particle is considered as a different agent. McNabb et al. propose a map reduced PSO [McNabb et al., 2007]. In the map, each particle is updated; velocity, position and local best. In the reduce phase, global best is updated by taking information from all particles. The synchronous PSO is parallelized easier than asynchronous PSO [Cui and Weile, 2005]. In parallel PSO, if the design points are selected synchronously, all design points must be evaluated before the next iteration processes started. Although this synchronous approach may not be efficient as in asynchronous approach especially in heterogeneous networks [Venter and Sobieszczanski-Sobieski, 2006], in our parallel approach, we executes all distinct swarms separately, and at the end of the algorithms converge then they exchange information. So, developing algorithm in an asynchronous manner is not necessary for the present. Mussi et al. efficiently categorize parallel PSO approaches studied in the literature [Mussi et al., 2011]. They have classified them into three paradigms; master-slave, coarse-grained and fine-grained. If the swarms are divided into different swarms, then communication topology and messaging protocols must be defined. However, in our case, we do not need to messaging protocol until the end of the algorithm.

### 2.3.5 Drawbacks of PSO

Almost in all researches, the dependency of the particle is not considered while updating position of the particle. Also both in modified PSO and classic PSO, there is no relationship between the elements of the position of particles. Nonetheless, in our situation, if we update the position of particles separately, then the particles can easily be outside of the domain of particles. After updating position of particle, if particle is not feasible, then adjustment

technique is proposed as in the master thesis of Lin to handle this situation [Lin, 2005].

However, it is so problem oriented and cannot be generalized. To propose more general approach for this type of domain, we propose a new mathematical formula for the velocity and position update that considers the relationship between elements of particles. Each particle in our problem domain must be unit binary vectors i.e. contains at most only one 1 and the other elements must be 0. Each particle can be regarded as a  $d$  dimensional orthonormal unit vector.

Although there are many variants of PSO exists in the literature, applying all them is not possible and selecting suitable variant is an effective start point for the effectiveness of the algorithm. In fact, when we analyze our problem, using multi-swarm is inevitable. Because, since we have more than one different resource, we have to allocate them separately. Otherwise, designing particle for the problem becomes pointless when we combine all of them. Although PSO yields an effective result on our problem, new heuristics algorithms are applied like Whale Optimization and Sine Cosine algorithms which are also swarm intelligence algorithms.





## CHAPTER 3

### METHODOLOGY

This chapter presents the proposed mathematical cost model formulation of the cloud services allocation system and describes the proposed solution approaches for the model. Cost reflects the amount of resources allocated and consumed. In such manner, service selecting is extremely critical step for the modelling. For the VoD applications, two different approaches are adapted from the standpoint of services used in our theory. First approach considers the IaaS cloud resources. This kind of cloud systems propose on-demand or reservation virtual-machine disk image library, raw block storage, and file or object storage, firewalls, load balancers, IP addresses, virtual local area networks (VLANs), and software bundles. In the first phase, we take into virtual machines and storages to stream videos. The second approach conducts on the SaaS besides IaaS resources. Due to the essentiality of the CDN for the video on demand applications, CDN and Transcoders for the cloud systems are much more preferable new solutions provided by the CSP. For the SaaS resources, CDN and transcoders are included in the design of the model with the security options. We design a resource manager that takes the required QoS specification which are analyzed systematically in detail in the following chapters. During this study, user requirements considered, cloud resources used and constraints handled are given in the following Table 3.1.

Table 3.1: Summary of the Problem

<b>User Requirements</b>	<b>System Resources</b>	<b>Experimental Constraints</b>
Video size and duration	Cost of Storage	Cost Constraint
Video type (HD / SD)	Cost of VM	Quality Constraint
User Location (Latency time)	Bandwidth Cost of Resources	Time Constraint
Video Format	Cost of CDN Bandwidth	Security Constraint
Priority	Cost of Transcoder	

In Section 3.1, the details of the offered services are analyzed and explanation of the reasons whether we handle the services or not are detailed. The mathematical model formulation for two approaches is given in Section 3.2 and 3.3 respectively. Section 3.4 describes the proposed solution algorithms to the problem discussed on the previous two chapters.

### 3.1 A Pre-Analysis for the Cloud Resources Selection and Algorithm

In the first part, we firstly analyze the decision of whether keeping the different type of formats of each video in storage or, transcoding each video in each time in terms of cost and time contexts. Next, if the video is streamed over CDN, we will need to decide time to live (TTL) value.

#### 3.1.1 Deciding whether encoded video will be saved to storage or not

Adaptation to encoding quality in response to network conditions and user satisfaction is certainly necessary to maximize user perceived quality. Due to the QoS attributes, network conditions and differences of user devices that requests multimedia data, different type of versions of data may be required. Since in the cloud, there is not any opportunity to detect network conditions and they charge according to size of the data that are sending over cloud systems, we adjust video compression type according to user perceived quality requirements and cost. In Amazon web services, Amazon Elastic Transcoder transcodes multimedia data in cloud from their source format into versions that will playback on devices that are compatible like smart phones, tablets and PCs. It uses content duration based pricing model; it is paid based on the length of output of content. It will need to be decided whether to keep the transcoded video in a storage or not by considering the costs used for transcoding and storing in cloud storage and the popularity i.e. frequency of the video requested. We define a threshold model to distinguish range of value to predict whether to store or not in cloud database instead of transcoding in each time. Let define function of each transaction; if  $t_{\delta t}(x)$  is 1 then the type of that video will be stored instead of transcoding in each time. If its value is zero, then no need to store them in cloud storages. This analysis is done under the assumptions stated below.

$$t_{\delta t}(x) = \begin{cases} 0 & f_{\delta t}(x) \leq 1 \\ 1 & f_{\delta t}(x) \geq 1 \end{cases} \quad (3.1)$$

$$f_{\delta t}(x) = \frac{\text{transcoding}_{cost}}{\text{storage}_{cost}} \quad (3.2)$$

$$f_{\delta t}(x) = \frac{\text{freq}(x) \times \text{dur}(x) \times \text{trans}_{price}(x) + 2 \times \text{bandwidth}_{price}(x) \times \text{size}(x)}{\text{number}_{storages} \times \text{size}(x) \times \text{storage}_{price}(x)} \quad (3.3)$$

$$f_{\delta t}(x) = \frac{2 \times \text{bandwidth}_{price}(x)}{\text{number}_{storages} \times \text{storage}_{price}(x)} + \frac{\text{freq}(x) \times \text{dur}(x) \times \text{trans}_{price}(x)}{\text{number}_{storages} \times \text{size}(x) \times \text{storage}_{price}(x)} \quad (3.4)$$

Since bandwidth cost with in the same cloud regions are equal to 0,

$$\frac{2 \times \text{bandwidth}_{price}(x)}{\text{number}_{storages} \times \text{storage}_{price}(x)} = 0 \quad (3.5)$$



If  $f_{\delta t}(x) < 1$  then

$$\frac{freq(x) \times dur(x) \times trans_{price}(x)}{size(x) \times storage_{price}(x)} \leq 1 \quad (3.6)$$

Using Amazon Elastic Transcoder, Amazon S3 and Amazon CloudFront, you can store, transcode and deliver your content. For the transcoding video, bandwidth cost will be incurred while transferring data from storage to elastic transcoder and back to storage. For the amazon case study, since bandwidth price within services is zero then second part of the equation is zero. If the following equation is satisfied, then any transcoded video will not be needed to kept in storages.

$$freq(x) \times dur(x) \times trans_{price}(x) \leq number_{storages} \times size(x) \times storage_{price}(x) \quad (3.7)$$

Under the following assumptions;

- We take the two extreme points which is highest quality video format (RGB 3 × 16) and lowest quality video format (HULU HD) video.
- Assume that the video duration is 1 minute.
- For 1 second video, video size is 17.9 GB (299 MB/s) for RGB and 10.4 MB (173 KB/second) for HULU HD.
- For Amazon S3, minimum cost is 0.023 for 1 GB and transcoding cost for each minute is \$0.030.
- There are 14 different storage regions exists in AWS.

Under these circumstances, equation becomes;

$$(High\ quality\ of\ video\ case) \quad freq(x) \times 1 \times 0,030 \leq 14 \times 17,9 \times 0,023 \quad (3.8)$$

$$(High\ quality\ of\ video\ case) \quad freq(x) \leq 192,13 \approx 192 \quad (3.9)$$

$$(Low\ quality\ of\ video\ case) \quad freq(x) \times 1 \times 0,030 \leq 14 \times 0,0104 \times 0,023 \quad (3.10)$$

$$(Low\ quality\ of\ video\ case) \quad freq(x) \leq 0.11 \approx 0 \quad (3.11)$$

To sum up, if the video quality is low and number of request of video is high then keeping all versions in storage is acceptable; however, if the number of request is low and quality of the video is high, then keeping in storage is not reasonable in terms of cost within a month.

### 3.1.2 TTL Value Deciding

TTL value needed to be adjusted for the field of multimedia data. While deciding the value of TTL, whether uploaded object has dynamic content or static content will be decided. Keeping TTL value small means it provides customer to serve dynamic content up to date however, increasing TTL duration means customers get better performance in terms of both time and cost of using this service. Since objects are more likely to be served directly from edge cache. The caching of the object becomes an important subject with the explosion of World Wide Web. In the literature, there exist different caching policies. Caching policies are classified into three groups; direct extensions of traditional policies (LRU, LFU and FIFO), key-based policies, function-based replacement policies. These all policies does not consider cloud case which differs in that cloud resources does not charge storage cost in CDN resources. In our case, we offer a new caching policy for cloud resources that contains parameters; transfer time cost (depends on the size of the object), last access time and access frequency. If the content in CDN has changed and we need to invalidate cached object, we need to pay extra price for invalidation. So, deciding optimum TTL provides us low cost and higher quality of the objects. In cloud resources, TTL is directly proportional to transfer cost ( $size \times cost$ ) and frequency (number of request per time) and inversely proportional to amount of content dynamic.

$$TTL \approx \frac{size \times frequency}{dynamicity} \quad (3.12)$$

We define the dynamicity of content between 0 and 1. In our case, since content of the video does not change often, then take dynamicity approaches to 0. In addition to this, if the frequency of video increases (goes to infinity), then;

$$\lim_{\substack{dynamicity \rightarrow 0 \\ frequency \rightarrow \infty}} TTL = \infty$$

In our case study, Amazon Cloud Front TTL value is assumed to be maximum due to the low dynamicity of video contents.

## 3.2 Dynamic Public Cloud Resources (Virtual Machines and Storage) Allocation Problem for VoD Applications - VoDRAP<sub>VMS</sub>

The aim of this study is to achieve minimum cost by provisioning of cloud resources for the video streaming applications. In this study, cost is the main objective function that has to be minimized while satisfying QoS attributes under constraint programming. The cost reflects the amount of cloud resources allocated by the providers. In this context, the cloud resources that need to be considered for video streaming applications are storages, VMs and network bandwidth. We use the integer programming to model our problem. To decide the amount of required resources, video format characteristics must be well-defined. Since, according to video characteristics, time needed to stream and size needed to store can change. Some of these characteristics detailed in Final Cut Pro manual are given in the following list [Inc., 2010];

- The medium used to store data (one type of medium is selected in this study. But, for future work, it may be changed.)

- The video standard supported
- The aspect ratio of the video frame: The ratio of the frame width to the frame height
- The dimensions of the video frame: The number of pixels per line, and the number of lines per frame. (We assume that, HD video with 1080 lines uses 1920 pixels for each line and SD video with 720 lines uses 486 pixels for each line)
- The frame rate: The number of frames recorded per second.
- Color recording method: RGB, component (YUV), S-Video (Y/C), or composite.
- Compressor (or codec): A video compressor attempts to reduce the amount of digital data required to store each frame without compromising the quality of the image. (We assume that all type of versions of video exists on storage.)

### 3.2.1 Parameters

Parameters used for the problem formulation are detailed as;

- A set of Virtual Machine,  $VM = v_1, \dots, v_n$ , that are used for video streaming.
- Each VM has specific characteristics; core number  $core(v_i)$ , existence zone  $zone(v_i)$ , cost per second  $VM(v_i)$ , bandwidth cost per gigabyte  $B(v_i)$
- A set of storage,  $S = s_1, \dots, s_s$ , each one is used to store video.
- Each storage has specific characteristics; existence zone  $zone(s_i)$ , cost per gb  $S(s_i)$
- A set of user video requests  $U = u_1, \dots, u_m$  in two different units; time unit and size unit. Since, cloud providers charges their virtual machines per second and storages per megabyte.
- Each video request has some characteristics: type of video,  $type(u_i)$ , which shows whether it is high definition (HD) or standard definition (SD) and frames of video,  $f(v_i)$
- User video requests arrive in batches minutely. Each video has a frame according to uniform distribution with parameters mean and variance.
- Video time and size is evaluated according to type and number of frames.

### 3.2.2 Assumptions

The problem is formulated according to the following assumptions;

- At time slot  $t$ , let  $m$  different type of videos that are requested by users.
- Same video is requested by more than one user at the same time slot, but handled as one video request.

- The data transfer between storages and servers in different zones in Amazon services are not charged however in Azure Cloud systems data transfer cost is the same as from VM to internet. Due to this dilemma, storages and VM are allocated from same zone to reduce data transfer cost and to decrease the latency within cloud services.
- HD and SD videos are assumed to be type of JPEG2000.
- In each time interval, the ratio of HD videos requested are 30% and SD videos are 70%.
- Play rate for SD video is 30 fps and HD video is 24 fps.
- Uniform distribution parameters for SD videos are:  $\mu = 250frames; \sigma = 20frames$
- Uniform distribution parameters for HD videos are:  $\mu = 75frames; \sigma = 7frames$
- We assume that, for HD videos, play rate is 24 frames per second. For SD videos, play rate is 30 fps. HD ( $1920 \times 1080$ ) video file size is 15.3 MB for JPEG2000 type of videos for 1 second. SD ( $720 \times 486$ ) video file size is 2.99 MB for JPEG2000 type of videos for 1 second [Forret, 2017]. (Since higher coding bit rate means high in quality for video and more advanced compression means higher latency [AC, 2018], compression standards differ in bitrates, quality and latency. In this study, there is only one type of video is handled which is JPEG2000 for both HD and SD videos. In the following section, we will introduce that to increase the customer satisfaction, according to user need for video quality, video format can be selected.

### 3.2.3 Constraints

- In one VM, more than one video may be encoded or streamed up to maximum number of cores of each VM.
- Each video is streamed from only one VM.
- All storages store all different version types of video. (reducing time and cost for transcoding)
- To reduce the reduction in cost of data transfer between storages and VMs, we assume that the video is processed in storage and VM in the same zone. Otherwise, the objective function becomes non-linear that results in harder question and solution which is handled in Section 3.3.

### 3.2.4 Mathematical Modelling

For the resource allocation, the minimization of resource allocation cost is considered as the decision objective. Virtual machine and storage are used cloud resources during video streaming. In each time interval, cost of used resources is evaluated dynamically.

In general;  $Cost = Cost_{VM} + Cost_{storage} + Cost_{transfer}$  Where the transfer cost is equal to data transfer cost between cloud systems and from cloud systems to Internet. Total cost for storing video data, encoding and streaming video is given respectively;

$$\begin{aligned}
Cost_{VM} &= Price_{VMpermin} \times VideoDuration \\
Cost_{storage} &= Price_{StorageperGB} \times VideoSize \\
Cost_{transfer} &= Price_{TransferperGB} \times VideoSize
\end{aligned}$$

So, total cost is stated below;

$$J = Price_{VMpermin} \times VideoDuration + Price_{StorageperGB} \times VideoSize + Price_{TransferperGB} \times VideoSize$$

The designed cloud architecture contains six VMs grouped as binary according to zones inter-connected with varying bandwidth and having its own storage resources.

The problem can be stated as: "Assign each video request into cloud resources such that while estimating the used VM cost and storage cost, lowest total cost and best performance among all VMs and storage resources must be achieved." Let  $C(VM)_t$  be the total cost of all requested videos streamed from all VM at time t. Let n be the total requested video at time t. Let m be the total VM existed in cloud system. Each  $v_i$  can be streamed from only one  $VM_j$ . So,  $x_{ij}$  shows us that whether  $v_i$  is assigned to  $VM_j$  (1) or not (0). For all videos streamed, summation of  $x_{ij}$  must be equal to 1 which means that video is assigned to only one VM.  $C(S)_t$  is the total storage cost of all requested videos from storage  $S$  at time t. Let s be the total number of storages in cloud systems. Let  $C(B)_t$  be the total bandwidth cost for streaming videos from VM to internet at time t.

Cost functions of cloud resources; VM, storage and transfer are given respectively. The parameters used during the formulation is given in Table 3.2.

Table 3.2: Symbols and Definitions

Symbol	Definition
$n$	Number of videos requested at a time interval $\delta t$
$m$	Number of CDNs used at a time interval $\delta t$
$s$	Number of storages used at a time interval $\delta t$
$x_{ij}$	Decision variable - $video_i$ is streamed from $VM_j$ or not
$y_{ij}$	Decision variable - $video_i$ is stored in $S_j$ or not
$VM_i$	Cost of $VM_i$ per minute
$S_i$	Storage cost of $storage_i$
$B_i$	Bandwidth cost from $storage_i$
$size_i$	Total size of $video_i$
$duration_i$	Total duration of $video_i$
$zone$	Zone of the cloud resources

$$C(VM)_t = \sum_i (\sum_j (x_{ij} \times VM_j)) \times duration_i \quad \forall i \in [1, n], \sum_j x_{ij} = 1, x_{ij} \in 0, 1 \quad (3.13)$$

$$C(S)_t = \sum_i (\sum_j (y_{ij} \times S_j)) \times size_i \quad \forall i \in [1, n], \sum_j y_{ij} = 1, y_{ij} \in 0, 1 \quad (3.14)$$

$$C(B)_t = \sum_i (\sum_j (x_{ij} \times B_j)) \times size_i \quad \forall i \in [1, n], \sum_j x_{ij} = 1, x_{ij} \in 0, 1 \quad (3.15)$$

$$C_t = C(VM)_t + C(S)_t + C(B)_t \quad (3.16)$$

$$\text{Minimize}(C_t) \quad \forall t \quad (3.17)$$

By combining above equations, integer linear programming problem is expressed as the following;

Variables:

$$x_{ij} \quad \text{and} \quad y_{ij} \quad (3.18)$$

$$x_{ij} = \begin{cases} 1 & \text{if } video_i \in VM_j \\ 0 & \text{if } video_i \notin VM_j \end{cases} \quad (3.19)$$

$$y_{ij} = \begin{cases} 1 & \text{if } video_i \in S_j \\ 0 & \text{if } video_i \notin S_j \end{cases} \quad (3.20)$$

Objective Function:

$$J_{\delta t} = \sum_{i=1}^n \left( \sum_{j=1}^m (x_{ij} \times VM_j) \right) \times duration_i + \sum_{i=1}^n \left( \sum_{j=1}^s (y_{ij} \times S_j) \right) \times size_i + \sum_{i=1}^n \left( \sum_{j=1}^m (x_{ij} \times B_j) \right) \times size_i \quad (3.21)$$

such that

Constraints:

$$\forall i \in [1, n], \sum_j^m x_{ij} = 1, x_{ij} \in 0, 1 \quad (3.22)$$

$$\forall i \in [1, n], \sum_j^m x_{ij} \leq \#of\ core, x_{ij} \in 0, 1$$

$$\forall i \in [1, n], \sum_j^s y_{ij} = 1, y_{ij} \in 0, 1$$

$$\forall i \in [1, n], zone(VM_i) = zone(S_i)$$

Matrix Representation

$$\begin{aligned}
J_{\delta t} = & \begin{bmatrix} duration_1 \\ duration_2 \\ \vdots \\ duration_n \end{bmatrix}^T \begin{bmatrix} x_{11} & x_{12} & x_{1m} \\ x_{21} & x_{22} & x_{2m} \\ \vdots & & \\ x_{n1} & x_{n2} & x_{nm} \end{bmatrix} \begin{bmatrix} VM_1 \\ VM_2 \\ \vdots \\ VM_m \end{bmatrix} + \begin{bmatrix} size_1 \\ size_2 \\ \vdots \\ size_n \end{bmatrix}^T \begin{bmatrix} y_{11} & y_{12} & y_{1s} \\ y_{21} & y_{22} & y_{2s} \\ \vdots & & \\ y_{n1} & y_{n2} & y_{ns} \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_s \end{bmatrix} \\
& + \begin{bmatrix} size_1 \\ size_2 \\ \vdots \\ size_n \end{bmatrix}^T \begin{bmatrix} x_{11} & x_{12} & x_{1m} \\ x_{21} & x_{22} & x_{2m} \\ \vdots & & \\ x_{n1} & x_{n2} & x_{nm} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_m \end{bmatrix}
\end{aligned} \tag{3.23}$$

### 3.3 Dynamic Public Cloud Resources (Content Delivery Network, Transcoders and Storage) Allocation Problem for VoD Applications - VoDRAP<sub>CDNTS</sub>

Cloud providers supply their services in three types; private, public and hybrid. In public cloud, each service is provided over the internet by cloud providers despite the fact that some CSPs provide virtual private networks for large customers. During this study, we take into consideration of resources of public cloud services. And so forth, since the data is transformed over internet, then security becomes a tension for data flow. So, encryption is a leading perception for the customer satisfaction. Since without usage of any cloud resources it is not priced; number and capacity of services in the cloud architecture are not limited except the constraints provided by providers. So, in this work, we assume that we have infinite number of cloud resources. For video streaming applications, we need storages to store data and CDNs to stream it. Today, cloud providers such as Amazon, Azure or Google provide cloud CDN solutions for its customers. Although they have different pricing strategies, they offer same services in common. Cloud CDN over traditional CDN streaming usage has mainly two advantages;

- One can customize his CDN infrastructure without the high cost of owning or operating geographically dispersed data centers as in another cloud service.
- Small companies can rent cloud CDN service from cloud storage providers and adopt the pay-as-demand way to save the bill.

Besides necessity of CDN, by reason of the difference in user profile and devices, transcoding becomes very essential part for video streaming applications which is used generally to change format, bit rates or resize the frame of the video. Within this context, main sources that are allocated are CDNs, storages and transcoders. Furthermore, user video requests must be well-defined to allocate resources in minimum cost and maximum user satisfaction. Videos are characterized as two groups; video format characteristics and QoS characteristics. Video format characteristics are given as video type (HD or SD), and video format. QoS characteristics are security (https and encryption), latency (minimize the streaming time) and quality (selecting suitable format for the user situation (device type, rate of data transfer)).

We use a discrete time slot model, for the modelling and parameters, assumptions and constraints are given below respectively. Cloud architecture handled during this study is given in Figure 3.1.

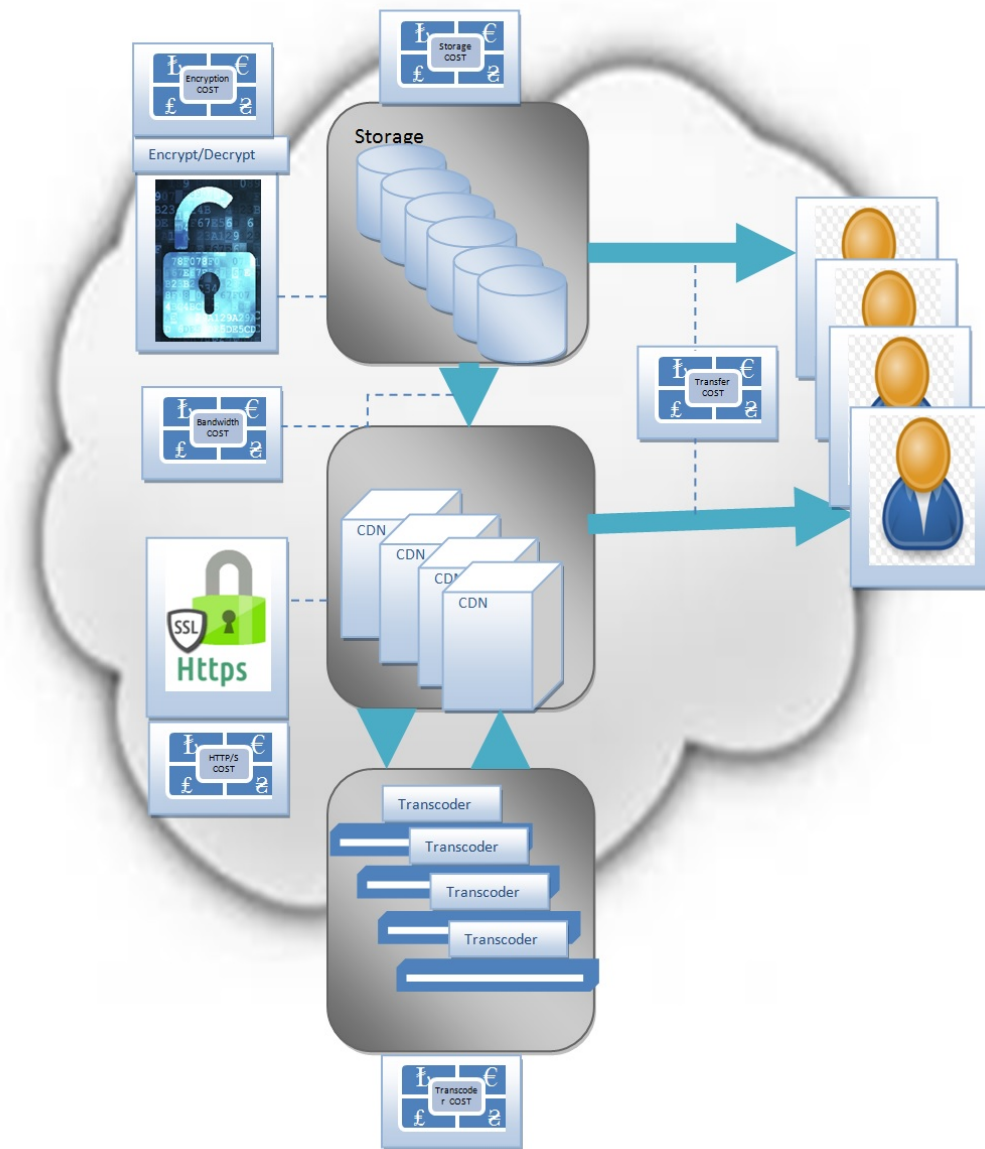


Figure 3.1: Architecture of Cloud Computing Systems



### 3.3.1 Parameters

- A set of storages  $S = \{s_1, \dots, s_s\}$  each one is used for storing videos.
- Each storage has specific characteristics; existence zone;  $zone(s_i)$ , bandwidth cost to internet and other cloud services per GB;  $b(s_i)$ , encryption cost;  $e(s_i)$
- A set of Content Delivery Network;  $CDN = \{c_1, \dots, c_c\}$ . Each one is used to stream video.
- Each CDN has specific characteristics; existence zone;  $zone(c_i)$ , bandwidth cost per GB;  $c(c_i)$ , streaming over secured http cost;  $h(c_i)$
- A set of Elastic Transcoder  $T = \{t_1, \dots, t_t\}$  each one is used to transcode video into desired format.
- Each Transcoder has specific characteristics; existence zone;  $zone(t_i)$ , bandwidth cost per GB;  $bt(t_i)$ , transcoding cost per minute;  $t(t_i)$
- A set of user video requests  $U = u_1, \dots, u_m$  in two different unit: Time unit and size unit. Since, cloud providers charged video transcoder per second and storages and CDN per GB.
- Each requested video has some characteristics: Type of video whether it is high definition (HD) or standard definition (SD);  $type(v_i)$  and duration of video;  $r(v_i)$ , frequency of video (number of video requested at time interval t);  $fr(v_i)$ , region of user;  $r(v_i)$ , binary info about whether video is transcoded or not, streamed over https or over http, encrypted or not and latency priority.
- The size of video is evaluated according to the demand from customers. The detailed information is given in Table 4.1.
- User video requests arrive in batches minutely. Duration of each video is evaluated according to aggregated normal distribution with parameters mean, variance and percentage of videos detailed in Table 4.2.

### 3.3.2 Assumptions and Constraints

- At time slot t, let  $n_{\delta t}$  different videos that are requested by users.
- The data transfer between Storage and CDNs in different zones in Amazon services is not charged; however, in Azure Cloud solutions data transfer cost is the same as from CDN to internet.
- HD and SD videos can be requested in different formats. So, cloud transcoder price will be used according to the format requested.
- In each time interval, the ratios of HD videos requested are 20% and SD videos are 80%.
- Play rate for SD video is 30 fps and HD video is 24 fps.
- There are 15 different video bit rates regarded. Some of them are detailed in Table 4.1.
- Each video is streamed from a unique storage or CDN.

- Each CDN has streamed maximum 10 Gigabits per second and 15,000 requests per second.
- Put, Copy, Post, List, or Get and all other Requests are ignored due to the low differences between regions in terms of pricing.
- In general, cloud service providers do not offer reserved prices for transferring from storages and CDNs in their web sites. So, differences between on-demand and reservation pricing are not considered.
- For each transcoder, at the same time, maximum 100,000 videos are transcoded.

Table 3.3: Symbols and Definitions

Symbol	Definition
$n$	Number of videos requested at a time interval $\delta t$
$m$	Number of CDNs used at a time interval $\delta t$
$s$	Number of storages used at a time interval $\delta t$
$t$	Number of transcoders used at a time interval $\delta t$
$x_{ij}$	Decision variable - $video_i$ is stored in $storage_j$ or not
$y_{ij}$	Decision variable - $video_i$ is streamed from $CDN_j$ or not
$z_{ij}$	Decision variable - $video_i$ is transcoded in $transcoder_j$ or not
$cost\_storage_j$	Cost of $storage_j$
$cost\_bandwidthstorage_j$	Bandwidth cost of $storage_j$
$size_i$	Total size of $video_i$
$zone_i$	Zone of $video_i$
$duration_i$	Total duration of $video_i$
$I\_latency_i$	Indicator of the latency for $video_i$ is important or not
$cost\_bandwidthstoragecdn_{jk}$	Bandwidth cost from $storage_j$ to $CDN_k$
$cost\_bandwidthstoragetrans_{jk}$	Bandwidth cost from $storage_j$ to $transcoder_k$
$cost\_bandwidthcdn_j$	Bandwidth cost of $CDN_j$ to Internet
$cost\_encryption_j$	Encryption cost of $storage_j$
$I\_encryption_i$	Indicator of the $video_i$ is encrypted or not
$frequency_i$	Number of video requested at a time interval $\delta t$
$cost\_https_j$	Https cost of $CDN_j$
$I\_https_i$	Indicator of the $video_i$ is streamed over https or http
$cost\_transcodersd_j$	Transcoding cost of $transcoder_j$ for SD videos
$cost\_transcoderhd_j$	Transcoding cost of $transcoder_j$ for HD videos
$I\_transcoderi$	Indicator of the $video_i$ is transcoded or not
$I\_type_i$	Indicator of the type $video_i$ (SD = 1 and HD = 0)
$v$	$n$ dimensional vector - Video characteristics
$v\_costbandwidthcdn$	$m$ dimensional vector- Bandwidth cost of CDN
$v\_costencryption$	$s$ dimensional vector- Encryption cost of CDN
$v\_costbandwithstorage$	$s$ dimensional vector - Bandwidth cost of storages
$D\_Latency$	$n \times n$ diagonal matrix - Diagonal entries of the latency of video

Table 3.3: Symbols and Definitions

Symbol	Definition
$D\_Size$	$n \times n$ diagonal matrix – Diagonal entries of the size of requested videos
$D\_Frequency$	$n \times n$ diagonal matrix – Diagonal entries of the frequency of requested videos
$D\_Duration$	$n \times n$ diagonal matrix – Diagonal entries of the duration of requested videos
$v\_costhttps$	$m$ dimensional vector - https cost of CDN
$v\_costtranscoder$	$t$ dimensional vector - Transcoding cost of Transcoders
$v\_encrypt$	$n$ dimensional vector - elements of the vector show video be encrypted or not
$v\_https$	$n$ dimensional vector - elements of the vector show video be streamed over https or not
$v\_transcoder$	$n$ dimensional vector - elements of the vector show video be transcoded or not
$C\_BandwidthStorageCDN$	$s \times m$ matrix – Bandwidth cost between storage and CDN
$C\_BandwidthStorageTranscoder$	$s \times t$ matrix – Bandwidth cost between storage and transcoder
X	$n \times s$ matrix - User control of video request assigned to storage or not
Y	$n \times m$ matrix – User control of video request assigned to CDN or not
Z	$n \times t$ matrix – User control of video request assigned to transcoder or not

### 3.3.3 Mathematical Modelling

In this study, we design a mathematical problem modelling represents the usage of cloud computing services in terms of cost while satisfying the constraints for the performance (security and latency) of the proposed VoD application. This problem is a kind of a multi-objective problem. We approach this issue as defining the cost as the main objective and QoS as constraints. We create a minimization optimization problem which the function includes the cost of security services cost (encryption cost, https cost), transfer cost between the services (to internet and between storage and CDN), CDN cost and transcoding cost. These are analyzed and explained separately in the following sub-chapters.

The symbols and definitions used for mathematical formulation are given in the following Table 3.3.

#### 3.3.3.1 Latency

It is generally reported that when we consider the QoS for VoD applications, latency is the main argument that are discussed by many researchers. So, to keep the customer persistence on the

usage of VoD services, elapsed time between the sending request and starting to watch for the video has to be minimized. This situation becomes more serious when serves application over cloud resources. On this wise, we have two distinct approaches. At the beginning of the study, we assume that each cloud data center stores all distinct videos uploaded. Although storage cost compared to bandwidth cost is relatively low, it also brings cost to financial budget or lessen profit of the organizations. So, by just storing videos on a part of data centers, we need to evaluate the latency to maximize user satisfaction. For the latency information, take the latency value from the video user's machine to the existing cloud data centers, and then continue with that data center locations who have first two levels of latency classes. For the second approach, to decrease the latency over cloud services, content delivery networks are used detailed in the next section. Thus, minimizing time to stream video is very important issue for these kind of services. Since public cloud services will be used during this work, minimizing latency problem over internet is another main problem which is also tried to be solved by many academic researchers. But, this problem is out of scope of our study. We try to minimize time to stream video by taking two approaches into consideration which are added to the problem formulation as constraints;

- We try to keep both user and sources in the same cloud region to reduce distance so time. First approach is to use latency data between the defined regions, shown in Table 4.6. This data is created as taken the average of the previous 24 hours of data collected. latency matrix between the region groups that categorizes latency into three class: Low (<100ms), Medium(100-180 ms) and High (>180ms).
- We use the CDNs for streaming videos to reduce latency. However, using CDNs increase the cost by just adding transfer cost between cloud services. So, according to user requirements, for each request, CDN usage will be decided dynamically.

### 3.3.3.2 Storage Cost

Storage for cloud services is considered as an IaaS service. Since the size of videos constitutes the big part of the storage, cost for this service is an important point. For the storage cost, there are two major pricing items;

- Storage pricing to store video according to different regions.
- Transfer pricing (bandwidth cost) to transfer video to the destination (internet or any other cloud service).

Since we evaluate cost dynamically and storage cost is priced per month, it is not added to instantaneous evaluation. But, it can be added to the cost analysis done for one month to see the total frame. Transfer pricing (bandwidth cost) constitutes the huge part of the total cost, charged according to size of the data transferred. By reason of the fact that cost function changes conforming to the destination of the video, if the data from storage is transferred to CDN, transcoders or directly to the user (internet) then the cost function becomes  $totalcost_{bandwidthstoragecdn}$ ,  $totalcost_{bandwidthstoragetrans}$ ,  $totalcost_{bandwidthstorage}$  respectively.

Also, two indicator function is created;  $I_{latency}$  and  $I_{transcoder}$ .  $I_{latency}$  expresses that whether latency is an important for the satisfaction of the customer of the video or not. As stated in

Section 3.3.3.1, we have two approaches to reduce latency time; selecting data centers who has low latency value and using CDN. In all cases, selecting low latency value is added as a constraint. However, since usage of CDN generates an additional cost, if this indicator function is equal to 1, video is transferred to CDN that lowers network latency.  $I_{transcoder}$  indicates that the video is needed to be transcoded into different formats. For the cloud services, transcoders are used to change the format of the video. So, if the video is transcoded then additional expense arises. These expenses cover two primary processes; transferring between storage and transcoder (bandwidth cost) and converting into different formats (transcoder cost). These expenses are analyzed in detail in Section 3.3.3.4.

$$\begin{aligned} totalcost_{bandwidthstorage} &= \sum_i (1 - I_{latency}(v_i)) \times size(v_i) \\ &\quad \times cost\_bandwidthstorage(zone(v_i)) \quad \forall i \in \Delta t \end{aligned} \quad (3.24)$$

$$\begin{aligned} totalcost_{bandwidthstoragecdn} &= \sum_i I_{latency}(v_i) \times size(v_i) \\ &\quad \times cost\_bandwidthstoragecdn(zone(v_i)) \quad \forall i \in \Delta t \end{aligned} \quad (3.25)$$

$$totalcost_{bandwidthstoragetranscoder} = \sum_i I_{transcoder}(v_i) \times size(v_i) \quad (3.26)$$

$$\begin{aligned} &\quad \times cost\_bandwidthstoragetrans(zone(v_i)) \\ &\quad \forall i \in \Delta t \end{aligned} \quad (3.27)$$

where

$$I_{latency}(v) = \begin{cases} 1 & \text{if latency is important} \\ 0 & \text{otherwise} \end{cases} \quad (3.28)$$

$$I_{transcoder}(v) = \begin{cases} 1 & \text{if the video needs to be transcoded} \\ 0 & \text{otherwise} \end{cases} \quad (3.29)$$

### 3.3.3.3 Content Delivery Network (CDN) Cost

For video streaming applications, CDN is an important concept and one of the basic terms for both traditional methods and cloud systems. CDN services are also provided for both on demand video content or live video. For the on demand video content, there are two download strategies; progressive download and streaming video. Progressive download means that CDN begins the delivering the download but a viewer can begin watching content within 3-5 seconds. In video streaming, video streams are divided into fragments; each fragment's length is between 2-10 seconds. Since cost is evaluated according to bit downloaded, the

second approach results in less cost for cloud users. Therefore, this strategy is considered in our study. Today, cloud providers such as Amazon, Azure, and Google provide cloud CDN solution for its customers. Although they have different pricing strategies, they offer same services in common. Cloud CDN usage has mainly two advantages;

- One can customize his CDN infrastructure without the high cost of owning or operating geographically dispersed data centers.
- Small companies can rent cloud CDN service from cloud storage providers and adopt the pay-as-demand way to save the bill.

There are two issues that must be considered for CDN and transcoder type of cloud resources. Firstly, we have to decide time to live value for CDN. Secondly, we have to decide whether keeping the different type of videos in storage or transcoded in each request which are argued in Section 3.1.

If service transfers big size data as in our case, CDN services are required. In addition to this, if video streaming service is provided, for streaming video with full HD quality, usage of CDN is inevitable [Maksiweb, 2017]. So, as mentioned before, overcoming the problem of latency and providing better quality video, CDN usage is necessary services for the services like video streaming services. For the CDNs, yet companies does not offer to select data centers of CDN, however they offer price classes. In each price class, maximum price is given as the price of that class. The details of these classes used in the case studies are given in Appendix A. Although CDN seems to be strictly necessary technology for the VoD applications, it also introduces additional cost which results in making a decision whether to use it or not. So, as in explained in previous chapter, we model our formulation based on the customer decision. If the latency is important for the video user, then CDN is to be put into use.

$$\begin{aligned}
 totalcost_{bandwidthcdn} &= \sum_i (I_{latency}(v_i)) \times size(v_i) \\
 &\quad \times cost\_bandwidthcdn(zone(v_i)) \\
 \forall i \in \Delta t & \tag{3.30}
 \end{aligned}$$

where

$$I_{latency}(v) = \begin{cases} 1 & \text{if latency is important} \\ 0 & \text{otherwise} \end{cases} \tag{3.31}$$

### 3.3.3.4 Transcoder Cost

Transcoders are very important for video streaming applications. To stream video, format of the video requested should be suitable for the user devices type. For the video transcoding cost, in addition to transcoding cost, bandwidth cost will be incurred while transferring data from storage to transcoder and back to storage. Thus, for the transcoder cloud services, there are two leading costs; transcoding cost and transferring cost between storage and transcoder which are  $totalcost_{transcoder}$ ,  $totalcost_{bandwidthstoragetrans}$  respectively. For the cloud transcoder, as

transcoding is charged per second, duration of the video plays an important role as long as the size of the video.

$$\begin{aligned}
totalcost_{transcoder} = \sum_i I_{transcoder}(v_i) \times dur(v_i) \times \\
((I\_type \times cost\_transcodersd)(zone(v_i)) + \\
((1 - I\_type) \times cost\_transcoderhd)(zone(v_i))) \\
\forall i \in \Delta t
\end{aligned} \tag{3.32}$$

$$\begin{aligned}
totalcost_{bandwidthstoragetrans} = \sum_i I_{transcoder}(v_i) \times size(v_i) \times \\
cost\_bandwidthstoragetrans(zone(v_i)) \quad \forall i \in \Delta t
\end{aligned} \tag{3.33}$$

where

$$I_{transcoder}(v) = \begin{cases} 1 & \text{if the video needs to be transcoded} \\ 0 & \text{otherwise} \end{cases} \tag{3.34}$$

Although we use public cloud services bring about infinite resources, according to cloud service providers, there are some constraints on the usage of services. For instance, for the Amazon, the number of video transcoded at the same time can be maximum 100.000. These constraints are detailed and handled during the implementation of case studies.

### 3.3.3.5 Security Options Cost

Encryption and watermarking are deeply critical two issues for the security of multimedia data. These are the services provided by most of cloud service providers. However, if we use Amazon Web Services (AWS) services, the usage of watermarking is not charged. So, we ignore this one for the Amazon Case Study. Since SSL does not change according to the zone and type of AWS services, it does not change the data center selection strategy so mathematical modelling does not include the cost of SSL usage; therefore it becomes static cost and which is also ignored. During this study, due to the above reasons, security options include only streaming video over https and, keeping video encrypted in the storage or not. These are examined and detailed in the following sub-chapters.

#### HTTP/S

CDN provides private content by signed URL or signed http cookie by user. CDN detect the user device and user geographical location, so suitable version of the requested file can be streamed and http/s is priced for each request and differs according to the region. If the customer requests video over secured http, then for each request, it is charged. Hence, the number of request for the video is added in to the formula of this additional cost. Besides, new characteristic function is defined that show the customer preference on the usage of https.

$$totalcost_{https} = \sum_i I_{https}(v_i) \times freq(v_i) \times cost\_https(zone(v_i)) \quad \forall i \in \Delta t \quad (3.35)$$

where

$$I_{https}(v) = \begin{cases} 1 & \text{if video } v \in https_{list} \\ 0 & \text{if video } v \notin https_{list} \end{cases} \quad (3.36)$$

Encryption

Customer master key is priced per month. Videlicet, we ignore it for each video requested in a time interval. If the multimedia file is encrypted, the cost equals the sum of coding cost and the requested number of decoding cost. Then cost becomes as the following;

$$totalcost_{encryption} = \sum_i I_{encrypt}(v_i) \times (freq(v_i)) \times cost\_encryption(zone(v_i)) \quad \forall i \in \Delta t \quad (3.37)$$

$$I_{encryption}(v) = \begin{cases} 1 & \text{if video } v \in encrypted_{list} \\ 0 & \text{if video } v \notin encrypted_{list} \end{cases} \quad (3.38)$$

### 3.3.3.6 Formulation

In each time interval, we define number of video requests as;

$$v_{\delta t} = (v_1, v_2, \dots, v_n)_{\delta t}$$

Each video has some characteristics;

$$v_i = (time, size, loc, format, encrypt, https, frequency, latency)$$

Latency is a big problem for the cloud services that does not guarantee to the customers to get the video in right time. But, if the latency is an important issue for the customer, then CDN usage becomes compulsory. Assume that at time interval  $\delta t$ , there are n different video requests exists and assume that m different type of content delivery networks (CDN), s different type of storages and t different type of transcoders exists. We should decide the values of three binary variables x, y and z where

$x_{ij}$  show us  $i^{th}$  video is requested form  $j^{th}$  storage,

$y_{ij}$  show us  $i^{th}$  video is requested from  $j^{th}$  CDN,

$z_{ij}$  show us  $i^{th}$  video is transcoded in  $j^{th}$  transcoder.

The problem can be stated as: "Assign each video request into cloud resources, such that



while minimizing the used storage, CDN and transcoder costs with the best performance that is satisfying quality of service requirements of the user such as high video quality, security and time efficiency.” This problem formulation becomes a quadratic problem in which objective function contains product of two decision variables. Since, there are quadratic terms in the objective function, the problem is termed a Mixed Integer Quadratic Program (MIQP).

### General representation of the mathematical model:

**Variables :**

$$x_{ij}, y_{ij} \text{ and } z_{ij}$$

**Parameters :**

$$I_{https}(v) = \begin{cases} 1 & \text{if video } v \text{ is streamed over http secure} \\ 0 & \text{otherwise} \end{cases} \quad (3.39)$$

$$I_{trans}(v) = \begin{cases} 1 & \text{if the video needs to be transcoded} \\ 0 & \text{otherwise} \end{cases} \quad (3.40)$$

$$I_{encrypt}(v) = \begin{cases} 1 & \text{if video } v \text{ is encrypted} \\ 0 & \text{otherwise} \end{cases} \quad (3.41)$$

$$I_{latency}(v) = s \begin{cases} 1 & \text{if latency is important} \\ 0 & \text{otherwise} \end{cases} \quad (3.42)$$

**Objective Function:**

$$J_{\delta t} = (cost_{bandwidth} + cost_{encrypt})^{Storage} + (cost_{bandwidth} + cost_{http/s})^{CDN} + (cost_{transcoder})^{Transcoder} + (cost_{bandwidth})^{Storage-CDN} + (cost_{bandwidth})^{Storage-Transcoder} \quad (3.43)$$

$$J_{\delta t} = \sum_i \sum_j \sum_k \sum_l (1 - I_{latency}(v_i)) \times size(v_i) \times cost_{bandwidthstorage}_j(zone(v_i)) + I_{encryption}(v_i) \times (frequency(v_i)) \times cost_{encryption}_j(zone(v_i)) + I_{https}(v_i) \times cost_{https}_k(zone(v_i)) \times frequency(v_i) + I_{latency}(v_i) \times cost_{bandwidthCDN}_k(zone(v_i)) \times size(v_i) + I_{transcoding}(v_i) \times cost_{transcoding}_l(type(v_i)) \times duration(v_i) + I_{latency}(v_i) \times cost_{bandwidthstoragecdn}_{jk}(zone(v_i)) \times size(v_i) + I_{transcoder}(v_i) \times cost_{bandwidthstoragetranscoder}_{jl}(zone(v_i)) \times size(v_i) \quad \forall i \in \Delta t \quad (3.44)$$

such that

Constraints:

$$\begin{aligned}
\forall i \in [1, n], \quad j \in [1, m] \quad \sum_{i=1}^n \sum_{j=1}^m y_{ij} &\leq 100,000, \quad y_{ij} \in 0, 1 \\
\forall i \in [1, n], \quad j \in [1, t] \quad \sum_{i=1}^n \sum_{j=1}^t z_{ij} &\leq 3,840, \quad z_{ij} \in 0, 1 \\
\forall j \in [1, t], \quad \sum_{i=1}^n y_{ij} &\leq 20,000, \quad z_{ij} \in 0, 1 \\
\forall j \in [1, m], \quad \sum_{i=1}^n z_{ij} &\leq 480, \quad z_{ij} \in 0, 1
\end{aligned} \tag{3.45}$$

Variable Restrictions:

$$\begin{aligned}
\forall i \in [1, n], \quad \sum_{j=1}^s x_{ij} &= 1, \quad x_{ij} \in 0, 1 \\
\forall i \in [1, n], \quad \sum_{j=1}^c y_{ij} &\leq 1 \vee 0, \quad y_{ij} \in 0, 1 \\
\forall i \in [1, n], \quad \sum_{j=1}^t z_{ij} &\leq 1 \vee 0, \quad z_{ij} \in 0, 1
\end{aligned} \tag{3.46}$$

### 3.3.3.7 Linear Algebra Representation

The parameters used during the linear algebra representation of the problem are given in Table 3.3 in detail. Eventually, the total cost for streaming video is defined as the following;

$$\begin{aligned}
J_{\delta t} = & v((X)^T \times D\_Latency) \times v\_costbandwidthstorage) + (v\_costencryption \times (X)^T \times D\_Frequency) \\
& \times v_{encryption}) + v((Y)^T \times D\_Latency) \times v\_costbandwidthcdn) + ((v\_costhttps \times (Y)^T \times \\
& D\_Frequency) \times v\_https) + ((v\_costtranscoder \times (Y)^T \times D\_Duration) \times v\_transcoder) + \\
& trc(((X)^T \times D\_Size) \times Y) \times (C\_BandwidthStorageCdn)^T) + \\
& trc(((X)^T \times D\_Size) \times Z) \times C\_BandwidthStorageTranscoder^T)
\end{aligned}$$

$$\begin{aligned}
J_{\delta t} = & \sum_{i=1}^n \left( \sum_{j=1}^s (x_{ij} \times \text{cost\_bandwidthstorage}_j) \times \text{size}_i \times (1 - I\_latency_i) \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^m (y_{ij} \times \text{cost\_bandwidthcdn}_j) \times \text{size}_i \times I\_latency_i \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^s (x_{ij} \times \text{cost\_encryption}_j) \times I\_encryption_i \times (\text{frequency}_i) \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^m (y_{ij} \times \text{cost\_https}_j) \times I\_https_i \times \text{frequency}_i \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^l (z_{ij} \times (\text{cost\_transcodingsd}_j \times (1 - I\_type)) + (\text{cost\_transcodinghd}_j \right. \\
& \quad \left. \times (I\_type)) \times I\_transcoding_i \times \text{duration}_i \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^s \left( \sum_{k=1}^l (x_{ij} \times z_{ik}) \times \text{cost\_bandwidthstoragetranscoder}_{jk} \right) \times \text{size}_i \times I\_transcoder_i \right) + \\
& \sum_{i=1}^n \left( \sum_{j=1}^s \left( \sum_{k=1}^m (x_{ij} \times y_{ik}) \times \text{cost\_bandwidthstoragecdn}_{jk} \right) \times \text{size}_i \times I\_latency_i \right)
\end{aligned}$$

### 3.3.4 Computational Complexity of the Problem

After the problem is defined, problem properties should be analyzed to select appropriate solution methodologies. NP-completeness cannot be directly applied to optimization problems; it should be resolved on a decision problem. Since our problem defined is an optimization problem, we need to cast it as decision problem. Computational Complexity focuses on decision problem since classifying problem complexity of them is easier than optimization problems. It is known that, any optimization problem can be transformed to a decision problem by imposing a bound on the objective value. The answer of the decision problem should be simply ‘yes’ or ‘no’ or 1 or 0 as in our case. After transforming to decision problem, to prove that our problem is NP hard problem, the steps below must be followed:

1. Show the problem is in NP class.
  2. Reduce the known NP complete problem P' to our problem P through a poly-time algorithm f, i.e. f has polynomial time complexity.
- We transform the minimization optimization problem to the decision problem for examining whether cost of a solution is lower than or equal to a value  $\mathbf{X}$ .

$$\forall v \in V \quad \sum_{a \in A', d_r(a)=t} 1 \leq 1, \forall t \in T \quad \sum_{a \in A, u_r(a)=t} m_r(a) \leq e_r(t), \sum_{a \in A} g_r(a) \leq \mathbf{X} \quad (3.47)$$

- V is the number of all videos
- A is the set of all allocations to cloud resources

- $A'$  is the compared problem solutions  $A' \subset A$
- $T$  is the set of all Resources
- $m_r(a)$  is the total number of resources used for allocation  $a$ ,  $m_r(a) \in Z^+$
- $u_r(a)$  is the resources used for allocation  $a$ ,  $u_r(a) \in T$
- $d_r(a)$  is the target resources used for allocation  $a$ ,  $u_r(a) \in T$
- $e_r(t)$  is the number of existing resources in resource  $t$ ,  $e_r(t) \in Z^+$
- $d_r(t)$  is the number of existing resources in resource  $t$ ,  $e_r(t) \in Z^+$
- $g_r(a)$  is the total cost for allocation  $a$ ,  $g_r(a) \in R^+$

The cost  $g_r(a)$  is the multiplication of the cost of the resource  $t$  and the size of video  $v \in V$ . Decision problem of our case is stated as "Given a set of video requests  $V = 1, 2, \dots, n$  with sizes  $s_1 \geq s_2 \geq \dots \geq s_n$  and set of cloud resources and non-negative integer  $\mathbf{X}$ , are there some number of sets  $S_1, \dots, S_\nu$  called bins, such that  $\sum_a f(|S_i|) \leq \mathbf{X}$  is satisfied."

- To prove that our problem is NP hard problem, we should state that problem is in NP class. To decide simply a decision problem is in NP, for any input for which the answer is yes, the correct answer should be verified in polynomial time.

Let  $x$  be an instance of the problem P. Apply the polynomial function  $f$  which is objective function and if the result is smaller than the value  $\mathbf{X}$ , and it does not exceed the capacity of resources, then we get the result of instance is 'yes' in a polynomial time.

- Find the known NP hard problem  $P'$  that is reduced to our problem P through a poly-time algorithm  $f$ . To reduce problem P to another problem  $P'$  if any instance of P can be stated as an instance of  $P'$ . The reduction function maps any instance  $x$  of P represented by  $L_1$  language to  $f(x)$  instance of  $P'$  represented by language  $L_2$ . Besides, for each instance  $x$  of P, the answer of instance  $x$  is 'yes' if and only if the answer of instance of  $f(x)$  is 'yes'.

We reduce it from the Generalized Bin packing problem which is proved that it is not only NP hard problem but also finding a feasible solution is NP hard as well [Baldi and Bruglieri, 2017]. In the generalized bin packing decision problem, there are  $n$  items of different weights and profits and  $m$  various bins each of capacity and cost, assign each item to a bin such that total cost is minimized [Baldi and Bruglieri, 2017]. It may be assumed that all items have weights smaller than bin capacity. In our problem, we have  $t$  number of resources which relates to  $m$  number of bins and  $v$  number of video requests of different size relates to  $n$  items of different weights.  $\mathbf{X}$  is the objective value. The aim of our problem is to find minimum cost spend on cloud resources corresponds to the minimum cost of bins in the known NP hard problem.  $u_r$  is the total number of resources used represents the weight of bins in NP problem. We should also show that result of any instance of  $x$  of P is 'yes' iff the result of instance  $f(x)$  in problem  $P'$  is 'yes'. Without loss of generality, we order the resources according to the cost of each resource such that  $t_1 \leq t_2 \leq t_3 \dots \leq t_t$ . Let there is an instance  $x$  that the cost of that  $x$  has value smaller than  $\mathbf{X}$ . To reach the minimum cost, the most sized videos are assigned to resources with has lowest cost and since each of request is 1 then there is a density to the first  $n$  resources having lower cost. Then trivially, this results into minimum cost of bins used for the bin packing problem. In the opposite case, if the optimal solution that finds minimum cost of bins; which is used to put items into bins and items. Then, without loss of generality, in our problem domain, we order resources from lowest cost

to the highest cost and order the items from larger sizes to the smaller sizes. After all, assign each ordered bin to the ordered resources respectively and assign the items in the bin to the videos whose sizes more respectively, then we get the optimal solution in our problem P.

Also, specific variants of the min-max knapsack problem were also explored in the study of Kasperski et al. for the case where the item sizes are all equal to 1, and one has to choose a given number of items so as to minimize the total cost under the existing scenarios. In this study, they showed that the problem is not approximable within a constant factor unless  $P = NP$  [Pinto et al., 2015].

### 3.4 Solution Approaches to the Proposed Mathematical Modelling of the Defined Problem

#### 3.4.1 Integer Linear Programming

The first linear programming formulation of a problem that is equivalent to the general linear programming problem was given by Leonid Kantorovich who also proposed a method for solving it in 1939 [Schrijver, 1986, Kennedy and Eberhart, 1997]. The linear programming is a model of constraint optimization and applied for determining the optimal allocation of such resources while maximizing/minimizing profit/cost under some constraints. It is used widely in many areas such as business, economics and engineering. Linear programming problem construction contains three steps: Variables, Constraints and Objective function. Constraints and objective function must be linear function or linearizable function. In matrix notation, linear program is as the following;

$$\begin{aligned} & \text{maximize } c^T x \\ & \text{subject to } Ax \leq b \end{aligned} \tag{3.48}$$

Where  $x$  represents variables,  $c$  and  $b$  are known as coefficients,  $A$  is a matrix of coefficients and  $(.)^T$  is the matrix transpose. The expression to be maximized or minimized is called the objective function ( $c^T x$  in this case). The inequality  $Ax \leq b$  is the constraint in which the objective function is to be optimized over. In the first part of this study, it is assumed that transfer cost between cloud services equal to zero. Hereby, function becomes linear and solved by using linear programming by using simplex method. There are many programs exist in industry to solve linear problems. In this study, we used the java programming language and IBM Ilog Cplex for the simulation of our problem and to solve the mathematical problems.

---

**Algorithm 1: Linear Programming (LP) Algorithm.**

---

- 1: Set the costs of all cloud resources stated in the architecture of cloud
  - 2: Set the constraints that must be satisfied
  - 3: For all requested videos  $v_i \in V(t)$
  - 4:     Evaluate defined costs using equation defined in Section 3.3
  - 5: End For
  - 6: Apply the linear programming algorithm
  - 7: Return the minimum cost
- 

### 3.4.2 Mixed Integer Quadratic Programming

The quadratic programming, a special version of mathematical optimization technique that the function must be minimized or maximized, is a quadratic function of several variables under linear constraints. Due to the transfer cost between two resources (Storage-CDN or Storage-Transcoder), our problem function becomes quadratic function which contains the multiplication of two variables. Ultimately, our mathematical optimization problem formulated in previous chapter become linearly constraint quadratic optimization problem. Due to the domain of the cost function is binary, then it is not a convex set. However, the cost function is convex, then we make continuous relaxation binary domain and make it convex set  $[0,1]$  to solve convex quadratic programming. Since domain is binary, special application of branch and cut method is applied to this problem. Although there are many exact solution methods are proposed for this problems, branch and cut method is used to solve the problem due to the fact that branch and cut technique combines the reliability advantage of Branch and Bound with the fastness of Gomory Cutting Planes scenery [Albert, 1999]. This method is based on the partition of feasible set into smaller subsets of solutions which are evaluated until the best solution is found. The details of the algorithm applied are given in the Algorithm 2.

### 3.4.3 Binary Particle Swarm Optimization

There are many exact solution suggestions for the linear programming algorithms which provide the best solution unlike heuristics algorithms. However, for the real and big data, solving this problem by using branch and cut method is very time consuming, which necessitates the use of other algorithms. Researchers apply different allocation methods and implement wide variety of heuristic algorithms in the literature. One of these heuristics algorithms is the Particle Swarm Optimization (PSO) which simulates social behavior of flock of birds. PSO algorithm works by having population of candidate solutions that travels in the problem space to reach optimum solution [Jamali et al., 2016]. It was proposed by Kennedy and Elbart in 1995. PSO was used in articles for task scheduling and job scheduling respectively and there are many researches use particle swarm optimization technique to tackle with optimization problems [Salman et al., 2002, Zhang et al., 2008, Prasad et al., 2009, Hu and Eberhart, 2002]. This kind of problems consist of three basic components: variables, fitness function and constraints that specify feasibility of variables. Although genetic algorithm (GA) is as popular as PSO, especially in terms of convergence and complexity of the parameters of the algorithms, usage of PSO is a little ahead [Karpas and Özel, 2006]. This technique has easier implementation and need less parameter than other heuristic algorithms. But, of course, there

are points to pay attention to.

One of them is that defining particle is very crucial step to maximize efficiency of the algorithm. In our study, there exists  $s$  number of storages,  $c$  number of CDNs and  $t$  number of transcoders for  $n$  number of video streaming applications. There are two different approaches in this case. In the first approach, we design algorithm as one particle (multi-dimensional) which attaches all different resource particles. So, in this way, there is only one particle exists and classical binary PSO is applied. In the second approach, since we have three decision variables, three different particles are defined as matrix in  $n \times s$  dimension,  $n \times c$  dimension and  $n \times t$  dimension respectively. Each cell of matrix is 1 or 0 and since each task is executed on only one storage, CDN and transcoder, each row is binary unit vector. In this situation, multi-swarm is applied. In each step, particle position is evaluated according to fitness function; cost function. In our case, nature of each particle is designed as binary (0-1) that shows whether it is assigned to storage or not. But, classic PSO is designed for continuous data. Due to the nature of our study, we have to implement binary version of PSO. The developers of continuous PSO algorithm also proposed reworking of the algorithm to operate on the binary variables. Position of particles changes in the probability of coordinates of position takes 1 or 0 value.

However, binary PSO is not effective as continuous binary PSO due to the problems defined especially in selecting inertia weight value in [Khanesar et al., 2007]. So, to increase the effectiveness of the solutions, Novel binary PSO proposed in the study of Khanesar et al. is used [Khanesar et al., 2007]. This proposed algorithm yields better results than the binary version of the PSO. Yet, in terms of execution time, since the proposed novel PSO algorithm is much more complex, taking all resources as one particle results in worse results from the standpoint of the execution time of the algorithm. Taking each resource as one swarm also improves the results of the algorithm in the light of two decisive points of the performance of the algorithm; execution time and accuracy. Since we have three variables that must be decided, we design particles as the combination of these variables as one particle and design our PSO as one swarm. Because, fitness value contains these three variables cost separately as well as multiplication of these variables. So, considering them as multiple swarms is not proper enough for handling fitness value completely. Moreover, in our case, the problem has some set of constraints which are detailed in the Section 3.3. Constrained optimization is also another problem that needs to modify PSO algorithm. With the multi-swarm approach, methods used to handle these two issues, binary and constrained, are detailed from section 3.4.4 to section 3.4.7.

---

**Algorithm 2:** MIQP with Branch and Cut Algorithm

---

- 1: Set the costs of all cloud resources stated in the architecture of cloud
  - 2: Set the constraints that must be satisfied
  - 3: for all time  $t \in \delta t$ 
    - 1: Tree is initialized with the root node
    - 2: Continues relaxation of binary set to [0,1]
    - 3: Initial node solution of convex quadratic programming is achieved in polynomial time
    - 4: Branching is occur
    - 5a: One node is 0 lower bound.
    - 6a: If the solution of the node satisfies constraints and value of objective function of the node is better than previous incumbent
    - 7a: New value of incumbent becomes the solution
    - 8a: If not
    - 9a: Branching will continue with the variable that not satisfies integrality constraint
    - 5b: One node is 1 upper bound.
    - 6b: If the solution of the node satisfies constraints and value of objective function of the node is better than previous incumbent
    - 7b: New value of incumbent becomes the solution
    - 8b: If not
    - 9b: Branching will continue with the variable that not satisfies integrality constraint
  - 10: At each point of algorithm, there is best node whose value is minimum than all the others
  - 11: Evaluate the MIP Gap value between best node value with the current incumbent value.
  - 12: If MIP Gap value becomes lower than 0,0001
  - 13: Terminate the search
  - 14: Else
  - 15: Continue with the step of 4 16:
- end  
for
- 17: Return the minimum cost
- 

### 3.4.4 Novel Binary Particle Swarm Optimization

In this binary model, particles best and global best of the particles are defined as in the classic version. And velocity is updated as in continuous case:

$$x_{ij}(t+1) = w \times v_{ij}(t) + c_1 \times rand() \times (pb_{ij}(t) - x_{ij}(t)) + c_2 \times rand() \times (gb_{ij}(t) - x_{ij}(t)) \quad (3.49)$$

where  $w$  is the inertia,  $c_1$  and  $c_2$  are learning factors called cognitive and social scaling parameters respectively.  $pb$  is the best value of that particle called particle best and  $gb$  is the best value of the swarm called global best. However, velocities of the particles are defined as probabilities that a bit will change to 1 or 0. According to this definition velocity interval is in range [0 1]. New position of particle is defined as:



$$x_{ij}(t+1) = \begin{cases} 1 & \text{if } r_{ij} < \text{sig}(v_{ij}(t+1)) \\ 0 & \text{otherwise} \end{cases}$$

where  $r_{ij}$  is a uniform random number in the range of (0, 1)

The implementation of the discrete version of binary PSO has many drawbacks and does not give efficient results like as in the continuous version. Nonetheless, during this work, Novel binary PSO proposed in Khanesar et al. is used to solve our problem [Khanesar et al., 2007]. Best particle position and best global position is updated as in continuous or binary version. The difference is velocity interpretation. Two vectors for each particle are defined:  $\vec{v}_i^0$ ,  $\vec{v}_i^1$ .  $\vec{v}_i^1$  is the probability of the particles bits to change 1 while  $\vec{v}_i^0$  is the probability to change 0. Since these are not complementary, velocity function is defined as;

$$V_{ij} = \begin{cases} V_{ij}^0 & \text{if } x_{ij} = 0 \\ V_{ij}^1 & \text{if } x_{ij} = 1 \end{cases}$$

Velocity update is as follows; let  $p_{ibest}$  is the best position of the particle  $i$  and  $p_{gbest}$  is the best position of all particles. Also consider the  $j^{th}$  bit of  $i^{th}$  best particle is 1. So, the velocity of change to one ( $\vec{v}_{ij}^1$ ) for that particle increases and the velocity of change to 0 ( $\vec{v}_{ij}^0$ ) decreases [Khanesar et al., 2007]. The velocity update is defined as;

$$\begin{aligned} \text{If } p_{ibestij} = 1 & \quad \text{Then } d_{ij1}^1 = c1r1 \quad \text{and} \quad d_{ij1}^0 = -c1r1 \\ \text{If } p_{ibestij} = 0 & \quad \text{Then } d_{ij1}^0 = c1r1 \quad \text{and} \quad d_{ij1}^1 = -c1r1 \\ \text{If } p_{gbestij} = 1 & \quad \text{Then } d_{ij2}^1 = c2r2 \quad \text{and} \quad d_{ij2}^0 = -c2r2 \\ \text{If } p_{gbestij} = 0 & \quad \text{Then } d_{ij2}^0 = c2r2 \quad \text{and} \quad d_{ij2}^1 = -c2r2 \end{aligned}$$

Where  $d_{ij}^0$ ,  $d_{ij}^1$  are two temporary values,  $r1$  and  $r2$  are two random numbers in the range of [0, 1] and  $c1$  and  $c2$  are two fixed variables acceleration coefficients which are determined by user. Then velocity is updated according to the following equation;

$$\begin{aligned} V_{ij}^0 &= wV_{ij}^0 + d_{ij1}^0 + d_{ij2}^0 \\ V_{ij}^1 &= wV_{ij}^1 + d_{ij1}^1 + d_{ij2}^1 \end{aligned}$$

We also add local neighbourhood topology (ring topology) while updating velocity. Only a specific number of particles can affect the velocity of a given particle. Then velocity update equation becomes as the following;

$$\begin{aligned} V_{ij}^0 &= wV_{ij}^0 + d_{ij1}^0 + d_{ij2}^0 + d_{ij3}^0 \\ V_{ij}^1 &= wV_{ij}^1 + d_{ij1}^1 + d_{ij2}^1 + d_{ij3}^1 \end{aligned}$$

where

$$\begin{aligned} \text{If } p_{nbest}^j = 1 & \quad \text{Then } d_{ij3}^1 = c3r3 \quad \text{and} \quad d_{ij3}^0 = -c3r3 \\ \text{If } p_{nbest}^j = 0 & \quad \text{Then } d_{ij3}^0 = c3r3 \quad \text{and} \quad d_{ij3}^1 = -c3r3 \end{aligned}$$

where  $r_3$  is random number in the range of  $[0, 1]$  and  $c_3$  is the fixed variables defined by user.  $c_1$ ,  $c_2$  and  $c_3$  are particle, global and neighbourhood increments respectively. The sigmoid function is used to normalize velocity function. There is a new acceleration coefficient defined for the neighbourhood topology. Besides influenced by global and own best value, it is also affected by the neighbourhoods with in a size of two (ring topology). Particle position update is as the following;

$$x_{ij}(t+1) = \begin{cases} \overline{x_{ij}(t)} & \text{if } r_{ij} \leq v_{ij} \\ x_{ij}(t) & \text{if } r_{ij} > v_{ij} \end{cases} \quad (3.50)$$

Pseudo-code for the proposed algorithm is given in Algorithm 3.

---

**Algorithm 3:** Proposed PSO Algorithm

---

```

1: Initialize time to zero
2: While time <= maxTime
3:   Call Data_Generate_Function
4:   Initialize the acceleration coefficients,  $c_1$ ,  $c_2$  and  $c_3$ 
5:   Define maximum number of iterations; maxIterations, neighbourhood size, size of the swarm.
6:   Do Until  $i = n$ 
7:     Call initialize_swarm_function
8:   End Do
9:   while number of iterations < maxIterations do
10:    Do While  $i=1, \dots, n$ 
11:      evaluate position of best solution found by particle  $i$ 's neighbourhood so far
12:      evaluate position of best solution particle  $i$  has found so far
13:      evaluate position of global best solution has found so far
14:      evaluate velocity and position of the particle  $i$ 
15:    End While
15:    Update particle_best, global_best and neighbourhood_best particle
16: End While

```

---

### 3.4.5 Constrained Particle Swarm Optimization

Constrained optimization algorithms, which are inherently problematic in the real life, have been a topic of study for many years. Like all the other optimization algorithms, the original PSO method needs to be modified to handle constraints. Koziel et al. grouped constrained PSO into four categories clearly: methods based on preserving feasibility of solutions; methods based on penalty functions; methods that make a clear distinction between feasible and infeasible solutions; and hybrid methods [Koziel and Michalewicz, 1999]. During this study, swarm is initialized from the feasible set of solutions which is the number of resources dimensional orthonormal unit binary vectors. After updating position of the particles, they can stay outside of the domain and/or can not satisfy the constraints caused by the service providers. If the updated solution does not satisfy the constraints, then to discard this particles, we define penalty value to the fitness value of that particle. Since the problem is a minimization

optimization problem, giving enough high value for that solution can automatically eliminates those particles. Also, after updating the particles, particles can easily become out side of the domain due to the binary and unit nature of the particles. After updating particles (velocities) using the equations in the study of Khanesar et al., variables may be outside of the domain [Khanesar et al., 2007]. So, we define an adjustment method to keep the particles in the domain as stated in the thesis of Fraser [Lin, 2005]. To keep particles inside of the domain, two different strategies are adopted. The first strategy is as follows;

- If the updated position of the particle is out of domain, position of the global best or neighbourhood best particle is taken.

The second strategy is as follows;

- If the updated particle is out of domain i.e. particle contain more than one 1, since each particle has at most one 1, then select randomly one of them and assign 0 to all other 1s.
- If the updated particle is out of domain i.e. particle does not contain any 1, then randomly select the position and assign 1 to that position of the particle.

The first strategy of the adjustment method does not give enough good results as expected. On account of the fact that the first strategy triggers the one of the main drawback which is falling into local optima of PSO. In brief, randomly selecting the position of the particle improves the exploratory of the search ability of the algorithm. Details of the algorithm are given in Algorithm 4.

---

**Algorithm 4:** Particle Update

---

- 1: Repeat
  - 2: For each particle  $j$  in  $n$ ; *numberofvideos*
  - 3: Update Velocity and Particle according to type
  - 4: If particle not in Domain  $D$
  - 5: If particle does not contain any 1s
  - 6: Assign 1 to the randomly chosen position of the particle
  - 7: If particle contains 1 more than one.
  - 8: set 0 to all 1s and assign 1 to the randomly chosen position  
of the particle
  - 9: End for
- 

Although this approach overcomes the in-feasibility of the solution, it complicates the algorithm. As a result of this, execution time suffers from this situation. Then, we propose a new approach which is the division of swarm into sub-swarms. Due to the different type of resources, this approach was quite appropriate for the problem area constructed during this study. Thus, PSO is modified as multi-swarm PSO which is detailed in the following section.

### 3.4.6 Multi-Swarm Binary Particle Swarm Optimization with Greedy Heuristic Algorithm

Particularly, for solving the multimodal function optimization like in our situation, balancing between global exploration and local exploitation and keeping diversity of the particles of PSO is inappropriate [Ye et al., 2017]. To overcome this problem, we introduce multi-swarm into PSO by just separating several system optima from each other. Since, we have three distinct sources that must be used to allocate requested videos, we define three swarms, and each one optimizes the cost of each resource. To handle the cost between these resources, we use greedy algorithm which is applied to reach the optimum solution between these swarms. In spite of the fact that greedy algorithm arrives likely local best solution rather than global best solution, it reduces the execution time drastically [Kiziloz et al., 2018]. Since due to the nature of our problem domain, using evolutionary algorithm instead of greedy algorithm is unseemly and pointless, we implement the greedy between swarms to increase time efficiency. Details of this technique are represented in the Figure 3.2.

Since, we have three distinct sources that must be used to allocate requested videos, we define three sub swarms, and each one optimizes the cost of each resource. To handle the cost between these resources, first best n solutions of the swarms are taken and greedy algorithm is applied to reach the optimum solution between them. Even though the transfer (bandwidth) cost between the resources constitute the big part of the cost, separating them and evaluating these cost outside of the optimization algorithm yields better results in terms of both accuracy and execution time. Because, the cost of the one resource is proportional to the cost of the other resource in the same region. In short, if a resource in region A has the highest charge, then the other resource in region A also has a highest charge. This correlation brings about good results at the proposed algorithm. The pseudo-code of the algorithm is given in Algorithm 5.

---

**Algorithm 5: Multi-Swarm PSO Algorithm**

---

- 1: Initialize time to zero
  - 2: While time  $\leq$  maxTime
  - 3:     Call Data\_Generate\_Function
  - 4:     Initialize the acceleration coefficients, c1, c2 and c3
  - 5:     Define maximum number of iterations; maxIterations
  - 6:     Initialize multi dimensional array bestParticles within size of resource type size\_resource times best n values
  - 7:     Do Until i = size\_resource
  - 8:         Call Swarm\_PSO with resource(i)
  - 9:         Return bestParticles
  - 10:     End Do
  - 11:     Call Greedy\_Algorithm with bestParticles
  - 12: End While
- 

Greedy algorithm obtains the locally optimal choice at that moment expecting to find global optimum solution. In our case, pseudo-code of the algorithm is as;

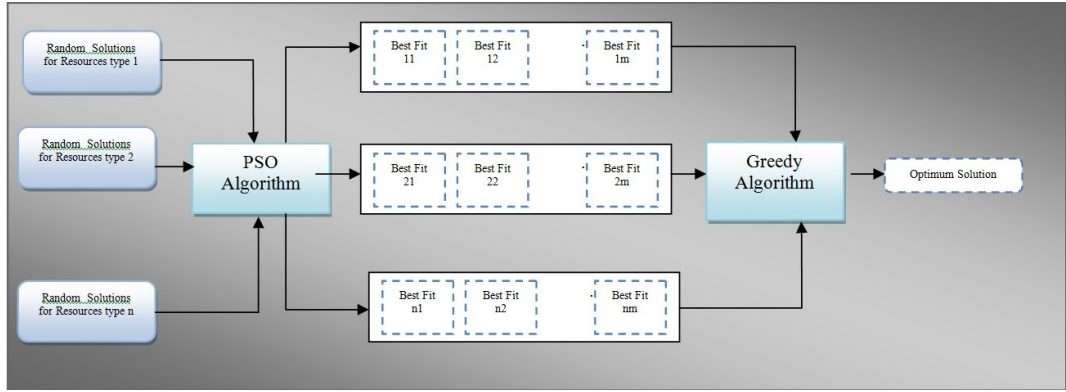


Figure 3.2: Multi Swarm PSO with Greedy Algorithm

---

**Algorithm 6:** Greedy\_Algorithm

---

- 1: Input bestParticles
  - 2: Do Until i size\_resource
  - 3:   Select randomly particle in the bests of  $i^{th}$  resource type
  - 4:   Evaluate the cost between the selected particle of  $i^{th}$  resource type and all particles of  $(i + 1)^{th}$  resources
  - 5:   Select the particle of  $(i + 1)^{th}$  resource type whose transfer cost from the previous selected particle is minimum.
  - 6: End Do
  - 7: Evaluate the total cost by using the selected particles
  - 8: Return minimized total cost
- 

### 3.4.7 Parallel Multi-Swarm Binary PSO

Although PSO is efficient when compared to other evolutionary algorithms, when the swarm size increases and multi dimensional problems exist, then it still suffers from performance loss. In the previous section, we propose an efficient modification of classic PSO to increase the performance. In this section, the proposed algorithm is extended by using the parallelization technique. Because, the implementation of parallel programming into the proposed algorithm is an inevitable process to decrease computing time of the algorithm. Instead of executing each sub-swarm sequentially, each sub-swarm PSO are executed concurrently. In summary, each resource (services) is considered as a one objective, then the problem becomes multi-objective and each objective is solved by using only one swarm. At the end of the parallel algorithm, they exchanges information with each other and by applying greedy algorithm into these outputs, optimum solution is achieved. Java multi-threading and concurrency is used to process each PSO concurrently. The pseudo-code and the figure for the parallel programming algorithm is given below;

– **Algorithm 7:** Parallel Multi-Swarm PSO with Greedy Algorithm

---

- 1: Create size\_resource worker threads
  - 2: Do Until size\_resource
  - 3:     Call Data\_Generate\_Function
  - 4:     Initialize the acceleration coefficients, c1, c2 and c3
  - 5:     Call Greedy\_Algorithm
  - 6:     Define maximum number of iterations; maxIterations, Neighbourhood size, size of the swarm.
  - 7:     Initialize multi dimensional array bestParticles within size of resource type size\_resource times best n values
  - 8:         Do Until i size\_resource
  - 9:             Create a thread i in size\_resource
  - 10:             Run Swarm\_PSO with resource(i) on thread i concurrently
  - 11:             Return bestParticles
  - 12:         End Do
  - 13:     Call Greedy\_Algorithm with bestParticles
-

## CHAPTER 4

### RESULTS

This chapter presents the application of the current proposed PSO which are detailed in the methodology chapter and proposed PSO approach to solve the proposed mathematical modelling for the cloud services resource allocation problem for VoD applications. Section 4.1 analyzes the benchmark data set for the YouTube case study and characterizes the requested video and user. To validate the proposed approach, two approaches are applied. In the first approach, at a specific time interval, the achieved best value, the mean of the values had and the total time to execute are compared. for the second approach some methods used to show that how the proposed method solution approaches to the real values of the problem (achieved by using linear programming) and how well the values fits actual data described in Section 4.2. For the cloud services, Amazon Web Services solutions are used and the detail charging information given in Appendix A. The results of the solutions of the proposed methods and the accuracy metrics in detail explained in Section 4.3.1 and Section 4.3.2.

#### 4.1 Data and Implementation

Since streaming video comprises more than half of the network (internet) traffic all over the world, benchmark data for the VoD applications are studied extensively in the literature. Our benchmark data is also based on the YouTube which is well known distribution channel and discussed widely [Burgess and Green, 2013]. In this study we use the progressive download of the videos as in the case of YouTube (chunks are handled.) In the literature, there are different approaches for the shape of video streaming distribution popularity of the YouTube Video server [Summers et al., 2012]. In the study of Summers et al., there is no found correlation between the popularity and the duration of the video. Benchmark load includes video characteristics (popularity, duration and bit rates) and how much of it is downloaded. Characteristics of video is designed as video format, video length, file size and encoding bit rate [Finamore et al., 2011]. For the video popularity, in the study of Cheng et al., they found that Wei-bull and Gamma distribution both fit more than Zipf distribution [Cheng et al., 2007]. During this research, for the video duration, aggregation of normal distributions statistics are used as described in [Cheng et al., 2007]. They take the video bit rate as constant value, 419 kbps. However, in our study, we dynamically assign video bit rate and frame rate according to the user requests. Fifteen different video bit rates and frame rates of each video format; standard definition (SD) and high definition (HD) are used [Forret, 2017]. Some of them are detailed in the following Table 4.1.

The other video formats and the relation between video size and time are given in Forret's file

Table 4.1: Samples for Video Size Evaluation

Video Type	Video Format	Video Bit Rates
1920 × 1080 with 24 fps	Hulu HD	173 KB/s
	Netflix 4K	468 KB/s
	Youtube HD	960 KB/s
720 × 486 with 29.97 fps	BluRay H.264	1,42 MB/s
	Digital Cinema JPEG2000	1,85 MB/s
	Apple Prores422	5,56 MB/s

Table 4.2: Aggregated Normal Distribution Parameters

Parameter (second)	Peak 1	Peak 2	Peak 3	Rest
mean	16	208	583	295
variance	62	58	16	172
r	48,6 %	26,2 %	2,7 %	22,5 %

size calculator [Forret, 2017].

Video durations are generated by using aggregated normal distribution with the defined means and variances [Cheng et al., 2007]. We assumed that, HD video with 1080 lines uses 1920 pixels for each line and SD video with 720 lines uses 486 pixels for each line. To compute requested length of video segment in terms of time and size, we used the parameters of uniform distribution stated in Table 4.2 with the different video bit rate (vbr) of videos stated in Table 4.1.

During this study, two different network traffics - data load patterns are designed. First data load pattern has two peaks and increases in a iterative manner. In each time interval, number of videos increases by 40. Percentage of videos is 20% and 80% for SD and HD videos respectively In the second pattern, there are 6 peaks and load increases progressively. In each time interval, number of request of videos increases by 40. Percentage of SD videos is 40% and it is 60% for HD videos. Details of the data load patterns are given in the Table 4.3;

Due to the usage of public cloud services, there is not any restriction in the usage of cloud

Table 4.3: Data Load Pattern Details

Characteristics	Pattern 1 Details	Pattern 2 Details
Percentage of HD Videos	20	40
Percentage of SD Videos	80	60
Number of Peaks	2	6
Total Time	40 min	40 min
Increment of Videos Requested	40	60



services, except the constraints stated by cloud service providers. The other assumptions that are considered during this study is listed below.

**Assumptions:**

- Data transfer out from Amazon and Azure to internet up to 10 TB/month price is handled.
- Data transfer out from Azure CDN to internet up to 50 TB/month price is handled.
- Maximum number of video transcoding in each transcoder on Amazon can be 480.
- Maximum number of video streaming over in each CDN can be maximum 20,000.
- Queuing analysis is not handled for now due to the usage of public cloud services.
- Storage cost up to 1 TB is handled to increase flexibility of usage any data store in any zone.
- To generate data according to conditions defined on Table 4.2 and Table 4.3, Eclipse IDE with Java programming language is used. Algorithm details are given in the previous chapters.

For the case study, to optimize the first problem, Microsoft Azure Cloud Services are used. For the second problem, cloud systems of Amazon Web Services are used. In the multimedia applications (video streaming), we need three cloud resources that companies serve; storage resources where videos are stored, CDN to cache videos and stream videos and transcoders to transform requested video into different formats depending on the customers' needs. Costs for Azure services used are given in Table 4.4.

Cost and type of storages, CDN and transcoders used for the case study of second problem are given in the Appendix A. Amazon Simple Storage Service (S3) is an object storage to store and retrieve any amount of data from anywhere. It is ideal for videos [Services, 2017]. Currently, we do not have the ability to assign CDN to the desires that we want, so we can choose from the price classes. The cost is determined by taking the highest cost of the CDN region existing in the price class. We calculate the total cost of cloud resources used according to the user video requests. Costs of cloud resources are storage cost, transfer cost from storage, transfer cost from content delivery network, stream over https cost, decryption of encrypted data cost, transcoding cost, transfer cost from storage to transcoder and transfer cost from storage to CDN. The initial step is to load user video requests, video type and user constraints on streaming video. According to type and number of frames, calculate the videos duration and size. These steps are repeated until all video requests are assigned in each time interval. There are two different data load patterns defined during this study. In the first pattern, the network traffic is categorized as two peaks. After each peak, the traffic (number of requests) starts from 0 and increases by 40 number of videos in each time interval linearly. In the second pattern, there are 6 peaks and after each peak, the traffic starts from 0 and increases by 40 number of videos in each time interval. Also, while creating desired videos, QoS attributes of the customer for the video will be added. QoS attributes are;

- Video latency is important for the customer or not. (Latency)

Table 4.4: Cost of Azure Web Services

Zone (Region)	Transfer Cost per GB (\$)	Type of Services	Core Number	VM Cost per second (\$)
Zone 1	0.087	VM - Brazil South - A8	16	0.0408
		VM - US Center - A9	32	0.0817
		Storage - US Center	NA	0.03
		Storage - US Center_RA	NA	0.061
Zone 2	0.138	VM - Brazil South - A8	16	0.0498
		VM - Brazil South - A9	32	0.0996
		Storage-Brazil South	NA	0.0408
		Storage-Brazil South_RA	NA	0.0832
Zone 3	0.181	VM - Australia Central - A8	16	0.0465
		VM - Australia Central - A9	32	0.0930
		Storage - Australia Central	NA	0.033
		Storage - Australia Central_RA	NA	0.0671

- Video is requested over https or not. (Security)
- Requested video is encrypted or not. (Security)
- Video needs to be transcoded according to the user's requirements or not. (Latency)

There are totally 14 different regions defined for Amazon Simple Storage Service, S3. Standard Storage pricing is used. There are 8 different region that includes Amazon Elastic Transcoder application. There are 3 different location group which are defined as price class in Amazon. Each price class contains some regions that are closer each other. Users' zone are created as the union of Azure and Amazon zones. The Amazon regions are demonstrated in the Figure 4.1.

While creating the number and aspects of the incoming video requests, the algorithm of which the pseudo-code is given in Algorithm 8 is used.

## 4.2 Validation Methods of the Proposed Model

Throughout this study, to compare the effectiveness of the algorithms proposed and to show how the solution approaches to the real values; analysis and discussion on a unique data is performed. Besides it, to see the big picture and to show the fluctuation under different loads, we used some forecast accuracy metrics. Used metrics are explained in the following Section 4.2.1 detail.



Figure 4.1: Amazon Regions

Table 4.5: Pseudo-Code for the Benchmark Data Generation

---

**Algorithm 8:** Data\_Generate.

---

- 1: While ( $\delta t$ )
  - 2:   Calculate the number of standard definition and high definition of videos at time interval  $t_i$  by using statistics given in Table 4.3
  - 3:   Calculate the number of videos for each peak defined in the Table 4.2.
  - 4:   Calculate the time of the video  $duration_i$ , according to aggregated normal distribution for each type of video
  - 5:   For all requested videos  $v_i \in V(t)$
  - 6:     Compute the size  $r_i$  of videos according to the time, bit rate and frame size of a video,  $V(t_i, f_i)$  (a set of time  $t \in T$ )
  - 7:     Assign the QoS attributes (zone, frequency, latency, https, encryption and user device type (for transcoding)) to each requested video
  - 8:   End For
  - 9: End While
-

### 4.2.1 Forecast Accuracy Metrics

To decide how far the proposed algorithms approaches accuracy, three different accuracy metrics are used. The definitions, mathematical formulas and the motivation is given below;

#### 4.2.1.1 Root Mean Squared Error (RMSE)

Root mean square error is the spread from the actual measures. If the forecast measures are further away from the actual measure, then the value of RMSE becomes greater.

$$RMSE = \sqrt{\sum[(\frac{F-A}{N})^2]} \quad (4.1)$$

#### 4.2.1.2 Normalized Root Mean Squared Error (NRMSE)

$$NRMSE = \frac{RMSE}{F_{max} - F_{min}} = \frac{\sqrt{\sum[(\frac{F-A}{N})^2]}}{F_{max} - F_{min}} \quad (4.2)$$

Although it is popular accuracy metric, it takes the squared of residual then large errors gain high weight. However these errors are not undesirable because it is the natural result of this situation. So, this approach may give bad results like the RMSE. Therefore, weighted mean absolute percentage error is decided to use.

#### 4.2.1.3 Weighted Mean Absolute Percentage Error (WMAPE)

Weighted Mean Absolute Percentage Error (WMAPE) reports are particularly useful and are becoming very popular [Kolassa and Schütz, 2007]. They are easily calculated and give a concise forecast accuracy measurement that can be used to summarize performance at any detailed level. The formula WMAPE is given below;

$$WMAPE = \frac{\sum[(\frac{|F-A|}{A}) \times 100 \times A]}{\sum A} \quad (4.3)$$

Nonetheless, in the existence of positive and negative errors, interpretation of WMAPE should be done carefully. Since, by reason of the fact that the residual of our data cannot be negative, this metric seems quite appropriate.

### 4.2.2 Implementation Details

Proposed Algorithm is implemented in Java using Eclipse Java Neon framework. Firstly, First in First Out (FIFO) basic scheduling algorithm is used to compare the efficiency of the proposed mathematical modelling. FIFO algorithm is implemented in Java and Linear and mixed integer quadratic programming algorithm is implemented using IBM Ilog Cplex library in Java. For the adaptation of variants of PSO, jswarm-pso\_2\_08.jar library is used.

### 4.3 Experimental Results

In the study, two models are proposed to minimize the cost for cloud solutions of VoD applications. In the first model, virtual machine and storage services (IaaS) are used. This model has a limited source and since the mathematical model is linear, integer programming is used to solve the problem. To show the necessity of this kind of approach, the proposed model are compared with one of the simplest methods; FIFO resource allocation method. For the case study of this approach, we use Microsoft Azure services.

In the second approach, CDN which is inevitable when the subject is VoD applications; transcoders which is necessary due to the heterogeneity of the devices; and storages are used. Since the mathematical model proposed for the second approach is non-linear, then branch and cut method is used to solve mixed integer quadratic problem. Dash claimed that branch and cut approach for the type of 0-1 integer problems have exponential running time in worst case [Dash, 2005]. When the demand for the video rises, then execution time becomes a problematic issue for the problem. Therefore, evolutionary algorithms are recommended to use. These methods are detailed in the Chapter 3. All algorithms are executed in the computer which has one processor installed. The processor is Intel64 Family 6 Model 78, 2492 Mhz. Total physical memory is 8 GB and maximum virtual memory is 9,38 GB. In the following sub-sections 4.3.1, 4.3.2 and 4.3.3, we show the efficiency and effectiveness of the proposed algorithms. For the second approach, there are two case studies on two different network traffic patterns are applied.

#### 4.3.1 Case Study for the Model $CRAP\_VOD_{VMS}$ under Microsoft Azure Cloud Services

In this approach, VMs and storages are used as resources and, videos are assigned to these services within a minimum cost. The problem presented in Section 3.2 is solved by using integer programming. Firstly, incoming requests are assigned to limited sources as a FIFO strategy, i.e. each video is streamed by using the first empty and suitable VM and taking out of the first storage in the list without regarding the price of used services and taking care of customer satisfaction. This is achieved without regarding the price of used services and taking care of customer satisfaction. Thereafter, cost is tried to be minimized by solving the presented mathematical formula and unequivocally, the charge is decreased and customer satisfaction is provided. In this approach, some assumptions and constraints based on this study are given in the following list.

- HD and SD videos are assumed to be type of JPEG2000. Then, the size of the video is evaluated according to these formats.
- 20% of videos are assumed to be HD and 80% of videos are assumed to be SD.
- We assume that HD video with 1080 lines uses 1920 pixels for each line and SD video with 720 lines uses 486 pixels for each line.
- In the architecture of the cloud services, limited number of storage and VMs are used and the type of these resources with the cost is given in Table 4.4.
- PSO parameters and benchmark data load is given in the Table 4.6 and Table 4.3 respectively.

Table 4.6: PSO Parameters

Parameter Name	Parameter Value
Number Of Particles	25
Inertia	0.9
Global Increment	2
Particle Increment	2
Neighbourhood Increment	0.5

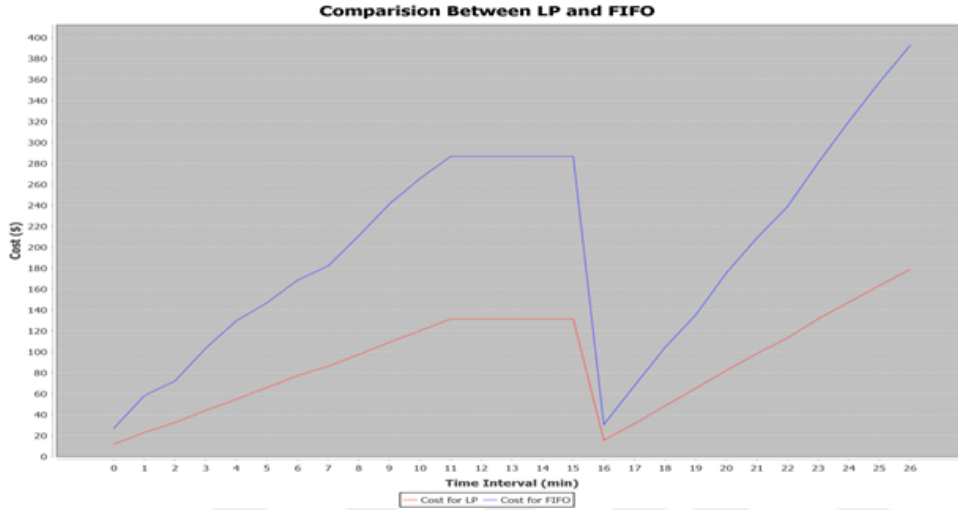


Figure 4.2: Comparison of bandwidth cost between LP and FIFO based algorithm when varying user video requests

In the case study, Microsoft Azure solutions (Azure VM and Blob Storage) are used. The requests from users are handled within each minute and Azure VM has constraints on the core numbers. It is charged per minute. A8 and A9 VMs are preferred due to the suitability to video coding. In every execution, transfer cost and VM cost is evaluated. In general, x-axis of the graphs show us the time and Y-axis represents the cost of the used services for all requests within that time interval. In each time interval, number of requested videos increase progressively. Videos are streamed from storage which is in the same zone of VM assigned. This not only decreases latency problem, also decreases the cost due to the fact that in the same zone transfer cost between cloud resources is zero. Also, this assumption provide the problem to become linear. So, it is solved easily by Integer programming. While streaming video, transfer cost from VM to the Internet constitutes huge part of the total cost. In the Figure 4.2, the comparison between FIFO and LP algorithm is given. We obviously get that LP results much better than FIFO algorithm in the total cost. As can be seen clearly from the figure, even with very little data, quite sharp differences emerge. Increasing in the size of video request decreases the spent on cloud resources for VoD application provider. Thus, proposing such an algorithm not only improves the performance of the provider but also reduces the cost on cloud solutions. The main reason for LP to perform better is that it considers the region of resources, cost of the resources and size of the videos while assigning incoming requests.

To increase the user satisfaction in terms of latency, the distance between user and cloud resource is minimized. Location of users and zone of cloud resources are tried to be kept in

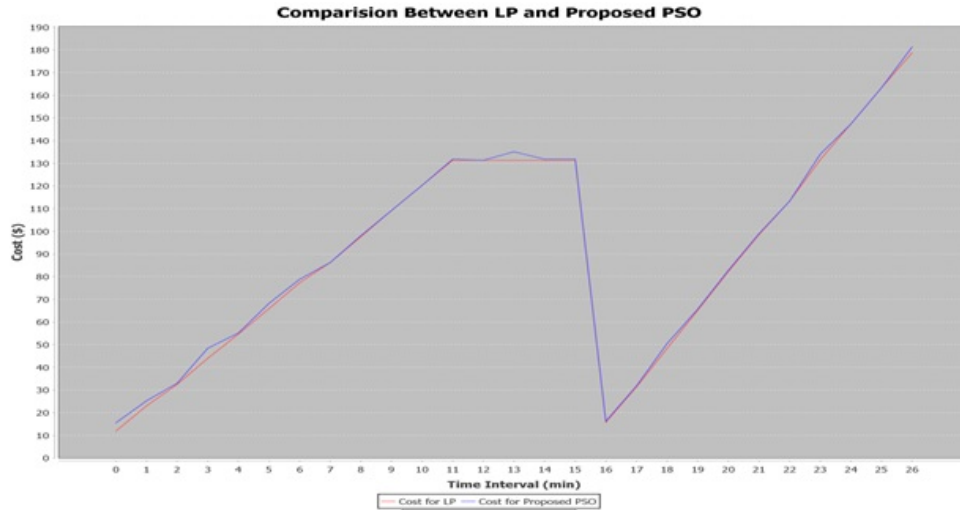


Figure 4.3: Comparison of the results of 1,000 iterative Proposed PSO optimization technique with LP results in varying user video requests

same. This is added as a constraint to the LP model. Although adding this constraint increases the total cost, results are still pretty much better than the results of FIFO algorithm.

As stated in the introduction chapter, we use PSO evolutionary algorithm to solve the problem within time efficiency. To show the efficiency of the proposed PSO algorithm, we use the weighted mean absolute percentage error (WMAPE) metric. This metric is easily calculated and gives a concise forecast accuracy measurement that can be used to summarize the performance at any detailed level. The optimum solution is tried to be achieved by using the proposed PSO under binary constrained with neighbourhood topology. This proposed algorithm is compared with the study of Khanesar et al. who proposes the Novel Binary PSO (NBPSO) [Khanesar et al., 2007]. The Figure 4.3 shows us the differences between NBPSO and LP. Under 100 iterations, WMAPE becomes 0.035 which is quite high to not choose the novel binary PSO. When the number of iterations increases to 1,000, WMAPE decreases to the 0.024, which is better, as expected.

The Figure 4.4 represents the comparison of the results of Proposed PSO with LP. The proposed PSO has better results than NBPSO. Under 100 iterations, WMAPE becomes the 0.0204 and under 1,000 iteration, it decreases to 0.0034 as expected. And, this value shows that the proposed algorithm is more satisfactory than the benchmark algorithms to use the algorithm to reach optimum solution of the presented problem.

The summary of the comparison of the algorithms are given in the below Table 4.7. WMAPE metric values falls from 0.024 to the 0.0034 value in Table 4.7 clearly shows that proposed algorithm yields considerably better results than Novel Binary PSO.

In addition; Figure 4.5 shows that proposed PSO algorithm also yields better results than the Novel Binary PSO. Novel PSO algorithm executes in 574.05 ms and Proposed PSO algorithm runs in 517.74 ms which has an advantage from time of 10%.

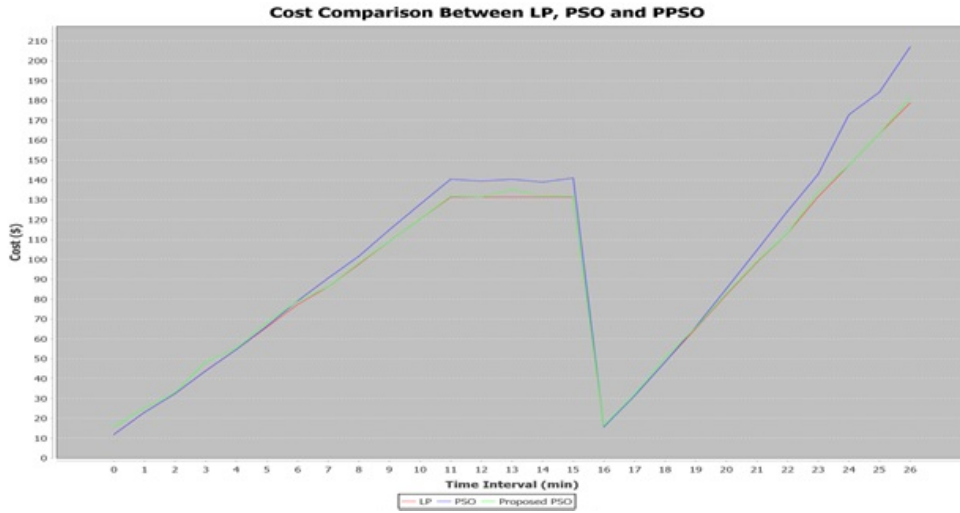


Figure 4.4: Comparison of the results of 1,000 iterative Proposed BPSO and Novel BPSO optimization techniques with LP results in varying user video requests

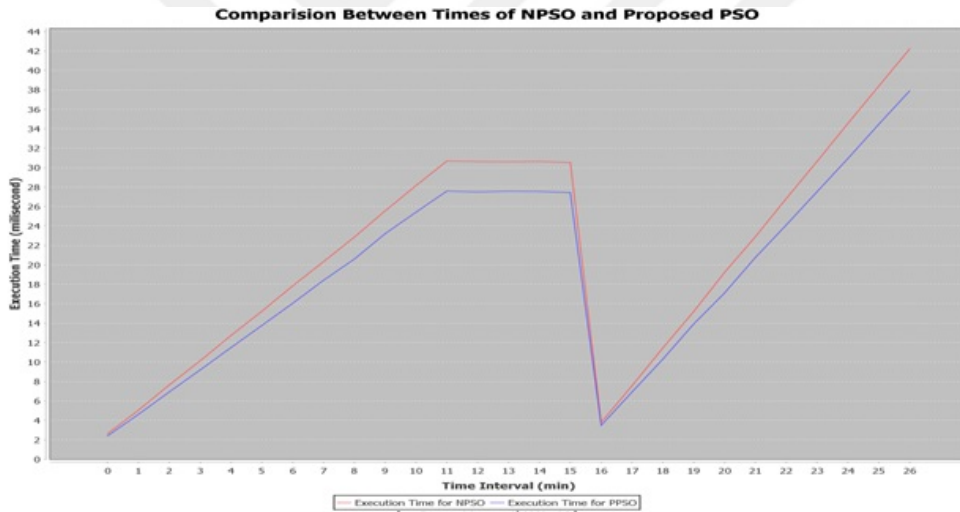


Figure 4.5: Comparison of the execution time of 1,000 iterative Proposed BPSO and Novel BPSO optimization techniques

Table 4.7: Comparison of the Algorithms

Optimization Technique	Number of Iterations	WMAPE
NBPSO	100	0.035
NBPSO	1,000	0.024
Proposed BPSO	100	0.0204
Proposed BPSO	1,000	0.0034



Table 4.8: PSO Parameters

Parameter Name	Parameter Value
Number Of Particles	25
Inertia	0.9
Global Increment	2
Particle Increment	2
Neighbourhood Increment	-2.9

#### 4.3.2 Case Study for the Model $CRAP\_VOD_{CDNTS}$ under Amazon Web Services

In this experiment, cost and type of cloud services offered by AWS are used. Cost and type of services are given in Section 4.1. The charging strategy for on-demand Amazon Web Services are detailed in Appendix A.

In the previous solution of the model, to minimize latency, keeping user and all cloud services in the same region strategy is developed. But in case of an unexpected situation like service interruption, it is necessary to develop a different strategy in order to continue the service. For this reason, CDN services are used in addition to using latency data between the location of user and services. Cloud latency can be measured by using some techniques to select the most suitable services for the video streaming to satisfy the QoS of users. The latency data between Amazon web services are given in the Figure 4.6. The data is evaluated by using the averages of the previous 24 hours of data collected. It is categorized as three groups; high latency ( $>180$  ms), medium latency (between 100ms and 180 ms) and low latency ( $<100$  ms). In the study, the data transfer between services which has low or medium latency is allowed.

Security options which are streamed over https and encryption of the content are added as an additional QoS attributes to the problem domain. Analysis of deciding TTL of video for caching and keeping different versions of video according to the popularity of video is totally another main subject and it is handled in Section 3.1. Amazon does not give customer a permission to select the region. However, it offers price classes for CloudFront CDN and customer may make a selection in a price class level. Amazon CloudFront minimizes end user latency by delivering the content from edge locations. This means that you may pay more to deliver your content with low latency to end users. By excluding Amazon CloudFront more expensive edge locations (price classes), delivery prices may be reduced but latency may increase. Different versions of video by using video transcoded services are provided. Amazon Elastic Transcoder converts your media files in S3 to various formats depending on the device of viewers. Then, it is stored back in S3. It is charged per second so video time is the crucial for the evaluation of the transcoder cost.

## AWS Inter-Region Latency

The data in the table below represents the averages of the previous 24 hours of data collected. Hover over a cell for the weekly average.

Destination Region	ap-northeast-1	ap-northeast-2	ap-south-1	ap-southeast-1	ap-southeast-2	ca-central-1	eu-central-1	eu-west-1	eu-west-2	eu-west-3	sa-east-1	us-east-1	us-east-2	us-west-1	us-west-2
Source Region															
ap-northeast-1	44.29	135.54	87.91	125.99	157.98	247.05	251.79	274.53	279.09	284.70	181.59	173.76	121.64	121.19	
ap-northeast-2	34.45	167.75	123.88	141.87	178.28	287.06	270.89	271.65	285.85	305.99	200.78	199.20	142.36	145.56	
ap-south-1	153.02	173.70	68.01	241.86	210.51	124.70	129.93	117.47	115.48	325.75	188.81	203.47	269.80	235.22	
ap-southeast-1	72.45	111.80	71.36	173.19	213.85	177.98	188.26	179.51	167.60	333.74	257.12	239.43	192.05	166.99	
ap-southeast-2	134.37	151.10	185.73	252.27	241.50	321.02	308.79	315.44	303.35	347.01	228.53	211.91	164.04	171.87	
ca-central-1	145.87	172.35	203.89	210.06	215.10	104.95	82.36	92.26	96.60	128.81	16.89	28.11	81.85	68.39	
eu-central-1	240.83	272.14	120.04	172.28	294.34	105.96	25.50	17.10	12.86	210.22	90.39	101.72	150.68	166.06	
eu-west-1	223.95	254.75	134.12	188.27	287.94	83.27	25.24	14.46	20.46	189.72	78.38	90.28	143.19	143.02	
eu-west-2	233.56	261.32	119.38	173.64	283.79	94.18	14.62	10.91	10.91	199.91	78.97	88.98	140.96	157.74	
eu-west-3	256.30	281.66	109.97	171.72	288.54	95.38	13.91	12.55	12.55	205.00	81.80	92.69	146.69	160.32	
sa-east-1	289.59	337.51	389.81	351.22	348.68	176.86	209.28	239.98	246.03	137.36	143.47	183.63	191.99	191.99	
us-east-1	161.53	185.12	190.25	250.12	210.92	18.50	80.90	79.36	83.60	125.63	13.51	67.99	84.91	84.91	
us-east-2	161.79	191.36	202.01	242.98	202.34	29.99	101.55	93.71	95.99	147.53	17.23	55.53	73.03	73.03	
us-west-1	113.59	146.44	264.66	189.64	149.52	82.14	149.31	141.27	144.64	194.07	66.88	52.52	23.07	23.07	
us-west-2	107.43	139.63	236.44	170.89	175.81	72.64	172.85	165.25	172.11	198.97	91.15	74.82	26.20	26.20	

Latency: ● < 100ms ● 100-180ms ● > 180ms  
All times are in milliseconds.

Figure 4.6: Amazon Web Service Latency Information

**Comparison Between First In First Out Resource Allocation (FIFO) and Mixed Integer Quadratic Programming (MIQP)**

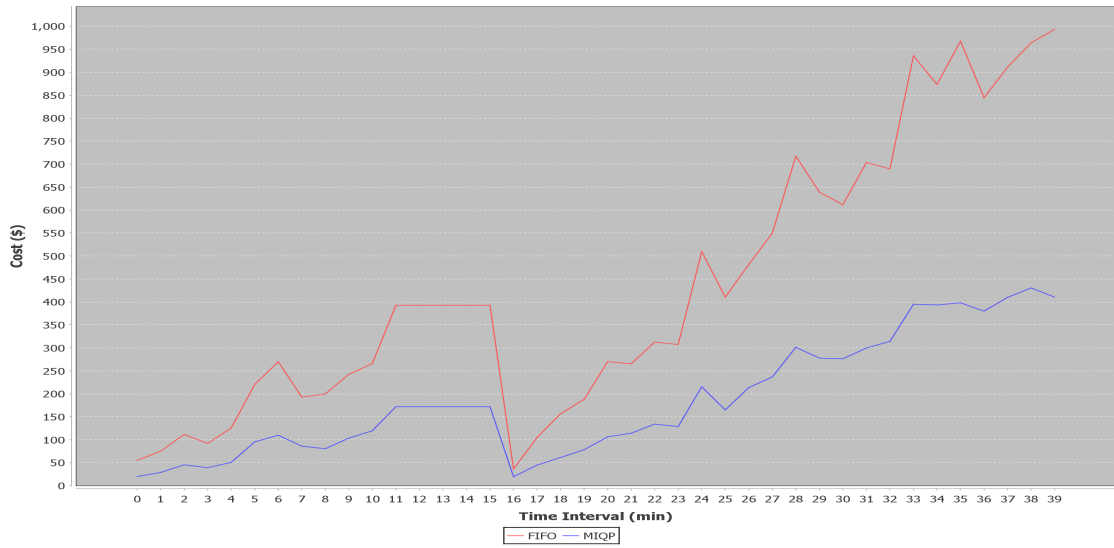


Figure 4.7: Comparison between the results of FIFO resource allocation technique with MIQP with varying user video requests

As a performance metric, we consider both the cost of all cloud resources used for video request in each minute and the time to run algorithms. We evaluated minimum cost of different modified versions of PSO and time to run these algorithms while satisfying QoS attributes of users. In every execution (on every minute), transfer cost between cloud resources and internet, QoS cost and transcoder cost are evaluated. X-axis parameters of the graphs show the time interval (one minute in our case), Y-axis represents the cost and time according to the graph content in general. As clarified in Chapter 3, algorithms are developed by considering the missing parts of the previous version. We start with the exact solution provided by the MIQP and compare the proposed algorithms with the exact solution to show how the proposed optimization technique approaches to the real values.

#### 4.3.2.1 Experimental Results of Mixed Integer Quadratic Programming

The results are very crucial for us to show that the performance of our proposed heuristic algorithm and most importantly how much such a modelling is required. If there is no algorithm for the videos allocated to the proposed cloud resources, and randomly FIFO is used, then the cost will be very high. Thus, the results point out absolutely essentiality of the proposed algorithm to reduce the cost spends on cloud services. If the resources are not allocated well then cloud users come up against higher cost and lower customer satisfaction for their applications. The Figure 4.7 presents the proposed formula yields superior results. This result show that there is significant cost saving in cloud resources while increasing the user satisfaction. When the number of video requests are 1,440 then the difference between the cost are \$578 which is quite high to use such an approach. The effectiveness of the results totally depends on considering the cost of the services in addition to the latency information and security. Although there is an usual belief that the use of the cloud services which has the lowest charging cut down the cost on them, this approach is ridiculous for the satisfaction of user. Because, when the latency values between regions are investigated, then the latency value between south 1 and east 1 is 389.81 milliseconds in Figure 4.6 which makes the user dissatisfied and so refuses

**Comparison Between the Mixed Integer Quadratic Programming (MIQP), Binary PSO (BPSO) and Novel Binary PSO (NBPSO)**

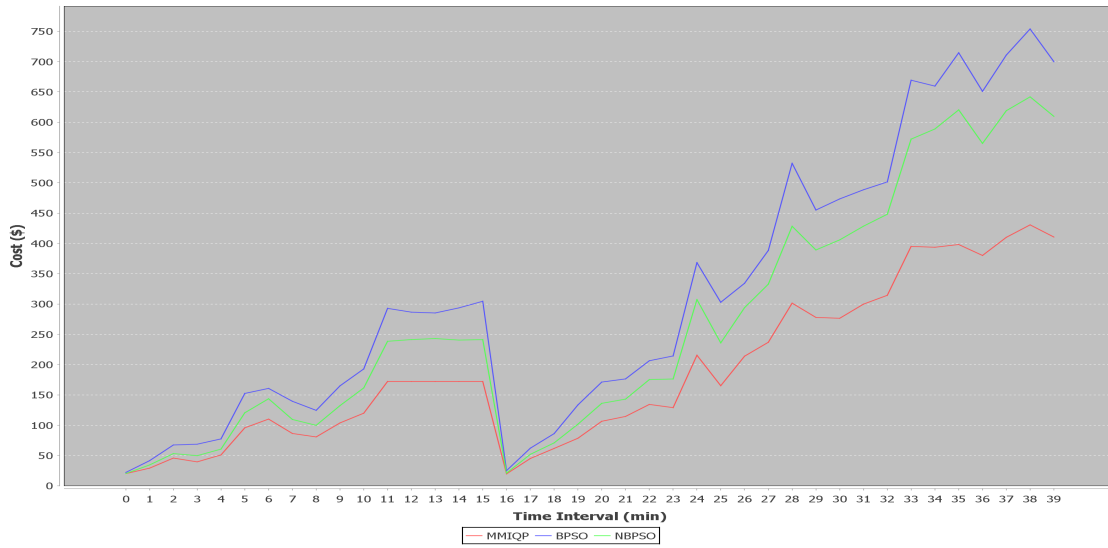


Figure 4.8: Comparison between the results of 1,000 iterative Binary BPSO, Novel Binary PSO and MIQP in varying user video requests

to use the service. While minimizing the cost, satisfaction of user should not be discarded. When we examine the results for a given time interval, we notice that the service provider’s expenditure on cloud resources has fallen by almost half compared to the model created with the FIFO approach. If we approach more systematically to the services allocation as in the MIQP, cost is decreased from \$998.04 to \$420.82 as can be seen in Table 4.9. Due to the time inefficiency to solve problem, we propose an evolutionary algorithm which is modification of PSO. In the Section 4.3.2.2 we will examine the results of this optimization technique in detail.

#### 4.3.2.2 Experimental Results of Modified Versions of Binary PSO

When the first binary PSO proposed by the creators of continuous PSO, the results were disappointing. Binary PSO algorithm was not effective as in continues PSO and weighted mean absolute percentage error and normalized mean squared error are 0.0669 and 0.507 respectively as seen in Table 4.10 which are fairly high. Then, the study of Khanesar et al. which is respected and cited mostly in the literature proposes a Novel Binary PSO. Although the results show that applying Novel Binary PSO is better than BPSO, it still suffers and needs to be modified for the problem described. Furthermore, due to the algorithm complexity, it is worse than BPSO when the execution time is regarded. Execution time rises from 24,255 milliseconds to the 50,131 milliseconds; which can be seen in Table 4.9. Despite this great difference in time, the results were not as effective as expected. When the ring neighbourhood topology is added to the algorithm, although there is no difference in terms of execution time, there is a nice increase in approaching to exact values. By adding ring topology, each particle not only affected globally, it is also affected by the previous and next particles.

In the Figure 4.8, the blue line represents the cost of BPSO and green line represents the cost of Novel BPSO. The aim of this optimization technique is to reach the values of the red line which is the indicator of exact solutions. Novel BPSO is better than Binary PSO as can be seen evidently in Figure 4.8. In the Table 4.10, comparison of the metrics of all used optimization

Table 4.9: Comparison of Algorithms Proposed in a Time Interval

<b>Optimization Technique</b>	<b>Tech-</b>	<b>Number of Genera-</b>	<b>Running</b>	<b>Total Cost (\$)</b>
		<b>tions</b>	<b>Time (ms)</b>	
FIFO		NA	-	998.04
MIQP		NA	-	420.82
Binary PSO		50	24	756.74
		100	695	751.64
		200	1,360	828.67
		500	3,393	730.81
		1,000	6,695	699.84
Novel Binary PSO		50	50	671.10
		100	1,554	685.06
		200	2,951	749.19
		500	7,626	653.97
		1,000	14,700	609.60
Neighbourhood Novel Binary PSO		50	48	625.83
		100	1,507	613.06
		200	2,842	675.01
		500	7,362	552.60
		1,000	14,300	531.87
Multi-Swarm Neighbourhood Novel Binary PSO		50	44	486.89
		100	1,362	477.86
		200	2,615	519.22
		500	6,666	447.50
		1,000	12,800	422.41
Parallel Multi-Swarm Neighbourhood Novel Binary PSO		50	33	492.62
		100	1,016	479.06
		200	2,004	518.62
		500	4,995	444.73
		1,000	10,000	423.98

Table 4.10: Comparison of Different Variants of Binary PSO

Optimization Technique	Number of Generations	WMAPE	RMSE	NRMSE
Binary PSO	50	0.669	148.05	0.507
	100	0.669	151.43	0.522
	200	0.662	149.05	0.517
	500	0.689	158.93	0.527
	1,000	0.671	155.49	0.512
Novel BPSO	50	0.490	113.85	0.437
	100	0.482	113.84	0.442
	200	0.458	108.14	0.428
	500	0.440	107.69	0.419
	1,000	0.416	101.84	0.395
Neighbourhood Novel BPSO	50	0.367	88.49	0.369
	100	0.327	80.50	0.349
	200	0.281	71.12	0.320
	500	0.247	65.08	0.292
	1,000	0.226	61.07	0.274
Multi-Swarm Neighbourhood Novel BPSO	50	0.092	25.77	0.135
	100	0.059	17.04	0.092
	200	0.035	10.98	0.061
	500	0.020	6.55	0.036
	1,000	0.013	4.49	0.024
Parallel Multi-Swarm Neighbourhood Novel BPSO	50	0.092	26.11	0.137
	100	0.058	16.78	0.091
	200	0.036	11.23	0.062
	500	0.020	6.29	0.035
	1,000	0.014	4.80	0.026

techniques in this study under different number of generations is given. The table presents that the results of the algorithms become more impressive when the number of generation increases as expected. WMAPE metric is more applicable for the problem we have constructed than NRMSE as explained in the Section 4.2. All metrics approve the improvement of our proposed algorithm.

Figure 4.9 shows that the differences between the results of Novel Binary PSO and Neighbourhood topology Novel Binary PSO. Both PSO models are run under 1,000 number of generations. The results produced clearly reveals that the proposed new topology PSO achieves better results than Novel BPSO. On the contrary, the algorithm is not enhanced within the execution time as effective as the correctness. The time to execute algorithm is dropped from 98,417 milliseconds to 95,826 milliseconds which can be seen in Table 4.9.

**Comparison Between the Novel Binary PSO(NBPSO), Neighborhood Novel Binary PSO (NNBPSO) and Multi-Swarm Neighborhood Novel Binary PSO (MNNBPSO)**

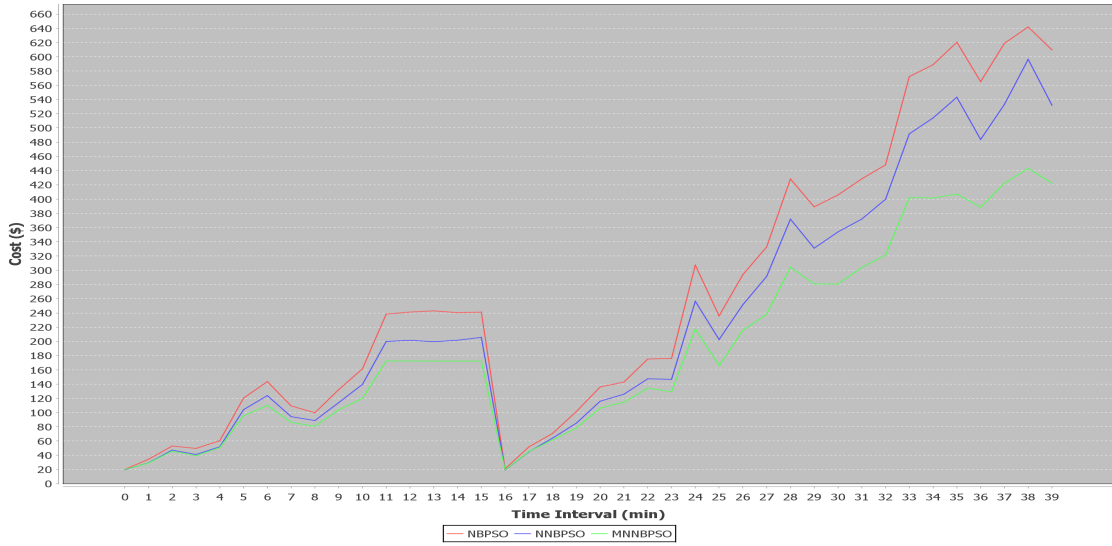


Figure 4.9: Comparison between the results of 1,000 iterative Novel BPSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

#### 4.3.2.3 Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm

Until now, even if some improvements in the algorithm are reported, the results produced are not dramatic and impressive. Heretofore, we tried to solve the problem with a single swarm approach as it is the classic PSO. However, this approach not only enlarged the size of the particle but also made it difficult to manage the particle since each particle represents the combined state of more than one cloud services which are CDN, transcoder and storage in this problem. Trying to optimize different services has come up with the idea of creating a swarm for each service according to the regions. Although this approach seems not to give a better result because of the strong communication between services at first, we obtained good results because the cost of each service decreases or increases in a similar way according to the region. For each resource, cost of each type in the region is proportional to each other. For instance, if one type of resource has the lowest cost in the region, then other resources has also the lowest cost in that region. Since the nature of the swarms is very similar in terms of cost diversity through region, multi-swarm technique is much more suitable for our situation. Three sub swarms, each swarm represents each cloud resource type, are created. And each is run independently of each other. Because of the cost proportionality in the regions, the greedy algorithm, which is a simpler algorithm, is studied to optimize the cost between the swarms. The Figure 4.9 apparently present that the proposed algorithm produces the desired result. In Table 4.10, WMAPE metric value is decreased from 0.23 to the 0.013 which shows the impressive and sharp enhancement in the optimization technique. Along with, there is a decrease in execution time drastically. Figure 4.10 manifests that Multi-Swarm PSO run faster than the other algorithms implemented up to now.

As shown in Table 4.9, the cost falls from 532 to 431 dollars by the help of these algorithms and execution time falls from 95,826 milliseconds to 86,635 milliseconds.

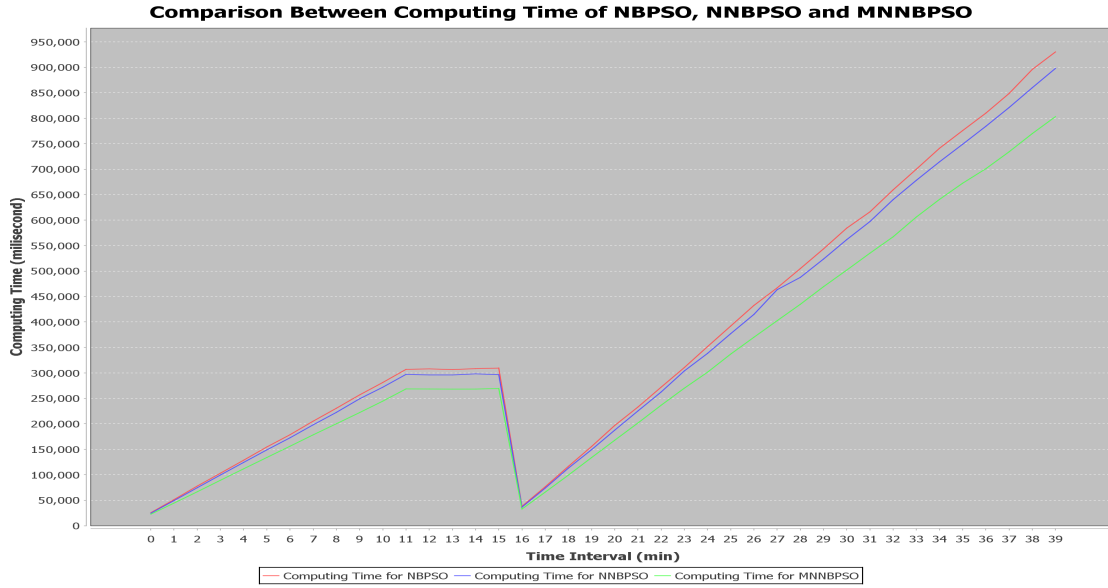


Figure 4.10: Comparison between the computing time of 1,000 iterative Novel Binary PSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

#### 4.3.2.4 Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm using Parallel Programming

Despite the fact that multi-swarm algorithm yields better solution in terms of both accuracy and execution time, we propose parallel multi-swarm algorithm to increase the efficiency in execution time of the algorithm. Besides, since there is huge number of solution types offered by cloud service providers, size of space increases unavoidably. When the dimension of the search space increases, many optimization algorithms end up with worse results and it takes long time to reach solution. To tackle this problem and due to the compatibility of the nature of the proposed algorithm with parallel programming, we implement multi PSO with parallel programming. We define three threads and run each PSO swarm of multi-swarm algorithm simultaneously. It reduces the execution time of the algorithm noticeably. After parallelizing multi-swarm PSO, there is a significant decline in execution time of the algorithm. Figure 4.11 shows that parallel algorithm is run less time than the Multi-Swarm PSO. And, as expected, this approach does not show an improvement in the accuracy of the algorithm. As stated in Table 4.9, execution time is decreased by almost 25%. This is very substantial and satisfactory value in terms of the increasing the performance.

To sum up, Table 4.9 and Table 4.10 show the results of the algorithms under different number of generations. We easily get that each proposed algorithm improves the Novel Binary PSO and optimizes our problem pleasurable. Table 4.9 shows the total cost of application evaluated by algorithms and the time to run these algorithms under the condition of different numbers of iterations in a time interval in which the data intensity is high. On the other hand, Table 4.10 contains different metrics that show how close the total cost is to the lowest payoff, taking the entire time interval into consideration. If we disregard the algorithms developed while reaching the proposed algorithm, we can figure out that the algorithm proposed in this study is better both in terms of accuracy and time than Novel PSO in Figure 4.12 and Figure 4.13.



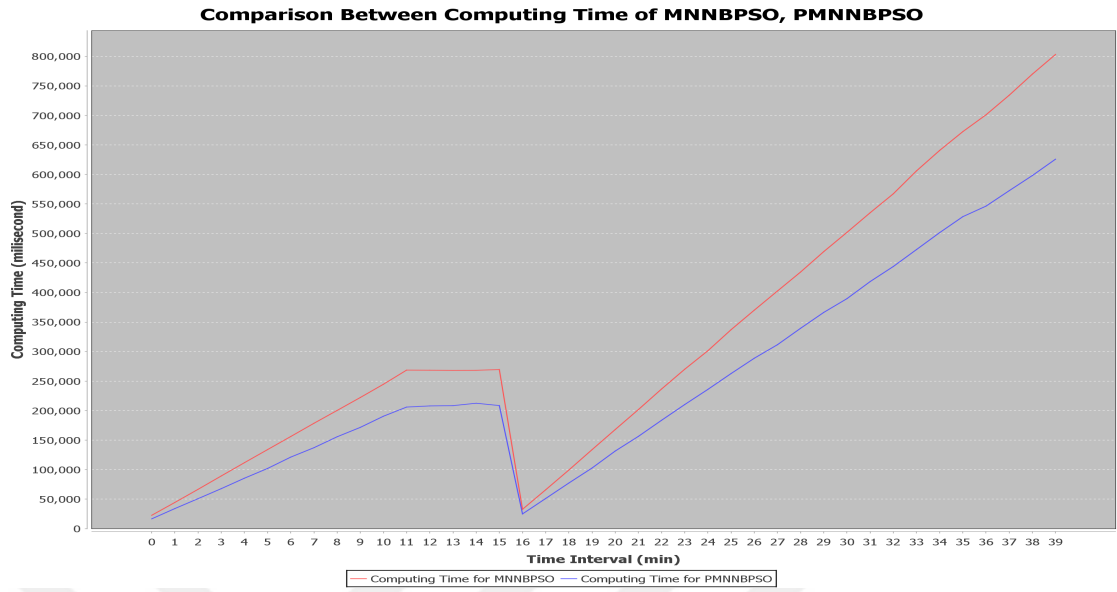


Figure 4.11: Comparison between the computing time of Multi-Swarm Neighbourhood Novel Binary PSO with Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

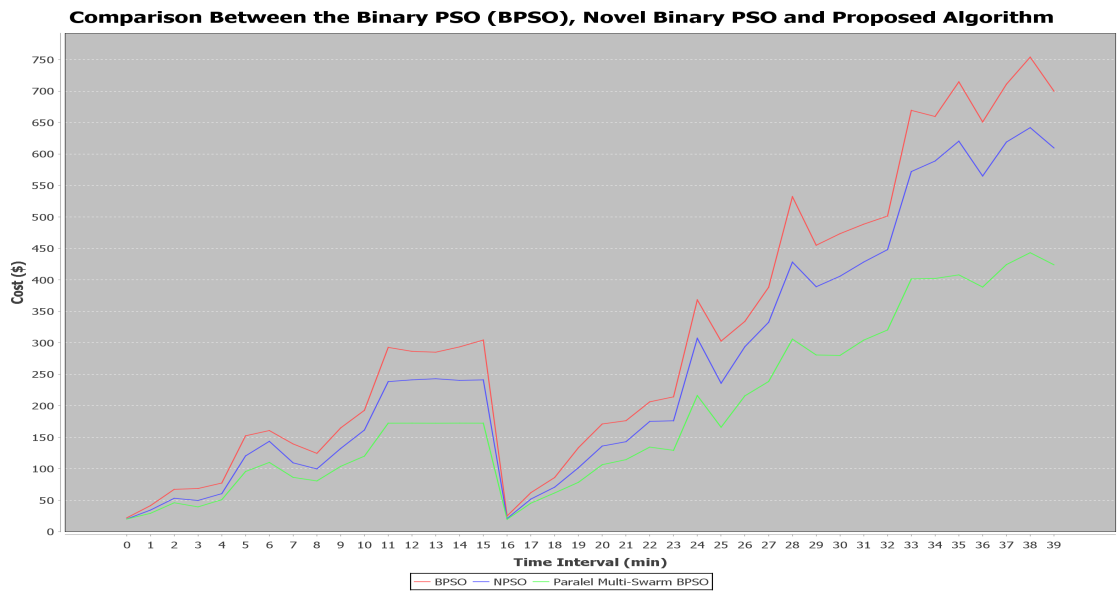


Figure 4.12: Comparison of the results of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests

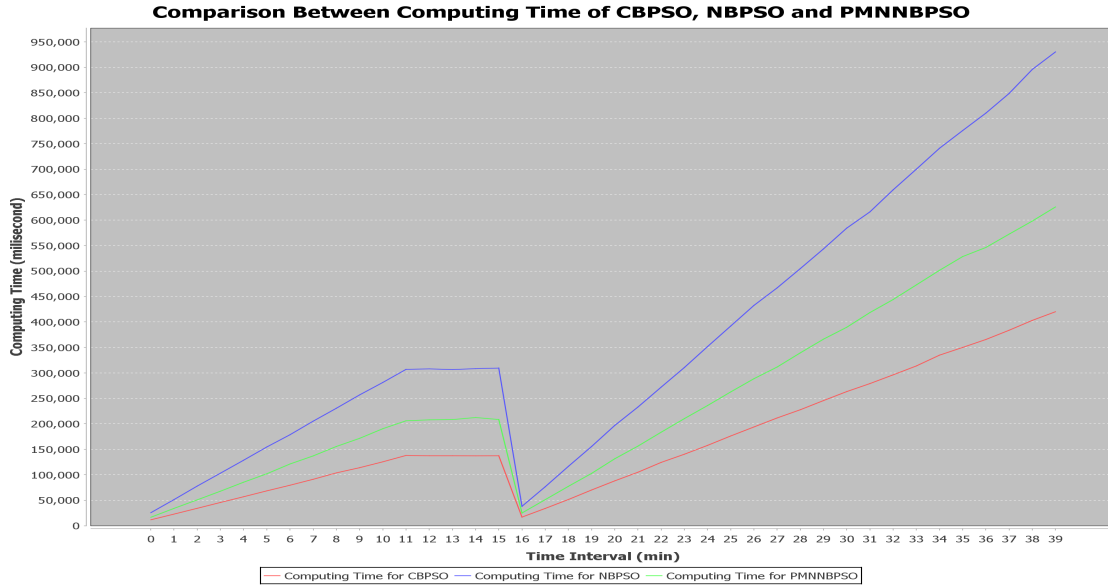


Figure 4.13: Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests

In conclusion, cost of the VoD service providers based on cloud services dropped from \$993.50 to \$423.98 which is almost 57% cost saving.

The total picture of algorithms under different number of iterations is given in the following Figure 4.14 and Figure 4.15. Figure 4.14 shows that as the number of iterations increases, the performance of the algorithm increases in the same way. On the contrary, the execution time of the algorithm increases rapidly. When the optimization technique is run under 50 iterations, then the cost falls from \$998.04 to \$492.62 which provides 51% cost reduction. Under 1,000 iterations, there is a falling from \$993.50 to \$ 423.98 which is 57% in cost saving. On the contrary, execution time is about 19 times higher when the number of iterations increases from 50 to 1,000.

Even though during this study, the execution time is tried to be minimized, sometimes for the organizations, the minimized time is not enough satisfactory and they prefer to compromise on the accuracy of the algorithm, i.e. a little deviation from the real values can be acceptable to reduce the time. So, the proposed algorithm is run in a limited time instead of iteration and the results of how algorithm approaches the real values are given in the Table 4.11. The case study result of the MIQP algorithm is \$482.82 and FIFO algorithm is \$1069.69. The values in Table 4.11 presents that the value of the proposed algorithm get closer to the value of MIQP by the increase in time limit as expected.

### 4.3.3 Case Study for the Model $CRAP\_VOD_{CDNTS}$ under Microsoft Azure Cloud Services

In this experiment, cost and type of cloud services offered by Microsoft Azure are used. Cost and pricing strategy of the Microsoft cloud services defined in Section 4.1 is given in Appendix A.

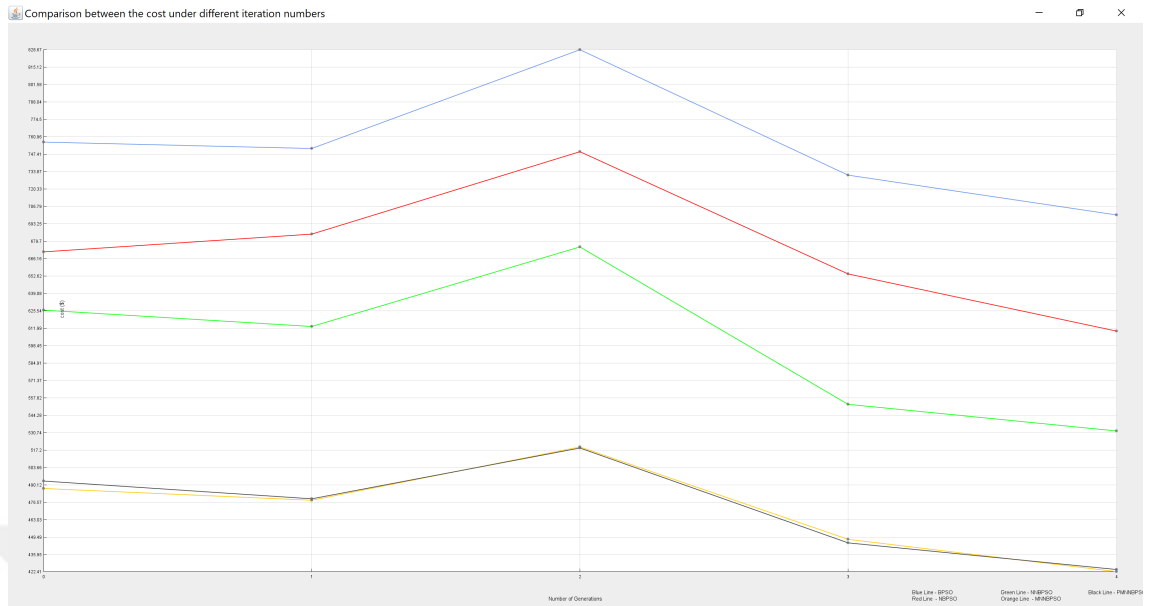


Figure 4.14: Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests

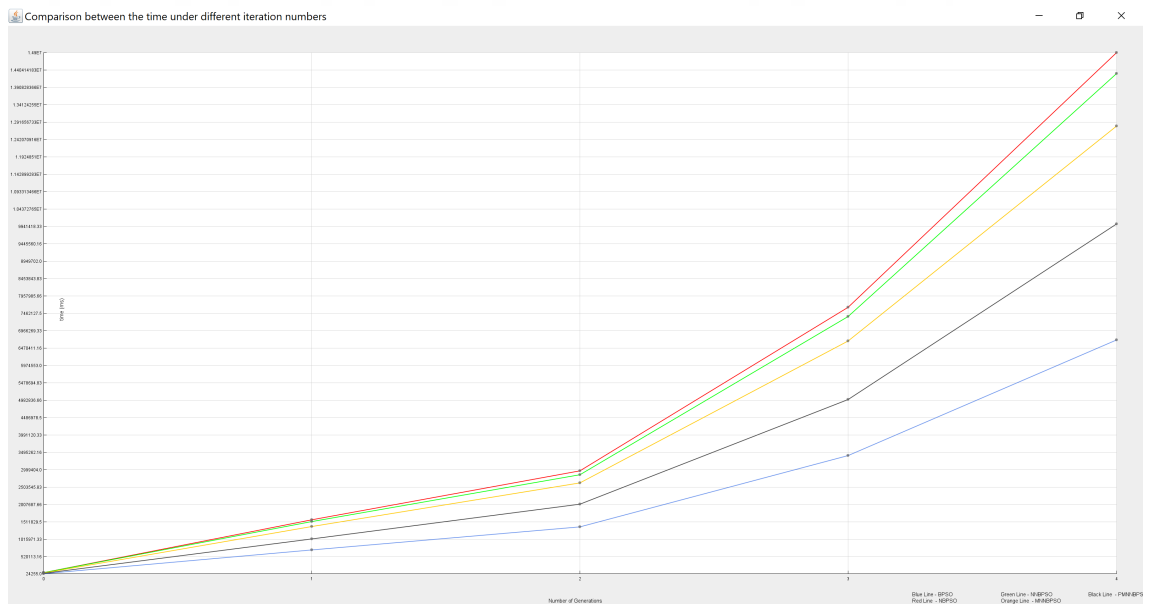


Figure 4.15: Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

Table 4.11: Comparison of the Results of PSO Run Under Different Time Limits

Time Limits (second)	Total Cost (\$)	WMAPE	RMSE	NRMSE
5	589.29	0.119	36.88	0.187
15	542.87	0.064	21.49	0.114
30	532.06	0.042	14.33	0.078
45	516.33	0.031	10.63	0.058
60	515.03	0.029	9.55	0.052

Table 4.12: PSO Parameters

Parameter Name	Parameter Value
Number Of Particles	25
Inertia	0.9
Global Increment	2
Particle Increment	2
Neighbourhood Increment	-2.9

Load Pattern 2, the network traffic defined in Chapter 4.3, is used to show the efficiency of the proposed algorithm. QoS attributes like security options defined in the previous study are handled during this case study as well. For the content delivery networks, Azure does not allow choosing the region at the lowest level like Amazon. However; it offers 5 zone levels in different pricing strategies. Different versions of video by using media services are provided. Azure media services convert media files to various formats depending on the device of viewers. It is charged per second so video time is also crucial for the evaluation of the transcoder cost.

For the comparison of the algorithms, both accuracy and performance of execution time metrics are used. We compare algorithms by both looking the results taken in unique interval time and accuracy metrics. The reason why accuracy metrics are used is that seeing the total picture of the algorithms' performance in all data loads.

In general, X-axis parameters of the graphs show us the time interval and Y-axis represents the cost and time according to the content of the graph. The results of the each algorithm defined in Chapter 3 are given in the subsections from 4.3.3.1 to 4.3.3.4. We start with the exact solution provided by the MIQP and compare the proposed algorithms with the exact solution to show how the proposed optimization technique approaches to the real values.

#### 4.3.3.1 Experimental Results of Mixed Integer Quadratic Programming

The case study of Microsoft Azure services under second network traffic also shows us the necessity of the proposed algorithm when we compare it with FIFO algorithm. The difference between the result of the FIFO algorithm and MIQP algorithm is \$96.89 where actual result is \$129.15 and the result of FIFO is \$226.04. This algorithm provides suppliers a gain almost 43% in terms of cost by virtue of this systematic approach for the allocation. The proposed algorithm not only reduces the cost spend on cloud solutions, but also increases the

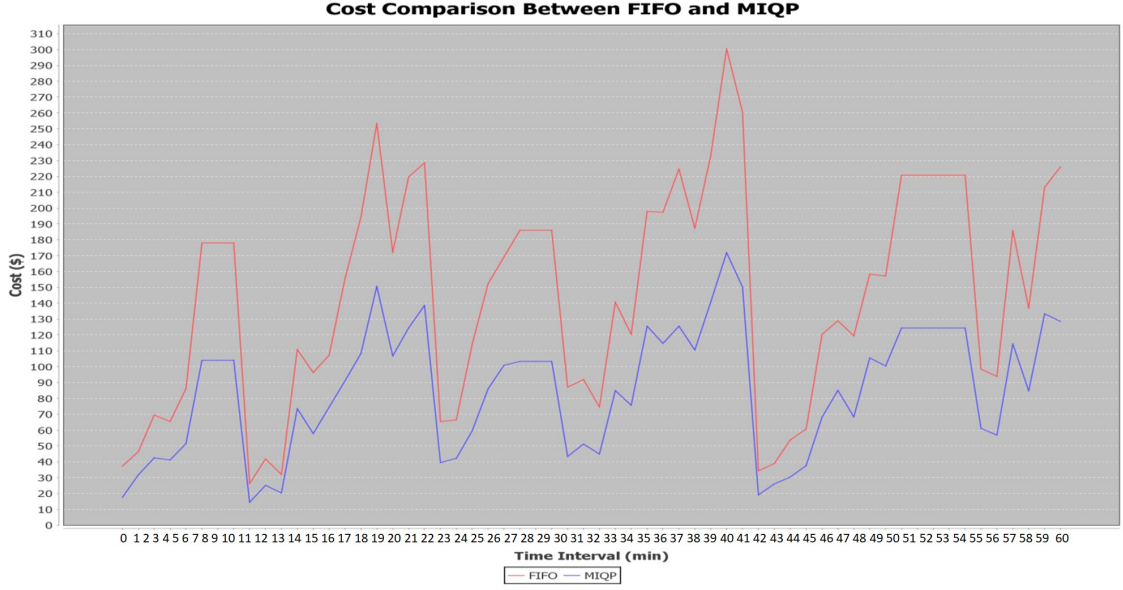


Figure 4.16: Comparison between the results of FIFO resource allocation technique with MIQP with varying user video requests

customer satisfaction by considering quality of services like latency and security. The Figure 4.16 represents that under all high and low data load, the proposed model yields better results. This result shows that there is significant cost saving in cloud resources while increasing the user satisfaction. In the proposed mathematical model, since resources have constraints and different cost strategies, it minimizes the total cost of used cloud resources by assigning videos of larger sizes to the resources of lower cost and assigning videos of smaller sizes to the resources of larger cost which obviously demonstrates the performance of our proposed algorithm. However, due to the complexity of the proposed algorithm, heuristic techniques which are modifications of PSO evolutionary technique are used to reach almost optimal solution. In the Section 4.3.3.2 we examined the results of this optimization technique in detail.

#### 4.3.3.2 Experimental Results of Modified Versions of Binary PSO

As in the case study of 4.3.2, in this part, we compare the existing variants of Binary PSO techniques which are classic Binary PSO and Novel Binary PSO in the literature to our proposed Neighbourhood Novel Binary PSO. As in the above case, classic binary PSO does not yield good results when we compared to the Novel Binary PSO. WMAPE and NRMS values of Classic Binary PSO are 0.24 and 0.23 respectively, which are fairly high values, obviously seen in the Table 4.13. Proposed Novel Binary PSO in the study of Khanesar et al. yields better results when we compared it to classic Binary PSO but it still suffers from approaching to accurate results and also it results in higher execution time due to the complexity of the proposed algorithm solution [Khanesar et al., 2007]. Execution time arises from 4,829 ms to 11,376 ms and total cost is decreased from \$165.5 to \$157.89. The WMAPE value of Novel Binary PSO is 0.19. These consequences results in a necessity to propose a new algorithm. Our proposed Neighbourhood Novel Binary PSO yields better results than both classic BPSO and Novel BPSO; however, it still has worse experience in execution time.

Table 4.13: Comparison of Algorithms Proposed in a Time Interval

Optimization Technique	Number of Generations	Total Running Time (ms)	Cost (\$)
FIFO	50	NA	207.61
	100	NA	203.53
	200	NA	205.61
	500	NA	168.67
	1,000	NA	226.04
MIQP	50	NA	129.34
	100	NA	128.43
	200	NA	123.93
	500	NA	98.17
	1,000	NA	128.53
Binary PSO	50	222	171.84
	100	465	168.45
	200	953	158.0
	500	2,462	126.37
	1,000	4,829	165.5
Novel Binary PSO	50	552	164.16
	100	1,113	160.04
	200	2,264	155.39
	500	5,756	123.41
	1,000	11,376	157.89
Neighbourhood Novel Binary PSO	50	531	146.41
	100	1,078	143.44
	200	2,182	136.66
	500	5,548	106.21
	1,000	10,957	135.09
Multi-Swarm Neighbourhood Novel Binary PSO	50	425	138.43
	100	852	131.78
	200	1,737	126.18
	500	4,440	99.14
	1,000	8,852	129.03
Parallel Multi-Swarm Neighbourhood Novel Binary PSO	50	262	138.20
	100	538	131.49
	200	1,129	126.17
	500	3,072	99.21
	1,000	6,006	129.15

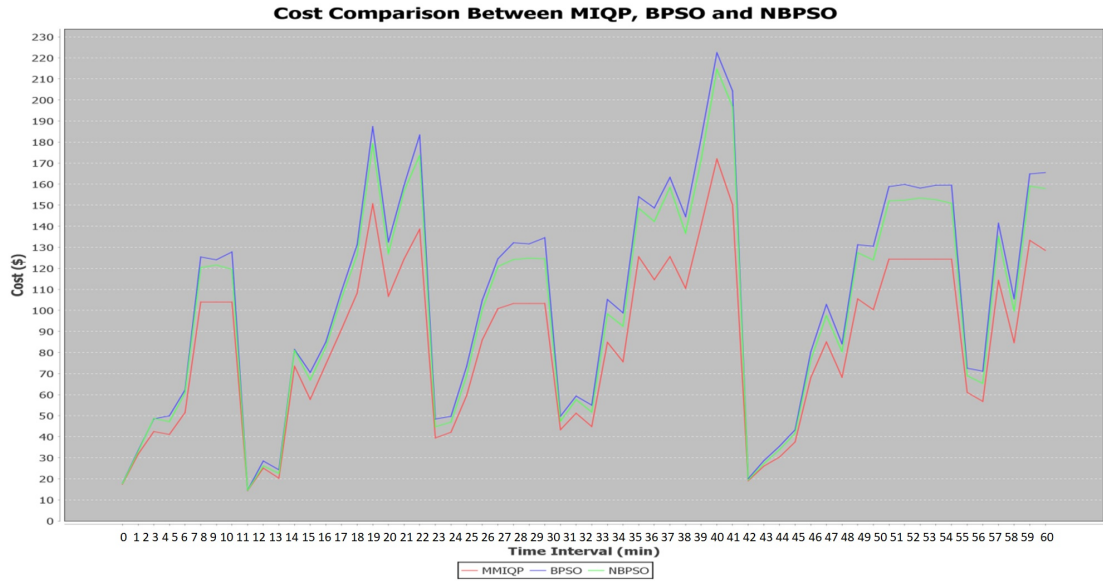


Figure 4.17: Comparison between the results of 1,000 iterative Binary BPSO, Novel Binary PSO and MIQP in varying user video requests

In the Figure 4.17, the blue line represents the cost of BPSO and green line represents the cost of Novel BPSO. The aim of this optimization technique is to reach the values of the red line which is the indicator of exact solutions of MIQP. Novel BPSO is better than Binary PSO as can be seen evidently in Figure 4.17. In the Table 4.14, comparison of the metrics of all used optimization techniques in this study under different number of generations are given. The table presents that the results of the algorithms become more impressive when the number of generation increases as expected. Results of the all forecast metrics approve the improvement of our proposed algorithm.

Figure 4.18 shows the improvement of Neighbourhood Novel BPSO by comparing to Novel BPSO. The results apparently reveal that proposed PSO gives better results than Novel BPSO. WMAPE value becomes 0.03 and cost becomes \$135.09 where actual cost is \$128.53. On the other hand, the proposed algorithm does not improve the Novel BPSO in terms of the execution time. Run time is decreased from 11,376 ms to only 10,957 ms under 1,000 iteration seen in Table 4.13.

#### 4.3.3.3 Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm

Up to now, all algorithms used is an approach of single swarm and does not give impressive results especially in terms of execution time. Multi-swarm application is also used for different network traffics and Microsoft Azure cloud service providers. This approach also yields very good results like in the previous case due to the rationality between the regions in cost of resources like Amazon case study. Also, since Microsoft Azure does not charge bandwidth cost between their regions (inbound data transfer cost is equal to 0), then sticking into local optima caused by greedy algorithm in the proposed algorithm does not affect this Azure case. The Figure 4.18 clearly shows that Multi-swarm approach yields dramatic results from both accuracy and run time. In Table 4.14, WMAPE metric value is decreased from 0.19

Table 4.14: Comparison of Different Variants of Binary PSO

<b>Optimization Technique</b>	<b>Number of Generations</b>	<b>WMAPE</b>	<b>RMSE</b>	<b>NRMSE</b>
Binary PSO	50	0.28	26.17	0.25
	100	0.27	25.40	0.25
	200	0.26	24.55	0.25
	500	0.25	24.63	0.24
	1,000	0.24	24.67	0.23
Novel BPSO	100	0.24	22.93	0.23
	100	0.22	21.07	0.21
	200	0.21	20.43	0.21
	500	0.19	19.94	0.21
	1,000	0.19	19.66	0.19
Neighbourhood Novel BPSO	100	0.12	12.32	0.13
	100	0.09	9.35	0.11
	200	0.07	7.25	0.08
	500	0.05	5.99	0.07
	1,000	0.03	4.49	0.05
Multi-Swarm Neighbourhood Novel BPSO	100	0.04	5.34	0.06
	100	0.02	2.92	0.03
	200	0.01	1.42	0.018
	500	0.005	0.74	0.009
	1,000	0.002	0.44	.05
Parallel Multi-Swarm Neighbourhood Novel BPSO	50	0.05	5.52	0.06
	100	0.02	2.65	0.03
	200	0.01	1.43	0.018
	500	0.005	0.67	0.008
	1,000	0.002	0.47	0.005



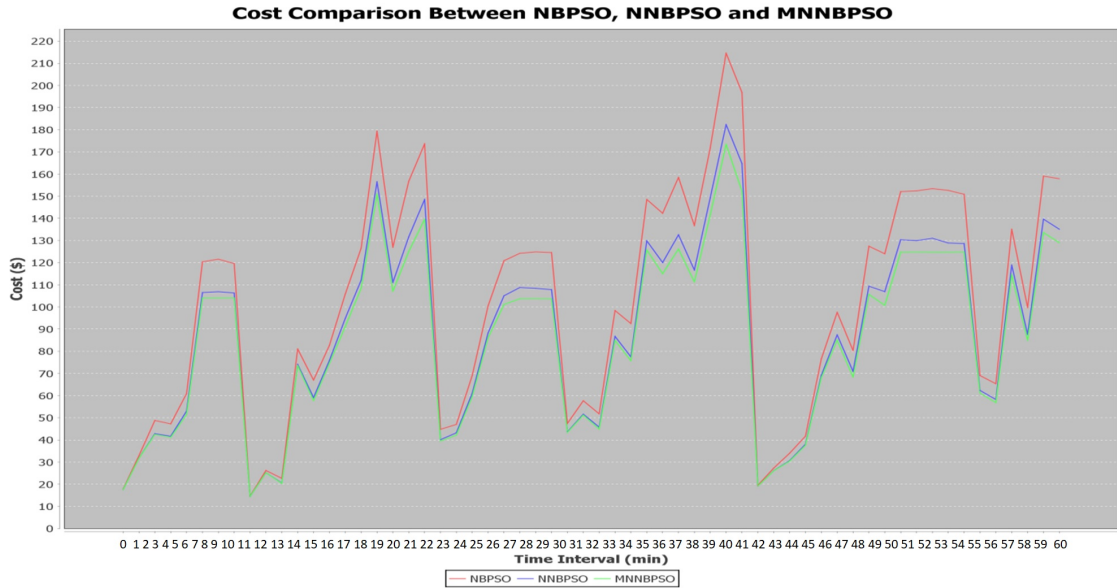


Figure 4.18: Comparison between the results of 1,000 iterative Novel BPSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

of NBPSO to the 0.002 of the proposed algorithm which shows the impressive and sharp enhancement in the optimization technique. Along with, there is a decrease in execution time drastically. Figure 4.19 manifests that Multi-Swarm PSO run faster than the other algorithms implemented up to now.

As a result, as shown in Table 4.13, the predicted cost falls from \$157.89 to \$129.03 by the help of these algorithms and execution time falls from 11,375 ms to 8,852 ms when we compare our proposed Multi-swarm PSO with Novel Binary PSO.

#### 4.3.3.4 Experimental Results of Multi-Swarm Particle Swarm Optimization with Greedy Algorithm using Parallel Programming

Although we achieve considerable drops in execution time of the algorithm, to get a better results in execution time, we parallelized the multi-swarm PSO. After parallelizing multi-swarm PSO, there is a significant decline in execution time of the algorithm. Figure 4.20 shows that parallel algorithm is run faster than the Multi-Swarm PSO. And, as expected, this approach does not show an improvement in the accuracy of the algorithm. As stated in Table 4.9, execution time is decreased from 8,852 ms to 6,006 ms ( $\approx 33\%$ ). This is very substantial and satisfactory value in terms of the increasing the performance.

To conclude, although algorithms are run under different network traffics and different service providers, they give similar results like in the previous case study. Each this study, proposed algorithm improves the existing algorithms in terms of both accuracy and execution time. Table 4.13 and Table 4.14 show the results of the algorithms under different number of generations. Table 4.13 shows the execution time and cost evaluated by the algorithms in a given unique time interval. Moreover, Table 4.14 shows the metrics results for each algorithm under 5 distinct numbers of iterations. Instead of a specified data load, these metrics analyze the

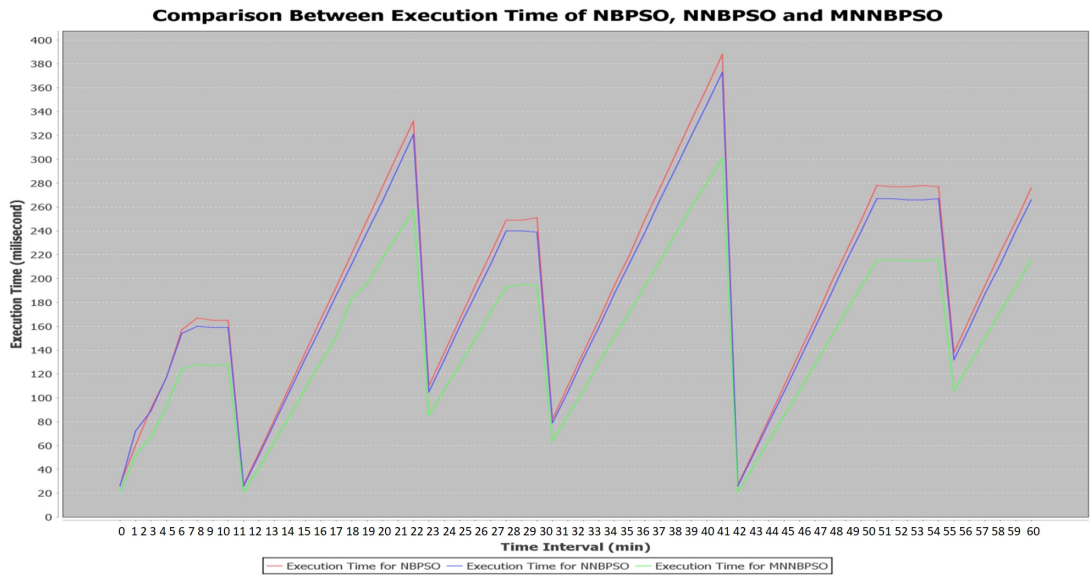


Figure 4.19: Comparison between the computing time of 1,000 iterative Novel Binary PSO, Neighbourhood Topology Novel BPSO and Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

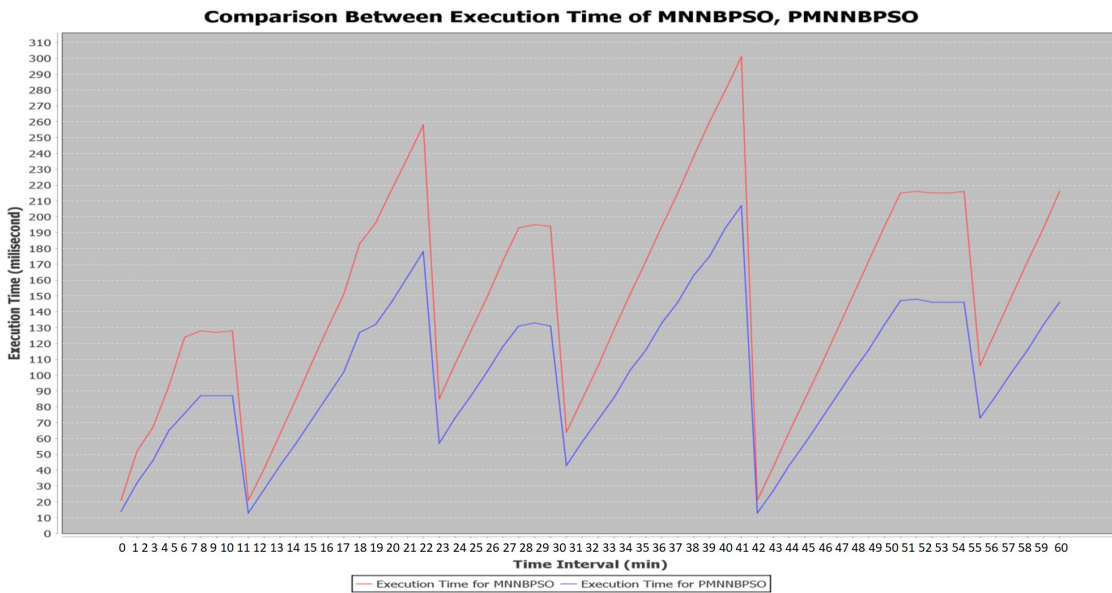


Figure 4.20: Comparison between the computing time of Multi-Swarm Neighbourhood Novel Binary PSO with Parallel Multi-Swarm Neighbourhood Novel Binary PSO with varying user video requests

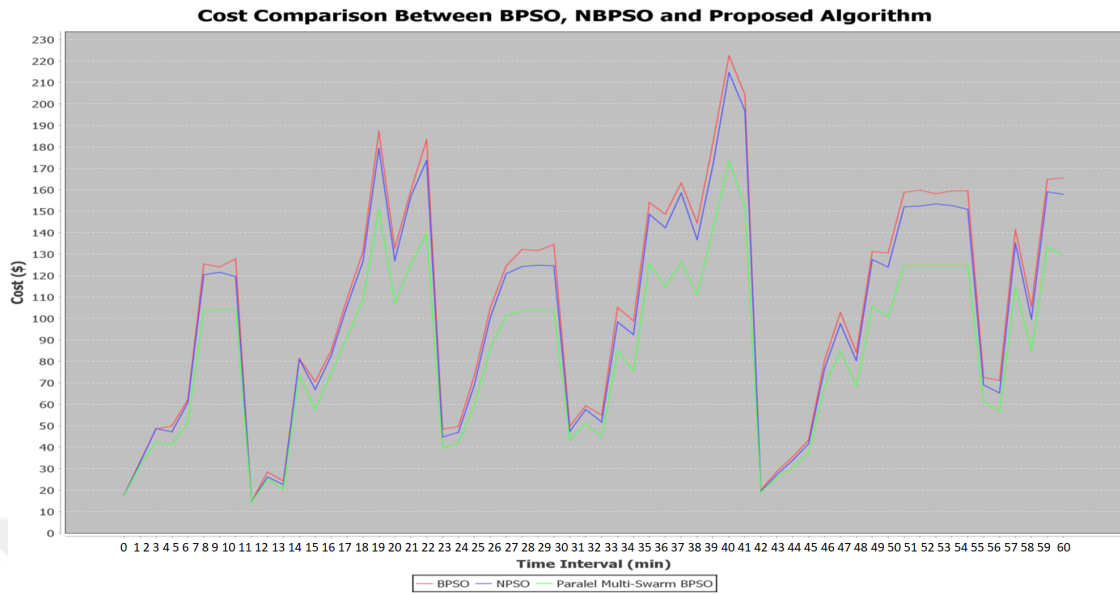


Figure 4.21: Comparison of the results of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests

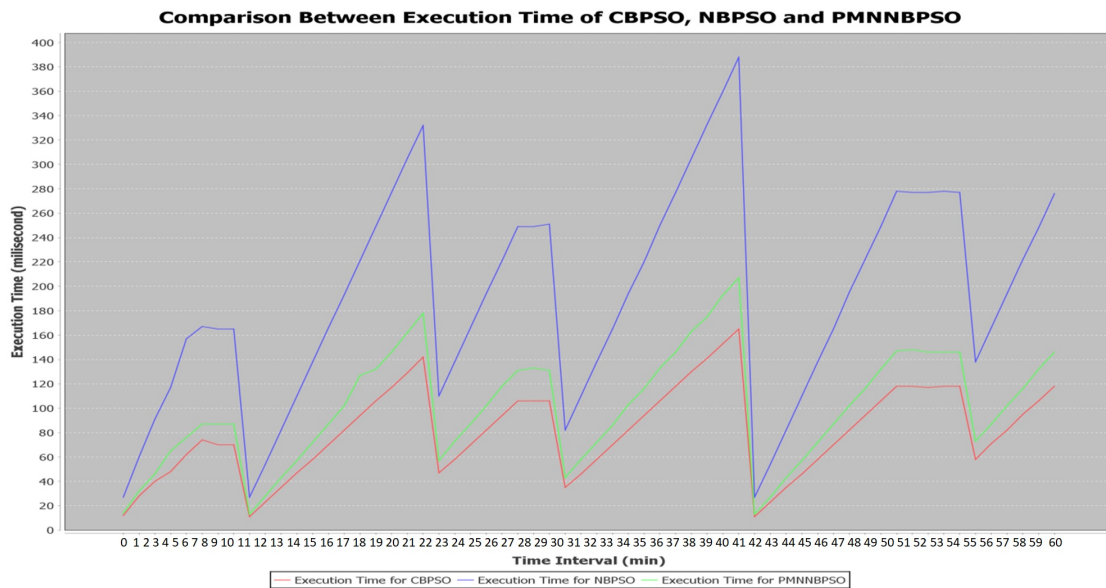


Figure 4.22: Comparison between the computing time of Binary PSO, Novel Binary PSO and Parallel Multi-Swarm Neighbourhood Novel Binary PSO in varying user video requests

algorithms under different loads. Our proposed algorithm solves the problem efficiently and achieves very close results to MIQP results as we see in the Figure ?? when we compare to Classic Binary PSo and Novel Binary PSO. In addition, cost of the VoD service providers based on cloud services dropped from \$226.04 to \$129.15 which is almost 43% cost saving by virtue of our proposed algorithm and proposed solution.



## CHAPTER 5

### CONCLUSION

Now that cloud computing technology is considered to be the key and the indispensable part of the future, and since the cost is always important to organizations, optimizing the expenditures on cloud computing becomes a very important issue. With VoD applications becoming very popular, it is inevitable to provide such applications over cloud services due to the easiness of management and no need to invest on physical resources. In this thesis, we provide a mathematical model to minimize the cost spend on cloud services while satisfying the QoS attributes of customer. Summary of the thesis and contribution are provided in Section 5.1. In the Section 5.2 presents the suggestions for the future work.

#### 5.1 Summary and Contributions of the Thesis Study

Although cloud resource allocation in cloud data centers is a widely studied and researched subject for cloud service providers, for the cloud services allocation based on CSP, there is insufficient study in the literature. Additionally, especially for video on demand applications, since needing huge number of storages and high computing capacity for streaming, using cloud services for VoD applications is not a smart approach for specifically small companies. At the same time, big companies try to minimize the cost spend on cloud services to increase their profits. For these reasons, the algorithm recommended to reduce cloud services spending is really a necessity for the business world where everything is built on the cloud. While minimizing the cost on cloud resources, we work on a very popular topic which is video streaming which makes this study more purposeful and feasible.

Cloud computing service providers offer many different services generally grouped into 3 main classes; SaaS, PaaS and IaaS. While optimizing services, VMs, type of IaaS, are popular and studied mostly in the literature, in this study; SaaS solutions are preferred too in addition to IaaS solutions. In the first approach, VMs and storages are used for video streaming. With the development of technology, other resources; CDN and transcoder that are indispensable for VoD applications are used in the formulation of the problem. Although the cost seems to be the main target, QoS parameters should be considered for such applications to ensure the continuity of the users or decrease the churn rate.

The latency is one of the first QoS that has attention when talking about video applications. Since, public cloud services are used; internet is the network that transfers data from cloud to users. Minimizing latency in the Internet world is the totally another subject and studied for many years in the literature. In the first problem formulation, to reduce latency, the regions

of users and services tried to be kept same. This not only reduces latency but also reduces the cost. By considering cost and QoS parameters, first mathematical model is constructed which is dynamic than can be adapted to different cloud system providers and different number of videos by simply setting the inputs. Since our problem is multi-objective, we use the approach that takes the cost as the main objective and the other objectives are handled as constraints i.e. while exploring our problem, all QoS parameters are added as constraints and the main objective of the problem is the cost function that must be minimized. Cost function is designed for scheduling requested videos to cloud system services to minimize the total cost. In the first problem, the transfer cost between cloud services are ignored which makes our problem linear. Then the problem is solved by using Linear Programming algorithm. We compared the results obtained by LP against FIFO scheduling algorithm and we found that LP based scheduling yields better results than basic scheduling algorithm, FIFO, in terms of cost.

Designed mathematical model achieves QoS requirements defined according to customer satisfaction. Due to the time inefficiency, then PSO evolutionary algorithm is used. By reason of the fact that our problem domain is binary and there are some constraints, we apply the modified version of PSO. To compare the proposed PSO performance, classic Binary PSO and Novel Binary PSO algorithms are used. This study not only reduces the cost spend on cloud services with the help of the proposed mathematical model for the problem but also improves the existing PSO in terms of both accuracy and execution time. Part of this study that deals with the first problem is published in one of the conferences in the area of computer science and engineering, International Conference on Computer Science and Engineering in 2017 [Aygün et al., 2017].

From there on, second problem formulation is developed. This problem contains the new cloud solutions; transcoder and CDN. Although the main objective function is still cost, the cost parameters differ according to the used cloud resources. Besides, in the second problem formulation new QoS parameters are put in. Firstly, content delivery networks are begun to be used which are very popular networks for VoD applications to minimize latency between cloud services and users. Along with, resource allocation by looking at the latency values between the service and the user, it also reduces the time required to serve the video. So, while allocating videos to the reasonable resources, services that have high latency values from the customers are ignored. Video quality is another QoS that is very important for the satisfaction of the user. According to the user profile, suitable video format should be streamed. In that case, transcoders cloud services should be used which unfortunately adds extra cost to the VoD provider. While serving the appropriate video format improves quality, selecting the appropriate transcoder needed to translate that format reduces the cost. Since public cloud resources are used, security is another QoS parameter that should be consider for the satisfaction of the customer. For the videos, streaming them over secured http and encrypting them in the storage are two approaches that are studied during this study. Considering these additional QoS parameters put additional burden on the cost that makes the cost function expanded. These additional burden include security options cost provided by cloud providers, transcoding cost to change format of the video, content delivery network cost to cache and send video fast. Besides, defining security constraints, location constraints, and video quality constraints makes the problem more sophisticated. Moreover, solving this kind of problem needs more sophisticated algorithms such as evolutionary algorithms.

By virtue of these reasons and due to the complexity of the problem, new versions of the previous proposed methods are proposed. Firstly, since the function becomes the quadratic caused by the transfer cost between services, solving it by using linear programming becomes

impossible. Branch and cut method is used to solve the problem. Also, a new version of PSO algorithm is proposed to increase the accuracy and to decrease the execution time. This approach divides one swarm into multiple swarms (the size of the sub-swarm is equal to the number of different cloud services) and each sub-swarm is run separately. This optimization technique not only increases the accuracy but also reduces the execution time drastically. Because, as a communication between sub-swarms, greedy algorithm is defined in which the time to execute the algorithm decreases significantly. After that, since the nature of proposed algorithm is reasonable for parallel programming, then each swarm runs in parallel. This last approach considerably reduces the execution time. For the execution of the algorithm, input video data is created by using the statistics of YouTube. For the first approach for the solution of the problem, price of Microsoft Azure Cloud Services, and for the second approach, price of Amazon Web Services are applied. The results represent that such a mathematical model reduces the cost of cloud services by almost 57 %. Furthermore, the proposed algorithm yields better results and have more robust performance than the popular PSO algorithms proposed in the literature. Besides all this, by using the services provided by CSP, how to satisfy the QoS parameters are analyzed and detailed during the study.

## 5.2 Future Work

In the study, only public cloud services are examined and there is no restriction in the capacity of the resources. For the rest of this study as a future work, the network can be designed using private cloud resources, and quality of service for video streaming applications can be expanded. Also, to concentrate the security of the services intensively, Virtual Private Networks (VPN) can be designed. Besides, for charging strategy only on-demand pricing strategy of the companies is handled. Other pricing approaches such as bidding or reservation may be accounted for the cost of cloud resources. In video requests, even if a video has already been requested many times, every request is treated as if it were requested for the first time. Considering previous requests by using prediction methods, different formats of frequently used videos can be kept in existing storage which significantly reduces the time and cost required for the transcoder. Besides, although this approach is designed for CBSP, the requests for CSP can be solved by using the proposed algorithms.





## REFERENCES

- [Abd-El-Wahed et al., 2011] Abd-El-Wahed, W., Mousa, A., and El-Shorbagy, M. (2011). Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems. *Journal of Computational and Applied Mathematics*, 235(5):1446 – 1453.
- [AC, 2018] AC, A. C. (2018). *Latency in live network video surveillance*. [https://www.axis.com/files/whitepaper/wp\\_latency\\_live\\_netvid\\_63380\\_external\\_en\\_1504\\_lo.pdf](https://www.axis.com/files/whitepaper/wp_latency_live_netvid_63380_external_en_1504_lo.pdf).
- [Adamuthe et al., 2013] Adamuthe, A. C., Bhise, V. K., and Thampi, G. T. (2013). Solving resource provisioning in cloud using gas and pso. In *2013 Nirma University International Conference on Engineering (NUiCONE)*, pages 1–5.
- [Albert, 1999] Albert, S. (1999). Solving mixed integer linear programs using branch and cut algorithm. *North Carolina State University, pdf.[Online]. Available: <http://www4.ncsu.edu/~kksivara/mastersthesis/shon-thesis.pdf> (visited on 07/05/2017)*.
- [Ali-Eldin et al., 2015] Ali-Eldin, A., Kihl, M., Tordsson, J., and Elmroth, E. (2015). Analysis and characterization of a video-on-demand service workload. In *Proceedings of the 6th ACM Multimedia Systems Conference, MMSys '15*, pages 189–200, New York, NY, USA. ACM.
- [Amazon Web Services, 2017] Amazon Web Services, I. (2018 (accessed November 15, 2017)). *Amazon Web Services*. <https://aws.amazon.com>.
- [Andrzejak et al., 2010] Andrzejak, A., Kondo, D., and Yi, S. (2010). Decision model for cloud computing under sla constraints. In *2010 IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 257–266.
- [Aoun et al., 2010] Aoun, R., Doumith, E. A., and Gagnaire, M. (2010). Resource provisioning for enriched services in cloud environment. In *2nd IEEE International Conference on Cloud Computing Technology and Science*, pages 296–303. IEEE.
- [Armbrust et al., 2010] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.
- [Armonk, 2013] Armonk, N. (2013). *IBM Study Reveals Businesses Using Cloud Computing for Competitive Advantage Can Generate Double Revenue and Profit Compared to their Peers*. <https://www.prnewswire.com/news-releases/ibm-study-reveals-businesses-using-cloud-computing-for-competitive-advantage>.

-can-generate-double-revenue-and-profit-compared-to-their-peers-229098871.html.

- [Aurrecochea et al., 1998] Aurrecochea, C., Campbell, A. T., and Hauw, L. (1998). A survey of qos architectures. *Multimedia Systems*, 6(3):138–151.
- [Aygün et al., 2017] Aygün, B., Arıcı, N., and Coşar, A. (2017). Optimal resource allocation of dynamic video streaming applications in a public cloud environment. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pages 445–450.
- [Bai, 2010] Bai, Q. (2010). Analysis of particle swarm optimization algorithm. *Computer and information science*, 3(1):180.
- [Baldi and Bruglieri, 2017] Baldi, M. M. and Bruglieri, M. (2017). On the generalized bin packing problem. *International Transactions in Operational Research*, 24(3):425–438.
- [Barba-Jimenez et al., 2016] Barba-Jimenez, C., Ramirez-Velarde, R., Tchernykh, A., Rodríguez-Dagnino, R., Nolasco-Flores, J., and Perez-Cazares, R. (2016). Cloud based video-on-demand service model ensuring quality of service and scalability. *Journal of Network and Computer Applications*, 70:102 – 113.
- [Bhise and Mali, 2013] Bhise, V. K. and Mali, A. S. (2013). Cloud resource provisioning for amazon ec2. In *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, pages 1–7. IEEE.
- [Botta et al., 2016] Botta, A., de Donato, W., Persico, V., and Pescapé, A. (2016). Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*, 56:684–700.
- [Brasileiro et al., 2017] Brasileiro, R. C., Souza, V. L., and Oliveira, A. L. (2017). Automatic trading method based on piecewise aggregate approximation and multi-swarm of improved self-adaptive particle swarm optimization with validation. *Decision Support Systems*, 104:79 – 91.
- [Broberg et al., 2009] Broberg, J., Buyya, R., and Tari, Z. (2009). Metacdn: Harnessing ‘storage clouds’ for high performance content delivery. *Journal of Network and Computer Applications*, 32(5):1012 – 1022. Next Generation Content Networks.
- [Burgess and Green, 2013] Burgess, J. and Green, J. (2013). *YouTube: Online video and participatory culture*. John Wiley & Sons.
- [Cagnina et al., 2008] Cagnina, L. C., Esquivel, S. C., and Coello, C. A. C. (2008). Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32(3).
- [Chaisiri et al., 2011a] Chaisiri, S., Kaewpuang, R., Lee, B. S., and Niyato, D. (2011a). Cost minimization for provisioning virtual servers in amazon elastic compute cloud. In *2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, pages 85–95.

- [Chaisiri et al., 2009] Chaisiri, S., Lee, B.-S., and Niyato, D. (2009). Optimal virtual machine placement across multiple cloud providers. In *Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific*, pages 103–110. IEEE.
- [Chaisiri et al., 2011b] Chaisiri, S., Lee, B. S., and Niyato, D. (2011b). Optimization of resource provisioning cost in cloud computing. *5(2):164–177*.
- [Chen et al., 2014] Chen, K. T., Chang, Y. C., Hsu, H. J., Chen, D. Y., Huang, C. Y., and Hsu, C. H. (2014). On the quality of service of cloud gaming systems. *IEEE Transactions on Multimedia*, 16(2):480–495.
- [Chen et al., 2015a] Chen, L., Duan, Y., Qiu, M., Xiong, J., and Gai, K. (2015a). Adaptive resource allocation optimization in heterogeneous mobile cloud systems. In *The 2nd IEEE International Conference on Cyber Security and Cloud Computing*, pages 19–24.
- [Chen et al., 2015b] Chen, W., Lu, Y. H., and Hacker, T. J. (2015b). Adaptive cloud resource allocation for analysing many video streams. In *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 17–24.
- [Chen et al., 2015c] Chen, Y., Wu, K., and Zhang, Q. (2015c). From qos to qoe: A tutorial on video quality assessment. *IEEE Communications Surveys Tutorials*, 17(2):1126–1165.
- [Cheng et al., 2007] Cheng, X., Dale, C., and Liu, J. (2007). Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study. *ArXiv e-prints*.
- [Cores and Subsystems, 2017] Cores, D. I. and Subsystems (2013 (accessed October 15, 2017)). *White Paper: Understanding—and Reducing—Latency in Video Compression Systems*. <http://www.cast-inc.com/blog/white-paper-understanding-and-reducing-latency-in-video-compression-systems>.
- [Cui and Weile, 2005] Cui, S. and Weile, D. S. (2005). Application of a parallel particle swarm optimization scheme to the design of electromagnetic absorbers. *IEEE Transactions on Antennas and Propagation*, 53(11):3616–3624.
- [Dash, 2005] Dash, S. (2005). Exponential lower bounds on the lengths of some classes of branch-and-cut proofs. *Mathematics of Operations Research*, 30(3):678–700.
- [Deb, 2000] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 186(2-4):311–338.
- [Dhir et al., 2016] Dhir, V., Chakraverti, A. K., and Chakraverti, S. (2016). Architectural and qos issues in mobile cloud computing environment for real-time video streaming. *International Journal of Advanced Computer Science and Applications ISSN*.
- [Eberhart and Shi, 2001] Eberhart, R. C. and Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pages 94–100. IEEE.

- [Espadas et al., 2013] Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., and Concha, D. (2013). A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures. *Future Gener. Comput. Syst.*, 29(1):273–286.
- [Esposito et al., 2016] Esposito, C., Ficco, M., Palmieri, F., and Castiglione, A. (2016). Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory. *IEEE Transactions on Computers*, 65(8):2348–2362.
- [Finamore et al., 2011] Finamore, A., Mellia, M., Munafò, M. M., Torres, R., and Rao, S. G. (2011). Youtube everywhere: Impact of device and infrastructure synergies on user experience. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 345–360. ACM.
- [Forret, 2017] Forret, P. (2017). *Video filesize calculator*. <https://toolstud.io/video/filesize.php>.
- [Google, 2017] Google (2018 (accessed October, 2017)). *Google Cloud Platform*. <https://cloud.google.com>.
- [Gorde et al., 2014] Gorde, A., Gandhi, N., Mishra, R., Pathak, R., and Padmavathi, B. (2014). Prediction based outcome for media streaming applications. In *International Journal of Computer Application*.
- [He et al., 2014] He, J., Wen, Y., Huang, J., and Wu, D. (2014). On the cost x2013;qoe trade-off for cloud-based video streaming under amazon ec2’s pricing models. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(4):669–680.
- [He et al., 2012] He, Q., Han, J., Yang, Y., Grundy, J., and Jin, H. (2012). Qos-driven service selection for multi-tenant saas. In *Cloud computing (cloud), 2012 ieee 5th international conference on*, pages 566–573. IEEE.
- [He et al., 2016] He, Q., Liu, J., Wang, C., and Li, B. (2016). Coping with heterogeneous video contributors and viewers in crowdsourced live streaming: A cloud-based approach. *IEEE Transactions on Multimedia*, 18(5):916–928.
- [Hossain et al., 2012] Hossain, M. S., Hassan, M. M., Qurishi, M. A., and Alghamdi, A. (2012). Resource allocation for service composition in cloud-based video surveillance platform. In *2012 IEEE International Conference on Multimedia and Expo Workshops*, pages 408–412.
- [Hu et al., 2016] Hu, H., Wen, Y., Chua, T. S., Huang, J., Zhu, W., and Li, X. (2016). Joint content replication and request routing for social video distribution over cloud cdn: A community clustering method. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(7):1320–1333.
- [Hu and Eberhart, 2002] Hu, X. and Eberhart, R. (2002). Solving constrained nonlinear optimization problems with particle swarm optimization. 2002:203–206.

- [Hu et al., 2003] Hu, X., Eberhart, R. C., and Shi, Y. (2003). Engineering optimization with particle swarm. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 53–57.
- [Inc., 2010] Inc., A. (2010). *Apple Final Cut Pro 7 User Manual*. [https://documentation.apple.com/en/finalcutpro/usermanual/Final%20Cut%20Pro%207%20User%20Manual%20\(en\).pdf](https://documentation.apple.com/en/finalcutpro/usermanual/Final%20Cut%20Pro%207%20User%20Manual%20(en).pdf).
- [Irvine and Levin, 2000] Irvine, C. and Levin, T. (2000). Quality of security service. In *Proceedings of the 2000 Workshop on New Security Paradigms*, NSPW '00, pages 91–99, New York, NY, USA. ACM.
- [Jamali et al., 2016] Jamali, S., Alizadeh, F., and Sadeqi, S. (2016). Task scheduling in cloud computing using particle swarm optimization. In *International Caucasus Universities Association Graduate Students Symposium*, pages 192–198.
- [Jeong et al., 2010] Jeong, Y. W., Park, J. B., Jang, S. H., and Lee, K. Y. (2010). A new quantum-inspired binary pso: Application to unit commitment problems for power systems. *IEEE Transactions on Power Systems*, 25(3):1486–1495.
- [Juluri et al., 2016] Juluri, P., Tamarapalli, V., and Medhi, D. (2016). Measurement of quality of experience of video-on-demand services: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):401–418.
- [Karpát and Özel, 2006] Karpát, Y. and Özel, T. (2006). Swarm-intelligent neural network system (sinns) based multi-objective optimization of hard turning. *Transactions of NAMRI/SME*, 34:9–16.
- [Kaur and Mehta, 2017] Kaur, P. and Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented shuffled frog leaping algorithm. *Journal of Parallel and Distributed Computing*, 101:41 – 50.
- [Kennedy and Eberhart, 1997] Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, volume 5, pages 4104–4108. IEEE.
- [Khanesar et al., 2007] Khanesar, M. A., Teshnehlab, M., and Shoorehdeli, M. A. (2007). A novel binary particle swarm optimization. In *2007 Mediterranean Conference on Control Automation*, pages 1–6.
- [Kim et al., 2011] Kim, J.-Y., Mun, K.-J., Kim, H.-S., and Park, J. H. (2011). Optimal power system operation using parallel processing system and pso algorithm. *International Journal of Electrical Power Energy Systems*, 33(8):1457 – 1461.
- [Kim et al., 2008] Kim, T.-H., Maruta, I., and Sugie, T. (2008). Robust pid controller tuning based on the constrained particle swarm optimization. *Automatica*, 44(4):1104 – 1110.

- [Kiziloz et al., 2018] Kiziloz, H. E., Deniz, A., Dokeroglu, T., and Cosar, A. (2018). Novel multiobjective tlbo algorithms for the feature subset selection problem. *Neurocomputing*.
- [Klymash et al., 2014] Klymash, M., Beshley, M., Strykhalyuk, B., and Maksymyuk, T. (2014). Research and development the methods of quality of service provision in mobile cloud systems. In *2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 160–164.
- [Ko et al., 2014] Ko, R. K. L., Tan, A. Y. S., and Ng, G. P. Y. (2014). 'time' for cloud? design and implementation of a time-based cloud resource management system. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 530–537.
- [Koh et al., 2006] Koh, B.-I., George, A. D., Haftka, R. T., and Fregly, B. J. (2006). Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering*, 67(4):578–595.
- [Kolassa and Schütz, 2007] Kolassa, S. and Schütz, W. (2007). Advantages of the mad/mean ratio over the mape. 6:40–43.
- [Koziel and Michalewicz, 1999] Koziel, S. and Michalewicz, Z. (1999). Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evol. Comput.*, 7(1):19–44.
- [Lee and b. Park, 2006] Lee, K. Y. and b. Park, J. (2006). Application of particle swarm optimization to economic dispatch problem: Advantages and disadvantages. In *2006 IEEE PES Power Systems Conference and Exposition*, pages 188–192.
- [Legillon et al., 2013a] Legillon, F., Melab, N., Renard, D., and Talbi, E. G. (2013a). Cost minimization of service deployment in a multi-cloud environment. In *2013 IEEE Congress on Evolutionary Computation*, pages 2580–2587.
- [Legillon et al., 2013b] Legillon, F., Melab, N., Renard, D., and Talbi, E. G. (2013b). Cost minimization of service deployment in a public cloud environment. In *2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pages 491–498.
- [Li et al., 2016] Li, X., Salehi, M. A., Bayoumi, M., and Buyya, R. (2016). Cvss: A cost-efficient and qos-aware video streaming using cloud services. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 106–115.
- [Li et al., 2010] Li, Z., Bin, Z., Ying, L., Yan, G., and Zhi-Liang, Z. (2010). A web service qos prediction approach based on collaborative filtering. In *Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific*, pages 725–731. IEEE.
- [Liang and Suganthan, 2005] Liang, J. J. and Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer with local search. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 522–528 Vol.1.

- [Lin, 2005] Lin, I.-L. (2005). Particle swarm optimization for solving constraint satisfaction problems. Master’s thesis, Simon Fraser University.
- [Maksiweb, 2017] Maksiweb (2018 (accessed March 3, 2017)). *CDN nedir?* <https://www.maksiweb.com/barindirma-hizmetleri/cdn/>.
- [Malawski et al., 2013] Malawski, M., Figiela, K., and Nabrzyski, J. (2013). Cost minimization for computational applications on hybrid cloud infrastructures. *Future Generation Computer Systems*, 29(7):1786 – 1794. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services Cloud Computing and Scientific Applications — Big Data, Scalable Analytics, and Beyond.
- [Manuel, 2015] Manuel, P. (2015). A trust model of cloud computing based on quality of service. *Annals of Operations Research*, 233(1):281–292.
- [Mao et al., 2013] Mao, L., Yang, Y., and Xu, H. (2013). Design and optimization of cloud-oriented workflow system. *JSW*, 8(1):251–258.
- [Mao and Humphrey, 2011a] Mao, M. and Humphrey, M. (2011a). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12.
- [Mao and Humphrey, 2011b] Mao, M. and Humphrey, M. (2011b). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12.
- [McNabb et al., 2007] McNabb, A. W., Monson, C. K., and Seppi, K. D. (2007). Parallel pso using mapreduce. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 7–14. IEEE.
- [mei Yu et al., 2004] mei Yu, X., yin Xiong, X., and wu Wu, Y. (2004). A pso-based approach to optimal capacitor placement with harmonic distortion consideration. *Electric Power Systems Research*, 71(1):27 – 33.
- [Mell and Grance, 2009] Mell, P. and Grance, T. (2009). The NIST definition of Cloud computing Natl. Inst. Stand. Technol. 53(6):50.
- [Meng et al., 2010] Meng, K., Wang, H. G., Dong, Z., and Wong, K. P. (2010). Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE Transactions on Power Systems*, 25(1):215–222.
- [Menhas et al., 2012] Menhas, M. I., Wang, L., Fei, M., and Pan, H. (2012). Comparative performance analysis of various binary coded pso algorithms in multivariable pid controller design. *Expert Systems with Applications*, 39(4):4390 – 4401.
- [Microsoft, 2016] Microsoft (2018 (accessed February, 2016)). *Microsoft Azure*. <http://nsrcac.rutgers.edu/tass1/projects/Adaptive%20QoS%20for%20Multimedia%20Applications%20in%20Heterogeneous%20Networks>.

- [Microsoft, 2017] Microsoft (2018 (accessed October 15, 2017)). *Microsoft Azure*. <https://azure.microsoft.com>.
- [Mussi et al., 2011] Mussi, L., Daolio, F., and Cagnoni, S. (2011). Evaluation of parallel particle swarm optimization algorithms within the cuda<sup>TM</sup> architecture. *Information Sciences*, 181(20):4642 – 4657. Special Issue on Interpretable Fuzzy Systems.
- [Nahrstedt and Steinmetz, 1994] Nahrstedt, K. and Steinmetz, R. (1994). Resource management in multimedia networked systems. Technical report, University of Pennsylvania Department of Computer and Information Science.
- [Nan et al., 2014] Nan, G., Mao, Z., Li, M., Zhang, Y., Gjessing, S., Wang, H., and Guizani, M. (2014). Distributed resource allocation in cloud-based wireless multimedia social networks. *IEEE Network*, 28(4):74–80.
- [Niu et al., 2007] Niu, B., Zhu, Y., He, X., and Wu, H. (2007). Mcpsos: A multi-swarm cooperative particle swarm optimizer. *Applied Mathematics and computation*, 185(2):1050–1062.
- [Niu et al., 2012] Niu, D., Xu, H., Li, B., and Zhao, S. (2012). Quality-assured cloud bandwidth auto-scaling for video-on-demand applications. In *2012 Proceedings IEEE INFOCOM*, pages 460–468.
- [Nolle, 2009] Nolle, T. (2009). Network considerations in cloud computing. Technical report, techTarget.
- [Ostadrhimi et al., 2012] Ostadrhimi, L., Mariño, M. A., and Afshar, A. (2012). Multi-reservoir operation rules: Multi-swarm pso-based optimization approach. *Water Resources Management*, 26(2):407–427.
- [Pandey et al., 2010] Pandey, S., Wu, L., Guru, S. M., and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on*, pages 400–407. IEEE.
- [Parsopoulos et al., 2002] Parsopoulos, K. E., Vrahatis, M. N., et al. (2002). Particle swarm optimization method for constrained optimization problems. *Intelligent Technologies—Theory and Application: New Trends in Intelligent Technologies*, 76(1):214–220.
- [Passarella, 2012] Passarella, A. (2012). A survey on content-centric technologies for the current internet: Cdn and p2p solutions. *Computer Communications*, 35(1):1 – 32.
- [Paudyal et al., 2014] Paudyal, P., Battisti, F., and Carli, M. (2014). A study on the effects of quality of service parameters on perceived video quality. In *Visual Information Processing (EUVIP), 2014 5th European Workshop on*, pages 1–6. IEEE.
- [Pinto et al., 2015] Pinto, T., Alves, C., Mansi, R., and Valério de Carvalho, J. (2015). Solving the multiscenario max-min knapsack problem exactly with column generation and branch-and-bound. *Mathematical Problems in Engineering*, 2015.



- [Prasad et al., 2009] Prasad, S. K., Routray, S., Khurana, R., and Sahni, S., editors (2009). *A Novel Particle Swarm Optimization Approach for Grid Job Scheduling*, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Pulido and Coello, 2004] Pulido, G. T. and Coello, C. A. C. (2004). A constraint-handling mechanism for particle swarm optimization. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1396–1403. Ieee.
- [Qu and Buyya, 2014] Qu, C. and Buyya, R. (2014). A cloud trust evaluation system using hierarchical fuzzy inference system for service selection. In *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pages 850–857.
- [Rouse, 2018] Rouse, M. (2015 (accessed June 15, 2018)). *cloud service latency*. <http://whatis.techtarget.com/definition/cloud-service-latency>.
- [Salman et al., 2002] Salman, A., Ahmad, I., and Al-Madani, S. (2002). Particle swarm optimization for task assignment problem. 26:363–371.
- [Sandhu and Sood, 2015] Sandhu, R. and Sood, S. K. (2015). Scheduling of big data applications on distributed cloud based on qos parameters. *Cluster Computing*, 18(2):817–828.
- [Schrijver, 1986] Schrijver, A. (1986). *Theory of Linear and Integer Programming*. John Wiley Sons, Inc.
- [Services, 2017] Services, A. W. (2018 (accessed October, 2017)). *Amazon Simple Storage Service S3*. [https://aws.amazon.com/s3/?nc2=h\\_m1](https://aws.amazon.com/s3/?nc2=h_m1).
- [Shi and Eberhart, 1998] Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73. IEEE.
- [Solomon et al., 2011] Solomon, S., Thulasiraman, P., and Thulasiram, R. (2011). Collaborative multi-swarm pso for task matching using graphics processing units. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO '11*, pages 1563–1570, New York, NY, USA. ACM.
- [Su-hua et al., 2006] Su-hua, L., Yao-wu, W., Xin-yin, X., and Guang-yu, T. (2006). A parallel pso approach to multi-objective reactive power optimization with static voltage stability consideration. In *2005/2006 IEEE/PES Transmission and Distribution Conference and Exhibition*, pages 172–176.
- [Sujatha et al., 2007] Sujatha, D., Girish, K., Venugopal, K., and Patnaik, L. (2007). An integrated quality-of-service model for video-on-demand application. *IAENG International Journal of Computer Science*, 34(1).
- [Summers et al., 2012] Summers, J., Brecht, T., Eager, D., and Wong, B. (2012). Methodologies for generating http streaming video workloads to evaluate web server performance. In *Proceedings of the 5th Annual International Systems and Storage Conference, SYSTOR '12*, pages 2:1–2:12, New York, NY, USA. ACM.

- [Sun et al., 2014] Sun, L., Dong, H., Hussain, F. K., Hussain, O. K., and Chang, E. (2014). Cloud service selection: State-of-the-art and future research directions. *Journal of Network and Computer Applications*, 45:134 – 150.
- [Um et al., 2014] Um, T.-W., Lee, H., Ryu, W., and Choi, J. K. (2014). Dynamic resource allocation and scheduling for cloud-based virtual content delivery networks. *ETRI Journal*, 36(2):197–205.
- [Usman et al., 2016] Usman, M., He, X., Lam, K. M., Xu, M., Bokhari, S. M. M., and Chen, J. (2016). Frame interpolation for cloud-based mobile video streaming. *IEEE Transactions on Multimedia*, 18(5):831–839.
- [Van den Bergh and Engelbrecht, 2004] Van den Bergh, F. and Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation*, 8(3):225–239.
- [van der Zwet and Strom, 2018] van der Zwet, J. F. and Strom, D. (2013 (accessed July 3, 2018)). *Cloud Latency Issues? Dedicated Network Connections Will Help*. <https://techcrunch.com/2013/02/02/overcoming-obstacles-in-cloud-latency/>.
- [Venter and Sobieszczanski-Sobieski, 2006] Venter, G. and Sobieszczanski-Sobieski, J. (2006). Parallel particle swarm optimization algorithm accelerated by asynchronous evaluations. *Journal of Aerospace Computing, Information, and Communication*, 3(3):123–137.
- [Vieira et al., 2014] Vieira, C. C. A., Bittencourt, L. F., and Madeira, E. R. M. (2014). Reducing costs in cloud application execution using redundancy-based scheduling. In *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC '14*, pages 117–126, Washington, DC, USA. IEEE Computer Society.
- [Vieira et al., 2015] Vieira, C. C. A., Bittencourt, L. F., and Madeira, E. R. M. (2015). A scheduling strategy based on redundancy of service requests on iaas providers. *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pages 497–504.
- [Wang et al., 2014] Wang, T., Liu, Z., Chen, Y., Xu, Y., and Dai, X. (2014). Load Balancing Task Scheduling Based On Genetic Algorithm In Cloud Computing. In *Dependable Autonomous And Secure Computing (Dasc) 2014 IEEE 12th International Conference On*, pages 146–152, Dalian, China. IEEE.
- [Wei et al., 2010] Wei, G., Vasilakos, A. V., Zheng, Y., and Xiong, N. (2010). A game-theoretic method of fair resource allocation for cloud computing services. *J. Supercomput.*, 54(2):252–269.
- [Welch et al., 1998] Welch, L. R., Shirazi, B. A., Ravindran, B., and Bruggeman, C. (1998). Desiderata: Qos management technology for dynamic, scalable, dependable, real-time systems. *IFAC Proceedings Volumes*, 31(32):7 – 12. 15th IFAC Workshop on Distributed Computer Control Systems (DCCS'98), Como, Italy, 9-11 September 1998.

- [Wimalajeewa and Jayaweera, 2008] Wimalajeewa, T. and Jayaweera, S. K. (2008). Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained pso. *IEEE Transactions on Wireless Communications*, 7(9):3608–3618.
- [Worasuchep, 2008] Worasuchep, C. (2008). Solving constrained engineering optimization problems by the constrained pso-dd. In *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 1, pages 5–8.
- [Wu et al., 2012] Wu, Y., Wu, C., Li, B., Zhang, L., Li, Z., and Lau, F. C. M. (2012). Scaling social media applications into geo-distributed clouds. In *2012 Proceedings IEEE INFOCOM*, pages 684–692.
- [Yang et al., 2007] Yang, X., Yuan, J., Yuan, J., and Mao, H. (2007). A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation*, 189(2):1205–1213.
- [Ye et al., 2017] Ye, W., Feng, W., and Fan, S. (2017). A novel multi-swarm particle swarm optimization with dynamic learning strategy. *Applied Soft Computing*, 61:832 – 843.
- [Ye et al., 2012] Ye, Z., Bouguettaya, A., and Zhou, X. (2012). Qos-aware cloud service composition based on economic models. In *International Conference on Service-Oriented Computing*, pages 111–126. Springer.
- [Zhang et al., 2004] Zhang, H., Keahey, K., and Allcock, W. (2004). Providing data transfer with qos as agreement-based service. In *Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on*, pages 344–353. IEEE.
- [Zhang et al., 2008] Zhang, L., Chen, Y., Sun, R., Jing, S., and Yang, B. (2008). A task scheduling algorithm based on pso for grid computing. 4:37–43.
- [Zhang et al., 2011] Zhang, Q., Zhu, Q., and Boutaba, R. (2011). Dynamic resource allocation for spot markets in cloud computing environments. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, pages 178–185.
- [Zhang et al., 2015] Zhang, Y.-D., Wang, S., and ji, G. (2015). A comprehensive survey on particle swarm optimization algorithm and its applications. 2015:1–38.
- [Zhang et al., 2014] Zhang, Z., Xi, H., and Song, C. (2014). Dynamic optimal resource provisioning for vod services under amazon ec2’s pricing models. pages 5527–5532.
- [Zhang-Jian, 2013] Zhang-Jian, R.-H. H. . C.-N. L. . Y.-R. C. . D.-J. (2013). Cost optimization of elasticity cloud resource subscription policy. 7:561 – 574.
- [Zhao et al., 2014] Zhao, Y., Jiang, H., Zhou, K., Huang, Z., and Huang, P. (2014). Meeting service level agreement cost-effectively for video-on-demand applications in the cloud. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 298–306.

- [Zheng et al., 2013a] Zheng, Z., Wu, X., Zhang, Y., Lyu, M. R., and Wang, J. (2013a). Qos ranking prediction for cloud services. *IEEE transactions on parallel and distributed systems*, 24(6):1213–1222.
- [Zheng et al., 2013b] Zheng, Z., Wu, X., Zhang, Y., Lyu, M. R., and Wang, J. (2013b). Qos ranking prediction for cloud services. *IEEE Transactions on Parallel and Distributed Systems*, 24(6):1213–1222.
- [Zhu et al., 2013] Zhu, Z., Li, S., and Chen, X. (2013). Design qos-aware multi-path provisioning strategies for efficient cloud-assisted svc video streaming to heterogeneous clients. *IEEE Transactions on Multimedia*, 15(4):758–768.



# APPENDIX A

## COST OF CLOUD WEB SERVICES

### A.1 The Cost of the Amazon Web Services

In this part, the pricing information for the cloud services used during the case study is given.

Table A.1: Cost and Type of Amazon Simple Storage Services (S3)

Region	Type	Cost (First 50 TB / month)
Region	Type	Cost (First 50 TB / month)
US East (N. Virginia)	SS <sup>1</sup>	\$0.023
	SIAS <sup>2</sup>	\$0.0125
	GS <sup>3</sup>	\$0.004
US East (Ohio)	SS	\$0.023
	SIAS	\$0.0125
	GS	\$0.004
US West (Northern California)	SS	\$0.026
	SIAS	\$0.019
	GS	\$0.005
US West (Oregon)	SS	\$0.023
	SIAS	\$0.0125
	GS	\$0.004
Asia Pacific (Mumbai)	SS	\$0.025
	SIAS	\$0.019
	GS	\$0.005
Asia Pacific (Seoul)	SS	\$0.025
	SIAS	\$0.018
	GS	\$0.005
Asia Pacific (Singapore)	SS	\$0.025
	SIAS	\$0.02
	GS	N/A
Asia Pacific (Sydney)	SS	\$0.025
	SIAS	\$0.019
	GS	\$0.005
Asia Pacific (Tokyo)	SS	\$0.025

---

<sup>1</sup> SS = Standard Storage

<sup>2</sup> SIAS = Standard- Infrequent Access Storage

<sup>3</sup> GS = Glacier Storage

Table A.1: Cost and Type of Amazon Simple Storage Services (S3)

Region	Type	Cost (First 50 TB / month)
	SIAS	\$0.019
	GS	\$0.005
Canada (Central)	SS	\$0.025
	SIAS	\$0.0138
	GS	\$0.0045
EU (Frankfurt)	SS	\$0.0245
	SIAS	\$0.0135
	GS	\$0.0045
EU (Ireland)	SS	\$0.023
	SIAS	\$0.0125
	GS	\$0.004
EU (London)	SS	\$0.024
	SIAS	\$0.0131
	GS	\$0.0045
South America (Sao Paulo)	SS	\$0.0405
	SIAS	\$0.026
	GS	N/A

Table A.2: Amazon Simple Storage Service(S3) Transfer Pricing

Region From	Data Transfer OUT From Amazon S3 To	Cost per GB (First 50 TB / month)
US East (N. Virginia)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
US East (Ohio)	Another AWS Region	\$0.020
	US East(Ohio)	\$0.010
	Internet(to 10TB/month)	\$0.090
US West (Northern California)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
US West (Oregon)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.010
	Internet(to 10TB/month)	\$0.090
Asia Pacific (Mumbai)	Another AWS Region	\$0.086
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.1093
Asia Pacific (Seoul)	Another AWS Region	\$0.080
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.126
Asia Pacific (Singapore)	Another AWS Region	\$0.090
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.120
Asia Pacific (Sydney)	Another AWS Region	\$0.140

Table A.2: Amazon Simple Storage Service(S3) Transfer Pricing

<b>Region From</b>	<b>Data Transfer OUT From Amazon S3 To</b>	<b>Cost per GB (First 50 TB / month)</b>
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.140
Asia Pacific (Tokyo)	Another AWS Region	\$0.090
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.140
Canada (Central)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
EU (Frankfurt)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
EU (Ireland)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
EU (London)	Another AWS Region	\$0.020
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.090
South America (Sao Paulo)	Another AWS Region	\$0.160
	Amazon CloudFront	\$0.000
	Internet(to 10TB/month)	\$0.250

Table A.3: Amazon Elastic Transcoder Transcoding Pricing

<b>Region</b>	<b>Video Type</b>	<b>Cost per minute</b>
US East (N. Virginia)	Standard Definition (Resolution of less than 720p)	\$0.015
	High Definition (Resolution of 720p or above)	\$0.030
US West (Northern California)	Standard Definition (Resolution of less than 720p)	\$0.017
	High Definition (Resolution of 720p or above)	\$0.034
Asia Pacific (Mumbai)	Standard Definition (Resolution of less than 720p)	\$0.015
	High Definition (Resolution of 720p or above)	\$0.030
Asia Pacific (Singapore)	Standard Definition (Resolution of less than 720p)	\$0.017
	High Definition (Resolution of 720p or above)	\$0.034
Asia Pacific (Sydney)	Standard Definition (Resolution of less than 720p)	\$0.017

Table A.3: Amazon Elastic Transcoder Transcoding Pricing

Region	Video Type	Cost per minute
	High Definition (Resolution of 720p or above)	\$0.034
Asia Pacific (Tokyo)	Standard Definition (Resolution of less than 720p)	\$0.017
	High Definition (Resolution of 720p or above)	\$0.034
EU (Ireland)	Standard Definition (Resolution of less than 720p)	\$0.017
	High Definition (Resolution of 720p or above)	\$0.034

Table A.4: Streaming from CDN Transfer Pricing

Region From	Data Transfer OUT From CDN To	Cost per GB
United States	Origin	\$0.020
	Internet(to 10TB/month)	\$0.090
Canada	Origin	\$0.020
	Internet(to 10TB/month)	\$0.09
Europe	Origin	\$0.020
	Internet(to 10TB/month)	\$0.090
Hong Kong, Philippines, S. Korea, Singapore and Taiwan	Origin	\$0.060
	Internet(to 10TB/month)	\$0.140
Japan	Origin	\$0.060
	Internet(to 10TB/month)	\$0.140
South America	Origin	\$0.130
	Internet(to 10TB/month)	\$0.250
Australia	Origin	\$0.100
	Internet(to 10TB/month)	\$0.140
India	Origin	\$0.160
	Internet(to 10TB/month)	\$0.170

Table A.5: Streaming over Secured HTTP Transfer Pricing

Region From	Request Type	Cost per 10.000 requests
United States	HTTP request	\$0.0075
	HTTP/S request	\$0.0100
Canada	HTTP request	\$0.0075
	HTTP/S request	\$0.010
Europe	HTTP request	\$0.0075
	HTTP/S request	\$0.010
Hong Kong, Philippines, S. Korea, Singapore and Taiwan	HTTP request	\$0.009
	HTTP/S request	\$0.012



Table A.5: Streaming over Secured HTTP Transfer Pricing

Region From	Request Type	Cost per 10.000 requests
Japan	HTTP request	\$0.009
	HTTP/S request	\$0.012
South America	HTTP request	\$0.0160
	HTTP/S request	\$0.0220
Australia	HTTP request	\$0.0090
	HTTP/S request	\$0.0125
India	HTTP request	\$0.0090
	HTTP/S request	\$0.0120

During the implementation, since Amazon Cloud Front does not give customers to select zone in a region level, price classes are defined. So, the price for the case study used for Cloud Front is taken in a price class level.

Table A.6: Amazon CDN(Cloud Front) Transfer Pricing

Price Class	Data Transfer OUT From CDN To	Cost per GB
Price Class All	Regional Data Transfer Out to Internet	\$0.250
	Regional Data Transfer Out to Origin	\$0.160
Price Class 200)	Regional Data Transfer Out to Internet	\$0.170
	Regional Data Transfer Out to Origin	\$0.160
Price Class 100	Regional Data Transfer Out to Internet	\$0.085
	Regional Data Transfer Out to Origin	\$0.020

Table A.7: Request Pricing for All HTTP- HTTPS Methods (per 10.000 requests)

Price Class	Request Type	Cost per 10.000 requests
Price Class All	HTTP requests	\$0.160
	HTTPS requests	\$0.0220
Price Class 200	HTTP requests	\$0.090
	HTTPS requests	\$0.0120
Price Class 100	HTTP requests	\$0.0075
	HTTPS requests	\$0.0100

## A.2 The Cost of the Microsoft Azure Services

In this section, the cost of Microsoft Azure Cloud solutions are given.

Table A.8: Cost and Region of Azure Media Services for transcoding HD and SD videos

<b>Zone</b>	<b>Region</b>	<b>SD Transcoding Cost per min</b>	<b>HD Transcoding Cost per min</b>
United States	Central US	\$0.015	\$0.030
	East US	\$0.015	\$0.030
	East US 2	NA	NA
	South Central US	\$0.015	\$0.030
	North Central US	NA	NA
	West US	\$0.015	\$0.030
	West US 2	NA	NA
	West Central US	NA	NA
Europe	North Europe	\$0.015	\$0.030
	West Europe	\$0.015	\$0.030
Asia Pacific	East Asia	\$0.017	\$0.034
	Southeast Asia	\$0.017	\$0.034
Japan	Japan East	\$0.017	\$0.034
	Japan West	\$0.015	\$0.030
Brazil	Brazil South	\$0.017	\$0.034
Australia	Australia Central	NA	NA
	Australia Central 2	NA	NA
	Australia East	\$0.017	\$0.034
	Australia South-East	\$0.017	\$0.034
India	Central India	\$0.015	\$0.030
	South India	\$0.015	\$0.030
	West India	\$0.015	\$0.030
Canada	Canada Central	\$0.015	\$0.030
	Canada East	\$0.015	\$0.030
Azure Germany	Germany Central	NA	NA
	Germany North-East	NA	NA
France	France Central	NA	NA
	France South	NA	NA
Korea	Korea Central	NA	NA
	Korea South	NA	NA
Azure Government	US Gov Arizone	NA	NA
	US Gov Iowa	\$0.019	\$0.038
	US Gov Texas	NA	NA
	US Gov Virginia	\$0.019	\$0.038

Table A.9: Amazon CDN(Cloud Front) Transfer Pricing

<b>Zone</b>	<b>Data Transfer OUT From CDN To</b>	<b>Cost per GB</b>
North America, Europe, Middle East and Africa	Regional Data Transfer Out to Internet	\$0.087
	Regional Data Transfer Out to Origin	\$0
Asia Pacific (including Japan)	Regional Data Transfer Out to Internet	\$0.138
	Regional Data Transfer Out to Origin	\$0
South America	Regional Data Transfer Out to Internet	\$0.25
	Regional Data Transfer Out to Origin	\$0
Australia	Regional Data Transfer Out to Internet	\$0.14
	Regional Data Transfer Out to Origin	\$0
India	Regional Data Transfer Out to Internet	\$0.17
	Regional Data Transfer Out to Origin	\$0

Table A.10: Cost and Region of Azure Media Services for transcoding HD and SD videos

<b>Zone</b>	<b>Region</b>	<b>Transfer Cost from Storage</b>	<b>Encryption Cost per 10.000 requests</b>
United States	Central US	\$0.087	\$0.030
	East US	\$0.087	\$0.030
	East US 2	\$0.087	\$0.030
	South Central US	\$0.087	\$0.030
	North Central US	\$0.087	\$0.030
	West US	\$0.087	\$0.030
	West US 2	\$0.087	\$0.030
	West Central US	\$0.087	\$0.030
Europe	North Europe	\$0.087	\$0.030
	West Europe	\$0.087	\$0.030
Asia Pacific	East Asia	\$0.12	\$0.030
	Southeast Asia	\$0.12	\$0.030
Japan	Japan East	\$0.12	\$0.030
	Japan West	\$0.12	\$0.030
Brazil	Brazil South	\$0.181	\$0.030
Australia	Australia Central	NA	NA
	Australia Central 2	NA	NA
	Australia East	\$0.12	\$0.30
	Australia South-east	\$0.12	\$0.030
India	Central India	\$0.12	\$0.030
	South India	\$0.12	\$0.030
	West India	\$0.12	\$0.030
Canada	Canada Central	\$0.087	\$0.030
	Canada East	\$0.087	\$0.030

Table A.10: Cost and Region of Azure Media Services for transcoding HD and SD videos

<b>Zone</b>	<b>Region</b>	<b>Transfer Cost from Storage</b>	<b>Encryption Cost per 10.000 requests</b>
Azure Germany	Germany Central	\$0.10	\$0.030
	Germany North-east	\$0.10	\$0.030
United Kingdom	France Central	\$0.087	\$0.030
	France South	\$0.087	\$0.030
France	France Central	\$0.087	\$0.030
	France South	\$0.087	\$0.030
Korea	Korea Central	\$0.12	NA
	Korea South	\$0.12	NA
Azure Government	US Gov Arizone	\$0.109	\$0.030
	US Gov Iowa	\$0.109	\$0.030
	US Gov Texas	\$0.109	\$0.030
	US Gov Virginia	\$0.109	\$0.030

# CURRICULUM VITAE

## PERSONAL INFORMATION

**Surname, Name:** Aygün, Betül  
**Nationality:** Turkish (TC)  
**Date and Place of Birth:** 30.07.1983, Ankara  
**Marital Status:** Married

## EDUCATION

Degree	Institution	Year of Graduation
M.S.	M.S. Informatics Institute Information Systems	2007 - 2010
B.S.	METU – Mathematics BS	2001 - 2006
High School	Ankara Kurtulus High School	1997 - 2001

## PROFESSIONAL EXPERIENCE

- 02.2015 – cont. Public Procurement Authority, Department of Electronic Procurement Software Engineer - Software Developer(C#) - Business Intelligence Specialist - Web Service Developer
- 02.2009 - 02.2015 Middle East Technical University- Information Systems (METU) Research Assistant
- 08.2007 - 02.2009 Turkish Arm Forces -JGNK Java Developer
- 06.2006 – 11.2006 SEBIT – Education and Information Technologies Inc. Educational Designer
- 01.2006 – 06.2006 Middle East Technical University - Mathematics (METU) Student Research Assistant – Linear Algebra

## PUBLICATIONS

### International Conference Publications

- Aygün, B., Yılal, E., Arifoğlu, A. (2011). ‘IT Process Models: Unification of IT Process Models into A Simple Framework Supplemented by Turkish Web Based Application’. Saarbrücken: LAP LAMBERT Academic Publishing
- Aygün, B., Arıcı, N., Coşar, A. (2017). ‘Optimal Resource Allocation of Dynamic Video Streaming Applications in a Public Cloud Environment’, in International Conference on Com-

puter Science and Engineering. Heads of Departments of Computer Engineering Council, Antalya, Turkey. 2017

- Aygün, B., Coşar, A. (2018). “Particle Swarm Optimization with Levenberg Marquardt Algorithm for Artificial Neural Network Training”, in 8th International Conference on Bioinspired Optimization Methods and their Applications. Paris, France. 2018.

- Aygün, B., Coşar, A. (Submitted). “Mathematical Modelling and Optimization Solution of Public Cloud Resources Allocation System for Video on Demand (VoD) Applications”, Applied Soft Computing.

