

An Optimization Framework Using Multiple Economic Models in Grid Computing: A Switching Mechanism

by

S M Aminul Haque

Submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

February 2012

School of Information Technology

MONASH UNIVERSITY

Copyright Notices

Notice 1

Under the Copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Notice 2

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Table of Contents

Abstract.....	v
Declaration	vii
Acknowledgements	viii
Figures	x
Tables	xiii
Chapter 1 Introduction	1
1.1 Background.....	1
1.1.1 Grid Computing.....	1
1.1.2 Economic Models in Grid Resource Management	3
1.2 Scope of the Thesis	5
1.3 Problem Description	5
1.3.1 Research Hypothesis.....	5
1.3.2 Research Questions	5
1.3.3 Proposed Solutions	6
1.4 Research Contributions	7
1.5 Thesis Organization.....	9
1.6 Publication Record	11
Chapter 2 Economic-based Resource Management in Grid Computing	13
2.1 Introduction.....	13
2.2 Grid Computing: A Transition from Mainframe to Meta-computing.....	14
2.3 A Layered Grid Architecture	16
2.4 Inspiration for Economic Grids.....	18
2.5 Market-oriented Grid Computing	20
2.6 Criteria to Judge Economic Grids.....	22
2.7 Economic Models and their Strengths and Weaknesses in Grid Computing ..	24
2.8 Discussion.....	40
2.9 Conclusions	44
Chapter 3 Towards Developing an Evaluation Framework for Economic-based Grid Resource Management	47
3.1 Introduction.....	47
3.2 Tools for Simulating Grid Environment.....	48
3.3 GridSim: A Discrete Event Simulation Toolkit	51
3.3.1 Applications and Resources.....	51
3.3.2 Communication Model	53
3.4 Properties of Economically-inspired Grid Entities	54
3.4.1 User	55

3.4.2	Broker.....	55
3.4.3	Resource.....	56
3.5	Resource Configuration and Simulation Methodology.....	56
3.5.1	Grid Scenarios and Simulation Space.....	56
3.5.2	Simulation Parameters and Its Statistical Significance.....	59
3.5.3	Methodology for Data Analysis	60
3.5.4	Fairness in Evaluation.....	61
3.6	Simulation using Monash Sun Grid.....	62
3.7	Conclusions	65
Chapter 4	Grid Resource Management Using Commodity Markets.....	67
4.1	Introduction	67
4.2	Resource Management Using Commodity Market Model.....	68
4.2.1	Concurrent Market	71
4.2.2	Market Equilibrium	72
4.3	Output Analysis for Commodity Market Model.....	74
4.4	Resource Management Using Bargaining Model.....	82
4.5	Output Analysis for Bargaining Model	86
4.6	A Comparative Analysis among the Commodity Models	94
4.7	Summary and Conclusions.....	99
Chapter 5	Grid Resource Management Using Auctions	101
5.1	Introduction	101
5.2	Auctions: A Historical Background	103
5.3	A Framework for Grid Resource Management Using English Auction.....	104
5.3.1	Understanding Different Auction Parties.....	105
5.3.2	Design and Development of English Auction	106
5.3.3	Concurrent Auction Procedure.....	108
5.4	Performance Analysis for Grid Resource Management using English Auction Protocol	110
5.5	Grid Resource Management Using Double Auction Protocol.....	119
5.6	Performance Analysis for Continuous Double Auction Protocol	122
5.7	Contract Net Protocol Based Grid Resource Management.....	131
5.7.1	Meta-scheduling Process	133
5.7.2	An Example of Resource Negotiation in Contract Net Protocol	134
5.8	Performance Analysis for Contract Net Protocol Based Grid Resource Management	135
5.9	A Comparative Analysis of the Auction Protocols	143
5.10	Summary and Conclusion.....	147
Chapter 6	Identification of Domains of Strengths of Different Economic Models in Grid Computing.....	149
6.1	Introduction	149
6.2	Inspiration for Domains of Strengths Specification	150
6.3	Revenue Analysis	151

6.4	Communication Efficiency Analysis.....	155
6.5	Success Rate Analysis.....	160
6.6	Analysis for Average Turn-around Time per Gridlet.....	164
6.7	Total Simulation Time Analysis.....	167
6.8	Resource Utilization Analysis.....	170
6.9	User Utility Analysis	171
6.10	Resource Utility Analysis	173
6.11	Social Welfare Analysis	174
6.12	Discussion and Research Proposal	177
6.13	Conclusion	179
Chapter 7	A Switching Framework for Optimization in Economic Grids	181
7.1	Introduction.....	181
7.2	Inspiration for Optimization and Switching Framework	182
7.3	A Prototype for Switching Framework	184
7.4	Towards Developing a Protocol-generic Switching Framework in Grid.....	186
7.4.1	User, Broker and Resource Models.....	186
7.4.2	An Autonomous Switching Framework.....	190
7.4.3	Switching Agent.....	192
7.5	Performance Analysis for Switching Framework.....	195
7.5.1	Optimization for Revenue over Communication Overhead	197
7.5.2	Optimization for Communication Efficiency	201
7.5.3	Optimization for Average Turn-around Time per Gridlet	204
7.5.4	Optimization for Social Welfare	208
7.6	Switching Mechanism in Cloud Computing: Challenges and Opportunities ..	210
7.7	Conclusions	212
Chapter 8	Conclusions and Future Directions	215
8.1	Summary	215
8.2	Research Limitations	216
8.3	Conclusions	217
8.4	Future Directions.....	220
8.4.1	Supporting Multiple Application Types	220
8.4.2	Measuring Real-time Adaptability	220
8.4.3	Supporting Multi-criteria Optimization and Improving the Switching Agent.....	221
8.4.4	Supporting Cloud Computing Infrastructure.....	221
	Bibliography	223

An Optimization Framework Using Multiple Economic Models in Grid Computing: A Switching Mechanism

Doctoral candidate: S M Aminul Haque

Doctoral Advisors: Dr Saadat M Alhashmi and Dr Rajendran Parthiban

School of Information Technology, Monash University
Sunway Campus, Malaysia

Abstract

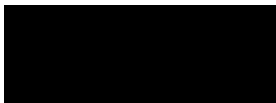
Grid computing solves computationally complex problems such as climate modelling in a cost effective and standardized way. It requires seamless collaboration of computational resources distributed across different administrative domains worldwide. However, distributed ownerships and heterogeneous (independent) nature of these resources impose a challenge to this collaboration. Since economic-based resource management approaches have been found efficient and sustainable for various distributed computing platforms such as Grid, significant efforts are being made to evaluate the effectiveness of various economic models for distributed resource management. Several economic models have been proposed for Grid computing based on both micro and macro-economic principles. In spite of the potential of economic-based resource management, there is no consensus on choosing a particular economic model for the Grid as different researchers have proposed different models in different times. Therefore, a comprehensive understanding about various economic models in the context of Grid computing is essential to discover the problem of choosing a common model.

The primary contribution of this thesis in identifying this problem is the process of a survey on existing economic models in the Grid. The survey identifies that one model is different from another in terms of pricing methodology and working principle. Moreover, the survey claims the suitability of different models for different scenarios. For example, Bargaining economic model supports utility-based negotiation between a resource user and a resource provider (microeconomic), whereas Commodity Market Model is suitable for maintaining equilibrium between supply and demand of resources in the environment (macro-economic). The major reason to this problem is the limitation of a single model to cope with large-scale dynamic characteristic of the Grid. This limitation demonstrates a need to analyze the effectiveness of different economic models in Grid resource management. Therefore, this thesis conducts an extensive experimental analysis on the five most widely proposed economic models in the Grid – Commodity Market, Bargaining, English Auction, Continuous Double Auction and Contract Net Protocol. The experimental results demonstrate the compatibility with existing literature that a single economic model is not suitable for all circumstances in a Grid's lifecycle. A quantitative analysis on the performances of different economic models helps identify the regions (domains) where one model outperforms all the other models in different scenarios. This variation in performances shows the opportunity of designing an optimization framework through utilization of the potentials of different models in different scenarios based on the domains of strengths of the models. To facilitate the optimization process, an adaptive switching mechanism that dynamically switches from one economic model to another depending on the function needed to be optimized, has been developed. The roles and responsibilities of the Grid entities to adapt with changing scenarios (one model to another model) in a dynamic environment have been justified and presented. The thesis further provides formal definitions to these domains of strengths of individual models to

ensure that the switching decision can be carried out without much delay and computational power. The effectiveness of the switching framework in distributed resource management has been evaluated through a series of experiments. The results of these experiments show that the switching model can bring promising outcomes in collaborating distributed resources in an economic Grid.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other institution and to the best of my knowledge, the thesis contains no material previously published or written by another person, except where due reference is made in the text of thesis. I further affirm that the thesis contains less than 100,000 words, exclusive of tables, figures, footnotes and bibliographies.



S M Aminul Haque

February 2012

Acknowledgements

This thesis is dedicated to those who have explicitly or implicitly contributed to the successful completion of the thesis.

First of all, I am grateful to my supervisors – Dr Saadat M Alhashmi and Dr Rajendran Parthiban, for giving me the opportunity to work under their guidance. Their systematic guidance and constructive comments have immensely helped to improve my performance over time. Their continuous support and inspiration further helped me to deal with new challenges successfully. In particular, I would like to mention their passion that significantly motivated me to put tireless effort into research.

I would like to thank Robert Gray and Dr Jefferson Tan (member, Centre for Distributed Systems and Software Engineering, Monash, Australia), for their kind cooperation in introducing with their Grid Computing facilities while my stay in Melbourne for a week.

My sincere thanks to Professor Chris Messom (Academic Deputy President, Monash, Malaysia) for giving comments and suggestion on my confirmation report. I admire the help of Dr. Karen Jennifer Golden and Dr. Miriam Sang-Ah Park (Lecturer, School of Medicine, Monash, Malaysia), who have been kind enough to provide comments on some of my work. I am also grateful to thank numerous reviewers from international journals and conferences for giving valuable feedback towards shaping my work. I thank to those who have posed questions and comments on my work from during my presentations in various conferences and seminars.

I would like to express my sincere thanks to Blair Bethwaite, Philip Chan, Shahaan Ayyub, and Slavisa Garic (member, Monash eScience and Grid Engineering Laboratory, Australia), for helping to run my experiment on Monash Sun Grid. Their prompt response and sincerity appreciably helped to accomplish my tasks successfully and in a due manner.

I thank to Monash University, for giving me the scholarship for conducting PhD and an excellent environment, which is filled with ample resources. I also thank the university, for giving me grants to travel to various countries for attending conferences and workshops. I appreciate the help and support I have received from all academic, administrative and technical staff members of the School of Information Technology, Monash, Malaysia. Without their cooperation and advice, it would have been hard to complete my work in due time.

I would like to take this opportunity to thank Monash University Postgraduate Association, Malaysia, for their events and gathering, which have provided the opportunity to exchange our ideas among people from different fields of research. In addition, their outdoor and sporting activities have truly made life fun while working.

I appreciate the inspiration and support from my colleagues. Their talents and experience inspired me to remain focused on achieving my goal. Overall, due to their sincerity and love, I have been able to work in a familylike environment. I thank Radi Jarrar, for proofreading a part of the thesis. My thanks to Abdul Khalique Shaikh. By being in the same area, I have been able to discuss and share many topics in the field with him.

Last but not least, I am thankful to my family, my parents, my wife, brothers, and sisters, for their love, patience, and understanding. My specific thanks to my elder brother and sister-in-law, for their inspiration and guidance while my stay in Malaysia. Above all, I am indebted

to the Almighty, for all His supports and wishes that have made it possible to complete the thesis successfully.

Figures

Figure 1.1: A High Level Overview of Grid Computing	2
Figure 1.2: Grid Resource Management in a Dynamic and Heterogeneous Environment	2
Figure 1.3: Organization of Thesis Chapters.....	10
Figure 2.1: Major Technological Shifts in Distributed Computing	15
Figure 2.2: Layered Grid Architecture with Examples.....	17
Figure 2.3: Conceptual View of an Economic Grid.....	20
Figure 2.4: A reference Market Oriented Overview of a Layered Grid Architecture (Neumann et al., 2008).....	21
Figure 2.5: Adoption of Economic Models as Per Year	42
Figure 2.6: Major Economic Models in Grid.....	43
Figure 3.1: A Typical Overview of GridSim Simulator – From a Developer’s Point of View	51
Figure 3.2: A Conceptual View of Communication between Two GridSim Entities	53
Figure 3.3: A Broker’s Life-cycle on Gridlet Execution	55
Figure 3.4: Simulation Space designed for Our Framework	57
Figure 3.5: Application Execution Environment in a Real Grid.....	58
Figure 3.6: Application Execution Environment in Our Simulated Grid	58
Figure 3.7: A Holistic Diagram of the Evaluation Model	62
Figure 3.8: Monash Sun Grid: A Layered Overview	62
Figure 3.9: Major Records Related to an Application in MSG.....	64
Figure 4.1: An Event Diagram for the Interaction between Different Entities in CMM	69
Figure 4.2: Change in Equilibrium Price, P due to Supply and Demand Shift	72
Figure 4.3: The Effect of Spot and Random Prices over Supply and Demand.....	74
Figure 4.4: Revenue Distribution for CMM	75
Figure 4.5: Distribution for Communication Overhead.....	76
Figure 4.6: Distribution for Success Rate	77
Figure 4.7: Distribution for Average Turn-around Time per Gridlet.....	78
Figure 4.8: Total Simulation Time Distribution.....	79
Figure 4.9: Distribution for Resource Utilization.....	80
Figure 4.10: Distribution for User Utility.....	81
Figure 4.11: User’s Strategy for Bargaining Model (θc = current number of rounds)	84
Figure 4.12: Resource’s Strategy for Bargaining Model (θc = current number of rounds)	85
Figure 4.13: Revenue Distribution	86
Figure 4.14: Distribution for Communication Overhead	87
Figure 4.15: Distribution for Success Rate	88
Figure 4.16: Distribution for Average Turn-around Time per Gridlet	89
Figure 4.17: Total Simulation Time Distribution	90
Figure 4.18: Distribution for Resource Utilization	91
Figure 4.19: Distribution for User Utility.....	92
Figure 4.20: Resource Utility Distribution	93
Figure 4.21: Distribution for Social Welfare.....	94
Figure 4.22: Comparison for Revenue over Communication Overhead	95

Figure 4.23: Comparison for Success Rate	97
Figure 4.24: Comparison for Total Simulation Time.....	98
Figure 5.1: Sotheby's Book Auction, London, 1888	103
Figure 5.2: An Event Diagram in the English Auction Framework.....	107
Figure 5.3: Group Formation and Independent Auction Process by the Auctioneer	109
Figure 5.4: A Competitive Bidding History among Different Brokers in English Auction	110
Figure 5.5: Revenue Distribution.....	111
Figure 5.6: Distribution for Communication Overhead	112
Figure 5.7: Distribution for Success Rate	113
Figure 5.8: Distribution for Average Turn-around Time per Gridlet	114
Figure 5.9: Total Simulation Time Distribution	115
Figure 5.10: Distribution for Resource Utilization.....	116
Figure 5.11: Distribution for User Utility	117
Figure 5.12: Resource Utility Distribution	118
Figure 5.13: Distribution for Social Welfare	119
Figure 5.14: Sample Order Books for Grid Users and Resources in Continuous Double Auction	122
Figure 5.15: Revenue Distribution	123
Figure 5.16: Distribution for Communication Overhead.....	124
Figure 5.17: Distribution for Success Rate.....	125
Figure 5.18: Distribution for Average Turn-around Time per Gridlet.....	126
Figure 5.19: Total Simulation Time Distribution	127
Figure 5.20: Distribution for Resource Utilization	128
Figure 5.21: Distribution for User Utility	129
Figure 5.22: Resource Utility Distribution.....	130
Figure 5.23: Distribution for Social Welfare.....	131
Figure 5.24: Structure of a cfp-template and Behavior of a Contract Node	133
Figure 5.25: An Event Diagram for Resource Negotiation in Contract Net Protocol	134
Figure 5.26: Revenue Distribution	135
Figure 5.27: Distribution for Communication Overhead	137
Figure 5.28: Distribution for Success Rate	138
Figure 5.29: Distribution for Average Turn-around Time per Gridlet	139
Figure 5.30: Distribution for Total Simulation Time.....	140
Figure 5.31: Distribution for Resource Utilization.....	141
Figure 5.32: Distribution for User Utility.....	142
Figure 5.33: Comparison for Revenue.....	143
Figure 5.34: Comparison for Communication Overhead	144
Figure 5.35: Comparison for Success Rate	145
Figure 5.36: Social Welfare Comparison	146
Figure 6.1: Revenue Comparison	151
Figure 6.2: Domain Formalization for Revenue over Communication Cost	154
Figure 6.3: Communication Efficiency Comparison.....	155
Figure 6.4: Domain Formalization for Communication Efficiency-1	156
Figure 6.5: Domain Formalization for Communication Efficiency-2	158
Figure 6.6: Success Rate Comparison.....	160
Figure 6.7: Domain Formalization for Success Rate	162
Figure 6.8: Comparison for Average Turn-around Time per Gridlet	164
Figure 6.9: Domain Formalization for Average Turn-around Time per Gridlet.....	165
Figure 6.10: Total Simulation Time Comparison	168
Figure 6.11: Domain Formalization for Total Simulation Time	168
Figure 6.12: Comparison for Resource Utilization.....	170

Figure 6.13: User Utility Comparison	171
Figure 6.14: Resource Utility Comparison.....	172
Figure 6.15: Resource Utility over Communication Overhead Comparison.....	172
Figure 6.16: Domain Formalization for Resource Utility over Communication Overhead	173
Figure 6.17: Social Welfare Comparison	175
Figure 6.18: Domain Formalization for Social Welfare	176
Figure 7.1: Agent-driven Switching Framework for Grid Resource Management .	184
Figure 7.2: The User Defines QoS Parameters for Switching Framework	186
Figure 7.3: Scenario Illustrated for Revenue over Communication Overhead Optimization	197
Figure 7.4: Average Turn-around Time per Gridlet while Optimizing Revenue over Communication Overhead.....	199
Figure 7.5: Social Welfare while Optimizing Revenue over Communication Overhead	200
Figure 7.6: Scenario Illustrated for Optimizing Communication Efficiency.....	201
Figure 7.7: Average Turn-around Time per Gridlet while Minimizing Communication Overhead.....	202
Figure 7.8: Revenue over Communication Overhead while Minimizing Communication Overhead.....	203
Figure 7.9: Social Welfare while Minimizing Communication Overhead	204
Figure 7.10: Scenario Illustrated for Optimizing Average Turn-around Time per Gridlet	205
Figure 7.11: Revenue over Communication Overhead while Minimizing Average Turn-around Time per Gridlet	206
Figure 7.12: Social Welfare while Minimizing Average Turn-around Time per Gridlet	206
Figure 7.13: Scenario Illustrated for Optimizing Total Simulation Time	207
Figure 7.14: Scenario Illustrated for Social Welfare Optimization	208
Figure 7.15: Revenue over Communication Overhead while Optimizing Social Welfare	209
Figure 7.16: Average Turn-around Time per Gridlet while Optimizing Social Welfare	210

Tables

Table 2.1: Strengths of Economic Models in Grid Computing	25
Table 2.2: A Comparative View among Different Economic Models in Grid Computing.....	36
Table 3.1: A Comparative Approach among Different Grid Simulators	49
Table 3.2: Resource Configuration	59
Table 4.1: CMM versus BAR over Different Supply and Demand Regions	100
Table 5.1: The Strengths of Different Auction Models for Varied Supply and Demand	148
Table 6.1: Domains of Strengths of Economic Models in Grid Computing	177
Table 7.1: Different Learning Methods in Agent Technology	193
Table 7.2: Resource Configuration for Switching Model	196

Introduction

This chapter discusses the fundamentals of the thesis. It briefly describes the inspiration for Grid computing, economic-based resource management in the Grid and our work. The chapter further presents the scope of this thesis and summarizes the research methodology. The chapter ends with a detailed version of the thesis contributions and organization of the rest of the chapters of the thesis.

1.1 Background

To understand the key components of the thesis, this section briefly describes the Grid, resource management and the value of studying economic models in Grid computing.

1.1.1 Grid Computing

The Grid is a distributed computing platform. The root of distributed computing goes back in 1965 with the paper titled “*Solution of a problem in concurrent programming control*” (Dijkstra, 1965). According to one of the renowned computer scientists Laslie Lamport, “*This paper starts the field of concurrent and distributed algorithms*”, as stated in <http://www.podc.org/influential/2002.html>. The continuous advancement in science and technology delivers the necessary constituents such as E-mail, Ethernet, and Internet in paving the path of distributed resource sharing mechanisms. Consequently, this helps to bring about a large-scale, high performance, low cost, secured, and reliable computing platform, which, today, we call “Grid computing”. The Grid was officially introduced in mid-1990s (Foster and Kesselman, 2003). The major motivation for the Grid is driven by solving computationally complex scientific applications such as protein folding in a cost effective and standard way, which is not possible with a cluster of computers or even with a supercomputer.

The technology of Grid computing supports identification, negotiation, selection, and collaboration of geographically distributed heterogeneous computing resources over the Internet to facilitate the accomplishment of large-scale scientific applications. Figure 1.1 illustrates a conceptual view of Grid computing. The Grid is highly distinguished due to its resource characteristic; therefore, it is better start describing the Grid resources at first.

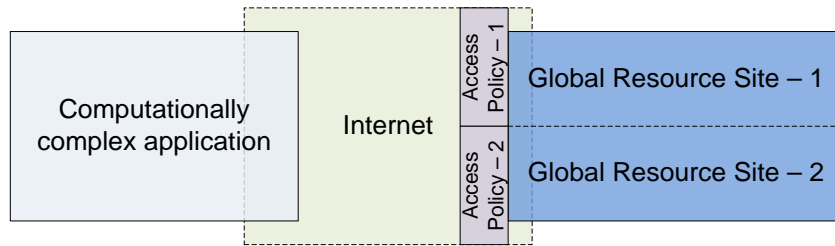


Figure 1.1: A High Level Overview of Grid Computing

The resources in a Grid are not under a centralized control, which means, the resource are highly heterogeneous and they run by their own administrative policies. The resources also vary in terms of architectures, Operating System, performance, and capability (computational resource, e.g. processing unit, storage or scientific instruments and so on). On the other hand, Grid computing possesses users/applications that require dynamic, scalable, seamless, and secure aggregation of these resources. Thus, the presentation of these highly heterogeneous resources to meet the requirements of the applications becomes a key challenge. Figure 1.2 demonstrates an overview of heterogeneous and dynamic resource management scenario in a Grid environment.

Therefore, resource management or scheduling becomes the most crucial issue in Grid computing (Buyya, 2002, Garg et al., 2009, Buyya et al., 2000a). The key role of Grid resource management architecture is to hide the heterogeneous nature of the Grid and deliver a reliable homogeneous environment to the end users. Thus, to enable a large-scale dynamic computing platform, a Grid resource management system must provide:

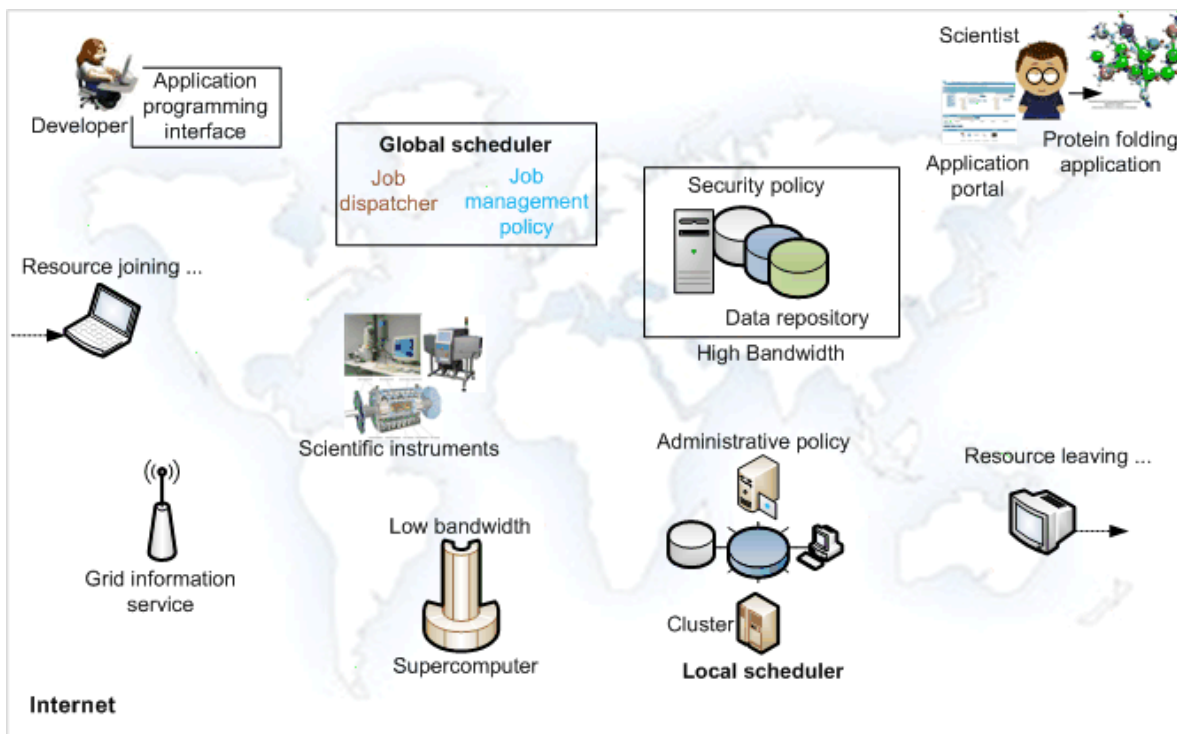


Figure 1.2: Grid Resource Management in a Dynamic and Heterogeneous Environment

- technologies to collaborate resources that are not under centralized control. To this collaboration, Grid schedulers/brokers are used. The schedulers are deployed to identify suitable resources, negotiate to understand resource usage policies, and facilitate the execution environment for the applications
- standards to communicate with resources. To enable a world-wide virtual computing platform the system should utilize necessary standards, negotiation models/ protocols, and interfaces
- technologies to understand the Quality of Service (QoS) parameters as defined in the application. Because, ultimately the Grid will be evaluated based on the value it delivers, not based on its architecture; therefore, it is crucial for any Grid model to be able to define a user's requirements
- components to define providers' interest. Multiple resource sites in Grid computing are typically federated towards meeting a common objective. This is also a key requirement for building a large-scale virtual computing platform; therefore, modeling the role and expected benefit by a provider in the environment must be analyzed beforehand

To meet the aforementioned challenges, various resource management techniques are proposed for the Grid including both economic and noneconomic based approaches. However, it has been identified that economic-based approaches are more suitable compared to noneconomic approaches for Grid resource management (Buyya et al., 2002, Mills and Dabrowski, 2008, Beck et al., 2008). Economic-based approaches not only provide the answer for distributed resource management problem but also provide sufficient motivation to the resource providers, for contributing their resources on the Grid; therefore, pricing becomes one of the key elements of any economic-based resource management system. It has been found that price can be used to understand the value of different resources successfully, to measure resource usage, to differentiate among QoS parameters, and to regulate system dynamics. The dynamics in Grid computing is typically referred to as the random variation of resource supply and demand in the environment. Depending on pricing methodologies and interaction protocols (between a user and a provider), various economic models are proposed for Grid computing. The following section summarizes the existing work on different economic models in Grid computing and presents the focus of this thesis.

1.1.2 Economic Models in Grid Resource Management

Economic models are significant in Grid resource management, because they can control and regulate the behavior of Grid entities (e.g. users and providers) through providing standards

related to pricing and interaction protocols; therefore, studying the effectiveness of different economic models in terms of heterogeneous and dynamic resource management/collaboration is crucial. Buyya proposes a set of economic models including Commodity Market, Bargaining, and Contract Net Protocol to deal with distributed resource collaboration (Buyya, 2002). Over the years, it can be observed that the value of economic-based approaches is increasing (Haque et al., 2011). The economic models that Buyya proposed were only based on the hypothetical suitability of the models. However not all of these models achieve similar popularity in the Grid (Haque et al., 2011). Over the past decade, only five economic models have widely been discussed – Commodity Market, Bargaining, English Auction, Continuous Double Auction and Contract Net Protocol. Therefore, this thesis focuses the study on these five economic models.

In spite of the potential of economic-based resource management in Grid computing until now, there is no general agreement on which economic model should be used for the Grid. The economic models vary from one another in terms of pricing methods and interaction protocols; therefore, the impact of different models on large-scale resource collaboration is likely to be varied. On the other hand, due to the highly dynamic nature of the Grid, the performance by a model changes over time; therefore, evaluating the effectiveness of an economic model for a large number of scenarios is crucial.

Realizing the ambiguity of choosing a suitable economic model for the Grid, in this thesis, we conduct a comprehensive literature review to gain insights on different models in Grid resource management (Haque et al., 2011). The review identifies the suitability of different models for different scenarios. For example, we identify that English Auction protocol is suitable for maximizing revenue for providers; however, it generates higher communication cost. Commodity Market model is suitable for maintaining market equilibrium, whereas Bargaining model shows its strength for supporting utility-based negotiation. We further identify that one model has been criticized by another in terms of various criteria.

Despite the variation of different economic models in terms of suitability for Grid resource collaboration, no research has identified the importance of analyzing the effectiveness of these economic models using a common evaluation platform. Therefore, our focus in this thesis is,

“to analyze and contrast the performances of widely proposed economic models for a wide range of scenarios and identify the domains where one economic model outperforms other economic models.”

1.2 Scope of the Thesis

In order to fulfill the focus of this thesis, we have the following scopes:

- ❖ To identify a set of key performance metrics that can be used to evaluate the performances of different economic models in Grid resource management
- ❖ To investigate and analyze the performances of the economic models, develop an evaluation framework suitable for measuring the effectiveness of the models for a wide range of scenarios in the Grid
- ❖ To compare among the performances of the economic models, and to identify the domains of strengths and weaknesses of individual models in terms of various performance metrics
- ❖ To identify the possibility of optimizing different performance metrics, through utilizing the potential of different economic models in different scenarios in a dynamic Grid computing environment

The following section describes the problem definition related to these scopes and proposes possible solutions to the problem.

1.3 Problem Description

We address the possibility of optimizing different performance metrics in a dynamic Grid environment through utilizing the potential of different economic models in different scenarios. In this section, we provide our research hypothesis, questions that rise from the hypothesis and possible solutions that support the hypothesis.

1.3.1 Research Hypothesis

The thesis is based on the following hypothesis:

“The diversity of different economic models can add value in solving distributed resource management problem in Grid Computing”

However, how one would be able to employ this diversity in a dynamic Grid computing scenario in order to optimize different objective functions, is a key issue. Therefore, we organize the following questions that help to understand more about the issue.

1.3.2 Research Questions

The following research questions are developed for this thesis:

- How one can understand the strengths of an economic model over other models in highly dynamic and distributed computing scenarios?

- How the scenarios are defined and relevant to the evaluation of the performances of the economic models?
- How one can justify the strengths of the economic models for unknown scenarios?
- How one can utilize the potential of individual economic models in a highly dynamic environment to deal with the diverse nature of the Grid?

In the following sub-section, we provide the solutions or research methodology used to solve these questions.

1.3.3 Proposed Solutions

For answering the first question, we propose the development of an evaluation framework in consideration of a wide range of scenarios in the Grid. The framework needs to incorporate the development and a consistent evaluation method for different economic models in the Grid. To deliver a comprehensive evaluation process, it should consider a wide range of performance metrics such as revenue earned by providers, communication cost by an economic model, and social welfare obtained from the environment. To be able to identify the strengths of a model, it would help perform a comparative analysis among the performances of different models.

It has been identified that the performance of an economic-based Grid management system is considerably influenced by supply (resources) and demand (the application requirements) (Buyya et al., 2002). In addition, the dynamics of Grid environment is also characterized by supply and demand, which is, the dynamic joining and leaving of Grid entities. Therefore, we propose to use supply and demand to define the scenarios as an answer to the second question. Any Grid system must consider this dynamic nature, before it can be evaluated through a simulated environment. Therefore, in this thesis, we would analyze the performances of different economic models in terms of all possible supply and demand variations when the total supply and demand is 100 units each; therefore, for each economic model, we will have 10^4 (100 by 100 matrix) different scenarios, which describe the simulation space on which the evaluation would take place. Such an evaluation would help to analyze the performances by a model for any given scenario in the space.

The nature of our simulation space would help identify the regions where one economic model shows its strengths over the others. To answer the third question, we propose to model the strengths of individual economic models mathematically. The formalization models would show the feasibility of considering extended scenarios to define the strengths of individual economic models. This means that the strength of a model would persist as long as the both known/unknown scenarios satisfy the formulae of the strengths for the model.

To be able to answer the fourth question, we propose to develop a switching mechanism, which would dynamically switch from one economic model to another depending on the models' domains of strengths. We describe the adaptive behavior by the Grid participants to deal with different models dynamically. To facilitate the autonomous switching, we propose the development of a switching agent that keeps track of scenarios in the environment and dynamically decides which model to choose when and for what reason. We would explore the necessary components in building such an agent that can adapt with the system as quickly as possible and could fulfill its role in the environment.

1.4 Research Contributions

To deal with the dynamic and diverse nature of Grid computing, this thesis describes the key features towards enabling the development of an adaptive economic-based resource management system. The contributions of this thesis are mentioned as follows:

1. This thesis develops an evaluation framework to investigate and analyze the performances of different economic models in the Grid. Economic models help collaborate distributed resources in Grid computing in a coherent and seamless manner. Until now, different researchers have conducted the evaluation process of different economic models using different evaluation platforms. However, for consistency, it is crucial to study the models using a same evaluation platform and on a same simulation space. The thesis develops such an evaluation framework using the widely discussed discrete-event Grid simulation toolkit – GridSim (Buyya and Murshed, 2002). The framework incorporates the five most widely proposed economic models in the Grid. The thesis contributes the development of Commodity Market Model, Bargaining Model and Contract Net Protocol to the current GridSim distributions. It further identifies a range of performance metrics suitable for evaluating any economic-based resource management system in the Grid. The framework has been configured with all of these evaluation metrics. To highlight the significance of the evaluation framework, the dynamic nature of the Grid is incorporated, that is, the framework evaluates the effectiveness of each economic models for all possible scenarios of supply and demand when the maximum value of supply and demand is 100 units each; therefore, the simulation space has been defined as a 100 by 100 matrix. Ultimately, these characteristics help to provide a suitable framework to investigate the performances of any economic model for a comprehensive set of scenarios in the Grid.

2. This thesis analyzes and contrasts among the performances of widely proposed economic models in the Grid, which facilitates to identify the regions where one economic model shows its strengths over other models. It further describes the regions (domains) of strengths of the models in terms of the parameters that defining those models. It performs a comparative analysis among the performances of all the five economic models and for a wide range of performance metrics. In essence, it facilitates to understand the strengths and weaknesses of a model quantitatively for a wide range of scenarios in the Grid. The thesis further provides mathematical definitions for these domains of strengths of the models. A scalability analysis of these strengths is significant for ensuring the suitability of the models for the Grid. The thesis identifies and formalizes the domains of strengths for individual models with respect to different performance functions. The simulation space in the evaluation framework helps to obtain clear trends for the strengths of the models. Therefore, even if we define the simulation space as 100 by 100, our formalization models still show consistency for considering the strength for extended and unknown scenarios. The thesis, therefore, delivers a mechanism to a reasonable understanding about the effectiveness of different models for distributed resource collaboration in the Grid.
3. This thesis designs and develops an optimization framework through the utilization of the potentials of different economic models in a dynamic Grid environment. The thesis identifies the opportunity of optimization by considering multiple economic models as different models perform better in different scenarios of a Grid's lifecycle. To facilitate the optimization process, a switching framework is developed that dynamically switches from one economic model to another depending on the models' domains of strengths. The framework provides necessary elements to enable the Grid entities to deal with dynamic changes of economic models in the environment. The adaptive capability of the framework further ensures seamless resource collaboration while obtaining desired optimization from the environment. To facilitate the decision process on which economic model to use when and for what reason, the thesis develops a switching agent. The agent is capable of keeping track of different scenarios and of providing decision, on which economic model to use at a scenario depending on previously identified models' domains of strengths. To understand the language/decision of the agent and work accordingly, extended broker and resource models are developed in the thesis. Overall, the thesis develops a framework towards providing support for dynamic organization of different economic models to deal with different scenarios in the Grid.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 contains the literature survey. It begins with a general overview of Grid architecture and economic-based resource management system in the Grid. It presents a wide range of performance metrics to evaluate the performances of different economic models in the Grid. Then, it analyzes different economic models in terms of their strengths and weaknesses as identified in the Grid literature. The chapter ends with identifying some research gaps that help in building up the key components of the thesis.

In Chapter 3, we present the development process of our evaluation framework. The chapter explains the simulation space and parameters used in the framework, to evaluate the performances of the economic models. It describes the statistical significance of the parameters and its relative significance in terms of identifying the effectiveness of a model. The chapter further provides the significance of the framework in terms of fair evaluation among different models.

Chapter 4 describes the design and development process of two commodity-based economic models – Commodity Market and Bargaining. We explain the key principles of the models and the foundation for Grid resource management system in terms of the models. Finally, the chapter describes the results obtained from the simulations of the two economic models, and performs a comparative analysis to a better understanding about the strengths and weaknesses of individual models.

Chapter 5 describes the development process of auction models – English Auction, Continuous Double Auction and Contract Net Protocol. It describes the respective roles of Auctioneer, users, brokers and resources in the environment. The chapter further explains the results obtained for the three economic models and conducts a comparative analysis among the performances of the models. The chapter ends with analyzing the strengths and weaknesses of the three models in terms of various performance metrics.

Chapter 6 presents a comparative analysis considering the performances of both Commodity and Auction models. It explains the domains of strengths of different economic models in terms of a wide range of performance metrics. Finally, it mathematically models the domains of strengths of individual models, and elaborates the possibility of optimization in a dynamic Grid.

Chapter 7 deals with designing and developing of the optimization framework. It proposes a switching framework and describes the working principles of the framework. It further explains the switching agent and its role in the dynamic environment. Finally, through a series of experiments, it presents the evaluation of the effectiveness of the switching mechanism.

Chapter 8 provides a brief summary of the thesis, conclusions and future research directions related to our work. Figure 1.3 illustrates the organization of this thesis.

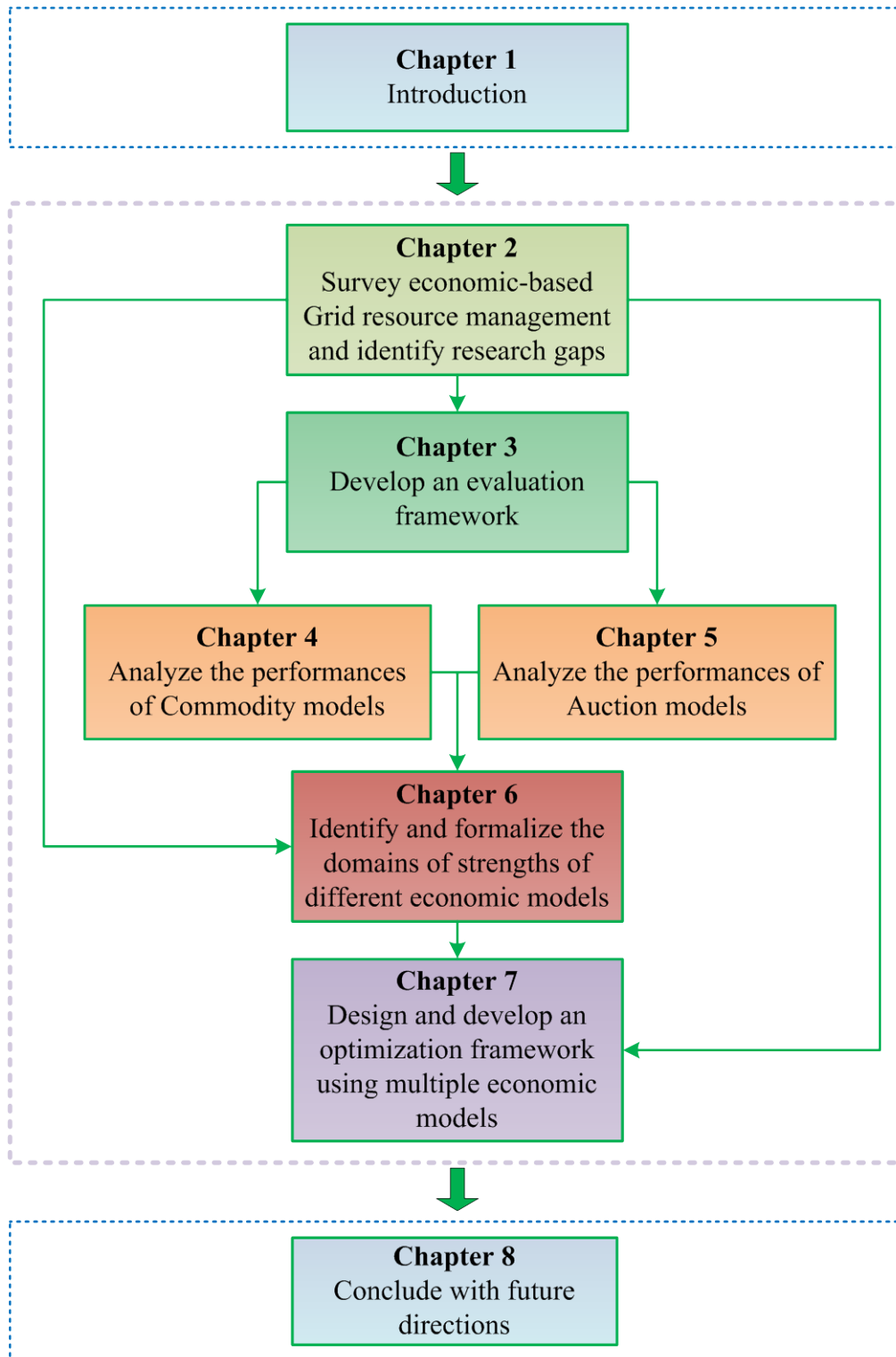


Figure 1.3: Organization of Thesis Chapters

1.6 Publication Record

Portion of the work presented in this thesis has been communicated to the research community through the following *publication during the tenure of my Ph.D. candidature*.

Chapter 1 and Chapter 2 have been partially derived from the following publications.

- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2011). “A Survey of Economic Models in Grid Computing”. *Future Generation Computer Systems*. The International Journal of Grid Computing and eScience, Volume 27(8), pp. 1057-1069, ISSN: 0167-739X.
- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2009). “Some Reflections on Research Issues in Grid Computing”, Proceedings of 12th International Business Information Management Association (12th IBIMA), 29-30 June, Kuala Lumpur, Malaysia, pp. 1616-1623, ISBN: 978-0-9821-4891-4.

Chapter 3, Chapter 4 and Chapter 5 have been partially derived from the following publications.

- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2010). “Implementation of Autonomous Pricing Mechanism in Grid Computing”, Proceedings of 12th International Asia-Pacific Web Conference (12th APWeb), 6-8 April, Busan, Korea, pp. 381-383, ISBN: 978-0-7695-4012-2.
- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2011). “Towards Better Understanding of Two Economic Models: A Grid Perspective”. *Communications of the IBIMA*, Volume 2011(2011), Article ID 223098, pp. 1-13, ISSN: 1943-7765.
- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2010). “Continuous Double Auction in Grid Computing: An Agent-based Approach to Maximize Profit for Providers”, Proceedings of IEEE/WIC/ACM International Conference of Agent Technology (IAT 2010), 1-3 September, Toronto, Canada, pp. 347-351, ISBN: 978-0-7695-4191-4.
- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2010). “A Comparative Economic Approach to Maximize Profit for Providers in Grid Computing”, Proceedings of 14th International Business Information Management Association (14th IBIMA), 23-24 June, Istanbul, Turkey, pp. 869-881, ISBN: 978-0-9821-4893-8.

Chapter 6 has been partially derived from the following publication.

- **Aminul Haque**, Saadat M Alhashmi and Rajendran Parthiban (2012). “An Inspiration for Solving Grid Resource Management Problem Using Multiple Economic Models”, Proceedings of 8th International Workshop on the Economics and Business of Grids, Clouds, Systems, and Services (GECON 2011), in conjunction with 9th International Conference on Service Oriented Computing (ICSOC 2011), LNCS – 7150, pp. 1-16, 5-8 December, Paphos, Cyprus, ISBN: 978-3-642-28674-2.

Economic-based Resource Management in Grid Computing

This chapter analyzes economic-based resource management models in Grid computing. It also discusses significant technological evolutions that gave rise to Grid computing. The motivation for economic-based resource management and various criteria to judge it are presented. Finally, a clear definition of the research gap has been given and the potential of the research problem has been made transparent.

2.1 Introduction

Investigation of some problems in science, engineering and commerce, such as protein analysis, the properties of materials and economic forecasting are computationally complex. Realizing the insufficiency of a single computer, a cluster of computers or even a supercomputer to solve these problems, Grid computing was introduced in the mid 1990s (Roure et al., 2003), which is the technology that aggregates distributed computer resources over the Internet. Coordination of distributed and heterogeneous computing resources create virtual organizations (VO) that support utilization of idle resources (Neumann et al., 2008). However, seamless collaboration is a challenge due to the extreme heterogeneity of these resources. This heterogeneity is due to varying architectures of physical resources (e.g., clusters, supercomputers, ordinary PCs), different administrative domains (e.g., country, enterprise) and multiple operating systems (e.g., UNIX variants, Windows). There is also a lack of a uniform way to use these resources.

As Grid resources typically belong to different administrations, dynamic resource management and scheduling is challenging. Numerous research has been conducted and it has been identified that economic based approaches have the potential to meet this challenge (Buyya et al., 2002, Mills and Dabrowski, 2008, Beck et al., 2008). Economic-based approaches not only help to solve distributed scheduling problem, but also bring sufficient motivation for resource providers to be a part of the Grid. Various economic models, such as Commodity Market, Double Auction and English Auction are proposed for Grid computing from its initialization through 2011. Economic models define the interaction protocol between

a user and a provider, and a method for pricing resources, which ultimately helps in building a standard large-scale computing platform. Thus, the organization of economic-based Grid systems are more complex compared to noneconomic based systems. Different researchers have proposed different economic models at different times for the Grid. Choosing a model from a set of models is challenging due to several reasons:

- the standards of a model would be static; however, Grid is dynamic,
- a model could only provide limited features to utilize the potential of a Grid, while Grid users could have different aspirations from the Grid and,
- the examination of the sustainability of a model for various scenarios is hard.

Realizing this complexity of choosing an appropriate model, we conduct a comprehensive survey on existing economic models in Grid computing (Haque et al., 2011). This chapter is dedicated to this survey. We define a comprehensive set of metrics to judge various Grid-based economic models. The principles of different economic models and their relative consequences on Grid environment in terms of those metrics are analyzed. The rest of the chapter is organized as follows.

In Section 2.2, we introduce Grid computing and the inspiration behind it. Section 2.3 describes a typical Grid architecture. Section 2.4 and Section 2.5 incorporates the motivation and architectural requirements for economic-based Grid systems respectively. Various performance criteria to evaluate economic models are introduced in Section 2.6. Then, Section 2.7 presents the analysis of different economic models in the light of existing Grid literature. The significance of economic models in the Grid since its initialization and some open research issues are discussed in Section 2.8. Finally, Section 2.9 draws the conclusion of this chapter.

2.2 Grid Computing: A Transition from Mainframe to Meta-computing

In this section, we briefly introduce Grid computing and discuss its evolution. In the next section, we explain its architecture and use it to inspire the discussion of economic models in the field. Figure 2.1 illustrates two main eras in distributed computing. The eras are mainframe and meta-computing. Needless to say, the continuous technological advances during the era of mainframe computing help us to think about computing as another utility like electricity or water, available on demand. Despite the different positions of Grid, P2P and Cloud under meta-computing, they share similar goals and hence tackle similar challenges, which we outline at the end of this section.

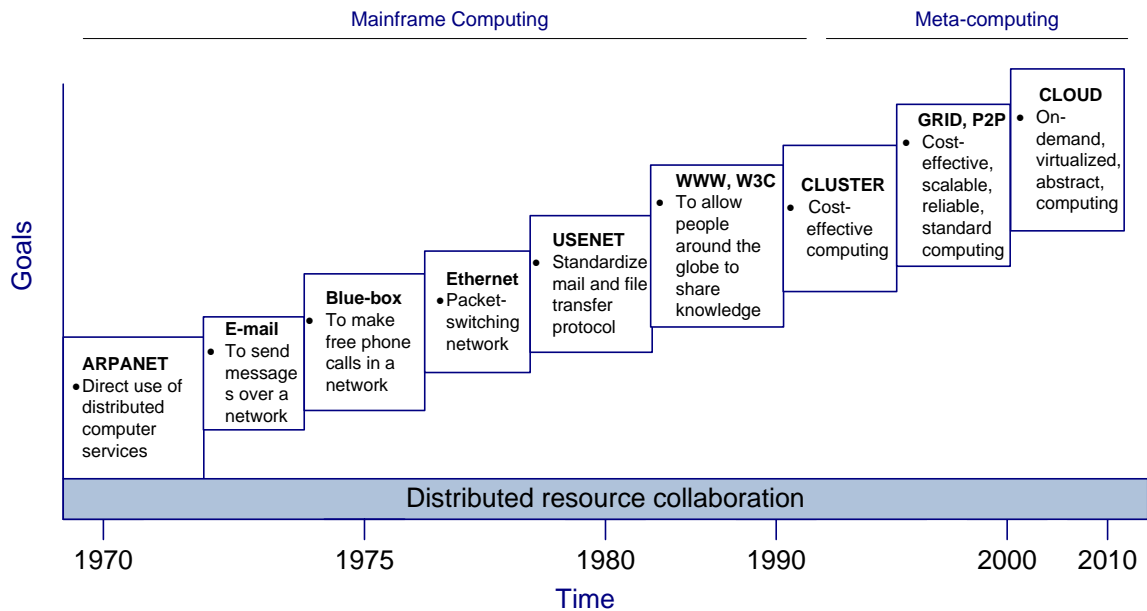


Figure 2.1: Major Technological Shifts in Distributed Computing

Advanced Research Projects Agency Network (ARPANET) in early 1970s established the first resource-sharing network with four computing nodes. The primary goal was to retrieve computer resources from a remote node (Museum, 2006). This establishment worked as a revolution for subsequent inventions in network technologies. In 1971, Ray Tomlinson first sent an e-mail through ARPANET network. Following this, Steve Wozniak built “Blue Box” to make free phone calls over a network. Another shift in distributed computing happened, when Robert Metcalfe invented Ethernet in 1973. Messages could be broken down into several packets and sent to a destination, due to this invention.

The hope of modern large-scale interaction was first seen when “Usenet” established. Duke University and the University of North Carolina developed it as a joint project. The project enabled its users to e-mail and transfer files using a communication standard known as UUCP (Unix-to-Unix Copy). This consequently helped to give rise of WWW (World Wide Web) on a global scale, helping people to interact and share their knowledge through one platform. W3 Consortium was introduced to coordinate WWW developments.

With the advances of networking technologies and falling prices of computers, clusters of computers were looked at as competitors to mainframe computers (supercomputers). The primary goal of cluster computing was to serve the scientific community in a cost-effective manner. However, the computing could not satisfy the community’s objective, because a cluster could not break down the barriers of organizational boundaries. The main reason for this could be the architectural and administrative constraints over resources in expanding a cluster. As a result, the need for a special computing, which shares resources across many

organizational boundaries becomes transparent. Realizing this, Grid Computing was introduced. Grid computing, also faced several challenges, such as security, reliability, availability, different administrative policies and a uniform method to use these resources, due to the large scale of resource collaboration over the Internet. Several Grid middleware, such as Globus, Condor and Legion were initiated to meet these challenges. P2P (Peer to Peer) is another contemporary computing method of the Grid. P2P plays a vital role in forming different Grid networks especially in multi-organizational Grid systems (Plaszczak and Wellner, 2005).

Until the Grid, scientific applications such as prediction of earthquakes or hurricanes would require supercomputers costing millions of dollars. Only rich countries were able to purchase such machines. Grid computing was more cost-effective and brought hope for countries with limited resources to find solutions for many life-threatening situations, which was articulated in <<http://www.buyya.com/press/TechBusiness02.pdf>>.

Some tools are available for running Grid-related experiments. Nimrod/K tool supports the workflow design and optimization through necessary parameterization for solving computationally complex problems in Quantum Chemistry and Cardiac Science (Abramson et al., 2011). FOLDING@Home is another initiative that uses distributed computing resources to study on protein folding to search for cures to diseases. The details can be found at <<http://folding.stanford.edu/English/HomePage>>.

The main objective of meta-computing is to provide computing as a service by hiding all the resource management procedures from its end-users. Recently, another computing platform has emerged – Cloud Computing. Different from the Grid, Cloud does not access resources directly, but through an abstract or more precisely a virtual layer. However, Grid is still the backbone of Cloud computing and they mainly share similar visions (Foster et al., 2008). As a result, Cloud is facing the same challenges as Grid does, which are dynamic resource management, standards for using resources and reliability of the resources.

In this thesis, we discuss resource management in Grid computing. As Grid aims to share resources across geographical boundaries, required technologies for seamless resource collaboration need to be defined. The following section presents the key components that form a typical Grid as part of that defining process.

2.3 A Layered Grid Architecture

Prior to discuss economic models in Grid computing, it is necessary to understand the organization of a typical Grid. Therefore, this section discusses the necessary materials that constitute a general Grid. Figure 2.2 shows the layers and different components that constitute

a typical Grid. The layered Grid architecture usually rests on the *fabric layer* that consists of servers, clusters, monitors and all other distributed computing resources around the world. Mercury (Kacsuk et al., 2003) system is a good example for this layer. The layer that controls and allows users secure access to the components of the fabric layer is called the *core middleware layer*. This layer also supports trading and information updating of resources. Globus (Foster, 2006) is a well-known middleware service, which allows resource discovery, management and security. On the other hand, Gridbus (Buyya et al., 2009a) middleware supports business driven technologies aimed at utility based computing. Gridbus uses economic models that aid the efficient management of shared resources through maintaining the supply and demand of distributed resources.

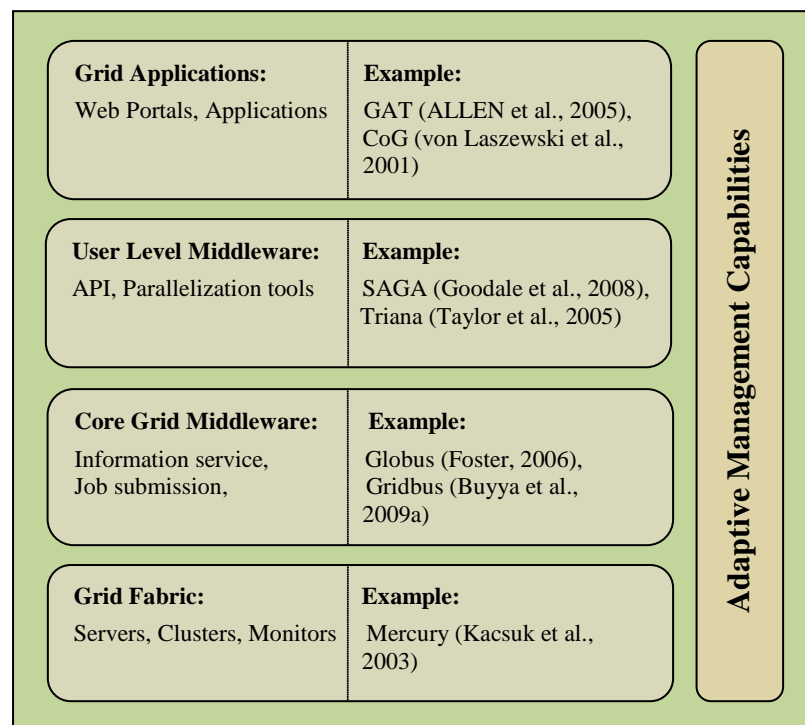


Figure 2.2: Layered Grid Architecture with Examples

In this thesis, we focus on suitable economic models in Grid computing and their practicality of usage in different perspectives in the field. The upper level of core Grid middleware is called *user level middleware*. This layer provides API (Application Programming Interface), libraries, application development environments and the resource mediator. The resource mediator negotiates between users and providers, and it schedules application tasks for execution on global resources. The Simple API Grid Applications (SAGA) (Goodale et al., 2008) and Triana (Taylor et al., 2005) are two examples of user level middleware. This middleware is used to communicate with the core middleware. *Grid applications*, the fourth layer, is typically developed using components of the user level middleware. This layer supports users execute their applications on remote resources and collect results from the

resources using web portals or applications such as the Grid Application Toolkit (GAT) (ALLEN et al., 2005) and java Commodity Grid kit (CoG) (von Laszewski et al., 2001).

The major part of the Grid is called the core Grid middleware, because it offers all the necessary functions, such as scheduling, security, data transfer, trading and communication (Caracas and Altmann, 2007). The main objective of this middleware is to hide the heterogeneous nature and provide a homogeneous and flexible environment to end users. Above all, there should be adaptive management capabilities across the layers to achieve seamless resource collaboration.

Grid resources are typically owned by different providers across educational, business, and commercial institutions. Thus, trading becomes one of the main reasons that motivate resource providers to contribute their resources on the Grid. In addition, price is a key decision factor in resource use (Kenyon and Cheliotis, 2003). Price can further maintain equilibrium between supply and demand, distinguish different QoS (Quality of Service) requirements and to maximize the utilization of idle resources. A market oriented modeling can be used in solving distributed resource management problems, such as site autonomy problem, objective optimization problem and cost management problem (Ernemann and Yahyapour, 2004). The site autonomy problem could occur when accessing resources that belong to different administrative domains. The objective optimization problem occurs when users want to optimize their QoS and when providers want to maximize their profit. Grid resource providers need to support seamless management of different requests from different users simultaneously, creating cost management problem. In the following section, we further emphasize the importance of economic-based approaches for the Grid.

2.4 Inspiration for Economic Grids

This section provides a brief discussion on the significance of applying economic-based resource management in Grid computing. Technology, business and policy are interdependent; without technology there are no services and products to be invented, and without business models no policies are needed to regulate their actions (Eymann et al., 2008). Buyya argues that the Grid's heterogeneity and decentralization are similar to the present standard human economy (Buyya et al., 2002), where market based mechanisms could be used to manage the environment successfully. He further argues that this approach would be efficient for balancing supply and demand and it is scalable (no need for central coordinator during negotiation). Additionally, it improves utilization of idle resources and distinguishes different quality of services. Similar measurements of a market based Grid can be seen in

(Howie et al., 2007), (Beck et al., 2008), (Sandholm et al., 2006), (Buyya, 2002), (Czajkowski et al., 1998).

Traditional market pricing models for managing Grid resources would also be applicable for managing self- interested and self-regulating entities (resource providers and users) (Cohen and Feser, 2003). A study undertaken by Cohen and Feser (Kenyon and Cheliotis, 2003) demonstrates the possible macroeconomic value for the introduction of Grid computing and forecasts a huge amount of gain through the deployment of high performance Grid and web service applications. The paper argues that price impact could be particularly beneficial for industrial firms, which often use the Grid. Using it, the firms become more competitive than might otherwise be expected. Grid technology enables the compilation of resources across many budget boundaries (accessing different economic goals). Therefore, an appropriate business model would be the key term for fair dynamic resource collaboration. Price can also be a key decision factor in resource use. In a market oriented approach, uncertainty drives a large portion of the decisions, -questions such as what are available when and for what price (Kenyon and Cheliotis, 2003). According to one of the leading Grid computing resource institutes, The 451 Group, the application of resource trading and allocation models is one of the crucial success factors for establishing commercial Grids (Fellows and Wallage, 2007). Therefore, a suitable pricing model for Internet services is one of the main prerequisites for successfully running the implementation of an accounting and charging system. Shin et al. (Yeo and Buyya, 2007) focus on a pricing mechanism to support utility driven management and allocation of resources. Accordingly, the providers should have mechanisms for generic pricing schemes to increase system utilization and protocols that help them offer competitive services.

In early days, considerable amount of money was spent for designing and developing a single supercomputer to solve complex scientific problems. Once the experiment finished, the computer became useless due to its high architectural dependency. There was no need for a pricing model in such a scenario. Now computing resources that are distributed across the globe are being used for the same purpose. Hence, capital expenditure (budget) also needs to be distributed in an intelligent and efficient way, which gives additional complexity. As such, a sustainable market model is crucial for Grid computing. Figure 2.3 illustrates a conceptual view of a typical economic Grid. There are different heterogeneous sites along with their unique access policies. Clients have their demand and budget for using those heterogeneous resources. Section 2.5 elaborates this concept from an architectural point of view.

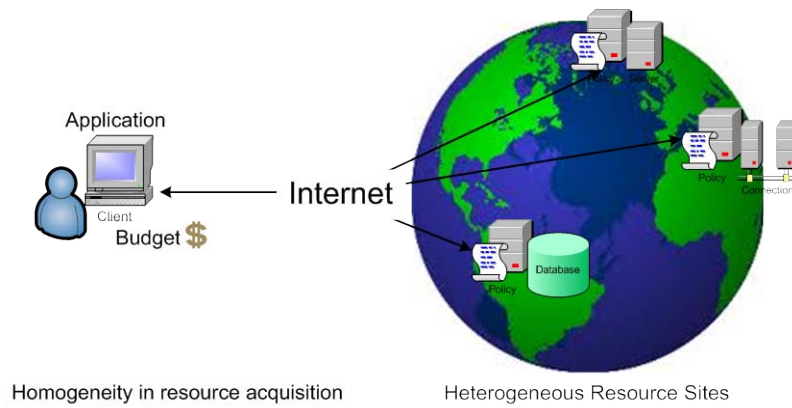


Figure 2.3: Conceptual View of an Economic Grid

2.5 Market-oriented Grid Computing

Standardization, usability and business models have been accepted as the main success factors for next generation computing systems (Neubauer et al., 2006). However, market based computing mechanisms are different from the traditional mechanisms in terms of the value (i.e. QoS) delivered to a user. The value could be measured by the following parameters:

- flexibility in parameterization of user driven jobs,
- suitability of business models for different user requirements and strategies and
- adaptation to changes in resource availability, capability and pricing.

To justify such parameters, market oriented computing organizations need to be more complex than the traditional systems. This section outlines the key elements in developing a market-based architecture for the Grid.

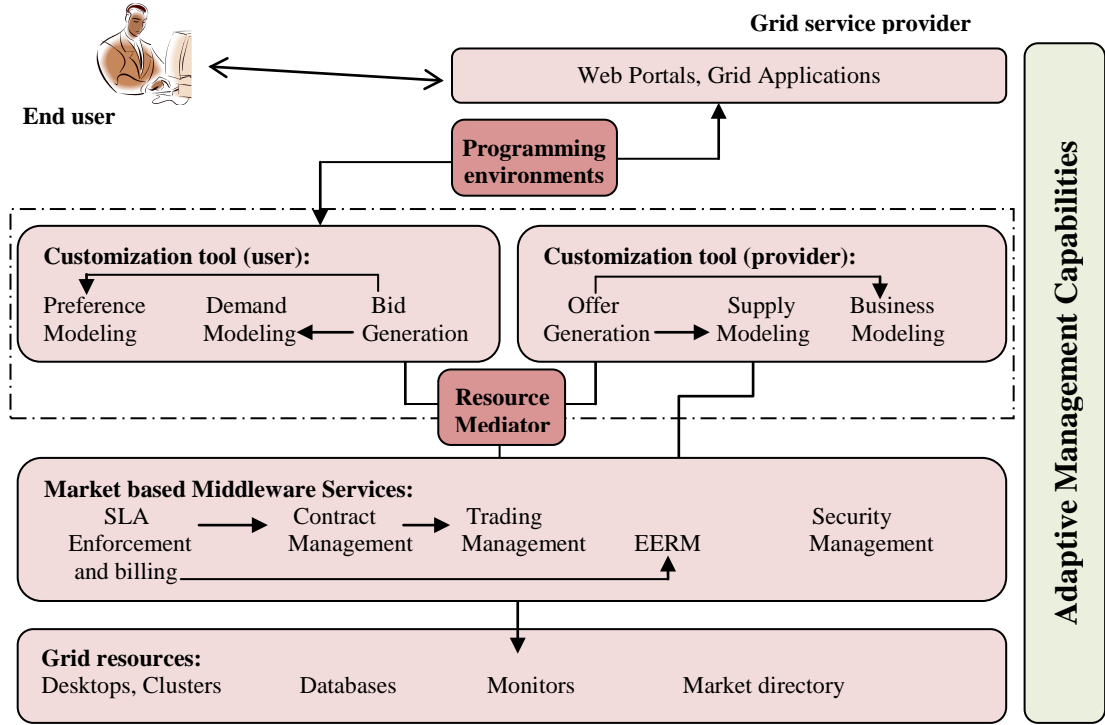


Figure 2.4: A reference Market Oriented Overview of a Layered Grid Architecture (Neumann et al., 2008)

Figure 2.4 presents the four Grid layers of Figure 2.2 in terms of a market oriented modeling environment. Each layer has some additional functional entities along with the dependencies among them. The arrow from A to B refers to the dependency of A on B. In a market-oriented architecture, market directory service keeps the resource information updated and helps generate a competitive market price for a resource. Market based middleware supports market participants to trade Grid resources. It performs trading activities, such as SLA (Service Level Agreement) enforcement and billing, contract and trading management. All of these activities are performed to decide which resources are allocated to which user, for what price and for how long. Economically Enhanced Resource Management (EERM) isolates users from its providers based on certain market relevant features to increase its functionality. EERM also keeps itself updated with the resource state through monitoring services and reports to the SLA enforcement. EERM gets information from supply modeling and assists to form SLA. Supply modeling depends on demand modeling, which provides necessary tools to specify resource properties. Offers are generated based on both supply and business models. On the other hand, bids are generated based on users' demands and preferences (e.g., economic preferences). A resource mediator negotiates between resource users and providers. A resource mediator could be a resource broker or a resource agent.

Market oriented Grids require adaptive management capabilities among different functional entities to enhance the service quality delivered to users as well as to optimize providers' goals.

Economic model is a key component for market-based Grid, because the model describes the behavior of the entities in the market. Therefore, a robust and viable economic model is required to deal with pricing distributed resources across multiple administrative domains. In addition, economic models help providers treating different users differently based on their requirements and organizing corresponding SLAs, which collectively would construct a rigid market oriented computing environment. Buyya proposes several economic models (such as Commodity Market, posted price, and bartering) including both micro and macroeconomic principles for distributed resource management (Buyya et al., 2002). However, he only discusses a hypothetical suitability of these models for a Grid environment. Not all the models proposed are suitable to deal with all different scenarios, as we will see in Section 2.7. In Section 2.7, we will discuss the models that have been studied and analyzed by different Grid researchers. We address the strengths and weaknesses of these models in terms of managing heterogeneous Grid resources. Before starting the discussion on various economic models in the Grid, let us introduce the criteria that can be used as probes to judge different economic Grids.

2.6 Criteria to Judge Economic Grids

Economic models are different from one another in terms of the methods (i) they use for interaction among users and providers, (ii) they use for pricing resources, and (iii) they adopt to evaluate different resource requirements. In a Grid computing environment, the strengths and weaknesses of an economic model can be evaluated using several criteria. Some of them are mentioned here with a brief explanation.

Admission control: Admission control refers to the control of submitting new jobs in a Grid for execution. This feature plays a pivotal role in maintaining market equilibrium. In a market oriented Grid, dynamic pricing can be used to maintain equilibrium between supply and demand as an effective means.

Broadcasting overhead: This is also known as *distribution and communication overhead/cost*. It is the delay incurred to disseminate information regarding resource availability, pricing bids, and so on. over the Internet. It also depends on the geographic distance of computing endpoints, Internet speed and communication protocols.

Computation efficiency: This is the amount of computation time that is consumed by a model while evaluating users' requests. Models that consume less computation cycles are considered as computationally efficient.

Decentralization: Decentralization in an economic based Grid refers to freedom to set a resource price by a provider. In a distributed environment such as Grid computing, decentralization in pricing is expected to achieve large-scale resource collaboration. It can also be used to evaluate *global allocation efficiency*.

Evaluating market price: The market price or economic price of a resource could be manipulated by the price offered for the same resource by different providers. The demand on a resource also contributes to determine the market price of that resource. A true market price is crucial in achieving a competitive Grid market.

Handling a large number of users: This criterion refers to the ability to contact and evaluate a large number of requests with their different QoS requirements within a particular period. Typically the Grid resources are utilized by global users over the Internet. Hence, there could be many users in general. This criterion could also refer to the *scalability* of a model in terms of dealing with many users.

Job cancellation rate: This is the rate at which requested or submitted jobs are cancelled by a market. Job cancellation could occur due to the disagreement of prices, unavailability of resources or failure while executing the jobs in the Grid.

Price stability: This criterion explains the stability of a market price for a resource for a specific period. Inflation is the opposite of price stability. Inflation is the rising of resource prices. Again price stability is crucial to ensure scheduling stability (Wolski et al., 2001).

Pareto optimal allocation: This is a resource allocation process, in which allocating of a resource is not supposed to affect other resources that are currently being allocated or executed (Gradwell and Padget, 2005). This process is necessary in Grid computing to get the jobs done according to their deadlines. Economic models play a crucial role here in setting different SLAs for different users.

Utility based negotiation: This is also called *individual rationality*, which refers to the payoff gained through participating in the negotiation. In an agent based negotiation, individual rationality means that, all the agents in the system agree to participate in negotiation, because all of them individually are assured of receiving better payoffs than in the case of not participating (Dash et al., 2003). In a Grid market, negotiation could happen between a user and a provider individually in order to optimize their goals or objectives.

Resource allocation efficiency: The ability to allocate an appropriate amount of resources according to the needs of users is called resource allocation efficiency (Ardaiz et al., 2006). It helps users to get their jobs executed within their job deadlines. Similarly, providers would also benefit through provisioning their resources to the users.

Market Liquidity: If there is a likelihood of matching a large number of users' requests and providers' availability in a market, the market is referred to as highly liquid (Risch et al., 2010). Market liquidity is necessary to ensure that there are a significant number of tradable resources in the market. However, the differences of Grid resources impose a challenge in defining a reasonable liquidity (setting resource prices) in a market to ensure that providers are not selling their resources at a loss.

Economic efficiency: The economic efficiency of an economic model defines how efficient the model is in utilizing idle resources as well as maximizing profit for providers. Profit for a period could be presented as the difference between the total revenue gained and total expenses associated with providing services (e.g., communication cost) throughout that period. In principle, economic efficiency depends on all the aforementioned criteria. In addition, from a user's point of view, a model can be efficient if it supports the user's requirements. Therefore, the economic efficiency could also be treated as *user provider efficiency*. *Social welfare* can be used to determine economic efficiency. *Social welfare* is calculated by aggregating users' and providers' *utility*, which for an entity (e.g., user/provider) is defined as the difference between his/her reservation-price¹ and the agreed-price² for a service.

2.7 Economic Models and their Strengths and Weaknesses in Grid Computing

In this section, we investigate different economic models proposed by different Grid computing researchers since the initiation of the Grid. The section further incorporates the discussion on the potential of the models in recently emerged Cloud computing paradigm. We provide a brief explanation about different economic models, because an extensive explanation on different models has already been given by (Buyya et al., 2002). The focus of this section is to present the economic models in terms of their strengths and weaknesses as identified by different Grid computing researchers at different times. At first, we present the strengths of different models in Table 2.1. The first column of Table 2.1 presents the economic models along with a brief definition. Column 2 presents the relative strengths of the models. At last, column 3 incorporates various approaches by different papers using the economic models. The methodologies used by these papers are based on the strengths mentioned in column 2. A detailed discussion on how these methodologies of different economic models work is given after the table.

¹ Price limit at which a user/provider agrees to buy/sell a particular service

² Price at which both user and provider are satisfied to trade

Table 2.1: Strengths of Economic Models in Grid Computing

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
Commodity Market: In general, resources are priced in such a way so that equilibrium between supply and demand is maintained. There are two types of Commodity Market Models in general: flat pricing model and supply and demand based pricing model. The latter is more popular in Grid research because it has the capability to maintain equilibrium between resource supply and demand by changing price behavior	Admission control, computing efficiency, economic efficiency, flexibility in evaluating market price, Pareto optimal allocation, resource allocation efficiency, price stability	(Vanmechelen et al., 2011): Integration of future contracts in supply and demand driven market to eliminate the risk associated with the deadline of short-term tasks while ensuring profit for providers
		(Turner et al., 2010): Utilizing the concept of spot and future markets in efficient bandwidth allocation for internet-based applications
		(Abdelkader et al., 2010): Investigating the robustness of commodity market resource allocation considering large varieties of interchangeable resources
		(Gomes and Kowalczyk, 2010): Ensure competitive system welfare for both individual and global rational agents in a distributed environment
		(Bossenbroek et al., 2009): Minimize the risk associated with service offering/requesting due to price volatility by adopting hedge strategy ³
		(Garg et al., 2009): Minimizing cost and execution time using meta-scheduling heuristics
		(Buyya and Abramson, 2009): Economic framework for service driven resource collaboration
		(Buyya et al., 2009b): The potential of market-oriented Cloud computing in delivering a service-oriented computing facilities to world-wide consumers
		(Mills and Dabrowski, 2008): Compare economic based approach over non economic centralized approaches
		(Ni et al., 2007): Utilize Grid resources based on appropriate service selection

³ Make contracts to obtain the rights of buying/selling a particular service within a specific period and at a specific price

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
		(Yeo and Buyya, 2007): Support user centric job specification for suitable allocation decisions
		(Stuer et al., 2007): Achieve effective allocation while ensuring price stability and service fairness
		(Sonmez and Gursoy, 2007): Economic based scheduling for time cost optimization and parameter sweep applications
		(Lu and Yang, 2006): Minimizing job cancellation rate, while ensuring scalability and load balance
		(Li and Li, 2006): Optimize aggregate utilization for Grid users, while maximizing revenue for providers
		(Li et al., 2005): Study utility based allocation algorithm properties under budget and time constraints
		(Buyya et al., 2005): Schedule computationally complex and data intensive applications under economic driven Grid
		(Tianfield, 2005): Agent based negotiation for distributed resource management
		(Chen et al., 2004): Effective task scheduling in supply and demand driven Grid computing
		(Wolski et al., 2001): Compare commodity and auction protocols' effectiveness in terms of market control
		(Buyya et al., 2000a): Dynamic resource trading for flexible application scheduling
		(Buyya et al., 2000c): Scheduling using adaptive management of computational resources
		(Veit et al., 2007): A comparative study between Commodity Market and Vickrey Auction models in case of price stability, fairness of allocations and communication requirements
Double Auction:	Market	(Prodan et al., 2011): Minimizing execution time and

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
<p>Providers are arranged in ascending order and users in descending order in terms of demand and budget respectively. If a user's request matches with a provider's offer, the trade is performed. There are two types of Double Auctions – Continuous Double Auction (CDA) and Periodic Double Auction. In CDA, users post their requirements and budgets, and service providers post their offers at any time during the trading period whereas in Periodic Double Auction, an</p>	competition, resource allocation efficiency, broadcasting overhead, computing efficiency,	cost by understanding scheduler behavior over resource allocation
	handling a large number of users, price stability, decentralization, economic efficiency	(Kun et al., 2010): Improving resource allocation process by considering trust and security concerns in a system
		(Borissov et al., 2010): Study bidding policy for users and providers to maximize their utilities in an independent market scenario
		(Buyya et al., 2010): Dynamic service distribution across different Cloud providers to meet dynamic QoS patterns requested by users
		(Izakian et al., 2009): Maximize task completion rate, utilization of resources and profit for providers
		(Li et al., 2009): Support combinatorial bids and exhibit incentive characteristics for both users and providers
		(Wang and Wang, 2009): Motivate users and providers through supporting individual rationality
		(Lynar et al., 2009): Study variation of time and energy consumption by applying different auction protocols
		(Suri and Singh, 2009): Maintain price stability using knowledge based policy
		(Valkenhoef et al., 2010): Compare TCDA ⁴ with traditional CDA in terms of execution uncertainty
		(Wieczorek et al., 2008): Study workflow behavior to support faster and cheaper execution
		(Streitberger et al., 2008): Compare centralized and decentralized service allocation in terms of time and utility function
		(Pourebrahimi et al., 2007): Decision making agents

⁴ Trust based Continuous Double Auction: supports agents to commit to trades they trust

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
<p> auction continues for a specific time as defined by the Auctioneer. The former type is mostly discussed in the Grid literature. </p>		<p>adapt to dynamic network environment and pricing</p>
		<p>(Tan and Gurd, 2007): Investigate market price through providing a bidding adjustment strategy</p>
		<p>(Chuliang et al., 2007): Resource type based modeling to support dynamic adjustment of pricing</p>
		<p>(Assuncao and Buyya, 2006): Compare communication overhead and profit for different auction models</p>
		<p>(Pourebrahimi et al., 2006): Study pricing function in balanced and unbalanced networks with self interested agents</p>
		<p>(Placek and Buyya, 2006): Support organizations to federate storage services among them and lease them globally</p>
		<p>(Eymann et al., 2006): Design ALN⁵ for both centralized and decentralized organizations</p>
		<p>(Kant and Grosu, 2005): A comparative approach of different Double Auction protocols to maximize resource utilization</p>
		<p>(Buyya et al., 2005): Schedule computationally complex and data intensive applications under economic driven Grid</p>
		<p>(Gradwell and Padget, 2005): Multiple single item auction to avoid complexity with combinatorial bids</p>
		<p>(Chen et al., 2004): Study pricing algorithms for users and providers separately to measure the integrity of requests and offers</p>
		<p>(Grosu and Das, 2004): Study economic efficiency and system performance of three auction models</p>
		<p>(Gomoluch and Schroeder, 2003): Study system load, heterogeneity and communication delay with three</p>

⁵ Application Layer Network: hides heterogeneity of a service network from users' view

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
		different models
		(He et al., 2003): Manipulate and adapt to dynamic market price using fuzzy logic
English Auction: According to this auction, users are free to increase their bids overtaking others. When no bidder is interested to increase the price anymore, then, the auction ends. The Auctioneer declares the highest bidder as the winner. Bids can be proposed for a single item (or single attribute) or multiple items (or multiple attributes)	QoS, economic efficiency, revenue, resource allocation efficiency	(Xing and Lan, 2009): Develop resource mapping algorithms using iterative combinatorial auction mechanism
		(Schnizler, 2008): Design an auction based constructive economic model to assist Grid users in expressing their true demands
		(Beck et al., 2008): Study economic efficiency while providing suitable allocation and learning models
		(Middleton et al., 2007): Implement a simulation toolkit considering the rising needs from medical services
		(Attanasio et al., 2006): Develop auction mechanisms, while ensuring minimal communication overhead with efficient resource usage
		(Brunelle et al., 2006): Simplify job scheduling through an economic platform in decentralized Grids
		(Gradwell and Padget, 2005): Auction with many items to avoid complexity in combinatorial bids
		(Tianfield, 2005): Study agent technology for adaptive, run time efficiency and autonomous Grid
		(AuYoung et al., 2004): Analyze scalability, efficiency and long term behavior for resources allocated in federated Grid
Bargaining: In this model, users like to get a lower access price and higher usage duration. The providers like to	Utility based negotiation	(Borissov et al., 2010): Resource allocation using bargaining models especially in the context of Grid dynamics, utility and suitable relaxation of bargaining terms
		(An et al., 2010): Scaling Cloud resources by employing agent-guided flexible negotiation strategies

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
<p>get more profit through bargaining. The users might start with a very low price and providers with a higher price. Bargaining may continue over multiple attributes (e.g., price, deadline/job-execution-time).</p>		(Sim, 2010): The role of complex negotiation process for a reasonable SLA establishment among Cloud market entities
		(Subrata et al., 2009): Develop semi-static scheduling algorithm to maximize utility for providers
		(Zhao and Li, 2009): Maintain market equilibrium and maximize profit through self adaptive autonomous negotiation
		(Minh and Altmann, 2008): Determine market price based on deadline, urgency of workflow management and Grid state
		(Assuncao and Buyya, 2008): Optimize resource utilization and load balance across federated Grids
		(Wang et al., 2008): Analyze agent based mechanism of the resource discovery for insufficient budgeted users
		(Assuncao et al., 2007): Implement a simulation environment suitable for utility based Grid computing
		(Sim, 2007): Determine appropriate amount of relaxation in negotiation criteria to maximize utility and success rate
		(Jiadao and Yahyapour, 2007): Analyze time and learning based negotiation strategies for adapting with dynamic Grid
		(Ghosh et al., 2005): Harness computing power in mobile devices through an efficient pricing strategy to allocate jobs on them
		(SIM, 2005): Study Bargaining Models by considering Grid dynamics and appropriate relaxation of bargaining terms
Proportional Share Based	Economic efficiency,	(Lai et al., 2005): Allocate hosts efficiently in a cluster. Develop an agent based approach to allocate

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
Auction: ⁶	scalability	resources
		(Leon et al., 2010): Manage resource load on distributed large scale infrastructure by controlling resource prices across a Grid network
Proportional resource share: In this mechanism, the percentage of the resource share allocation to a user's application is proportional to the bid value in comparison to the other users' bids.	Less job cancellation rate	(Falavinha et al., 2009): Allocate resources fairly through owner share enforcement policy and distributed ownership concept
		(Gomoluch and Schroeder, 2003): Study and compare Double Auction and proportional resource share in terms of system load, heterogeneity and communication delay
		(Li and Li, 2004): Apply agent technology to maximize job accomplishment rate, while minimizing the cost accrued
		(Sherwani et al., 2004): Share deadline and budget strategically by considering user utility rather than system performance
First Price Sealed Bid Auction: ⁷	Resource allocation	(Danak, 2011): To improve bidding strategy for budget-constrained users in the Grid
	efficiency, global allocation efficiency	(Chun et al., 2005): Deploy testbed resources to computing users in an aggregated manner through combinatorial auction
Contract Net Protocol: According to this model, a user is called a manager and a provider is called contractor. Here, a manager	Utility based negotiation, scalability, resource co-operation, meta-scheduling	(Thabet et al., 2011): A macro-level study towards improving Grid scheduling process considering non-deterministic Grid entities
		(Kakarontzas et al., 2011): Improving resource management process by considering application constraints in Grid environments
		(Paurobally, 2010): A motivational framework for Grid resource providers to form VO through

⁶ This is like the English Auction, except, after the auction process, resources are shared among the participants according to their bids.

⁷ A number of users submit their bids only once to get a service, without knowing others bids. The highest bidder wins the service at the price he/she bids.

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
declares his/her requirements and invites bids from available contractors. Interested contractors evaluate the demands and respond by submitting their bids. The manager evaluates the bids and selects a contractor to proceed.		negotiation
		(Ganzha et al., 2010): Replicate market information to increase sustainability of the system in terms of failure
		(Gutierrez-Garcia and Sim, 2010): Analyze scalability, and heterogeneous (capability) and homogeneous (performance) resource composition capability in multi-agent based Cloud computing scenario
		(Chao et al., 2009): Group Grid nodes in terms of their respective desires to optimize system performance
		(Caramia and Giordani, 2008): Optimize system performance through negotiating distributed schedulers
		(Goswami and Gupta, 2008): Maximize success rate while minimizing time and cost constraints at different job arrival periods
		(Stefano and Santoro, 2008): Optimize QoS by adopting CDN ⁸ concept in Grid
		(Ranjan et al., 2008): Optimize QoS and resource allocation decisions by SLA based super scheduling in federated Grids
		(Dominiak et al., 2007): Implement CIC ⁹ to facilitate forming teams of different provisioning and specialization
		(Paurobally, 2007): Develop multi-agent negotiation techniques to facilitate building adaptive and autonomous Grid
		(Ouelhadj et al., 2005): Design SLA based negotiation to deal with uncertainties in resource co-

⁸ Content Distribution Network: duplicates web resources (owned by the same organization) from an origin server to different replica servers

⁹ Client Information Centre: stores information so that a user agent can interact and negotiate on pre-execution entities (e.g., price, QoS) or join an agent team

ECONOMIC MODEL	STRENGTHS	PROPOSED BY: RESEARCH FOCUS/CONTRIBUTION
		operation, system flexibility and scalability
		(Juhasz and Paul, 2002): Study system performance in terms of system size, agent load and deadline
		(Sapra et al., 2010): Propose multi-agent based hierarchical bidding mechanism to meta-schedule Grid services. Support re-negotiation among agents subject to uncertainty of job execution

Table 2.1 demonstrates the significance of different economic models in Grid computing. The papers are presented in descending chronological orders so that the reader can easily identify recent works in the context. It can be seen from the table that the Commodity Market and Double Auction are the most widely proposed models in the Grid. The Commodity Market Model can maintain market equilibrium, which is crucial for any market-oriented Grid environment. Maintaining supply and demand by regulating price behavior ensures a higher probability to deliver requested QoS as well as increased system performance. The main principle behind this model is to determine an equilibrium/spot price at which the aggregated supply and demand of the market can be diminished. For example, if demand for a resource exceeds its supply at a state, the price of that resource increases in such a way so that the demand function shifts to a point closer to the available supply. Various techniques are used to determine the equilibrium/spot price in the literature (Stuer et al., 2007). Double Auction, on the other hand is a suitable model for the Grid due to its decentralized nature and the ability to handle a large number of users. In Grid computing, users and providers are self-interested entities and appear with their individual optimization strategies. Thus, Double Auction supports them by sorting their valuations and thus expediting the trading phase without any requirement for global information.

English Auction is another compelling model in the Grid. In this model, an Auctioneer seeks to obtain the true market value of the resource that has been set for auction. Usually, users are free to increase their bids exceeding others for the resource that they are competing. When no bidder is willing to increase his or her bids anymore, then, the auction ends and the Auctioneer checks the reservation price with the last highest bid and determines the winner. This model is found to be suitable for increasing revenue, because it supports competition among users and finally selects the user who bids the highest by using iterative bidding policy. This characteristic also helps identify the demand of a resource in the market. However, English Auction, in a distributed environment may produce network congestion due to its

high communication demand. English Auction, by nature is an iterative model and hence causes too many messages to be exchanged during the auction process (Assuncao and Buyya, 2006).

By using the Bargaining Model in Grid computing, users and providers can optimize their various preference functions (time/cost). The model allows participants to negotiate on their preferences and finally to construct a satisfactory SLA. In Grid computing, the preferences could be over budget/job-execution-cost, deadline/job-execution-time or any such criteria. However, successful negotiation also depends on preference values. For example, if a user and a provider negotiate the same preference (e.g., deadline & job-execution-time) value, either the negotiation will finish with minimum optimization of the preference or it will fail. The optimization of a preference value for a user can be measured by using the utility function of the optimization criteria for that user. Researchers have already analyzed how to relax different negotiation terms during the bargaining process, so that better optimization can be achieved (Sim, 2007), (SIM, 2005). The Bargaining Model also requires a high communication demand due to the multi-round communication process, which may not be suitable when there is a large number of a user.

Proportional Share-based Auction is efficient for Grid computing because it allows for the sharing of resources according to the values represented by the users. The model also helps construct a large-scale collaborative infrastructure, which is one of the main goals of Grid computing. Because of the sharing of the same resources by multiple users, utilization for the resources increases and thus job cancellation rate decreases. However, failure to provide a sustainable sharing mechanism may cause lower QoS to be received by users or even cancellation of jobs.

Contract Net Protocol allows users to choose the appropriate service providers based on their varied requirements. The model permits users to optimize their preferences (e.g., budget, deadline) by selecting one or more appropriate providers out of multiple providers in the Grid. Providers are allowed to cooperate among themselves in order to ensure that the service for users is as per the contract. This model focuses on users' side rather than for providers to optimize their preferences. Hence, utility for users is greater than it is for providers.

Table 2.1 further incorporates different economic models proposed to be suitable for resource management in Cloud computing. The manipulation of Cloud resource prices by considering supply and demand function rather than a fixed pricing model across the network is suggested by (Buyya et al., 2009b). (An et al., 2010) and (Sim, 2010) realize the significance of automated negotiation towards a service oriented SLA establishment for the Cloud. (Buyya et al., 2010) argue for Continuous Double Auction in provisioning large-scale distributed

resource requirement, whereas (Gutierrez-Garcia and Sim, 2010) propose Contract Net Protocol to target world-wide consumer satisfaction. All these researchers realize the immense potential of these models in delivering computing as a utility for the next generation computing applications. Unfortunately, most of the existing commercial Cloud computing providers such as Amazon EC2 <<http://aws.amazon.com/ec2/pricing/>> and Windows Azure <<http://www.microsoft.com/windowsazure/pricing/>> still adopt a very basic pricing mechanism – Pay-as-you-go, to provision their resources. Therefore, the potential of economic-based distributed resource collaboration in the domain is being hindered. As identified by many researchers, Pay-as-you-go model is not incentive enough for scaling Cloud resources through aggregating resources across private/public domains worldwide (An et al., 2010), (Buyya et al., 2009b).

If we look at the Table 2.1 as a whole, we can see that different models are suitable for different scenarios. For example, Commodity Market Model is suitable for market equilibrium, admission control and Pareto optimal allocation, whereas Double Auction model shows strengths in handling a large number of users, decentralization and time efficiency. English Auction model is suitable for optimizing QoS related to jobs, and maximizing revenue for providers. Additionally, it is suitable for efficient resource allocation. The Bargaining Model and Contract Net Protocol support utility-based negotiation. Contract Net Protocol can further help in cooperating distributed resources and maintaining scalability. Proportional Share Based Auction model is suitable for economic efficiency and revenue, whereas the Proportional resource share model decreases job cancellation rate. First Price Sealed Bid Auction model provides globally efficient resource allocation.

Apart from this, Neumann et al. (Neumann et al., 2008) identify two main modes of applications: batch¹⁰ and interactive¹¹. They further distinguish Grid markets in terms of application dependency and application independency. Application dependency considers complex services, while application independency considers only physical resources. They propose different economic models for different application modes. For example, for the Batch mode within an application-dependent market, they propose either multi-attribute combinatorial auction or Proportional Share Based Auction. For the same mode within an application-independent market, they propose a bargaining protocol. However, their proposed market mechanisms for different classes are just hypotheses and not based on experimental proof. Applicability of a market mechanism in a distributed large-scale environment requires an extensive study on the mechanisms with real parameters. Until now, different economic

¹⁰ Planned execution time and expected termination time for this type of applications are possibly known in advance

¹¹ Planned execution time and expected termination time for this type of applications are usually known in advance

models have been studied by different researchers using different evaluation framework. *However, there is no research that has conducted the study of widely proposed economic models using a same framework and same evaluation metrics.* Therefore, the accuracy of their identification about different economic models in terms of a distributed Grid environment remains unclear.

In the literature, we also find that one model is being compared to another model using various criteria. Table 2.2 describes this information in detail: the first column presents different economic models, the second column lists the models that are being compared and the last column describes the different features used for comparison:

Table 2.2: A Comparative View among Different Economic Models in Grid Computing

ECONOMIC MODEL	COMPARED MODEL	FEATURES
Commodity Market (Wolski et al., 2001), (Stuer et al., 2007), (Veit et al., 2007)	English Auction (AuYoung et al., 2004)	<u>Commodity Market Model is:</u> <ul style="list-style-type: none"> • less complex for selecting a market to participate • more efficient in dividing the budget, if users want to explore different markets • more time efficient • more efficient in handling a large number of users • more suitable for price stability • more suitable for retaining market equilibrium • more suitable for increasing user provider efficiency • more scalable <u>English Auction model is:</u> <ul style="list-style-type: none"> • able to evaluate the market price
	Dutch Auction: The Auctioneer begins with a high price for a service, which is lowered until (a) some users are willing to	<u>Commodity Market Model is:</u> <ul style="list-style-type: none"> • more time efficient • more efficient in handling a large number of users

ECONOMIC MODEL	COMPARED MODEL	FEATURES
	accept the Auctioneer's price or (b) the provider's minimum demand is met.	
	Vickrey Auction , (Veit et al., 2007) Proportional resource share Vickrey Auction: This is very similar to the First Price Sealed Bid Auction model, except the highest bidder wins at the price of the second highest bidder.	<u>Commodity Market Model is:</u> <ul style="list-style-type: none"> • more suitable for price stability • more suitable for retaining market equilibrium • increases user provider efficiency <u>Vickrey Auction model is:</u> <ul style="list-style-type: none"> • more suitable for handling a large communication demand
	Flat pricing (deadline and budget based)	<u>Commodity Market Model is:</u> <ul style="list-style-type: none"> • more efficient for resource allocation • more time efficient • more scalable
Double Auction (Assuncao and Buyya, 2006), (Grosu and Das, 2004), (Tan and Gurd, 2007)	Dutch Auction, English Auction, First Price Sealed Bid Auction, Vickrey Auction	<u>Double Auction model:</u> <ul style="list-style-type: none"> • is more suitable for retaining market equilibrium • is more efficient for resource allocation • produces less broadcasting overhead • is more time efficient • is more suitable for handling a large number of users • is more suitable for price stability • is more suitable for increasing user and provider efficiency

ECONOMIC MODEL	COMPARED MODEL	FEATURES
		<ul style="list-style-type: none"> • is more efficient for global resource allocation <u>English Auction model is:</u> <ul style="list-style-type: none"> • more suitable to optimize QoS • more efficient for maximizing revenue for providers • better for the economic efficiency
	Proportional resource share	<u>Double Auction model is:</u> <ul style="list-style-type: none"> • more suitable for price stability • more suitable for retaining market equilibrium • more suitable for increasing user and provider efficiency
	Commodity Market	<u>Double Auction model is:</u> <ul style="list-style-type: none"> • more time efficient • more decentralized
English Auction (multi attribute) (Xing and Lan, 2009), (Schnizler, 2008)	Flat (fixed) pricing, Unit pricing	<u>English Auction model (multi attribute) is:</u> <ul style="list-style-type: none"> • more suitable to optimize QoS • better for the economic efficiency
	English Auction	<u>English Auction model (multi attribute) is:</u> <ul style="list-style-type: none"> • more suitable for considering combinatorial bids • more suitable to optimize QoS
	First Price Sealed Bid Auction, Vickrey Auction	<u>English Auction model (multi attribute) is:</u> <ul style="list-style-type: none"> • more suitable to optimize QoS • more efficient for maximizing revenue for providers • better for economic efficiency
Proportional	English Auction, Vickrey	<u>Proportional Share Based Auction</u>

ECONOMIC MODEL	COMPARED MODEL	FEATURES
Share Based Auction	Auction	<u>model is:</u> <ul style="list-style-type: none"> • more scalable • more suitable for efficient resource allocation
Proportional resource share (Sherwani et al., 2004)	PBS (Portable Batch System), FIFO (First in First out)	<u>Proportional resource share model is:</u> <ul style="list-style-type: none"> • suitable for minimizing job cancellation rate

Table 2.2 explains different economic models in the Grid and compares them with one another in terms of various criteria, such as market equilibrium, handling a large number of users and user and provider efficiency. One of the widely proposed models (according to Table 2.1), the Commodity Market Model is shown to be better here than the English Auction model in terms of having less complexity in selecting a market to participate, handling a large number of users and maintaining market equilibrium. The Commodity Market Model is more efficient in managing time and handling a large number of users than the Dutch Auction model. The Commodity Market Model is also found to be more suitable than the Flat pricing model for resource allocation efficiency, time efficiency and scalability. Another widely proposed model, the Double Auction, is better than the Dutch, English, First price sealed bid and Vickrey Auction models in terms of maintaining market equilibrium, broadcasting overhead and achieving user and provider efficiency. The Double Auction has better market equilibrium, user provider efficiency and price stability compared to Proportional share model. The Double Auction is also better than another popular model, the Commodity Market, in terms of time efficiency and decentralization. The English Auction (multi attribute) model is better than the Flat pricing and Unit (fixed) pricing models in terms of QoS and economic efficiency. The multi attribute English Auction model is better than the single attribute English Auction model in terms of QoS optimization and consideration of combinatorial bids. The English Auction model (multi attribute) is better than the First Price Sealed Bid Auction, Vickrey Auction model and Double Auction models in terms of QoS, revenue and economic efficiency. Even though the Commodity Market Model is one of the most widely proposed models, it is less efficient to evaluate the true market value of a resource compared to the English Auction (multi-attribute) model. The Proportional Share Based Auction model is better than English and the Vickrey Auction models in terms of scalability and resource allocation efficiency. The model is also found to be suitable for minimizing job cancellation

rate compared to the PBS (Portable Batch System) and the FIFO (First in First out). However, these comparisons are either based on theoretical backgrounds of the models or identifications made by different researcher but not based on any experimental proof. ***Therefore, the identification of regions of strength for individual models based on quantitative analysis is still an open issue.***

Analyzing Table 2.1 and Table 2.2, it can be seen that even though the economic models play a significant role in Grid computing, one model is not suitable for all the scenarios in Grid environment. In addition, due to the dynamic nature of the Grid, the application of a single model might not be able to harness the full potential from Grid. ***However, in the literature, there is no such mechanism for combining two or more economic models to utilize the strengths of multiple economic models in different scenarios.*** For example, there is no mechanism that uses Commodity Market Model to maintain market equilibrium and, then, switches to an auction model to bring more profit for providers. However, managing more than one economic model in a highly dynamic and heterogeneous environment poses other challenges, which we will explore in Chapter 7. The following section presents a summary and describes the popularity of different economic models in the literature.

2.8 Discussion

This section provides a brief discussion on the available economic models in the Grid and summarizes some open research issues in the field. Since the initiation of Grid computing, a number of economic models have been proposed to deal with the heterogeneity and utility based computing. However, not all models are suitable for all scenarios in the Grid environment. Through numerous research studies, experiments and simulations, only a few of the models have demonstrated their effectiveness in Grid environment. In addition, one model is different from another due to its distinct features and objectives of usage. Figure 2.5 summarizes different economic models that have been proposed over the years for usage in the Grid (Legend: from left to right). Figure 2.5 has been generated using the information from Table 2.1.

The adoption of economic based approaches started primarily at the beginning of this decade. Thus, Figure 2.5 shows papers from the year 2000 onwards. Figure 2.5 illustrates that the significance of economic models is increasing every year. The adoption of Commodity Market Model started in 2000 and continued through 2011 with a gap in 2002 and 2003. Many papers on Commodity Market Model were published in 2007, which is quite recent. The Commodity Market Model has the potential of maintaining equilibrium between supply and

demand, and it is economically efficient. This provides incentive to resource providers to contribute their resources in the Grid. The Double Auction model is another widely proposed model since 2003. In 2009, Double Auction was the most frequently proposed model. Double Auction has become more popular, especially due to its ability to handle a large number of participants, while producing less communication overhead. However, Double Auction is not precise economically efficient compared to the Commodity Market Model or English Auction model. The English Auction model has achieved popularity in the Grid due to its efficient resource allocation and economic efficiency. Thus, the English Auction model has been continuously proposed since 2004. However, the English Auction model is not suitable for handling a large number of users and is not decentralized. The Bargaining Model has been proposed since 2005 and has become popular, because it supports negotiation among Grid participants, which assists to form utility based computing. In spite of the utility-based negotiation, the Bargaining Model is not precise economically efficient and it produces high communication overhead. The next three models, Proportional Share Based Auction, proportional resource share and First Price Sealed Bid Auction, have not achieved much popularity in the Grid. Only a few papers have proposed these models across the years. Among these three models, proportional resource share is discussed the most because it supports fairness in resource sharing. Finally, Contract Net Protocol has some potential to meet the vision of large-scale resource collaboration, because it supports cooperation among different Grid organizations to optimize resource QoS. However, co-operation among the organizations would be a complex undertaking in the Grid due to their distinct administrative rules and policies. In addition, the Contract Net Protocol provides incentives to users through optimizing their utility entities, such as time, QoS, and budget, but it does not provide sufficient motivations for providers to achieve their goals. In 2008, the maximum number of papers proposed the Contract Net Protocol model.

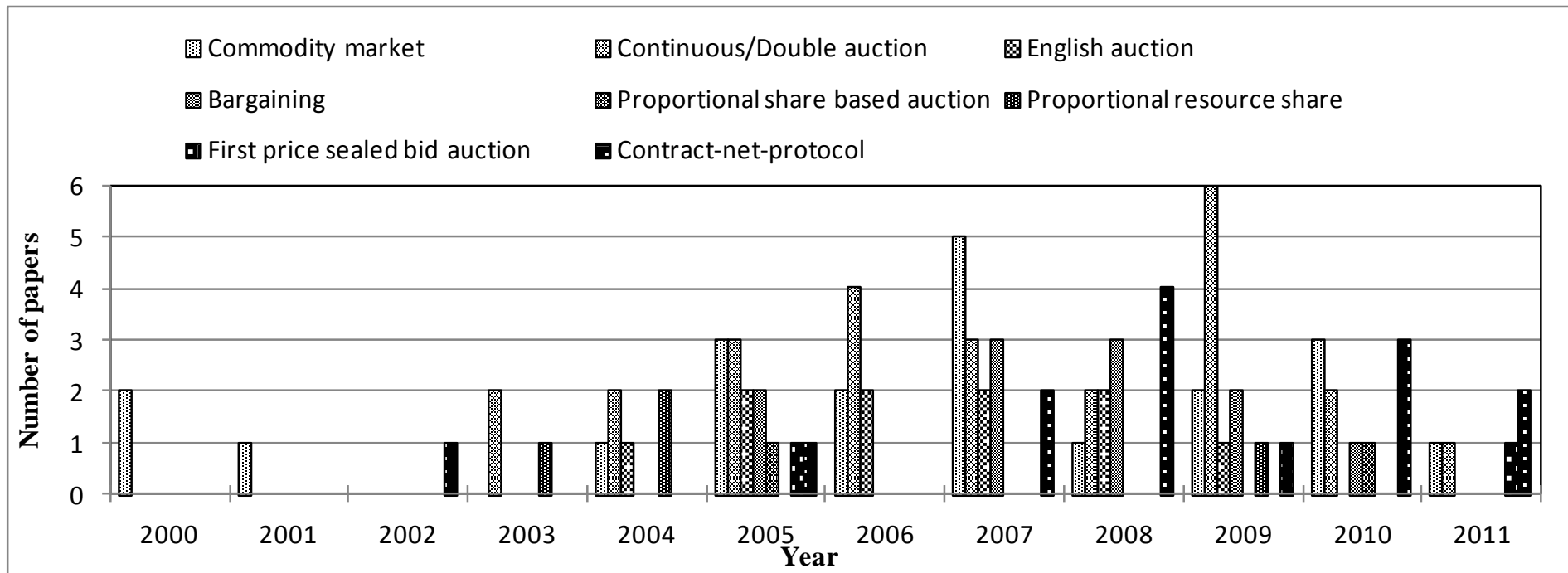


Figure 2.5: Adoption of Economic Models as Per Year

Based on the frequency of publishing different economic models as appeared in Figure 2.5, we detect five most frequently proposed models - Commodity Market, Bargaining, English Auction, Continuous Double Auction and Contract Net Protocol. This thesis focuses on analyzing the performances of these five models. Moreover, from Table 2.2, we realize that there are two groups of researchers – one talk on favor of Commodity models and the other talk on favor of Auction models. The two models – Commodity Market and Bargaining fall under the former type. On the other hand, the other three models – English Auction, Continuous Double Auction and Contract Net Protocol fall under the Auction type. Therefore, we dedicate two different chapters – Chapter 4 and Chapter 5 to study the efficacy of Commodity and Auction models. Auction models can further be classified into two types – One-sided and Two-sided – depending on the structure of the auction. One-sided auction supports one-to-many negotiation. For example, this type of auction could happen between many users and one provider such as English Auction. Contract Net Protocol is not directly an auction type. However, based on the workflow of the protocol (with one user and many providers), we accommodate it into the auction type. Again, two-sided auction allows flexibility to both users and providers i.e., auction happens among many user and many providers such as Double Auction. Based on these characteristics, and for a better understanding, a classification of these models is provided in Figure 2.6.

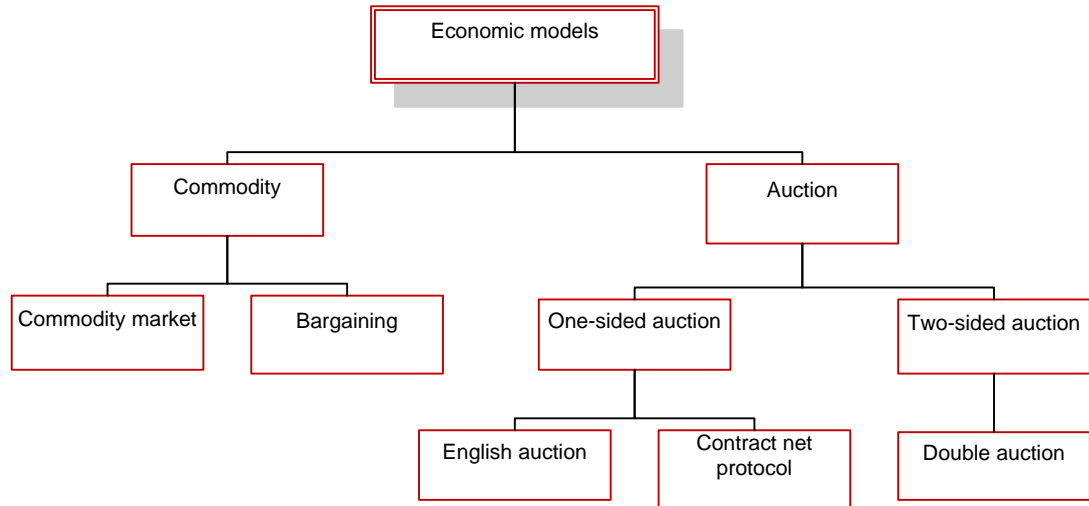


Figure 2.6: Major Economic Models in Grid

The Commodity market, where price works to regulate the behavior of market entities as the main principle, can be adopted to satisfy the entities (Cheliotisy et al., 2003), (Stuer et al., 2007). However, price volatility in such a market is also anticipated as a harmful catalyst, because it might degrade the users' QoS (Bossenbroek et al., 2009). To avoid price volatility, based on current market conditions different hedging strategies are proposed by constructing hedging portfolios by the contract issuing service for individuals. Therefore, the organization

of different portfolios obviously deserves meaningful consideration due to the higher level of uncertainty involved in the system. On the other hand, different auction theories are suitable for distributed environments. However, auction models cannot always guarantee market efficiency and it thus become difficult to maintain consistency in supply and demand. It has also been identified that an individual auction model is not suitable to construct a precise and complete solution for a large scale distributed system (Cheliotisy et al., 2003).

Though the demand for economic models increases over time in Grid computing, an individual model cannot provide all the benefits in different scenarios. Hence, the following issues are still open and could be addressed as:

- To date, different economic models in Grid computing have been studied by different researchers using different evaluation platforms. No research has considered the study of widely proposed economic models using a common platform and common performance metrics to ensure consistency in the evaluation process.
- Until now, one economic model has been compared with another based on others identification or theoretical background of the models. However, no research considers a comprehensive investigation and comparative analysis on the performances of widely proposed economic models for Grid resource management.
- In spite of the suitability of different economic models in different scenarios, no research attempts to identify the regions of strengths of individual models based on quantitative measures.
- No research has identified the potential of utilizing the strengths of different economic models in different scenarios through switching between the models towards developing an optimization-based Grid computing framework.

2.9 Conclusions

Grid computing was initiated with the promise of delivering a cost-effective and standard computing. The distributed, heterogeneous and dynamic natures of the Grid resources impose challenge in seamless collaboration of the resources. Economic-based approaches provide sufficient motivation to achieve this collaboration. Economic models also provide standards to manage the operation in a Grid environment. However, different economic models follow different principles and users and providers in the environment must obey the principles. We identified the key performance metrics that can be used to evaluate the performances of various economic models in Grid computing. Our survey on existing economic models in Grid computing identified that different economic models are suitable for different scenarios.

Therefore, among providers, there is an ambiguity on choosing a model for maximizing their profit. A comparative approach among different economic models further emphasized the inability of selecting a single model for the Grid. Finally, we identified some open research issues that could be considered to deliver a sustainable economic-based Grid computing framework. We identified the need of an evaluation framework suitable for investigation and comparison among the performances of different economic models in Grid resource management. We further shown the possibility of identifying the domains of strengths of different models in terms of quantitative measures and the opportunity of developing an optimization framework based on the domains of strengths.

To deal with the first issue, in the next chapter, we describe the design and development process of such a framework, which can be used to analyze the performances of different economic models in the Grid. We identify the five most widely proposed economic models in the Grid – Commodity Market, Bargaining, English Auction, Continuous Double Auction and Contract Net Protocol. To investigate and analyze the effectiveness of these models are agenda of this thesis. Chapter 4 and Chapter 5 are dedicated to dealing with the second issue. The third and fourth issues will be discussed in Chapter 6 and Chapter 7 consecutively.

Towards Developing an Evaluation Framework for Economic-based Grid Resource Management

To deal with the first research issue addressed in the previous chapter, this chapter incorporates the materials for developing a framework suitable for assessing the performances of different economic models in Grid resource collaboration. In this regard, the chapter discusses the necessary tools, characteristics of the Grid entities, and simulation space for the framework.

3.1 Introduction

The performance of the execution environment in Grid computing is significantly influenced by resource scheduling/management strategies. Therefore, the scheduling strategies must be evaluated before they can be deployed in the real world. As such, an evaluation framework needs to support components that facilitate to analyze the strategies as much as possible. In this chapter, we provide such a framework suitable for evaluating various economic-based scheduling strategies in the Grid.

The entities in a Grid framework can be characterized depending on their distinct behaviors. Grid users can typically be described in terms of their applications. The description of budget and deadline parameters in the applications further enables to understand the values of the applications. Grid broker is an entity in the environment that typically works on behalf of a user. A broker performs all the crucial tasks such as resource discovery, resource selection through negotiation with the resource nodes, job-submission on selected nodes and finally obtains results from executed-nodes. On the other hand, resources are characterized based on their capability, availability and reliability.

To evaluate the performances of different economic-based resource scheduling strategies, a more comprehensive and reconfigurable economic-based framework is required compared to traditional systems. Simulation-based evaluation is a cost-effective and quick method to justify the efficiency of Grid systems comprising of thousands of resource nodes and users. For hardware-based evaluation, considerable amount of money needs to be spent to install

servers. In addition, managing so many active users could be extremely difficult in this environment (Buyya and Murshed, 2002, Vanmechelen et al., 2008). Realizing this difficulty, several simulation toolkits depending on their distinct purposes, values and extensibility are proposed for the Grid (Buyya and Murshed, 2002, Casanova, 2001, Klusacek and Rudova, 2007). The widely discussed discrete-event simulation toolkit – GridSim has been used to develop our framework.

The performance in a Grid environment further varies due to the dynamic nature of Grid entities. This nature explains the behavior by the entities for leaving and joining in a Grid environment arbitrarily; therefore, the evaluation framework needs to account for this dynamic nature through suitably defined parameters. Our simulation space defined in the framework supports such parameters. We show the significance of using Monash Sun Grid (MSG) for our simulations. MSG provides high performance cluster resources for high scale experiments. The details of MSG can be viewed at <http://www.monash.edu.au/eresearch/services/mcg/msg.html>.

The major motivation of our framework is driven by the evaluation of different economic models using a common platform. Using a common platform not only brings consistency in different evaluation processes but also facilitates to conduct a comparative analysis among the performances on various models. The parameters and their statistical significance in our framework support the fairness among the evaluations of different models.

The rest of the chapter is organized as follows. Section 3.2 presents and contrasts among different simulation toolkits for the Grid. The GridSim and its working behavior have been discussed in Section 3.3. Section 3.4 describes the key entities played in our framework. The definition of resources and simulation space that the framework incorporates has been discussed in Section 3.5. The section further presents the significance of the framework in terms of a fair evaluation among different economic models. Section 3.6 gives an overview of how we have conducted our simulations on MSG. Section 3.7 concludes the chapter.

3.2 Tools for Simulating Grid Environment

This section gives an overview of different simulation tools for Grid computing. As mentioned earlier, simulation-based approach is effective in assessing the quality of a large-scale resource management system. Grid simulators provide the necessary tools for this assessment. Several simulators have been proposed for evaluating Grid environments. They can be differentiated based on their distinct purposes, values, and extensibility. Table 3.1 presents and contrasts several simulators proposed for Grid computing.

Table 3.1: A Comparative Approach among Different Grid Simulators

Properties for comparison	GridSim	SimGrid	GangSim	GES	Alea
Economic-based Scheduling	√	×	×	√	×
Network Facility	√	√	×	×	GridSim-provided
Programming Platform	Java	C	Perl	Java	Java
Job-scheduling	Centralized and Decentralized	Centralized and Decentralized	Centralized	Centralized and Decentralized	Centralized
Parameter Sweep	×	√ (via external parser)	×	√	√
Entity Communication Standard (I/O model)	√	×	×	×	GridSim-provided

GridSim: It is a widely discussed and adopted toolkit for simulating Grid environment (Buyya and Murshed, 2002). It is a Java-based discrete-event simulation toolkit, which possesses multilayer architecture starting from various resource management components (such as job-management, resource allocation policy and advanced reservation), Grid broker and applications. GridSim runs on top of SimJava library, which provides the basics of discrete-event simulation. The toolkit also supports the Grid entities in a simulated network and provides the necessary characteristics for network architecture, which makes the toolkit as close as possible to a real Grid scenario. However, GridSim does not support multiple-runs which is crucial when a simulation needs to be conducted over different sets of parameters automatically (parameter sweep) to evaluate the sustainability of the proposed mechanism. The drawback of GridSim can be viewed at <<http://www.buyya.com/gridsim/doc/faq.txt>>. In current GridSim, one needs to conduct this manually for different sets of parameters, which is a substantial drawback of this tool. Additionally, GridSim supports testing and analyzing economic-based resource management scheduling. Finally, the Entity Communication Standard describes the model for communication between two entities. GridSim uses the input-output method for communication among the entities, which is realistic.

SimGrid: It is another widely discussed Grid simulator developed in the University of California San Diego (UCSD). It is a C language-based simulation toolkit especially designed for simulating distributed applications in heterogeneous and distributed environments (Casanova, 2001). It supports both centralized and decentralized approaches of scheduling user jobs. It models time-shared resources and allows developers to transfer the simulated scenario into the real world Grid without code modification. The tool does not support economic-based resource management. Though it provides network facilities, unlike the GridSim methods, it does not use input-output method for communication. The tool uses store and forward method for passing messages through the network. Even though, the SimGrid supports multiple-runs, the procedure is not feasible, which can be viewed at <http://simgrid.gforge.inria.fr/faq.html>. In order to achieve multiple-runs, one needs to configure his/her own parser using FileXML. Afterward, the parser needs to be bypassed to the original code using some unique functions (SURF) from the SimGrid library.

GangSim: The toolkit is developed under the department of Computer Science, University of Chicago. It mainly models job-submission, execution monitoring and usage-policy structures. It adopts the concept of Resource-Site that typically comprises of several resource-nodes and VO's (Virtual Organizations), which aggregates users (Dumitrescu and Foster, 2005). It uses a centralized scheduler that queues user jobs and selects suitable Resource-Site for each job. The toolkit does not support economic-based resource management and network infrastructure. The library is developed using Perl language.

Grid Economics Simulator: The Grid Economics Simulator or GES is designed especially for the simulation of economic-based Grid organizations (Vanmechelen et al., 2008). The tool is written in Java language and provides major market models including Commodity and Auctions except Bargaining protocols. As there are scalability issues in GridSim and SimGrid due to their multithreading model, GES uses single thread-based simulation. However, it does not provide network services. It supports advanced reservation of Grid resources through Future Market mechanism. It uses a centralized scheduler to deal with noneconomic based scheduling policies.

Alea: Alea is an extension of GridSim simulator (Klusacek and Rudova, 2007). It is mainly envisioned to evaluate different scheduling strategies such as FCFS (First Come First Serve), Easy-Backfilling, EDF-Backfilling (Earliest Deadline First) or any user-defined scheduling strategies. It allows centralized scheduling services, which is not suitable for economic-based scheduling. Instead of using GridSim provided SimJava library, Alea uses a modified version of it to supports multiple-runs. As the tool is based on GridSim, it can use all other GridSim provided functions.

Based on extensibility, popularity and finally our economic-based requirements, we choose the GridSim simulator for our evaluation. The working principle of GridSim is described in the following section. Besides, we also use Alea provided SimJava to support multiple-runs. A detailed motivation for using this SimJava is discussed in sub-section 3.5.1.

3.3 GridSim: A Discrete Event Simulation Toolkit

This section presents an architectural overview and working principle of GridSim, which will help to understand the subsequent developments of different economic models. The primary goals of the simulator are described below (Buyya and Murshed, 2002);

- ❖ To analyze the effectiveness of various resource management techniques based on computational economy through simulation
- ❖ To support necessary components for allowing simulation with large-scale distributed heterogeneous resources
- ❖ To provide sophisticated tools for supporting fine-tuning of the simulation to deliver quality analysis

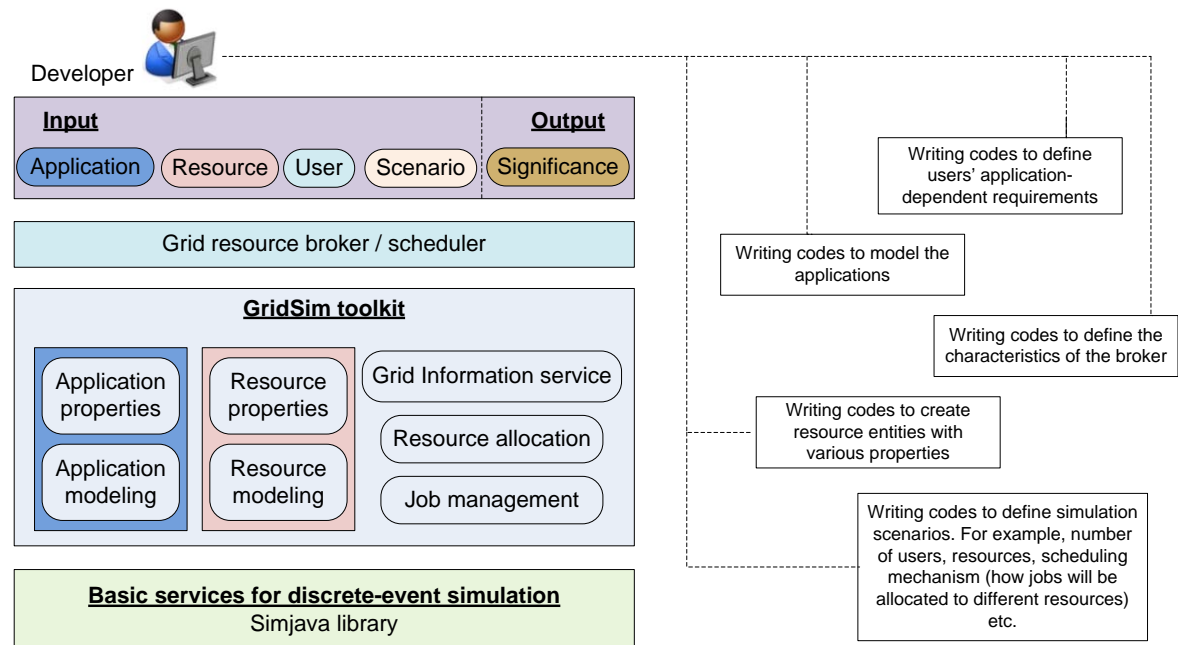


Figure 3.1: A Typical Overview of GridSim Simulator – From a Developer’s Point of View

3.3.1 Applications and Resources

Figure 3.1 illustrates a general overview of GridSim toolkit. In this sub-section, we describe the toolkit from a developer’s point of view. To this respect, it discusses application modeling, resource characteristics and significance of broker in a Grid. The Grid Application describes the original task that needs to be executed on different heterogeneous resources. Users can

define various constraints such as deadline, budget and information such as specific software packages required by the applications. Different users may execute different applications with different constraints at the same time. Grid applications are typically comprised of a large number of tasks. A task can contain a number of instructions, which need to be executed on a computing element/resource. Based on the relationship and dependency among tasks, Grid applications can be categorized into three types:

- **Bag-Of-Tasks:** This kind of application consists of multiple independent tasks requiring no communication among the tasks (Cirne et al., 2003). The final output is subjected to the completion of all individual tasks. Such applications are best suited for the scenarios such as data mining, parameter sweep simulations and computational biology (Cirne et al., 2003).
- **Message Passing Interface:** For such applications, tasks are interdependent. Therefore, they need to communicate with each other during the execution. The inter-task communication process is developed using Message Passing Interface libraries. The availability of the computing elements for different tasks is significant due to the interdependency among the tasks (Nascimento et al., 2005).
- **Workflow:** This type of applications can be represented as a Directed Acyclic Graph, where different tasks can be represented as nodes in the graph and the task-dependencies can be drawn as the directed arcs among the nodes (Ramakrishnan, 2008). The task, which has no parent task, is referred to as entry task and the one that does not have any child task is known as exit task. A child task must wait until all of its parent tasks finish their execution. Simulations in Bio-informatics, weather forecasting, astrophysics could be some of the examples of this type of application.

For our framework, we only consider the Bag-Of-Tasks application, because it simplifies the overall simulation process due to its task-independent characteristic. Moreover, this kind of applications is more suitable for execution in a Grid environment (Cirne et al., 2003). GridSim toolkit provides the necessary services for application modeling with the given properties such as application ID and number of processing elements (PE) required.

Grid resources typically vary in terms of their architecture, performance and allocation policy. A single computer, a cluster of computers or even a supercomputer could be a part of the Grid resources. Grid resources could also include some specific software packages or scientific instruments around the globe. Thus, considering the heterogeneity, various capability and allocation policies are crucial, while modeling Grid resources. GridSim supports modeling resources suitable for a distributed environment. Depending on the execution on a resource, resource allocation policy can be categorized mainly into two policies – time-shared and space-

shared. In a time-shared policy, the resource immediately starts executing all the jobs that are arriving or more precisely, all the available jobs share time on the resource. Therefore, the completion time for a job increases as the number of jobs being assigned to the resource increases. On the other hand, in space-shared policy, a job will start immediately if there is a free processing element (PE); otherwise, the job will wait until the resource becomes available. Additionally, the GridSim supports designing one's own resource allocation policy. For example, currently it supports modeling advanced reservation for resources. In our simulation, we use only space-shared policy throughout to simplify the evaluation process.

Grid resource broker works as a mediator between a user and many resources. The broker is also referred to as scheduler, because it schedules jobs on the resources. The broker performs all the crucial tasks for the user, such as discovering appropriate resource(s) for executing the application, negotiating with resources to choose suitable resource(s), submitting the application to the resources and getting the results back from the resources. The Grid Information Service (GIS) in the GridSim toolkit is designed to assist the brokers by providing available resource information. Whenever a new resource entity is created, it must register with the GIS. The GIS works similar to DNS (Domain Name Service) used by World Wide Web. Once a broker submits its job to a resource, the broker can keep track of the progress of the job using the Job Management Service (Figure 3.1). More details on the properties of Grid applications, resources and brokers are explained in Section 3.4. Communication is a vital part for the Grid entities to achieve their individual interests from the environment. Therefore, in the following sub-section, we describe the communication model used in GridSim toolkit.

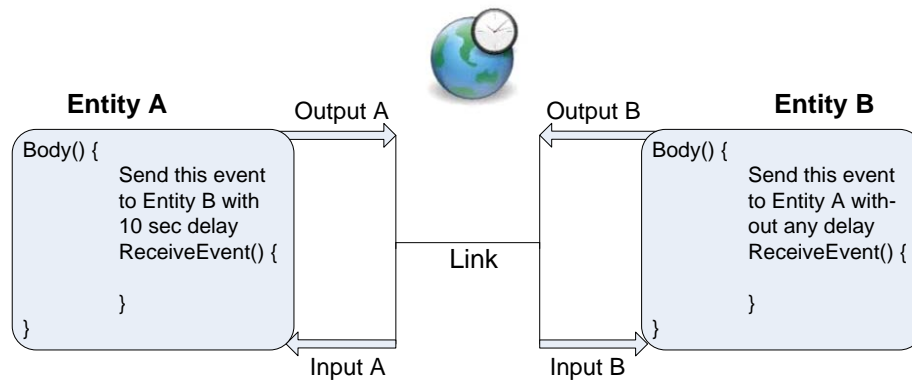


Figure 3.2: A Conceptual View of Communication between Two GridSim Entities

3.3.2 Communication Model

This sub-section provides a brief understanding of how different entities send and receive messages in GridSim. As noted before, GridSim toolkit runs on top of SimJava library. The library is encoded in java and responsible for the management of the overall simulation

process. The model (user or resource) with the capability of sending and receiving messages must be inherited from *Sim_entity* in SimJava. Thus, such models are known as entities or more precisely run-able entities in the simulation. The behavior or characteristics of an entity must be defined within the entity's *body ()* method. Each entity possesses a thread, if it is inherited from *Sim_entity* class. The messages in SimJava library are known as events and *Sim_event* class handles the management of all the simulation events. GridSim creates two additional entities during the creation of each entity (using *Sim_entity*) to facilitate the communication process – *Input* and *Output*. Every entity, during its communication with any other entity, uses its *Input* class to receive events and *Output* class to send events. To facilitate the parallel and concurrent communication, and estimate a transparent communication delay, GridSim extends the *Input* and *Output* classes also from *Sim_entity* class. Thus, both classes will have their individual threads. How the communication process is conducted between two different entities is depicted in Figure 3.2.

If two entities in GridSim would like to talk to each other, they need to establish a link using their *Input* and *Output* classes, before they start their communication process. GridSim's networking tools also support modeling router, packet and packet scheduler in the link. However, in our simulation, we use an ordinary Grid network where each entity can form a link to another entity for communication. We do not deploy the additional network components in the link. GridSim supports the specification of a particular amount of delay to schedule the events. Therefore, the events in a simulation are called time-driven, which further enables a developer to facilitate the event-management process in the simulation. For example, if a set of job-requests is sent to the resources, which are yet to be registered with GIS, there will be a conflict in the simulation; therefore, it is better put some delay (suitable time-period for the resources to be registered with GIS) before sending any job-request to the resources.

3.4 Properties of Economically-inspired Grid Entities

Grid entities are typically regulated depending on their own objectives. However, economically inspired Grid entities are more complex and more self-interested compared to noneconomic entities. As such, simulating such individually rational entities requires proper definition of their different properties. These properties will then play a role in defining their characteristics in the simulation. In this section, we describe the properties for users, brokers and resources, which are the three key roles in our simulation.

3.4.1 User

As an application represents a user, a task can represent an application. Such tasks are named “Gridlet” in GridSim toolkit. Tasks are created using Gridlet objects. A Gridlet is a package, which contains all information about the Gridlet. One Gridlet can be differentiated from another using the following properties;

- *Identity*: Each Gridlet has a unique ID
- *Length*: The processing length (in Million Instruction) of the Gridlet
- *Deadline*: The maximum execution time (in Simulation second) the Gridlet can allow
- *Budget*: Capital (Grid\$) available to spend on the execution for the Gridlet
- *Input-data*: Input file size (in Byte) of the Gridlet
- *Output-data*: Output file size (in Byte) of the Gridlet (once the processing the Gridlet is finished)

Once a Gridlet is created, the respective user sends it to its corresponding broker along with the properties. The broker, then, tries to finish executing the Gridlet on the Grid resources by maintaining the constraints as defined by the user.

3.4.2 Broker

A broker performs all the crucial tasks on behalf of the user. A complete life cycle of a broker is presented in Figure 3.3.

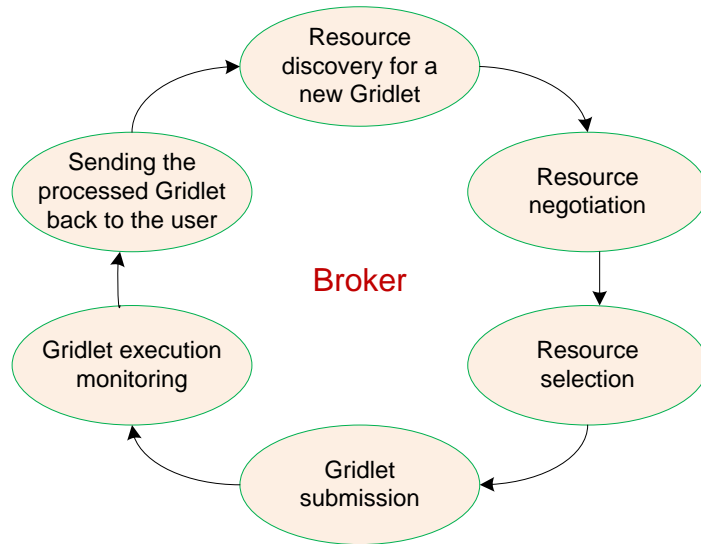


Figure 3.3: A Broker's Life-cycle on Gridlet Execution

A unique ID is assigned to each broker. At the first phase of its lifecycle, it collects resource availability information from GIS. Afterward, it starts communicating with resources and negotiating based on constraints (Figure 3.3). For example, it confirms with a resource

whether the resource can process the Gridlet within the available budget and deadline. However, the negotiation process varies over economic models, which we will see in the subsequent sections. The broker may find its suitable resource among the available resources or may fail. In terms of a failure, it sends the unprocessed Gridlet to its user. If the broker finds a suitable resource, it selects the resource to submit the Gridlet. The execution of the Gridlet on the resource can be monitored, if the broker wishes. Once the Gridlet is processed, the resource sends the Gridlet back to the broker along with the results. The broker, then, sends the processed Gridlet back to its user. In our simulation, a single broker deals with a single Gridlet. Nimrod is an example of real world Grid broker (Buyya et al., 2000b).

3.4.3 Resource

Grid resources are typically referred to as resource-nodes or only nodes. Each resource-node has several properties to distinguish it from other nodes.

- *Identity*: The node ID
- *Operating System*: The operating system, on which the node is running. For example, Linux, Windows
- *Architecture*: The architecture of the node (Apollo DN, Sun Ultra)
- *Machine List*: Number of machines (computers), of which the node is comprised. A machine further consists of one or more processing elements (PE). The performance of PE is denoted by MIPS (Million Instruction Per Sec) rating
- *Allocation policy*: The scheduling policy by the node to schedule different Gridlets on it
- *Cost*: The cost of using the node per second

We explain how the cost and time are calculated for a Gridlet by a node, in the following chapter.

3.5 Resource Configuration and Simulation Methodology

This section explains the simulation scenarios, different simulation parameters and their statistical significance. The section further describes the fairness in evaluation among different models.

3.5.1 Grid Scenarios and Simulation Space

Grid is dynamic in nature, i.e. Grid nodes can join and leave anytime during the Grid's lifecycle. In addition, market mechanism in a Grid environment is significantly influenced by the number of users and resources. Therefore, to reflect the impact of Grid dynamics, one needs to take into account a large number of possibilities while defining parameters for a

simulation. However, existing literature only considers a limited number of users and resources, and varies these numbers in large steps (e.g., 5, 10, 15...), which is not comprehensive enough to model the performance of economic models. Realizing this, we consider a parameter space consisting of a number of Gridlets and nodes, which is a 100×100 mesh of (s, d) . This takes into account all possible scenarios when the maximum number of Gridlets or nodes is 100. Here, s refers to the number of nodes and d refers to the Gridlets. As we are considering every possible scenario in the simulation space, we, at present, limit the simulation space within 100 by 100. We would like to evaluate our work for an extended simulation space in the future. If we depict our simulation space, it would look like Figure 3.4. A value along the Z-axis represents the performance obtained for a particular evaluation metric and a cell in the space. Thus, analyzing the effectiveness of an economic model for different supply-demand variation becomes feasible.

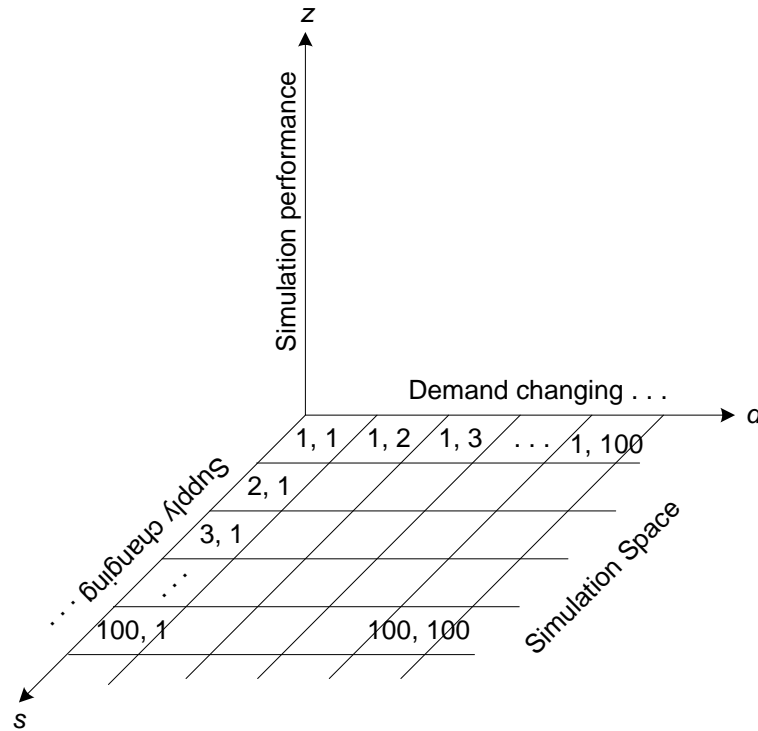


Figure 3.4: Simulation Space designed for Our Framework

Execution of an application on real Grid resources would look like Figure 3.5. Grid typically uses dispatcher (Nimrod-G Dispatcher) to break-down an application into several Gridlets to facilitate the deployment of the Gridlets on multiple resources (Buyya, 2002). Individual Gridlets is then submitted by their respective brokers to resources for execution. We have already described the parameters for a typical Grid resource in sub-section 3.4.3. A resource can be used to execute one or multiple Gridlets at a time. In a Grid computing

environment, each resource can be regarded as a cluster; because the resource typically consists of several machines and each machine could possess several PEs.

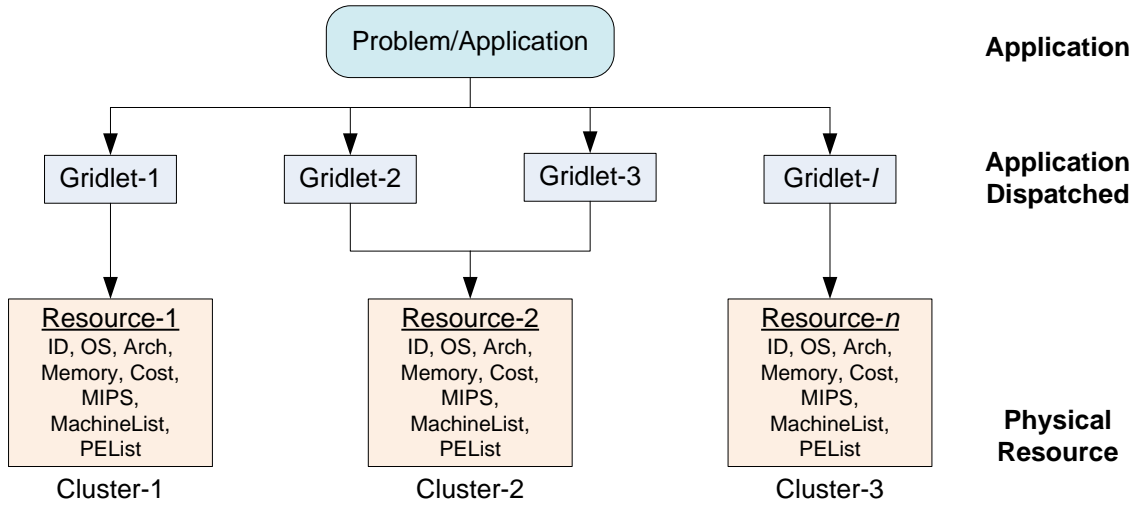


Figure 3.5: Application Execution Environment in a Real Grid

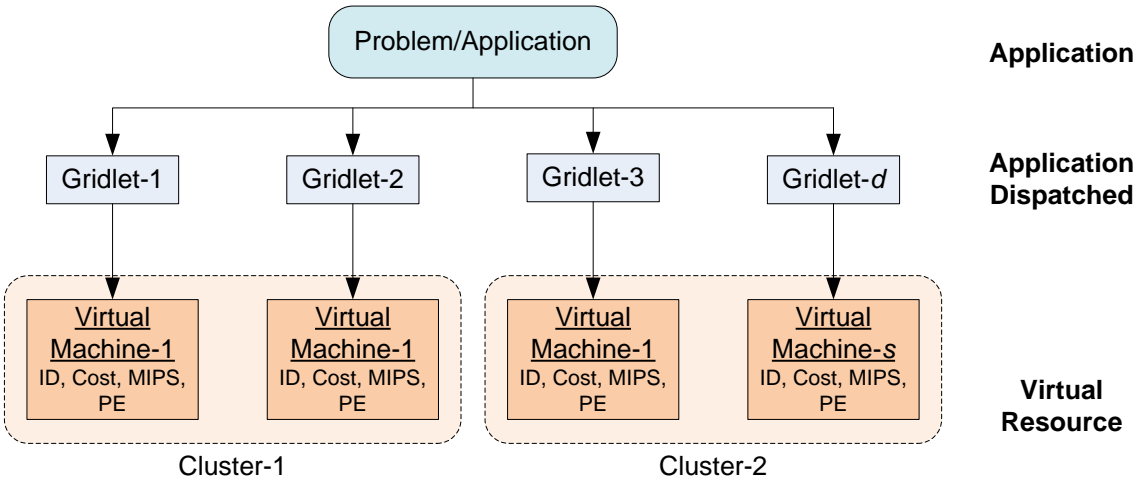


Figure 3.6: Application Execution Environment in Our Simulated Grid

Figure 3.6 presents the execution environment in our simulated Grid. In our simulation, we assume that a suitable dispatcher has already dispatched the application. We further vary the number of Gridlets in the application up to d (100 in our case). To be consistent with this, we assume the concept of virtual machine rather than real resource. We consider that a single resource works as a single virtual machine and one virtual machine can be used to execute only one Gridlet so that like Gridlet, we can vary the total number of machines up to s (100 in our case). In addition, currently GridSim supports processing a single Gridlet only on a single PE, which is stated in the GridSim Documentation at <http://www.buyya.com/gridsim/doc/faq.txt>. Thus, we advise each virtual machine with one PE. This helps us to understand the impact of changing supply and demand in the environment in a more precise way. Therefore, each virtual machine has its own ID, MIPS

rating, a PE and cost of using per second. We use constant values for other parameters because of their irrelevancy in our simulation. Several virtual machines could be under a single cluster/resource in reality. For explanation purpose, we continue referring to the virtual machines as resources/nodes.

GridSim rests on SimJava, which is a java-based discrete event simulation library. As mentioned earlier, GridSim provided SimJava, by default, does not support *multiple-runs*. *Multiple-runs* is essential for scenarios when one experiment needs to be conducted over hundreds or thousands of different parameter settings such as ours. In our case, we are experimenting each model with 10^4 ($=100 \times 100$) different set of simulations. For this reason, we use a modified version of SimJava provided by Alea-2, which is a GridSim based job scheduling simulator designed to support *multiple-runs* (Klusacek and Rudova, 2007). To let the *multiple-runs* work out, *setInComplete(true)* method of *Sim_System* class is invoked from the main program each time the experiment finishes with a setting. This will refresh the SimJava library; however, the parameters that have been using to define the entities' behaviors outside of the library also need to be refreshed to let the simulation work properly. The following subsection describes different simulation parameters and their experimental justification.

3.5.2 Simulation Parameters and Its Statistical Significance

Table 3.2 presents the parameters we have used to conduct the simulations. This configuration is applicable for all the five models we are dealing. Some parameters are only applicable for some models. For example, the number of rounds (θ) is only applicable for Bargaining and English Auction models.

Table 3.2: Resource Configuration

Parameters	Values
Number of rounds (θ)	10
Gridlet arrival time	(5, 20)
MIPS rating for a node (in MIPS)	(350, 450)
Cost-per-sec for using a node (in G\$) (<i>cost</i>)	(1, 2)
Gridlet length (in MI)	(1000, 10000)
Gridlet deadline (in simulation sec.) (<i>dl</i>)	(12, 22)
Gridlet budget (in G\$) / <i>max-budget</i>	(32, 45)
Gridlet budget (in G\$) / <i>min-budget</i>	(15, 18)
<i>min-bid</i> / <i>min-node-bid</i>	0
<i>max-bid</i> / <i>max-node-bid</i>	45

The behavior of Grid entities is stochastic; hence achieving same performance at different times by a model is not possible. To minimize this uncertainty, we test our models by using five different distributions (samples) and present only their averages.

The impact of economic models on Grid computing is extensive; hence, the complete evaluation for a model is almost impossible. However, it is better analyzing the performances for as many scenarios as possible. There are several metrics to evaluate the strengths of the economic models in Grid computing (Haque et al., 2011). Few of them are revenue, communication overhead, success rate, average turn-around time, total simulation time, resource utilization, user utility, resource utility and social welfare. As we consider the concept of reservation price¹², we ignore market liquidity in our simulation. As we are comparing the performance of different economic models, we apply this reservation price for all models. We test the five economic models in terms of all of these metrics. For each metric, we generate a 100 by 100 matrix, each of its cell stores the value of that performance metric in the corresponding $s-d$ value.

Currency (budget) for the Gridlets is injected into the system using the limits shown in Table 3.2. This currency injection is consistent with the existing literature (Broberg et al., 2008). Broberg et al. have explained the drawbacks of having unlimited currency to the Gridlets in a simulated system (Broberg et al., 2008). Some of the points they have raised are that having unlimited currency could lead to starvation (domination of higher budgeted Gridlets over lower budgeted ones), inflation and hoarding (hidden fund for future domination), which hamper a consistent resource allocation. In addition, Simjava provided *Random Uniform Distribution* is used to generate samples in the ranges shown in Table 3.2. We have chosen the seed values in the random number generator in such a way so that it can produce well-spaced sequences to remove correlation in the samples. This helps to simulate the entities with realistic configurations.

3.5.3 Methodology for Data Analysis

We conduct quantitative analysis to obtain the solutions of the problems related to our research hypothesis presented in Chapter 1. To this respect, the parameters describing our experiments have been explained in the previous sub-section. The raw data obtained from experiments are processed in several steps before they are suitable for analysis. We use Matlab tool to process and present the data. The data are firstly organized according to economic models and different performance metrics. For a single metric, we obtain a dataset, which is a 100 by 100, matrix (based on the simulation space). Therefore, different economic models

¹² The minimum price a user must pay to get access by a resource. This is the job execution cost by a node

produce different datasets for different metrics. To increase the significance of these datasets, multiple experiments are carried out with different distribution of input parameters. Datasets for a single metric are then averaged, and the averaged dataset is then plotted for presented. The simulation space defined before enabled us to analyze the plots in a comprehensive way (a 3-d mode of evaluation). For a comparative analysis, a cell-wise comparison is processed among the averaged datasets for different models. As we need to evaluate and compare among the performances of different models, we have to make sure that the experimental procedures are unbiased to produce these data. The following sub-section presents the justification of our methodology in terms of conducting a fair evaluation among different economic models.

3.5.4 Fairness in Evaluation

As the focus of this thesis is to investigate and analyze the performances of widely proposed economic models in Grid computing, a consistent evaluation across different models is, therefore, essential. This evaluation would further bring accuracy in a comparative analysis among the performances on the models. Our framework is able to deliver such an evaluation methodology due to the following reasons.

- Each economic model is evaluated using the same parameter configuration (Table 3.2)
- The economic models have been implemented only based on their basic principles. No further strategic behavior has been incorporated to improve any kind of performance in a model
- The simulation space remains same for all models
- The framework considers the concept of pseudorandom approach (seed) for generating values from a particular range. This consideration ensures a consistent generation of samples for all models. For example, for generating the MIPS rating for resources, the range is (350, 450). Now if we use a particular seed value (e.g. 7489113) to generate samples from this range, we could get samples like, 378, 401, 352... As long as a generator is seeded with the same seed value, it would always produce the same sequence of samples. In our simulation, we evaluate all the models using a same seed value, thus ensuring the same distribution of the parameters across all the models

To facilitate the understanding of our evaluation process, a holistic diagram is presented in Figure 3.7. The input and output sections go under the Application Management process. As stated before, we use different samples of *Uniform Random Distribution (URD)* as inputs for statistical need. Once, an experiment is executed on the defined simulation space and for a particular economic model, we receive different *Data Sets (DS)* due to different samples and

for different *Performance Metrics* (PM). These data sets are then averaged for evaluating the performances of the economic model through figures. The detailed experimental procedure on Monash Sun Grid is presented in the following sub-section.

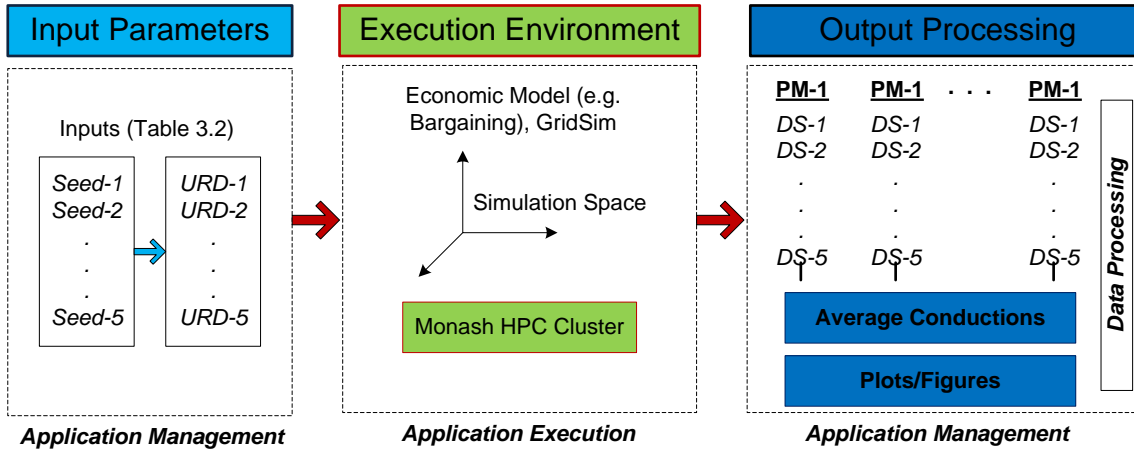


Figure 3.7: A Holistic Diagram of the Evaluation Model

3.6 Simulation using Monash Sun Grid

This sub-section outlines the key components of Monash Sun Grid (MSG) utilized to conduct our experiment. As we are dealing with an extensive simulation space, to expedite our simulation process, we use Monash University's High Performance Computing cluster called MSG. The detail resource configuration of MSG can be viewed at <http://www.monash.edu.au/eresearch/services/mcg/msg.html>. For a single economic model, we have 10^4 unique simulations. For statistical purposes, we conduct the same simulation for five different distributions; therefore, at the end, for a single model, we conduct 5×10^4 different simulations. Likewise, we do this for five different models. In addition, conducting such an extensive simulation on MSG is secure, reliable and easily manageable, because MSG is dedicated only for the Grid users worldwide.

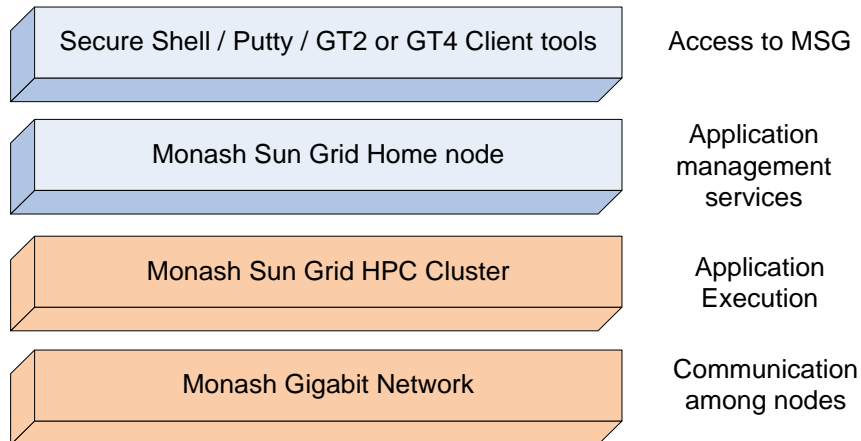


Figure 3.8: Monash Sun Grid: A Layered Overview

Figure 3.7 illustrates the layered architecture of MSG. First of all, one needs to create an account to get access to the cluster, which can be done by contacting the corresponding body. The contact details can be found at the MSG website at <http://www.monash.edu.au/eresearch/services/mcg/msg.html>. Otherwise, one can use Globus client tools to get access. Once the account is created, one can directly login into the site (msgln1/msgln2.its.monash.edu.au) using Secure Shell or Putty client tools. In our case, we use Putty. The next layer describes the application management services, which is referred to as “Home Node”. This node contains the application that needs to be executed on the Grid, relative scripting files for job submission to the execute nodes, and relative outputs once the application is finished. One must write scripting files to make the application suitable for submission on the cluster. The pseudo-code of the script files for a sample application is presented below.

The application must be submitted using *qsub* command in the console of the home node. The memory allocation should be enough for the application; otherwise, the execution will be halted. Once the application is executed, we can obtain the relative execution history. Figure 3.8 presents the screen-shot describing the execution records for our Bargaining model. In the following section, we present a taxonomy for widely proposed economic models in the Grid and advise the topics for two subsequent chapters.

```

App1.sh
#!/bin/sh
#$ -S /bin/bash
#$ -l proc=intel           //define preference on particular processors

Java -cp <application-dependent libraries (in our case, GridSim and SimJava)> <class containing the main method> <command-line variables>

```

```

SubApp1.sh
#!/bin/sh
#$ -S /bin/bash

qsub -cwd -s /bin/sh <-l define an estimated time for the app> <-l define a memory requirement for the app> App1.sh

```

```

msgln1.its.monash.edu.au - PuTTY
sahaq1@msgln1 ~$ qacct -j 2937110
=====
qname      tgu8
hostname   gn141.its.monash.edu.au
group      hpcimio
owner      sahaq1
project    NONE
department defaultdepartment
jobname     B&R.sh
jobnumber  2937110
taskid     undefined
account     sge
priority    0
qsub_time  Sat Sep 24 13:28:30 2011
start_time Sat Sep 24 13:35:35 2011
end_time   Sat Sep 24 20:17:37 2011
granted_pe NONE
slots      1
failed     0
exit_status 0
ru_wallclock 24122
ru_utime     14568.697
ru_stime     7426.569
ru_maxrss   0
ru_ixrss    0
ru_ismrss   0
ru_idrss    0
ru_ismrss   0
ru_minflt   90412601
ru_majflt   3
ru_nswap    0
ru_inblock  0
ru_oublock  0
ru_msgsnd   0
ru_msgrcv   0
ru_signals  0
ru_nvcsw    1285915757
ru_nivcsw    837633767
cpu         21995.266
mem         216374.187
io          0.000
iow         0.000
maxvmem     10.462G
arid        undefined
sahaq1@msgln1 ~$

```

Requesting execution records for the application ID 2937110

The execute-node

The name of the application

Time records for the application

Memory consumed by the application

Figure 3.9: Major Records Related to an Application in MSG

3.7 Conclusions

Realizing the evaluation process of different economic models by different researchers using different evaluation platforms, which may not guarantee a neutral assessment over the models, this chapter developed an evaluation framework for investigating and analyzing the performances of the models. The framework has been developed using GridSim toolkit. The framework described the key entities and their basic behaviors in a simulation environment. The communication model defined in the framework helped to understand how different entities communicate their interests among each other. The definition of resources and simulation space helped to understand the significance of our framework in terms of dealing with the dynamic nature of the Grid. The parameter configuration and its statistical significance further enabled to deliver a consistent and sustainable evaluation methodology. As all the models are evaluated over the same simulation space and parameter model, a fair evaluation process across the models is therefore ensured. The framework thus enabled to measure the performances of individual models in terms of their methodologies. The influences of varied platforms and design constraints over these performances are limited here. This approach would further enable a justifiable comparison process among the performances of the models, which we will see in the following chapters. The evaluation framework developed in this chapter, has therefore provided the materials to deal with the first research issue raised in Chapter 2. In the following chapter, we describe the working frameworks and analyze the performances of two commodity-based models – Commodity Market and Bargaining to evaluate their effectiveness for Grid resource management.

Grid Resource Management Using Commodity Markets

As a part of dealing with the second research issue addressed in Chapter 2, this chapter discusses the framework development and performance analysis of two most widely proposed commodity-based economic models – Commodity Market and Bargaining. Later, the models are comparatively analyzed to understand their strengths and weaknesses in different Grid scenarios.

4.1 Introduction

The Grid will not be evaluated based on its architecture, but the value it delivers to the users (Foster, 2002). Economic models are used to provide this value. Economic models facilitate Grid users to express their demands and Grid providers to advertise their resources. Commodity Market and Bargaining are the two most widely discussed economic models can be distinguished from one another in terms of communication and pricing methodologies of the models.

Commodity Market Model (CMM) has become remarkable in the Grid mainly because of its suitability for maintaining equilibrium between supply and demand of resources (Stuer et al., 2007, Abdelkader et al., 2010). The essence of CMM is to determine an equilibrium price based on current supply and demand function in a Grid environment. The price can change dynamically as the supply or demand for resources changes. An extensive research has been conducted to identify a suitable method for manipulating this price in the context of the Grid (Stuer et al., 2007). Overall, the model is more system-centric rather than incentive-centric for individual users or providers in the environment.

Bargaining Model (BM) on the other hand, is unique in the Grid due to its utility-based negotiation between a user and a resource provider. To deliver an effective and dynamic resource management system, researchers study various negotiation strategies for the model (Borissov et al., 2010, Sim, 2007). Typically, in BM, both the user and provider can try to optimize their individual objective functions (e.g. price, time) through multi-rounds

negotiation process. Therefore, the model is quite different from CMM in terms of working methodology.

The entities (e.g. broker, resource) and their characteristics in both CMM and BAR have been designed and developed in this chapter. The simulative study helps to analyze the effectiveness of the models for Grid resource management. A comparative analysis on the performances of the models further helps to identify the suitability of individual models in different Grid scenarios. We contribute the development of CMM and BAR to the current GridSim distributions.

The rest of the chapter is organized as follows. Section 4.2 describes the resource management process using CMM. The section includes the communication model for different entities, significance and determination process of equilibrium price in the model. The simulation results for the model have been discussed in Section 4.3, which incorporates the analysis of the model in terms of various performance metrics such as revenue, communication overhead, and social welfare. The design and development process of BM has been presented in Section 4.4. It describes the bidding strategies for both users and resources and working diagrams of the model. The performances of the model have been discussed in Section 4.5. Section 4.6 provides a comparative analysis among the performances of the models. The summary and conclusion of this chapter has been presented in Section 4.7.

4.2 Resource Management Using Commodity Market Model

Market mechanisms are constructed not only to manage the market environment successfully, but also to predict the market behavior for future. Therefore, it is necessary to understand the parameters that define a market mechanism. In this section, we analyze a distributed Grid market using Commodity Market Model (CMM). As mentioned in Chapter 2, there are two types of CMMs in the literature; one is flat pricing, in which price for a resource does not vary frequently, and another is supply and demand driven pricing, in which the price changes frequently depending on the market's supply and demand function. The latter is widely adopted and more popular in the Grid (Stuer et al., 2007, Buyya et al., 2002). Thus, we aim to develop the supply and demand driven CMM. A complete workflow diagram of the CMM is illustrated in Figure 4.1.

Figure 4.1 depicts the interaction among different entities in CMM. In our simulated Grid network, each of the entities is attached to network with a constant bandwidth. In our case, we use 10Kpbs. The user has an experiment, which needs to be executed on the Grid. At first, the user submits his/her experiment along with other preference parameters, such as budget and deadline to a broker. The broker collects available resource IDs from GIS (not shown in

the figure), before it starts talking to the resources (one at a time) about the possibility of executing the experiment. Each resource node has its own *Observer* and *Responder* to help the resource to make a decision (either acceptance or rejection) about the request. The *Observer* on behalf of the resource interprets message types and obtains the relative responses from the *Responder*. The *Responder* performs the original matchmaking process and sends the results back to the *Observer*.

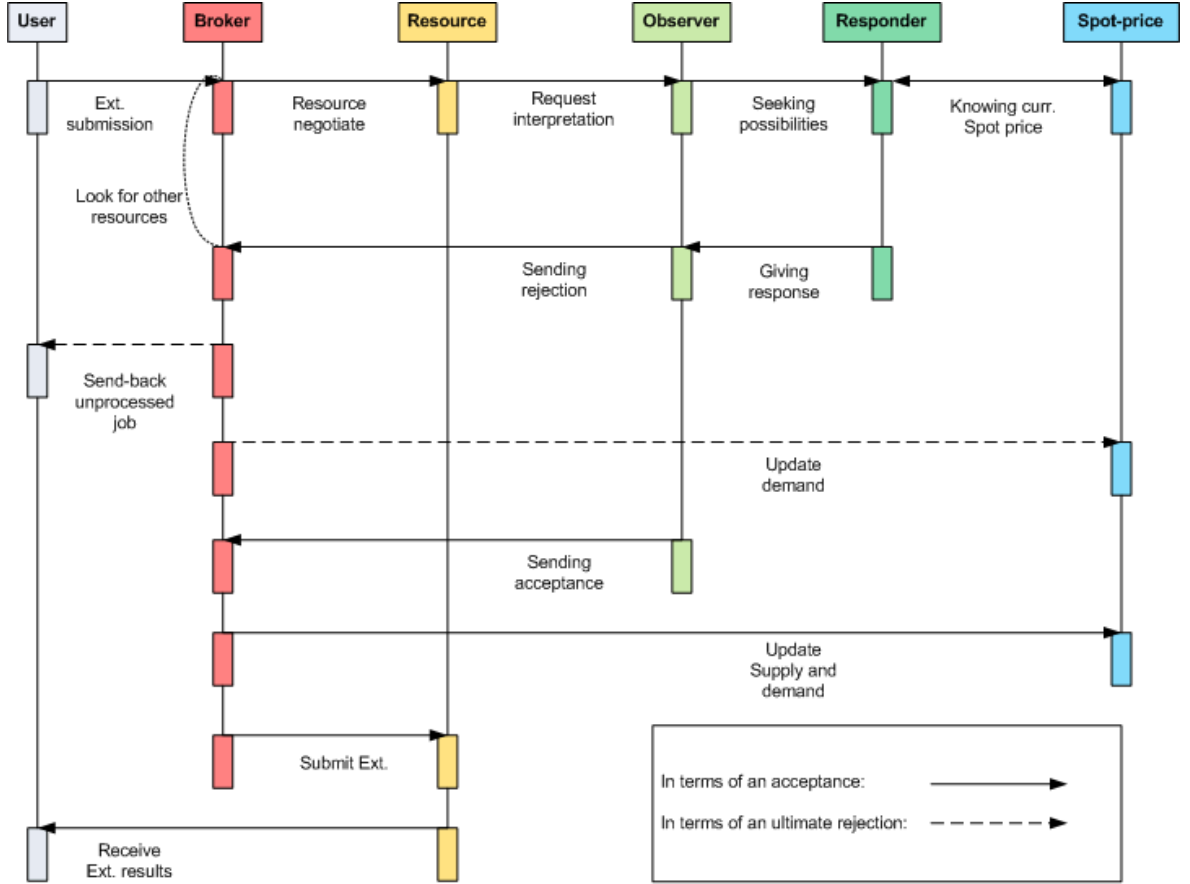


Figure 4.1: An Event Diagram for the Interaction between Different Entities in CMM

Realizing a Grid's extensibility and the possibility of receiving numerous requests by a resource, we distribute the workload among different entities (*Observer* and *Responder*) that work on behalf of the resource.

To check the possibility of executing a request on the resource-node, *Responder* considers the deadline and budget of the experiment with the node's MIPS rating and the execution cost. According to GridSim, the execution time, $cpuTime$ for a job (experiment) on a node, n is given by,

$$cpuTime = \frac{jobLength}{MIPS\ rating\ for\ n} \quad Eq - 4.1$$

On the other hand, cost for executing a job, $cpuCost$ on a node, n , is given as,

$$cpuCost = jobLength \times \left(\frac{cost\ per\ sec\ for\ n}{MIPS\ rating\ for\ n} \right) \quad Eq - 4.2$$

A job will be permitted to submit on the resource, if the following conditions are met:

$$job-dl \geq cpuTime\ and\ job-budget \geq cpuCost$$

Where, *job-dl* and *job-budget* refer to the deadline and budget for the job respectively. The essence of CMM is in the dynamic change of the *cost-per-sec* parameter with supply and demand in the environment. As the focus of the market is to determine an equilibrium price¹³ (also known as spot price) depending on the current supply and demand, we develop a central manager (spot-price determinator in Figure 4.1). The role of the manager is to keep track of supply and demand in the environment and to calculate the spot prices. The manager responds every time a *Responder* requests it. We explain how the spot prices are determined, in sub-section 4.2.2.

Once the decision of either acceptance or rejection is made, the *Observer* sends the message to the broker. If accepted, the broker submits the job to the resource. The central manager is then called to update both the supply and demand in the market. If rejected, the broker checks whether it has finished negotiation with all the resources in the market. If it has finished, it sends the unprocessed job to the user. At this stage, an ultimate rejection occurs. The central manager is then invoked to update only the available demand in the market. In case the broker has not finished negotiating with all the resources, it sends the request to another resource and the process continues. Once the job is executed by the resource, the resource sends the job back to the respective user along with the job-outputs (results). The methods used to model the behavior of broker and resource in this process is shown in Algorithm 1 and Algorithm 2 respectively. The methods are further explained in the following sub-section.

¹³ Price at which the accumulated supply and demand are equal to one another

Algorithm-1: Broker behavior in a concurrent CMM-based Grid:

resourceIDList: Obtaining available resource information from GIS
 Let *counter* = 0

sendingRequest(): Send this job to the first resource in the *resourceIDList* with an arbitrary delay

receiveFeedback():

Case-I: Acceptance

Submit the job to the resource

Case-II: Rejection

Remove the resource ID that has rejected from the *resourceIDList*

If (size of *resourceIDList* > 0) sendingRequest()

Case-III: Resource is busy

If (the size of the *counter* reaches to the size of *resourceIDList*)

[reset the *counter* with 0]

sendingRequest() with the resource as the *counter* suggests

Increment the *counter* by 1

Algorithm-2: Resource behavior in a concurrent CMM-based Grid:

Let *resource-status* free

receiveRequest():

If (The status of the resource is free)

```
{
    set resource-status busy
    start processing on the request
    sendingAcceptance() / sendingRejection()
    set resource-status free
}
```

Else sendingBusy() //asynchronous event

4.2.1 Concurrent Market

In our simulated market-based Grid environment, we use concurrent negotiation process, i.e. a broker will not wait for other brokers to finish. Such a characteristic brings real distributive non-deterministic nature in our simulation. However, this complicates the process of defining the behavior of entities for supporting concurrency. In such a scenario, a broker should have the ability to start negotiation with another resource, if the broker finds a resource busy at that time. On the other hand, the resource needs the capability to tell the broker at the same time that the resource is busy, which can be managed by employing some asynchronous properties¹⁴. Algorithm-1 and Algorithm-2 consider this.

In Algorithm-1, if the request is not being accepted, the broker keeps sending the request until the *resourceIDList* becomes zero. This means the broker has surveyed all the resources

¹⁴ When an entity raises an event and continues working with other activities without waiting for the completion of the event

without success. At this stage, the broker terminates surveying. The broker using some arbitrary delays schedules the requests to the resources. This is to overcome the overall contention¹⁵ problem in the environment. How the market price is determined by a resource in CMM is elaborated in the next sub-section.

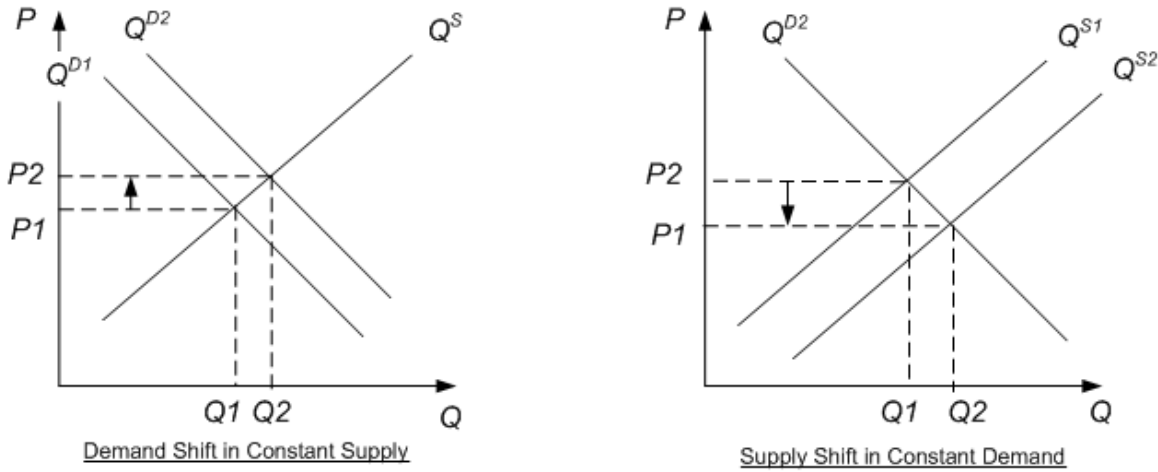


Figure 4.2: Change in Equilibrium Price, P due to Supply and Demand Shift

4.2.2 Market Equilibrium

In this particular sub-section, we discuss the value and method of price determination process in CMM. Market behavior is predominantly influenced by price. In the nineteenth century, French economist Walras identified that it is possible to analyze and identify a price, which has the ability to coordinate the allocation of resources in the market (Gomes and Kowalczyk, 2010). This price is called equilibrium price or spot price. When a market finds this price, the market enters into an economic equilibrium state, that is, the market behavior will not be influenced by an individual budget or cost. This helps in finding Pareto-optimal¹⁶ solution for resource allocation problem. The solution can also be regarded as consistency in resource allocation. However, in a distributed large-scale environment such as Grid, entities will have their own strategies to apply in the market, which might hamper in obtaining expected market equilibrium. In this thesis, we focus on the provider side and consider naïve users. We do not focus on developing any strategic behavior for users.

The spot price for a market can be determined using the market's supply and demand function. There are both linear and non-linear methods of investigating the price. We use linear method due to its simplicity (Hands, 2003). According to linear equilibrium theory (Hands, 2003), the demand and supply functions are given as,

¹⁵ Conflicts arise when the same resource is accessed by multiple requests at the same time

¹⁶ A solution is referred to as pareto-optimal, if there is no other solution that can improve the utility for an individual without worsening that of others. In other words, the magnitude of resource demands and availability is equal

$$\begin{aligned} QD &= -aP + b \\ QS &= cP + \alpha \end{aligned}$$

Where Q^D refers to the quantity demanded at any specific time and Q^S is for supply; a , b and c are the scalar parameters where a , c are the changes in demand and supply respectively, b is the current demand. The value for b is calculated using the current number of Gridlets still looking for resources in the market. The values for a and c are found using:

$$\begin{aligned} a &= \frac{\text{current demand in the market}}{\text{starting demand in the market}} \\ c &= \frac{\text{current supply in the market}}{\text{starting supply in the market}} \end{aligned}$$

Thus, the values for a and c usually range from zero to one. The negative sign in the demand function represents the relationship between price (P) and demand, which is, an increase in price will induce a decrease in the quantity demanded and vice versa (Figure 4.2). In the supply function, α refers to the shift in supply, which can be found using:

$$\begin{aligned} \alpha &= \frac{\text{starting supply} - \text{current supply}}{\text{starting supply}} \text{ Or,} \\ \alpha &= 1 - \frac{\text{current supply}}{\text{starting supply}} \end{aligned}$$

As we are using linear equilibrium theory, the definition of demand or supply is straightforward. The supply and demand are defined in terms of number of nodes and Gridlets respectively. Now, if we need to know the price at which, the total supply and total demand at any given state diminishes, we need to solve the supply demand functions for P when $Q^D = Q^S$. If P^* is the spot price, we get,

$$\begin{aligned} Q^D &= Q^S \\ -aP + b &= cP + \alpha \\ P^* &= \frac{b - \alpha}{a + c} \end{aligned} \quad \text{Eq - 4.3}$$

As we already know the values for a , b , c and α , we can determine the spot price at any given state in the market.

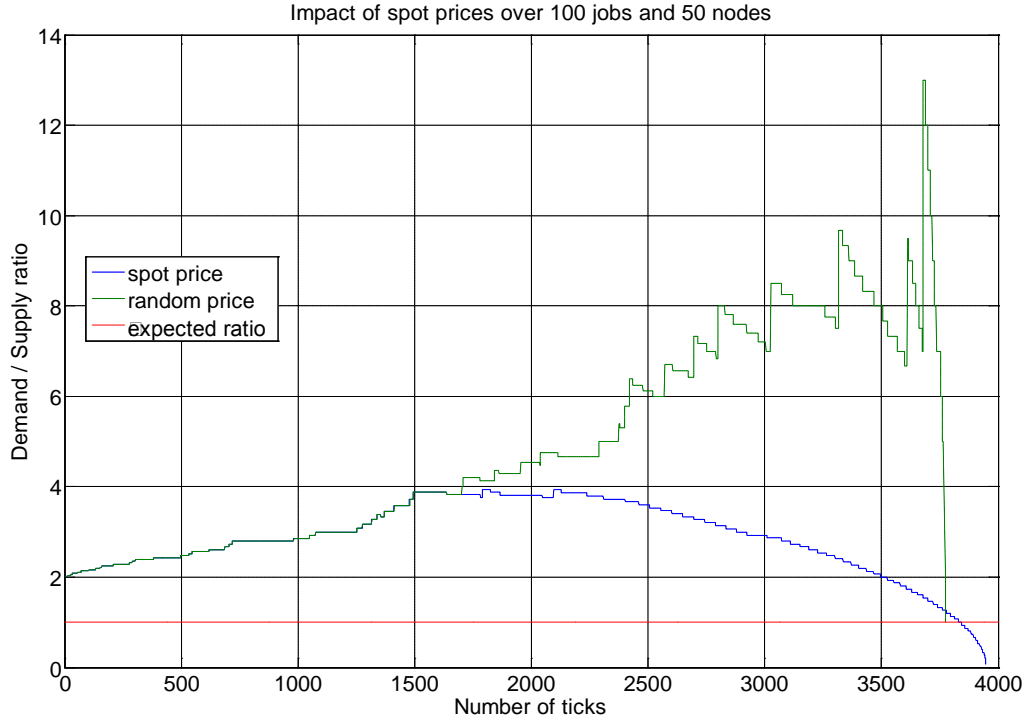


Figure 4.3: The Effect of Spot and Random Prices over Supply and Demand

In order to ensure that the defined spot price works for pushing the market into equilibrium, we conduct a sample experiment with 100 users and 50 nodes. The effect of random price and spot price over the market are shown in Figure 4.3. The trend for demand-supply ratio for spot price is smooth compared to the trend using random prices. On the other hand, supply and demand is not affected by the random prices; hence, the respective trend fluctuates a lot. Since we are using ranges of *MIPS* rating for the nodes, when demand exceeds supply, spot price becomes higher which results in some users choosing low-priced nodes instead of high-priced nodes or even reject the users to maintain equilibrium between supply and demand. In the following section, we describe simulation related materials and analyze the performances obtained for CMM.

4.3 Output Analysis for Commodity Market Model

Detailed simulation conditions are provided in Section 3.5.2. We use the same variables in Table 3.2 over all the models for the sake of a consistent evaluation. For each performance metric, five different simulations are conducted using 100×100 (s, d) matrixes. The average values of the five simulations are then taken and presented for each metric. To minimize the complexity in explanation, we use the term “Gridlet” to refer to the user, broker or application.

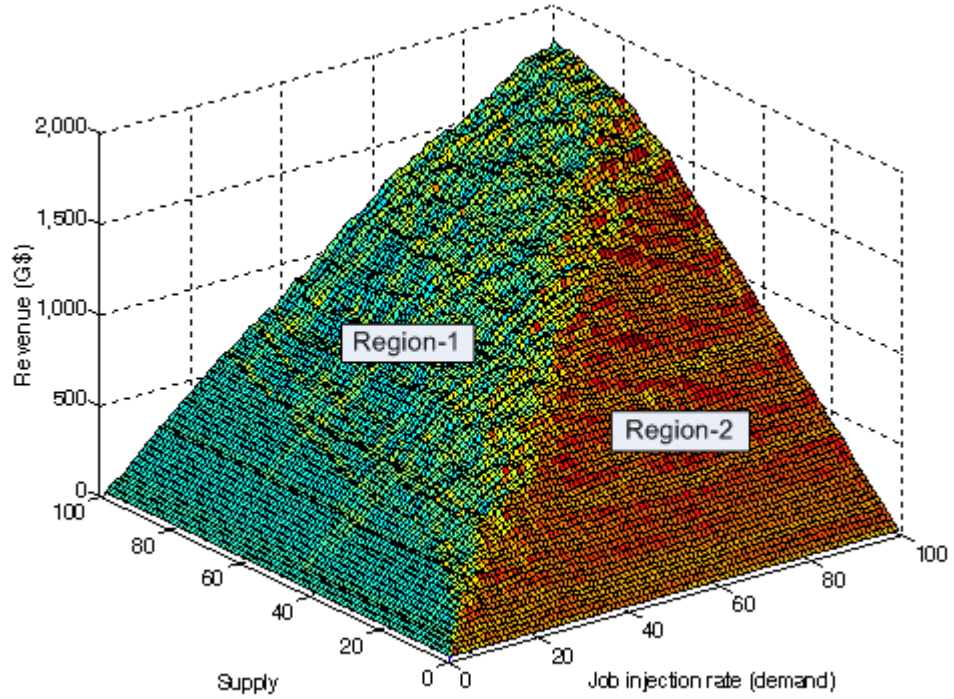


Figure 4.4: Revenue Distribution for CMM

Revenue: This means money (G\$) earned by the resources. For a particular simulation (a set of a supply and demand), the accumulated revenue is presented here. Figure 4.4 illustrates the revenue distribution for CMM. The X-axis of the figure represents the job/Gridlet injection rate¹⁷. This is also referred as demand rate. Y-axis represents the supply. The Z-axis represents the magnitude of revenue for all possible steps in supply and demand. It can be seen from the figure that the revenue increases along with the supply and demand. As both the supply and demand increases, the chance for the more Gridlets to be accepted also increases. Thus, the accumulated revenue goes up. However, there are two different regions characterized by the supply and demand variation. Region-1 is more flatten compared to Region-2. Region-1 is characterized using high supply and low demand. The resource costs (Eq-4.3) generated is lower, due to sufficient supply compared to demand. The system lowers the cost to utilize more resources to achieve equilibrium between the supply and demand; therefore, in this region, the Gridlets are accepted by the resources with low costs. This acceptance prevents the resources from maximizing their revenues. In addition, due to lower demand, the accumulated revenue is also lower for Region-1.

On the other hand, Region-2 explains the scenarios when there is higher demand compared to the supply. CMM generates higher spot prices, due to higher demand, which

¹⁷ The number of jobs sent per unit time period

helps to bring some Pareto-optimality in resource allocation (consistency in resource allocation in accordance with the Gridlets budgets). Therefore, some lower budgeted Gridlets, in Region-2, are pushed to choose lower performance resources (lower MIPS rating) or might be rejected to bring the market into equilibrium. As only the higher budgeted Gridlets can survive longer in such scenarios, resources are provided with higher revenue compared to Region-1. When there are 100 Gridlets and 100 resources, the highest revenue is obtained, which is about G\$1800.

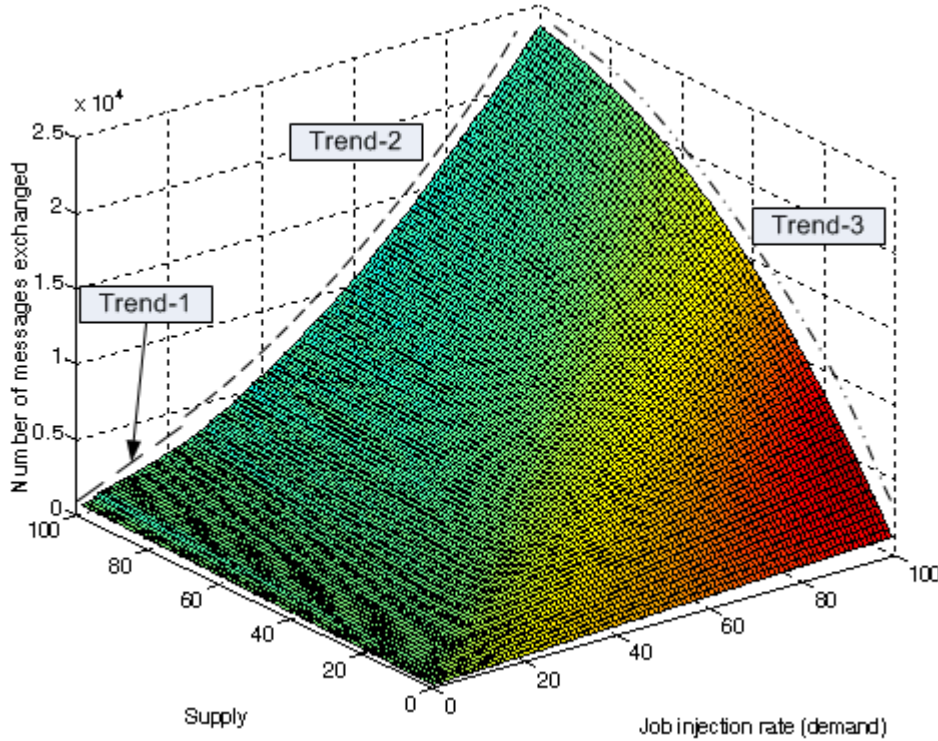


Figure 4.5: Distribution for Communication Overhead

Communication Overhead: The communication overhead is defined in terms of the total number of messages exchanged during a simulation's life cycle. Figure 4.5 depicts the distribution for communication overhead. There are three different trends in the plot. Trend-1 is almost straight; when the total number of Gridlets is lower than 20% irrespective of the supply. The lowest number of messages has been exchanged in this region. Gridlets can quickly identify their suitable resources from a pool of resources, due to lower demand and higher supply, because the resource costs are lower here. The number of messages exchanged is also lower, due to this quick acceptance of the Gridlets by the resources. However, as the demand starts increasing and the supply is still higher (Trend-2), spot prices also start increasing. This pushes some Gridlets to stay longer in the market and generates more messages. It can be seen that the growth of Trend-2 is slower compared to Trend-3. The reason behind this is that for Trend-3, the supply starts decreasing, when the demand is high.

The competition in Trend-3 is the highest, because there are not enough resources compared to the number of Gridlets. This raises the spot prices and compels the Gridlets to stay longer to look for suitable resources. Thus, for Trend-3, a higher number of messages need to be exchanged.

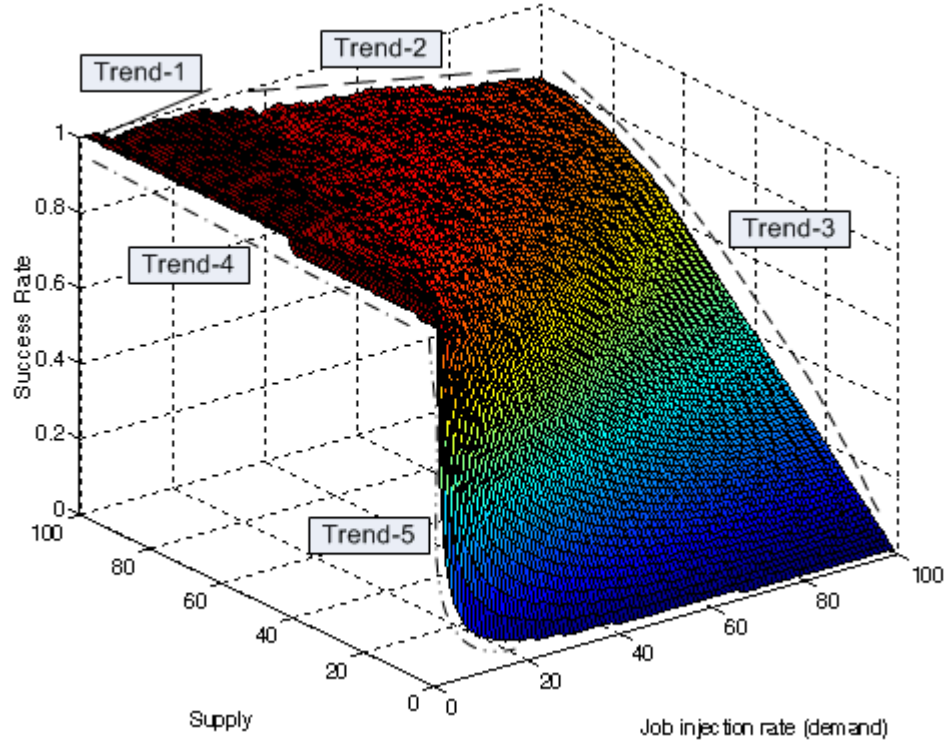


Figure 4.6: Distribution for Success Rate

Success Rate: The success rate is defined as the number of accepted Gridlets over the total number of Gridlets. Figure 4.6 shows this against supply and demand for CMM. There are five different trends in Figure 4.6. Trend-1 describes the scenario in which there is high supply and limited demand. The chance for a higher number of Gridlets to be accepted increases, due to available resources. In addition, the spot prices for this region are lower, which also helps for maximizing the acceptance ratio. As the demand starts increasing and the supply is still high (Trend-2), the ratio gradually decreases. For Trend-2, as the demand tends to be closer to supply, the chance of increasing the spot price grows slowly, because of the constant supply. However, as the supply starts decreasing and the demand is still high (Trend-3), the success ratio drastically drops. The reason is the higher spot prices for this region. When there is limited demand regardless of the supply (Trend-4), the success rate is almost constant. Change in spot prices does not occur frequently, due to the available resources with respect to demand; this helps the resources to obtain somewhat constant acceptance across the trend. When the supply becomes lower and the demand is still lower (Trend-5), the success ratio significantly drops. The Gridlets receive limited chance to explore the market, due to lower

supply. Therefore, a few of the Gridlets' rejection play a significant role in minimizing the overall success ratio.

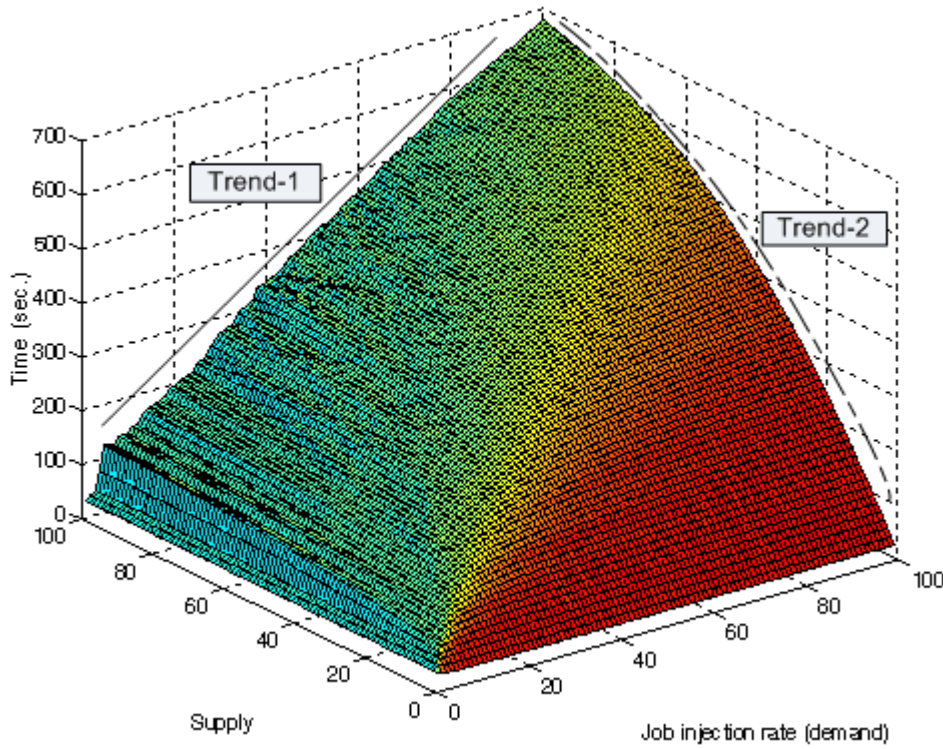


Figure 4.7: Distribution for Average Turn-around Time per Gridlet

Average Turn-around Time: Figure 4.7 illustrates the distribution for average turn-around time per Gridlet. The time is expressed in second. This is simulation second and varies from real time. We use simulation time throughout the thesis. There are two trends in this plot similar to the plot of revenue. This is the average time required for a Gridlet to know the message about its ultimate acceptance or rejection. When the supply starts increasing and the demand is limited (below 50%), the time fluctuates frequently. The spot prices are low, due to high supply and low demand. Though the acceptance rate is quick, resource allocation happens in somewhat random manner (in terms of timing), because the competition is lower for this region. As the demand starts increasing (Trend-1), competition also increases and Pareto-optimal resource allocation becomes apparent. Thus, the fluctuation in timing rarely happens in this region. Trend-2 can be explained similarly to that of the revenue. As the supply decreases and, the demand is high, the Gridlets stay longer in the market due to high spot prices. This increases the average time for a Gridlet.

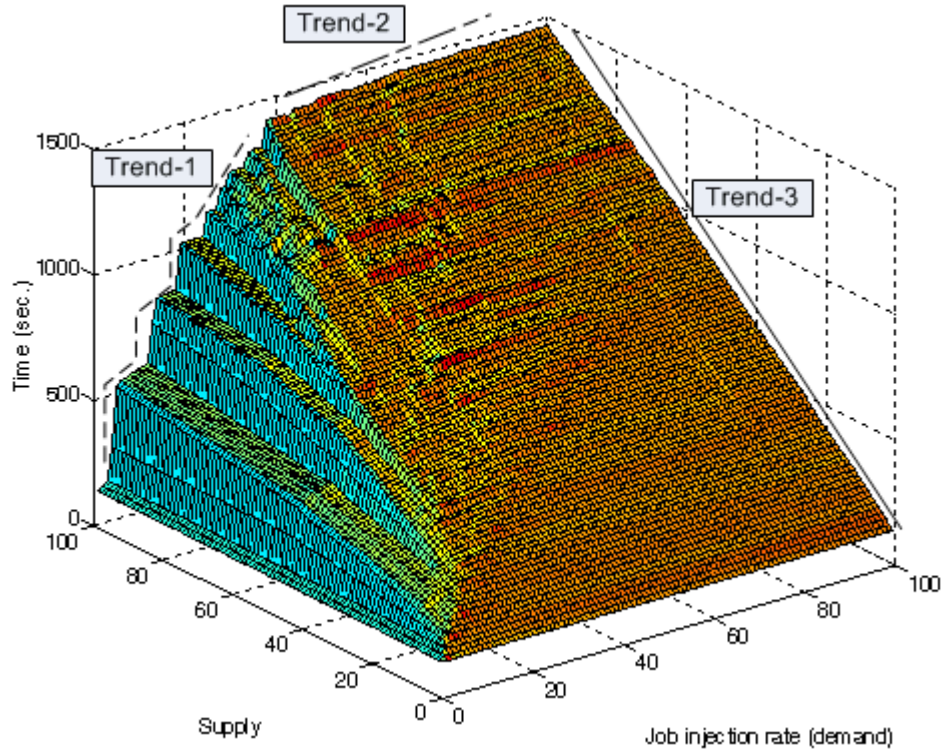


Figure 4.8: Total Simulation Time Distribution

Total Simulation Time: This particular metric is defined by the total time required to finish a simulation. As the supply increases and the demand increases up to $\sim 40\%$ (Trend-1), the simulation time increases in steps (Figure 4.8). When the supply is high and the demand is in between 15%-20%, the time remains constant. One of the possible reasons for such a trend could be due to the concurrent behavior of the entities. The Gridlets approach randomly to the resources, due to the concurrency. Even though it is random, as we are using seed values to define the arrival times (Table 3.2) for the requests, the system maintains an internal sequence for defining the times. The moment demand increases in the scenario, the overall sequence changes; therefore, the requests are sent in a different order. Thus, as it is not as stable as in the previous order, a Gridlet might need to stay longer in the market to discover its suitable resources. As such, the accumulated time for all the Gridlets to discover resources suddenly goes up. However, for a few Gridlets, the shift in the order does not change considerable, which might cause the time to be constant for some time. Again, as the number of Gridlets keeps increasing, the constant time period decreases and periods of steep jumps decreases. As the demand increases, even though the change in ordering the requests occurs, some Gridlets are quickly occupied. This causes some other Gridlets to be more quickly rejected. This minimizes the constant times. For the same reason, the periods of steep jumps keep decreasing. This effect gradually diminishes as the demand increases beyond 40%.

Due to high demand and supply (Trend-2), Gridlets rarely get a chance to stay longer in the market. It can be noticed that the time for both 50% and 100% demand remain constant. In terms of 50% demand, the spot prices are lower compared to 100% demand, which helps the Gridlet to be accepted more quickly. On the other hand, due to the higher prices at 100% demand, the chance of quick acceptance for the higher budgeted Gridlets increases and the rest of the Gridlets are not successful in getting the supply due to the competitive supply and demand in this region. Thus, the time does not go up in this case also. As the supply starts decreasing and the demand is high (Trend-3), the time proportionally decreases. The competition rises and the spot prices in this region rise, due to high demand. This causes the resources to be quickly occupied by the higher budgeted Gridlets and makes the other Gridlets to be quickly rejected.

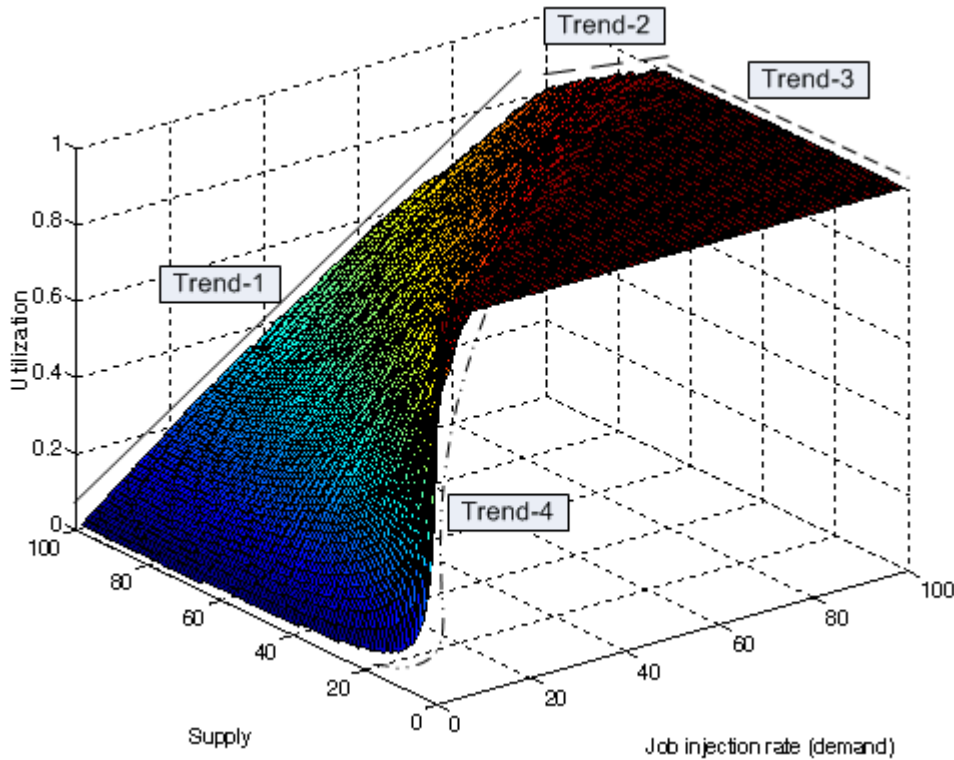


Figure 4.9: Distribution for Resource Utilization

Resource Utilization: The resource utilization of a particular simulation is characterized by the number of resources that has been used over the total number of resources of that simulation. Figure 4.9 presents the resource utilization for CMM. There are four trends observable in this figure. When the supply is high and the demand starts increasing (Trend-1), the utilization increases proportionally. There is little competition in the market, due to high supply; so the utilization increases with the number of Gridlets. However, when the demand approaches 80% or more (Trend-2), the competition is comparatively high. This results in somewhat constant utilization. As the supply starts decreasing and demand is high (Trend-3),

the competition for resources becomes significant. This helps most resources to be utilized. When the demand is low and the supply is also low (Trend-4), the utilization drastically drops. There is a low competition for resources, due to low demand, which results in low utilization of them.

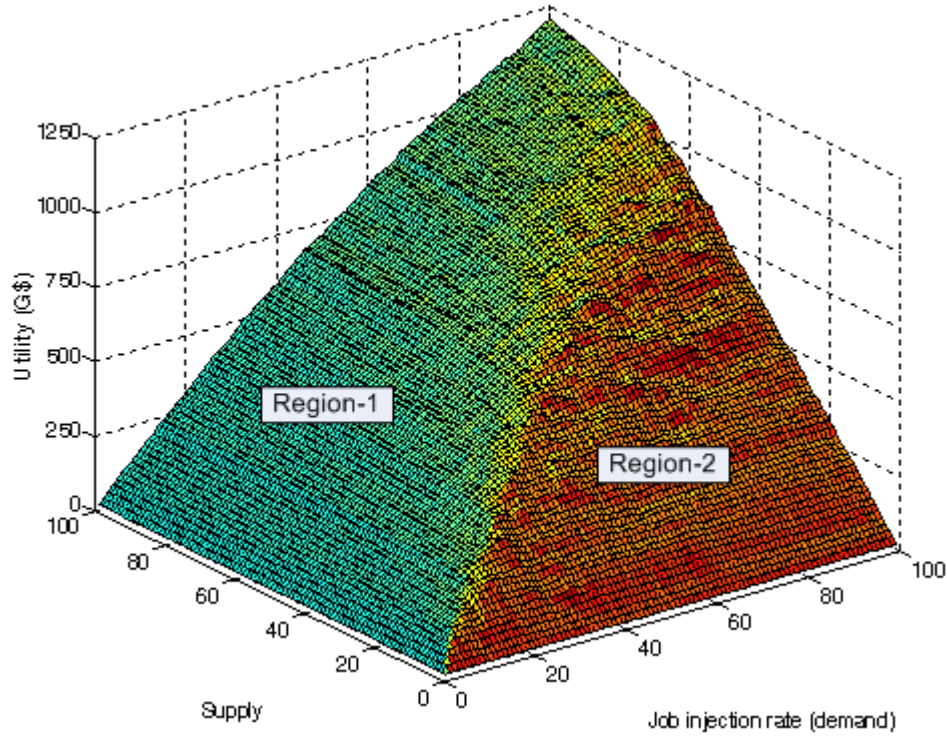


Figure 4.10: Distribution for User Utility

User Utility: As mentioned in Chapter 2, user/Gridlet utility is defined by the difference between a Gridlet's budget and the price the Gridlet has to pay for its execution (paid-cost). The utility is represented in G\$. In other words, one could say that this is the profit made by the Gridlets. Figure 4.10 illustrates the user utility for CMM. For a particular set of supply and demand, the figure plots the accumulated utilities for all the Gridlets. As the supply and demand increase, the utility also increases. The utility is only considered when a Gridlet is accepted. When the supply is low and demand is high (Region-2), the spot prices are high. One could expect this to result in low utility. However, as the demand is high in this region, the accumulated utility is higher compared to that of the Region-1. In addition, as the high budgeted Gridlets are accepted in Region-2, this helps to improve the overall utility for the region.

Resource Utility: This is defined using the difference between the agreed price and the resource's reservation price (job-cost). In CMM, the resources do not try to optimize their utilities and sell the resources at exactly their reservation prices; hence, the utility is zero for resources.

Social Welfare: Social welfare is the combination of the user and resource utilities. For CMM, the social welfare is equal to the user utility. In the following section, we demonstrate the development process of BAR and analyze the results obtained for the model.

4.4 Resource Management Using Bargaining Model

In this section, we extend our evaluation framework by incorporating another economic model – Bargaining (BAR). The section presents the development process and investigates the performances of BAR for distributed resource collaboration. BAR is one of the most widely proposed economic models in the Grid (Haque et al., 2011). Its special feature is that it supports the negotiation between users and providers over multiple rounds. Such a negotiation eventually helps to understand the requirements of the market participants better. It facilitates Service Level Agreement (SLA) processes and brings Pareto-optimal resource allocation among multiple users. In spite of the potential of the model for distributed resource negotiation, defining characteristics of the entities, i.e., advising the negotiation parameters over the rounds is challenging (Borissov et al., 2010, Sim, 2007). To simplify explanation, once again we use the term “Gridlet” to refer to the user, broker or application until the end of this section.

In a market-oriented Grid computing, both the Gridlets and resource-nodes try to optimize their individual objective functions. Gridlets try to get low cost access with sufficient CPU time, whereas nodes try to maximize utilities through bargaining. Bargaining protocol is suitable to find a common ground in these circumstances. Here, a Gridlet might start with a very low bid and a resource-node with a very high bid, and the negotiation process is continued until they reach a mutually agreeable condition or any of them does not show any interest to continue further (Buyya et al., 2002). Bargaining process could also be terminated at a certain number of rounds; because the higher number of rounds incurs higher communication cost. Hence, resource-nodes could define the total number of rounds for which bargaining will continue and this helps the Gridlets to decide how they should bid over the rounds. We set the total number of rounds (θ), 10 (Table 3.2). As the bid update process by users/resources over rounds is always consistent with the total number of rounds, we are not much concerned about defining the value of θ (the consistency is explained in the second last paragraph of this section). In addition, a higher value for the θ could lead the overall simulation process longer. Figure 4.11 shows the flow chart of how bargaining process is conducted between a Gridlet and a resource node. In this case, both the Gridlet and node use *Observer* and *Responder*, because they both need to make some decisions over the rounds. Figure 4.11 and Figure 4.12 show the Gridlet’s and resource’s side of negotiation strategy

respectively. The bargaining process begins with the agreement that the node is able to meet the Gridlet's deadline; otherwise, there is no means to bargain. In each round, the Gridlet uses the following strategy to update its *cfp-bid* (Call for Proposal¹⁸),

$$cfp\ bid = cfp\ bid + \left\{ \frac{max\ budget - min\ budget}{\theta - 1} \right\} \quad Eq - 4.4$$

The initial value for *cfp-bid* is zero, whereas the node starts with some higher bid (*max-bid*, Table 3.2). Node uses the following method to change its bid over the rounds,

$$node\ bid = node\ bid - \left\{ \frac{max\ node\ bid - min\ node\ bid}{\theta - 1} \right\} \quad Eq - 4.5$$

The aforementioned bid update process (Eq-4.4) is used by the English Auction protocol in GridSim. As we are comparing the performance of different economic models, to maintain consistency, we adopt the same bid update process for BAR. The bid update process is advised in such a way that if there is high number of rounds (θ), the increment/decrement of the bids over a round will be lower (the second half of Eq-4.4 and Eq-4.5) so that the negotiation can be continued for longer. This is very realistic, because the Gridlets can plan their relaxation amount over the rounds, because they know the total number of rounds prior to the start of their negotiation process.

The initial part of Figure 4.11 is the request submission and deadline verification process. Here, the Gridlet submits its *cfp* to node n . The node, then, verifies whether it can finish processing the Gridlet by the requested deadline. If the node can meet the deadline, the negotiation starts immediately, otherwise, the Gridlet initiates the same request with another node. The negotiation process is conducted over price. At the beginning of the negotiation, the node informs the Gridlet about the total number of rounds (θ) for negotiation. The process continues until they reach a mutually agreeable price. At every round, the node checks the current number of rounds (θ) and whether the bid from the Gridlet can meet the job cost. The middle part of Figure 4.11 shows the detail. If they are able to reach a mutually agreeable price within the total number of rounds, the negotiation succeeds; otherwise, the Gridlet starts looking for other nodes. Like CMM, BAR is also conducted in a concurrent way (Algorithm-1 and Algorithm-2), i.e., a Gridlet will not wait for other Gridlets to finish. The following subsection illustrates the results obtained for various performance metrics for BAR.

¹⁸ A message bid containing all the Gridlet related parameters

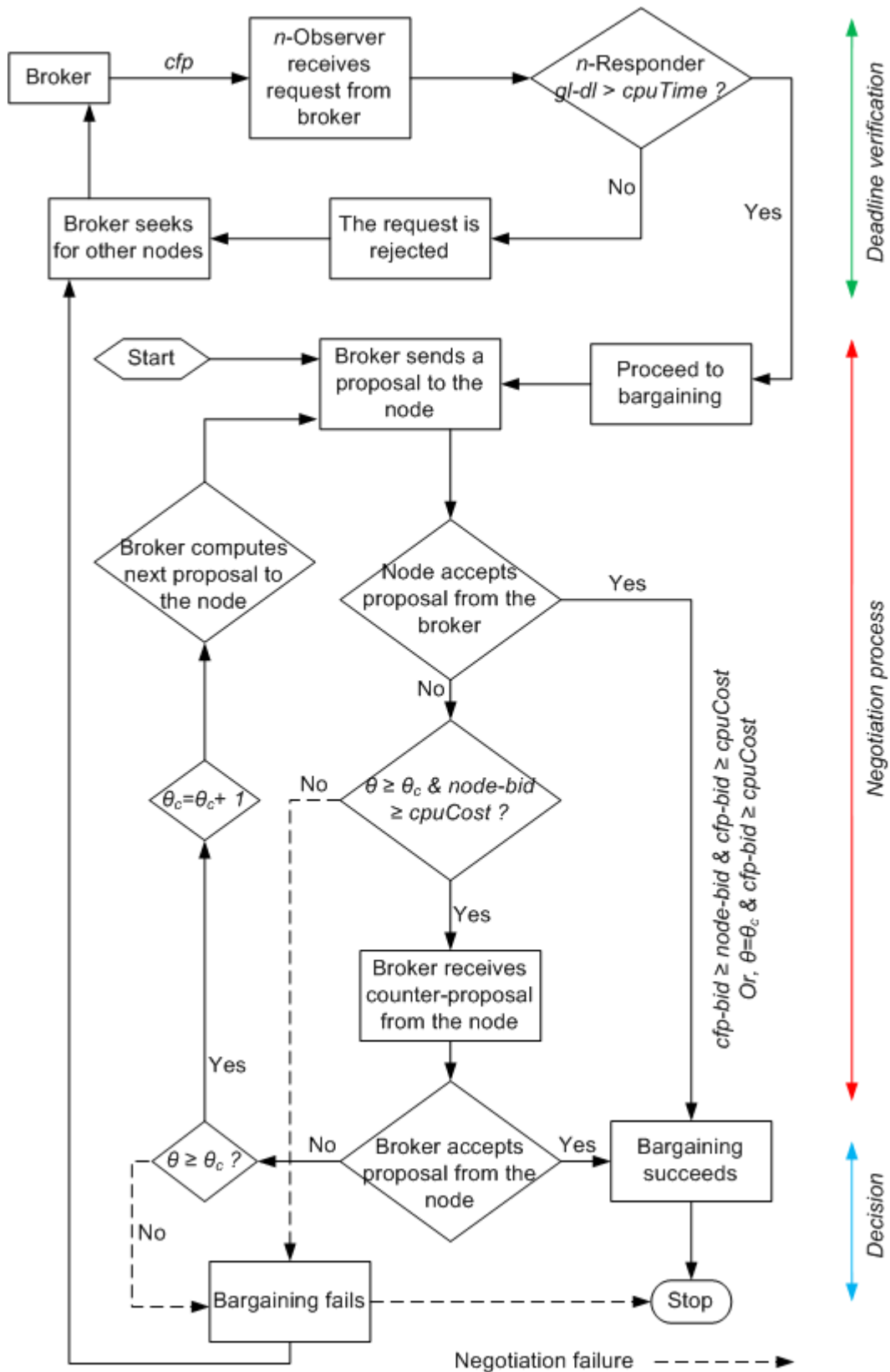


Figure 4.11: User's Strategy for Bargaining Model (θ_c = current number of rounds)

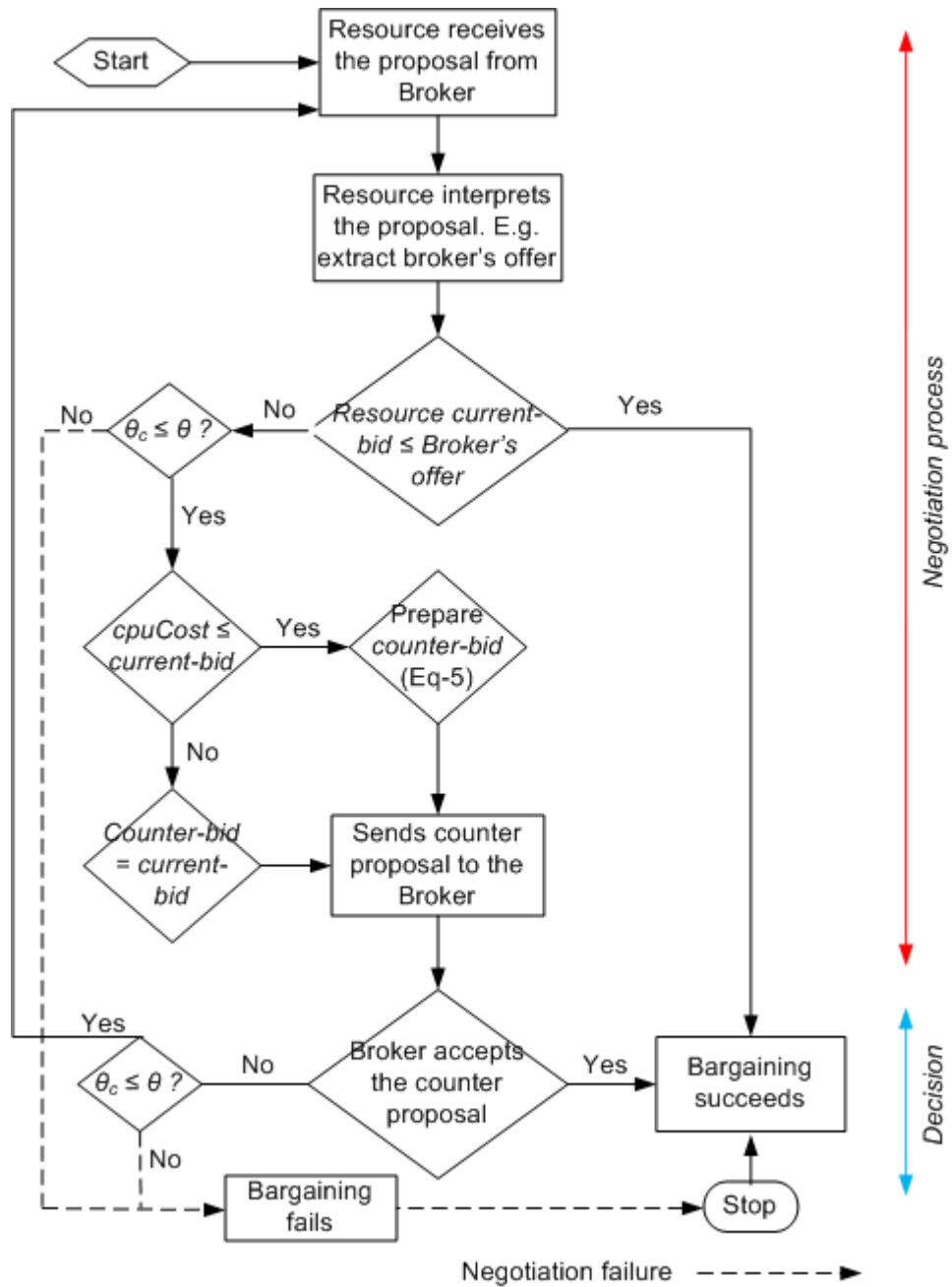


Figure 4.12: Resource's Strategy for Bargaining Model (θc = current number of rounds)

4.5 Output Analysis for Bargaining Model

This sub-section explains the results obtained for BAR.

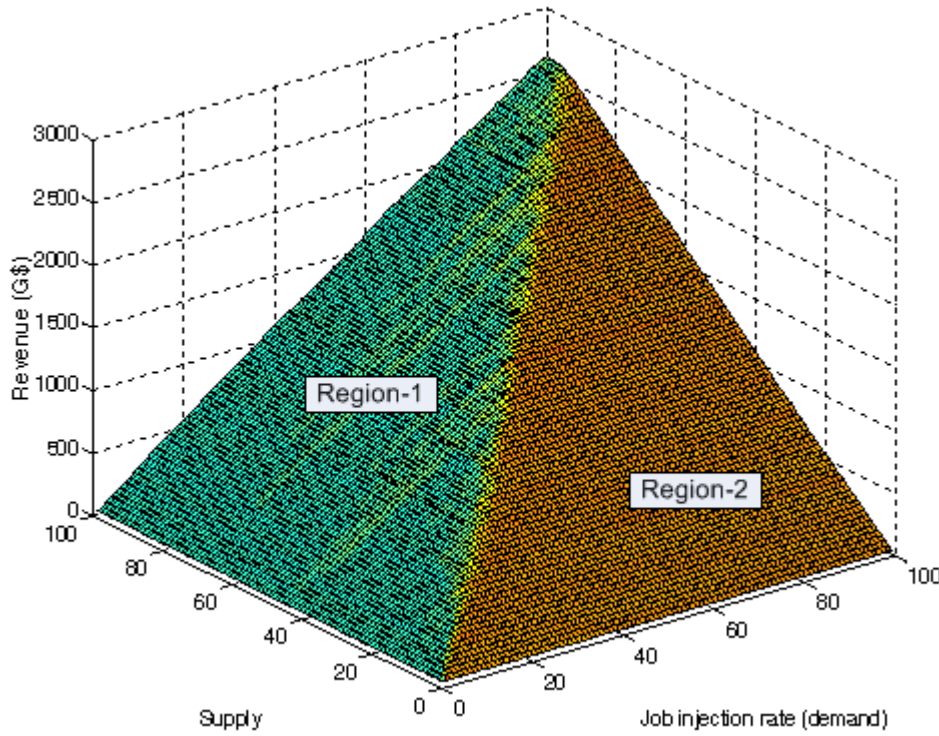


Figure 4.13: Revenue Distribution

Revenue: Figure 4.13 shows the revenue distribution for BAR. There are two different regions apparent for the revenue – Region-1 and Region-2. Different from CMM, Region-2 in this case, does not perform better compared to Region-1. In both cases, the revenue grows approximately in equal magnitude. In BAR, both Gridlets and Resources get an opportunity to negotiate for prices. This helps both Gridlets and resources to optimize their individual utilities, which brings some competition at every negotiation. Another reason for not maximizing the revenue in Region-2 compared to Region-1 is that, BAR does not vary prices over a simulation like the one CMM does (Figure 4.4). This prevents the resources from maximizing the revenue in Region-2. However, the overall revenue for BAR is higher compared to that of CMM. This can be explained using the Pareto-optimal resource allocation. The more the Pareto-optimality in resource allocation, the higher the success rate is. The aggregated revenue will also be higher, due to high success rate. We can see from Figure 4.15 and Figure 4.6 that the success rate is higher for BAR. In BAR, as the resource prices do not vary over supply and demand, Gridlets and resources can bargain based on their budgets and job costs respectively. Thus, there is a high chance that a Gridlet will get a

resource whose performance is closer to the Gridlet's requirements. This brings some kind of optimality in resource allocation, which ultimately helps to maximize the success rate.

However, for CMM, when the demand is high and supply is low (Region-2 in Figure 4.4), the market is dominated by the higher budgeted Gridlets; therefore, there is a high chance that these Gridlets will occupy the resources without considering consistency between the Gridlets' requirements and the resources' performance. On the other hand, when the supply is high and demand is low in CMM, the market is dominated by the resources, i.e., in this region (Region-1 in Figure 4.4), the spot prices are low. This triggers resources to accept Gridlets quickly without a significant consideration about the Pareto-optimality in resource allocation. Therefore, there is a chance that some Gridlets with higher resource requirements will fail to obtain their resources. Even BAR generates higher revenue; it is not suitable for the communication cost, which is illustrated in Figure 4.14.

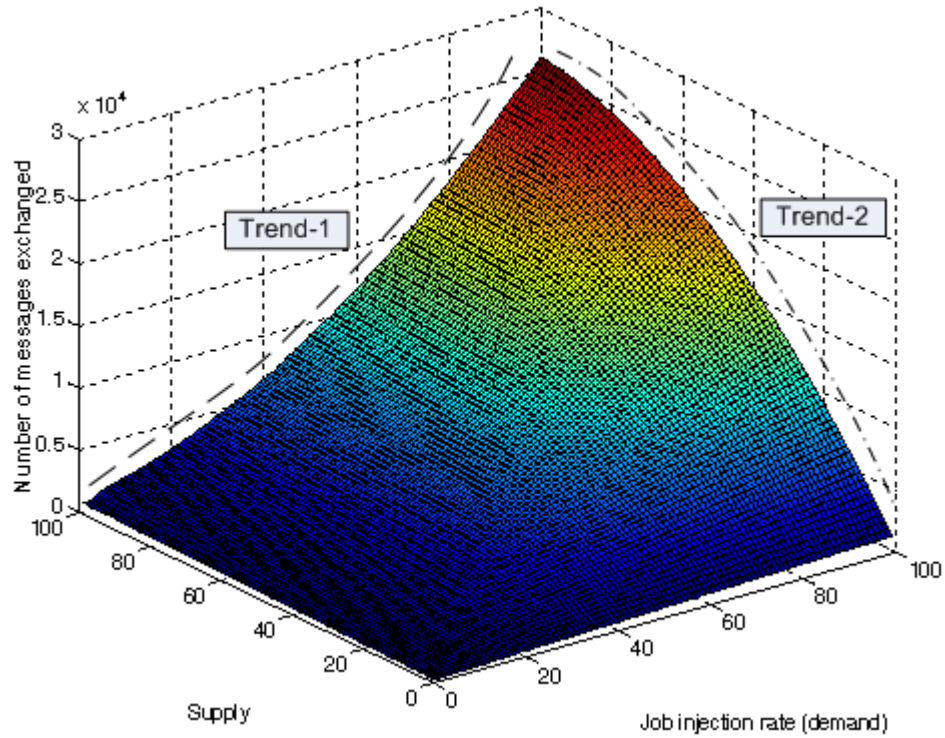


Figure 4.14: Distribution for Communication Overhead

Communication Overhead: Figure 4.14 depicts the distribution of communication overhead for BAR. The communication requirement is higher for BAR compared to that of CMM, due to several rounds in a negotiation process between a Gridlet and a resource. When there is sufficient supply and insufficient demand (Trend-1), the overhead rises slowly. For this region, the competition for the resources is low. Again, in our BAR, the bargaining process starts with the precondition that the resource has the ability to meet the Gridlet's deadline. This removes the unnecessary overhead from the scenarios. In addition, the Pareto-optimality in BAR

prevents the Gridlets from unnecessarily surveying longer in the market. This also helps to keep the number of the messages exchanged low in this region. However, as the demand goes up, the competition also rises and due to more Gridlets in the environment, a higher number of messages have been exchanged. Again, when the supply starts increases, and demand is high (Trend-2), the competition for resources further goes up. The number of messages exchanged for this region is also higher, due to the higher demand.

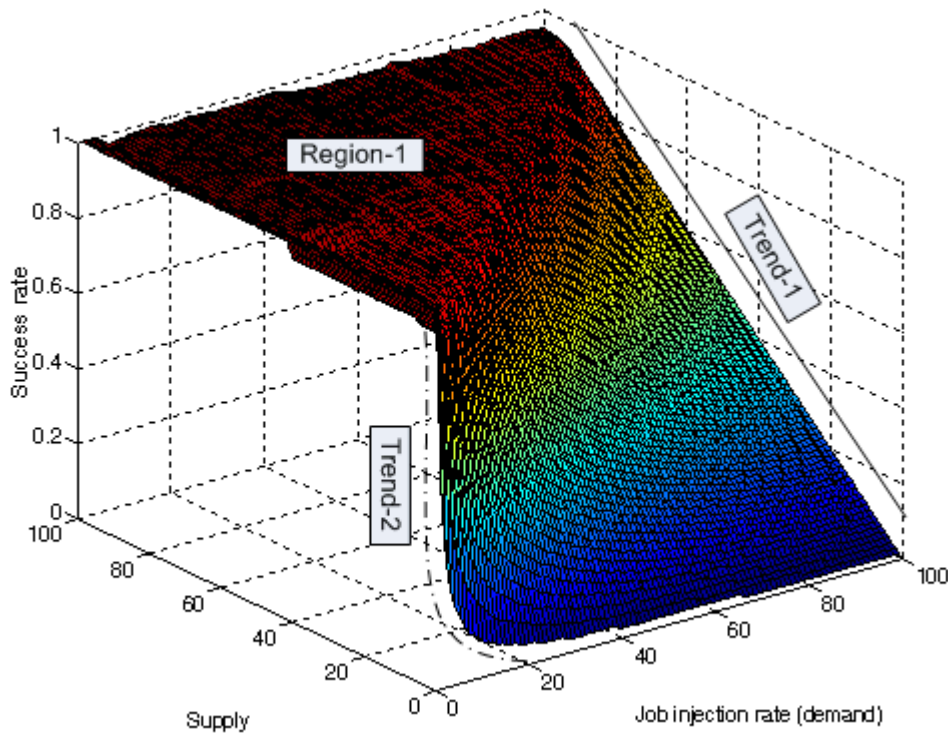


Figure 4.15: Distribution for Success Rate

Success Rate: The success rate for BAR remains almost constant when the supply is high and demand is low (Region-1 in Figure 4.15). Most of the Gridlets are accepted easily, due to the high supply and low demand. However, as the demand increases and supply decreases, the rate decreases proportionally (Trend-1). Most of the Gridlets are rejected in this region, due to the high demand and limited supply, thus minimizing the success ratio. Trend-2 can be explained using the similar explanation given for CMM (Trend-2 in Figure 4.6).

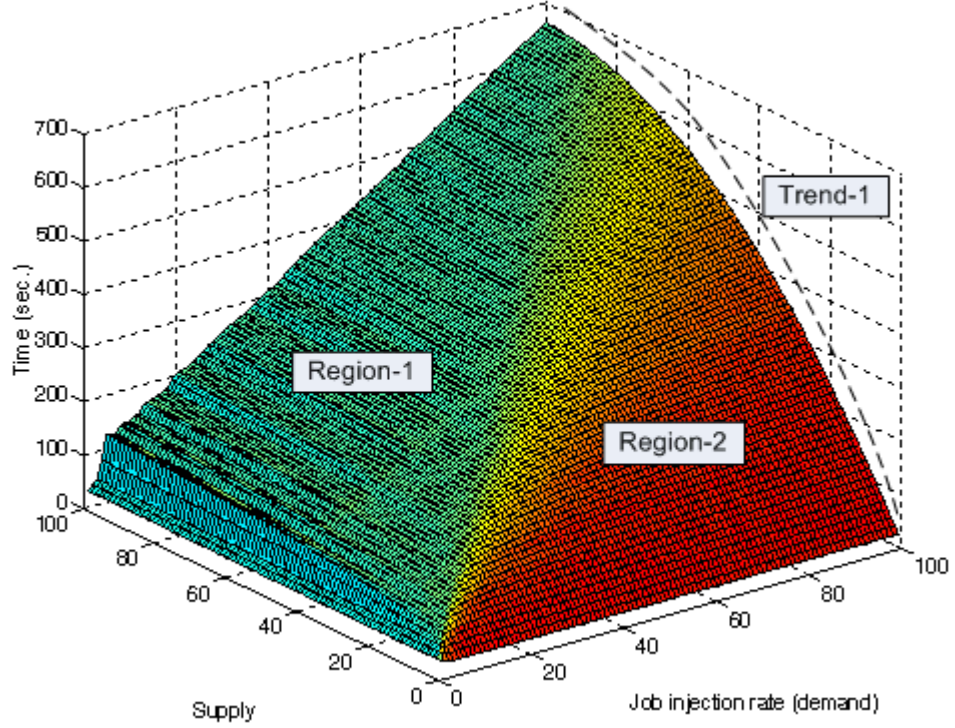


Figure 4.16: Distribution for Average Turn-around Time per Gridlet

Average Turn-around Time: Figure 4.16 illustrates the distribution for average turn-around time per Gridlet. It can be seen that there is no significant difference in timing if we compare it with CMM (Figure 4.7). One can argue that BAR should consume more time for a Gridlet compared to that of CMM, because BAR uses multiple rounds. However, BAR is comparatively suitable for Pareto-optimal allocation, as we have already discovered. Gridlets do not require to travel longer in the market for their suitable resources, due to this optimality. The chance for getting appropriate resources quickly by the Gridlets increases, because the allocation consistency for some Gridlets expedites discovering the suitable resources for other Gridlets. On the other hand, as CMM offers lower Pareto-optimality, Gridlets need to stay longer in the market to find out their suitable resources. Therefore, it is clear that the both models have some advantages and disadvantages, which ultimately helps in achieving a competitive performance for this metric. The Region-1, Region-2 and Trend-1 can be explained using the same explanation given for CMM (Figure 4.7), because they only depend on supply and demand functions.

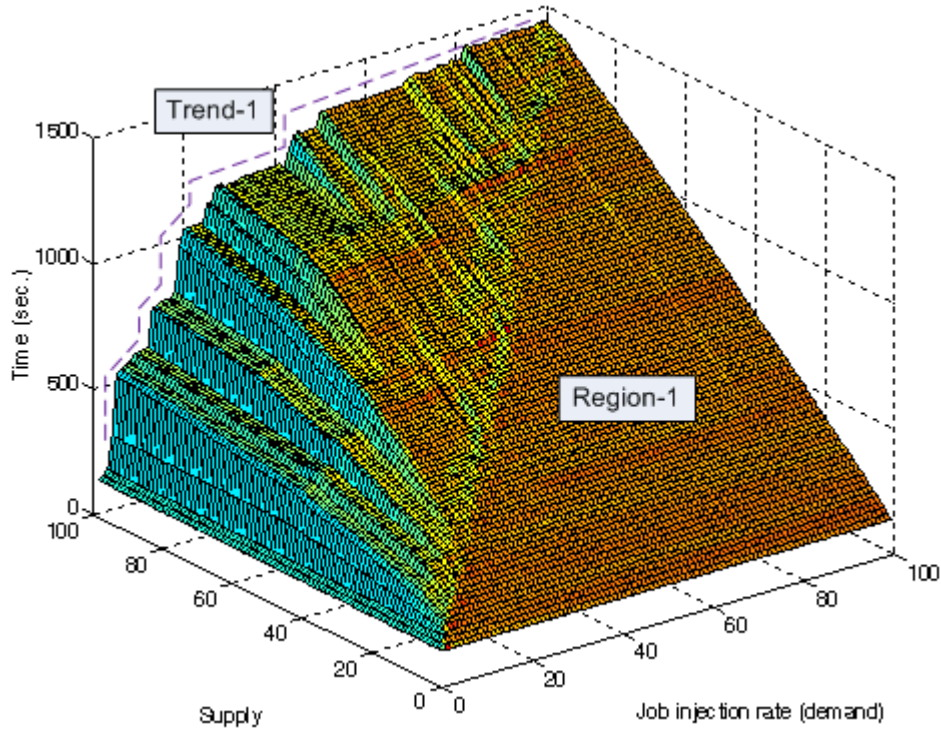


Figure 4.17: Total Simulation Time Distribution

Total Simulation Time: The distribution of simulation time (Figure 4.17) for BAR is similar to that of CMM (Figure 4.8) except the steps stay longer in case of BAR. The similar explanation used for CMM can be applied here also to define the steps (Trend-1). As mentioned earlier, different from CMM, BAR does not have more randomness in allocating resources. The state of the constant position for this region is longer, due to less random nature. However, this is only apparent when demand is high. The chance for staying longer in the market by the Gridlets decreases, due to the high demand and Pareto-optimality. This gives a constant timing even for a range of demand. When the demand is low, the chance for the Gridlets for staying longer in the market is also low. This prevents the Gridlets from staying longer in the same position (in terms of high supply and low demand). Region-1 can be explained as before. As the demand increases and supply decreases, the competition for resources equally increases. The time remains constant across Region-1, due to this tight competition.

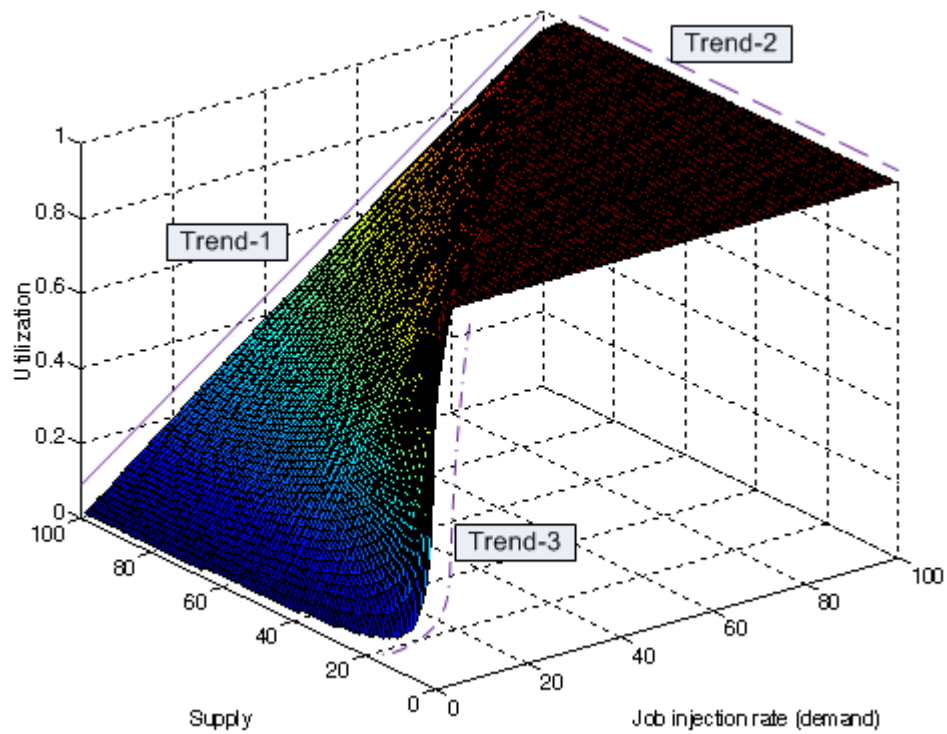


Figure 4.18: Distribution for Resource Utilization

Resource Utilization: Figure 4.18 demonstrates the distribution for resource utilization. The utilization proportionally increases with demand, and when the supply is high (Trend-1). Because of consistency in resource allocation, the utilization is proportional until the supply and demand reach the highest, which has not been appeared for CMM (Figure 4.9). Utilization during high demand regardless of supply (Trend-2) is the maximum, due to the same reason. As the demand starts decreasing, the utilization also starts decreasing. As there are a fewer demand and supply (Trend-3), the chance for resources to be utilized drastically drops as mentioned in CMM.

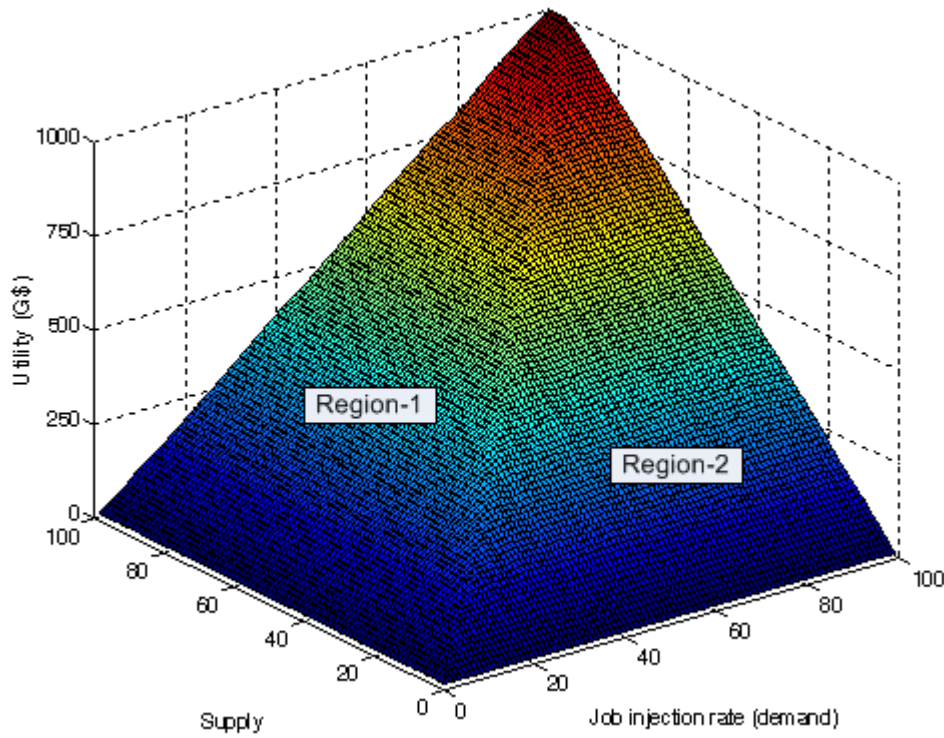


Figure 4.19: Distribution for User Utility

User Utility: The overall user-utility for BAR (Figure 4.19) is lower compared to that of CMM (Figure 4.10). In CMM, resources do not attempt to optimize their utilities, which help Gridlets to optimize their utilities. However, in BAR, resources also try to optimize their utilities along with Gridlets. This prevents the Gridlets from maximizing their utilities in this case. As both supply and demand increase, the utility increases, because the chance for the more users to be accepted also increases. Therefore, the accumulated utility for a particular scenario becomes higher. An explanation similar to the one given for CMM (Figure 4.10) is enough to explain Region-1 and Region-2 in this case.

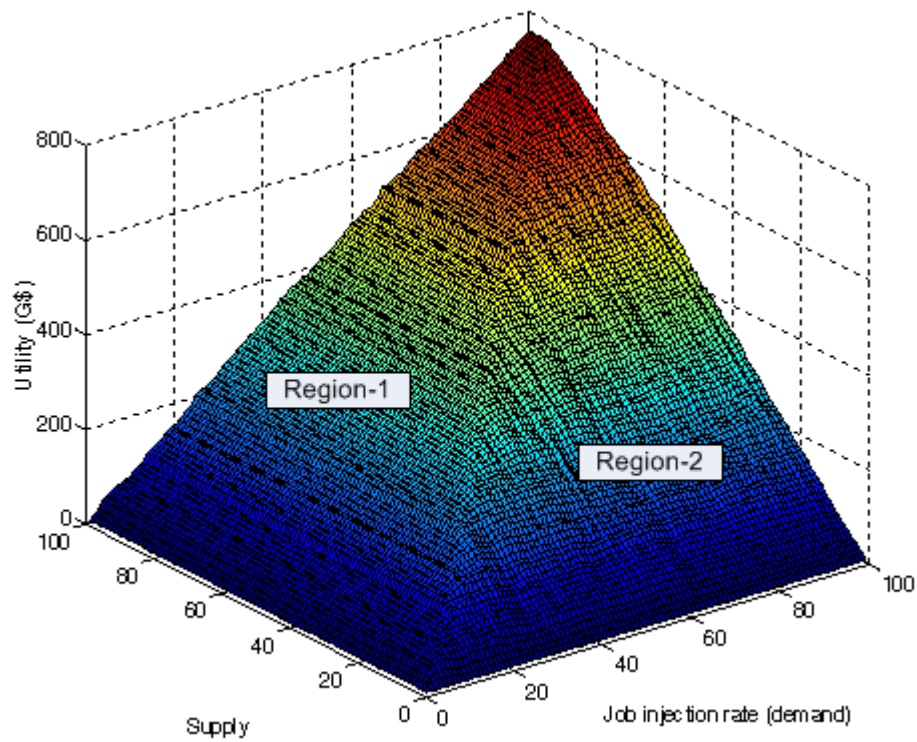


Figure 4.20: Resource Utility Distribution

Resource Utility: Through bargaining, resources in this case, also able to obtain utilities (Figure 4.20). This is consistent with the literature, i.e., BAR allows the utility-based negotiation for both Gridlets and resources. This might attract a wide range of Grid providers to adopt BAR for maximizing their revenue. For a similar reason as explained for user utility, in this case too, Region-2 generates more utility compared to Region-1. As more Gridlets are accepted in Region-2 due to the over-demand, which means, more resources are utilized. As a result, the accumulated utility for resources also increases.

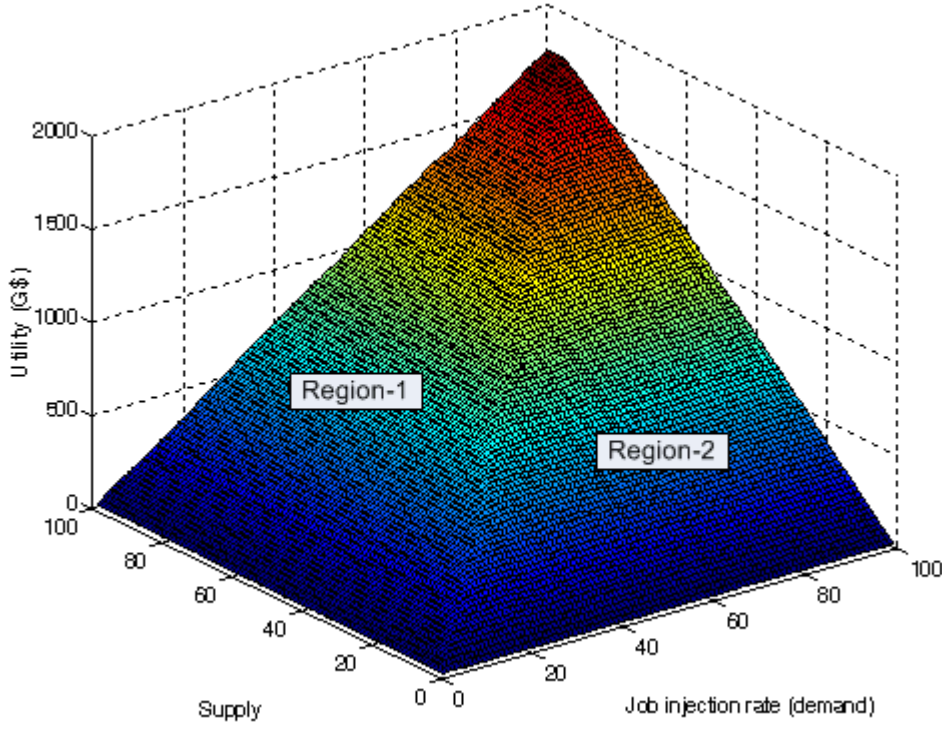


Figure 4.21: Distribution for Social Welfare

Social Welfare: Figure 4.21 presents the distribution for social welfare. As mentioned earlier, the social welfare is given by,

$$\text{Social welfare} = \text{user utility} + \text{resource utility}$$

As resources also contribute to the welfare, the welfare in this case, is much higher compared to that of CMM (In CMM, the welfare is equal to the user utility). As we are adding the utilities for both Gridlets and resources, the plot for the welfare follows a similar trend (for both Region-1 and Region-2) to those of the user and resource utilities. The highest welfare recorded for BAR is about G\$ 1700. In the following section, we contrast among performances obtained for the CMM and BAR, and identify the domains of strength for individual models.

4.6 A Comparative Analysis among the Commodity Models

We observe from Section 4.3 and Section 4.5 that performances for CMM and BAR vary for different scenarios and different performance metrics. Therefore, in this particular section, we identify the domains of strengths for CMM and BAR in terms of the different performance metrics.

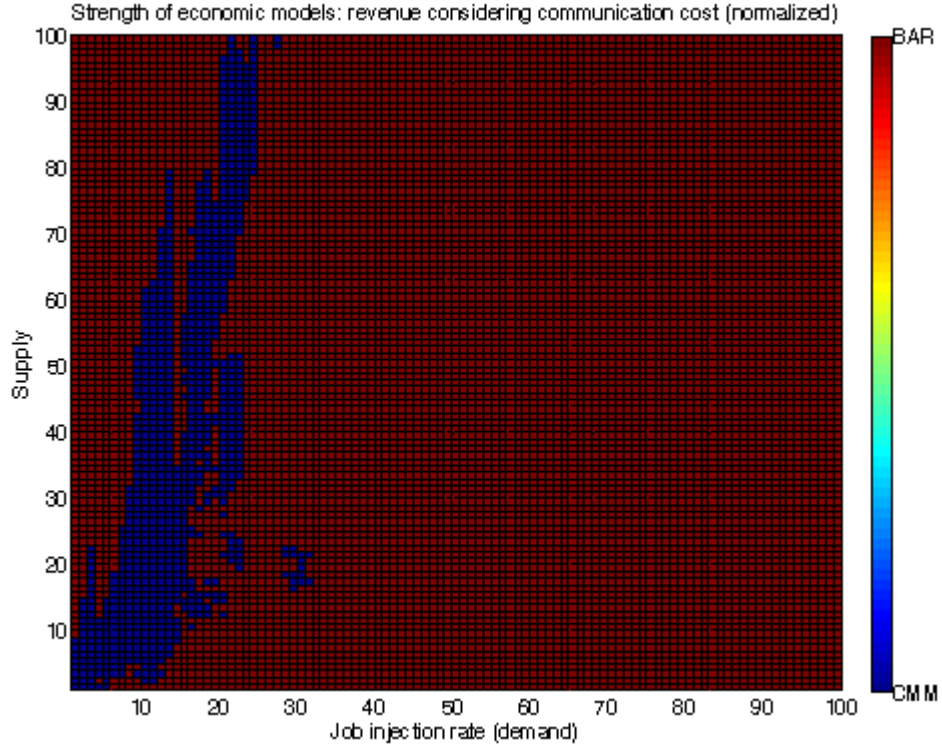


Figure 4.22: Comparison for Revenue over Communication Overhead

Revenue and Communication Overhead Based Comparison: As we are dealing with matrices for the performances of different models, we conduct cell-wise comparison to identify the strength of a particular model for a particular metric. For example, to compare which model generates more revenue and for which scenarios, we compare the matrices for CMM and BAR representing their revenues, and then, plot the results (in terms of Boolean) in a contour diagram. We identify that in terms of revenue, BAR always outperforms CMM (not shown here). However, as BAR has high communication overhead and the relative bandwidth for a Grid might become expensive, we re-evaluate the revenue considering communication overhead. We normalize the respective revenues and communication overheads for CMM and BAR and compute the revenue over communication overhead ratio. If the revenue matrixes for CMM and BAR are represented as $[CMM]_{rev}$ and $[BAR]_{rev}$ respectively, then, their normalizations are given by,

$$normalized [CMM]_{rev} = \frac{[CMM]_{rev}}{\max\{[CMM]_{rev} \text{ and } [BAR]_{rev}\}}$$

$$normalized [BAR]_{rev} = \frac{[BAR]_{rev}}{\max\{[CMM]_{rev} \text{ and } [BAR]_{rev}\}}$$

Similarly, for communication overhead, we get,

$$normalized [CMM]_{commOver} = \frac{[CMM]_{commOver}}{\max\{[CMM]_{commOver} \text{ and } [BAR]_{commOver}\}}$$

$$normalized [BAR]_{commOver} = \frac{[BAR]_{commOver}}{\max\{[CMM]_{commOver} \text{ and } [BAR]_{commOver}\}}$$

Now, as we would like to maximize revenue and minimize communication overhead from an economic point of view, we compute the normalized revenue over communication overhead ratios for both CMM and BAR.

$$[CMM]_{ratio} = \frac{normalized [CMM]_{rev}}{normalized [CMM]_{commOver}}$$

$$[BAR]_{ratio} = \frac{normalized [BAR]_{rev}}{normalized [BAR]_{commOver}}$$

Then, we perform a cell-wise comparison of these matrixes representing the ratios in terms of the normalized metric. The comparison results are shown in Figure 4.22.

In most scenarios, BAR outperforms CMM. Even though BAR generates more messages, they are not so much higher compared to that of CMM. For BAR, the number of messages exchanged does not considerably increase with greater supply and demand, due to the Pareto-optimality in resource allocation. This helps BAR to perform better overall. However, when the supply and demand are low, CMM outperforms BAR. The accumulated strength of the Pareto-optimal resource allocation is not high, due to low supply and demand. Because of this, CMM easily dominates BAR using its number of rounds parameter. Therefore, CMM outperforms BAR in this region. However, as the supply increases, and the demand is still low, the performance of CMM starts get worse. The spot prices generated by CMM are low, due to high supply and low demand. This prevents the model from maximizing the revenue in general. This ultimately minimizes the strength of CMM in the high supply and low demand region.

In terms of the communication overhead, CMM always outperforms BAR. Therefore, if a Grid has limited bandwidth, CMM would be of interest, because it requires lower communication cost.

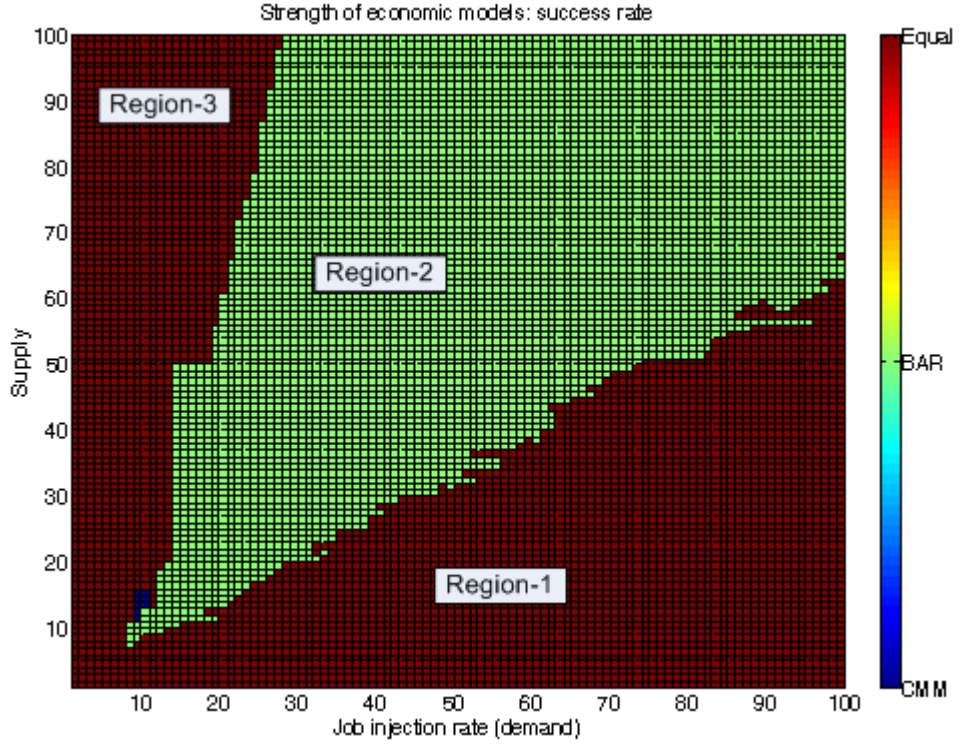


Figure 4.23: Comparison for Success Rate

Success Rate Comparison: Figure 4.23 illustrates the comparison for success rate. There are three different regions in the contour diagram. Region-1 denotes the scenario of low supply regardless of the demand. Both models have competitive performance in this region. The only reason for this is the higher competition over the resources by the demand. The similar performance can be seen for Region-3 also. Most of the Gridlets can easily be accepted irrespective of the model used, due to high supply compared to demand. However, when both the supply and demand increases, BAR starts outperforming CMM. For this region, the spot prices generated by CMM are somewhat constant, which does not ensure a better Pareto-optimality in resource allocation. Earlier we highlighted the suitability of BAR for Pareto-optimal resource allocation. The accumulated strength for the optimality also increases, due to high supply and demand. This helps BAR to accept more Gridlets in the market.

As we can directly observe from Figure 4.7 and Figure 4.16 that there is no significant difference in terms of average turn-around time per Gridlet for CMM and BAR, we ignore this comparison.

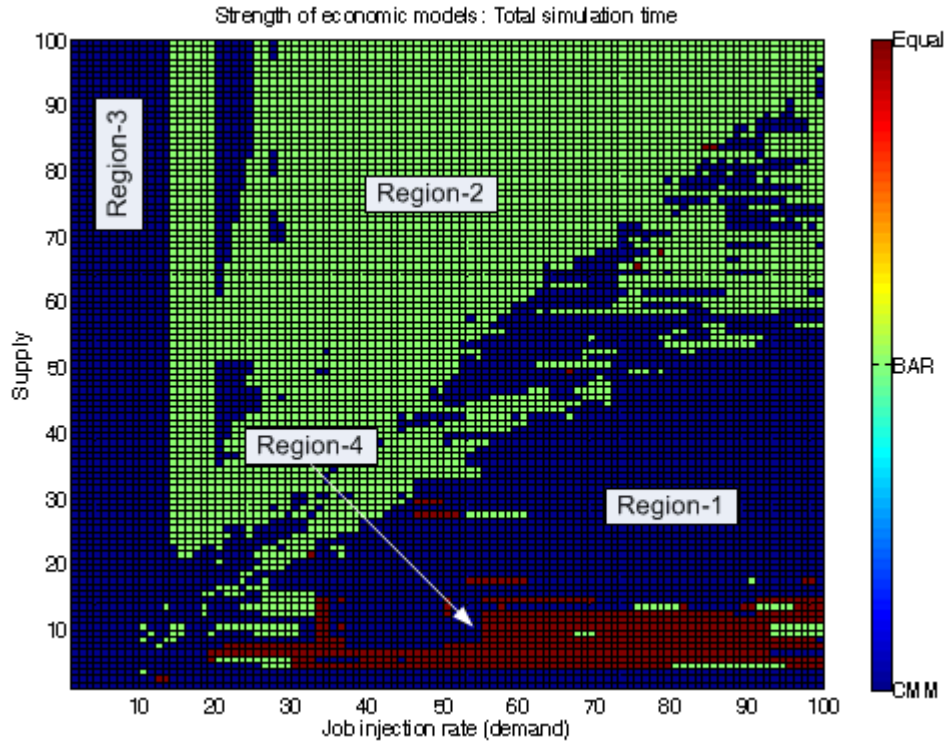


Figure 4.24: Comparison for Total Simulation Time

Total Simulation Time Comparison: There are four different regions, when simulation time is compared between BAR and CMM (Figure 4.24). When the supply is low and demand is high (Region-1), CMM requires lower time to complete simulations. The spot prices will be higher for this region, due to high demand and low supply. This means the higher budgeted Gridlets are quickly accepted and others are quickly rejected, because the supply is limited here. However, as the supply and demand increase (Region-2), BAR outperforms CMM. In this case, the spot prices change infrequently for CMM, which pushes the Gridlets to stay longer in the market compared to that of Region-1. When the demand is significantly low regardless of the supply, CMM again outperforms BAR. In this region, the Gridlets are accepted more quickly and the Pareto-optimality has little relevance for BAR due to low demand. This causes BAR to require more time in general. In the low supply and high demand region (Region-4), BAR and CMM have equal performance. The reason for such a competitive performance is due to the high competition over resources in this region. This helps both the models to occupy the resources quickly and reject the rest of the Gridlets as a consequence.

In terms of resource utilization, the contour obtained was similar to that of the success rate (Figure 4.23). For user utility, CMM always outperforms BAR, whereas for resource utility and social welfare, BAR outperforms CMM.

From the analysis above, we see that either CMM or BAR is not suitable for all scenarios and for all performance metrics. This demonstrates the compatibility of our experimental findings with our survey on different economic models in Chapter 2, which is, a single economic model is not suitable to cope with every scenario. Thus, one could be interested to use the combination of the models in a Grid in order to optimize his/her objectives.

4.7 Summary and Conclusions

Various market mechanisms are proposed to deal with large scale distributed computing environment such as Grid. Market mechanisms are significant in the Grid, because they describe the problem of successful resource management across many self-interested and self-regulating entities. Economic models play a key role in understanding the value of Grid entities. In this chapter, we developed and conducted simulations on two widely proposed economic models – Commodity Market and Bargaining, and analyzed their performance for a wide range of performance metrics. However, due to the variation in working principles of different models, to achieve expected utility by a single model all the time is hard.

We evaluated the performances of two economic models in distributed resource collaboration. We evaluated them in terms of users, resources and the system's utilities. The experimental findings are compatible with the observation made in the existing literature, i.e., in a highly dynamic environment, a single model is unable to cope with every scenario. For example, we identified that Commodity Market Model is suitable for minimizing communication overhead, whereas Bargaining Model is good for maximizing revenue. We further identified the domains where one model outperforms another model. For example, in terms of an objective model combining revenue and communication overhead, Commodity Market Model dominates some scenarios and some are by Bargaining Model. Table 4.1 provides an overview of these identifications. Realizing such a condition, one might be interested to utilize the potential of the both models in order to maximize his/her various objective functions.

We showed the suitability of adopting more than one economic model to deal with highly heterogeneous and dynamic Grid environment in this chapter. In the following chapter, we discuss three auction-based models – English Auction, Contract Net Protocol and Double Auction to identify their suitability in different scenarios as a part of our evaluation process.

Table 4.1: CMM versus BAR over Different Supply and Demand Regions

Performance metric	Low supply-low demand	Low supply-high demand	High supply-high demand	High supply-low demand
Revenue over Communication Overhead	CMM	BAR	BAR	CMM
Success Rate	BAR	Equal	BAR	Equal
Total Simulation Time	CMM	Competitive	BAR	CMM

Grid Resource Management Using Auctions

This chapter incorporates the study of Grid resource management using auction protocols. A historical overview of auction-based resource trading is given at the beginning. Thereafter, the discussion on working principles and performance analysis of English Auction and Double Auction protocols are presented respectively. The development of Contract Net Protocol and its simulative study are also incorporated in this chapter. Later we conduct a comparative analysis among the performances of these models to understand the effectiveness of individual models for distributed resource collaboration in Grid computing.

5.1 Introduction

Auctions are different from traditional commodity-based economic models in terms of organization and pricing methodology. Although the traditional commodity models are more common, auctions are also popular due to their long historical background and the ability to evaluate the values of resources in a market (Cassady, 1967). An auction is typically regarded as a non-cooperative game, where individual participants possess their individual strategies to win the game through competing for a resource. Internet opens the door for distributed resource trading and makes the games more competitive than might otherwise be thought. As the Grid consists of a large number of self-interested entities worldwide, auctions become a potential method to provide motivation for the Grid resource providers to contribute their resources in the Grid.

Although the auction is generally popular among the traders, it has many forms and many purposes of using. A few of the widely discussed auction models are English, Dutch, Sealed-bid, Vickrey, and Double. However, in Grid computing, English and Double Auctions are the most widely discussed models (Haque et al., 2011). English Auction (ascending-bid) has become popular in the Grid due to its suitability for maximizing revenue for providers (Buyya et al., 2005, Beck et al., 2008). At the same time, it has some drawback in terms of communication overhead. A central Auctioneer, in this case is responsible for putting the resources up for the auction and receiving bids from potential bidders (users). The Auctioneer keeps increasing the bids over rounds, and finally declares the highest bidder as winner. We

present an English Auction (EA) protocol suitable for trading Grid resources and analyze the model's effectiveness in terms of distributed resource collaboration.

Double Auction (DA) is another popular economic model in the Grid. The model is highly distinguished from other models due to its ability to handle a large number of users, immediate resource allocation, and finally the price formation process (Tan and Gurd, 2007, Grosu and Das, 2004). In such an auction setting, the auction happens among many users and many providers. The Auctioneer receives bids (application parameters) from potential users and asks (resource parameters) from potential providers, and performs the matchmaking process. The suitability of DA for various performance metrics is analyzed in this chapter. We further explore the Grid resource management using Contract Net Protocol (CNP). An extensive research has been conducted on this protocol and has identified its potential in terms of a significant understanding about user QoS (Thabet et al., 2011, Paurobally, 2010). In such an auction form, the users invite bids from potential resource providers. Once the users receive the list of potential providers, they decide which provider to choose (meta-scheduling), depending on their preference criteria. Thus, users here are able to optimize their individual preference functions through the meta-scheduling process.

We further conduct a comparative analysis among the performances of these three auction models. The analysis demonstrates the insufficiency of a single model to deal with every scenario in the Grid. Finally, we evaluate and identify the regions of strengths of the models in terms of various performance metrics. The rest of the chapter is organized as follows.

Section 5.2 presents a historical perspective of auction models. The design and development of EA-based Grid resource management is discussed in Section 5.3. The section includes the description of activities of different auction parties, their communication model and the procedure for conducting multiple auctions concurrently. Section 5.4 analyzes the performances of the model. The definition, working behavior and pricing methodology of DA is explained in Section 5.5. The performance analysis for the model is presented in Section 5.6. The working behavior of the participants and the meta-scheduling process of CNP are discussed in Section 5.7. Section 5.8 does the simulative study for CNP. A comparative examination among the performances of these auction models is presented in Section 5.9. Finally, Section 5.10 presents the summary and conclusions of the chapter.

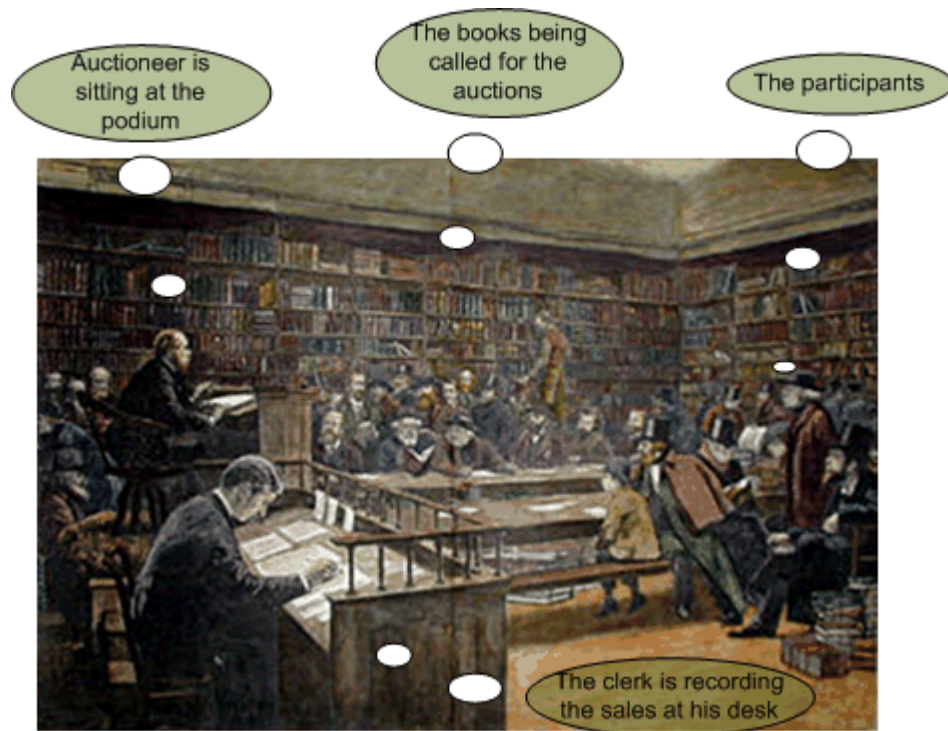


Figure 5.1: Sotheby's Book Auction, London, 1888

5.2 Auctions: A Historical Background

There is a high chance that English Auction is the oldest from the available auction variants; because the word “*auction*” itself has been derived from the Latin word “*augere*”, which refers to “*to increase*” or “*to augment*” (Krishna, 2002). The history of auctions roots back in 500 B.C. when the Greek historian Herodotus first reports the use of an auction. The ancient auctions were typically used to sell less attractive women for marrying, property or estate goods or some time to sell the goods acquired by soldiers (Cassady, 1967). In spite of the strong historical evidence of auctions, the early auctions were only conducted in some specific times or events. Auctions are usually conducted when the value of a particular good (resource) is unknown to the seller.

The earliest history of modern auctions was recorded in the Oxford English Dictionary in 1595. The major auction houses were created in the early 18th century. Sotheby's is one of the oldest houses in the world created in 1744, which is still in operation. Figure 5.1 illustrates a book sale at Sotheby's auction room. The graphic is reported at London in 1888, and the source is at <<http://www.georgeglazer.com/prints/business/bookauct.html>>. Over years, auctions have moved on to trade commodity goods (fruits, vegetables, and fish) and have become more frequent (Cassady, 1967). The development of the Internet technology has further raised the rapid deployment of auctions considering millions of bidders around the world, as stated in <<http://www.economist.com/node/226168>>. *eBay* is the most widely

adopted online auction at present. Apart from *eBay*, there are *uBid*, *Amazon*, and *Yahoo!* also offer auctions. Therefore, the distribution of resources has found a new and an attractive way to be utilized.

As the auctions become popular over times, economists around the world have began exploring the auction structure, bidders, pricing rules, and in essence, the auction standards (Milgrom and Robert, 1982, Parkes and Ungar, 2000). It has been identified by many researchers that auctions are suitable for distributed resource collaboration among autonomous and self-interested entities (Parkes and Ungar, 2000, Xing and Lan, 2009). As the auctions are conducted among self-interested entities such as Grid entities, the mechanisms are often referred to as non-cooperative games where each individual tries to maximize his/her own utility function. The independent private value auctions are broadly adopted in the subject (Milgrom and Robert, 1982). Therefore, in this chapter, we model our auction bidders according to the private value auctions. In a private value auction, a bidder is considered as risk-neutral and knows the value of the auction only to himself, however, do not know the value to others. In the following section, we describe the development process for EA-based Grid resource management.

5.3 A Framework for Grid Resource Management Using English Auction

Auction protocols describe the behavior of the participants in the auctions and analyze the properties of the auction markets. It also studies competitive bidding strategies and market throughputs such as revenue. This section discusses the design and development process for EA-based Grid resource management framework. Depending of the structure, auctions can broadly be categorized into two kinds – One-sided and Double. We will explore the Double Auction mechanism in Section 5.5. In a One-sided Auction, multiple bidders compete for a single resource. Again, based on the movement of the price (ascending or descending) and the roles of buyers and sellers, One-sided auctions can be classified into two ways; Forward Auction (ascending-bid) and Reverse Auction (descending-bid). In an ascending-bid auction such as Forward English Auction (FEA), multiple buyers compete for a single resource through increasing their bids over rounds. On the other hand, in a Reverse Auction such as Dutch or Reverse English Auction, the roles of buyers and sellers are reversed. In this case, multiple sellers compete to service a single buyer and continuously lower their prices over rounds until they reach up to their reservation prices. English Auction model has the both variants. However, the FEA or typically known as EA is more popular in the Grid literature (Buyya et al., 2005, Beck et al., 2008). One of the possible reasons of this popularity is the

suitability of EA for generating higher revenue for providers. Thus, we design and develop the framework for EA. The following sub-section describes the key roles involved in the framework.

5.3.1 Understanding Different Auction Parties

This sub-section presents an overview of the participants and their general activities in the auction. The user-model remains unchanged for almost all the economic protocols. Therefore, the parameters used to describe a user in the earlier models (CMM or BAR) remain the same also for EA. Broker needs to deal with the auction on behalf of a user. Each broker is assigned in a particular group for an auction. The detailed group formation process is described in sub-section 5.3.3. In a group, one broker competes with other brokers to get the resource for which the auction has been set. The auction is typically conducted over multiple rounds. As mentioned earlier, we use the concept of private value auctions, in which each broker knows the value of the resource only to itself but do not know the values to others; therefore, a broker actively keeps bidding in the auction until it knows that it has reached up to its private value (budget) of the resource. The broker regards this value as optimal reply or optimal value. If the auction further continues after this, the broker is termed as a loser, because other brokers have made the auction continue by bidding higher than that the broker has. However, if the auction finishes at this stage, the Auctioneer checks its reservation price and selects the broker as winner. In case of failure, the broker can again participate in another auction and continue the competition. To maintain consistency with the models discussed in the previous chapter, we adopt the idea of *Exclusive-or* bid for the brokers in the EA, that is, a broker cannot obtain more than a single resource (Parkes and Ungar, 2000).

The resources, in EA, have no significant role to play, because the Auctioneer works on behalf of the resources. An auction can be single-attribute or multi-attribute according to the literature. In a single-attribute auction, bidders compete for a single item/resource whereas in a multi-attribute or combinatorial auction, bidders compete for a set of items in combination. It has been identified that the latter is more efficient for distributed resource management (Parkes and Ungar, 2000). In addition, as the Grid applications require a set of resources (OS, MIPS, memory, storage etc) together, the combinatorial auction is more suitable for the Grid rather than having multiple individual auctions for each of the items. Even currently, we use only MIPS parameter to define a resource; we assume the concept of combinatorial auction in our work. Before the auctions start, the Auctioneer receives information about the resources from GIS (Grid Information Service). The Auctioneer, then, sets those resources for auctions based on the interests by the brokers on those resources. At the end, the resource follows if it

has been committed for any broker (the winner) by the Auctioneer. Generally, the Auctioneer receives some percentage from resources for conducting the auctions on behalf of the resources. However, for simplicity, we ignore this percentage and assume that the Auctioneer services are also developed from provider-side and, therefore, whatever revenue earned belongs to the resources.

The most crucial party in an auction is the Auctioneer. An Auctioneer typically sells resources on behalf of a resource owner. The Auctioneer is an agent and obeys the responsibilities of acting in the owner's interests. A broker is generally different from an Auctioneer, i.e. , the activity of the broker is ranging from both buying and selling whereas, the activity of the Auctioneer is limited only in selling (Cassady, 1967). The Auctioneer performs all the crucial tasks such as initiating the auction process by inviting bids from potential bidders, conducting the auction process and, finally determining appropriate winner. In the following sub-section, we explain the working model of EA framework.

5.3.2 Design and Development of English Auction

This sub-section presents an event diagram, bid-update procedure by the Auctioneer and winner determination process in EA. Figure 5.2 presents an event diagram for EA-based Grid resource trading. The figure represents the scenario of a single auction process for a single resource. However, in our environment, multiple auctions can occur independently, which we describe in sub-section 5.3.3. At the beginning of the auction, the resource registers with the GIS. The Auctioneer first obtains the resource information from the GIS and requests for competition from potential brokers before starting the auction process. The Auctioneer has its corresponding *Observer* and the Auction (*English Auction*) class itself to help deciding bids over rounds. Each broker has its own *Observer* and *Responder* (not shown in the diagram) to process decision and response to the Auctioneer over rounds. The EA-class is again inherited from *One-sided-auction* and the *One-sided-auction* is inherited from the class *Auction*. The *Auction* class deals with information such as, auction ID, auction protocol, auction bidders and so on., which are common to all auctions. *One-sided-auction* manages information such as the number of rounds, reservation price, winner and so on, which are specific only for such auctions. Consequently, EA-class is more particular about the activities of this auction. The activities include when to start/close the auction, how to increase the bids over rounds and so on.

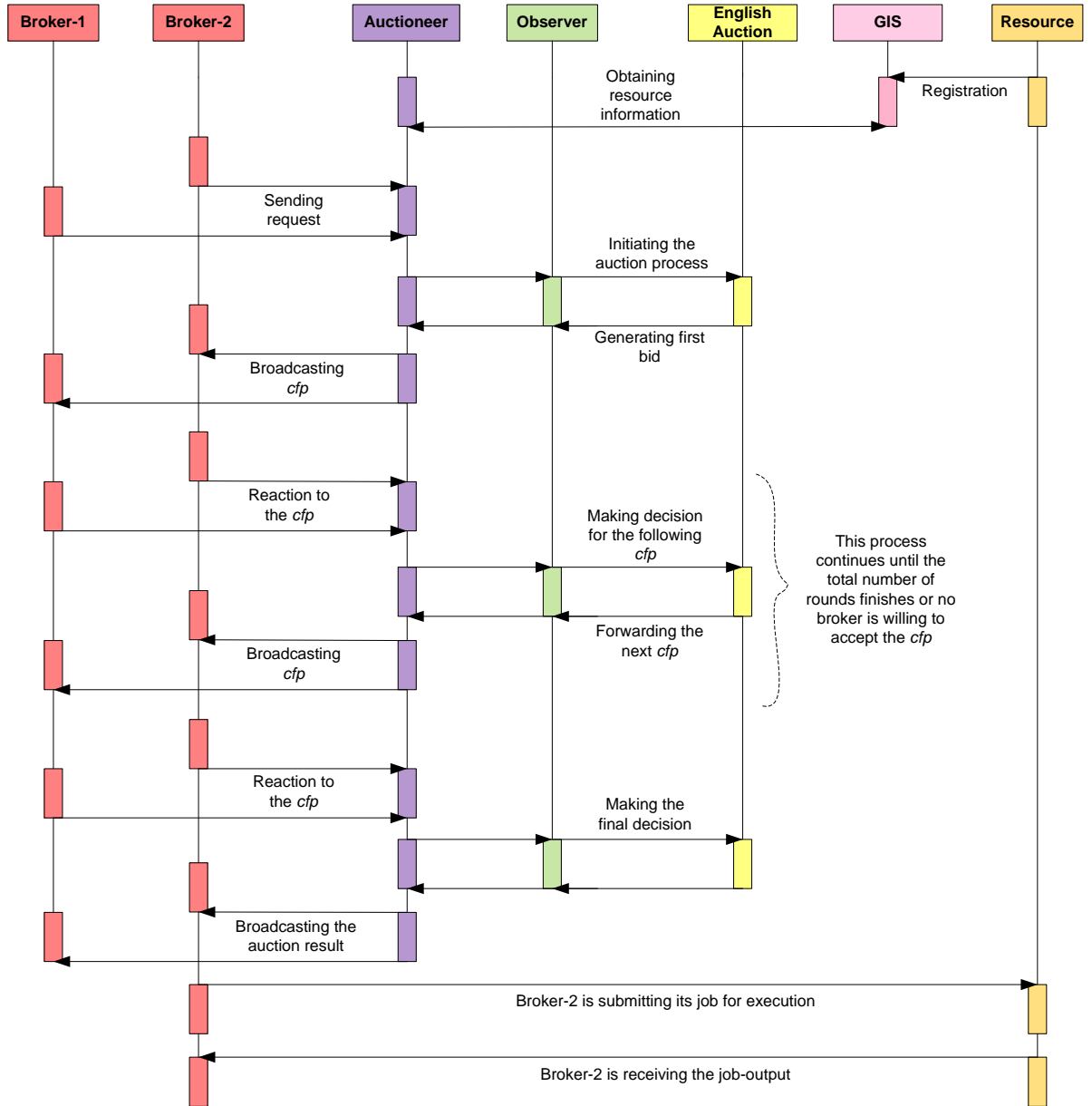


Figure 5.2: An Event Diagram in the English Auction Framework

Upon receiving requests from the brokers, the Auctioneer first generates the bid and broadcasts the *cfp* to the brokers. The first bid is usually a value far below than the current market value of the resource or zero. In our case, we set it zero. Each broker, then, compares its budget with the bid inside the *cfp* sent by the Auctioneer. If the budget is higher or equal to the bid, the broker sends a message (reaction) to the Auctioneer showing its interest. If the budget is lower compared to the bid, the broker triggers a *rejectCallForBid* message to the Auctioneer. In this case, the Auctioneer updates the total number of bidders in the auction. The Auctioneer manipulates bids at every round depending on the current number of rounds. The bid update process by the Auctioneer is given by,

$$cfp\ bid = cfp\ bid + \left\{ \frac{max\ bid - min\ bid}{\theta - 1} \right\} \quad Eq - 5.1$$

Table 3.2 in Chapter 3 provides the values of the parameters in Eq-5.1. The total number of rounds is represented by θ . We set 10 to be the value of θ to maintain consistency with the total number of rounds in Bargaining protocol in Chapter 4.

Winner Determination: This proposal and counter-proposal by the Auctioneer and the brokers respectively continues until the total number of rounds finishes or no broker is willing to accept the current cfp by the Auctioneer. At this stage, the Auctioneer considers the latest bid, which has been accepted by the broker and checks the reservation price (Eq-4.2 in Chapter 4) with the bid. EA in GridSim does not support the concept of reservation price by default. We modify the existing EA-class to support reservation price, because we are using this concept for all the economic models. The ultimate winner is determined based on the reservation price. A broker will be considered as the winner if the following conditions are met.

- The broker that accepts the bid in the latest cfp and
- The bid must satisfy the $cpuCost$ relative to the Gridlet

According to the Figure 5.2, broker-2 has been selected as the winner. Thus, broker-2 is submitting its job to the resource and finally getting the results back from the resource. Broker-1 can take participate again in another auction subject to resource availability. The following sub-section explains the procedure for conducting multiple auction instances concurrently.

5.3.3 Concurrent Auction Procedure

In our simulated environment, there are as many auctions as many resources. Generally, each broker will not be interested in each auction (resource). The decision by a broker to take participate in an auction depends on the broker requirement and the resource capability. Therefore, in this sub-section, we explain the role performed by the Auctioneer in forming multiple groups of interested bidders for multiple auctions and in conducting the auctions simultaneously.

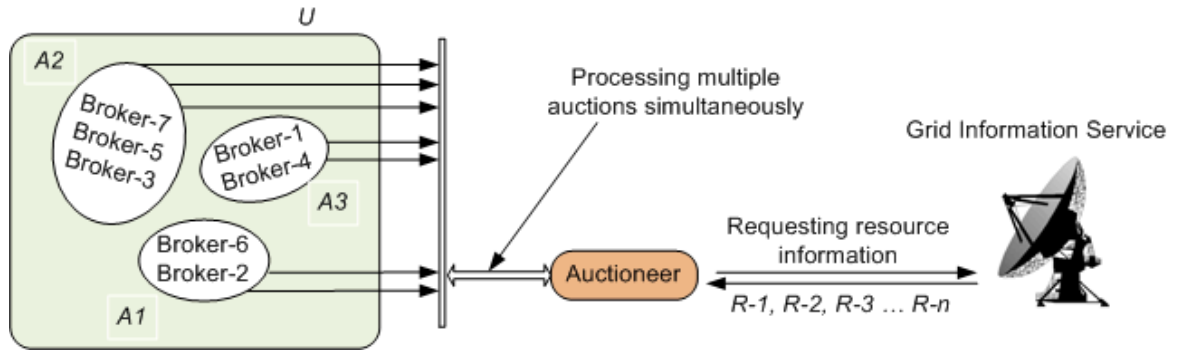


Figure 5.3: Group Formation and Independent Auction Process by the Auctioneer

As we are discussing concurrent auction process, at the beginning of such a scenario, the Auctioneer scans all the broker requirements and resource properties and forms groups. However, in such a scenario, a single broker cannot participate in multiple groups (auctions) at the same time. The Auctioneer first obtains information about all resources from the GIS. As different brokers could choose different resources to compete, the Auctioneer, then, processes the group formation process, that is, which broker would like to compete for which resource. This scenario is illustrated in Figure 5.3, which describes how different brokers have chosen different resources for competition. It can be seen from Figure 5.3 that Broker-1 and Broker-4 have selected node-3 for competition; since they form the group as A_3 . If the set of available brokers in the market is U and the set of interested brokers for a particular resource n is A_n , then,

$$A_n \subseteq U \quad \text{where, } |A_n| \geq 1$$

Let a broker be element of A_n if the respective resource n can meet the relative Gridlet's deadline; because if the resource is unable to meet the Gridlet's deadline, there is no reason by the broker to compete for that resource. Therefore, before start processing the auction(s), Auctioneer groups the brokers depending on the Gridlets' deadlines and resources' capabilities. The *cpuTime* for a particular Gridlet on a particular resource node is given by Eq-4.1 in Chapter 4. If there are multiple groups for multiple nodes, then, each group starts its auction independently with the brokers inside the group. Figure 5.2 illustrates the auction process of such a single group. The next group formation process with the rest of the resources (for which no auctions have yet been set) by the Auctioneer is conducted once all the auctions finish at the first stage. Whenever an auction finishes, the resource list is updated by removing the relative resource's ID from the available resource list. The process continues until there is one potential broker and one potential resource in the market. Figure 5.4 illustrates a conceptual bidding history of a five-round-auction process.

The following section describes the results obtained for Grid resource collaboration using EA.

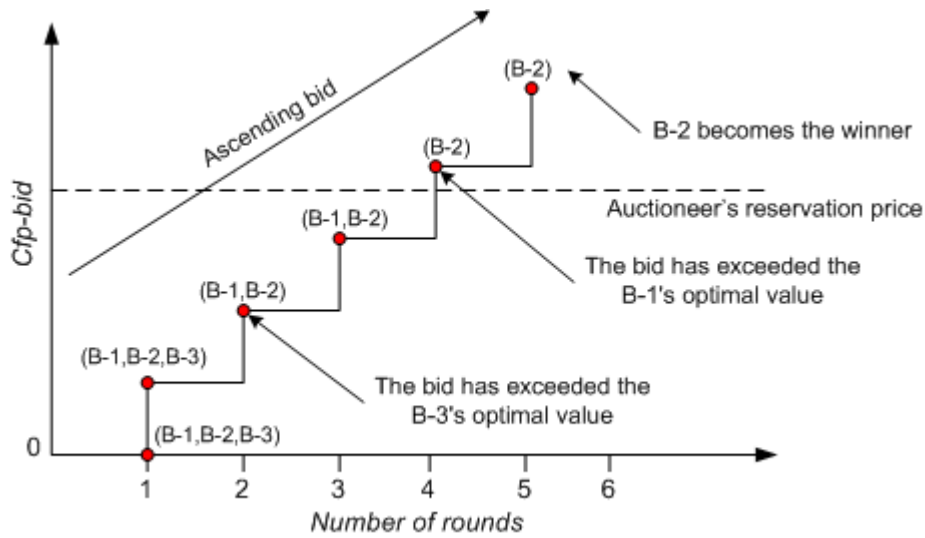


Figure 5.4: A Competitive Bidding History among Different Brokers in English Auction

5.4 Performance Analysis for Grid Resource Management using English Auction Protocol

A detailed description on simulation space has already been discussed in Chapter 3. We use the same simulation space to evaluate the performance of EA. In this section, we analyze the effectiveness of EA for Grid resource management for a wide range of performance metrics. As usual, the simulation is conducted using five different samples and only their average results are presented. To minimize the complexity in the explanation, we use the term “Gridlet” to refer to the user/broker. In addition, the terms Gridlet and competitors will be used interchangeably.

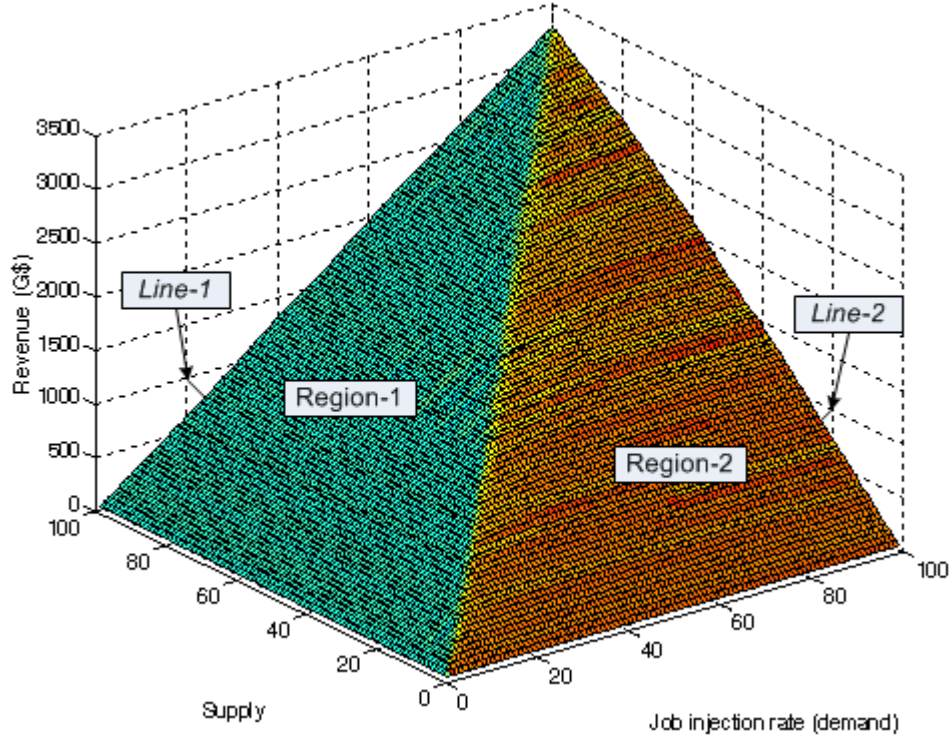


Figure 5.5: Revenue Distribution

Revenue: Figure 5.5 illustrates the revenue distribution for EA. The highest revenue (~G\$3300) is recorded when there are the maximum number of Gridlets and maximum number of resource nodes in the environment. It can be observed that the revenue distribution is very straight forward; because we only plot the aggregated revenue of a complete simulation (a set of Gridlets and nodes). Thus, as both the number of Gridlets and nodes increase, the chance of getting more Gridlets to be accepted also increases; therefore, the aggregated revenue also increases. However, the revenue obtained for Region-2 is higher (*line-2*) compared to the Region-1 (*line-1*). There are two reasons for such a behavior. Firstly, when the demand keeps increasing and the supply is high (Region-1), the aggregated revenue is low because of the low demand. Secondly, due to the high supply and low demand, the average number of competitors for an auction/group is low. This prevents the nodes from maximizing their revenue, because the auctions are finished quickly in this case due to the limited Gridlets per auction. On the other hand, when the demand is high and supply keeps decreasing (Region-2), all the nodes are sufficiently utilized (refer to Figure 5.10), which leads to rise the revenue. In addition, due to the high demand, in this region, the average number of competitors per node increases. This causes the auctions to be conducted for longer. As a result, only the higher budgeted Gridlets become the winners. Therefore, the aggregated revenue increases in general.

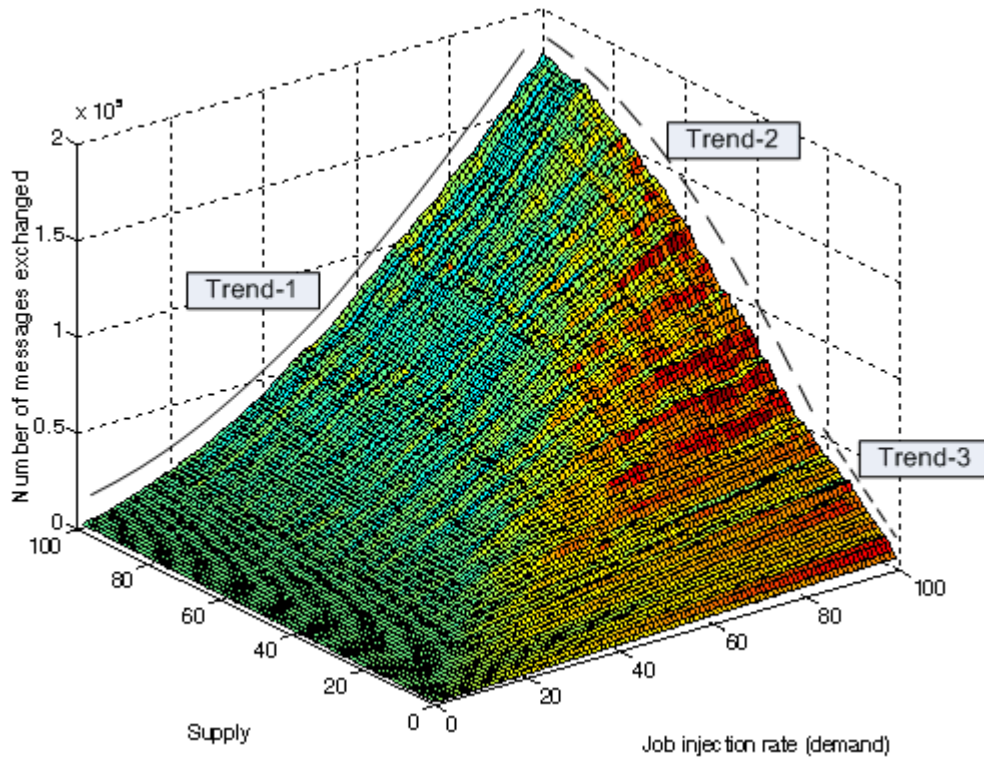


Figure 5.6: Distribution for Communication Overhead

Communication Overhead: The distribution for the number of messages exchanged in EA is depicted in Figure 5.6. Trend-1 describes the scenario when the supply is high and the demand grows from 1% until 100%. However, when the demand is low ($<20\%$), the trend is almost constant. The average number of Gridlets per auction is also very low, due to the very low demand. This fails to utilize the total number of rounds (10) of an auction, which ultimately prevents from increasing the total number of messages exchanged per auction. As the demand increases, the number of Gridlets per auction also increases. This results in moving up the overhead. The overhead is comparatively high in case of Trend-2 and Trend-3 than the Trend-1. Because of the high demand and low supply (Trend-2), the chance that a particular auction can be conducted with a higher number of competitors increases. This increases the probability of utilizing the total number of rounds, which causes the auctions to be longer. This eventually causes to generate a higher number of messages in the simulations. As the supply keeps increasing (Trend-2), the Gridlets get a chance to participate in other auctions in case of failure. This causes to raise the overhead in a non-proportional mode. Again, when the supply is further high and the demand is still high, the market possesses almost equal competition per node. This again results in decreasing the overhead slightly compared to the down of the Trend-2.

However, when the demand is high and supply is lower Trend-3), the trend proportionally increases. Trend-3 describes the scenario of highest competition and due to the very low

supply, the Gridlets cannot participate in further auctions, thus causing the Gridlets directly to be terminated. This maintains a proportional performance across the trend.

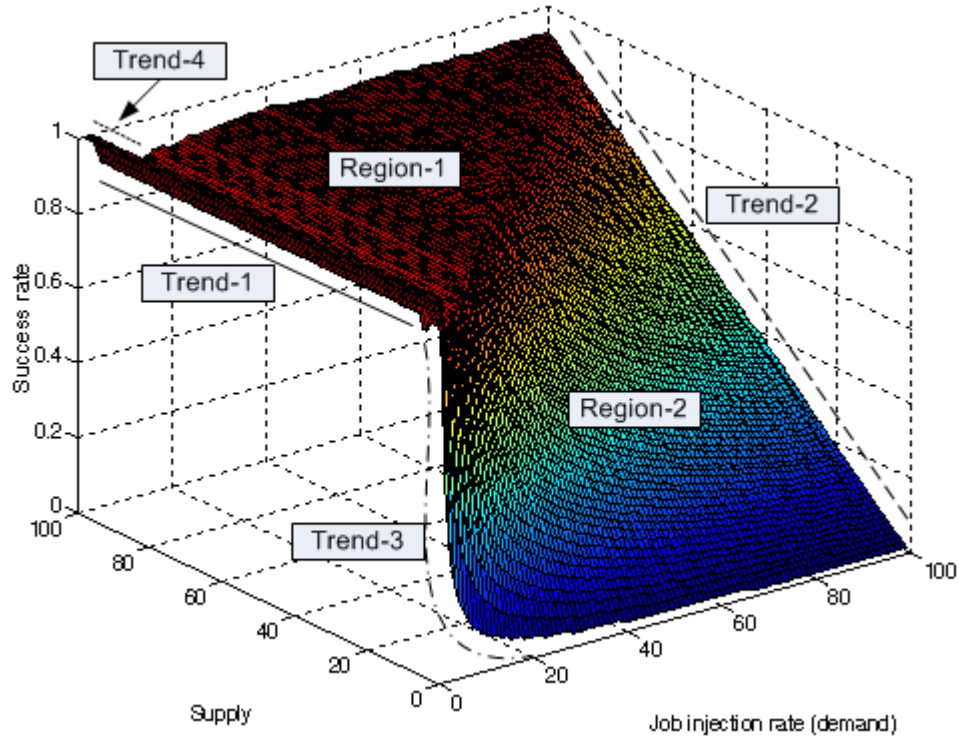


Figure 5.7: Distribution for Success Rate

Success Rate: Figure 5.7 describes the performance pattern obtained for success rate of EA. The highest performance is recorded when the demand is low regardless of the supply (Trend-1). In this case, the Gridlets can easily obtain resources due to the abundance of supply. As the demand increases, and the supply is still high, the rate suddenly drops and continues until the demand reaches the highest (Trend-4). The reason for this sudden-drop can be explained using the reservation price. As the demand slightly increases from Trend-1, the possibility of variation in the requirements of the Gridlets also increases. This variation results the Gridlets in participating in a wide range of auction groups (refer to \mathcal{A}_n in Section 5.3.3). This leads to the average number of competitors somehow decrease per group. Therefore, reaching up to the reservation price by the limited Gridlets in an auction becomes harder, thus causing a sudden declination in the performance (Trend-4). However, as the demand increases, the chance for reaching up to the reservation prices increases and the rate remains almost constant across Region-1.

As the supply starts decreasing (Region-2), the rate gradually falls down. When there is equal supply and demand (the upper middle part of Figure 5.7), the overall competition for an auction is equal. This leads to maintain the constant performance along the region. As the

supply starts decreasing and the demand is high (Trend-2), the success rate proportionally drops. As the supply decreases, the competition proportionally increases for an auction; therefore, the rate becomes completely dependent on the number of nodes in the market. When the supply is extremely low and the demand is also low (Trend-3), either meeting the reservation price or exploring the market by the Gridlets further become impossible. Therefore, the rate abruptly drops in this region.

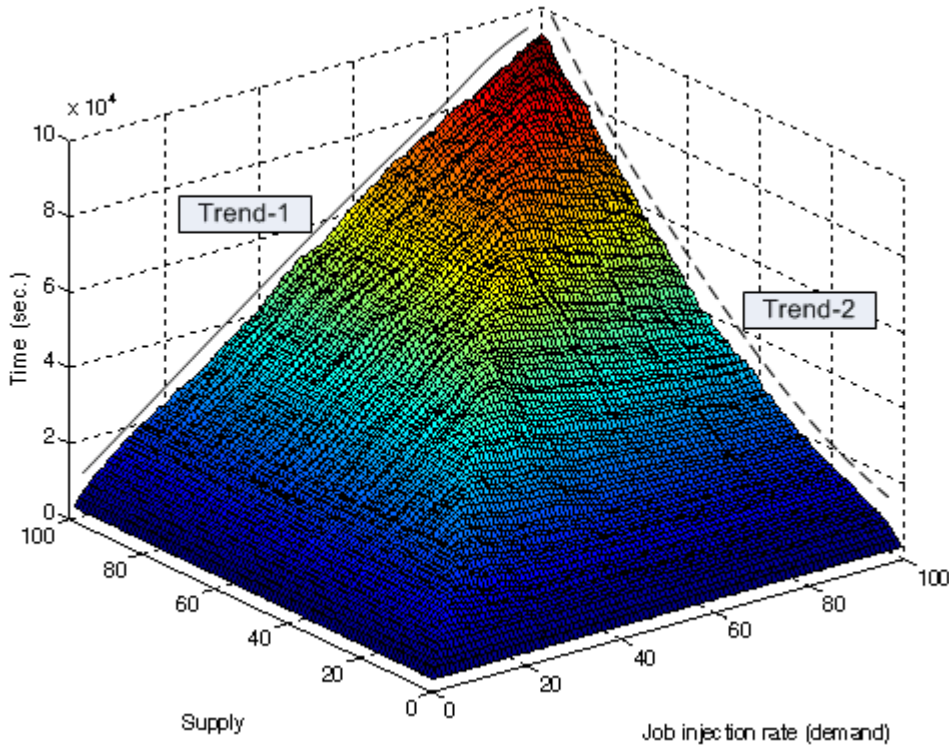


Figure 5.8: Distribution for Average Turn-around Time per Gridlet

Average Turn-around Time per Gridlet: The time required to receive the ultimate decision about its acceptance or rejection from the environment by a Gridlet is presented in Figure 5.8. There are two different trends in this figure. When the supply is high and demand keeps increasing (Trend-1), the time is comparatively higher than when the demand is high and supply is low (Trend-2). In terms of Trend-1, due to the high supply, average competition for an auction is low. This leads a set of auctions to be failed and to be continued further auctions with the rest of the nodes. The number of auctions increases in this case, due to the high supply. This scenario results the Gridlets in staying longer in the market and in continuing competition for the auctions. Therefore, the average time increases for a Gridlet to obtain its final decision.

On the other hand, when the demand is high and supply keeps decreasing (Trend-2), the average number of Gridlets per auction increases. This helps the auctions to be succeeded quickly and easily. In addition, in case of auction failure, the Gridlets possess limited chance to

explore the market further due to the low supply, which leads the Gridlets to be rejected quickly from the environment. This shortens the average turn-around time for individual Gridlets.

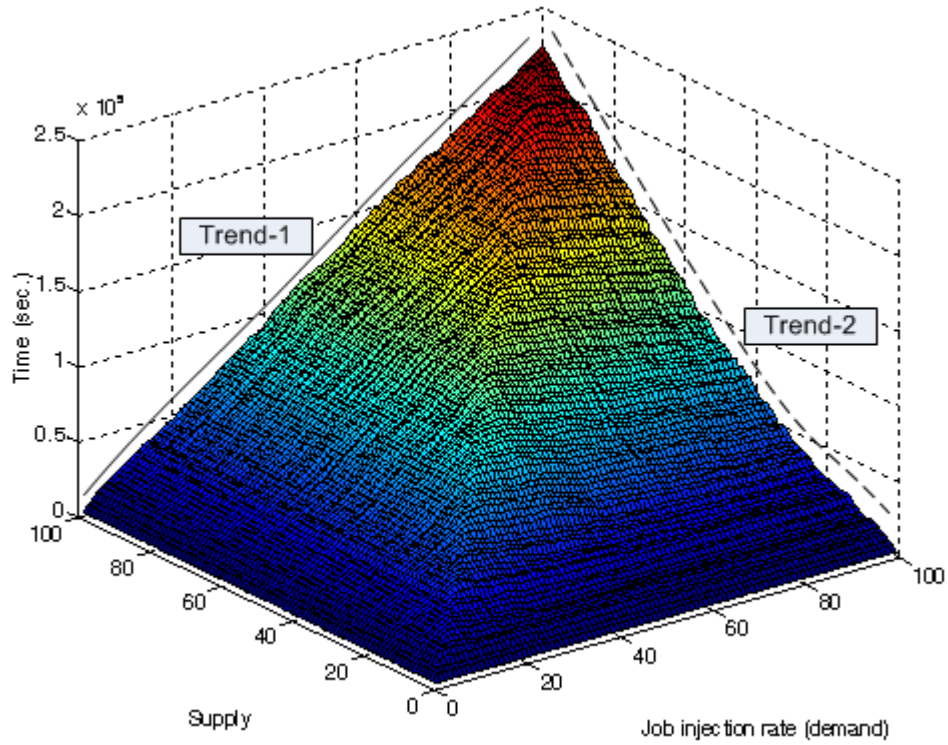


Figure 5.9: Total Simulation Time Distribution

Total Simulation Time: Figure 5.9 depicts the distribution for total simulation time. The pattern is almost similar to Figure 5.8. Therefore, a similar explanation can be given for this figure also. When the supply is high and demand keeps increasing (Trend-1), due to the lower competition per auction, auctions are likely to be failed. This compels the Gridlets to go for further auctions and thus longer the average simulation time. For Trend-2, the competition per auction increases due to the high demand and low supply. The finishing time for an auction, therefore, increases due to the higher competition. However, the Gridlets in this case cannot explore the market longer due to the limited supply. This turns down the overall simulation time for Trend-2.

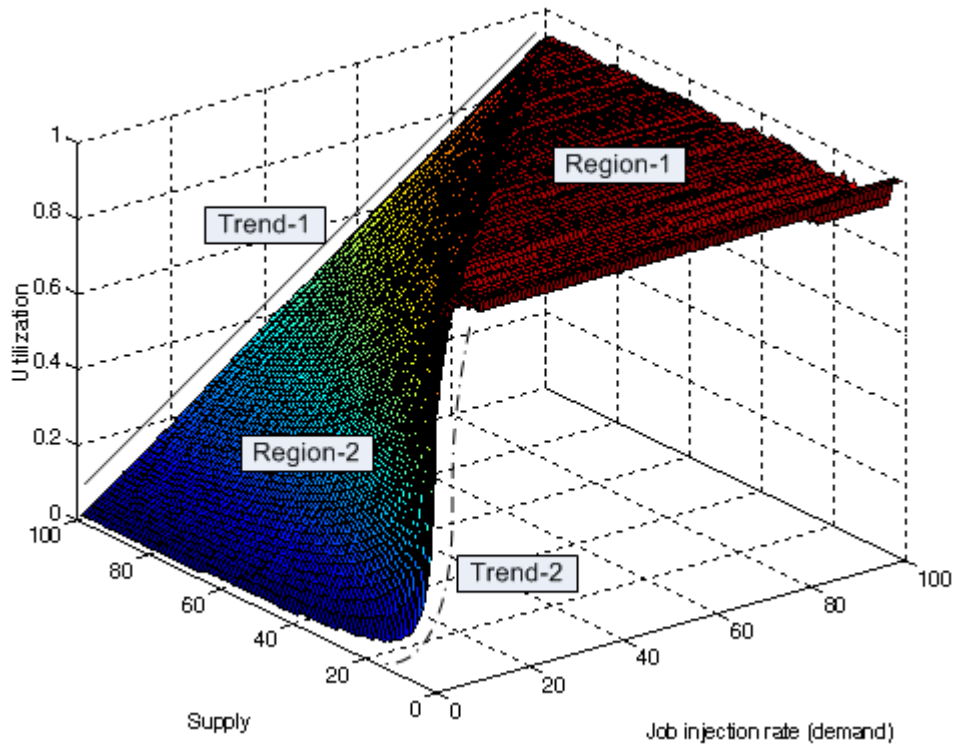


Figure 5.10: Distribution for Resource Utilization

Resource Utilization: The performance of EA for resource utilization is presented in Figure 5.10. When the supply is high regardless of the demand (Trend-1), the utilization rate proportionally increases. The rate completely depends on the number of competitors per auction, due to the high supply. If the number of competitors per auction increases, the rate increases and vice versa. However, as the supply decreases and the demand is low (Region-2), the utilization drops dramatically. The competition per auction decreases, due to the low demand. This unable to reach up to the reservation prices of the auctions and causes the auctions to be failed. However, when the demand is high and the supply keeps decreasing (Region-1), the utilization is almost constant across the region. In this region, the number of Gridlets per auction is higher due to the high demand. This helps the auctions to be succeeded and due to the limited supply, almost all the resources are utilized. This leads to the highest utilization in this region. On the other hand, when the demand is low and supply is also low (Trend-2), reaching up to the reservation prices by the few Gridlets becomes impossible. This causes the abrupt declination to the performance.

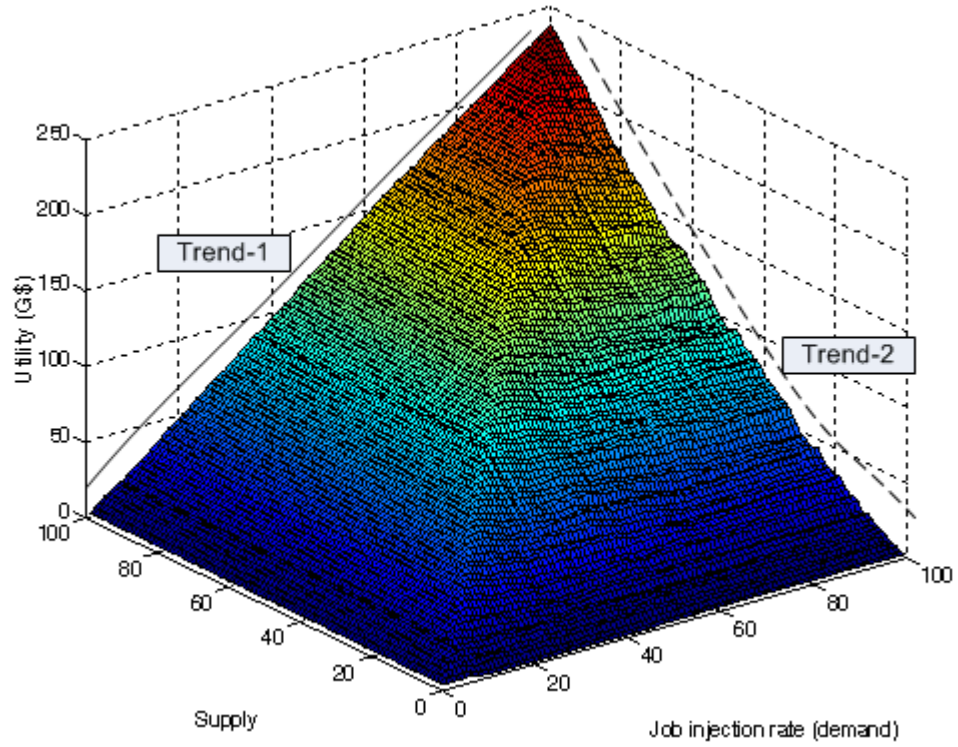


Figure 5.11: Distribution for User Utility

User Utility: Figure 5.11 demonstrates the distribution for user utility in EA. The trend is similar to the simulation time (Figure 5.9). It can be observed that the utility for Trend-1 is higher compared to that of the Trend-2. When the supply is high and demand keeps increasing (Trend-1), the auction-groups are formed with a limited number of Gridlets. Therefore, due to the lower competition per auction, the Auctioneer cannot extend the auction process. This results in finishing the auctions with lower prices. Thus, the Gridlets get a chance to maximize their utilities. In other words, EA is not suitable to utilize the Gridlets' budgets in this region. On the other hand, when the demand is high and supply keeps decreasing (Trend-2), the competition per auction increases. This helps the Auctioneer to auction out the nodes with higher prices. Each Gridlet has to compete through utilizing their budgets for longer, due to the higher competition. In most of the cases, the Gridlets, in this case, requires to reach up to their optimal reply in their respective auctions. Therefore, it becomes harder for the Gridlets to increase their utilities in this particular region.

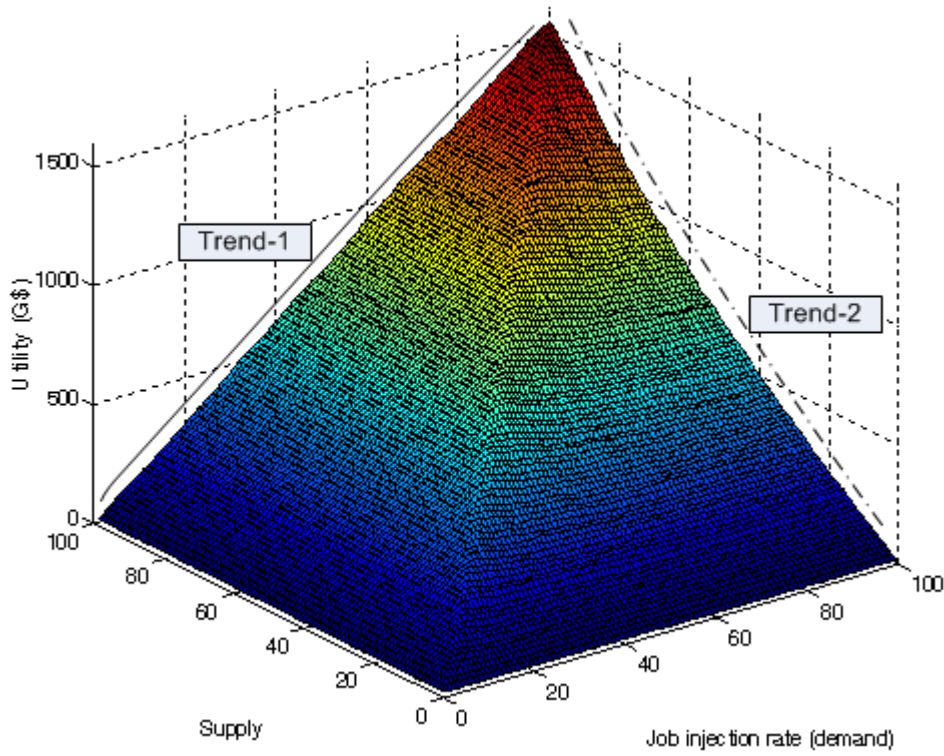


Figure 5.12: Resource Utility Distribution

Resource Utility: The distribution for resource utility is illustrated in Figure 5.12. The trend of Figure 5.12 is similar to that of the Figure 5.11. However, the overall magnitude for the resource utility is much higher compared to that of user utility. If we look at the structure of EA (Section 5.3), it can be observed that the protocol is suitable for maximizing resource utility. The Auctioneer, in this protocol, let the Gridlets compete for nodes through increasing the bid-values over rounds; therefore, the nodes are likely to be sold with far greater than their original reservation prices are. This helps the protocol to maximize the utility for resources-nodes. When the supply is high and demand keeps increasing (Trend-1), due to the limited Gridlets per auction the winning prices cannot go higher, which should minimize the overall utility. However, due to the high supply, a higher number of Gridlets can be accepted. This maximizes the overall utility in this region. On the other hand, when the demand is high and the supply keeps decreasing (Trend-2), due to the higher competition per auction, the winning prices can reach far greater than their corresponding reservation prices. However, due to the limited supply, a lower number of Gridlets can be accepted, which moves down the overall performance in this region.

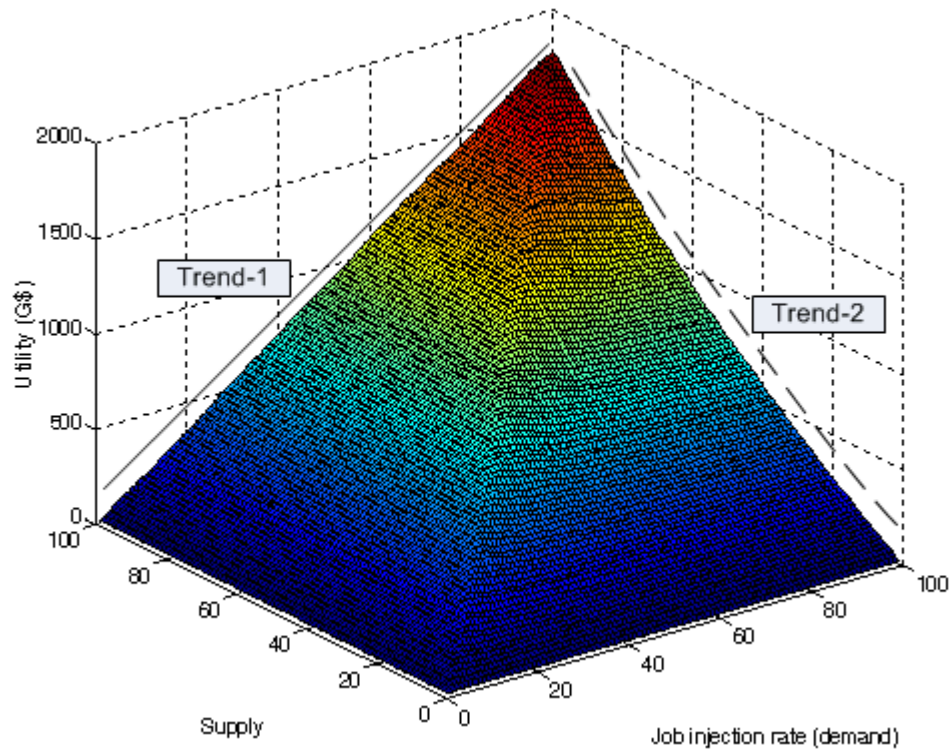


Figure 5.13: Distribution for Social Welfare

Social Welfare: The combination of user and resource utilities (social welfare) is depicted in Figure 5.13. The figure shows the similar trend to the user or resource utility. As the user utility and resource utility follow a similar trend, the social welfare also maintains the trend. Trend-1 and Trend-2 can be explained with the similar explanation given for user and resource utilities. The highest welfare is recorded when there are the maximum number of nodes and Gridlets in the market, which is about G\$1800.

In the following section, the working behavior of CDA and its organization suitable for Grid resource management are discussed.

5.5 Grid Resource Management Using Double Auction Protocol

This section presents the working framework for CDA and the necessary components, which build up the model. In a Double Auction model (DA), the auction takes place among many users and many providers and is conducted by a central Auctioneer. The requests placed by the users are known as *bids* and the properties/availability placed by the providers are referred to as *asks*. The *bids* and *asks* can be placed any time during the auction phase. A real example of DA-based market scenario is New York Stock Exchange (NYSE), as mentioned in <http://www.sci.brooklyn.cuny.edu/%7Eparsons/projects/mech-design/publications/cda.pdf>.

There are two variants of DA in the literature – Periodic DA and Continuous DA (Pourebrahimi et al., 2007). In a Periodic DA, the Auctioneer collects *bids*, *asks* for every specific period, and determines a standard price to clear the market. On the other hand, in a Continuous DA (CDA), the Auctioneer immediately goes for a matchmaking process whenever it receives a new *bid* or a new *ask* from the market. The latter is more popular in the Grid (Haque et al., 2011). The CDA is well-known in Grid computing because of its scalability and immediate resource allocation (Tan and Gurd, 2007). Therefore, we aim to design framework for CDA. GridSim, by default provides CDA. We modify the existing CDA to fit into our framework. For example, we extend the current *Continuous Double Auction* class to support deadline parameter. In GridSim, CDA has been extended from *Double Auction* class whereas *Double Auction* class has been extended from the *Auction* class.

We design the CDA of its most popular form – open cry with order queue (Tan and Gurd, 2007). In this form, the resource costs (*asks*) are generated continuously until the Auctioneer finds a match between a *bid* and an *ask*. The *cost-per-sec* parameter in Table 3.2 in Chapter 3 has been used to generate prices continuously by the resources. Different resources, in this case, use different seed values to generate the prices so that the sequences of the prices for different resources do not match with each other. Outstanding *bids* and *asks* are maintained in an Order Book while *bids* are sorted in descending order and *asks* are in ascending order. Two individual comparators have been used to do the ascending and descending processes. We consider the Zero Intelligence Strategy for defining *bids* and *ask*, which is, the participants in this case, do not attempt to learn or optimize from previous observations. The most crucial part of the protocol, the Auctioneer is described in Algorithm-3.

Algorithm-3: On_Receive_Bid (Synchronized):*Input: bid, ask-Order-Book, bid-Order-Book**Output: match/mismatch, Order-Books*

```

If (size of the ask-Order-Book > 0)
{
    Get the first ask from the ask-Order-Book
    Cast the Gridlet from the received bid
    Cast the node's properties from the ask
    Get budget and deadline from the Gridlet
    Compute cpuTime and cpuCost using Eq-4.1 and Eq-4.2

    if (budget >= cpuCost and deadline >= cpuTime and node-status is free)
    {
        Inform the auctioneer, the respective broker and resource about the match
        finalPrice = (cpuCost + budget) / 2
        Update the ask-Order-Book by removing the ask
        set the node-status busy
    }
    // Keep track of the asks that can never match this bid
    else-if (cpuTime > deadline)
    {
        let Z be the set of node IDs that this bid has dealt with
        if (current node-ID ∉ Z)
            Z = Z ∪ node-ID
        if (Z >= number of available nodes in the market)
        {
            Inform the auctioneer and the respective broker about the mismatch
            Update the bid-Order-Book by removing the received bid
        }
    }
    else add the bid in the bid-Order-Book
}
Else add the received bid in the bid-Order-Book

```

Algorithm-3 is an extended version of the existing CDA in GridSim to support deadline parameter. Algorithm-3 is synchronized, which is, it will not consider a new *bid* to process until and unless it finishes the current *bid* by letting the upcoming *bids* in a queue. A similar algorithm has been designed to handle new *asks*. Unsuccessful *bids* and *asks* are maintained in their respective Order Books. The *bids* are sorted in terms of their budgets in the *Bid-Order-Book* and *asks* are sorted in terms of their *cost-per-sec* parameters in the *ask-Order-Book*. Equal valued *bids* or *asks* are ordered by the time of submission. Final *cpuTime* and *cpuCost* is calculated inside Algorithm-3 by using Eq-4.1 and Eq-4.2 respectively. Whenever there is a match, the Auctioneer immediately informs the message to the corresponding resource so that it can stop generating further *asks*. The Auctioneer also sends a message to the respective broker so that it can submit its Gridlet to the resource.

In the second half of the algorithm, we keep track of the *asks* which are unable to meet the deadline of a *bid*. Once a *bid* receives all the available nodes in Z (set of nodes unable to meet

the deadline of a *bid*), the *bid* will be removed from the respective *Bid-Order-Book* and will be terminated by assuming that there is not a single node, which can serve the *bid*. An ultimate mismatch/rejection occurs at this stage. The Auctioneer, then, sends a message to the corresponding broker stating the rejection. If there is a match between a *bid* and an *ask*, the Auctioneer compute a special price (the *finalPrice* in Algorithm-3) for the both parties, which is half of the combination of the *bid*-price (budget) and *ask*-price (*cpuCost*). We apply this same price formation process as defined in GridSim. Similar methodology for defining the price can also be found in (Izakian et al., 2009, Grosu and Das, 2004, Wiecek et al., 2008). This price is also regarded as equilibrium price for CDA. A sample Order Book suitable for Grid resource trading is presented in Figure 5.14. In the Bid Order Book, Gridlet-125 secured the first position, because it has the highest budget. The subsequent Gridlets are positioned according to their budget values. On the other hand, in the Ask Order Book, Node-7 obtained the first position, because it offers the cheapest price.

The following section discusses the results obtained from CDA-based distributed resource collaboration.

<u>Bid Order Book</u>				<u>Ask Order Book</u>		
<u>Gridlet-ID</u>	<u>Gridlet-length</u>	<u>Deadline</u>	<u>Budget</u>	<u>Node-ID</u>	<u>MIPS rating</u>	<u>Cost per Sec.</u>
125	6557	15	39	7	400	1.2
147	8953	17	38	12	390	1.5
186	7000	14	33	59	420	1.65
205	9500	12	32	44	450	1.90

Figure 5.14: Sample Order Books for Grid Users and Resources in Continuous Double Auction

5.6 Performance Analysis for Continuous Double Auction Protocol

This section describes the outputs of the CDA-based Grid resource management. For example, it discusses revenue, communication requirement, resource utilization, and social welfare as obtained from the simulation. The simulation space remains as before.

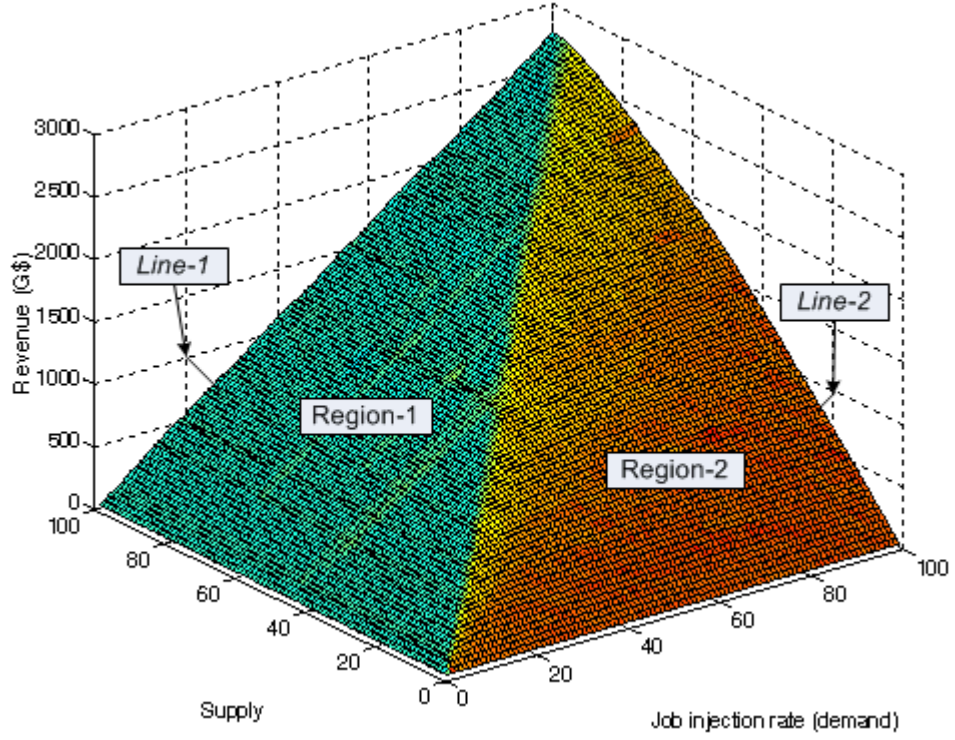


Figure 5.15: Revenue Distribution

Revenue: Revenue obtained for CDA is illustrated in Figure 5.15. The overall revenue for CDA is lower compared to that of the EA (Figure 5.5). The scheduling process for CDA is quite different from that of the EA. In addition, the price formation process (*finalPrice* in Algorithm-3) is also different in CDA. These prevent CDA to be competitive with EA in terms of revenue. There are two different regions in Figure 5.15. The revenue for Region-1 (*line-1*) is comparatively low than that of the Region-2 (*line-2*). When the demand keeps increasing and supply is high (Region-1), firstly, due to the low demand, the aggregated revenue is lower. Secondly, due to the high supply, the *Ask-Order-Book* is able to provide low cost *asks* for longer (refer to Figure 5.14). Again, due to the limited demand, the *Bid-Order-Book* cannot provide a higher number of higher-budgeted *bids*. As the trade executes between the lowest-cost *ask* and the highest-budgeted *bid*, maximizing the revenue becomes harder in this region.

On the other hand, when the demand is high and supply keeps decreasing (Region-2), due to the high demand, the aggregated revenue for a scenario (a set of Gridlets and nodes) is also high. In addition, due to the low supply, the chance for filling the *Ask-Order-Book* with low-cost *asks* decreases. Again, due to the high demand, the nodes are quickly occupied by the higher-budgeted Gridlets; therefore, the trades execute between the higher-cost nodes and higher-budgeted Gridlets. This helps to maximize the revenue for this region. However, as the

supply increases (the peak of the Region-2), the chance for filling the *Ask-Order-Book* with low-cost *asks* increases. This brings the revenue slightly down in this particular region.

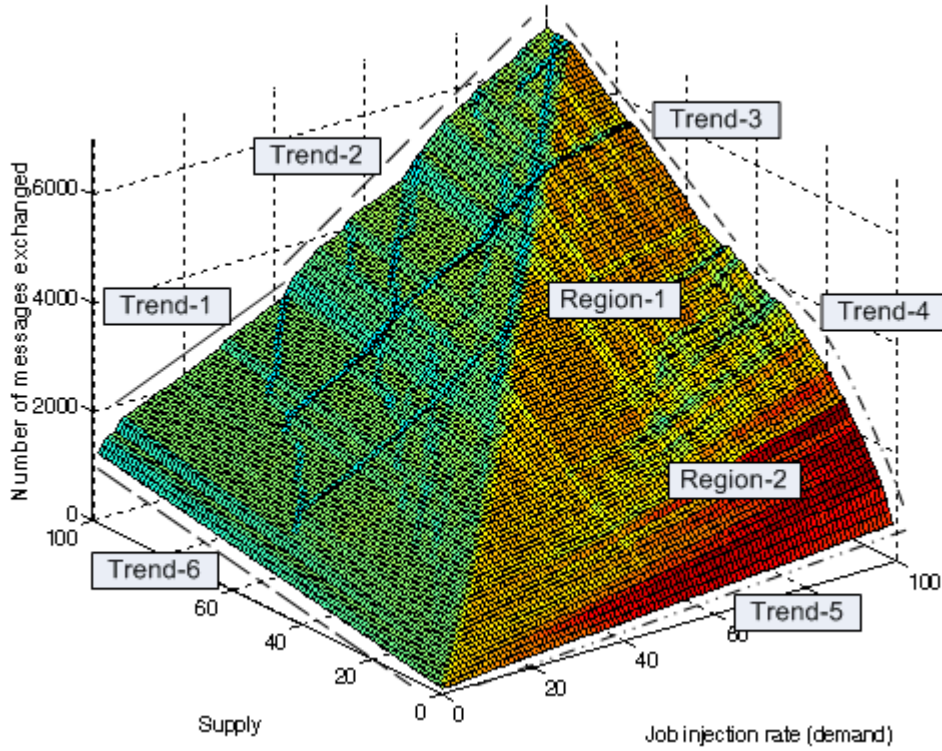


Figure 5.16: Distribution for Communication Overhead

Communication Overhead: Figure 5.16 presents the distribution for communication overhead. One can observe that the overhead for CDA is much lower compared to that of the EA (Figure 5.6). The working methodology of CDA is advised in a way (section 5.5) so that the trades can be executed very quickly. In this process, the *bids* are sorted in descending order while putting the highest-*bid* at first and *asks* are sorted in ascending order while putting the lowest-*ask* at first. Then the trade executes between the first *bid* and the first *ask*. This sorting process helps to clear the market quickly for CDA. Therefore, the total number of messages exchanged in the environment is dramatically lower.

When the supply is high and demand is low (Trend-1), the overhead increases slowly. The limited number of Gridlets is quickly accepted, due to the high supply. This prevents the rest of the resource-nodes from continuously generating *asks*, because there are no more Gridlets in the market. However, as the demand increases (from 40% to above), the trend (Trend-2) increases rapidly. The higher budgeted Gridlets are quickly accepted and left the market with lower-budgeted Gridlets, due to the higher demand; therefore, matching the reservation prices of the nodes by these Gridlets becomes harder, which keeps the nodes longer for generating *asks*. This causes the exchange of a higher number message in this region. As the supply decreases, and the demand is high (Trend-3), the overhead decreases proportionally. Most of

the Gridlets quickly occupy a higher number of nodes, due to the excess demand. This causes to the rest of the Gridlets to be quickly rejected (refer to a similar region in Figure 5.17); because making a final decision by the Auctioneer for these Gridlets takes shorter time due to the limited supply (refer to the second part of Algorithm-3). The scenario is almost flat across the Region-1, due to the same reason.

When the demand is high and the supply is lower (from 42% to below), the figure possesses a kind of exponential trend (Trend-4). Even there is available demand, due to the lower supply, the chance for filling the *Ask-Order-Book* with low-cost *asks* decreases, which unable some Gridlets to meet the reservation prices of the nodes; therefore, the nodes are required to keep generating the *asks* for longer. This causes to a kind of swelling scenario in Region-2. When the demand is high and supply is low (Trend-5), the exchange of messages is mainly due to the submission *bids* by the brokers. The contribution from *asks* is limited for the overhead, due to the very limited supply. However, when the supply is high and demand is low (Trend-6), the contribution is mainly from the *asks*; because the nodes quickly start generating *asks* from the beginning of a simulation until they have been notified that there are no more Gridlets in the market. The similar trends like Trend-5 and Trend-6 also exist in EA (Figure 5.6). The trends were not apparent in Figure 5.6, due to the much higher magnitude.

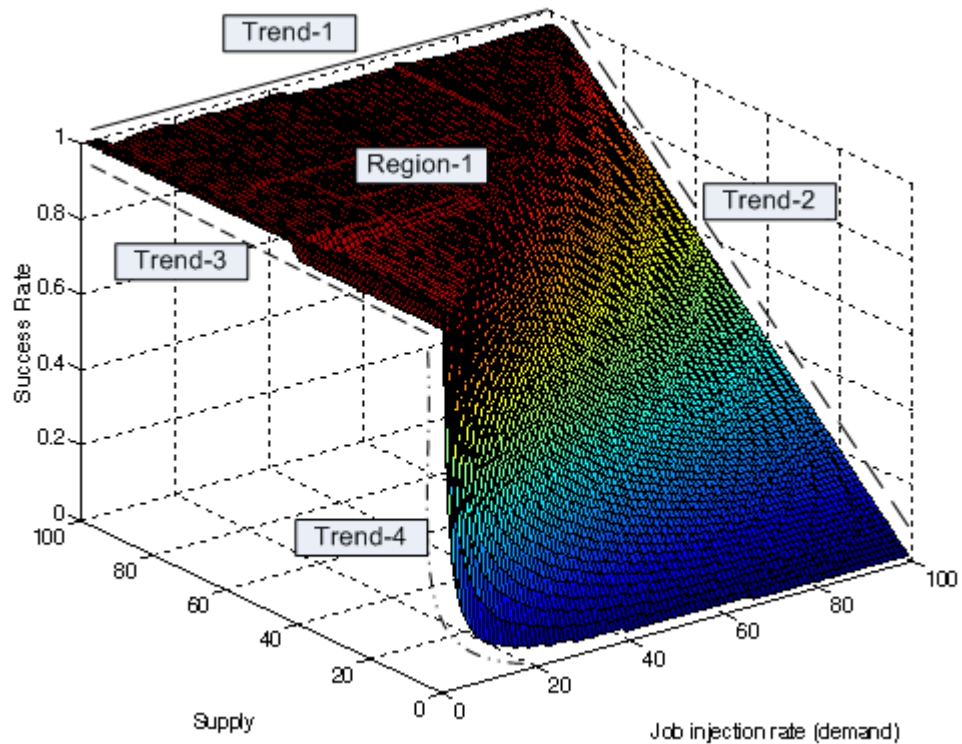


Figure 5.17: Distribution for Success Rate

Success Rate: The distribution of success rate for CDA is depicted in Figure 5.17. It can be observed that the overall success rate for CDA is better than that of the EA (Figure 5.7). EA targets only on higher-budgeted Gridlets regardless of the node performance, whereas CDA's sorting process of *asks* and *bids* helps to bring some Pareto-optimality in resource allocation. As the higher-budgeted Gridlets are matched with lower-costs nodes, acceptance of more Gridlets becomes feasible for CDA.

When the supply is high regardless of the demand (Trend-1), the rate is almost constant. The acceptance becomes easier because of the lower-cost nodes in the respective book, due to the high supply. However, as the demand increases (Trend-1), because of the Pareto-optimality, the performance remains almost constant. When the supply decreases, and demand is low (Region-1), due to the limited supply, there is a slight breakdown in the performance. As the supply decreases, and demand is high (Trend-2), the rate falls proportionally. As mentioned earlier (Figure 5.7), the rate in this case becomes dependent mainly on supply; thus making the trend proportional. Again, when the demand is low regardless of the supply (Trend-3), due to the sorting process of the nodes, the few Gridlets can easily be accepted. Therefore, the rate remains almost constant. Trend-4 can be explained with the similar explanation given for Figure 5.7.

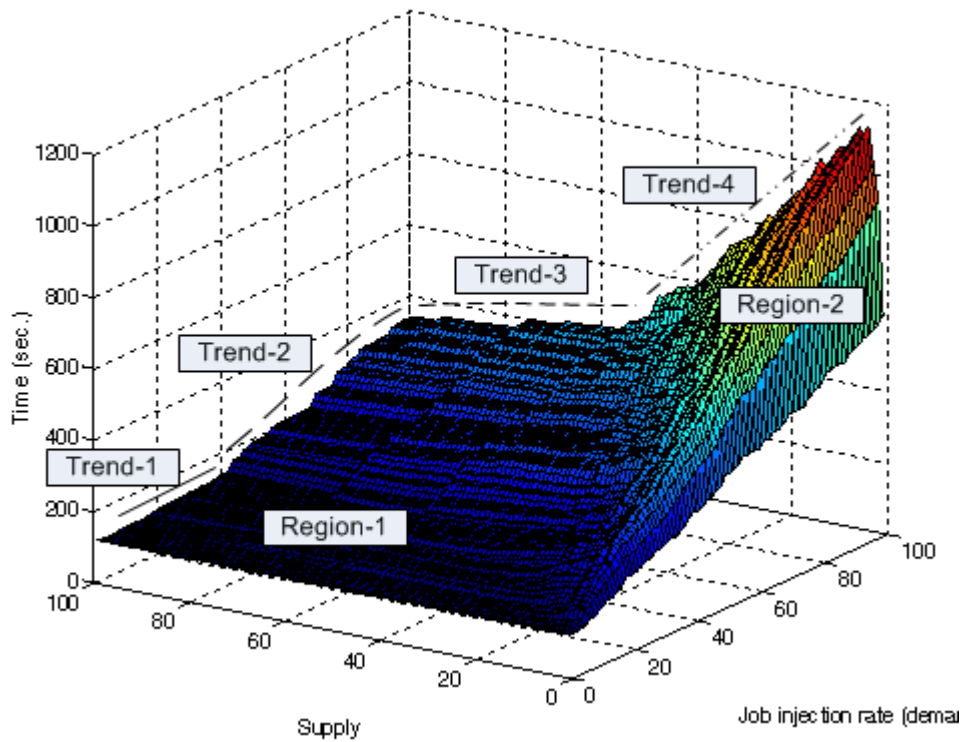


Figure 5.18: Distribution for Average Turn-around Time per Gridlet

Average Turn-around Time per Gridlet: Figure 5.18 demonstrates the scenario for average turn-around time per Gridlet. When there is available supply in the market, the overall time is dramatically lower. As the supply decreases, the time increases. When the supply is high and demand is low (up to 40%) (Trend-1), the time is almost constant. The *Ask-Order-Book* is able to provide a higher number of low-cost *asks*, due to the high supply. This helps the few Gridlets to be accepted very quickly; therefore, there is no significant difference in timing across the region (Region-1). However, as the demand increases (Trend-2), some lower-budgeted Gridlets are remaining in the market for which, the nodes are requiring to generate *asks* for some time more. This raises the time a little bit.

As the supply decreases, and demand is high (Trend-3), the time increases slowly. Most of the higher-budgeted Gridlets are quickly accepted, due to the high demand; because the supply is moderately high in this region. Because of the moderately high supply, the chance of getting low-cost nodes moderately increases. However, the moment supply further decreases (Trend-4), the chance of filling the *Ask-Order-Book* with low-cost *asks* decreases. This helps only to the sufficiently higher-budgeted Gridlets to be accepted whereas, for the lower-budgeted Gridlets (the major part in this case), the rest of the nodes continuously generate *asks*. In other words, the time to settle down the decision for this higher number of Gridlets with the nodes becomes longer (refer to the second part of Algorithm-3).

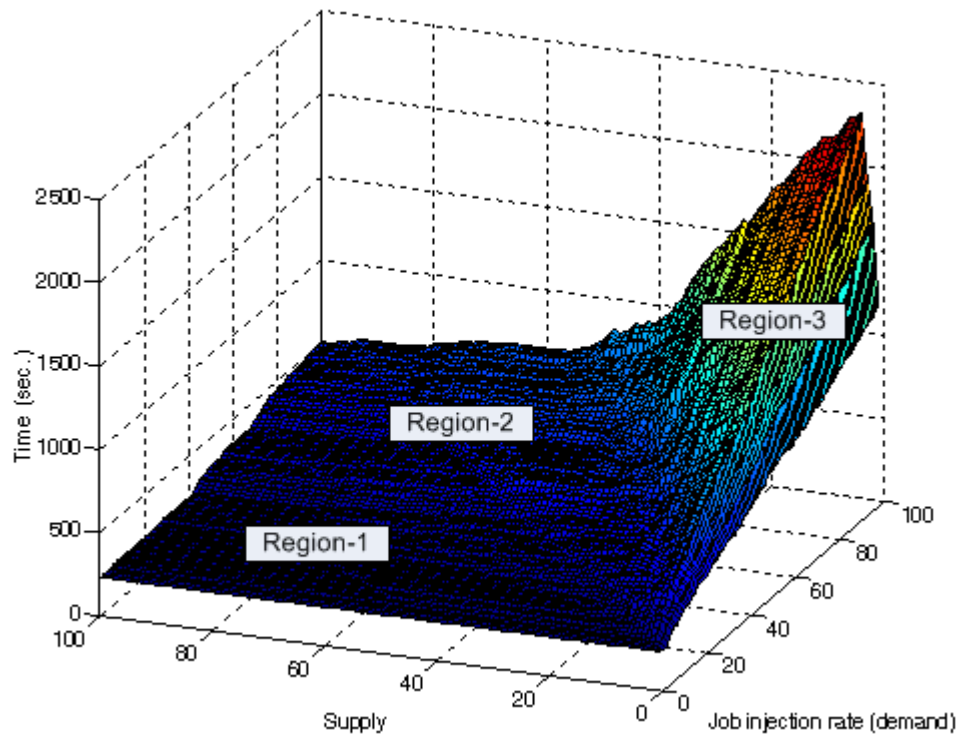


Figure 5.19: Total Simulation Time Distribution

Total Simulation Time: Figure 5.19 presents the distribution for total simulation time. This can be observed that in this case the pattern is similar to the Figure 5.18. Therefore, a similar explanation can be imposed for this figure also. The overall magnitude is higher for the total simulation time than that of the Figure 5.18. Region-1 describes the scenarios where Gridlets are able to be accepted quickly. Region-2 takes slightly longer time, as the competition increases due to the additional demand, whereas Region-3 causes the longest time due to insufficient low-cost nodes in the market.

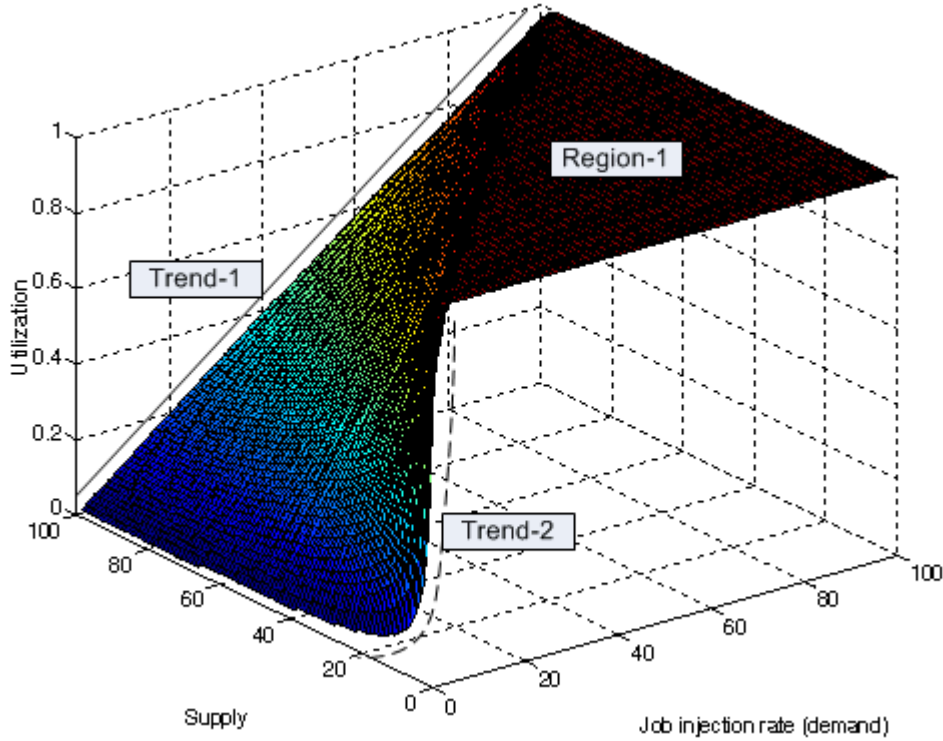


Figure 5.20: Distribution for Resource Utilization

Resource Utilization: We have observed that the success rate for CDA (Figure 5.17) is better than that of the EA (Figure 5.7). The utilization for CDA (Figure 5.20) is better compared to the utilization for EA (Figure 5.10), due to the same reason. When the demand is high and supply keeps decreasing (Region-1), the maximum utilization is obtained. Most of the nodes are easily utilized, due to the high demand. In addition, the better Pareto-optimality of CDA helps to obtain the higher utilization.

As the demand decreases (Trend-1), the rate decreases proportionally. The rate becomes dependent only on demand, due to the abundant supply. Because of the constant high supply, whatever demand there is in the market, can be accepted quickly. Therefore, the utilization tends to be the function of only demand in this case. When both the demand and supply become low (Trend-2), the rate drops drastically. The chance of getting low-cost nodes is very

low, due to the very limited supply, which hardly helps the Gridlets to be accepted, thus bringing the utilization closer to the X-Y plane.

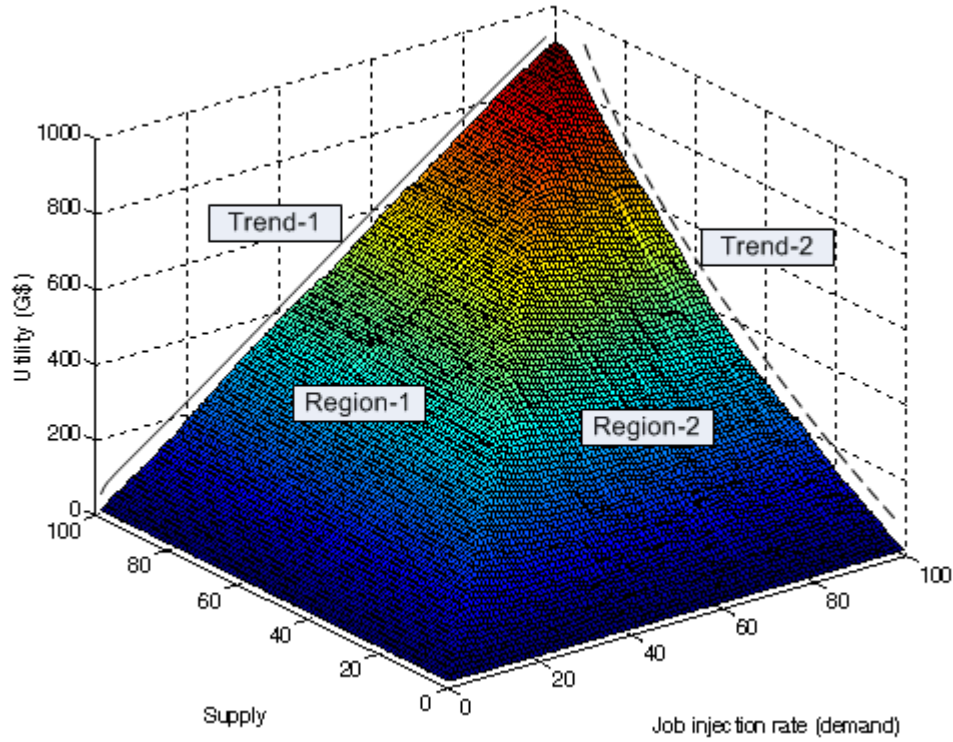


Figure 5.21: Distribution for User Utility

User Utility: Figure 5.21 depicts the distribution for user utility. One can observe that the utility for CDA is much better than that of the EA (Figure 5.11). If we look at the structure of CDA, the price formation process (*finalPrice* in Algorithm-3) is designed in such a way, it can maximizing the utility for both Gridlets and resource nodes. The utility for Region-1 is higher compared to the utility for Region-2. When the supply is high regardless of demand (Trend-1), the utility increases proportionally. The *Ask-Order-Book* is able to provide a higher number of low-cost *asks*, due to the high supply. This helps the Gridlets to execute trades with low cost prices. Therefore, the *finalPrice* tends to favor Gridlets, which helps to maximize the utility in this region. As long as the supply is high, this situation remains constant and leads to the constant performance. However, when the demand is high regardless of the supply (Trend-2), the performance drops exponentially. As the supply starts decreasing, the chance for providing low-cost nodes by the book decreases. This compels the Gridlets to execute trades with higher-prices, which minimizes the utility for Gridlets. Therefore, the overall performance across Region-2 is down.

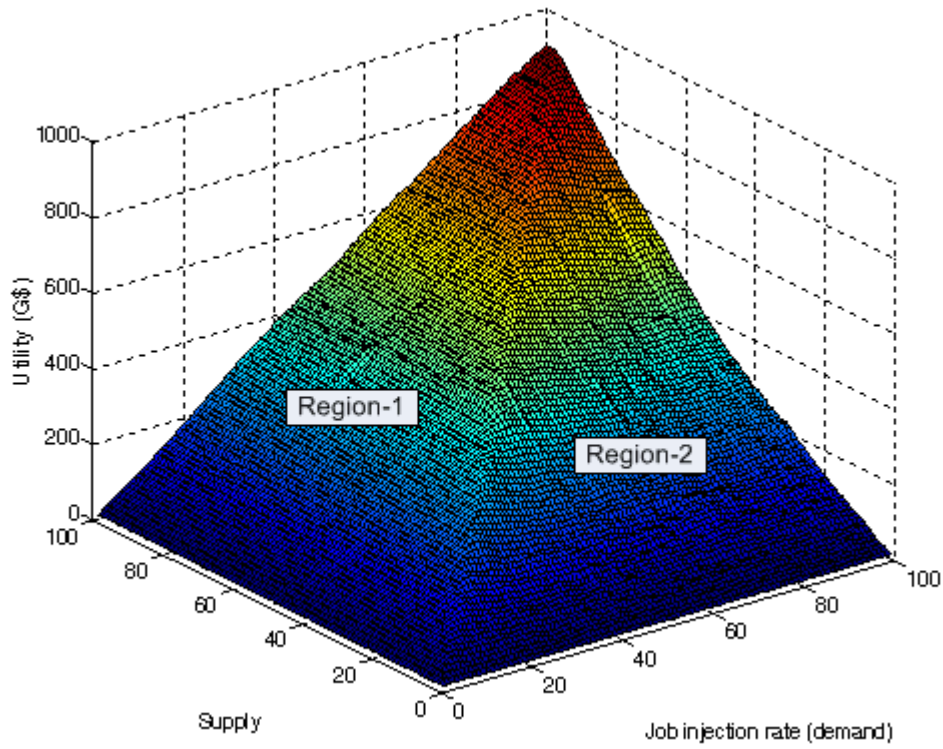


Figure 5.22: Resource Utility Distribution

Resource Utility: The resource utility (Figure 5.22) follows the similar trend to the user utility (Figure 5.21). One can argue that in case of Region-1, due to the high supply, the trades execute with low-cost prices; therefore, the resource utility should be lower for this region compared to Region-2. However, there are two factors, which can be used to explain the reason for higher utility in Region-1. Firstly, due to the abundance of supply, the aggregated utility is higher. Secondly, due to the special price formation process in CDA, nodes are also getting some opportunity to maximize their utilities. When the demand is high and supply starts decreasing (Region-2), due to the limited supply, the aggregated utility for the nodes becomes lower. Therefore, the performance is down across the region.

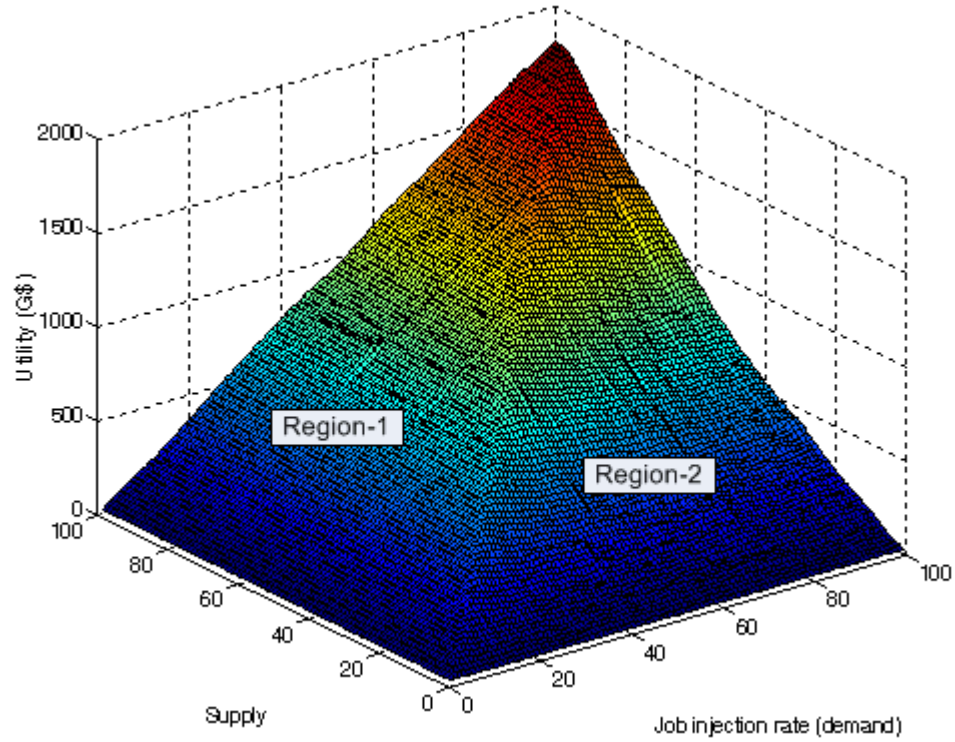


Figure 5.23: Distribution for Social Welfare

Social Welfare: As we combine the user and resource utilities for social welfare, the pattern is similar to either of those (Figure 5.23). Therefore, a similar explanation to user and resource utilities can be used to explain the welfare. Even there is a big difference between the magnitudes of user utilities for CDA and EA, the overall performance for social welfare is competitive (Figure 5.13 and Figure 5.23). In the EA, the major part of welfare is contributed by the resource utility. However, even the resource utility for CDA is lower than that of the EA, CDA is able to provide a balance value for both user and resource utilities. The highest welfare for CDA is recorded (about G\$1800) when there are the maximum number of Gridlets and nodes in the market.

The following section presents the discussion on design and development of CNP-based scheduling strategy for Grid resource management.

5.7 Contract Net Protocol Based Grid Resource Management

This section explains the scheduling strategy, the major roles played in the environment and the negotiation model for CNP. As mentioned earlier, Contract Net Protocol (CNP) is not directly an auction type; however, it supports one-to-many negotiation in a distributed Grid configuration. The effectiveness of the CNP in distributed problem solving is not recent (Smith, 1980). In Grid computing, an extensive research has been conducted to investigate the suitability of CNP-based resource negotiation (Stefano and Santoro, 2008, Ranjan et al., 2008,

Dominiak et al., 2007). The CNP is popular in Grid computing, especially because of its support for meta-scheduling and for resource cooperation to deliver a strong QoS to the users. We design and develop CNP in the current GridSim distribution.

A broker and a resource-node are known as manager and contractor respectively in this scenario (Buyya et al., 2002). In such a market scenario, a manager tries to optimize its scheduling process (meta-scheduling) by selecting one or more suitable contractors from available contractors in the market. The selection process is typically conducted according to the manager's preference values. In terms of a task execution in a distributed environment such as Grid, the manager could either prefer time or budget to be optimized. Considering the optimization function, the manager, then, advises its *cfp*. Afterward, the manager broadcasts the *cfp* to available contractors in the market seeking for potential contractors. The contractors that can satisfy the requirements of the *cfp* are regarded as potential contractors. However, in a concurrent market environment, processing multiple *cfps* concurrently could lead some inefficiency to the user QoS. For example, the optimization for a *cfp* might be significantly hampered if at the same time, some contractors are busy negotiating with other *cfps*. A manager in our scenario would eventually awards only one contractor regardless of how many potential contractors are there. This maintains consistency with the previous models. Different *cfps* are then broadcasted randomly to the available contractors one by one.

The contractors, then, evaluate the *cfp* based on their performance characteristics such as MIPS rating. The evaluation process is very much similar to the other economic models such as Commodity Market. Depending on the *cpuTime* and *cpuCost* (Eq-4.1 and Eq-4.2 in Chapter 4), a contractor determines its interest on a *cfp*. Each contractor has its respective *Observer* and *Responder* to deal with (1) the reception of a new *cfp*, (2) the evaluation of the *cfp* and finally (3) the decision (accept/reject) on the *cfp*. Each contractor must send a message even if the contractor rejects the *cfp*, which ensures the manager that it has received responses from all the contractors. This assurance prevents the manager from keep waiting for a longer time for responses from the contractors. A contractor will be interested on a *cfp* if the following conditions are satisfied,

$$\begin{aligned} &cpuTime \leq \text{The relative Gridlet's deadline} \\ &cpuCost \leq \text{The relative Gridlet's budget} \end{aligned}$$

Otherwise, it will reject the proposal. Once the manager obtains the list of potential contractors, it executes the meta-scheduling process. A detailed explanation on the meta-scheduling process is given in sub-section 5.7.1. We develop a CNP-Manager class that conducts all the decision making process on behalf of a broker. A typical *cfp-template* and the

behavior of a contract-node are presented in Figure 5.24. A user could define a deadline to receive bid from available nodes inside the template (not shown here). This prevents users from waiting longer if some nodes are idle or taking a long time to provide their feedback. However, we do not design such idle nodes in our simulation; therefore, we ignore this parameter from the *cfp*.

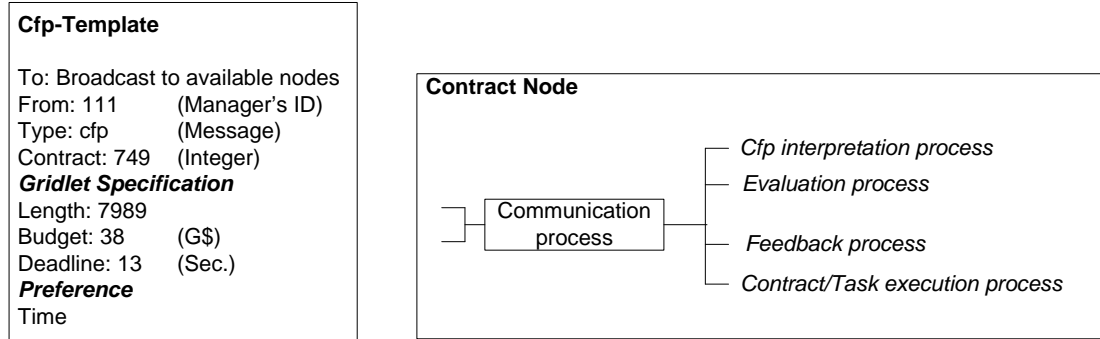


Figure 5.24: Structure of a cfp-template and Behavior of a Contract Node

5.7.1 Meta-scheduling Process

Scheduling jobs to appropriate resources is a key topic in Grid resource management (Garg et al., 2009, Sapra et al., 2010). CNP delivers necessary components in this regard to obtain QoS-based execution of Grid applications. Currently, our CNP-manager can handle to deal with either time or budget-based optimization. However a cross-value optimization e.g., 40% time optimization and 60% budget optimization is also possible, which we would like to explore in future.

The manager receives a set of potential contractors, P_n those have shown their interest to accept the *cfp*. The manager, then, makes a decision to award one from the available potential contractors by considering the preference function.

Preference: Budget-optimization. If the manager selects the node n from a set of available nodes P_n , we can write,

$$n = (n \in P_n \mid cpuCost(n) \leq cpuCost(\forall n' (n' \in (P_n - n)))) \quad Eq - 5.2$$

Where, $|P_n| \geq 1$, which means, there must be at least one potential node for the proposal. Otherwise, the user corresponding to the proposal will be terminated by assuming that there is no potential node in the market to accept the proposal. n' is a potential node which has been rejected by the manager and $(P_n - n)$ indicates the set of rejected potential nodes. Eq-5.2 helps the manager to identify the contract-node with minimum cost. We use a comparator to do the process. The comparator compares relative *cpuCosts* of the potential nodes and selects the node with minimum cost.

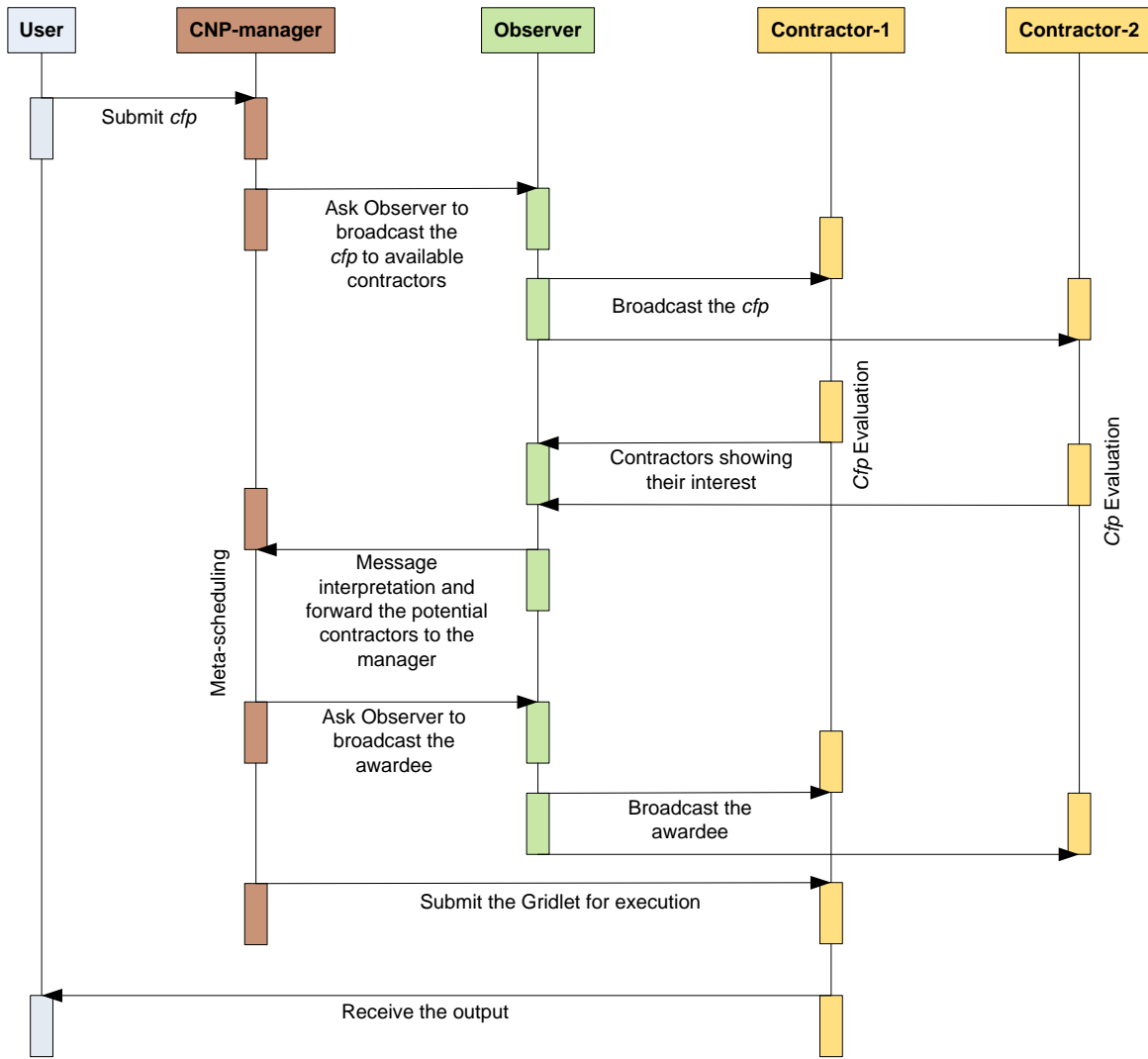


Figure 5.25: An Event Diagram for Resource Negotiation in Contract Net Protocol

Preference: Time-optimization. In terms of time optimization, the node n is represented as,

$$n = (n \in P_n \mid \text{cpuTime}(n) \leq \text{cpuTime}(\forall n' (n' \in (P_n - n)))) \quad \text{Eq} - 5.3$$

This identifies the contract-node that gives the minimum execution time. The manager awards the first node if two nodes provide same execution time. This is the manager's responsibility to notify the result of the award to every node in the set of potential nodes $(P_n - n)$, which helps the nodes to decide whether to take participate in the following proposals.

5.7.2 An Example of Resource Negotiation in Contract Net Protocol

Figure 5.25 illustrates the negotiation process between a CNP-manager and several contractors. At the beginning of the negotiation process, the user submits its *cfp* to the CNP-Manager. The manager obtains the available contractors' (resource) IDs from GIS (not shown in the diagram). The manager, then, invites bids from the available contractors through

broadcasting the *cfp* using its *Observer*. Contractor-1 and Contractor-2, then, evaluate the *cfp* and reply showing their interest. The *Observer* interprets the replies from the contractors and passes the list of potential contractors to the manager. The manager, then, decides which contractor to be awarded depending on the user's preference function. The final decision by the manager is then broadcasted to the both contractors. In this case, Contractor-1 has been awarded. As the decision about the award has been broadcasted to the both contractors, Contractor-2 can now accept an invitation from a future proposal. Therefore, the manager submits its Gridlet to Contractor-1 for execution. Finally, the user receives the Gridlet with the corresponding outputs from Contractor-1.

The following section presents and discusses the simulation results obtained from the CNP-based resource management.

5.8 Performance Analysis for Contract Net Protocol Based Grid Resource Management

In this section, we discuss the performance of CNP for distributed resource management in terms of various evaluation metrics. For explanation purpose, we use the terms Gridlet and node/resource-node to refer to the terms user/manager and contractor respectively.

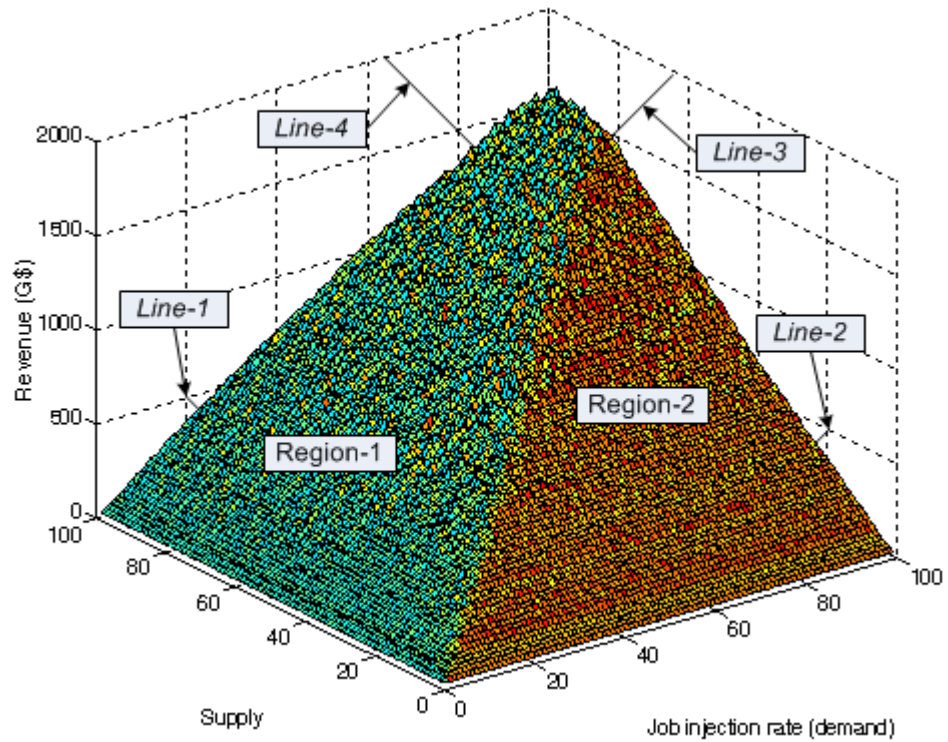


Figure 5.26: Revenue Distribution

Revenue: Figure 5.26 illustrates the revenue distribution for CNP. It can be observed that the overall revenue for CNP is far lower compared to that of the EA (Figure 5.5) and CDA

(Figure 5.15). If we look at the organization of CNP, the Gridlets in this case have the opportunity to optimize their preference functions through the meta-scheduling process (Eq-5.2 and Eq-5.3). This opportunity of optimization by the Gridlets prevents the nodes from maximizing their revenue. The fluctuation in revenue is apparent across the plot. This fluctuation is due to the random nature of assigning preference values (time/cost) to the Gridlets. For example, in terms of time optimization, the overall revenue is generally higher compared to the budget optimization by the Gridlets. If a set of Gridlets comes with time optimization, the nodes are likely to get higher revenue, because the Gridlets in this case, are not much worried about their budget. However, in terms of budget optimization, nodes are not able to maximize their revenue, as the Gridlets would prefer low-cost nodes in this case.

When the supply is high and demand starts increasing (Region-1), the most of the Gridlets are easily accepted and they receive higher chance to optimize their preferences due to the high supply. Even the opportunity of maximizing the preferences by the Gridlets in Region-1, the revenue for nodes is supposed to be down compared to the Region-2. However, as the optimization happens in terms of both time and budget randomly (section 5.7.1), the decrease in revenue in case of budget optimization can somehow be covered with the increased revenue in terms of time optimization. This prevents the model from bringing the performance sufficiently down in Region-1.

On the other hand, when the demand is high and the supply keeps decreasing (Region-2), getting the opportunity by the Gridlets to optimize their preferences becomes harder. This helps the nodes to obtain Gridlets with higher prices, which should result in obtaining higher revenue in this region. However, Region-2 does not provide more revenue compared to the Region-1. The low supply in Region-2 should cause the nodes to obtain more revenue, because the Gridlets receive limited chance to maximize their preferences in this case. However, a set of Gridlets is also rejected due to the shortage of supply; therefore, the aggregated revenue for a particular simulation is low. This prevents the model from maximizing the revenue in the region; therefore, there is no significant difference in revenue for both Region-1 and Region-2 (if we compare *line-1* and *line-2*). However, as the supply increases and demand is still high (the upper part of Region-2), more Gridlets are starting to be accepted. This higher acceptance causes to increase the revenue in this region (*line-3*). This condition continues until the supply is about 80%. As the supply further increases (the upper part of Region-1), the possibility for receiving higher optimization by the Gridlets increases. This higher possibility of optimization by the Gridlets prevents the model from maximizing the revenue in this region (*line-4*).

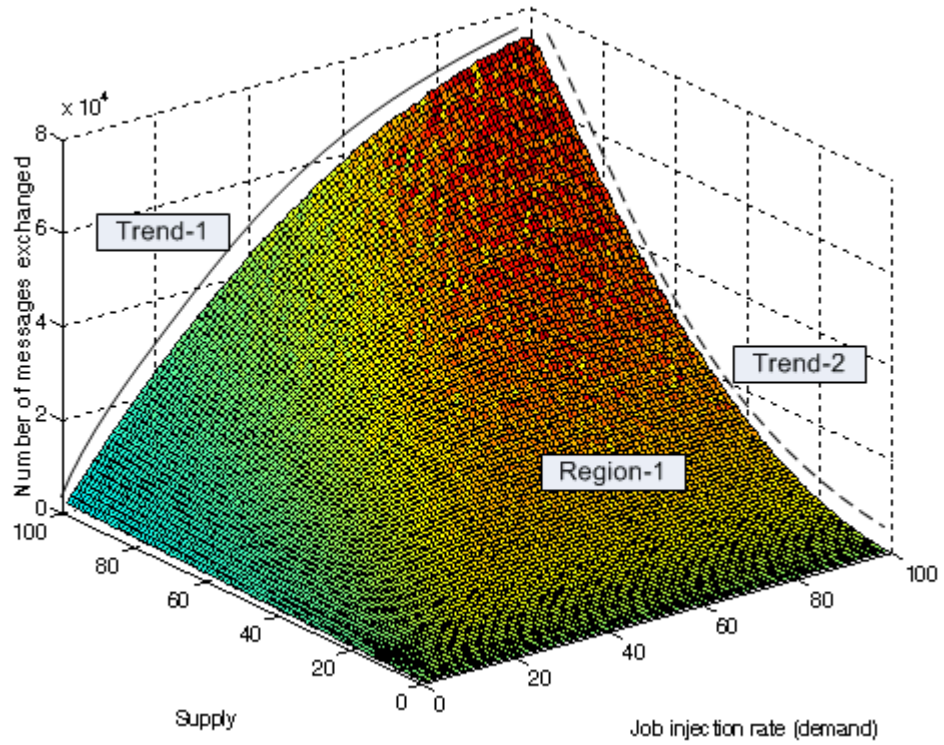


Figure 5.27: Distribution for Communication Overhead

Communication Overhead: Figure 5.27 presents the distribution for communication overhead. When the supply is high and demand starts increasing (Trend-1), the overhead increases exponentially. The size of the set of potential nodes P_n for a Gridlet increases, due to the high supply and low demand. In CNP, if a node receives a *cfp*, the node must response back to the respective Gridlet even the node is not interested with the *cfp*. Again, even a Gridlet awards only one potential node, the Gridlet is responsible to notify the decision of the award to all the potential nodes. These cases increase the overall overhead in this region (Trend-1). However, as the demand increases over time, the overhead tends to be decreased (the upper part of the Trend-1). A set of suitable nodes is quickly occupied by a set of early-arrived Gridlets, due to the higher demand. This leads the slow Gridlets to obtain a limited size of potential nodes (P_n), thus decreasing the overall number of messages exchanged in the environment.

As the supply decreases (Trend-2), the size of P_n starts decreasing for individual Gridlets. This helps to bring down the overhead drastically. Even the demand is high in the region the limited supply is quickly occupied by a set of Gridlets. Therefore, it does not require the rest of the Gridlets to send their *cfps* due to the unavailable supply. As the supply is sufficiently low

(Region-1), the overhead remains almost constant. The quick allocation of the very few nodes due to the high demand helps to obtain a constant performance across this region.

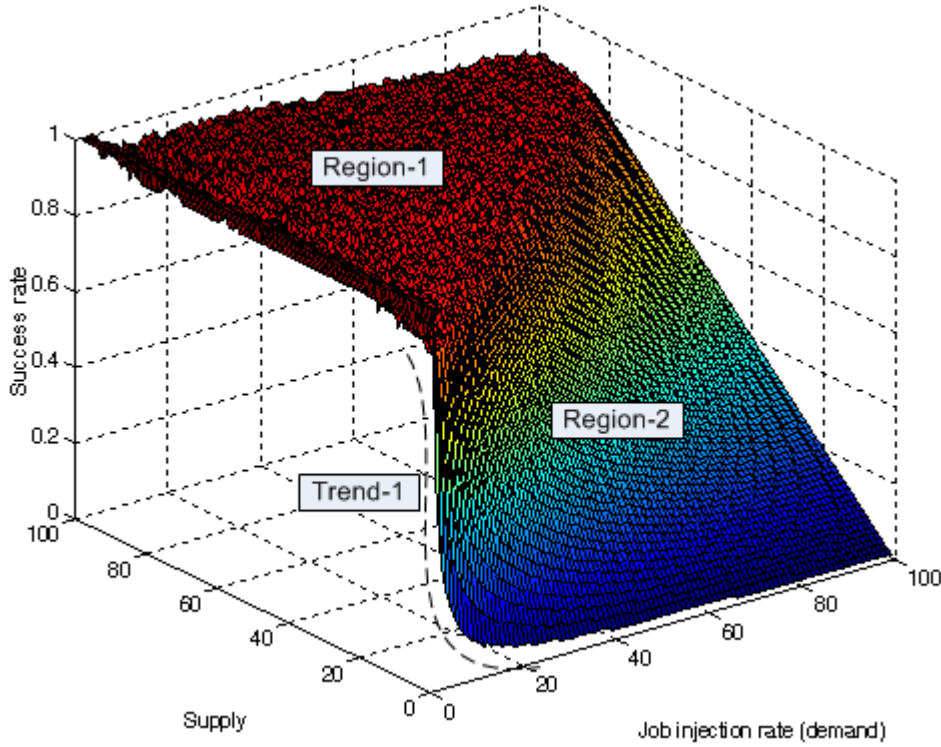


Figure 5.28: Distribution for Success Rate

Success Rate: Success rate for CNP is demonstrated in Figure 5.28. Region-1 describes the scenario when there is sufficient supply regardless of the demand. Most of the Gridlets are easily accepted, due to the high supply, which helps to obtain a higher rate across the region. However, the region is not flatten rather fluctuates very often. This is due to the low Pareto-optimality of resource allocation in CNP. CNP fully focuses on Gridlet-side without caring much about which node to be chosen for which Gridlet. In addition, Gridlets come with random preference values of either time or cost. These random behaviors by the model bring some randomness in resource distribution. Therefore, assuring constant success rate across the region becomes harder even the supply is high. As the supply starts decreasing (Region-2), the model receives limited opportunity to exhibit its low Pareto-optimality of resource allocation. Therefore, the performance moves down smoothly for this region. Trend-1 can be explained with the similar explanation given before (Figure 5.7).

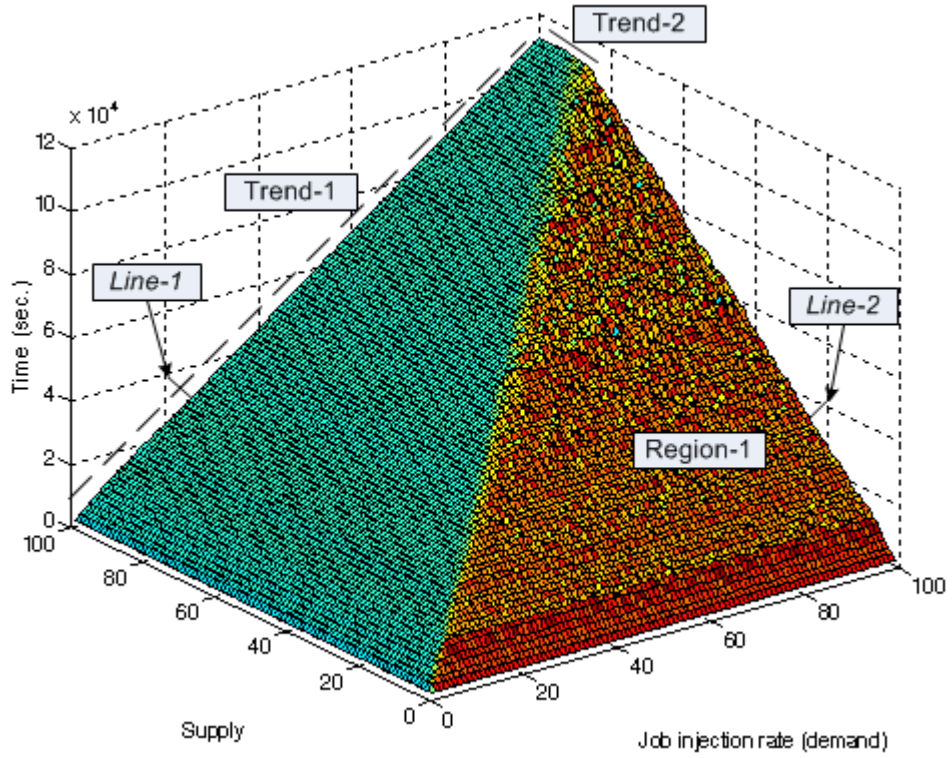


Figure 5.29: Distribution for Average Turn-around Time per Gridlet

Average Turn-around Time per Gridlet: Figure 5.29 demonstrates the distribution of average turn-around time per Gridlet for CNP. The overall time is higher for CNP compared to that of the EA and CDA (Figure 5.8 and Figure 5.18). As the Gridlets in CNP are processed one by one, the average time for the slow Gridlets are generally higher, which ultimately contributes in maximizing the average time for the fast Gridlets. This eventually extends the overall average time per Gridlet. When the supply is high and demand starts increasing (Trend-1), the time proportionally increases. The time for this region (high supply and low demand) is slightly lower (*line-1*) compared to that of Region-1 (*line-2*). The few Gridlets can quickly be processed due to the pool of nodes, due to the high supply and low demand. However, as the demand increases (The upper part of Trend-1), the additional time resulting from the slow Gridlets becomes significant. This causes to longer the time for this particular region.

As the supply slightly decreases (Trend-2) from 100% to 82% or so, the time remains almost constant. As there are high supply and high demand in this region, it is easier for a set of fast Gridlets to optimize their preferences. This leads the other Gridlets to optimize their preferences with the rest of the nodes. The scheduling process can be continued due to the high supply. As the supply further decreases (from 82% to below) (Region-1), the time dramatically drops. In this case, even the slow Gridlets cannot travel further due to unavailable

supply. The limited supply is quickly occupied by a set of fast Gridlets, thus causing the slow Gridlets directly to be terminated. This minimizes the average time per Gridlet in this region.

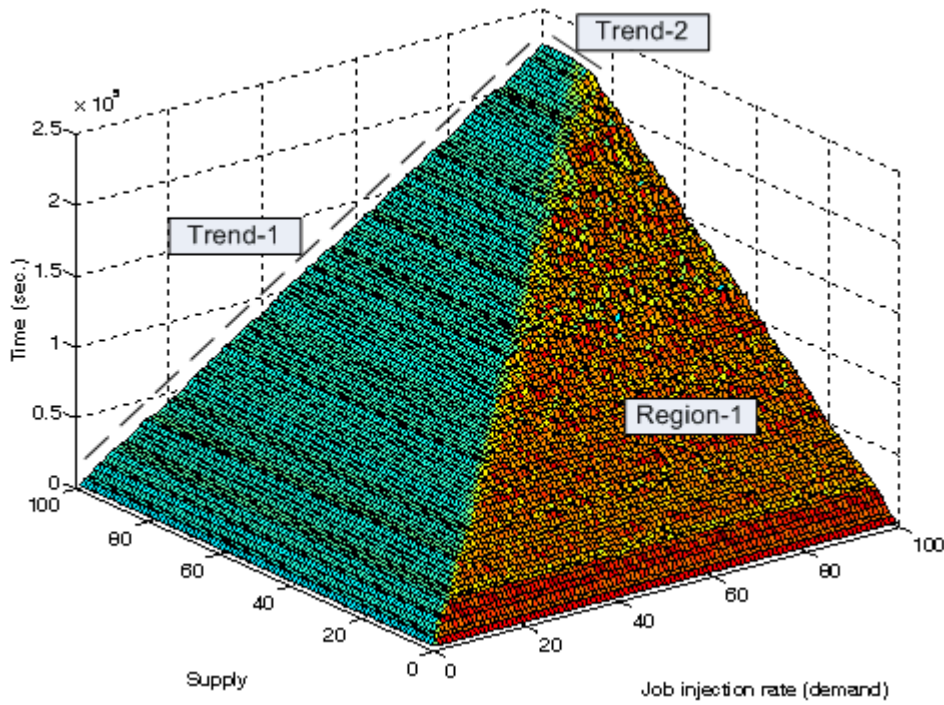


Figure 5.30: Distribution for Total Simulation Time

Total Simulation Time: Figure 5.30 presents the distribution for total simulation time. It follows a similar trend to the Figure 5.29; therefore, a similar explanation can be applied for Figure 5.30 also. The simulations take shorter time to finish compared to the Region-1, due to the high supply and low demand (Trend-1), because the few Gridlets can easily be accepted due to sufficient nodes in this case. When both the supply and demand is high (Trend-2), the time remains constant, because, due to the additional supply, the slow Gridlets still obtain chance to explore the market causing the simulation time remain higher.

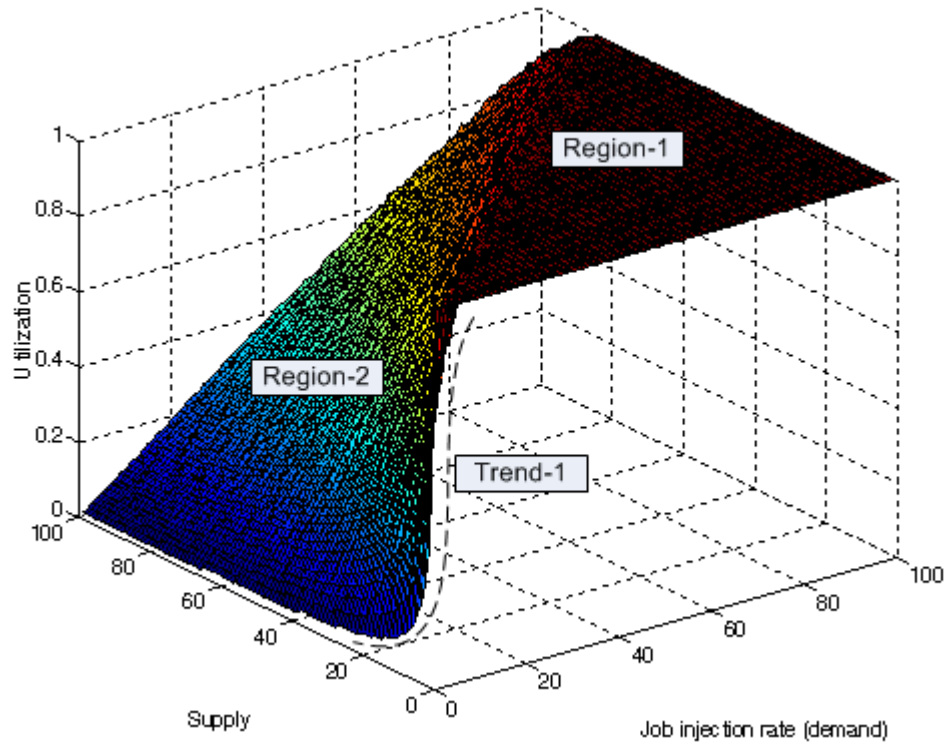


Figure 5.31: Distribution for Resource Utilization

Resource Utilization: The distribution for resource utilization is depicted in Figure 5.31. When the demand is high and supply keeps decreasing (Region-1), the nodes are easily utilized due to the abundant of demand. This causes the highest utilization to this region. However, as the demand decreases (Region-2), the rate drastically drops. As mentioned earlier, due to the low Pareto-optimality of resource allocation in CNP, the performance fluctuates in this region. Trend-1 can be explained with the similar explanation given in Figure 5.10.

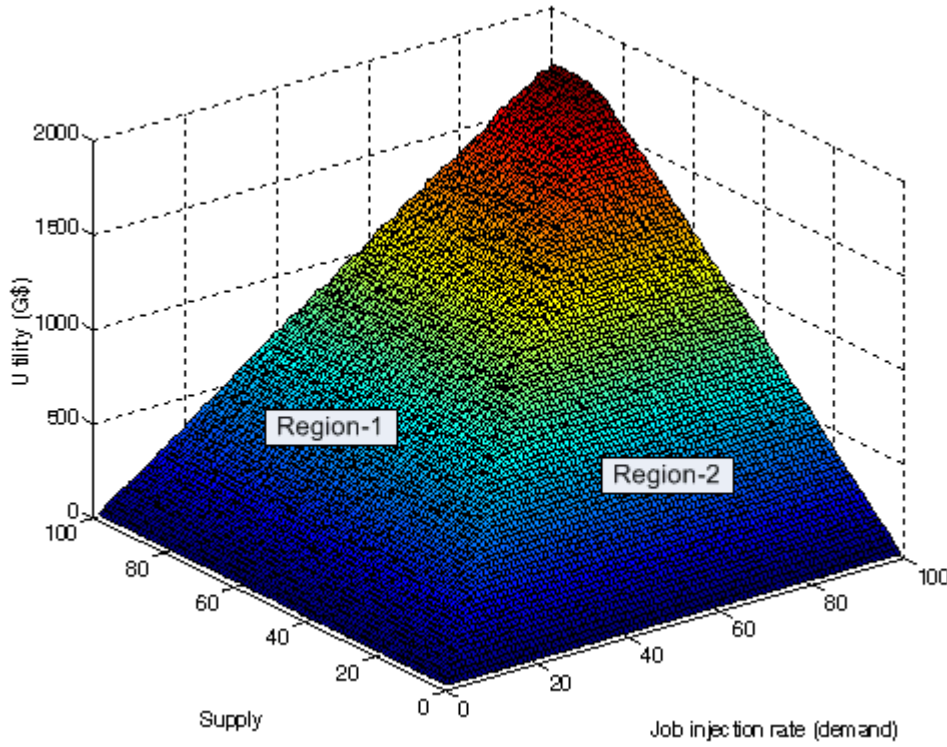


Figure 5.32: Distribution for User Utility

User Utility: It can be observed that the overall user utility (Figure 5.32) for CNP is higher compared to that of the EA and CNP (Figure 5.11 and Figure 5.21). As mentioned earlier, CNP is suitable for maximizing profit for users. Here, the Gridlets get an opportunity to optimize their preferences. Even a set of Gridlets receives the opportunity to maximize their utility through budget optimization. However, for Region-1, the magnitude is lower compared to the Region-2 because of insufficient demand. On the other hand, due to the enough demand in Region-2, the aggregated utility becomes higher.

As nodes in CNP are sold exactly with their reservation costs (like CMM in Chapter 4), there is no resource utility for CNP. The social welfare for CNP is equal to the user utility, due to the same reason.

Thus far, we have presented and analyzed the performances of three auction models in terms of distributed Grid resource management. The explanation of the performances using the parameters that describe the models demonstrates the accuracy in the implementation of the models. We have observed the suitability of different models in different scenarios. For example, even EA has generated the highest revenue CDA is more suitable for minimization of communication overhead. Therefore, to a better understanding about the strengths and weaknesses of these models, in the following section, we perform a comparative analysis among the performances of these models.

5.9 A Comparative Analysis of the Auction Protocols

It can be observed from the discussion on various auction results, a single protocol does not constantly perform better for every scenario. This variation in performance thus leads to identify the regions of strengths of the different auction protocols. Therefore, this section presents a comparative analysis among the performances of EA, CDA and CNP for Grid resource collaboration.

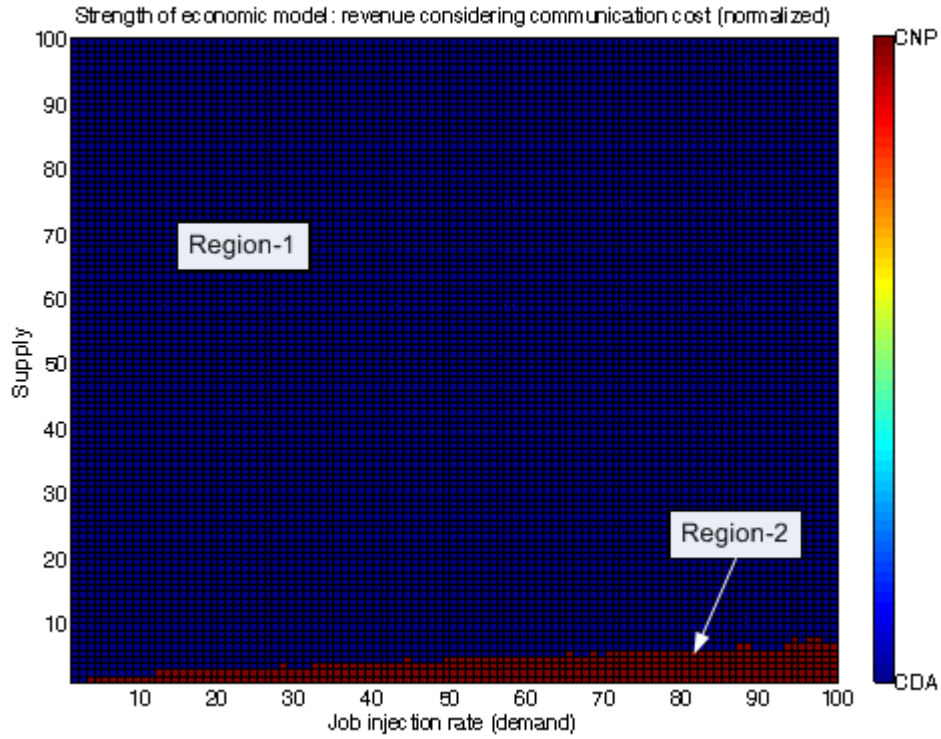


Figure 5.33: Comparison for Revenue

Revenue: To understand which model generates the highest revenue in which scenarios, we perform cell-wise comparison among the three matrices representing the revenues for three models – EA, CDA and CNP. We identify that EA always outperforms the other two models (not shown here). This identification proves the compatibility of our findings with the existing literature that EA is suitable for maximizing revenue (Buyya et al., 2005, Beck et al., 2008). The scheduling strategy of EA helps to identify the higher-budgeted Gridlets (Section 5.3). This strategy helps to generate higher revenue for the model. However, as we have seen that the EA at the same time produces a huge amount of communication overhead (Figure 5.6), we are inspired to investigate the revenue considering communication overhead objective function. Therefore, we normalize the relative matrices and perform the similar comparison once again. We follow the similar process for normalization as given in Section 4.6 in Chapter 4. The result is plotted in Figure 5.33. The contour plot (Figure 5.33) demonstrates some

interesting outcomes. The EA is completely absent here rather CDA and CNP are apparent. The major part of the contour is dominated by CDA (Region-1). It can be observed that the overall communication overhead of CDA (Figure 5.16) is very lower compared to that of the EA and CNP. The market is able to execute the trades quickly, due to the sorting process by the Auctioneer in CDA (Figure 5.14), thus preventing the model from exchanging a higher number of messages. Again, due to the higher Pareto-optimality of CDA compared to the EA and CNP and special price formation process (*finalPrice* in Algorithm-3), the protocol is able to generate somewhat moderate revenue (Figure 5.15). In most of the scenarios CDA outperforms the other two models, due to these two factors (sorting process and better Pareto-optimality).

However, when the supply is sufficiently low, CNP tends to be emerged. CNP, in this region, produces lower messages compared to that of the CDA (refer to Figure 5.16 and Figure 5.27). The few nodes are quickly occupied by the Gridlets, due to the low supply and high demand (Region-2). On the other hand, for CDA, even there are a few nodes, they start generating *asks* continuously from the beginning of a simulation. This causes the model to produce a higher number of messages in this region. Therefore, CNP receives some opportunity to show its strength in Region-2.

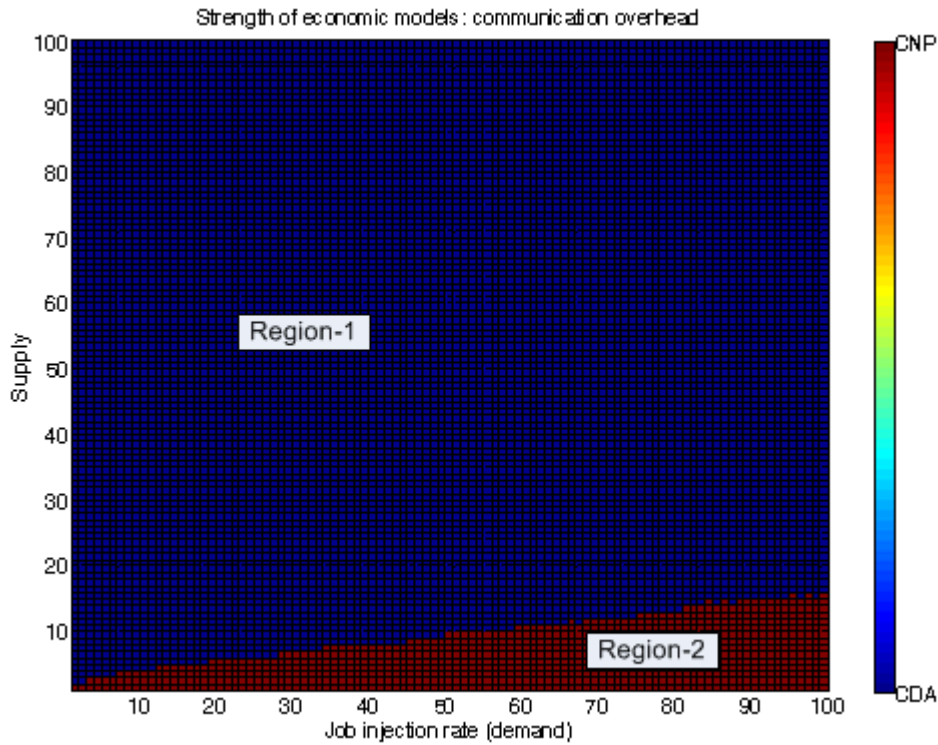


Figure 5.34: Comparison for Communication Overhead

Communication Overhead: Figure 5.34 illustrates the comparison for communication overhead. The similar explanation can be given here as given for Figure 5.33. In most of the scenarios (Region-1), CDA produces a lower number of messages compared to the other models whereas, when the supply is significantly low (Region-2), CNP performs better. The strength of CNP in Region-2 can be explained with the weakness of CDA in this region. The second paragraph of the previous contour provides the explanation of this weakness. The strength of a particular model, in this case, we mean that a lower number of messages have been exchanged compared to the others.

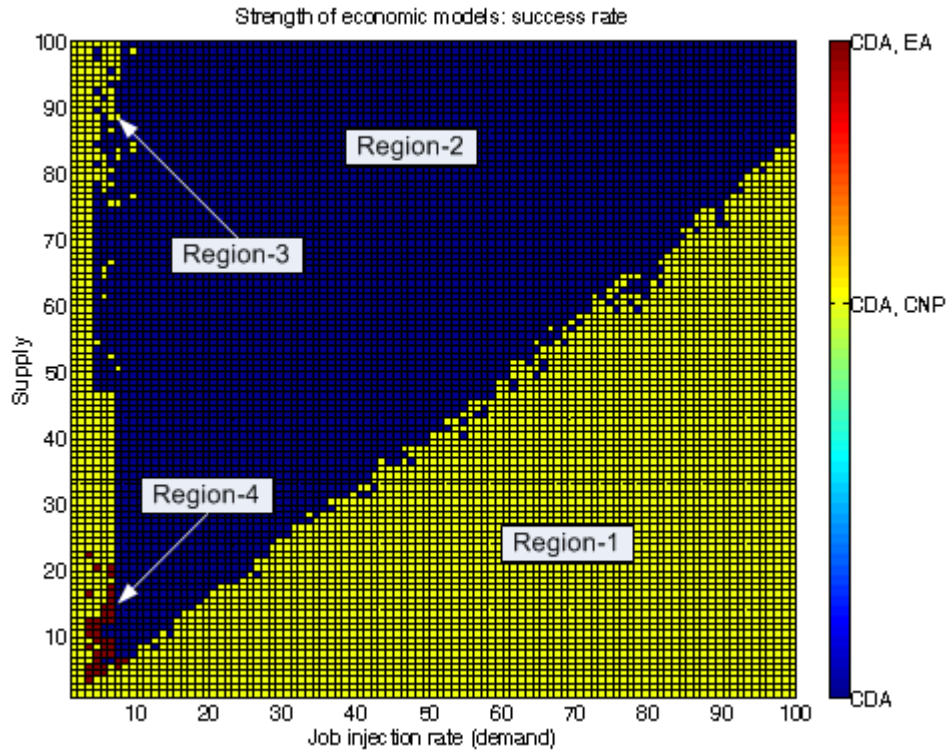


Figure 5.35: Comparison for Success Rate

Success Rate: Figure 5.35 illustrates the contour diagram for the comparison of success rate. There are four different regions, we can observe here. As we have seen that CDA is more suitable for Pareto-optimal resource allocation than that of the EA or CNP, CDA is present across all regions. Almost half of the contour is dominated by CDA itself (Region-2). However, when the supply starts decreasing irrespective of the demand (Region-1), both CDA and CNP perform equally better. As the supply decreases, CNP provides limited chance to the Gridlets to optimize their preferences. This limitation for the fast Gridlets somehow brings a better acceptance even for the slow Gridlets; therefore, CNP is able to accept more Gridlets in this case. Again, when the demand is low regardless of the supply (Region-3), CNP again competes with CDA. The acceptance becomes easier in this region, due to the high supply.

When both the supply and demand tend to be low (Region-4), EA takes part with CDA. We ignore this region, due to the limited contribution.

We have observed that CDA is suitable for quick resource allocation. This is conformant with existing literature that CDA offers immediate resource allocation (Tan and Gurd, 2007). Therefore, for the average turn-around time and total simulation time, CDA outperforms the other two auction models. In terms of resource utilization comparison, we obtain similar contour to the success rate comparison. If the success rate increases, the utilization generally increases. In addition, as we are considering one Gridlet per user and one Gridlet is executed on a single resource (Figure 3.6 in Chapter 3), the success rate and resource utilization exhibit a proportional relation to each other in our case.

In terms of user utility, CNP always outperforms EA and CDA. As discussed earlier, CNP is suitable for maximizing utility for users, because it allows the Gridlets to optimize their preferences through the meta-scheduling process. However, for resource utility, EA is always better. As we have seen, the Auctioneer in EA continuously increases the bid over rounds and finally determines the highest bidder as winner. This helps the resources to be sold with the prices much higher than their original reservation prices, thus maximizing the utility for resources.

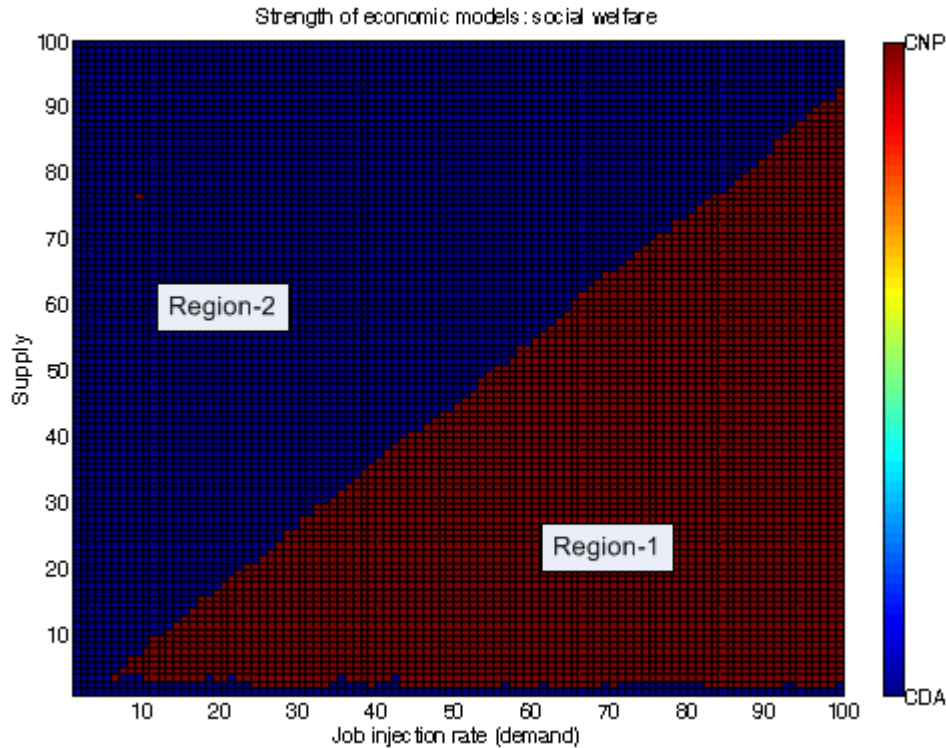


Figure 5.36: Social Welfare Comparison

Social Welfare: Figure 5.36 presents the comparison for social welfare. About half of the contour is dominated by the CDA (Region-2). Both the Gridlets and resources in CDA are

able to maximize their utilities, due to the special price formation process (*finalPrice* in Algorithm-3). This helps to maximize the combined utility (social welfare) by the model. The Gridlets are able to execute trades with low-cost nodes, due to the high supply in Region-2. This assists the Gridlets in maximizing their utilities. However, as the supply decreases (Region-1), CNP performs better. Even the supply is lower here the Gridlets can still optimize their preferences. In addition, due to the high demand, the aggregated utility for Gridlets is higher in this region. The overall welfare for CNP is higher in Region-1, due to the optimization and high demand.

5.10 Summary and Conclusion

The auction models are suitable for Grid resource management due to their decentralized nature and special price formation method. In this chapter, we presented the design and development of three most widely proposed auction models in Grid computing – English Auction, Continuous Double Auction and Contract Net Protocol. The models have been described in terms of their scheduling strategies and pricing structures. The results obtained for various performance metrics from the experimental study have then been analyzed. English Auction demonstrated its effectiveness in maximizing utility for resources, due to the competition process and winner determination methodology. However, due to the multi-rounds structure, the model produced a high communication cost. The Continuous Double Auction has been able to show its effectiveness in immediate resource allocation and social welfare. Again, the meta-scheduling process of Contract-Net-Protocol has helped to optimize the user utility as well as the social welfare.

For a more precise understanding about the potential of individual auction models, a comparative study has been conducted among the performances of the three models. The comparative study showed the compatibility with the existing literature that a single model is not suitable to cope with every scenario in Grid computing. For example, we have identified that two different models show their strengths in two different regions in terms of minimizing communication cost. When the supply is high regardless of demand, in most of the scenarios, Continuous Double Auction outperforms other two auction models. However, when the supply is low, Contract Net Protocol showed its strengths over the others. Again, we have observed that for some scenarios Continuous Double Auction and for some other scenarios, Contract Net Protocol is suitable for social welfare. Table 5.1 summarizes the findings of this chapter.

Despite the popularity of auction models, our identification demonstrated the insufficiency of a single model to deal with diverse and dynamic characteristic of the Grid. To

meet this challenge, one might be interested in our findings or in considering the deployment of multiple economic models in a Grid. This consideration would help him/her to obtain an environment where optimization of different performance metrics can be achieved through utilizing the potential of different economic models in different scenarios. In this regard, in the following chapter, we conduct a more comprehensive comparative analysis considering both commodity and auction-based economic models and attempt to generalize their domains of strengths for different performance metrics.

Table 5.1: The Strengths of Different Auction Models for Varied Supply and Demand

Performance metric	Low supply-low demand	Low supply-high demand	High supply-high demand	High supply-low demand
Revenue over Communication Overhead	CDA	CNP	CDA	CDA
Success Rate	CDA, CNP	CDA, CNP	CDA	CDA, CNP
Social Welfare	CDA, CNP	CNP	CDA, CNP	CDA

Identification of Domains of Strengths of Different Economic Models in Grid Computing

Despite the potential of economic-based resource management in Grid computing, to date, there is no consensus on which economic model to use for the Grid. The insufficiency of a single model to cope with the distributed large-scale environment, in fact, leads to this problem. Through an extensive simulative study on different economic models in the previous chapters, we identified this fact. To deal with the second and third issues as raised in Chapter 2, in this particular chapter, we attempt to identify the domains of strengths of the widely proposed economic models through a comprehensive comparative analysis among the models (including both commodity and auctions). We further formalize the domains of strength of the models for a feasible extension of our approach.

6.1 Introduction

Grid computing incorporates a good level of dynamism, which ultimately affects in execution of the applications. Therefore, before designing a framework of a Grid system, one needs to account this dynamism. The dynamism is typically described in terms of joining and leaving the users and providers randomly from the environment. A framework is said to have considered this characteristic, if it models the system feasible for evaluating every possible set of users and providers in the environment. This further helps to draw the regions where one model shows its strength over other models. In Chapter 3, we have described such a framework. Again, to make the framework more intuitive, it needs to support as many evaluation metrics as possible. We considered 10 such metrics to evaluate economic-based resource management systems. Therefore, evaluating the performance of an economic model using our system becomes more comprehensive, easily extensible, and suitable for the Grid. This chapter provides an extensive comparative analysis to understand the domains of strengths of individual economic models in a market-oriented Grid resource management.

Understanding different economic models in Grid computing is crucial, because the models provide the standards to deliver a sustainable computing platform. In Chapter 4 and

Chapter 5, we have only performed the comparison between commodity and auction models separately. In this chapter, we combine the five models to extend our comparative analysis. We identify the domains of strengths of individual economic models for individual metrics through this analysis. Along with identifying the models' domains of strengths, we further model the domains mathematically for a general understanding of the models' strengths and scalability of these strengths. We believe our investigation would provide a reasonable outcome for the Grid community to understand the insights about the models.

Through a comprehensive analysis even considering the five economic models, we identify that a single model is not suitable to cope with every scenario. For example, we identify that there are some scenarios where Commodity Market Model and for some scenarios, Continuous Double Auction is suitable for minimizing communication overhead. Therefore, we find the opportunity of utilizing the potentials of different economic models in different scenarios in order to optimize various performance metrics. In this respect, we draw some research proposals at the end of this chapter.

The rest of the chapter is organized as follows: In Section 6.2, we provide some motivation behind the identification of domains of strengths of different economic models in Grid computing. From Section 6.3 through Section 6.11, we describe the domains where a one model shows its strength over other models. We further formalize these domains in the respective sections. In Section 6.12, we draw the summary and make our proposal of utilizing the strengths of different economic models in a dynamic Grid environment. Section 6.13 presents the conclusion.

6.2 Inspiration for Domains of Strengths Specification

This section briefly introduces some motivations for identifying the domains of strengths of individual economic models in the Grid. Economic-based distributed resource collaboration is a vast and complex topic in the Grid (Haque et al., 2011, Buyya, 2002). It not only provides sufficient motivation for the resource provider to contribute their resources in the Grid but also solves distributed scheduling problem rising from millions of requests around the world. Therefore, understanding the pros and cons of different economic models is crucial for Grid service providers. We identify a few of a many reasons for which one might be interested to identify the domains of strengths of different economic models in Grid computing.

- What are the parameters that define the strengths of an economic model?
- What are the parameters that characterize the weaknesses of an economic model?
- Which model to use when and for what purpose?
- What are the preferences/optimization functions of the resource provider?

- What kind of application the user wants to execute on the Grid?
- Is there any preference values that the user wants to optimize?
- What is the value of the resources the provider is delivering to the application?
- What is the structure of the Grid network and how it is characterized?

Economic models have the ability to serve the Grid from various dimensions. It helps to understand a user's QoS requirements and resource values to the providers. Therefore, understanding the strengths and weaknesses of a particular economic model for a specific scenario is crucial. In spite of the value of economic models in the Grid, existing literature is lacking a clear demonstration of the models' domains of strengths. In this regard, in the subsequent sections, we present a comparative analysis of the five most widely proposed economic models to investigate their strengths and weaknesses for a wide range of performance metrics in the Grid.

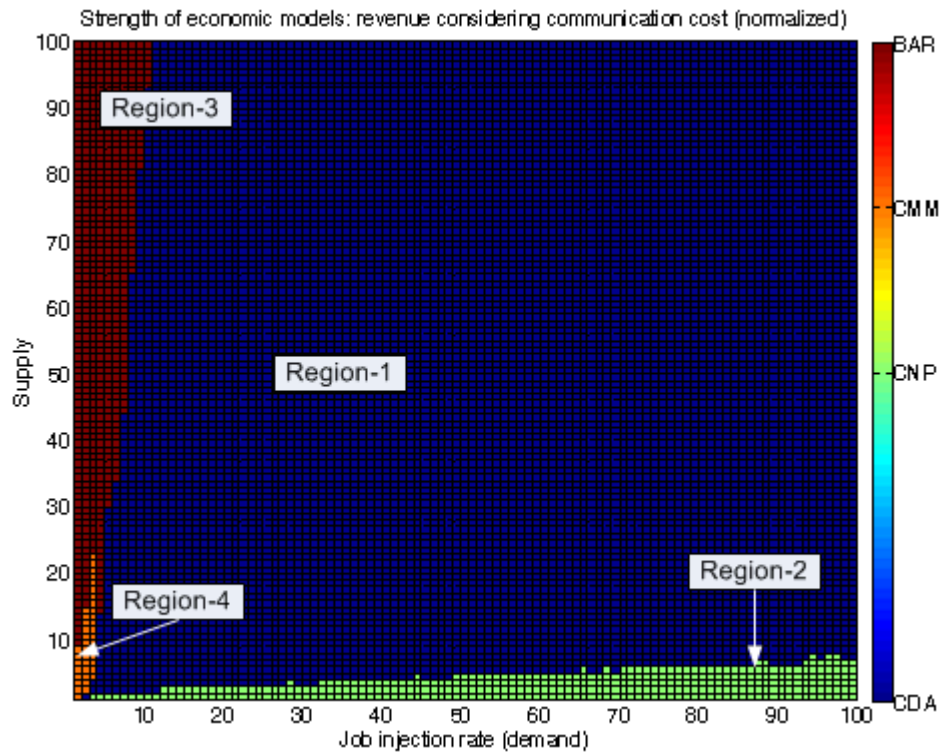


Figure 6.1: Revenue Comparison

6.3 Revenue Analysis

In order to identify which model (out of the five models) generates more revenue for resources, we perform a cell-wise comparison of the five matrices representing the revenues of the five models. We find that EA always outperforms other four models in this case. It becomes un-beatable even by the commodity models, due to the competition process in EA. As we have seen in Chapter 5, EA puts every resource in a competition where the potential

Gridlets compete for the resource by increasing their bids over rounds. It helps the protocol to obtain higher revenue at every scenario. While EA generates the highest revenue, at the same time, it produces huge communication overhead/cost. Thus, we are encouraged to investigate about revenue over communication overhead function. We normalize the both parameters and manipulate revenue over communication overhead matrices corresponding to all the five models. The manipulation is conducted according to the same process considered for revenue in Section 4.6 in Chapter 4. A cell-wise comparison is then performed among the normalized matrices. The contour diagram (Figure 6.1) shows this result. We can observe from the contour that the EA is fully absent here rather CDA, BAR, CNP and CMM are the models show their strengths at different scenarios.

Out of the four models, CDA shows the best performance. The greater part of the simulation space (Region-1) is dominated by CDA. There are two different reasons for showing such a performance by CDA. Firstly, CDA generates the lowest communication overhead in most of the scenarios (Figure 6.3). This prevents the model from lowering the revenue much. Secondly, due to the special price formation process (*finalPrice* in Algorithm-3), the revenue earned from a particular trade lies somewhere in between the Gridlet's budget and the resource's *cpuCost*; therefore, the ultimate revenue is neither very low nor very high. However, when the demand is sufficiently low irrespective of the supply (Region-3), BAR outperforms all the others. Gridlets get high chance to start bargaining with the appropriate resources, due to the high supply. Thus, the Gridlets make quick acceptance on the resources, which leads to a lower communication overhead for BAR. In addition, due to the low demand and high supply, Gridlets do not need to switch between the resources very often if a resource is busy (refer to Algorithm-1 and Algorithm-2 in Chapter 4). Even in Region-3, CMM shows lower overhead compared to that of the BAR (Figure 6.3), the ratio of revenue over communication overhead is higher for BAR than that of the CMM. There are two reasons for the CMM to produce lower revenue in this region. Firstly, the spot prices generated by CMM in this region are lower due to the high supply and low demand. Thus, the job execution costs are also lower. Secondly, in CMM, resources are only provisioned with the original job execution costs. These, prevents CMM from maximizing its revenue in this region. Even BAR, in this case, shows marginal performance compared to the CDA, analyzing the trend (which is upward) of BAR, one can understand that the performance for BAR would become significant for an immensely high supply.

When both supply and demand are low (Region-4), CMM tends to outperform other models. However, due to the little contribution, we ignore this region and assume that this is a

part of Region-3. As the supply is sufficiently low regardless of the demand (Region-2), CNP performs better than others do. CNP, in this region, produces lower communication overhead compared to that of the CDA (refer to Figure 6.3). The reason for such a characteristic of CNP is explained in Section 6.4. Moreover, as CNP supports both time and cost optimization for the Gridlets, the increased revenue could somehow cover the decreased revenue during budget optimization during time optimization scenarios. Therefore, like CDA, CNP produces a kind of intermediately revenue overall.

Domains of Strength: To define the domains of the strength of individual models, we manually choose the closest trends. To formalize the strengths, we use the terms “ s ” and “ d ” instead of Y and X to refer to the supply and demand respectively. Each domain is formalized anti-clockwise, that is, from supply to demand direction. Figure 6.2 illustrates the formalization process for revenue over communication overhead.

Domain for Bargaining Model: The first boundary that describes the domain for BAR is,

$$d = 1 \quad (\text{irrespective of supply})$$

The nature of the second boundary is straight and is a function of both supply and demand. The second boundary passes through (3.54, 1) and (11.15, 100). Now, the formula of a straight line passing through two coordinates is given by,

$$s = s_1 + \left(\frac{s_2 - s_1}{d_2 - d_1} \right) \times (d - d_1) \quad \text{Eq - 6.1}$$

Where, $s_1 = 1$; $s_2 = 100$; $d_1 = 3.54$; $d_2 = 11.15$

By substituting the values of s_1 , s_2 , d_1 , d_2 in Eq-6.1, we obtain the second boundary,

$$s = 13.01d - 46.06 \quad \text{Eq - 6.2}$$

The slope m of Eq-6.2 is given as,

$$m = \frac{s + 46.06}{d} = 13.01$$

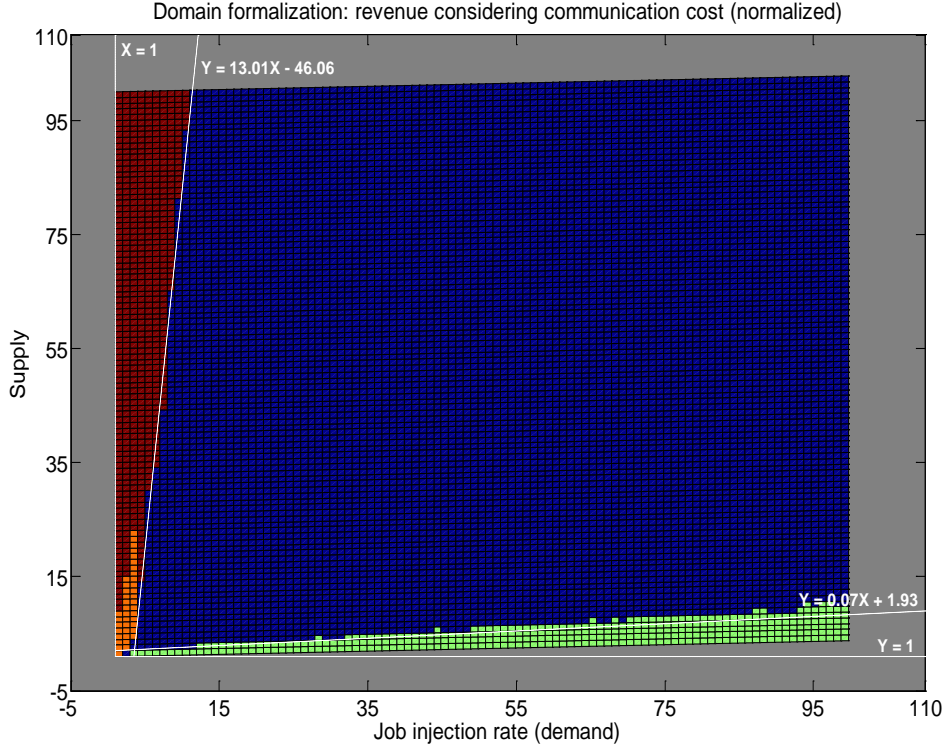


Figure 6.2: Domain Formalization for Revenue over Communication Cost

Therefore, the domain is characterized as, $d \geq 1$ and $m \geq 13.01$

Domain for Continuous Double Auction: The first boundary of this domain is the second boundary of BAR.

$$s = 13.01d - 46.06$$

For the second boundary, $s_1 = 2$; $s_2 = 9$; $d_1 = 1$; $d_2 = 100$. After substituting these values in Eq-6.1, we obtained the boundary as,

$$s = 0.07d + 1.93$$

The slopes for the first and second boundaries are,

$$m_1 = \frac{s + 46.06}{d} = 13.01 \text{ and } m_2 = \frac{s - 1.93}{d} = 0.07$$

Therefore, the domain is defined as, $m_1 < 13.01$ and $m_2 \geq 0.07$

Domain for Contract Net Protocol: The first and second boundaries of this domain are,

$$\begin{aligned} s &= 0.07d + 1.93 \\ s &= 1 \quad (\text{irrespective of demand}) \end{aligned}$$

Therefore, the domain is, $m < 0.07$ and $s \geq 1$

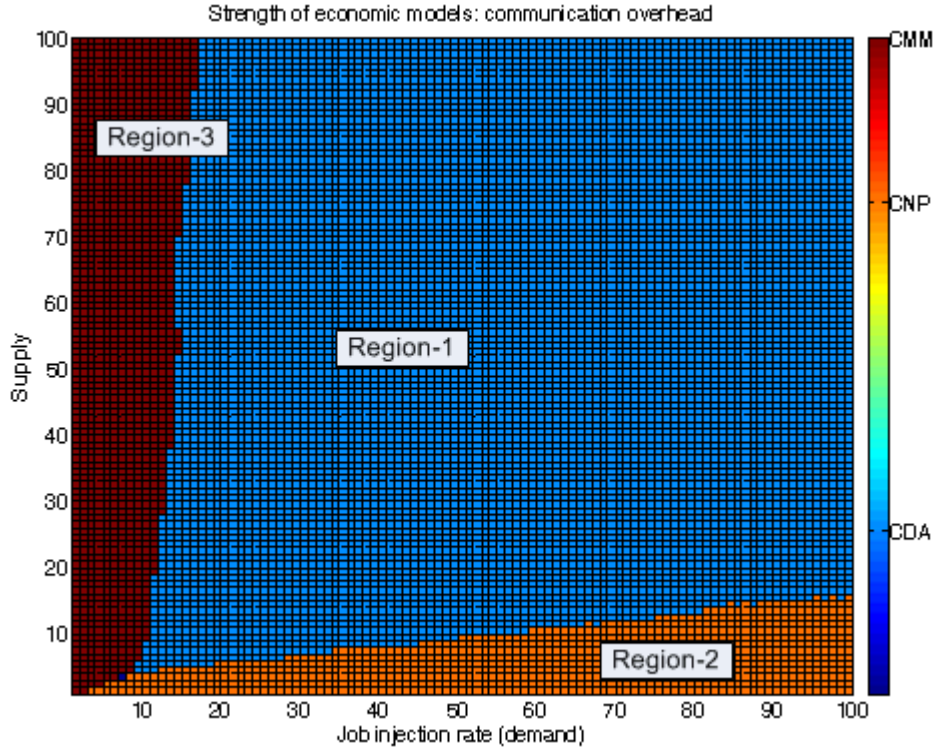


Figure 6.3: Communication Efficiency Comparison

6.4 Communication Efficiency Analysis

Figure 6.3 demonstrates the comparison for communication efficiency. There are three different regions dominated by three different economic models – CMM, CDA and CNP. To the strength of a particular model, in this case, we mean that a lower number of messages has been exchanged compared to other models. Because of having multiple rounds in BAR and EA, even to provision a single resource, they produce a higher number of messages compared to other models. Thus, these two models are absent from the whole simulation space.

In most of the scenarios (Region-1), CDA outperforms the other models. In CDA, even though resource proposals (*asks*) are sent in the environment continuously, the Auctioneer, in this case, sorts both *bids* and *asks* in an approach, it becomes much easier and quicker for the Auctioneer to clear the market. Therefore, at the end, a Gridlet or a node does not require to send too many requests/messages in the environment. However, when the demand is low irrespective of the supply (Region-3), CMM performs better. The spot prices generated by the model are low, due to the low demand. This helps the Gridlets to be accepted quickly without let them keep sending (look-up process) messages for longer. This prevents the model from generating a high number of messages. As the supply increases and demand remains almost

constant (the upper part of Region-3), the spot prices are lower than before. This leads to a quicker acceptance and helps the model to perform slightly better in this portion.

When the supply is sufficiently low regardless of the demand (Region-2), the CNP outperforms the other models. In CNP, the few nodes are occupied very quickly, due to the low supply. The strength of the CNP, in this case, can be well explained using the weaknesses of other models. As mentioned earlier, due to the multiple rounds, in BAR and EA, they always exchange a higher number of messages. For CMM, the spot prices are high for this region, which leads to a set of Gridlets to stay longer in the market. Finally, even the CDA is suitable for immediate allocation, the nodes start generating their *asks* continuously from the beginning of a simulation. Therefore, even during the lower supply, it produces a higher number of messages than that of the CNP. Thus CNP, in this region, receives some opportunity to show its strength over CDA.

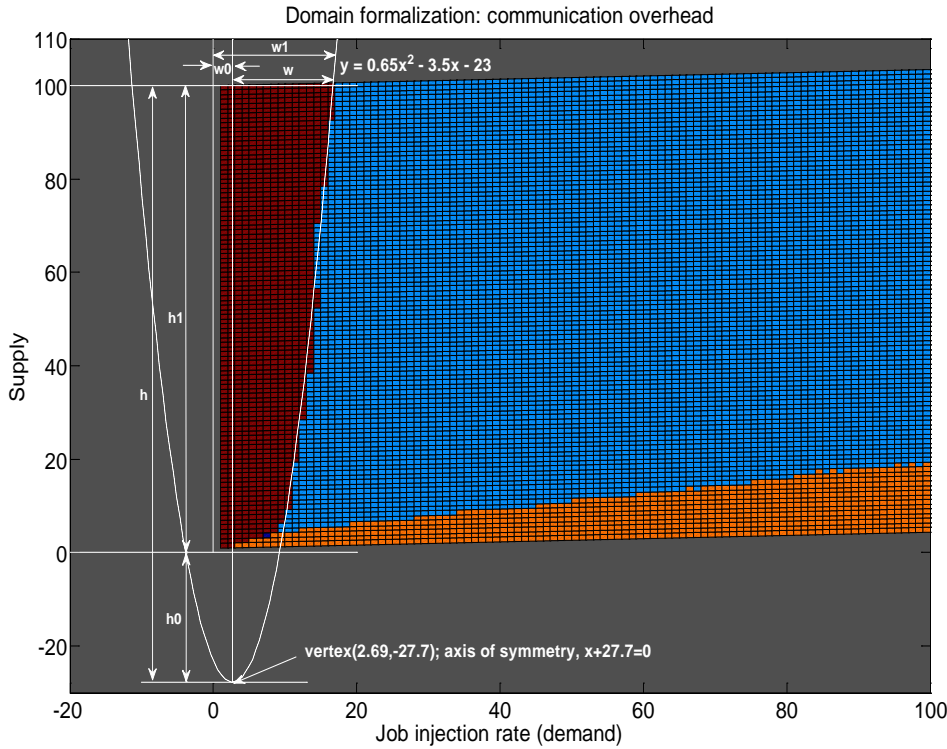


Figure 6.4: Domain Formalization for Communication Efficiency-1

Domains of Strength: Figure 6.4 presents the domain formalization for CMM.

Domain for Commodity Market Model: The first boundary of this domain is straight, which is,

$$d = 1 \quad (\text{irrespective of supply})$$

The nature of the second boundary is not straight rather quadratic (ax^2+bx+c). The equation of the closest trend is drawn as,

$$s = 0.65d^2 - 3.5d - 23 \quad \text{Eq - 6.3}$$

The slope of such a parabola is typically given by,

$$m = \frac{s}{d^2} \quad (\text{if the vertex is at } (0,0))$$

However, as the vertex of our parabola is not at $(0, 0)$, we will get a modified slope. From Figure 6.4, we get,

$$m = \frac{h}{w^2} \text{ Or,}$$

$$m = \frac{h_1 + h_0}{(w_1 - w_0)^2} \quad \text{Eq - 6.4}$$

Here, we are only interested in the absolute values of h_0 and w_0 . Now from Eq-6.3 and Eq-6.4, we can write,

$$m = \frac{h_1 + |h_0|}{(w_1 - |w_0|)^2} = 0.65 \quad \text{Eq - 6.5}$$

To obtain m of Eq-6.5, we need to know the vertex (w_0, h_0) of Eq-6.3. The x-coordinate of the vertex is given by,

$$-b/2a = 2.69 \quad \text{where, } b = -3.5 \text{ and } a = 0.65 \text{ (Comparing Eq-6.3 with } (ax^2+bx+c))$$

After solving Eq-6.3 with $d = 2.69$, we get the y-coordinate of the vertex, which is -27.7.

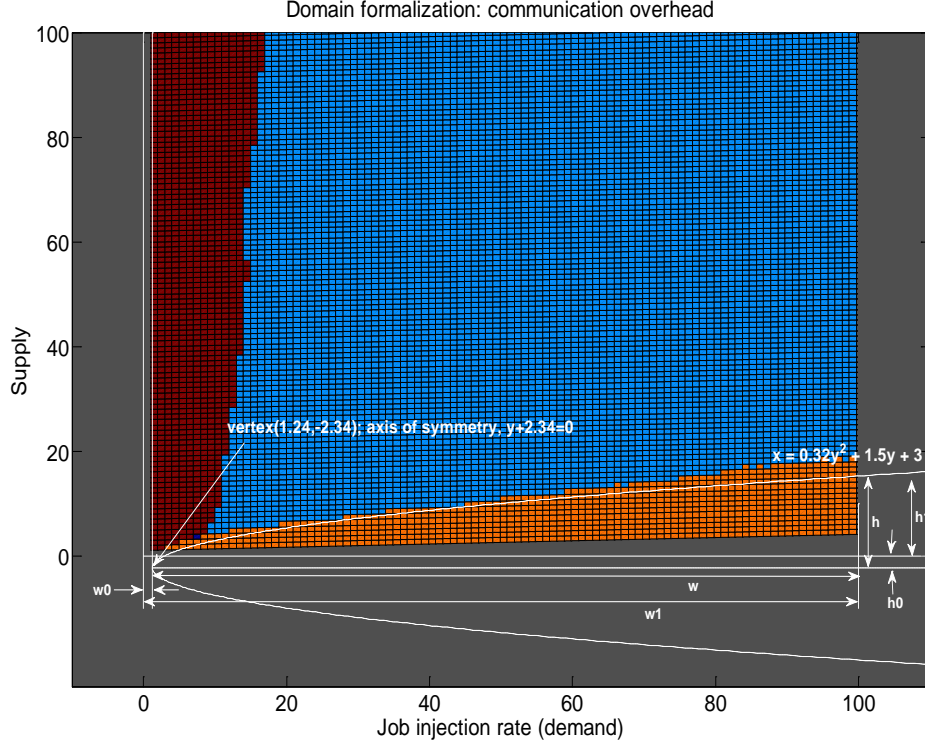


Figure 6.5: Domain Formalization for Communication Efficiency-2

Therefore,

$$w_0 = 2.69 \text{ and } h_0 = -27.7$$

Again, $|w_0| = 2.69$ and $|h_0| = 27.7$

Now Eq-6.5 gives,

$$m = \frac{h_1 + 27.7}{(w_1 - 2.69)^2} = 0.65$$

Replacing h_1 and w_1 with s and d , we get,

$$m = \frac{s + 27.7}{(d - 2.69)^2} = 0.65 \quad \text{Eq - 6.6}$$

Therefore, the domain is defined as, $d \geq 1$ and $m \geq 0.65$

Domain for Contract Net Protocol: Figure 6.5 depicts the domain formalization for CNP. The nature of the first boundary of this domain is quadratic. The equation is written as,

$$d = 0.32s^2 + 1.5s + 3 \quad \text{Eq - 6.7}$$

The slope of this horizontal parabola is given by,

$$m = \frac{d^2}{s}$$

Once again, the vertex is not at $(0, 0)$ From Figure 6.5, we get,

$$m = \frac{w^2}{h} \text{ Or,}$$

$$m = \frac{(w_1 - |w_0|)^2}{h_1 + |h_0|} \quad \text{Eq - 6.8}$$

From Eq-6.7 and Eq-6.8, one can write,

$$m = \frac{(w_1 - |w_0|)^2}{h_1 + |h_0|} = 0.32 \quad \text{Eq - 6.9}$$

Like before, Using Eq-6.7, we obtain the vertex (w_0, h_0) as,

$$w_0 = 1.24 \text{ and } h_0 = -2.34 \quad \text{where } h_0 = -b/2a$$

$$\text{Again, } |w_0| = 1.24 \text{ and } |h_0| = 2.34$$

From Eq-6.9,

$$m = \frac{(w_1 - 1.24)^2}{h_1 + 2.34} = 0.32$$

Now replacing w_1 and h_1 with d and s , we find,

$$m = \frac{(d - 1.24)^2}{s + 2.34} = 0.32 \quad \text{Eq - 6.10}$$

The second boundary is given by,

$$s = 1 \quad (\text{irrespective of demand})$$

Therefore, the domain is, $m < 0.32$ and $s \geq 1$

Domain for Continuous Double Auction: As the domain for the CDA is enclosed by the second boundary of CMM and first boundary of CNP, the first slope is given as,

From Eq-6.6,

$$m_1 = \frac{s + 27.7}{(d - 2.69)^2} = 0.65$$

From Eq-6.10, we get the slope of the second boundary,

$$m_2 = \frac{(d - 1.24)^2}{s + 2.34} = 0.32$$

Therefore, the domain is described as, $m_1 < 0.65$ and $m_2 \geq 0.32$

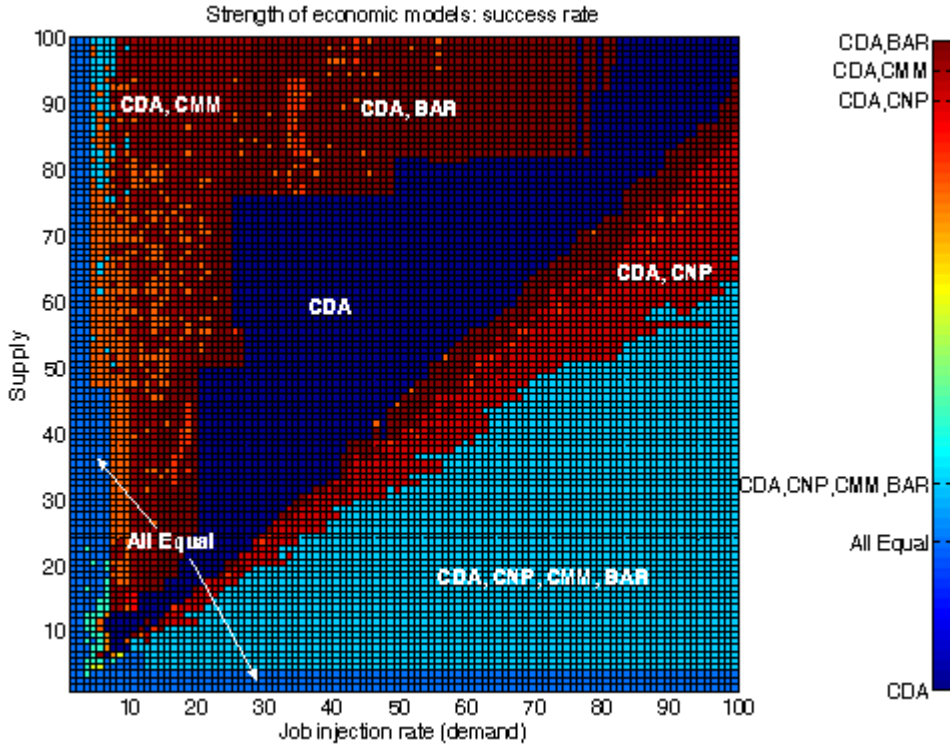


Figure 6.6: Success Rate Comparison

6.5 Success Rate Analysis

Figure 6.6 illustrates different regions representing the strengths of different models for success rate. We also consider the regions where more than one models perform equally better than other models. There are two different regions (All Equal) where all the five models perform equally better. The regions are – (1) when the demand is very low regardless of supply and (2) when the supply is very low regardless of the demand. For the first case, the few Gridlets are easily accepted without any hard competition by any of the models. For the second case, due to the low supply, there is an equal magnitude in competition for a particular resource for all models. This helps to provide an equal acceptance ratio for all the models for these two different scenarios.

As the supply starts moving up, all models except EA tend to perform equally better. To obtain a higher success ratio, Pareto-optimal resource allocation is crucial. CDA's sorting

process of *bids* and *asks* helps to give this optimal allocation. This ultimately helps to make higher success rate. Thus, CDA is present across the whole space. On the other hand, EA only focuses on higher-budgeted Gridlets rather than focusing on Pareto-optimal allocation. Therefore, the strength of EA, for this case is limited. Again, CNP, CMM and BAR are the models equally perform better with CDA; because of somewhat higher competition due to the additional demand compared to supply in this region. However, as the supply decreases (from 66%), the performance decreases proportionally. When the supply further increases (from 66% to above), the randomness in resource allocation becomes significant for CMM and BAR. Thus, the acceptance ratio decreases for these models, which helps the CDA and CNP to be persisted in the region.

When the supply further increases (from 83% to above), in CNP, the former Gridlets get a chance to optimize their preferences quickly, which leaves a kind of resources for which getting acceptance by the rest of the Gridlets becomes harder; therefore, CNP is missing here. However, BAR equally performs better with CDA in the region. The effect of lower Pareto-optimal resource allocation in BAR is slightly covered, due to the slightly higher supply, because some more Gridlets can be accepted due to the higher supply in this case. As the supply further increases (from 92% to above), the CDA alone outperforms all the other models. The Ask-Order-Book in CDA provides a high number of low-cost nodes, due to the high supply, which helps for more Gridlets to be accepted. As the demand starts decreasing (from 76% to down), multiple models again starts showing equal performance. When the demand is sufficiently low and supply is high, due to the low spot prices, a high number of Gridlets are accepted by CMM, which helps CMM to perform equally better with CDA in the region.

Domains of Strength: Figure 6.7 presents the domain formalization for success rate. We formalize the both domains separately where all the models show equal performance.

Domain-1 for All Equal: The first boundary in this case, is given as,

$$d = 1 \quad (\text{irrespective of supply})$$

The second boundary is,

$$d = 8 \quad (\text{irrespective of supply})$$

Therefore, the domain is, $1 \leq d \leq 8$

Domain-2 for All Equal: The boundaries for this domain is defined as,

$$\begin{aligned} s &= 1 & (\text{irrespective of demand}) \\ s &= 4 & (\text{irrespective of demand}) \end{aligned}$$

Therefore, the domain is, $1 \leq s \leq 4$

Domain for Continuous Double Auction and Commodity Market Model: The first boundary of this domain is given by,

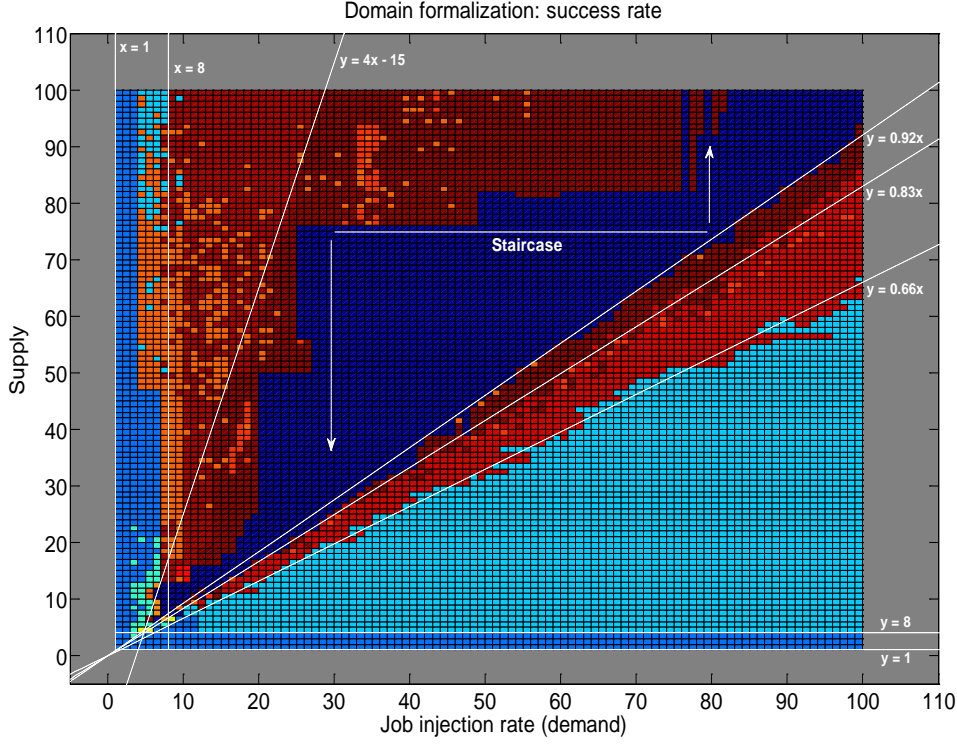


Figure 6.7: Domain Formalization for Success Rate

$$d = 8 \quad (\text{irrespective of supply})$$

The nature of the second boundary is straight, which passes through $(4, 1)$ and $(28.75, 100)$.

Therefore, Eq-6.1 gives,

$$s = 4d - 15 \quad \text{Eq - 6.11}$$

The slope of this equation is given as,

$$m = \frac{s + 15}{d} = 4$$

So, the domain is defined as, $d > 8$ and $m \geq 4$

Domain for Continuous Double Auction and Bargaining Model: For simplicity, we combine the both regions of CDA and BAR with the CDA. We have just identified the first boundary of this domain. From Eq-6.11,

$$s = 4d - 15$$

Where, slope $m_1 = (s + 15) / d = 4$

As the second boundary is passing through the origin, we can write,

$$s = 0.83d$$

Where, the slope $m_2 = s/d = 0.83$

Hence, the domain is defined as, $m_1 < 4$ and $m_2 \geq 0.83$

Domain for Continuous Double Auction and Contract Net Protocol: The first boundary of this domain is already defined, which is,

$$s = 0.83d$$

Where, the slope $m_1 = s/d = 0.83$

In addition, the second boundary is,

$$s = 0.66d$$

Where, the slope $m_2 = s/d = 0.66$

Thus, the domain is, $m_1 < 0.83$ and $m_2 \geq 0.66$

Domain for Continuous Double Auction, Contract Net Protocol, Commodity Market Model and Bargaining Model: The domain is enclosed by the following boundaries,

$$\begin{aligned} s &= 0.66d \\ s &= 4 \quad (\text{irrespective of demand}) \end{aligned}$$

Therefore, the domain is characterized as, $m < 0.66$ and $s \geq 4$

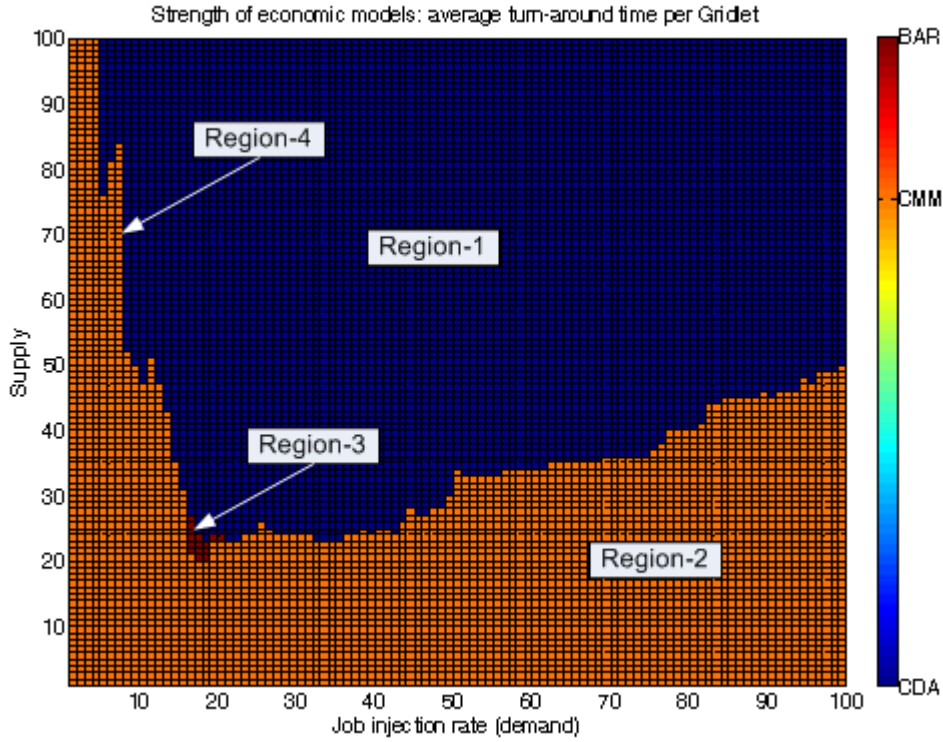


Figure 6.8: Comparison for Average Turn-around Time per Gridlet

6.6 Analysis for Average Turn-around Time per Gridlet

Figure 6.8 demonstrates the contour diagram representing the comparison for average turn-around time per Gridlet. As mentioned earlier, this metric refers to the average time required by a Gridlet to receive its final acceptance or rejection notification. The CDA, CMM and BAR are the three models those show their strength over different scenarios in this case. We ignore this, due to the little contribution of BAR (Region-3). In terms of the low demand regardless of the supply (Region-4), CMM performs better. Spot prices determined by the CMM are low, due to the low demand over resources. This helps the Gridlets to occupy the resources quickly. However, as the demand increases and the supply is still high (Region-1), spot prices start rising up. This forces a set of Gridlets to stay longer in the market to look for suitable resources. This works as a favor for CDA to perform better in this region. Again, when the supply starts decreasing and demand starts increasing, spot prices rises up. This helps the higher budgeted Gridlets to be accepted quickly, which causes the other Gridlets directly to fail due to the shortage of resources. This results in turning down the average time for a Gridlet in CMM. As mentioned earlier, CDA is suitable for immediate resource allocation, which is another reason for the model to perform better in Region-1.

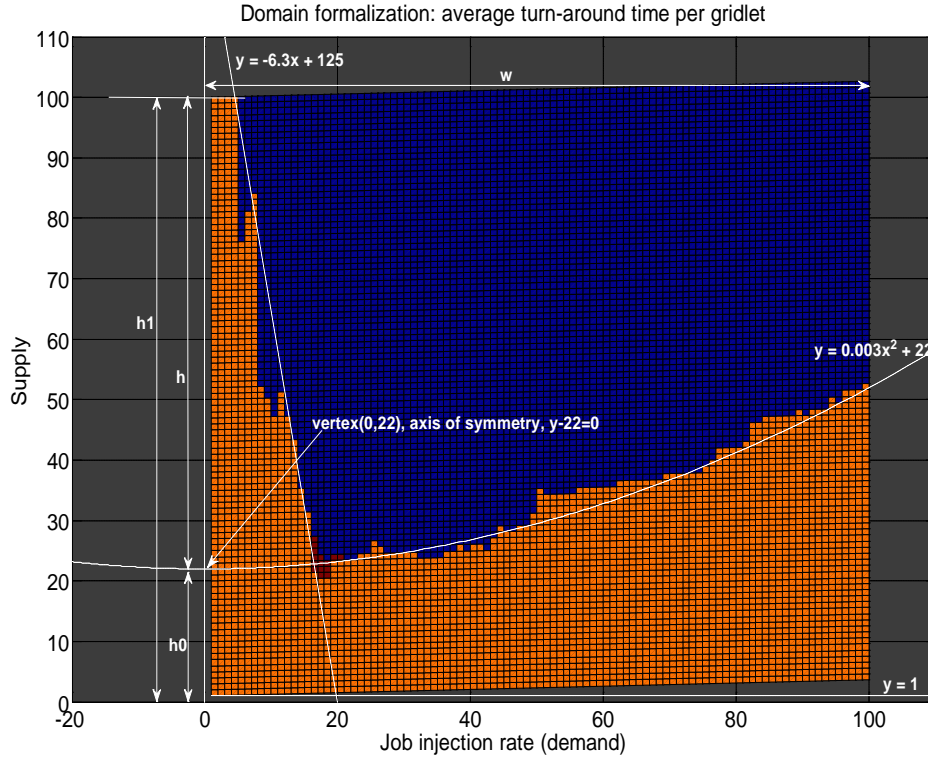


Figure 6.9: Domain Formalization for Average Turn-around Time per Gridlet

Domains of Strength: Figure 6.9 illustrates the domain formalization procedure for the metric.

Domain-1 for Commodity Market Model: The first boundary of this domain is given as,

$$d = 1 \quad (\text{irrespective of supply})$$

The second boundary passes through $(19.68, 1)$ and $(3.97, 100)$. From Eq-6.1, we obtain,

$$s = -6.3d + 125$$

The slope m of the equation is given by,

$$m = \frac{125 - s}{d} = 6.3$$

Therefore, the domain can be written as, $d \geq 1$ and $m \geq 6.3$

Domain-2 for Commodity Market Model: The nature of the first boundary of this domain is quadratic, which is given as,

$$s = 0.003d^2 + 22 \quad \text{Eq - 6.12}$$

The slope of Eq-6.12 is given by,

$$m = \frac{s}{d^2}$$

The Y-coordinate of the vertex is not at 0. Therefore, from Figure 6.9, we can write,

$$m = \frac{h}{w^2} = \frac{h_1 - |h_0|}{w^2} \quad Eq - 6.13$$

Again, from Eq-6.12 and Eq-6.13, we find,

$$m = \frac{h_1 - |h_0|}{w^2} = 0.003 \quad Eq - 6.14$$

The x-coordinate of the vertex is 0. Therefore, the y-coordinate will be 22 (solving Eq-6.12 with $d = 0$). So,

$$|h_0| = 22$$

From Eq-6.14, we get,

$$m = \frac{h_1 - 22}{w^2} = 0.003$$

Again, in terms of s and d ,

$$m = \frac{s - 22}{d^2} = 0.003$$

Now, the second boundary of the domain is given as,

$$s = 1 \quad (\text{irrespective of demand})$$

Therefore, the domain is defined as, $m < 0.003$ and $s \geq 1$

Domain for Continuous Double Auction: The first boundary of this domain will be,

$$s = -6.3d + 125$$

Where, the slope $m_i = (125 - s) / d = 6.3$

The second boundary is given as,

$$s = 0.003d^2 + 22$$

Where, the slope $m_2 = (s - 22) / d^2 = 0.003$

Therefore, the domain is formalized as, $m_1 < 6.3$ and $m_2 \geq 0.003$

6.7 Total Simulation Time Analysis

Figure 6.10 presents the comparison for total simulation time. The result is almost similar to Figure 6.8 except when both the supply and demand are low. Therefore, a similar explanation can be used to explain this figure also. The contribution of CDA is bigger in this case than that in Figure 6.8. As the supply decreases and demand is low (Region-5), the spot prices are higher compared to that of the upper part of Region-5. This forces some low-budgeted Gridlets to stay longer in the market, which ultimately extends the total simulation time. Thus, CMM cannot perform better in this region as it performed in average turn-around time per Gridlet case. However, in terms of Figure 6.8, because of the quick acceptance of higher budgeted Gridlets, the average time is still lower in this case. There are some regions (Region-4), where BAR outperforms all the other models. We ignore this, due to the little contribution. For the explanation of Region-1 and Region-2, please refer to Section 6.6. The competition rises equally for all models, due to the high demand and low supply (Region-3). This helps some Gridlets to be accepted quickly and causes the rest of the Gridlets to be failed due to unavailable resources. This brings a kind of equal performance in Region-3.

Domains of Strength: Figure 6.11 presents the domain formalization process for the total simulation time. We combine it with Region-2 for simplicity, due to the discrete contribution of Region-3.

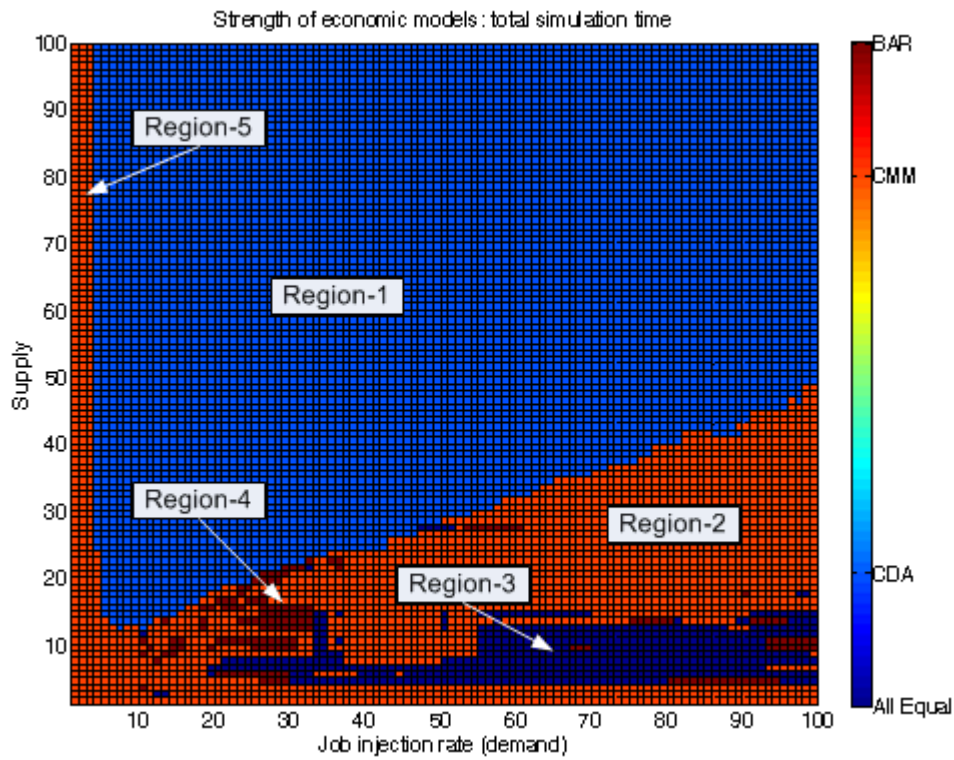


Figure 6.10: Total Simulation Time Comparison

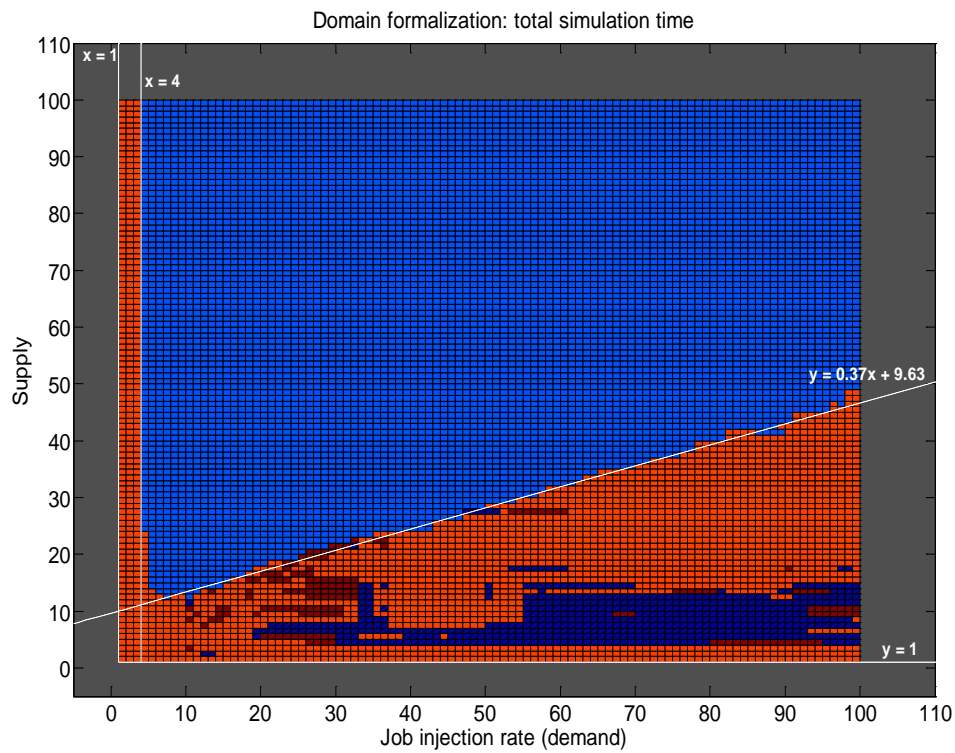


Figure 6.11: Domain Formalization for Total Simulation Time

Domain-1 for Commodity Market Model: The first and second boundaries of this domain are given by,

$$\begin{aligned} d &= 1 & (\text{irrespective of supply}) \\ d &= 4 & (\text{irrespective of supply}) \end{aligned}$$

Therefore, the domain is defined as, $1 \leq d \leq 4$

Domain-2 for Commodity Market Model: The nature of the first boundary of the domain is straight, which passes through the points $(1, 10)$ and $(100, 47)$. From Eq-6.1, we get,

$$s = 0.37d + 9.63 \quad \text{Eq} - 6.15$$

The slope $m = (s - 9.63) / d = 0.37$

The second boundary is given as,

$$s = 1 \quad (\text{irrespective of demand})$$

Therefore, we draw the domain as, $m < 0.37$ and $s \geq 1$

Domain for Continuous Double Auction: The first boundary of this domain is,

$$d = 4 \quad (\text{irrespective of supply})$$

From Eq-6.15, we get the second boundary as,

$$s = 0.37d + 9.63$$

Where, the slope $m = (s - 9.63) / d = 0.37$

Thus, the domain will be, $d > 4$ and $m \geq 0.37$

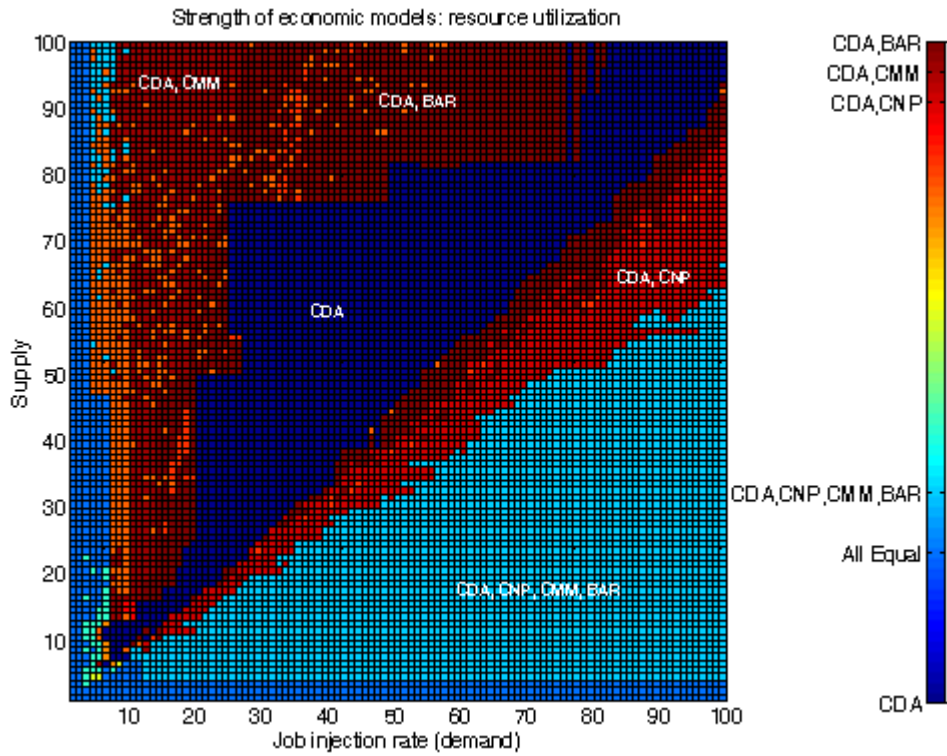


Figure 6.12: Comparison for Resource Utilization

6.8 Resource Utilization Analysis

Figure 6.12 depicts the different regions where different economic models show their strength in terms of resource utilization. As mentioned earlier, for our simulation scenario, resource utilization is just a reflection of success rate. Thus, we obtain a similar result to the success rate here. Therefore, a similar explanation is enough also for Figure 6.12.

In terms of domain formalization, a similar specification to the success rate would be enough for resource utilization.

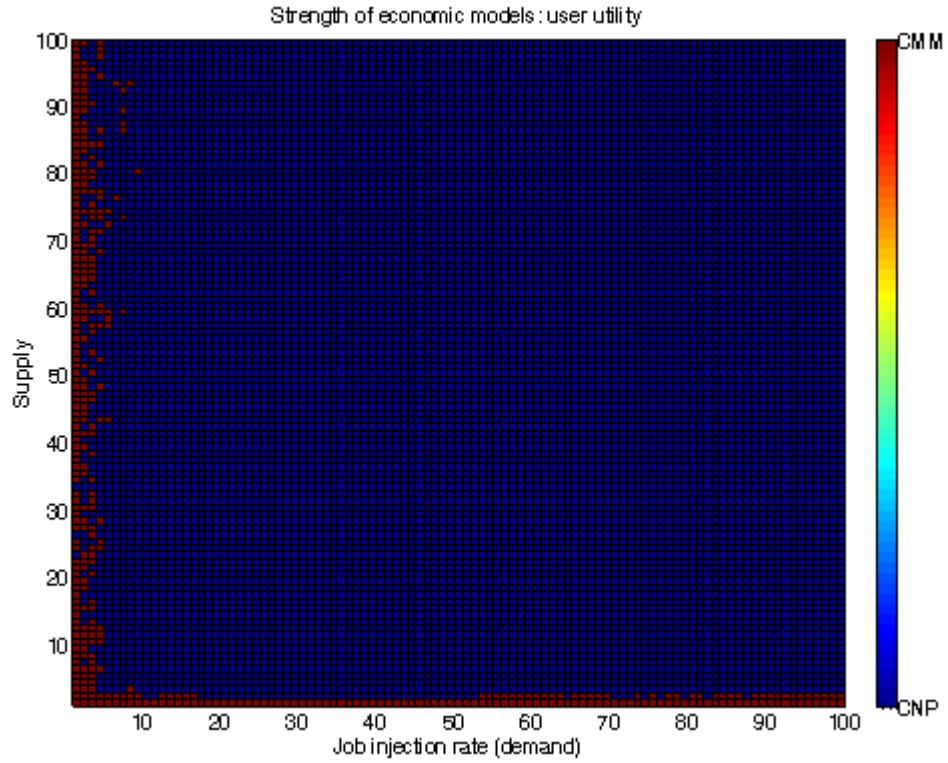


Figure 6.13: User Utility Comparison

6.9 User Utility Analysis

The comparison of the five economic models for user utility is shown in Figure 6.13. It can be observed that most of the scenarios are dominated by CNP. This is what the theory of CNP says that the CNP is suitable for optimizing user utility. If we look at the structure of CNP (Section 5.7 in Chapter 5), it provides an opportunity to the Gridlets to optimize their individual preference values through the meta-scheduling process (sub-section 5.7.1). Even the CNP supports for both time and budget optimization, and the user utility is measured in terms of price, the average utility per Gridlet is still higher compared to other models. However, when the demand is sufficiently low regardless of the supply, CMM tends to perform better. As mentioned earlier, for this region, the spot prices are low, which helps the Gridlets to trade with low costs. Thus, Gridlets receive an opportunity to maximize their utilities. Again, when the supply is sufficiently low irrespective of the demand, due to the high spot prices, in this region, only the high budgeted Gridlets are accepted. This cannot prevent lowering the utility for Gridlets much. The CNP is not performing better in this region, because the Gridlets cannot optimize their preferences much due to limited supply.

Due to the limited contribution of CMM, we assume that the whole space is dominated by CNP in this case. Therefore, the domain formalization is not much significant here.

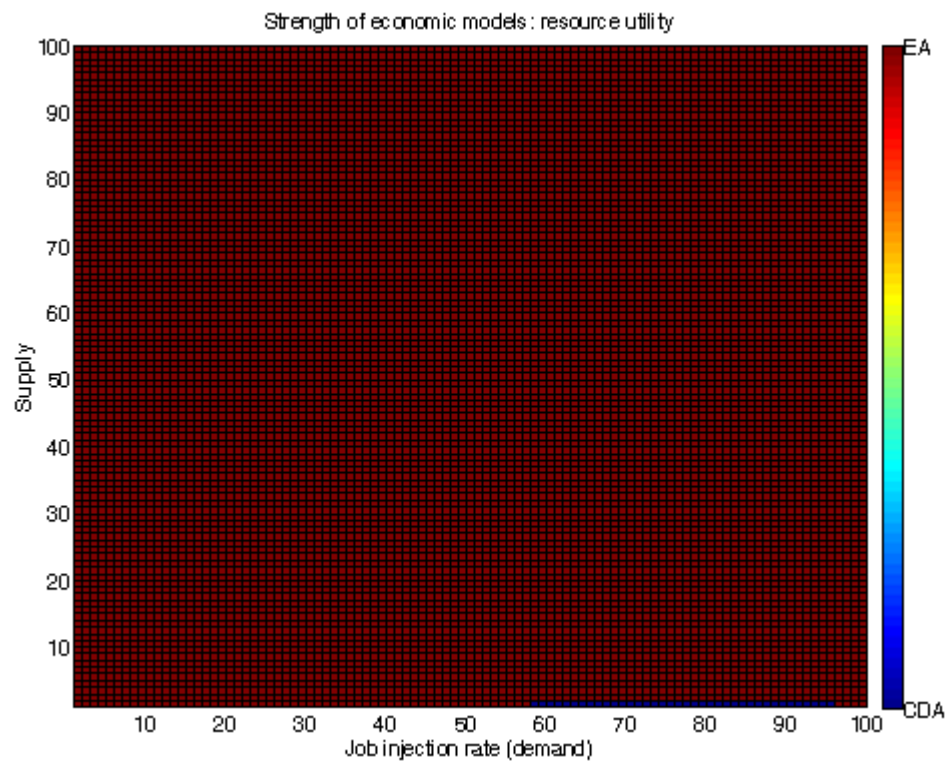


Figure 6.14: Resource Utility Comparison

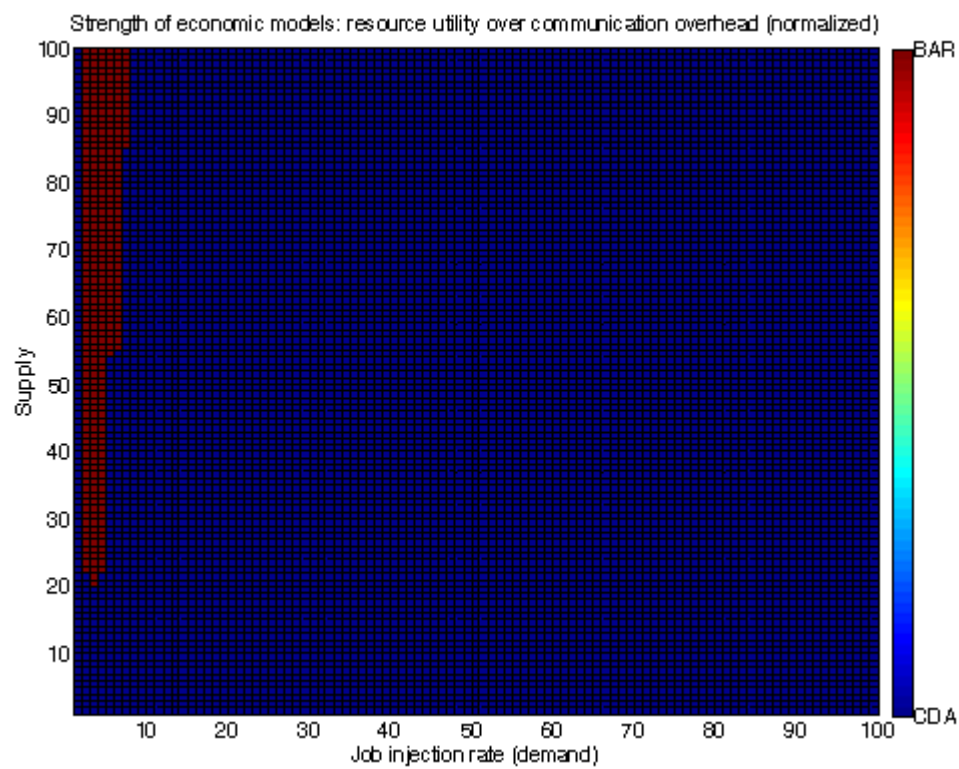


Figure 6.15: Resource Utility over Communication Overhead Comparison

6.10 Resource Utility Analysis

Figure 6.14 shows the comparison for resource utility. The EA is the protocol that outperforms other models in most of the scenarios. When the supply is sufficiently low and demand is high, CDA tends to emerge. However, due to its little contribution, we ignore this. As mention earlier, EA only focuses on higher-budgeted Gridlets through competition. This helps the model to sell the resources far above than their original reservation costs, thus maximizing the utility for resources. However, as EA is not suitable for communication overhead, a similar objective model can be considered as we considered for revenue, which is resource utility over communication overhead.

Figure 6.15 shows the contour representing resource utility over communication overhead. Once again, EA is completely absent in this measurement. The CDA and BAR are the models dominate the whole space. The CDA has already been identified as one of the suitable models for minimizing communication overhead (Figure 6.3). In addition, the price formation process in CDA, helps resources to optimize their utilities (*finalPrice* in Algorithm-3). However, when the demand is low irrespective of the supply, BAR tends to perform better. We can observe that for this particular region, CDA produces more overhead (Figure 4.16) compared to the BAR (Figure 4.14). Thus, in this case, BAR receives high opportunity to maximize the ratio for utility over communication overhead comparatively.

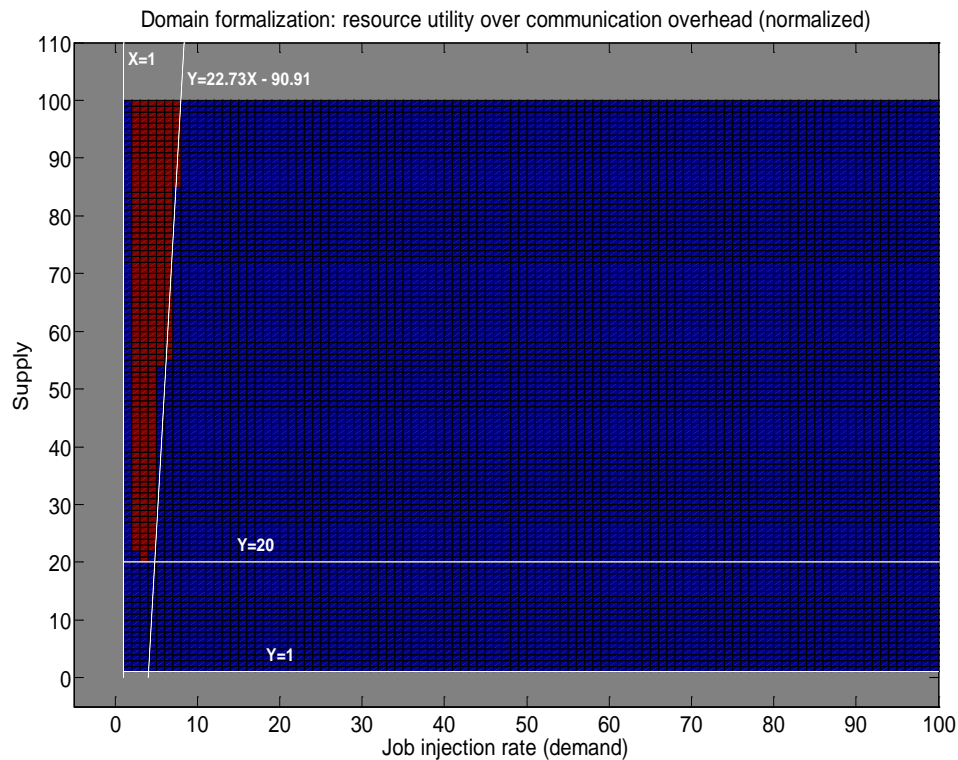


Figure 6.16: Domain Formalization for Resource Utility over Communication Overhead

Domains of Strength: Figure 6.16 presents the domain formalization for resource utility over communication overhead.

Domain for BAR: The first boundary in this case, is given as,

$$d = 1 \quad (\text{irrespective of supply})$$

The second boundary is passing through $(4, 0)$ and $(8.4, 100)$ coordinates. Therefore, from Eq-6.1, we get,

$$s = 22.73d - 90.91$$

The slope, m of this boundary is,

$$m = \frac{s + 90.91}{d} = 22.73$$

The third boundary is given as,

$$s = 20 \quad (\text{irrespective of demand})$$

Therefore, the domain is defined as, $d \geq 1$, $s \geq 20$ and $m \geq 22.73$

Domain for CDA: The boundaries for this domain is defined as,

$$\begin{aligned} s &= 22.73d - 90.91 \\ s &= 1 \quad (\text{irrespective of demand}) \\ s &= 20 \quad (\text{irrespective of demand}) \end{aligned}$$

Therefore, the domain is, $s \geq 1$, $m < 22.73$ or, $s < 20$

6.11 Social Welfare Analysis

As mentioned earlier, social welfare is the combined utility made by both Gridlets and resources in the environment. The contour diagram (Figure 6.17) demonstrates the regions representing the strengths of different economic models for social welfare. In this case, mainly CDA and CNP are the models those outperform all the other models in two different regions. When supply and demand are low (Region-3), BAR outperforms other models. However, due to its little contribution, we ignore this. In terms of CNP, the welfare mainly comes from user utility and for CDA; it comes from both user and resource utilities.

When the supply is high regardless of the demand (Region-1), CDA performs better. The respective Ask-Order-Book is filled with low-cost resources, due to the high supply, which helps to accept a high number of Gridlets with low costs. Thus, a majority of the welfare

comes from user utility. When the supply decreases and demand is high (Region-2), satisfying a high number of Gridlets by CDA becomes hard due to (1) high prices in the book and (2) shortage of resources. Even the supply is low, CNP in Region-2 still gets a chance to optimize user utility, which is higher compared to the combined utility (user + resource) in CDA. As both the supply and demand decreases, the welfare for CNP/CDA decreases proportionally. A low number of Gridlets are accepted, due to the low supply and demand, which results the aggregated welfare low in Region-2.

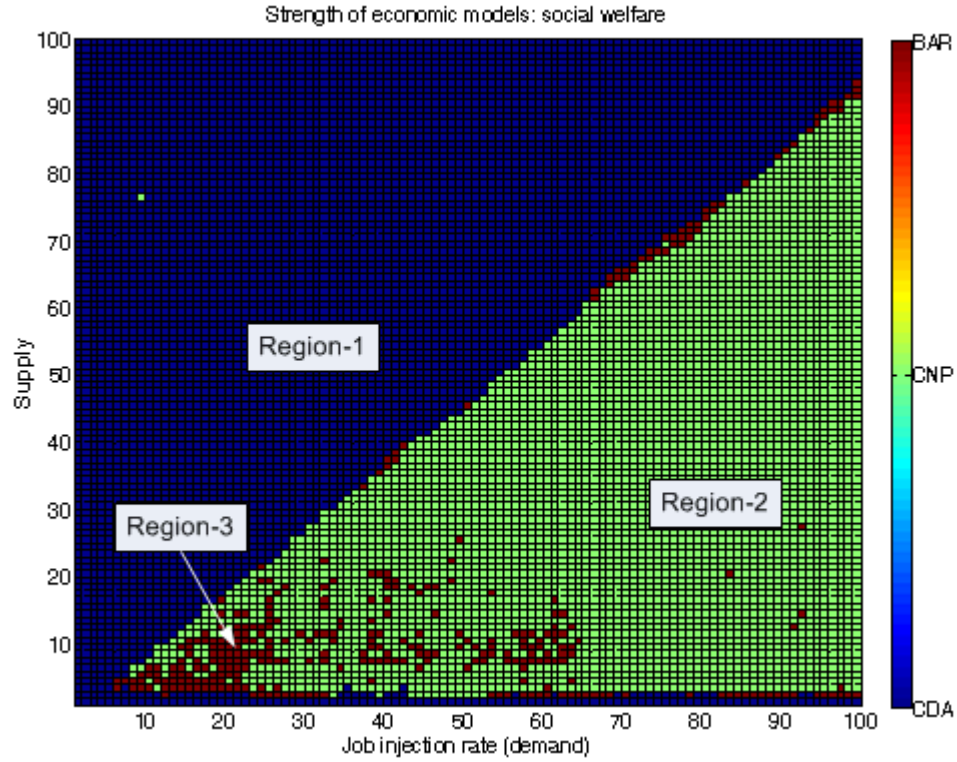


Figure 6.17: Social Welfare Comparison

Domains of Strength: Figure 6.18 illustrates the domain formalization procedure for social welfare.

Domain for Continuous Double Auction: The first boundary of this domain is given by,

$$d = 1 \quad (\text{irrespective of supply})$$

The second boundary is passing through the origin and its nature is straight.

$$s = 0.91d$$

The slope $m = s/d = 0.91$

Therefore, the domain is formalized as, $d \geq 1$ and $m \geq 0.91$

Domain for Contract Net Protocol: We just identified the first boundary, which is,

$$s = 0.91d$$

The slope $m = s/d = 0.91$

The second boundary is given by,

$$s = 1 \quad (\text{irrespective of demand})$$

So, the domain is, $m < 0.91$ and $s \geq 1$

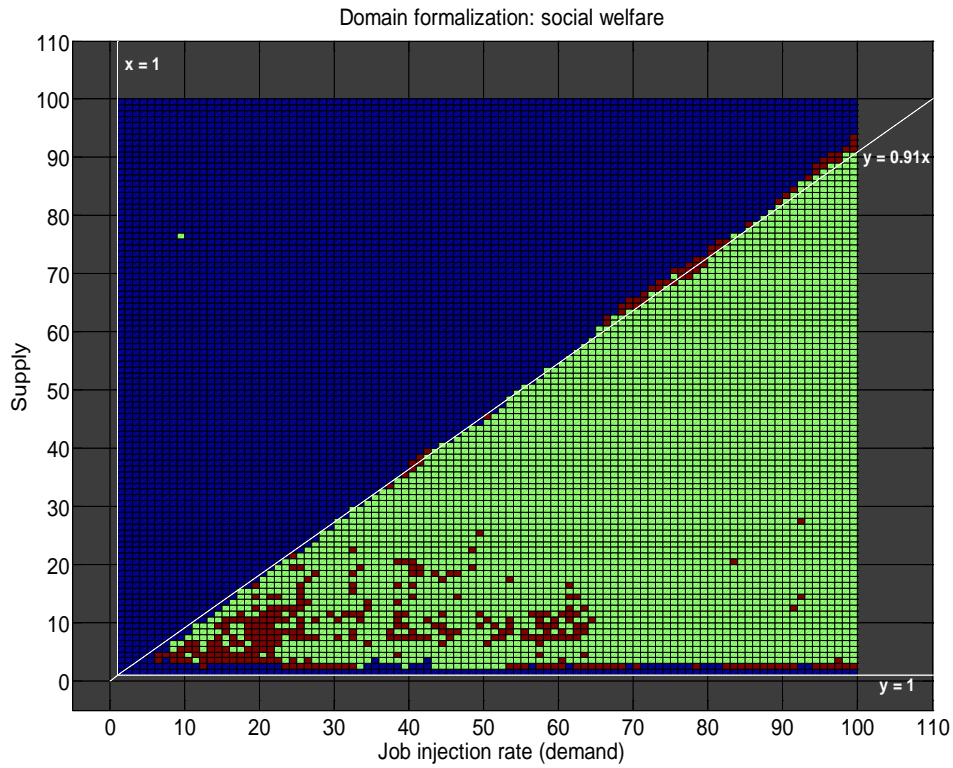


Figure 6.18: Domain Formalization for Social Welfare

6.12 Discussion and Research Proposal

We summarize our discussion on the domains of strengths of most widely proposed economic models in Grid computing in Table 6.1.

Table 6.1: Domains of Strengths of Economic Models in Grid Computing

Performance Metric	Economic Model (Domain of Strength) Space (s, d)
Revenue	<i>EA (whole space)</i>
Revenue over communication overhead	<i>BAR</i> $\{d \geq 1 \text{ and } ((s+46.06)/d) \geq 13.01\}$, <i>CDA</i> $\{((s+46.06)/d) < 13.01 \text{ and } ((s-1.93)/d) \geq 0.07\}$, <i>CNP</i> $\{((s-1.93)/d) < 0.07 \text{ and } s \geq 1\}$
Communication efficiency	<i>CMM</i> $\{d \geq 1 \text{ and } ((s+27.7)/(d-2.69)^2) \geq 0.65\}$, <i>CNP</i> $\{((d-1.24)^2/(s+2.34)) < 0.32 \text{ and } s \geq 1\}$, <i>CDA</i> $\{((s+27.7)/(d-2.69)^2) < 0.65 \text{ and } ((d-1.24)^2/(s+2.34)) \geq 0.32\}$
Success rate	<i>All Equal</i> $\{1 \leq d \leq 8 \mid 1 \leq s \leq 4\}$, <i>CDA or CMM</i> $\{d > 8 \text{ and } ((s+15)/d) \geq 4\}$, <i>CDA or BAR</i> $\{((s+15)/d) < 4 \text{ and } s/d \geq 0.83\}$, <i>CDA or CNP</i> $\{0.66 \leq s/d < 0.83\}$, <i>CDA or CNP or CMM or BAR</i> $\{s/d < 0.66 \text{ and } s \geq 4\}$
Average turn-around time per Gridlet	<i>CMM</i> $\{d \geq 1 \text{ and } ((125-s)/d) \geq 6.3 \mid ((s-22)/d^2) < 0.003 \text{ and } s \geq 1\}$, <i>CDA</i> $\{((125-s)/d) < 6.3 \text{ and } ((s-22)/d^2) \geq 0.003\}$
Total simulation time	<i>CMM</i> $\{1 \leq d \leq 4 \mid ((s-9.63)/d) < 0.37 \text{ and } s \geq 1\}$, <i>CDA</i> $\{d > 4 \text{ and } ((s-9.63)/d) \geq 0.37\}$
Resource utilization	<i>Similar to the success rate</i>
User utility	<i>CNP (approx. whole space)</i> <i>CMM (negligible)</i>
Resource utility	<i>EA (whole space)</i>
Resource utility over communication overhead	<i>BAR</i> $\{d \geq 1, s \geq 20 \text{ and } ((s+90.91)/d) \geq 22.73\}$, <i>CDA</i> $s \geq 1, ((s+90.91)/d) < 22.73 \mid s < 20\}$
Social welfare	<i>CDA</i> $\{d \geq 1 \text{ and } s/d \geq 0.91\}$, <i>CNP</i> $\{s/d < 0.91 \text{ and } s \geq 1\}$

One can observe from Table 6.1 that a single economic model does not perform better all the time. CMM has some potential for minimizing communication overhead and time. However, in terms of revenue, the model is not strong enough to compete with the other models. Even the BAR is not suitable for minimizing communication overhead; it contributes in revenue considering communication overhead metric. The model further shows its strength in the similar region for resource utility. This implies the suitability of the model from a provider's point of view. Even EA generates the highest utility/revenue for resources, due to its higher

communication overhead, the model has fully been degraded for other metrics. Therefore, there is a chance that the model might be undesirable by the resource community. In most of the performance metrics and a greater part of the simulation space, CDA performs better. Even CDA has significant contribution for revenue over communication efficiency compared to other models, this contribution decreases as we scale up the supply and demand function and the other two models – BAR and CNP take over this contribution gradually (Figure 6.1). The reason is that the CDA exhibits a downward, and BAR and CNP exhibits an upward trend in this case. Therefore, for an immensely high supply and demand, the performance for CDA might become marginal.

CNP, on the other hand, is also found suitable for revenue considering communication overhead, user utility and social welfare performance metrics. In terms of success rate or resource utilization, more than one model perform equally better compared the others. This indicates the more/less suitability of all the economic models except EA for this particular metric. In addition, as we have obtained clear trends of the model' strengths, the formalization models further ensure the feasibility of accounting the strength for extended unknown scenarios. Our findings show the compatibility with the existing literature, i.e., in a highly dynamic and distributed environment such as Grid, a single model is not suitable to cope with every scenario (Haque et al., 2011). We draw the conclusion of our experimental findings with the following propositions.

Proposition 6.1: There will be metrics for which no economic model, eM performs better than the others, at least not under all circumstances, G. That will depend also on the domain, g.

$$\forall (G) (\exists (g) (\text{one eM outperforms other models}))$$

For example, in terms of communication efficiency, we have identified three different regions where three economic models (CMM, CNP and CDA) perform better compared to each other.

Proposition 6.2: For a possible set of performance metrics, ρ , there will be metrics, $\varphi \subseteq \rho$ for which one economic model, eM outperforms other economic models at all circumstances.

$$\forall (\rho) (\exists (\varphi) (\text{one eM outperforms other models}))$$

For example, as we have seen, for user utility, CNP always outperforms other models.

From the discussion above, we find the opportunity of optimizing different performance metrics in a Grid computing environment by utilizing the potential of different economic models in different scenarios. *This helps us to deal with the fourth issue as identified in Chapter 2.*

However, *how one would be able to optimize his/her objective function(s) using different economic models in a highly dynamic environment* is a question. To answer this question and at the same time, to tackle the fourth issue, we make the following proposals.

- We propose a protocol-generic adaptive framework that couples multiple economic models suitable for Grid resource management. The framework is able to switch from one economic model to another dynamically depending on the models' domains of strengths. For example, if a Grid network has limited bandwidth and desired to minimize its communication overhead, the network can follow our findings for "communication efficiency" metric. That is, if the network sometimes notices an equal amount of supply and demand, it can use CDA. Again, if the network sometimes notices that the demand has been decreased and supply is available, it can switch to CMM; because CMM, in this region, generates lower communication overhead (refer to Figure 6.3). Likewise, the network can minimize its overall communication cost. Again, to support such a framework, the broker and resource models must have the adaptive capabilities to deal with different economic models. We will discuss the details in Chapter 7.
- Realizing the dynamic and distributed nature of the Grid, the switching must be conducted autonomously and without any considerable delay. Thus, secondly we propose a switching agent that senses the Grid environment and dynamically switches from one economic model to another model depending on the models' domains of strength (Table 6.1).

6.13 Conclusion

Economic models show the efficiency of distributed resource management in Grid computing. Thus, evaluating the performances of different economic models in terms of a wide range of scenarios is crucial. In this chapter, we analyzed various performances of five most widely proposed economic models in the Grid. Through a comprehensive comparative analysis, we realized that a single model is not suitable to meet the requirements of diverse scenarios in a Grid. We described the scenarios in terms of supply and demand as economic models are generally influenced by them. We further drew the domains where a one model outperforms all the other models. The domains were formalized mathematically for a feasible extension of our identification. Thereafter, we addressed the possibility of optimizing different performance metrics by employing multiple economic models in a Grid environment. In this regard, finally, we proposed a novel switching framework that automatically senses Grid

environment and dynamically switches from one economic model to another model depending on the models' domains of strengths.

In this chapter, we identify and formalize the domains of strengths of widely proposed economic models in Grid computing to deliver a comprehensive understanding about the models. In the following chapter, we discuss the design and development of the proposed switching framework followed by a detailed experimental analysis suitable for evaluating the framework.

A Switching Framework for Optimization in Economic Grids

To deal with the fourth issue identified in Chapter 2, this chapter discusses the necessary elements towards developing an economic-based protocol-generic Grid computing framework. We incorporate the five economic models in the framework. We discuss the extended version of the broker and resource models to support the framework. The framework is also analyzed in terms of autonomous switching between different economic models. A switching agent and its role in a dynamic Grid environment in terms of optimizing different objective functions by the provider, is also discussed. Finally, a series of experiments is conducted to evaluate the effectiveness of the framework.

7.1 Introduction

Economic models show the promise of efficient resource management in Grid computing. The models are different from one another in terms of their working principles and pricing strategies. We identified that a single model is not suitable to deal with a wide range of scenarios in the Grid. An extensive comparative analysis on the widely proposed economic models helped us to identify the domains of strengths of different economic models. Therefore, the diversity of the Grid might require a generic framework to deal with different scenarios using different models. We use the models as a solution parameter to deal with this diversity of the Grid.

In spite of the varied performance of different economic models in the Grid, a few researches have considered the requirement of a protocol-generic adaptive framework (Brandic et al., 2008, Resinas et al., 2006). However, these works are just a proposal and concentrate more on the structural requirements of the framework rather than investigating the performances of different models. Analyzing the performances of different models, we identify the opportunity of optimization through utilizing the potential of the models in a dynamic Grid environment. Therefore, in this chapter, we design and develop a generic framework that couples five widely proposed economic models in the Grid. We provide the necessary components to understand the adaptive nature of the framework— switching from

one economic model to another in a dynamic environment. To facilitate the optimization process, agent technology is incorporated in our system.

The incorporation of agent in Grid computing is not a new concept (Foster and Kesselman, 2003, Foster et al., 2004, Cao et al., 2002, Cao et al., 2001, Shen et al., 2002). The Grid and agent basically share the common vision of large scale open distributed systems (Foster et al., 2004). They are capable of effectively and dynamically deploy and redeploy computing resources to solve computationally complex problems. In addition, agents can make crucial decisions (re-track routing) during unpredicted failures (e.g. Network failure) without any considerable delay in order to optimize system utilization. In this chapter, we consider the agent mainly in terms of decision making relative to our optimization problem.

As we design the agent to provide decision on when and which economic model to switch, we refer the agent as switching agent. The agent has been developed in consistent with the framework, which is, the agent is able to sense the Grid environment and provide decision to the system about suitable economic models and the system on the other hand is able to configure its behavior according to changes in models. To facilitate the decision process, the domains of strengths of different models as identified in the previous chapter have been imported to the agent. The agent has then been learnt about the application models of these domains. To evaluate the performance of our switching framework, we conduct an experimental analysis to prove that the framework is able to optimize different performance metrics in a dynamic environment. The experimental results show promising outcomes. Based on these outcomes, we are inspired to investigate the opportunity of deploying the switching mechanism in Cloud computing environment. The rest of this chapter is organized as follows:

In Section 7.2, we describe the opportunity of optimization using multiple economic models and some motivations for switching framework. Section 7.3 presents a prototype of our framework and a brief overview of it. In Section 7.4, we describe the development procedure of our switching framework including the working principle of the switching agent. Section 7.5 explains the experimental analysis in terms of optimizing various performance metrics. The challenges and opportunities in deploying the switching mechanism is presented in Section 7.6. Section 7.7 draws the conclusion.

7.2 Inspiration for Optimization and Switching Framework

This section provides the motivation for optimizing different performance metrics in a dynamic Grid computing environment. It further provides the motivation for switching framework to enable the optimization process. Grid computing shares resources across geographical boundaries and possesses users from around the world. Expecting constant

performance all the time in such a dynamic and large scale computing platform is, therefore, subjected. An extensive economic-based research has been conducted to deal with this issue from the Grid's inception until now. Various optimization techniques are applied to deliver a sustainable computing platform. However, a very few research have studied with the fundamentals of economic-based approaches in the Grid. In this thesis, we propose a new dimension of optimization mechanism that works in elementary level of the models. Our work is not confined in a single model rather seek for the opportunity of utilizing the potential of different models in different scenarios. The following examples provide some motivation for optimization using multiple models in a dynamic Grid.

Example 1: Let us imagine a Grid network that would like to maximize its revenue and minimize communication overhead. The communication service for a network is not free in general. In addition, communication overhead has become a topic by many researchers in distributed resource collaboration in the Grid (Foster et al., 2008, Buyya et al., 2002, Andrzejak and Zhichen, 2002). Through a comparative analysis, we have identified that there three different economic models perform better in three different regions in terms of revenue over communication overhead objective function (Figure 6.1 in Chapter 6). Now, the possibility of maximizing the revenue is higher if the network employs the three different economic models in their respective domains rather than keep relying on a single model throughout the Domains.

For example, if the network notices that its current demand is low regardless of its supply, it can employ BAR as BAR has been identified as the suitable model for this scenario and for the optimization function (Figure 6.1 in Chapter 6). Likewise, if sometimes the network notices that its supply has been decreased moderately and demand has been increased, it can switch to CDA. Again, when the supply becomes sufficiently low and demand is still high, the network can switch to CNP. As a result, the network would be able to utilize the potential of different economic models in different scenarios, thus maximizing the defined objective function in general.

Example 2: Similarly, if a particular Grid network would like to satisfy both its users and providers as a part of its business strategy, the network might be interested to follow our findings for social welfare. For instance, if there are enough supply and limited demand in the network, it can use CDA. Again, if sometimes the supply decreases and demand increases, the network could switch the model to CNP. The network must have the capability to employ two different economic models in two individual regions (Figure 6.17 in Chapter 6). Similar optimization is also possible for other criteria where different models perform better

compared to each other at different regions. Such an optimization procedure would provide incentives to Grid providers, which in turn would help to deliver a viable market mechanism for future computing platform. However, to be consistent with the nature of Grid, such a multi-models architecture gives rise to the following questions.

- What motivates a Grid to follow such an optimization procedure?
- How one will utilize the potential of different economic models in a highly dynamic environment?
- Who is responsible to keep track of the scenarios?
- Who is responsible to decide which model to use, and when to use?
- Who is responsible for the corresponding effects of changing models?

To deliver the answers of these questions, we propose a switching model that dynamically switches from one economic model to another and able to adapt with the changing circumstances in the environment. To facilitate the switching process, we design a switching agent, which automatically decides which model to use when and for what purpose. A prototype of our switching model is presented in the following section.

7.3 A Prototype for Switching Framework

To understand the working behavior of our switching framework, in this section, we design a prototype that describes the key functionalities performed by the framework.

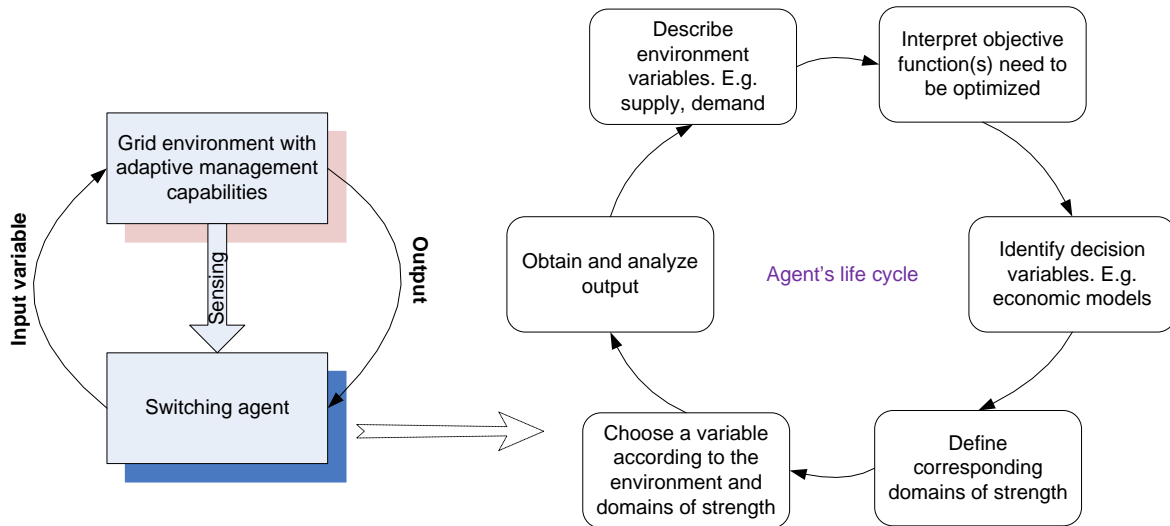


Figure 7.1: Agent-driven Switching Framework for Grid Resource Management

A prototype of our switching framework is illustrated in Figure 7.1. There are two different parts we will describe here – the Grid environment with adaptive management capabilities and the switching agent. The roles (broker, resource) in such an environment must

have capabilities to deal with different economic models dynamically. The broker, in this case, should be modeled in a way so that it can generally submit its request to a resource, bargain with the resource, or compete with other brokers whenever necessary. Therefore, the parameters and the organization of a broker model have to be designed in a way so that it can adapt with changing circumstances in the environment. A resource model, on the other hand, needs to have bargaining capability, generating *asks* continuously (for CDA) or evaluating a broker's request in general. The Auctioneers, in this case, must be prepared so that they can start processing as soon as any one of them is invoked by the system. In addition, every broker, resource and Auctioneer must understand the language of the agent; because the agent is providing crucial information to them. For example, the agent informs the participants in the environment about whether to continue with the current model or to get prepare for a new model. Therefore, the ability by the participants to interpret the agent's message and to work accordingly is significant.

The most crucial role played in the environment is the switching agent. There are six different stages in the agent's life cycle. The agent can sense (keep track of supply and demand in the network) the Grid environment in two different ways – time-based and event-based. In a time-based mode, the agent senses the environment at every specific period whereas in an event-based approach, the agent senses the environment whenever there is a change in the environment. We consider the event-based approach in our work and like to explore the time-based scenario in the future. Upon receiving an event the agent starts its first activity, which is analyzing the output (Figure 7.1). The logic behind analyzing the output is that whether the environment is really making the optimization with the input provided by the agent. For simplicity, we ignore this part in our work and consider the other procedures in practice by the agent.

In an event-based system, whenever there is a change in supply or demand, the agent is invoked to provide decision on whether to continue with the existing model or to switch to another model. The agent, at first, interprets the current supply and demand, and the objective function that needs to be optimized. Based on the objective function, the agent, then, identifies the available economic models those can help in optimizing the defined objective function, where the economic models are considered as decision variables. For example, in terms of communication efficiency, there are three different decision variables; CMM, CDA and CNP (Table 6.1 in Chapter 6). Each of these models shows the strength over other models in its respective regions for this particular metric. The following process by the agent is to identify the corresponding domains of strength of the decision variables. Depending on the values of current supply and demand, the agent, then, verifies the appropriate domain and

decides the model corresponding to that domain. If the decided model is different from the previous model, the new model works as an input variable to the environment (Figure 7.1). Otherwise, the system continues with the existing model. The process continues until and unless the supply or demand becomes empty from the environment. In the following section, we describe the development process of our switching framework.

7.4 Towards Developing a Protocol-generic Switching Framework in Grid

The rising demand of Grid computing has led to the requirement of a framework that support dynamic organization of different models. This section represents a framework suitable for dealing with five different economic organizations dynamically. First, we describe the user, broker and resource models in terms of fitting into the framework. Then, we describe the parameters those can be re-configured while switching from one economic model to another in order to understand the adaptive nature of the framework. Finally, we explain the role of switching agent in the framework.

7.4.1 User, Broker and Resource Models

The user, in this case, must define the parameters required to deal with all the economic models in the framework. This helps the corresponding broker to be adaptive with the environment where the interaction principles may change over times. Apart from some common parameters, the user must specify the parameters more particular to a model. Figure 7.2 presents such a user model.

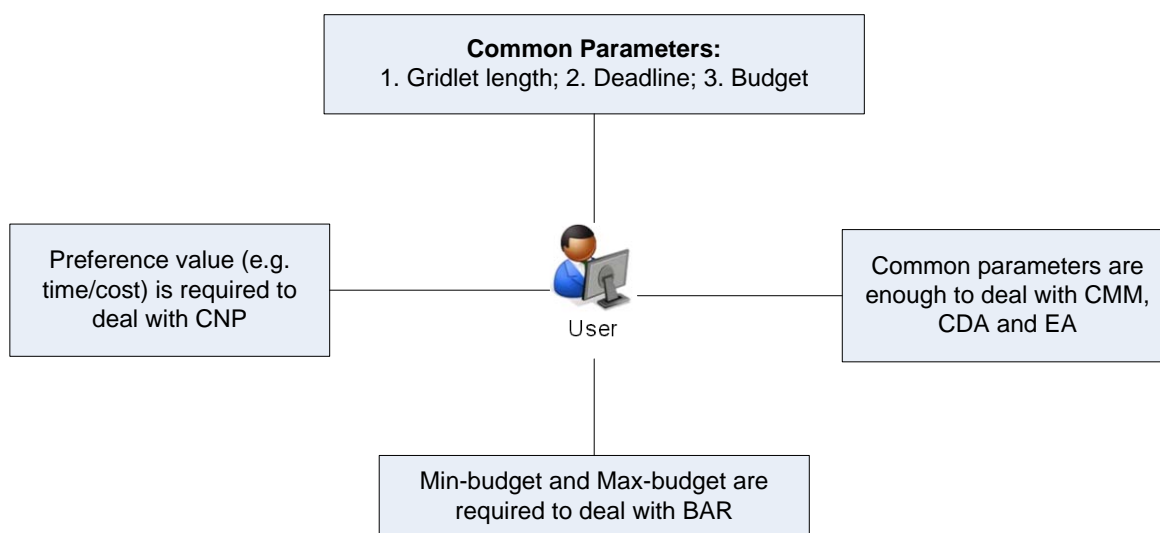


Figure 7.2: The User Defines QoS Parameters for Switching Framework

The consideration of additional parameters helps the user to maximize its success probability in the environment, as it may need to deal with multiple economic models now. There is not much work need to be done by the user; because its corresponding broker conducts all the activities on behalf of the user. The behavior of the broker is regulated according to the models in the environment. As mentioned earlier, the broker must understand the language of the agent to know the recommended model and to behave accordingly. The broker-model suitable for our switching framework is described in Algorithm-4.

Algorithm-4: Extended Broker Model:

Switch (Track messages using **Tag** values)

```
{
    Case: Message Received from the Agent //initiated by Algorithm-6
        Cast the message and identify the Recommended protocol
        if (Current protocol != Recommended protocol)
        {
            Stop sending request from this broker //to stop the current protocol
            Wait for a while //this is important if the broker is currently busy talking to any resources
            De-activate the parameters related to Current protocol
            if(The broker is still looking for the suitable resource)
            {
                let delay = 0;
                Current protocol = Recommended protocol
                Send a message to itself to prepare for the new protocol
                sendMessage(this.Broker, delay, Tag.Current protocol);
            }else terminate the broker
        }else keep continue with the Current protocol
    Break;

    Case: Commodity Market Protocol
        Prepare job specification according to this protocol
        Start sending the request to the resources one by one
    Break;

    Case: Bargaining Protocol
        Prepare job specification according to this protocol
        Start sending the request to the resources one by one
    Break;

    Case: English Auction Protocol
        Prepare job specification according to this protocol
        Send the request to the Auctioneer
    Break;

    Case: Continuous Double Auction Protocol
        Prepare job specification according to this protocol
        Send the request to the Auctioneer
    Break;

    Case: Contract Net Protocol
        Prepare job specification according to this protocol
        Send the request to the Manager
    Break;
}
```

For CMM and BAR, Algorithm-1 (in Chapter 4) works along with Algorithm-4. The Algorithm-4 is in action until the simulation finishes or the broker is terminated by itself. Apart from this algorithm, there are three different modules, which work in conjunction with this algorithm. The modules are *Observer-1*, *Observer-2* and *Manager*. The *Observer-1* deals with activities for BAR, whereas, *Observer-2* manages the activities for EA. The *Manager* describes the procedures to be done by the broker when it is CNP. The agent about the change of model in the environment also notifies the resources. Therefore, the resources must also have capabilities to deal with all the five economic models dynamically. Algorithm-5 describes the extended resource model.

Algorithm-5: Extended Resource Model:Switch (Track messages using **Tag** values)

```

{
    Case: Message Received from the Agent //initiated by Algorithm-6
        Cast the message and identify the Recommended protocol
        if (Current protocol != Recommended protocol)
        {
            Start rejecting request from any broker with the Current protocol
            Wait for a while //this is important if the resource is currently busy talking to any broker
            De-activate the parameters related to Current protocol
            if(The resource is not committed to any broker)
            {
                let delay = 0;
                Current protocol = Recommended protocol
                Ask itself to prepare for the new protocol
                sendMessage(this.Resource, delay, Tag.Current protocol);
            }else remove the resource from the available resource list
        }else keep continue with the Current protocol
    Break;

    Case: Commodity Market Protocol
        Prepare itself for CMM (e.g. to initiate spot price determination process)
        Start receiving requests from the brokers
    Break;

    Case: Bargaining Protocol
        Prepare itself for BAR (e.g. to initiate the corresponding Observer and to aggregate the parameters
            to compute bids over rounds)
        Start receiving requests from the brokers
    Break;

    Case: English Auction Protocol
        Get prepared for the auction (e.g. to notify the Auctioneer about the resource properties)
        Follow the commitment if the auction is succeed
    Break;

    Case: Continuous Double Auction Protocol
        Get ready for the protocol (e.g. to start generating asks and sending them to the Auctioneer
            continuously)
    Break;

    Case: Contract Net Protocol
        Be prepared for the protocol (e.g. to receive invitation from the manager and prepare response for
            that)
    Break;
}

```

Algorithm-5 helps a resource to deal with the different economic models. If the new model is any of the auctions (e.g. EA or CDA), the agent first notifies to the Auctioneer to be ready. Afterwards, it notifies to brokers and resources about the new model in the environment, because, in order to accept the request/call/proposal from the brokers or resources, the Auctioneer must be ready beforehand. Now, we discuss the characteristics of our framework when it switches from one economic model to another model. As we are dealing with five different models, there will be 20 possible ways of switching from one model to another model. Each of the ways is crucial to consider due to the distinct characteristics of

individual models. The following sub-section describes the framework in terms of an autonomous computing system.

7.4.2 An Autonomous Switching Framework

The rising complexity in computing systems has led to the challenge of a feasible understanding about the future of those systems to deal with dynamic and conflicting demands in the systems. The concept of “autonomic computing” is introduced to deal with this challenge (Kephart and Chess, 2003). An autonomic computing system is typically defined as a system that can evolve by self-management process in order to meet the system-administrator’s goal. In this sub-section, we aim to discuss such an autonomous system that is able to deal with dynamic integration of different economic models in an environment. We discuss our framework in terms of the main objectives of an autonomous system. The objectives are:

- **Self-configuration:** This describes the ability of the system to adapt with unknown scenarios automatically. This may include installation, configuration or integration. Individual components must know how to be introduced into the new configuration by themselves and the rest of the system should adapt with their presence without any disturbance in the system. We have already presented such a broker and resource models in the previous sub-section. We described them in terms of self-organization process to deal with multiple economic models seamlessly. For example, when either a broker or resource is notified about a new model, it starts configuring the parameters to deal with the new model, thus adapting with the new environment. Along with the participants, the system itself must also understand the rationality behind the changed behavior of the participants in the environment. The role performed by the system in such an environment will be explained in Algorithm-6.
- **Self-optimization:** An autonomous system must know how to improve the overall performance of the system. To achieve this, it monitors, experiment, or tune their parameters; overall, it tries to learn from the environment. In this thesis, we performed an extensive experimental analysis to understand the domains of strengths of different economic models in Grid computing. We, then, identified the possibility of optimization by dynamically tuning between different economic models. To this end, we developed the switching framework, which is able to switch automatically from one economic model to another model in order to optimize pre-defined objective functions. To facilitate the optimization process, we develop a switching agent, which is described in the following sub-section.

- **Self-healing:** This describes the problem of dealing with bugs or failures automatically by the system. For example, the system should have the ability to detect, diagnose, and repair such problems probably through analyzing the log files. As we are not dealing with such issues in our system, we assume that our system is free from such errors. However, we would like to explore the position of our system against such unpredicted failures in future.
- **Self-protection:** This describes, protecting itself from malicious program. This also discusses the protection from the problems those remain unsolved from the self-healing process. The system must be able to detect such attacks and take necessary steps to avoid or mitigate them. Once again, as we are not focusing any security issues in our work, we ignore this part from our system.

Upon receiving the message of a new model from the agent, the system starts processing its subsequent procedures. The procedures include dealing with the parameters, to control the behavior of the current model and inform the participants in the environment about the new model. Algorithm-6 describes these in detail.

The notification of a new economic model to the participants must occur without any considerable delay in the system. Otherwise, the application of the new model at desired scenario might be disturbed. We have organized our algorithms accordingly so that it can maximize the utilization of the switching model as much as possible. However, in reality, the dynamic nature of network latency might lead to some inefficiency in receiving the message of a new model by the participants quickly. We would like to explore such a behavior in the future. We have observed some undesirable situations in the simulation in terms of switching from CDA to any other models. This happens due to the force of CDA. The moment CDA starts processing, the resources start generating *asks* continuously; therefore, by the time the resources are informed about a new model, CDA already have generated a large number of *asks* for which a set of acceptations might incur. This fails to enable the system to apply the new model to a more desirable scenario. The deviation from the desired scenario becomes more significant if CDA stays longer in the environment, and then the switching happens. This characteristic, while switching from other models is quite negligible.

It is also possible that a set of users will not be interested with the new model and, therefore, would like to leave the market as a part of their optimization strategies. In such a case, reflecting on the desired supply and demand ratio might be harder. As mentioned earlier, we are focusing on the provider side. Therefore, we would like to explore some strategic behavior by our framework to tackle such a scenario in the future. For now, we assume that

users do not leave the market and they are happy as long as they can execute their jobs within the defined constraints. The main parameter, which drives the switching framework, is the optimization function by the Grid provider. The switching agent must be informed about this function before the simulation starts to perform the decision process properly. The following section describes the functions implemented by the switching agent.

Algorithm-6: Switching from Protocol-1 to Protocol-2:

```

The agent suggests protocol-2 for current network scenario
Stop initiating new auction instances if the protocol-1 is an auction
Wait for a while as some processes related to protocol-1 might be still in execution
De-activate the global parameters related to protocol-1

//Send a message about the protocol-2 to every broker in the environment
Foreach(broker in the broker-List) //we are updating the broker-List continuously as soon as a broker is terminated
                                     from the environment
    {
        Send a message about protocol-2 to the broker without any delay
    }

//Send a message about the protocol-2 to every resource in the environment
Foreach(resource in the resource-List) //we are updating the resource-List continuously as soon as a resource
                                     is committed to a broker
    {
        Send a message about protocol-2 to the resource without any delay
    }

```

7.4.3 Switching Agent

“Learning” – is an essential term in the field of agent technology. It is now a closely related topic between natural and computational systems. Learning is typically used to mean the improvement of a system based on experience. However, depending on the nature of a system, learning methods could vary from one another. In this sub-section, we describe the agent and identify the most suitable learning method out of a set of available methods that matches with the method that our agent adopts. To understand the basis of this section better, we provide a definition of learning according to Gerhard Weiss (Weiss, 2000),

“The acquisition of new knowledge and skills and the incorporation of the acquired knowledge and skills in future system activities provided that this acquisition and incorporation is conducted by the system itself and leads to an improvement in its performance.”

The objective of autonomous system and learning is closely coupled. However, leaning is more particular in dealing with individuals. We develop an agent that is able to apply its knowledge based on previous identification towards improving the performance of the system. Depending on the degree of freedom in taking decision, learning can be classified into two principal categories:

- **Centralized learning:** This is a kind of learning where an agent is independent in processing the learning activities and decision. The system's overall performance solely depends on this agent's learning and decision-making capabilities.
- **Decentralized learning:** In this case, individual agents are designed to carry out their respective activities towards improving the system performance.

As we are building a single agent trying to improve the system performance by deciding which economic model to use when, our approach goes under the centralized learning. Again, there are several methods that an agent can use to learn from environment (Weiss, 2000). Table 7.1 arranges different learning processes according to their complexity.

Table 7.1: Different Learning Methods in Agent Technology

Learning Methods	Description	Amount of learning efforts required increasing ↓
1. Rote learning	Direct implementation of knowledge	
2. Learning from instruction and advice	Operationalization – decision with prior knowledge	
3. Learning from example and practice	Refinement of knowledge from experience	
4. Learning by analogy	Transformation of knowledge from a solved to a similar but unsolved problem	
5. Learning by discovery	New knowledge by observation, conducting experiments, testing theories	

Analyzing Table 7.1, we identify that our agent falls in the second category – learning from instruction and advice. We have conducted an extensive experimental analysis to obtain the knowledge about the domains of strengths of different economic models in Grid computing. Afterward, we advise our agent with this knowledge so that it can make decisions based on the knowledge. In addition, the formalization of the domains further enables the agent to make decisions for unknown scenarios, which can still be satisfied by those domains. The agent now needs to apply this knowledge in a dynamic Grid environment and to prove its effectiveness in terms of optimizing different performance metrics. Section 7.5 demonstrates a detailed experimental analysis to prove this effectiveness. The latter learning mechanisms in Table 7.1 are more generic and natural, which we would like to investigate in future.

To the knowledge here, we mean that the domains of strengths identified in Chapter 6. Based on this knowledge, the agent now needs to decide which economic model to consider

at a specific scenario in a dynamic environment. An example of the agent's knowledge representation is presented below.

The Agent's Knowledge Representation: In our framework, the agent uses the economic models as input variables for optimization in the environment. Therefore, the problem space defined by the agent is,

$$\text{Problem Space} = \{CMM, BAR, EA, CDA, CNP\}$$

Now, based on the optimization function as defined by the Grid provider, the agent manipulates the decision space. To do this, the agent uses our identification in Chapter 6. For example, if the optimization function is minimization of communication overhead (communication efficiency metric), the decision space is described as,

$$\text{Decision Space} = \{CMM, CDA, CNP\}$$

As these three economic models have been identified suitable for minimizing communication overhead in their respective domains compared to other economic models (Figure 6.3 in Chapter 6), these are considered as decision variables. However, at a specific time in the environment, the agent can choose only one decision variable. To facilitate the decision process, the agent, then, converts the decision space in terms of their respective domains. Using Table 6.1 in Chapter 6, we can write,

$$\begin{aligned} \text{Decision Space} = \{ & (d \geq 1 \text{ and } ((s+27.7)/(d-2.69)^2) \geq 0.65), \\ & (\{((s+27.7)/(d-2.69)^2) < 0.65 \text{ and } ((d-1.24)^2/(s+2.34)) \geq 0.32\}, \\ & (\{((d-1.24)^2/(s+2.34)) < 0.32 \text{ and } s \geq 1\}) \} \end{aligned}$$

Where, s and d refer to the supply and demand respectively at any state in the environment. Now for a specific time, either one of the domains in the decision space would be true. As the agent is notified about every occurrence of supply or demand change in the environment, it can determine the true domain respective to s and d of any given state. Thereafter, the agent chooses the economic model corresponding to that domain as an input to the environment. The nature of s and d in the decision space ensures the extensibility of the domains; i.e. can be any value greater than or equal to one.

The most crucial role played in the environment is the agent. The agent keeps track of supply and demand in the environment, and based on this, it makes the decision on which economic model to choose at a specific scenario. Algorithm-7 provides the pseudo code that describes the major roles performed by the agent.

Algorithm-7: Sense and Decide:**Input:** *numberOfUser, numberOfResource, Optimization-metric, Domains of Strengths (Table 6.1), Slope***Output:** *Recommended protocol*

//Keep track of supply and demand

If a broker is terminated, update the demand function by decreasing the *numberOfUser* by 1If a resource is committed, update the supply function by decreasing the *numberOfResource* by 1If(*Optimization-metric* = *metric-n*) // where *n* is one of the 11 metrics in Table 6.1

{

Domain-*i*: Assuming that there are *i* number of different domains dominated by different economic protocols for this particular optimizationcompute the *Slope* using the current *numberOfUser* and *numberOfResource* for Domain-*i*if the *Slope* sits in Domain-*i*, select the corresponding protocol of the domain*Recommended protocol* = Protocol corresponding to the Domain-*i*

}

Let the system know about the *Recommended protocol*

The last line of Algorithm-7 triggers the Algorithm-6 to be initiated and Algorithm-6 is then used to let the participants know about the new model. The mathematical models of the domains of strength of different economic models help the agent to facilitate the decision process in the following ways:

- ✓ As mentioned earlier, the switching decision must be conducted as soon as possible to maximize the utilization of the optimization process. Because of the formalization of the domains of strengths, the agent now requires to consume only a little computational power and can make the decision very quickly, which, otherwise, would require to import the respective datasets (the whole matrices) to make relative decisions.
- ✓ The second case is about the scalability of our model. As we have obtained the clear trends of the models' strengths, the formalization models further ensure the feasibility of accounting the strength for extended scenarios. We will see the proof for extended scenario in the following section.

The following section describes the simulative study conducted to evaluate the switching framework.

7.5 Performance Analysis for Switching Framework

This section presents the evaluation process of our framework and discusses the simulation space in support of the evaluation process. Our evaluation process is different from the existing literature (Table 2.1) to the sense that existing literature deploy a single economic model to investigate its efficacy in Grid resource management whereas, we switch from one economic model to another based on the scenario perceived by the environment. We evaluate switching framework in terms of optimizing five different performance metrics individually.

Same parameter configuration is used here as used to evaluate the performances of the individual economic models in Chapter 3 (Table 3.2) to ensure fairness in the evaluation. For our switching model and for a particular optimization case, we start with a constant number for the total number of resources and continuously inject different number of users using some predefined ranges into the environment (4th column of Table 7.2). A new set of users is injected when the previous set is finished with the simulation. This scenario continues until the total number of resources finishes in the environment. For statistical need, once again, we run all the simulations with five different samples and present only their averages.

Table 7.2 presents the number of users and resources used for different simulations aimed for different optimization scenarios.

Table 7.2: Resource Configuration for Switching Model

Simulation No:	Optimization Metric	Number of Resources	Number of Users
1	Revenue over communication cost	200	(1, 20)
2	Communication efficiency	200	(1, 32)
3	Social welfare	100	(10, 50)
4	Average turn-around time per Gridlet	100	(20, 50)
5	Total simulation time	100	(1, 50)

The ranges used for the number-of-users column have been set deliberately so that we can obtain possible scenarios to enable the execution of switching model. For example, for the first simulation, the range of (1, 20) gives higher probability for obtaining different regions those are dominated by different models (Figure 6.1 in Chapter 6). Considering other ranges (e.g. (30, 40)) might unable to evaluate the switching model as for such ranges we would receive scenarios for which switching could ever take place. The number-of-resources has been extended to 200 for the first two functions to measure the extensibility of our mathematical models identified in the previous chapter.

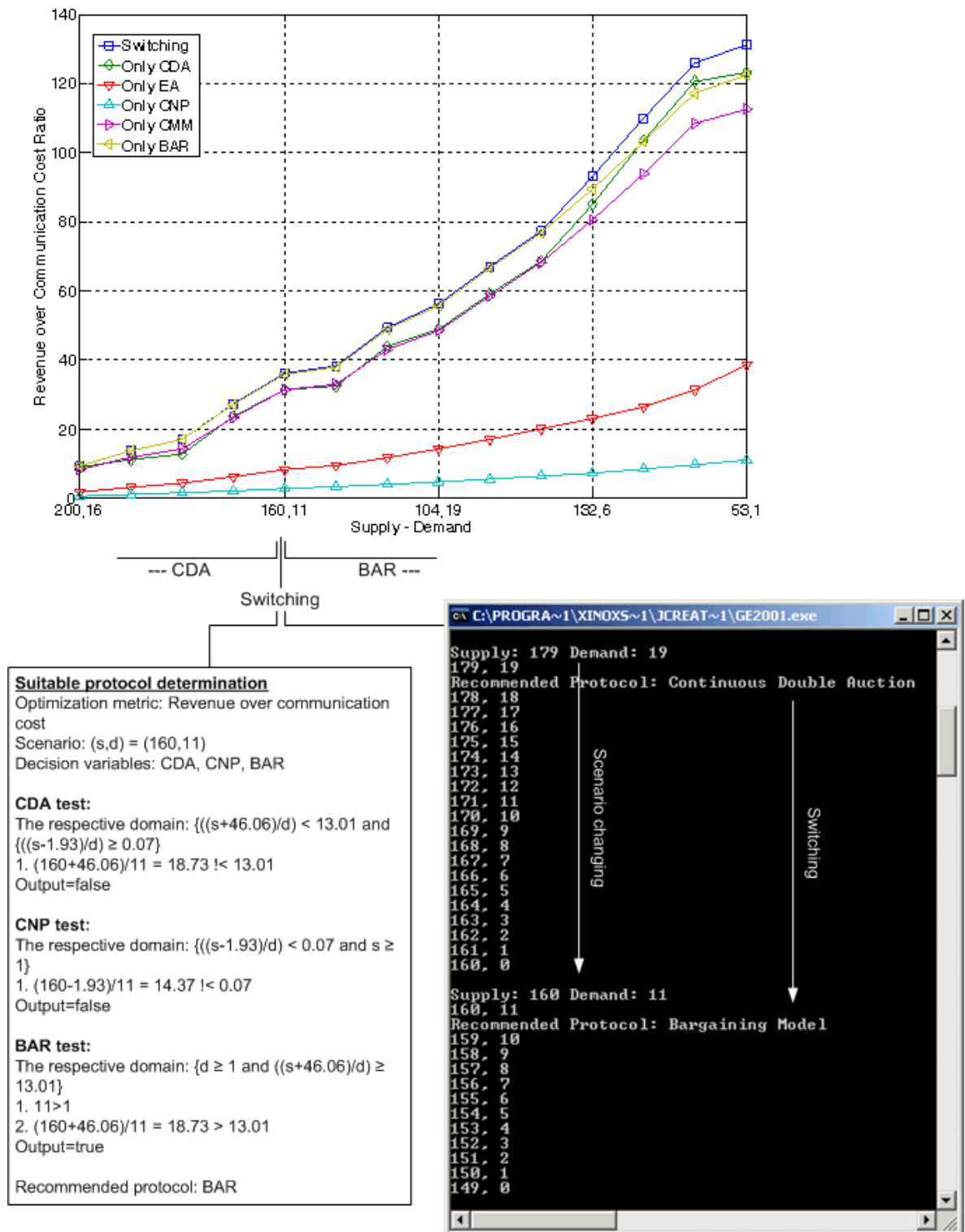


Figure 7.3: Scenario Illustrated for Revenue over Communication Overhead Optimization

7.5.1 Optimization for Revenue over Communication Overhead

In this case, the Grid would like to maximize its overall revenue while minimizing the communication overhead. Therefore, we let the agent know about the optimization function and the agent dynamically decides which model to use for what scenario based on the relative domains of strengths as identified in Chapter 6.

Figure 7.3 illustrates the results obtained during the optimization procedure for revenue over communication overhead. The upper plot compares the performances obtained by different models. The X-axis of the plot shows different supply-demand ratios at different times. The supply starts at 200 and keeps decreasing as occupied by the randomly injected users. For injecting users, we use the range (1, 20) in Table 7.2. The Y-axis represents the revenue over communication overhead ratio for different models. We normalize the both parameters, due to different units and, then, compute the ratios. For normalization, we follow the similar process as considered at Section 4.6 in Chapter 4. It can be observed that overall; the switching model performs better compared to any other individual models. The optimization could be further enhanced by utilizing all the three decision variables – CDA, CNP and BAR (Table 6.1). However, for this plot, our switching model only uses CDA and BAR, and switches between them based on their relative domains of strength. At each different scenario (along X-axis), the agent determines which model will be suitable to optimize the function as defined. For example, when the environment notices that the current supply is 160 units and demand is 11 units, the agent computes the suitable protocol for this particular supply-demand ratio. The down-left part of Figure 7.3 shows this computation process. The agent verifies all the decision variables in terms of their relative domains of strength using the current supply and demand parameters. After the domain test, the agent determines that BAR is the suitable protocol for the scenario. The downright part of Figure 7.3 shows the screen-shot of the system is switching from CDA to BAR due to changing scenario. Because of switching between the models, the system is able to utilize the relative strengths of the models better and ultimately to optimize the function. Thus, the switching model outperforms all the five economic models individually.

The EA and CNP show the lowest performance due to their highest communication overhead. Because of more supply than demand (Table 7.2), to provision a single user, CNP produces a huge amount of messages. The sizes of the groups of potential resources in this case, will be higher. This prevents CNP even to outperform EA (Figure 7.6). On the other hand, CDA, CMM and BAR show the quite competitive performance. As we have extended the total number of resources to 200 and still it is successful to make the defined optimization, this guarantees the scalability of the formalization models.

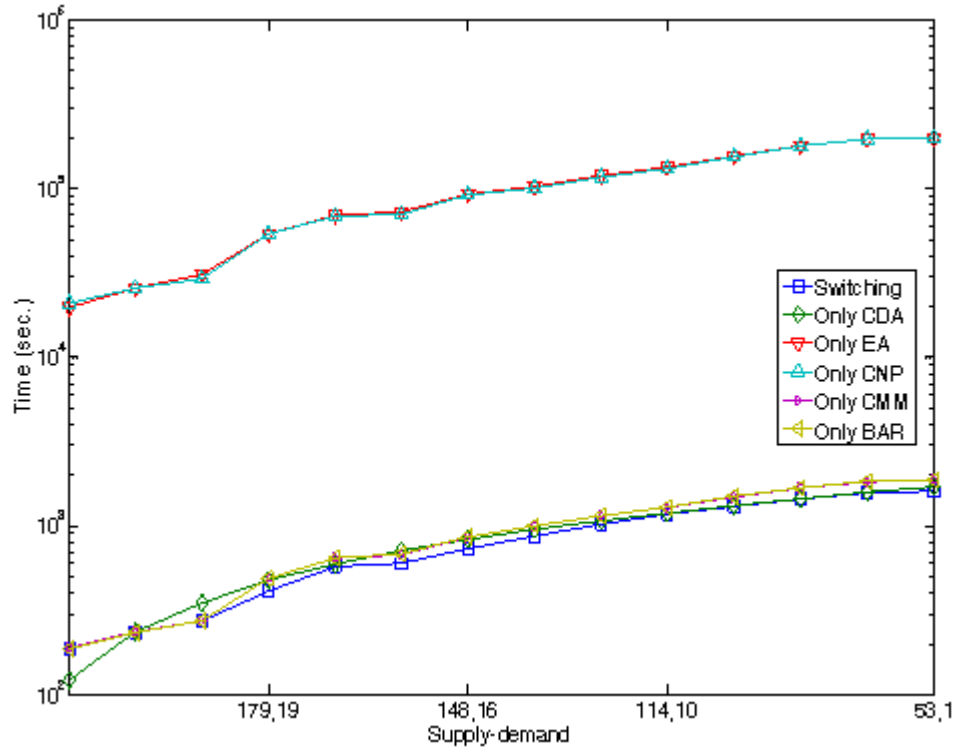


Figure 7.4: Average Turn-around Time per Gridlet while Optimizing Revenue over Communication Overhead

While we are optimizing the revenue over communication overhead function using switching model, we, at the same time also explore the conditions of other performance metrics. For example, Figure 7.4 shows the results for average turn-around time per Gridlet by the different models while optimizing revenue over communication overhead. The Y-axis is represented in logarithmic scale, due to the high magnitude. In this case, the performance of the switching model is quite competitive with that of the CDA. Because of switching, the systems uses both CDA and BAR. As it uses BAR, the overall negotiation process is extended. However, for a set of scenarios (from the beginning until (114, 10)), the switching model outperforms the CDA, which helps to obtain a competitive performance at the end. The EA and CNP take the highest time and pose a competitive performance.

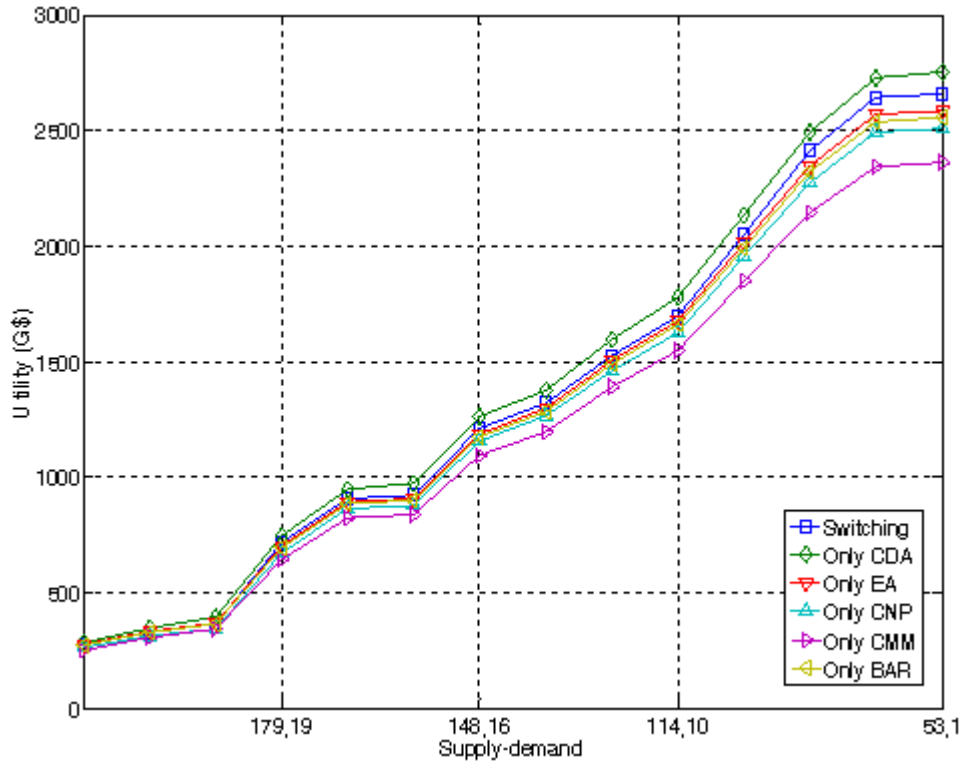


Figure 7.5: Social Welfare while Optimizing Revenue over Communication Overhead

Figure 7.5 illustrates the social welfare during the period of revenue over communication Overhead optimization. We can observe that the switching model cannot outperform all the models in this case. Overall, the CDA shows the best performance. The overall welfare for the model is lowered, due to the integration of BAR in the switching approach. The EA performs better compared to CNP, CMM and BAR. Because of additional resources, the average number of competitors per auction decreases; therefore, resources are provisioned with lower costs. This also gives chance users to optimize their budgets, which ultimately helps in maximizing the welfare for EA. Again, due to the higher supply than the demand (Table 7.2), the spot prices in CMM are always low. This prevents the resources from maximizing their revenue. In addition, as the welfare in CMM only comes from the user utility, CMM shows the lowest performance.

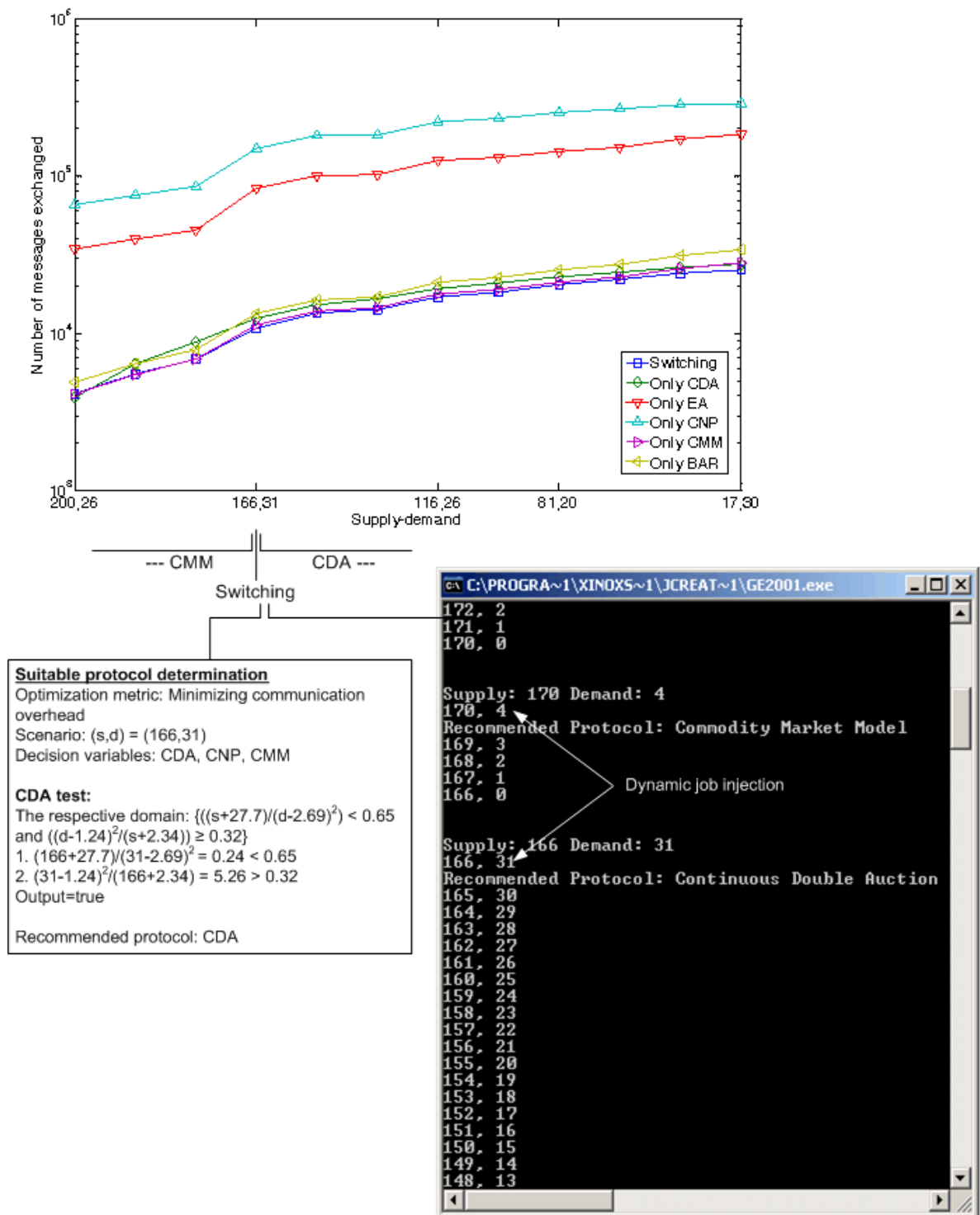


Figure 7.6: Scenario Illustrated for Optimizing Communication Efficiency

7.5.2 Optimization for Communication Efficiency

In this particular case, the Grid would like to minimize its overall communication overhead. Figure 7.6 depicts the results obtained from different economic models. The upper part of the figure does the performance comparison among different economic models. Overall, the switching model produces minimum communication overhead. Once again, the optimization

could be enhanced by utilizing all the decision variables. In this case, we are testing using only two variables, whereas, there are three decision variables for this metric – CDA, CMM and CNP (Table 6.1 in Chapter 6). Because of switching between the models according to their respective domains of strengths, switching model outperforms all the others. The CMM shows second best performance in this case. The spot prices in CMM are low, due to additional supply (Table 7.2), which causes most of users to be accepted very quickly. Thus, it prevents the model from exchanging a huge amount of messages. The CDA shows competitive performance with CMM at the end whereas; due to multiple rounds in BAR, it shows the third best performance. As mentioned earlier, because of the more supply, the sizes of the groups of potential resources in CNP are larger; therefore, to provision a single user, it requires exchanging a huge amount of messages. On the other hand, even EA uses multiple rounds, as the total number of users in the environment is low and the auction groups are only formed with interested users (Section 5.3 in Chapter 5), it results in exchanging a lower number of messages for a single auction. This is why; EA performs better than the CNP in this case.

The down-left part of Figure 7.6 shows the suitable protocol determination process by the agent when the scenario is defined as (166, 31). The decision variables, in this case, are CDA, CMM and CNP. The agent determines that the CDA to be the suitable for that particular scenario. The screen-shot records the behavioral details of the environment.

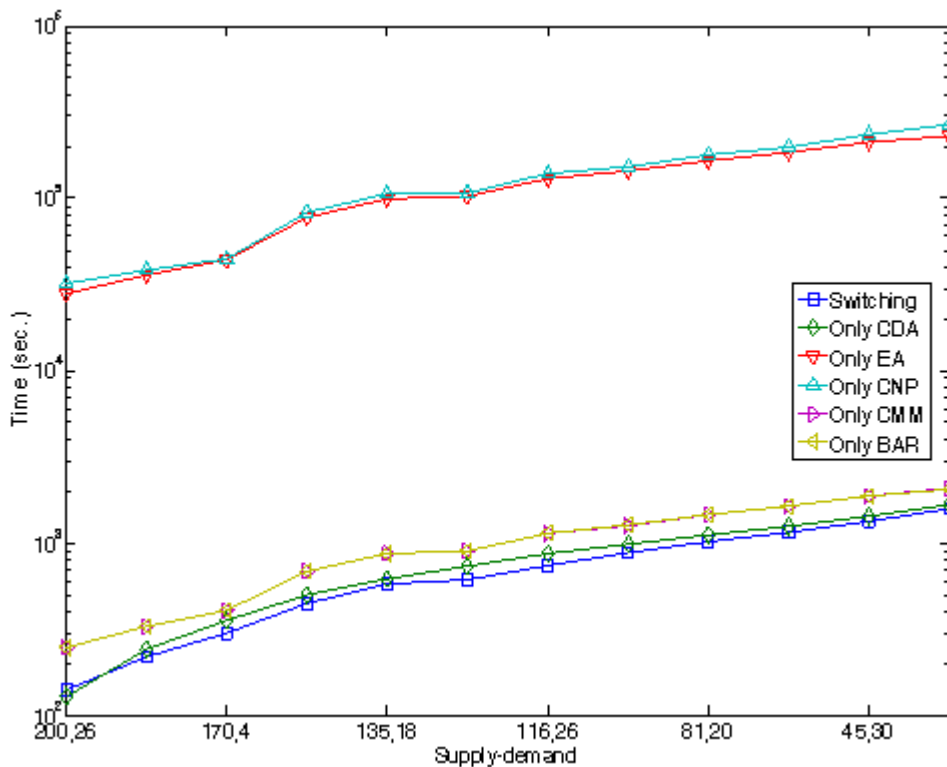


Figure 7.7: Average Turn-around Time per Gridlet while Minimizing Communication Overhead

Figure 7.7 presents the comparison for the average turn-around time per Gridlet during the minimization of communication overhead. It can be observed that in this case also switching model outperforms all the others. The models (CMM and CDA) used for minimizing communication overhead have been matched with the models used for minimizing average turn-around time (Table 6.1). This is the reason why the switching model still shows the best performance here. The CMM and the BAR show almost equal performance. Users do not have to explore the market longer, due to optimal resource allocation in BAR (Section 4.4 in Chapter 4), which helps the model to minimize the average time per user. On the other hand, in CMM, due to low prices for the resources, users do not require to explore the market longer. Thus, it minimizes the time for CMM also. The EA and CNP show the competitive performance in this case.

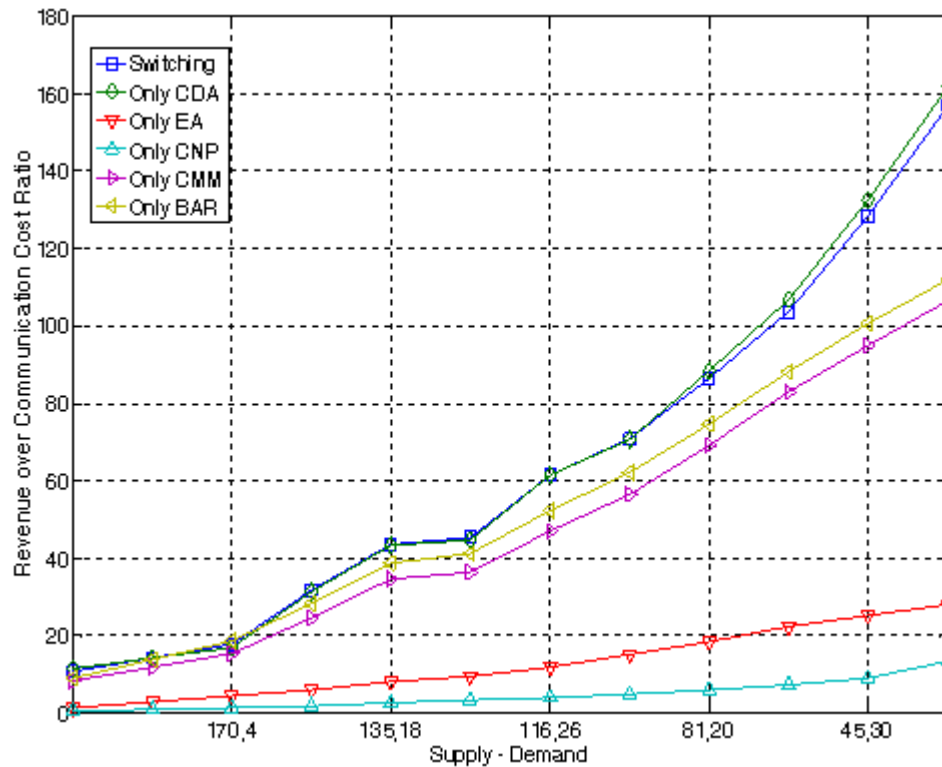


Figure 7.8: Revenue over Communication Overhead while Minimizing Communication Overhead

Figure 7.8 illustrates the performance analysis for revenue over communication overhead while minimizing communication overhead. For certain scenarios, switching and CDA shows competitive performance. However, ultimately the performance for switching is degraded. BAR generates more revenue, due to optimal allocation of resources, compared to CMM (Figure 4.22 in Chapter 4). Again, the users can better optimize their preference values in CNP, due to enough resources, which prevent the resources from maximizing their revenue. Thus, CNP, in this case, shows lower performance compared to the EA.

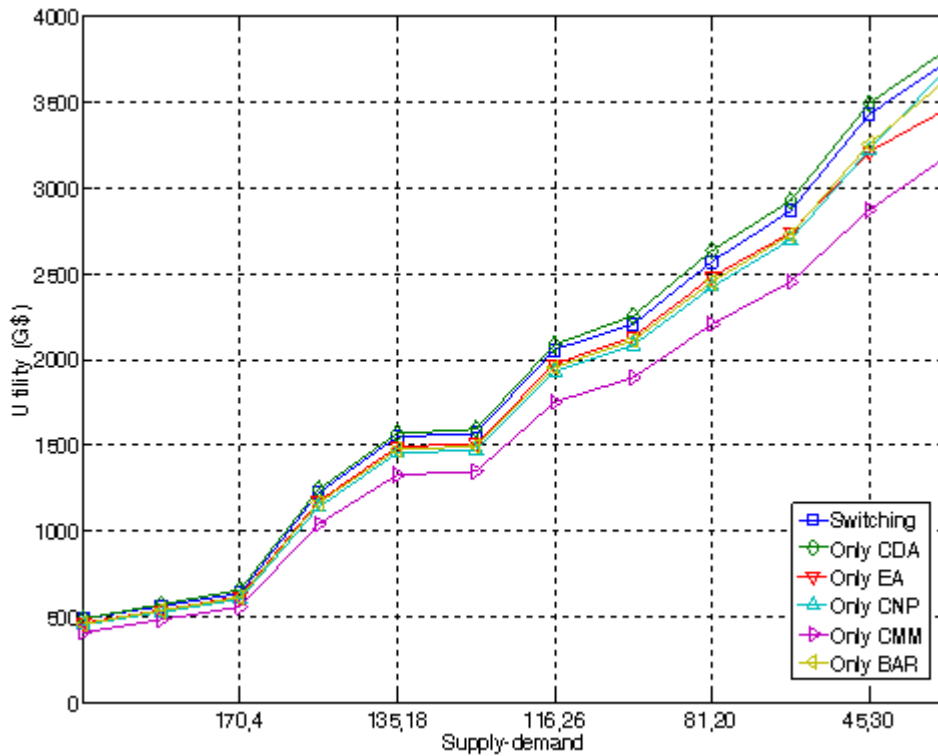


Figure 7.9: Social Welfare while Minimizing Communication Overhead

The comparison of social welfare in the period of minimizing communication overhead is presented in Figure 7.9. The CDA is the model that outperforms all the other models in this case. The EA, CNP and the BAR show competitive performance. The CMM shows the lowest performance. A similar explanation can be applied to explain the performance of CMM here too as used to explain in Figure 7.5.

7.5.3 Optimization for Average Turn-around Time per Gridlet

If the Grid would like to minimize the average turn-around time per Gridlet, this optimization procedure is processed. Figure 7.10 shows the results obtained for the metric in terms of different economic models. Overall, the switching model performs better than others do. There are two decision variables for this function – CDA and CMM (Table 6.1 in Chapter 6). As can be observed from Figure 7.10, for a certain scenarios (until (34, 22)), switching model shows equal performance with CDA. This indicates that until (34, 20), switching model uses CDA. At (34, 20), the model switches to CMM, because CMM has been determined as the suitable protocol for this scenario. Staying longer in the environment and keep switching between the models could enhance the optimization further. The BAR shows slightly better performance than the CMM. As mentioned earlier, due to the Pareto-optimality in resource allocation, BAR takes lower time. Their performances are much low, due to different working

principles of EA and CNP. Similar protocol determination process and screen-shot can be shown as shown during the previous optimization processes.

Figure 7.11 presents the revenue over communication overhead comparison during the optimization for average turn-around time per Gridlet. The CDA performs better than others do in this case. The moment switching model switches from CDA to CMM at (34, 22) (Figure 7.10) to minimize the time, the revenue gets down. As the total number of resources, for this metric, is lower compared to earlier (Table 7.2), BAR and CMM are not better able to utilize their strengths. Therefore, they show very lower performance compared to the switching and CDA. The EA and CNP show the overall equal performance. However, in the middle, EA performs better than the CNP.

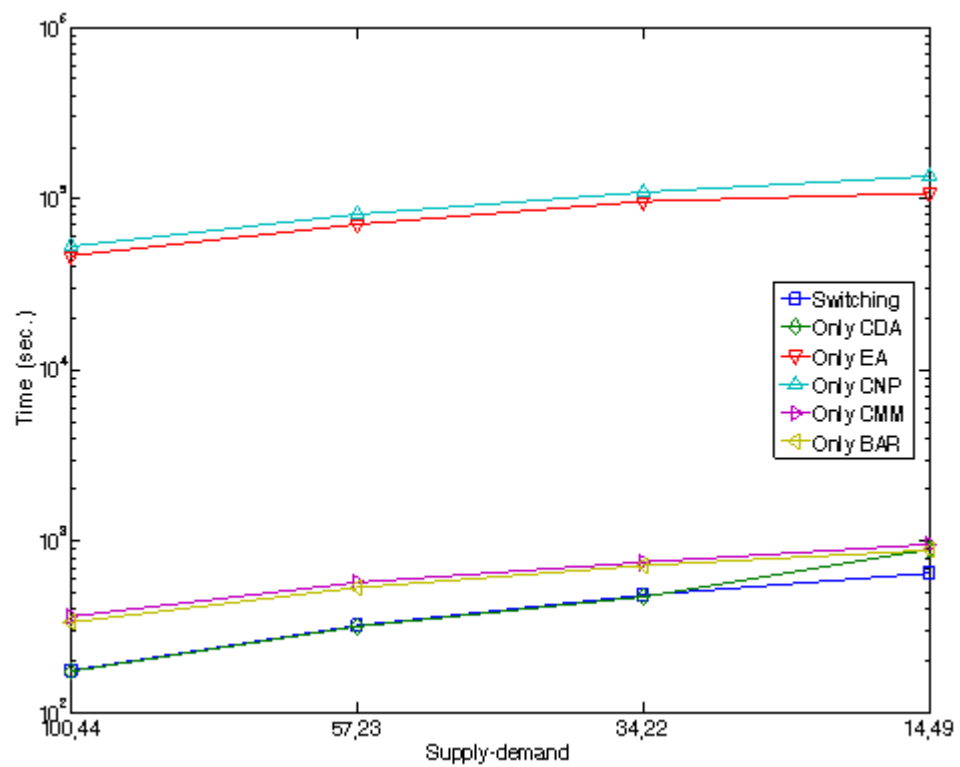


Figure 7.10: Scenario Illustrated for Optimizing Average Turn-around Time per Gridlet

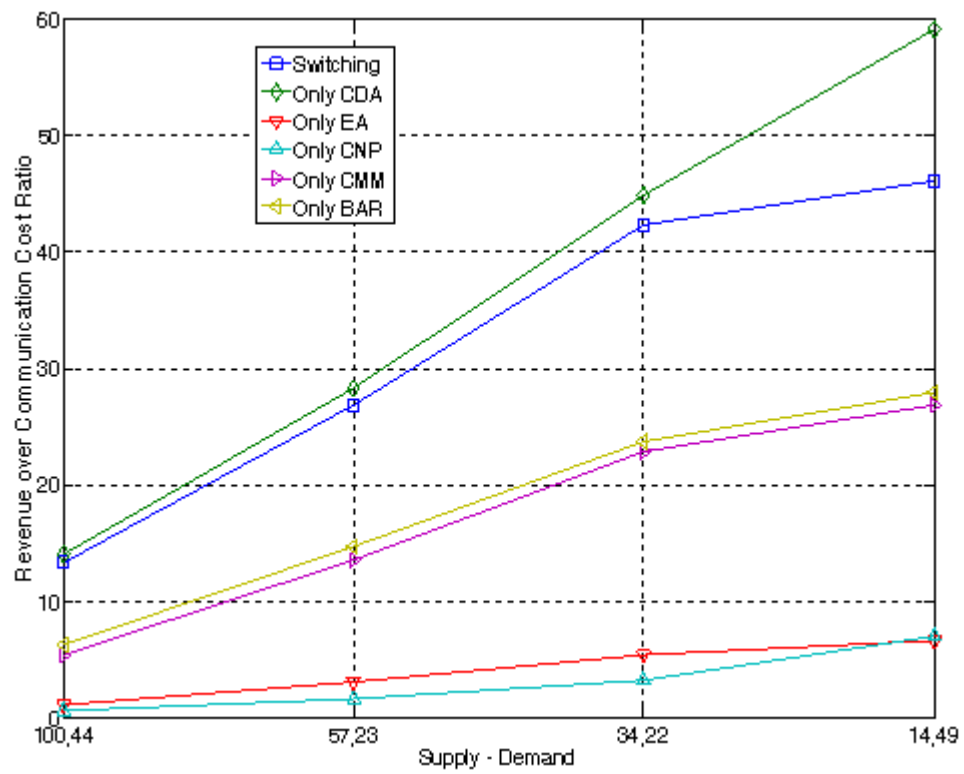


Figure 7.11: Revenue over Communication Overhead while Minimizing Average Turn-around Time per Gridlet

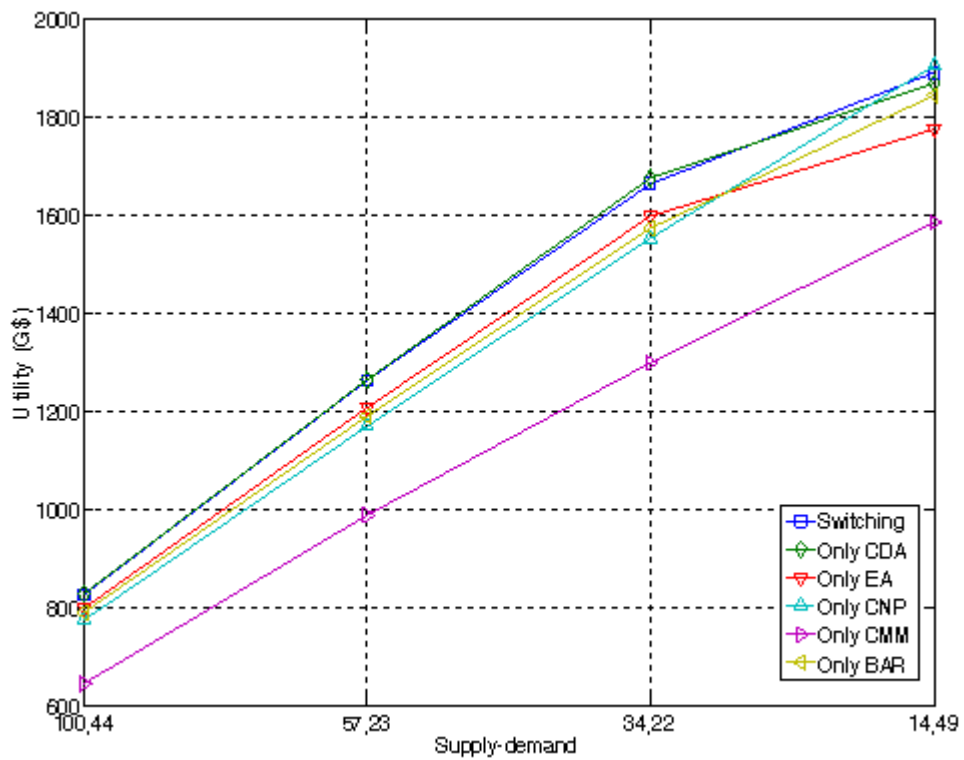


Figure 7.12: Social Welfare while Minimizing Average Turn-around Time per Gridlet

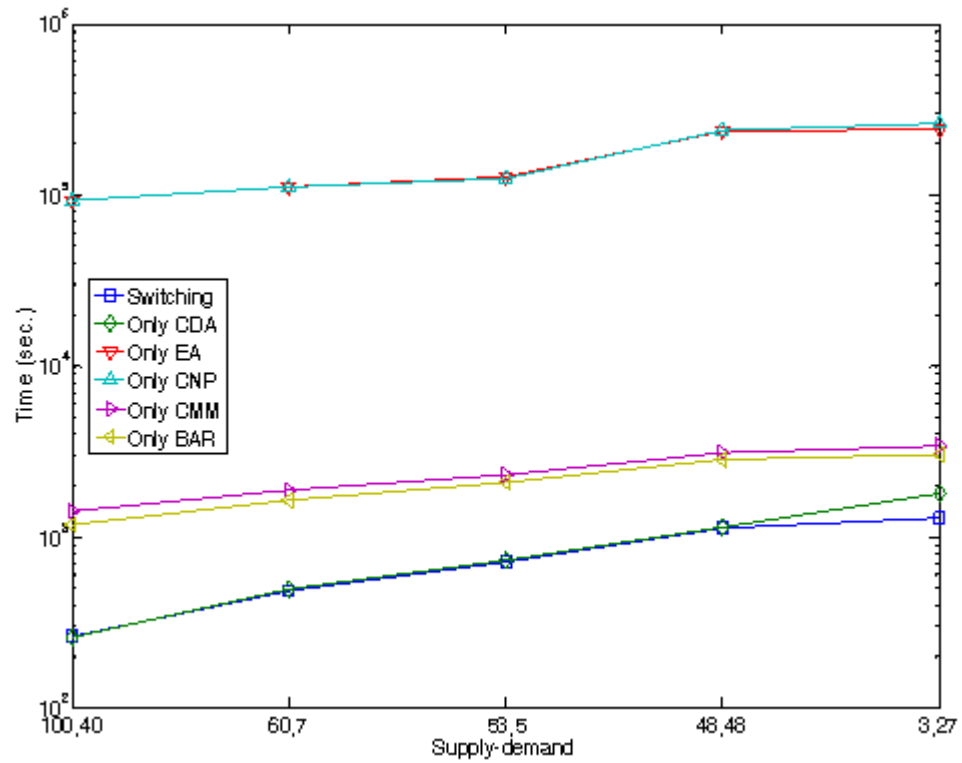


Figure 7.13: Scenario Illustrated for Optimizing Total Simulation Time

The comparison for social welfare during the optimization of average turn-around time per Gridlet is illustrated in Figure 7.12. The switching and CDA is quite competitive in this case. However, switching performs better at the end. Because of using BAR at the end, switching is able to satisfy both users and resources. This helps to maximize their utilities, which ultimately brings higher welfare in the market. The CNP shows the best performance ultimately. When the number of resources are decreased at the end, a set of users is unable to be accepted, which lowers the aggregated welfare in general for this region. As mentioned earlier, for CMM, the welfare comes only from user utility. Therefore, CMM shows the lowest performance across the scenarios.

Figure 7.13 shows the results obtained for optimizing total simulation time. Once again, overall the switching model outperforms all the other models. Figure 7.13 follows a similar trend as obtained for Figure 7.10. Therefore, a similar explanation can be used to explain Figure 7.13.

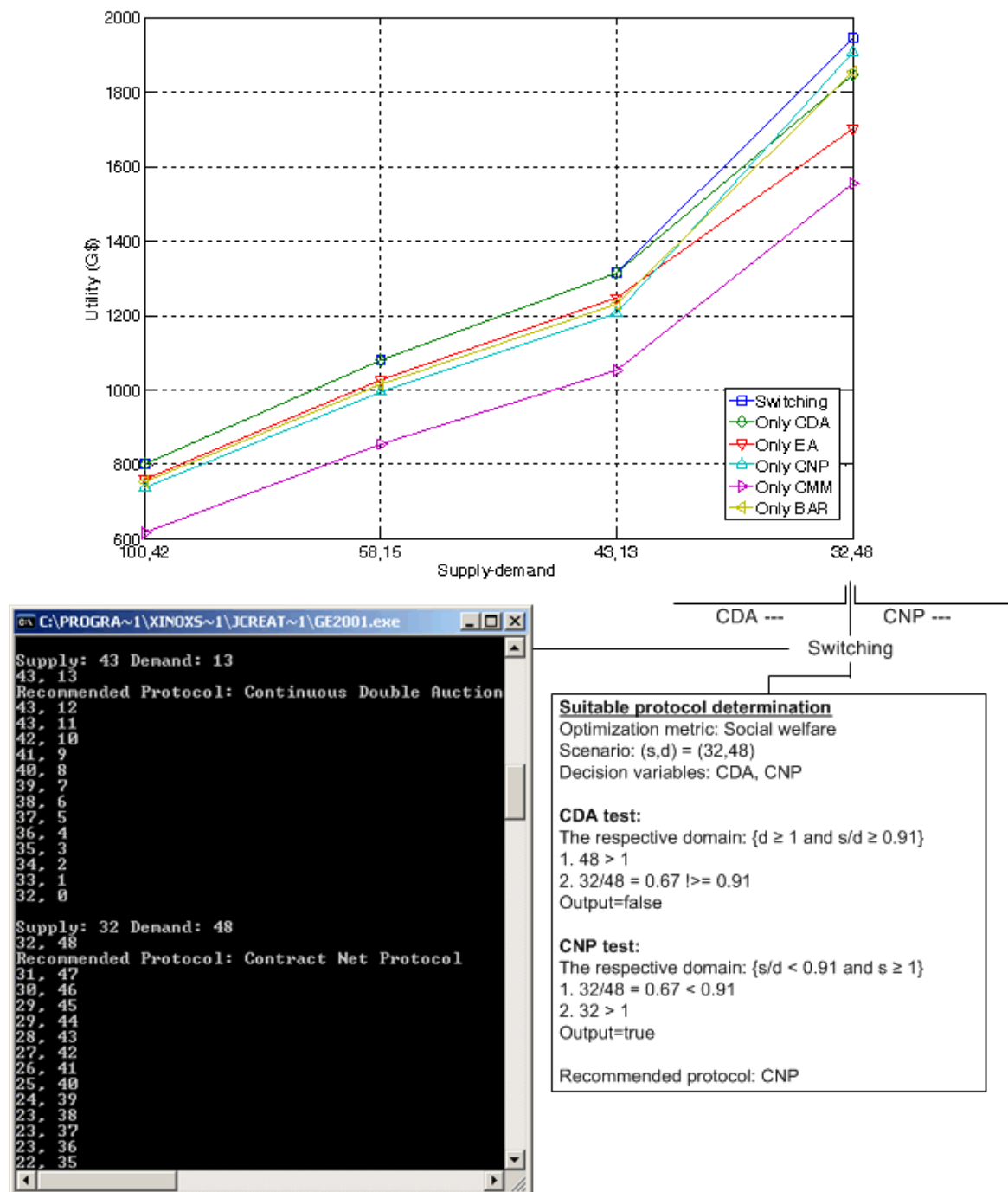


Figure 7.14: Scenario Illustrated for Social Welfare Optimization

7.5.4 Optimization for Social Welfare

In this case, the Grid would like to increase the utilities for both users and resources in the environment. Figure 7.14 demonstrates the overall performance achieved by the different models for social welfare. The switching model, overall, outperforms all the other models. Until the model switches to CNP at (43, 13), its performance coincides with that of the CDA. The moment it switches to CNP at (43, 13), the performance starts rising up; because CNP is the suitable protocol for this scenario as determined by the agent. CNP alone, in this region,

shows improved performance, due to the same reason. The BAR overcomes EA at (43, 13). The users in BAR can quickly obtain resources without much competition, due to the higher supply than the demand. This helps users to maximize their utilities, thus contributing to maximize the welfare. The CMM shows the lowest performance as usual. The downright part of Figure 7.14 presents the suitable protocol determination process. The current scenario is recorded at (32, 48) and the determination process is meant to this scenario. There are two decision variables for this case – CDA and CNP. The first test verifies that this scenario is not under the domain of CDA. The second test verifies that the scenario belongs to the domain of CNP. Therefore, CNP is decided as the suitable protocol to this scenario. The down-left part (the screen-shot) of Figure 7.14 shows the compatibility with the agent’s determination.

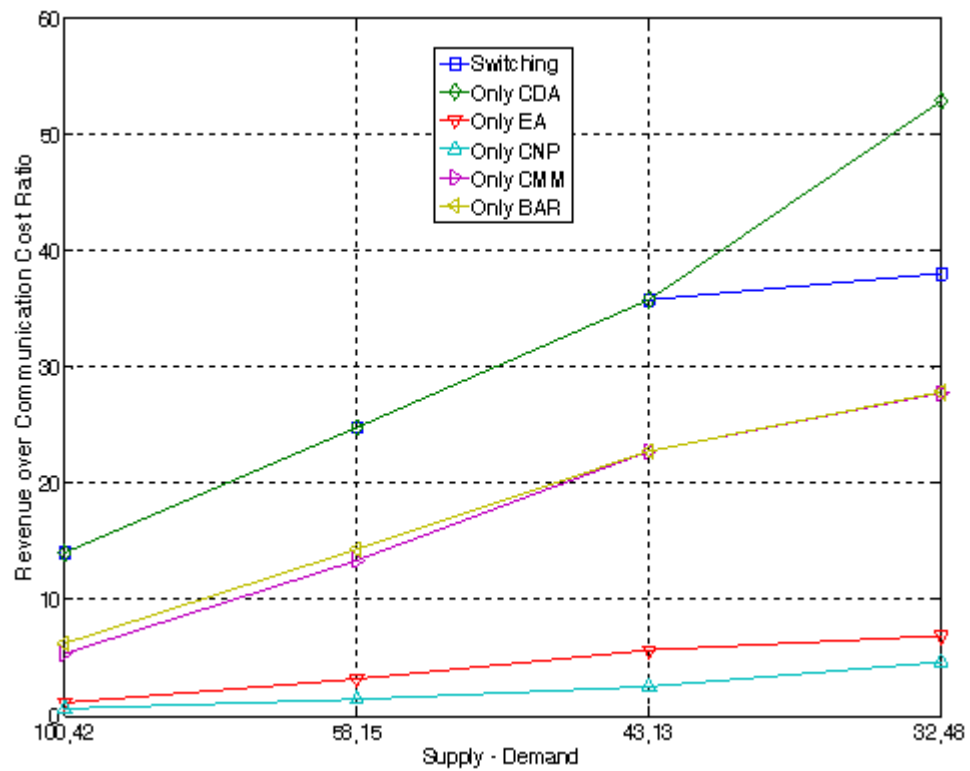


Figure 7.15: Revenue over Communication Overhead while Optimizing Social Welfare

Figure 7.15 depicts the results for revenue over communication overhead while optimizing social welfare. Overall, the switching model is not performing better here; rather CDA performs better than all the others do. The revenue has become lower for the switching model, due to switching from CDA to CNP. The CMM and the BAR show competitive performance in this case. Because of exchanging a huge amount of messages, EA and CNP show the lowest performance as usual.

The outcomes for average turn-around time per Gridlet while optimizing social welfare is demonstrated in Figure 7.16. The CDA once again, outperforms all the others for this criterion. The switching model consumes a greater amount of time, due to switching to CNP

at (43, 13), which causes the performance to dramatically worse at the end. Here also, the CMM and the BAR show competitive performance. The EA and CNP show the lowest performance as usual.

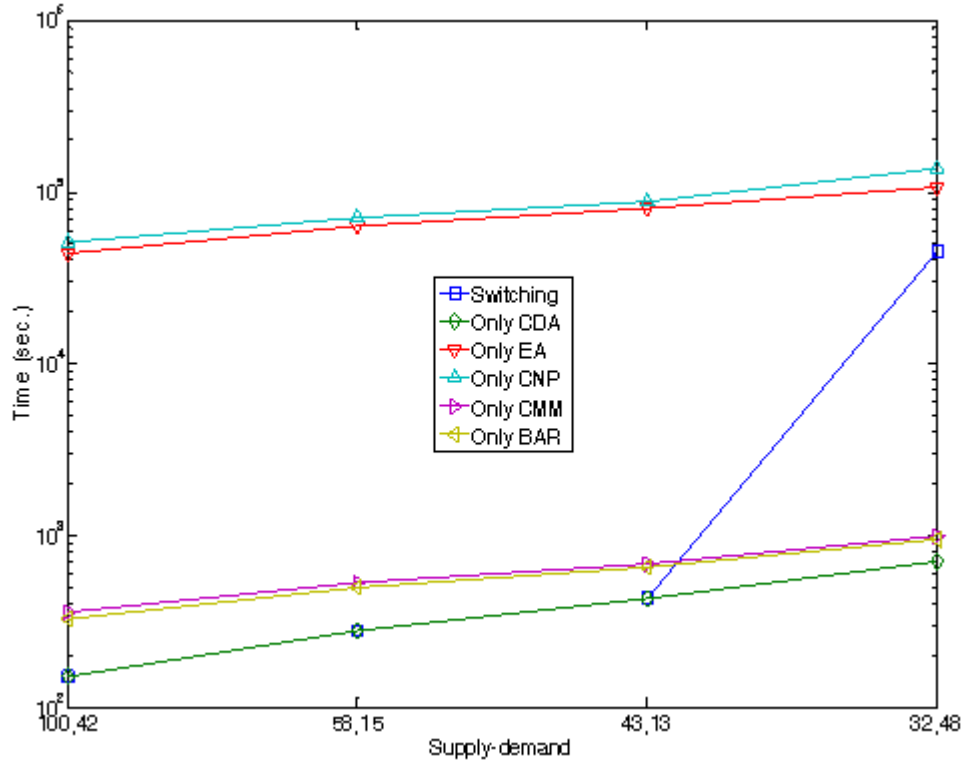


Figure 7.16: Average Turn-around Time per Gridlet while Optimizing Social Welfare

From the above discussion, we have seen that the switching model always outperforms all the other individual models in terms of optimization function. However, during switching, performance functions other than the optimization function have not been optimized. We observed the suitability of Continuous Double Auction for most of the functions other than the optimization function. The sorting and price formation methodology mainly resulted in achieving better performance for the model. Thus far, we have only focused on a single criterion optimization except the case of revenue and communication overhead. However, optimization for multiple criteria is also possible. In a multi-criteria optimization scenario, the Grid might want to optimize more than one performance function simultaneously. We would like to explore this multi-criteria optimization problem in the future.

7.6 Switching Mechanism in Cloud Computing: Challenges and Opportunities

Based on the promising outcomes of the switching model as discussed above, we are inspired to identify the possibilities of deploying the model in recently emerged Cloud computing

environment. As mentioned before, the Cloud can be distinguished from traditional computing paradigms because of its additional focus on scalability, reliability, and virtualization of the resources. One of the crucial parameters that drives the computing resource providers to move towards the Cloud is, obtaining economic benefit through statistical multiplexing of the resources. Unfortunately, the pricing strategies used in existing commercial Cloud computing providers are still in their infancy. Most providers in the Cloud impose a fixed pricing mechanism over resources (instances), that is, the price for a particular resource does not change over time. We study two commercial Cloud resource providers – Amazon EC2 (Ostermann et al., 2010) and Manjrasoft Aneka (Vecchiola et al., 2009) to identify the challenges and opportunities involved in deploying our switching mechanism in the Cloud.

- **Challenge – (i):** *Rapid Escalation of the Cloud.* The demand and supply for resources can scale up in order within a moment in the Cloud. The switching decision must be carried out as quickly as possible to adapt with the change

Opportunity – (i): A comprehensive monitoring service using prediction functions (probability distribution function) can be used in this case. The supply and demand can be known well in advance to facilitate the decision process

- **Challenge – (ii):** *Data Transfer Bottleneck.* Data transfer across virtualized layers or multiple data centers in the Cloud may affect the performance of the transfer. This relates our model with the dissemination of the data regarding the switching decision as the decision must be transferred in the Cloud without much delay

Opportunity – (ii): The Cloud needs to be re-configured with high Bandwidth switches to maximize the potential of the switching mechanism

- **Challenge – (iii):** *Different Application-level Languages.* User applications in the Cloud may be developed using different programming languages (e.g., Java, Python, .NET), whichever is suitable depending on the workflow and execution models of the applications. On the other hand, according to the switching mechanism, providers may come up with different pricing strategies with different times where different providers may define their strategies with different languages. Therefore, the scenario is lacking a common negotiation language between users and providers

Opportunity – (iii): A programming-level virtualization technique might be helpful in this case. The incorporation of a virtual layer consisting of various programming language compilers into the Cloud layered architecture (Vecchiola et al., 2009) would help communicating between application and provider level components

- **Challenge – (iv): *Supply Function for Public and Private Clouds.*** Depending on the resource usage policy, there are two different Clouds – Private and Public. In a private Cloud, the Cloud infrastructure is restricted to a specific set of users whereas, in public Clouds, anyone can access the resources on demand. According to the promise of Cloud computing, in public Clouds, there would always be unused resource by dint of the virtualization technique (Ostermann et al., 2010). Therefore, unlike Grid, the resource load in Cloud would not vary with time making it difficult to define the supply function in such a scenario. As the supply is one of the crucial parameters that drives our switching mechanism, a proper definition of the supply is essential. The definition must also be consistent with the function defined for private Clouds

Opportunity – (iv): A general supply function applicable for both private and public Clouds needs to be defined. Even if it is possible to define as many virtual instances as possible over physical instances, the performances of the virtual machines might not be guaranteed as the number scales up. Therefore, the supply function defined for public Clouds must consider the Pareto optimality so that creating a new virtual instance does not affect the performances of existing instances

- **Challenge – (v): *Policy Extensibility Issue.*** It is possible to acquire Cloud resources across clusters or datacenters distributed in different countries or organizations. Therefore, different providers could come up with different resource usage policies and may not agree upon the switching decision. The potential of the switching mechanism would be hampered if the case is ignored

Opportunity – (v): To deal with this issue, we may convince the resource providers denying the switching decision by providing them an estimated profit margin within a particular period if they accept to adopt the switching mechanism. Still, if a set of providers want to restrict with their own policies, the supply-demand monitoring system (such as CloudWatch in Amazon EC2 <<http://aws.amazon.com/cloudwatch/>>) should adapt accordingly. The monitoring system, in this case, must avoid the demand deployed on and supply available by the providers denying the switching decision

7.7 Conclusions

The demand of Grid computing has been broadening day by day. Economic models are found compelling because they can cope with the dynamics of the Grid. We have identified that

different economic models are suitable for different scenarios where the scenarios are characterized in terms of supply and demand. Based on our identification about the strengths of different economic models, in this chapter, we proposed an optimization framework in a dynamic Grid environment. We described the framework in terms of switching from one economic model to another based on the models' domains of strengths. We discussed the extended broker, resource and system's behavior in terms of an autonomous and learning environment. We described the agent and its role in the environment through automatically deciding which economic model to use at what scenario. Through an experimental analysis, we showed that our multi-model framework has successfully been able to optimize predefined objective functions better than any other individual models in a dynamic Grid computing environment. A study on existing commercial Cloud resource provider further enabled us to understand the challenges and opportunities of deploying the switching model in the Cloud.

Conclusions and Future Directions

8.1 Summary

Grid computing is a promising platform for distributed resource collaboration. The major motivation for this platform is driven by solving computationally complex problems such as drug design. However, the aggregation of the Grid resources becomes an issue due to the standalone characteristic and distributed ownership of the resources. The effectiveness of economic-based approaches in this aggregation is well known in the subject. An extensive research has been conducted to investigate the suitability of different economic models for distributed resource collaboration in the Grid. However, existing research is still lacking selecting a suitable model for the Grid. Through a comprehensive survey on different economic models, we have identified the reason of this ambiguity to choose a particular model. The reason is the suitability of different models in different scenarios. Thus, we realized the significance of performing a quantitative analysis on the performances of widely proposed economic models and of identifying the domains of strengths of individual models. This identification eventually inspired us to develop an adaptive economic-based Grid resource management architecture where different economic models can be used to deal with different scenarios.

Therefore, towards delivering a sustainable and an adaptive economic-based Grid computing platform, in this thesis, we have:

- identified a range of key performance metrics suitable for evaluating the performances of any economic-based resource management system in the Grid,
- developed an evaluation framework suitable for investigating and analyzing the scheduling strategies of widely proposed economic models for a wide range of scenarios in the Grid,
- extended widely adopted GridSim simulation toolkit by developing Commodity Market, Bargaining and Contract Net Protocol economic models in the distributions,

- compared the performances of the economic models and identified the regions where a one economic model shows its strength over the other economic models,
- formalized the domains of strengths of individual economic models in terms of various performance metrics,
- designed and developed an optimization framework that optimizes through dynamically switching from one economic model to another economic model depending on the models' domains of strengths, and
- designed and developed a switching agent that keeps track of supply and demand function of a Grid network and automatically provide decision on which economic model to use when depending on the function the network administrator would like to optimize.

8.2 Research Limitations

Along with the aforementioned contributions, we would like to address the following limitations in our research, which are still needed to consider for enabling a more reliable and robust framework.

- Currently, the definition of scenarios in our work was limited only into resource supply and demand. A more comprehensive definition by considering other QoS parameters such as trust and risk for a particular market should be added
- In our work, we took provider side and considered naïve users. However, in reality, users could also come up with their individual negotiation strategies. The incorporation of intelligent users would make difference in the performances of the economic models. We need to consider this issue in our work
- At present, we assumed that all users are happy even our mechanism switches to one economic model from another. In reality, a set of users might not be happy with the switched model and, therefore, would like to leave the market. Hence, a careful attention need to be taken while formatting SLAs between users and providers over the switching mechanism
- The simulation space was defined in low scale (100 by 100). The model should be evaluated on a high scale to conform with the large nature of the Grid
- The evaluation of our switching mechanism was based on only the previous identifications (Domains of Strengths). The extensibility of our mechanism by considering the impact of frequent switching or time-driven switching on the provider profit should be justified

8.3 Conclusions

Grid computing possesses diversity in terms of users, applications, and resource management strategies. To deliver a sustainable computing platform for the Grid, resource management systems need to be re-advisable and incentive-oriented. Economic-based resource management systems can be used in this respect, because they provide necessary components such as standards for using resources and motivation for sharing resources towards enabling a large-scale virtual computing platform. Therefore, studying the suitability of different economic models in distributed resource management environment is crucial. Our survey on existing economic models in the Grid provided key components for realizing the significance of different models in different aspects of Grid computing.

To obtain and contrast a clearer picture about the performance of different economic models in Grid resource management, an evaluation framework has been developed. The framework enabled the delivery of a consistent evaluation method for different models. The methodology of our framework is built upon the evaluation of different models using a common simulation and parameter model. The method, therefore, facilitated the evaluation process of various economic models by the researchers in the domain. The characteristics of the simulation space and parameter configuration of the framework further emphasized the significance of such a framework in Grid resource management paradigm. We defined the simulation space by considering an intuitive set of supply-demand ratio. This characteristic enabled to carry out a quantitative analysis on the performances of the models and to justify sustainability of the models in the domain. The framework showed its effectiveness through the conduction of successful simulations and analysis and contrast among the performances of widely proposed economic models in the Grid. Therefore, the research issue rose from the requirements of a common evaluation platform has been resolved by the development of such a framework.

The framework enabled understanding the impact of commodity market models (Commodity Market and Bargaining) on Grid resource management. The dynamic manipulation of equilibrium price based on resource supply and demand function in Commodity Market Model showed its efficacy towards maintaining equilibrium in the market. The variation in performances over the simulation space of the model ensured that the equilibrium pricing strategy has successfully interpreted the supply and demand function in the environment. The negotiation strategy developed in Bargaining Model demonstrated its effectiveness in utility-based resource collaboration. As the participants, in this model, can negotiate over their preferred terms to construct SLAs, an optimal resource allocation became feasible. This optimality helped increase the job success probability and revenue for providers

in the market. The evaluation of these two models over a comprehensive set of performance metrics helped realizing the robustness of individual models in distributed resource management. A comparative analysis among the performances of the models enabled understanding the strengths and weaknesses of individual models in terms of both micro and macro-economic principles. The analysis further helped identify the domains where one model outperformed the other. Clear trends of these domains strengthened our identification. Our investigation on the commodity models, thus, proved the compatibility with our research methodology that utilizing the potential of different models in different scenarios could bring reasonable benefit in Grid resource management. The identification of these domains of strengths of individual models, therefore, provided a remarkable impact on Grid resource providers in organizing their business strategies.

The evaluation framework has been further extended by incorporating the scheduling protocols of auction models – English Auction, Continuous Double Auction and Contract Net Protocol. Auctions are popular in distributed resource collaboration due to their decentralized and Pareto-optimal resource allocation. The development of English Auction Model helped understand the importance of conducting multiple auctions concurrently by the Auctioneer and the competition strategies by the brokers. In multiple auctions, users are given opportunity to choose their suitable resources and all auctions are carried out independently. The model showed its effectiveness in maximizing revenue for providers. However, the model has significant drawback in terms of communication efficiency. Therefore, networks with strong bandwidth may choose this model to maximize their profit. Our observation on Continuous Double Auction revealed its suitability for both user and resource communities. As a result, the model brought significant social welfare in the market. Networks with limited bandwidth can choose this model, as it further showed its significance in quick resource allocation. The meta-scheduling process in Contract Net Protocol tended to favor users as user, in this case, can select best resources based on the availability in the market.

A comparative analysis among the performances of these auction models provided a clear understanding about the suitability of different models in different scenarios in the Grid. This variation in findings proved the compatibility with the performances of individual models. We investigated that in terms of both communication efficiency and revenue, Continuous Double Auction dominated the others. Contract Net Protocol has significance over social welfare, in which, a major part is contributed by user utility. Overall, the comparative approach helped to deliver a better realization about the performances of different auction models in Grid computing. Therefore, our identification provided reasonable contribution to the resource markets particularly interested in auction protocols.

An extensive comparison considering the performances of both commodity and auction models helped deliver an overall understanding about the suitability of the models in different scenarios in Grid resource management. The domains of strengths of individual models identified from the comparison leveraged a crucial contribution to commodity and auction communities. Our identification paved the way of a long-term demand of a summarized and neutral evaluation of various economic models in Grid resource management. The suitability of different models in different scenarios proved the consistency with existing literature – a single model is not enough to cope with the diverseness of the Grid. Our observation on the domains of strength of different models showed the opportunity of developing an optimization mechanism through utilizing of the potentials of different models in different scenarios.

To deal with the diversity of Grid computing, an adaptive economic-based switching framework has been developed. The market entities are defined accordingly so that they can deal with different economic models dynamically without causing any perturbation in the environment. Realizing the dynamic nature of the Grid, an agent providing runtime decision on which economic model to be used when and for what purpose, has been developed. The agent is given relative domains of strengths information to facilitate the decision making process. To maximize the potential of switching mechanism, we realized that the decision making process and the broadcasting of the decision must be happened quickly. Therefore, we mathematically defined the domains of strengths of individual models, which prevented the agent from importing relative datasets for making decision. The scalability of the domains of strengths has been evaluated through experiments. Our switching framework showed its effectiveness in dynamically switching from one economic model to another depending on the models' domains of strengths. As the potential of different models in different scenarios are now utilized, the objective functions defined by network providers can easily be optimized in a dynamic environment. Therefore, our model provided significant incentives to Grid resource providers, which would encourage them to contribute their resources for solving many computationally complex problems. This incentive can ultimately help meeting the Grid's vision of a large-scale computing platform.

The growing and dynamic interest of Grid computing has led to the deployment of different market mechanisms. To be adaptive with future Grid computing environment, dynamic and reliable organization of different market mechanisms are essential. Therefore, understanding the languages and values of these market mechanisms would help to evolve the Grid's vision of a worldwide virtual computing organization.

8.4 Future Directions

The performance and suitability of different economic models in Grid computing have been examined in this thesis. It contributed a new dimension for analyzing the computational economy for future Grids. Accordingly, the thesis gave rise to some challenges such as practical-ability that need to be evaluated with importance.

This thesis has opened up a new door for economic-based distributed resource management in Grid computing and still requires to be furnished with the following works.

8.4.1 Supporting Multiple Application Types

The application of Grid computing has been widening. Therefore, the execution model must support multiple application types. Currently, our framework is suitable for only Bag-Of-Tasks (BOT) type applications, such as data mining, design exploration or parameter sweep. This kind of applications does not require the tasks/jobs/Gridlets to communicate among each other. However, there are other types of applications, such as Message Passing Interface (MPI) and workflow, those of which require communication among the tasks. Simulations in Bio-informatics and weather forecasting are examples of such kind of applications. We aim to design our framework to support MPI or workflow type and to evaluate the performances of the different economic models in terms of those applications.

The aforementioned applications are computationally intensive. However, a large community is also seeking the possibility of utilizing Grid resources for data-intensive applications such as investigating material properties. In data Grids, thousands or millions of datasets are stored and replicated across a Grid network. These datasets are, then, invoked and processed to generate meaningful results to users. A range of parameters, such as bandwidth for data replication, data storage capacity, computational requirements, and data security, can be managed and regulated successfully using economic models. Therefore, we would like to investigate the performance of different economic models also for data-intensive applications.

8.4.2 Measuring Real-time Adaptability

At present, the evaluation of our framework is limited only to simulation. A crucial idea would be to measure the real-time adaptability of our system using Grid computing test-beds. The test-beds constitute real Grid nodes that can be customized and networked to form virtual Grid organizations, to deliver a suitable platform for testing the performance of different application models against realistic scenarios. Currently, there are quite a few test-beds available in the community for evaluating Grid-based models – GridBuilder (Childs et al., 2006), Xen Grid Engine (Fallenbeck et al., 2006), and Grid Gateway (Childs et al., 2005).

We further plan to test our system using synthetic Grid workloads. Synthetic workloads are useful in terms of evaluating any system against real Grid applications. Synthetic workloads can be modeled using the parameters fetched from real traces or a set of predefined parameters by the workload designer. Grenchmark is an example of such a workload generating engine, which can be used to generate varieties of applications (MPI, BOT) (Iosup and Epema, 2006). A given model can then be evaluated in terms of the generated workloads.

8.4.3 Supporting Multi-criteria Optimization and Improving the Switching Agent

Currently, our switching framework supports optimization only for a single performance metric. We aim to improve the switching agent so that it can provide decisions suitable for optimizing multiple performance metrics. In such a scenario, a Grid provider might want to optimize a combination of metrics, such as revenue, -average turnaround time per job and social welfare. There are several methods including *Genetic* and *Evolutionary* algorithms in the literature to solve such problems. One of the suitable approaches within these algorithms that we plan to explore is the ranking method. According to this method, a number of objective functions that need to be optimized are ranked in terms of the provider's preference values. This preference relation is then applied over the solution space, which consists of the suitable models identified for those optimization functions. The agent will then dynamically decide which model to use based on the solution space.

The evaluation of our switching framework in real Grid computing networks might lead to some inefficiency in terms of reflecting on a desired supply and demand ratio due to the random nature of network latency. Therefore, we aim to design our agent using fuzzy intelligence so that it can conduct switching process depending on approximate values obtaining from the network behavior.

8.4.4 Supporting Cloud Computing Infrastructure

Cloud is another recently emerged distributed computing paradigm that can be distinguished from other traditional computing platforms for its focus on additional scalability, dynamic configuration and virtualized services. In spite of the additional focus on Cloud computing, Grid and Cloud basically share the similar concepts and, therefore, face similar challenges, such as dynamic resource configuration, resource utilization, scalability, and data management. Moreover, Grid computing is generally considered as the backbone of Cloud computing and they need each other for the evolution of future computing platform (Foster et al., 2008). Therefore, we would like to explore the possibility of employing our switching framework also

in Cloud services. The commercialized approach of Cloud computing further encourages us to evaluate our framework in that particular environment. For the scalability, we aim to test our system for an extended simulation space. We aim to explore the most widely discussed Cloud computing testbed “The Eucalyptus” (Nurmi et al., 2009).

Bibliography

- ABDELKADER, K., BROECKHOVE, J. & VANMECHELEN, K. 2010. Resource Pricing In A Dynamic Multi-Commodity Market For Computational Resources. *International Journal of Computer Networks & Communications*, 2, 74-87.
- ABRAMSON, D., BETHWAITE, B., ENTICOTT, C., GARIC, S. & PEACHEY, T. 2011. Parameter Exploration in Science and Engineering Using Many-Task Computing. *IEEE Trans. Parallel Distrib. Syst.*, 22, 960-973.
- ALLEN, G., DAVIS, K., GOODALE, T., HUTANU, A., KAISER, H., KIELMANN, T., MERZKY, A., NIEUWPOORT, R. V., REINEFELD, A., SCHINTKE, F., SCHÜTT, T., SEIDEL, E. & ULLMER, B. 2005. The Grid Application Toolkit: Toward Generic and Easy Application Programming Interfaces for the Grid. *Proceedings of the IEEE*, 93, 534-550.
- Amazon CloudWatch Service, 2012a, *Amazon CloudWatch*, viewed 31 August, 2012, <<http://aws.amazon.com/cloudwatch/>>.
- Amazon Web Services, 2012b, *Amazon EC2 Pricing*, viewed 29 August, 2012, <<http://aws.amazon.com/ec2/pricing/>>.
- AN, B., LESSER, V., IRWIN, D. & ZINK, M. 2010. Automated negotiation with decommitment for dynamic resource allocation in cloud computing. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*. Toronto, Canada: International Foundation for Autonomous Agents and Multiagent Systems.
- ANDRZEJAK, A. & ZHICHEN, X. Scalable, efficient range queries for grid information services. In *Proceedings of Second International Conference on Peer-to-Peer Computing*, (P2P 2002), 2002. 33-40.
- ARDAIZ, O., ARTIGAS, P., EYMANN, T., FREITAG, F., NAVARRO, L. & REINICKE, M. 2006. The catallaxy approach for decentralized economic-based allocation in Grid resource and service markets. *Applied Intelligence*, 25, 131-145.
- ASSUNCAO, M. & BUYYA, R. An evaluation of communication demand of auction protocols in grid environments. *Proceedings of the 3rd International Workshop on Grid Economics and Business (GECON 2006)*, 2006. World Scientific Press.
- ASSUNCAO, M. & BUYYA, R. A Cost-Aware Resource Exchange Mechanism for Load Management across Grids. *14th IEEE International Conference on Parallel and Distributed Systems (ICPADS '08)*, 2008. 213-220.
- ASSUNCAO, M., STREITBERGER, W., EYMANN, T. & BUYYA, R. Enabling the Simulation of Service-Oriented Computing and Provisioning Policies for

Autonomic Utility Grids. Proceedings of the 4th International Workshop on Grid Economics and Business (GECON'07), 2007 Rennes, France.

ATTANASIO, A., GHIANI, G., GRANDINETTI, L. & GUERRIERO, F. 2006. Auction algorithms for decentralized parallel machine scheduling. *Parallel Computing*, 32, 701-709.

AUYOUNG, A., CHUN, B. N., SNOEREN, A. C. & VAHDAT, A. Resource Allocation in Federated Distributed Computing Infrastructures. In Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure (OASIS 2004), 2004 Boston, USA.

BECK, R., SCHWIND, M. & HINZ, O. 2008. Grid Economics in Departmentalized Enterprises. *J Grid Computing*, 6, 277-290.

BORISSOV, N., NEUMANN, D. & WEINHARDT, C. 2010. Automated bidding in computational markets: an application in market-based allocation of computing services. *Autonomous Agents and Multi-Agent Systems*, 21, 115-142.

BOSSENBROEK, A., TIRADO-RAMOS, A. & SLOOT, P. M. A. 2009. Grid Resource Allocation by Means of Option Contracts. *Systems Journal, IEEE*, 3, 49-64.

BRANDIC, I., VENUGOPAL, S., MATTESS, M. & BUYYA, R. Towards a Meta-Negotiation Architecture for SLA-Aware Grid Services. Workshop on Service-Oriented Engineering and Optimizations 2008. In conjunction with International Conference on High Performance Computing 2008 (HiPC 2008), December 17–20 2008 Bangalore, India.

BROBERG, J., VENUGOPAL, S. & BUYYA, R. 2008. Market-oriented Grids and Utility Computing: The State-of-the-art and Future Directions. *Journal of Grid Computing*, 6, 255-276.

BRUNELLE, J. A., HURST, P., HUTH, J., KANG, L., NG, C., PARKES, D. C., SELTZE, M. I., SHANK, J. & YOUSSEF, S. 2006. EGG: an extensible and economics-inspired open Grid computing platform. *3rd International Workshop on Grid Economics and Business Models (GECON 2006)*, Singapore, 140-150.

BUYYA, R. *GridSim Documentation*, Cloud Computing and Distributed Systems (CLOUDS) Laboratory, viewed 15 August, 2012, <<http://www.buyya.com/gridsim/doc/faq.txt>>.

BUYYA, R. 2002. *Economic-based distributed resource management and scheduling for Grid computing*. Ph.D thesis, Monash University.

BUYYA, R. & ABRAMSON, D. 2009. The Nimrod/G Grid Resource Broker for Economic-based Scheduling. *Market-Oriented Grid and Utility Computing (Wiley Series on Parallel and Distributed Computing)*.

BUYYA, R., ABRAMSON, D. & GIDDY, J. An economy driven resource management architecture for global computational power grids. In proceedings of international

conference on parallel and distributed processing techniques and applications (PDPTA 2000), 2000a Las Vegas, USA.

- BUY YA, R., ABRAMSON, D. & GIDDY, J. Nimrod/G: An Architecture for a resource Management and Scheduling System in a Global Computational Grid. 4th International Conference on High Performance Computing 2000b Asia-Pacific Region, Beijing, China. IEEE Computer Society, USA.
- BUY YA, R., ABRAMSON, D. & GIDDY, J. Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid. The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, 2000c. 283-289.
- BUY YA, R., ABRAMSON, D., GIDDY, J. & STOCKINGER, H. 2002. Economic models for resource management and scheduling in Grid computing. *Concurrency and Computation: Practice and Experience*, 14, 1507-1542.
- BUY YA, R., ABRAMSON, D. & VENUGOPAL, S. 2005. The Grid Economy. *Proceedings of the IEEE*, 93, 698-714.
- BUY YA, R. & MURSHED, M. 2002. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency and Computation: Practice and Experience*, 14, 1175-1220.
- BUY YA, R., RANJAN, R. & CALHEIROS, R. 2010. InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services Algorithms and Architectures for Parallel Processing. *In: Ching-Hsien Hsu, Yang, Laurence, Park, Jong & Yeo, Sang-Soo (eds.). Springer Berlin / Heidelberg.*
- BUY YA, R., VENUGOPAL, S., RANJAN, R. & YEO, C. S. 2009a. The Gridbus Middleware for Market-Oriented Computing. *Market-Oriented Grid and Utility Computing*. John Wiley & Sons, Inc.
- BUY YA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J. & BRANDIC, I. 2009b. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25, 599-616.
- CAO, J., J., K. D. & R., N. G. Performance evaluation of an agent-based resource management infrastructure for grid computing. *Proceedings. First IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2001. 311-318.
- CAO, J., JARVIS, S. A., SAINI, S., KERBYSON, D. J. & NUDD, G. R. 2002. ARMS: An agent-based resource management system for grid computing. *Scientific Programming*, 10, 135-148.
- CARACAS, A. & ALTMANN, J. 2007. A pricing information service for grid computing. *Proceedings of the 5th international workshop on Middleware for grid computing: held at the ACM/IFIP/USENIX 8th International Middleware Conference*. Newport Beach, California: ACM.

- CARAMIA, M. & GIORDANI, S. 2008. Resource allocation in grid computing: an economic model. *J. WSEAS Trans. Comp.*, 3, 19-27.
- CASANOVA, H. Simgrid: a toolkit for the simulation of application scheduling. In Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001. 430-437.
- CASSADY, R. 1967. *Auctions and auctioneering*, University of California Press.
- CHAN, P. 2010, *Monash Sun Grid*, Monash University, Australia, viewed 16 August, 2012, <<http://www.monash.edu.au/eresearch/services/mcg/msg.html>>.
- CHAO, I., SANGUESA, R., ARDAIZ, O., JOITA, L. & RANA, O. F. 2009. Optimizing decentralised grid markets through group selection. *Int. J. Web Grid Serv.*, 4, 357-366.
- CHELIOTISY, G., KENYONY, C. & BUYYA, R. 2003. Grid Economics: 10 Lessons from Finance. *Joint Technical Report GRIDS-TR-2003-3, IBM Research Zurich and Grid Computing and Distributed Systems Laboratory and University of Melbourne*.
- CHEN, M., YANG, G. & LIU, X. 2004. Gridmarket: A Practical, Efficient Market Balancing Resource for Grid and P2P Computing Grid and Cooperative Computing. In: Minglu Li, Sun, Xian-He, Deng, Qianni & Ni, Jun (eds.). Springer Berlin / Heidelberg.
- CHILDS, S., COGHLAN, B. & MCCANDLESS, J. GridBuilder: A Tool for Creating Virtual Grid Testbeds. Second IEEE International Conference on e-Science and Grid Computing (e-Science '06), 2006. 77-77.
- CHILDS, S., COGHLAN, B., O'CALLAGHAN, D., QUIGLEY, G. & WALSH, J. A single-computer grid gateway using virtual machines. 19th International Conference on Advanced Information Networking and Applications (AINA 2005), 2005. 310-315.
- CHULIANG, W., MINGLU, L. & XINDA, L. 2007. Grid resource management based on economic mechanisms. *J. Supercomput.*, 42, 181-199.
- CHUN, B. N., BUONADONNA, P., AUYOUNG, A., NG, C., PARKES, D. C., SHNEIDMANY, J., SNOEREN, A. C. & VAHDAT, A. Mirage: A Microeconomic Resource Allocation System for Sensornet Testbeds. The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II., 2005. 19-28.
- CIRNE, W., BRASILEIRO, F., SAUVÉ, J., ANDRADE, N., PARANHOS, D., SANTOS-NETO, E., MEDEIROS, R. & CAMPINA, F. Grid computing for bag of tasks applications. Proceedings of the 3rd IFIP Conference on E-Commerce, E-Business and E-Government, 2003 Brazil.
- COHEN, R. & FESER, E. 2003. Grid computing, Projected Impact in North Carolina's Economy and Broadband Use through 2010. *Rural Internet Access Authority*.

- CZAJKOWSKI, K., FOSTER, I., KARONIS, N., KESSELMAN, C., MARTIN, S., SMITH, W. & TUECKE, S. 1998. A resource management Architecture for metacomputing systems. Berlin, Germany: Springer.
- DANAK, A. 2011. Efficient Bidding in Dynamic Grid Markets. *IEEE Transactions on Parallel and Distributed Systems*, 22, 1483-1496.
- DASH, R. K., JENNINGS, N. R. & PARKES, D. C. 2003. Computational-Mechanism Design: A Call to Arms. *IEEE Intelligent Systems*, 18, 40-47.
- DIJKSTRA, E. W. 1965. Solution of a problem in concurrent programming control. *Commun. ACM*, 8, 569.
- DOMINIAK, M., GANZHA, M., GAWINECKI, M., KURANOWSKI, W., PAPRZYCKI, M., MARGENOV, S. & LIRKOV, I. 2007. Utilizing Agent Teams in Grid Resource Brokering. *Multiagent and Grid Systems*.
- DUMITRESCU, C. L. & FOSTER, I. GangSim: a simulator for grid scheduling studies. IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), 2005. 1151-1158.
- ERNEMANN, C. & YAHYAPOUR, R. 2004. Applying economic scheduling methods to Grid environments. *Grid resource management: state of the art and future trends*. Kluwer Academic Publishers.
- EYMANN, T., KÖNIG, S. & MATROS, R. 2008. A Framework for Trust and Reputation in Grid Environments. *Journal of Grid Computing*, 6, 225-237.
- EYMANN, T., NEUMANN, D., REINICKE, M., SCHNIZLER, B., STREITBERGER, W. & VEIT, D. On the Design of a Two-Tiered Grid Market Structure. In Proceedings of the Multikonferenz Wirtschaftsinformatik (MKWI'06), 2006 Passau, Germany.
- FALAVINHA, J. N., MANACERO, A., LIVNY, M. & BRADLEY, D. The Owner Share Scheduler for a Distributed System. International Conference on Parallel Processing Workshops (ICPPW '09), 2009. 298-305.
- FALLENBECK, N., PICHT, H.-J., SMITH, M. & FREISLEBEN, B. Xen and the Art of Cluster Scheduling. First International Workshop on Virtualization Technology in Distributed Computing (VTDC 2006), 2006. 4-4.
- FELLOWS, W. & WALLAGE, S. 2007. Grid computing-The state of the Market *The 451 Group*.
- FOSTER, I. 2002. What is the Grid? - a three point checklist. *GRIDToday*, 1, 22-25.
- FOSTER, I. 2006. Globus Toolkit Version 4: Software for Service-oriented Systems. *Journal of Computer Science and Technology*, 21, 513-520.
- FOSTER, I., JENNINGS, N. R. & KESSELMAN, C. Brain meets brawn: why grid and agents need each other. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 2004. 8-15.

- FOSTER, I. & KESSELMAN, C. 2003. The Grid: Blueprint for a New Computing Infrastructure. *Morgan Kaufmann, Burlington, VT.*
- FOSTER, I., ZHAO, Y., RAICU, I. & LU, S. Cloud Computing and Grid Computing 360-Degree Compared. Grid Computing Environments Workshop, 2008. GCE '08, 2008. 1-10.
- George Glazer Gallery, GALLERY, G. G. 1997, *Sotheby's Book Auction, London*, Art and Antiques Dealers League of America, viewed 15 August, 2012, <<http://www.georgeglazer.com/prints/business/bookauct.html>>.
- GANZHA, M., PAPRZYCKI, M., DROZDOWICZ, M., SENOBARI, M., LIRKOV, I., IVANOVSKA, S., OLEJNIK, R. & TELEGIN, P. 2010. Information Flow and Mirroring in an Agent-Based Grid Resource Brokering System. *Large-Scale Scientific Computing*. Springer Berlin / Heidelberg.
- GARG, S. K., BUYYA, R. & SIEGEL, H. J. 2009. Time and cost trade-off management for scheduling parallel applications on Utility Grids. *Future Generation Computer Systems*, 26, 1344-1355.
- GHOSH, P., ROY, N., DAS, S. K. & BASU, K. 2005. A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework. *Journal of Parallel and Distributed Computing*, 65, 1366-1383.
- GOMES, E. & KOWALCZYK, R. 2010. The Dynamics of Multiagent Q-Learning in Commodity Market Resource Allocation. *Advances in Machine Learning II*. Springer Berlin / Heidelberg.
- GOMOLUCH, J. & SCHROEDER, M. Market-Based Resource Allocation for Grid Computing: A Model and Simulation. Proceedings of the First International Workshop on Middleware for Grid Computing, 2003.
- GOODALE, T., JHA, S., KAISER, H., KIELMANN, T., KLEIJER, P., LASZEWSKI, G. V., LEE, C., MERZKY, A., RAJIC, H. & SHALF, J. 2008. SAGA: A Simple API for Grid Applications. High-level application programming on the Grid.
- GOSWAMI, K. & GUPTA, A. Resource Selection in Grids Using Contract Net. 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), 2008. 105-109.
- GRADWELL, P. & PADGET, J. Distributed combinatorial resource scheduling. In Proceedings of the 1st International Workshop on Smart Grid Technologies (SGT-2005), 2005. 17-32.
- GROSU, D. & DAS, A. Auction-Based Resource Allocation Protocols in Grids. in the Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), 2004 Cambridge, MA. ACTA Press, 20-27.
- GUTIERREZ-GARCIA, J. O. & SIM, K.-M. 2010. Agent-Based Service Composition in Cloud Computing Grid and Distributed Computing, Control and Automation. *In:*

Tai-hoon Kim, Yau, Stephen S., Gervasi, Osvaldo, Kang, Byeong-Ho, Stoica, Adrian & Ślęzak, Dominik (eds.). Springer Berlin Heidelberg.

HANDS, D. W. 2003. *Introductory Mathematical Economics*, Oxford University Press.

HAQUE, A., ALHASHMI, S. M. & PARTHIBAN, R. 2011. A survey of economic models in grid computing. *Future Generation Computer Systems*, 27, 1056-1069.

HE, M., LEUNG, H.-F. & R., J. N. 2003. A fuzzy-logic based bidding strategy for autonomous agents in continuous double auctions. *Knowledge and Data Engineering, IEEE Transactions on*, 15, 1345-1363.

HOWIE, H. H., S., G. A. & F., K. J. You Can't Always Get What You Want: Achieving Differentiated Service Levels with Pricing Agents in a Storage Grid. IEEE/WIC/ACM International Conference on Web Intelligence, 2007. 123-131.

IOSUP, A. & EPEMA, D. GRENCHMARK: A Framework for Analyzing, Testing, and Comparing Grids. 6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID 06), 2006. 313-320.

IZAKIAN, H., ABRAHAM, A. & LADANI, B. T. 2009. An auction method for resource allocation in computational grids. *Future Generation Computer Systems*, 26, 228-235.

JIADAO, L. & YAHYAPOUR, R. R. A strategic negotiation models for grid scheduling. International transactions on systems science and applications, 2007. 411-420.

JUHASZ, Z. & PAUL, P. Scalability Analysis of the Contract Net Protocol. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2002. 346-346.

KACSUK, P., DOZSA, G., KOVACS, J., LOVAS, R., PODHORSZKI, N., BALATON, Z. & GOMBAS, G. 2003. P-GRADE: A Grid Programming Environment. *Journal of Grid Computing*, 1, 171-197.

KAKARONTZAS, G., SAVVAS, I. K. & STAMELOS, I. 2011. Agents, clusters and components: A synergistic approach to the GSP. *Future Generation Computer Systems*, 27, 999-1010.

KANT, U. & GROSU, D. Double auction protocols for resource allocation in grids. International Conference on Information Technology: Coding and Computing (ITCC 2005), 2005. 366-371.

KENYON, C. & CHELIOTIS, G. 2003. Grid Resource Commercialization: Economic Engineering and Delivery Scenarios. *IBM Zurich Research Lab*.

KEPHART, J. O. & CHESS, D. M. 2003. The vision of autonomic computing. *Computer*, 36, 41-50.

KLUSACEK, D. & RUDOVA, H. Alea 2 – Job Scheduling Simulator. Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010), ICST (2010), 2007 Malaga, Spain.

KRISHNA, V. 2002. *Auction Theory*, Elsevier.

KUN, W., LI, L., HAUSHEER, D., ZHIYONG, L., WEI, L., DENIAN, S., GUILI, H. & STILLER, B. A trust-incentive-based combinatorial double auction algorithm. Network Operations and Management Symposium (NOMS), 2010 IEEE, 19-23 April 2010 2010. 209-215.

LAI, K., RASMUSSEN, L., ADAR, E., ZHANG, L. & HUBERMAN, B. A. 2005. Tycoon: An implementation of a distributed, market-based resource allocation system. *Multiagent Grid System*, 1, 169-182.

LEGRAND, A. & QUINSON, M. 2001, *SimGrid: Scalable simulation of distributed systems*, The National Institute for Research in Computer Science and Control (INRIA), viewed 15 August, 2012, <<http://simgrid.gforge.inria.fr/faq.html>>.

LEON, X., TRINH, T. A. & NAVARRO, L. 2010. Using economic regulation to prevent resource congestion in large-scale shared infrastructures. *Future Generation Computer Systems*, 26, 599-607.

LI, C. & LI, L. 2004. Competitive proportional resource allocation policy for computational grid. *Future Generation Computer Systems*, 20, 1041-1054.

LI, C. & LI, L. 2006. Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Applied Intelligence*, 25, 147-158.

LI, C., LI, L. & LU, Z. 2005. Utility driven dynamic resource allocation using competitive markets in computational grid. *Advances in Engineering Software*, 36, 425-434.

LI, L., LIU, Y.-A., LIU, K.-M., MA, X.-L. & YANG, M. 2009. Pricing in combinatorial double auction-based grid allocation model. *The Journal of China Universities of Posts and Telecommunications*, 16, 59-65.

LU, L. & YANG, S. 2006. DIRSS-G: an Intelligent Resource Scheduling System for Grid Environment Based on Dynamic Pricing. *International Journal of Information Technology*, 12.

LYNAR, T. M., HERBERT, R. D. & SIMON, S. 2009. Auction resource allocation mechanisms in grids of heterogeneous computers. *J. WSEAS Transactions on Computers*, 8, 1671-1680.

Microsoft, 2012c, *Windows Azure Pricing*, viewed 31 August, 2012, <<http://www.microsoft.com/windowsazure/pricing/>>.

MIDDLETON, S. E., SURRIDGE, M., BENKNER, S. & ENGELBRECHT, G. 2007. Quality of Service Negotiation for Commercial Medical Grid Services *Journal of Grid Computing*, 5, 429-447.

- MILGROM, P. & ROBERT, W. 1982. A Theory of Auctions and Competitive Bidding. *Econometrica*, 50, 1089-1122.
- MILLS, K. L. & DABROWSKI, C. 2008. Can Economics-based Resource Allocation Prove Effective in a Computation Marketplace? *Journal of Grid Computing*, 6, 291-311.
- MINH, Q. D. & ALTMANN, J. Bilateral Bargaining Game and Fuzzy Logic in the System Handling SLA-Based Workflow. 22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW 2008), 2008. 1609-1614.
- NASCIMENTO, A. P., SENA, A. C., SILVA, J. A. D., VIANNA, D. Q. C., BOERES, C. & REBELLO, V. E. F. Managing the execution of large scale MPI applications on computational grids. 17th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2005), 2005. 69-76.
- NEIGER, G. 2003, *PODC Influential Paper Award: 2002*, ACM Symposium on Principles of Distributed Computing, viewed 16 August, 2012, <<http://www.podc.org/influential/2002.html>>.
- NEUBAUER, F., HOHEISEL, A. & GEILER, J. 2006. Workflow-based Grid applications. *Future Generation Computer Systems*, 22, 6-15.
- NEUMANN, D., STÖBER, J., WEINHARDT, C. & NIMIS, J. 2008. A Framework for Commercial Grids—Economic and Technical Challenges. *Journal of Grid Computing*, 6, 325-347.
- NI, W.-C., LIU, L.-C. & WU, C. 2007. Services selection based on commodity-market in grid workflow. *Journal of Computer Applications*, 27, 2973-2975.
- NURMI, D., WOLSKI, R., GRZEGORCZYK, C., OBERTELLI, G., SOMAN, S., YOUSEFF, L. & ZAGORODNOV, D. The Eucalyptus Open-Source Cloud-Computing System. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09), 2009. 124-131.
- OSTERMANN, S., IOSUP, A., YIGITBASI, N., PRODAN, R., FAHRINGER, T. & EPEMA, D. 2010. A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing Cloud Computing. In: Dimiter R. Avresky, Diaz, Michel, Bode, Arndt, Ciciani, Bruno & Dekel, Eliezer (eds.). Springer Berlin Heidelberg.
- OUELHADJ, D., GARIBALDI, J., MACLAREN, J., SAKELLARIOU, R. & KRISHNAKUMAR, K. 2005. A Multi-agent Infrastructure and a Service Level Agreement Negotiation Protocol for Robust Scheduling in Grid Computing. *Advances in Grid Computing-EGC 2005*. Springer-Verlag.
- PANDE, V. 2000, *Folding@home: distributed computing*, National Science Foundation, USA, viewed 15 August, 2012, <<http://folding.stanford.edu/English/HomePage>>.
- PARKES, D. C. & UNGAR, L. H. 2000. Iterative combinatorial auctions: theory and practice.

- PARSONS, S., MARCINKIEWICZ, M., NIU, J. & PHELPS, S. 2006, *Everything you wanted to know about double auctions, but were afraid to (bid or) ask*, Brooklyn College, New York, viewed 15 August, 2012, <<http://www.sci.brooklyn.cuny.edu/%7Eparsons/projects/mech-design/publications/cda.pdf>>.
- PAUROBALLY, S. 2007. A Framework for Web service negotiation. *J. ACM Transaction of Autonomous and Adaptive System*, 2, 14.
- PAUROBALLY, S. 2010. SLA Negotiation for VO Formation. *Grids and Service-Oriented Architectures for Service Level Agreements*. Springer US.
- PLACEK, M. & BUYYA, R. 2006. Storage Exchange: A Global Trading Platform for Storage Services. *12th International European Parallel Computing Conference (EuroPar)*, LNCS. Berlin, Germany: Springer-Verlag.
- PLASZCZAK, P. & WELLNER, R. 2005. *Grid Computing*, Morgan Kaufmann Publishers.
- POUREBRAHIMI, B., BERTELS, K., KANDRU, G. M. & VASSILIADIS, S. Market-Based Resource Allocation in Grids. Second IEEE International Conference on e-Science and Grid Computing, e-Science '06., 2006. 80-80.
- POUREBRAHIMI, B., BERTELS, K., VASSILIADIS, S. & ALIMA, L. O. 2007. A Dynamic Pricing and Bidding Strategy for Autonomous Agents in Grids. *6th International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2007)*, Honolulu, Hawaii, USA, May 14-18, 2007, *Revised and Selected Papers*. Springer-Verlag.
- PRODAN, R., WIECZOREK, M. & FARD, H. 2011. Double Auction-based Scheduling of Scientific Applications in Distributed Grid and Cloud Environments. *Journal of Grid Computing*, 1-18.
- RAMAKRISHNAN, L. 2008. A Survey of Distributed Workflow Characteristics and Resource Requirements. USA: Indiana University.
- RANJAN, R., HARWOOD, A. & BUYYA, R. 2008. Sla-based cooperative superscheduling algorithms for computational grids. *J. ACM Transaction of Autonomous and Adaptive System*.
- RESINAS, M., FERNANDEZ, P. & CORCHUELO, R. 2006. A Conceptual Framework for Automated Negotiation Systems Intelligent Data Engineering and Automated Learning – IDEAL 2006. In: Emilio Corchado, Yin, Hujun, Botti, Vicente & Fyfe, Colin (eds.). Springer Berlin / Heidelberg.
- RISCH, M., BRANDIC, I. & ALTMANN, J. 2010. Using SLA Mapping to Increase Market Liquidity. In: Asit Dan, Gittler, Frédéric & Toumani, Farouk (eds.) *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. Springer Berlin / Heidelberg.

- ROURE, D. D., BAKER, M. A. & JENNINGS, N. R. 2003. The Evolution of the Grid. *In: Fran Berman, Fox, Geoffrey & Hey, Tony (eds.) Grid computing: making the global infrastructure a reality* John Wiley & Sons Ltd.
- SANDHOLM, T., LAI, K., ORTIZ, J. A. & ODEBERG, J. Market-Based Resource Allocation using Price Prediction in a High Performance Computing Grid for Scientific Applications. 15th IEEE International Symposium on High Performance Distributed Computing, 2006. 132-143.
- SAPRA, P., MEMORIA, M. & SUNAINA 2010. Multi-Agent Systems for Adaptive and Efficient Job Scheduling Service in Grids. *International Journal of Computer Applications*, 1, 19-23.
- SCHNIZLER, B. 2008. MACE: A Multi-attribute Combinatorial Exchange. *Negotiation, Auctions, and Market Engineering*. Springer Berlin Heidelberg.
- SHEN, W., LI, Y., GHENNIWA, H. H. & WANG, C. 2002. Adaptive Negotiation for Agent-Based Grid Computing. *Integrated Manufacturing Technologies Institute National Research Council Canada*.
- SHERWANI, J., ALI, N., LOTIA, N., HAYAT, Z. & BUYYA, R. 2004. Libra: a computational economy-based job scheduling system for clusters. *Journal of Software: Practice and Experience*, 34, 573-590.
- SIM, K. 2010. Towards Complex Negotiation for Cloud Economy Advances in Grid and Pervasive Computing. *In: Paolo Bellavista, Chang, Ruay-Shiung, Chao, Han-Chieh, Lin, Shin-Feng & Sloot, Peter (eds.)*. Springer Berlin / Heidelberg.
- SIM, K. M. 2005. A Survey of Bargaining Models for Grid Resource Allocation. *ACM SIGecom Exchanges*, 5, 22-32.
- SIM, K. M. 2007. Relaxed-criteria G-negotiation for Grid resource co-allocation. *ACM SIGECOM: E-commerce Exchanges* 6, 37-46.
- SMITH, R. G. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *Computers, IEEE Transactions on*, C-29, 1104-1113.
- SONMEZ, O. O. & GURSOY, A. 2007. A Novel Economic-Based Scheduling Heuristic for Computational Grids. *International Journal of High Performance Computing Applications*, 21, 21-29.
- STEFANO, A. D. & SANTORO, C. 2008. An economic model for resource management in a Grid-based content distribution network. *Future Generation Computer Systems*, 24, 202-212.
- STREITBERGER, W., HUDERT, S., EYMANN, T., SCHNIZLER, B., ZINI, F. & CATALANO, M. 2008. On the Simulation of Grid Market Coordination Approaches. *Journal of Grid Computing*, 6, 349-366.

- STUER, G., VANMECHELEN, K. & BROECKHOVE, J. 2007. A commodity market algorithm for pricing substitutable Grid resources. *Future Generation Computer Systems*, 23, 688-701.
- SUBRATA, R., ZOMAYA, A. Y. & LANDFELDT, B. 2009. Cooperative power-aware scheduling in grid computing environments. *Journal of Parallel and Distributed Computing*, 70, 84-91.
- SURI, P. K. & SINGH, M. 2009. Auction Oriented Resource Allocation in Grid Using Knowledge Based Pricing Policy. *International Journal of Computer Science and Network Security*, 9.
- TAN, Z. & GURD, J. R. Market-based grid resource allocation using a stable continuous double auction. 8th IEEE/ACM International Conference on Grid Computing, 2007. 283-290.
- TAYLOR, I., WANG, I., SHIELDS, M. & MAJITHIA, S. 2005. Distributed computing with Triana on the Grid. *Concurrency and Computation: Practice and Experience*, 17, 1197-1214.
- THABET, I., BOUSLIMI, I., HANACHI, C. & GHÉDIRA, K. 2011. A Multi-agent Organizational Model for Grid Scheduling Agent and Multi-Agent Systems: Technologies and Applications. In: James O'Shea, Nguyen, Ngoc, Crockett, Keeley, Howlett, Robert & Jain, Lakhmi (eds.). Springer Berlin / Heidelberg.
- The Economist, 1999, *The Heyday of the Auction*, The Economist, viewed 19 August, 2012, <<http://www.economist.com/node/226168>>.
- TIANFIELD, H. Towards agent based grid resource management. IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2005), 2005. 590-597.
- TURNER, D. M., PREVELAKIS, V. & KEROMYTOS, A. D. 2010. A market-based bandwidth charging framework. *ACM Transactions on Internet Technology (TOIT)*, 10, 1-30.
- VALKENHOEF, G., RAMCHURN, S. D., VYTELINGUM, P., JENNINGS, N. R. & VERBRUGGE, R. 2010. Continuous Double Auctions with Execution Uncertainty. In: Esther David, Gerding, Enrico, Sarne, David & Shehory, Onn (eds.) *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer Berlin Heidelberg.
- VANMECHELEN, K., DEPOORTER, W. & BROECKHOVE, J. 2008. A Simulation Framework for Studying Economic Resource Management in Grids Computational Science – ICCS 2008. In: Marian Bubak, Albada, Geert, Dongarra, Jack & Sloot, Peter (eds.). Springer Berlin / Heidelberg.
- VANMECHELEN, K., DEPOORTER, W. & BROECKHOVE, J. 2011. Combining Futures and Spot Markets: A Hybrid Market Approach to Economic Grid Resource Management. *Journal of Grid Computing*, 9, 81-94.

- VECCHIOLA, C., CHU, X. & BUYYA, R. 2009. Aneka: A Software Platform for .NET-based Cloud Computing.
- VEIT, D., ALTMANN, J., VANMECHELEN, K. & BROECKHOVE, J. 2007. A Comparative Analysis of Single-Unit Vickrey Auctions and Commodity Markets for Realizing Grid Economies with Dynamic Pricing. *Grid Economics and Business Models*. Springer Berlin / Heidelberg.
- VON LASZEWSKI, G., FOSTER, I., GAWOR, J. & LANE, P. 2001. A Java commodity grid kit. *Concurrency and Computation: Practice and Experience*, 13, 645-662.
- WANG, J., XU, Y., LIU, G., PAN, Z. & HAO, Y. A New Resource Discovery Mechanism with Negotiate Solution Based on Agent in Grid Environments. The 3rd International Conference on Grid and Pervasive Computing Workshops (GPC Workshops '08), 2008. 23-28.
- WANG, Y. & WANG, X. A Novel Double Auction Mechanism Based Resource Allocation in the Grid. International Conference on Management of e-Commerce and e-Government, ICMECG '09, 2009. 420-423.
- WEISS, G. 2000. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Cambridge, Massachusetts, MIT Press.
- WIECZOREK, M., PODLIPNIG, S., PRODAN, R. & FAHRINGER, T. Applying double auctions for scheduling of workflows on the Grid. International Conference for High Performance Computing, Networking, Storage and Analysis (SC 2008), 2008. 1-11.
- WITHERS, S. 2002, *Power grid*, Technology & Business, viewed 15 August, 2012, <<http://www.buyya.com/press/TechBusiness02.pdf>>.
- WOLSKI, R., PLANK, J. S., BREVIK, J. & BRYAN, T. G-commerce: market formulations controlling resource allocation on the computational grid. Proceedings 15th International Parallel and Distributed Processing Symposium, 2001. 8 pp.
- XING, L. & LAN, Z. A Method Based on Iterative Combinatorial Auction Mechanism for Resource Allocation in Grid Multi-agent Systems. International Conference on Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09, 2009. 36-39.
- YEO, C. S. & BUYYA, R. 2007. Pricing for Utility-Driven Resource Management and Allocation in Clusters. *International Journal of High Performance Computing Applications*, 21, 405-418.
- ZHAO, H. & LI, X. Efficient Grid Task-Bundle Allocation Using Bargaining Based Self-Adaptive Auction. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09), 2009. 4-11.