THE DEADLINE-BASED SCHEDULING OF DIVISIBLE REAL-TIME WORKLOADS ON MULTIPROCESSOR PLATFORMS

SURIAYATI BT CHUPRAT

UNIVERSITI TEKNOLOGI MALAYSIA

THE DEADLINE-BASED SCHEDULING OF DIVISIBLE REAL-TIME WORKLOADS ON MULTIPROCESSOR PLATFORMS

SURIAYATI BT CHUPRAT

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Doctor of Philosophy (Mathematics)

Faculty of Science
Universiti Teknologi Malaysia

JUNE 2009

To

my beloved husband, Abd Hadi,

my loving childrens, Aizat and Ainaa,

and

my loving and supportive parents,

Hj Chuprat and Hajjah Rabahya.

ACKNOWLEDGEMENT

First of all, I thank ALLAH (SWT), the Lord Almighty, for giving me the health, strength, ability to complete this work and for blessing me with supportive supervisors, family and friends.

I wish to express my deepest appreciation to my supervisor, Professor Dr Shaharuddin Salleh for his idea, support, enthusiasm, and patience. I have learnt an enormous amount from working with him. My special thanks to my co-supervisor Professor Dr Sanjoy K. Baruah for his guidance, support, respect, and kindness. The opportunity to collaborate with him during my five months visit at the University of North Carolina, Chapel Hill, USA has benefited my research tremendously.

I would also like to thank Professor Dr James H. Anderson for giving me the opportunity to attend his class on Real-time Systems at UNC. Thanks to the Real Time Systems Group of UNC (Nathan, Bjeorn, John, Aaron, Hennadiy) for sharing their dynamic discussions during the “Real-time Lunch” weekly meeting.

I am forever indebted to my employer Universiti Teknologi Malaysia (UTM) for granted me the study leave, funds and the facilities for my research. Thanks to all management staff of KST Kuala Lumpur, HRD Skudai and Canselori UTMKL.

Life would be harder without the support of many good friends. Thanks to Dr Zuraini and Dr Ruzana for being such a good mentor, Arbai’ah and Haslina for sharing the challenging PhD years, Dr Nazli, Dr Maslin and Dr Liza for their support and motivations. Thank you to all my friends in KST Kuala Lumpur and UTM Skudai.

Finally, I thank to all my family members for their love, patient and uncountable supports.

ABSTRACT

Current formal models of real-time workloads were designed within the context of uniprocessor real-time systems; hence, they are often not able to accurately represent salient features of multiprocessor real-time systems. Researchers have recently attempted to overcome this shortcoming by applying workload models from *Divisible Load Theory* (DLT) to real-time systems. The resulting theory, referred to as *Real-time Divisible Load Theory* (RT-DLT), holds great promise for modeling an emergent class of massively parallel real-time workloads. However, the theory needs strong formal foundations before it can be widely used for the design and analysis of real-time systems. The goal of this thesis is to obtain such formal foundations, by generalizing and extending recent results and concepts from multiprocessor real-time scheduling theory. To achieve this, recent results from traditional multiprocessor scheduling theory were used to provide satisfactory explanations to some apparently anomalous observations that were previously made upon applying DLT to real-time systems. Further generalization of the RT-DLT model was then considered: this generalization assumes that processors become available at different instants of time. Two important problems for this model were solved: determining the minimum number of processors needed to complete a job by its deadline; and determining the earliest completion time for a job upon a given cluster of such processors. For the first problem, an optimal algorithm called MINPROCS was developed to compute the minimum number of processors that ensure each job completes by its deadline. For the second problem, a Linear Programming (LP) based solution called MIN-ξ was formulated to compute the earliest completion time upon given number of processors. Through formal proofs and extensive simulations both algorithms have been shown to improve the non-optimal approximate algorithms previously used to solve these problems.

ABSTRAK

Model formal bagi beban kerja masa nyata asalnya direkabentuk dalam konteks sistem masa-nyata satu pemproses. Model ini kadangkala gagal mewakilkan secara tepat ciri-ciri sistem masa-nyata pemproses berbilang. Masalah ini cuba diatasi oleh para penyelidik dengan mengaplikasikan model beban kerja yang digunakan di dalam Teori Pembahagian Beban (DLT) kepada sistem masa nyata. Hasil aplikasi ini dikenali sebagai Teori Pembahagian Beban Masa Nyata (RT-DLT). Teori ini menunjukkan potensi yang meyakinkan bagi memodelkan beban kerja masa nyata selari dalam kelas besar. Walaubagaimanapun, sebelum teori ini boleh digunakan dalam merekabentuk dan analisis sistem masa nyata, ia memerlukan asas formal yang kukuh. Tujuan kajian tesis ini adalah untuk menghasilkan asas formal yang dimaksudkan dengan memperluaskan hasil kajian terkini dan menggunakan konsep dari teori sistem masa nyata pemproses berbilang. Untuk mencapai tujuan ini, hasil kajian terkini daripada teori penjadualan sistem masa nyata pemproses berbilang digunakan bagi menerangkan pemerhatian yang luar-biasa apabila Teori Pembahagian Beban diaplikasikan kepada sistem masa nyata. Tesis ini seterusnya mengkaji model Teori Pembahagian Beban Masa Nyata apabila berlaku keadaan di mana masa sedia pemproses-pemproses di dalam kluster adalah berbeza-beza. Dua masalah utama berjaya diselesaikan dalam kajian ini: menentukan bilangan minimum pemproses yang diperlukan untuk menyiapkan beban kerja sebelum sampai masa tamat; menentukan masa yang paling awal bagi menyiapkan sesuatu beban kerja. Bagi masalah pertama, satu algoritma optimal dinamakan MINPROCS telah dihasilkan. Dan untuk masalah kedua satu penyelesaian berasaskan Pengaturcaraan Lelurus yang dinamakan MIN-ξ telah direkabentuk. Melalui pembuktian formal dan beberapa siri simulasi, telah dibuktikan bahawa kedua-dua penyelesaian adalah optimal dan sekaligus algoritma yang sebelumnya digunakan untuk menyelesaikan masalah yang sama diperbaiki.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AN	All Nodes
ATLAS	AToroidal LHC ApporatuS
CMS	Compact Muon Solenoid
DAG	Directed Acyclic Graph
DLT	Divisible Load Theory
DM	Deadline Monotonic
EDF	Earliest Deadline First
EDZL	Earliest Deadline Zero Laxity
EPR	Equal Partitioning
EPU	Effective Processor Utilization
FIFO	First In First Out
IIT	Inserted Idle Time
LLF	Least Laxity First
LP	Linear Programming
MN	Minimum Nodes
MWF	Maximum Workload Derivative First
OPR	Optimal Partitioning
RM	Rate Monotonic
RT-DLT	Real-time Divisible Load Theory
SMP	Symmetric Shared Memory Multiprocessor
UMA	Uniform Memory Access
WCET	Worst Case Execution Time

LIST OF SYMBOLS

a_i	-	Arrival Time of i^{th} job
c_i	-	Execution Requirement of i^{th} job
C_m	-	Communication Cost
C_p	-	Computation Cost
c_i^j	-	Computation Time
C_i	-	Worst Case Requirement of i^{th} task
d_i	-	Deadline of i^{th} job
D_i	-	Deadline of i^{th} task
e_i	-	Worst Case Execution Time of i^{th} job
f_i	-	Completion Time of i^{th} job
J_i	-	i^{th} Job
I	-	Collection of Jobs
L_i	-	i^{th} Link
n	-	Number of Processors
$n^{\min}$	-	Minimum Number of Processors
p_i	-	Period or Inter-arrival between successive jobs
P_i	-	i^{th} Processor
r_i	-	Ready Time i^{th} job
s_i	-	Start Time i^{th} job
$S(t)$	-	Schedule as Integer Step Function
T_i	-	Minimum Inter-arrival separation of i^{th} task
t	-	Time

U_i	-	Utilization of i^{th} task
$U_{sum}(\tau)$	-	Total Utilization of a task system τ
$U_{\max}(\tau)$	-	Maximum Utilization of a task system τ
$V_{\max}(\tau)$	-	Maximum Utilization of a task system τ in non-preemptive system
$V_{sum}(\tau)$	-	Total Utilization of a task system τ in non-preemptive system
$e_{\max}(\tau)$	-	Maximum execution time of task τ

Greek Symbols

τ_i	-	i^{th} Task
σ_i	-	i^{th} Workload
α_i	-	i^{th} Fraction of Workload
β	-	Ratio of C_p and ($C_p + C_m$)
δ	-	Density of i^{th} task
$\delta_{\max}(\tau)$	-	Total Density of a task system τ
$\delta_{sum}(\tau)$	-	Largest Density of a task system τ
ξ_i	-	Execution Time of i^{th} workload
ϕ	-	Off set
$\chi(n)$	-	Cost of executing a job

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Real-time computer application systems are systems in which the correctness of a computation depends upon both the *logical* and *temporal* properties of the result of the computation. Temporal constraints of real-time systems are commonly specified as deadlines within which activities should complete execution. For *hard-real-time systems*, meeting timing constraints is crucially important – failure to do so may cause critical failures and in some cases cause hazard to human life (Buttazzo, 2004). In *soft-real-time systems*, by contrast, the consequences of an occasional missed deadline are not as severe (Buttazzo et al., 2005). Given the central importance of meeting timing constraints in hard-real-time systems, such systems typically require guarantees prior to deployment – e.g., during system design time – that they will indeed always meet their timing constraints during run-time. This thesis is primarily concerned with hard-real-time systems.

Real-time computing will continue to play a crucial role in our society, as there are an increasing number of complex systems that needs computer control. Many next-generation computing applications such as automated manufacturing systems, defense systems (e.g. smart bombs, automotive, avionics and spacecraft control systems), high speed and multimedia communication systems, have significant real-time components (Liu, 2000; Buttazzo, 2004).

Such real-time application systems demand complex and significantly increased functionality and it is becoming unreasonable to expect to implement them upon *uniprocessor* platforms. Consequently, these systems are increasingly coming to be implemented upon *multiprocessor* platforms, with complex synchronization, data-sharing and parallelism requirements.

Formal models for representing *real-time workloads* have traditionally been designed for the modeling of processes that are expected to execute in uniprocessor environments. As real-time application systems increasingly come to be implemented upon multiprocessor environments, these same models have been used to model the multiprocessor task systems. However, these traditional models fail to capture some important characteristics of multiprocessor real-time systems; furthermore, they may impose additional restrictions ("additional" in the sense of being mandated by the limitations of the model rather than the inherent characteristics of the platform) upon system design and implementation.

One particular restriction that has been extended from uniprocessor models to multiprocessor ones is that each task may execute upon at most one processor at each instant in time. In other words, they do not allow task parallel execution. However, this is overly restrictive for many current multiprocessor platforms; to further exacerbate matters, this restriction is in fact one significant causal factor of much of the complexity of multiprocessor scheduling. Indeed, as Liu (1969) pointed out, *"the simple fact that a [job] can use only one processor even when several processors are free at the same time adds a surprising amount of difficulty to the scheduling of multiple processors."* Certainly, the next generation of embedded and real-time systems will demand parallel execution.

Recently, some researchers have studied extensions to the workload models traditionally used in real-time scheduling theory, to allow for the possibility that a single job may execute simultaneously on multiple processors. One of the more promising approaches in this respect has been the recent work of Lin et al. (2006a, 2006b, 2007a, 2007b, 2007c), that applies *Divisible Load Theory (DLT)* to multiprocessor real-time systems. The resulting theory is referred to as *Real-time Divisible Load Theory (RT-DLT).*

1.2 Research Problem and Motivations

Real-time Divisible Load Theory (RT-DLT) holds great promise for modeling an emergent class of massively parallel real-time workloads. ***However, the theory needs strong formal foundations before it can be widely used for the design and analysis of hard real-time safety-critical applications***. In this thesis, we address the general problem of obtaining such formal foundations, by generalizing and extending recent results and concepts from multiprocessor real-time scheduling theory. Within this general problem, here are some of the specific issues we address:

i. Prior research in RT-DLT has reported some apparently anomalous findings, in the sense that these findings are somewhat counter-intuitive when compared to results from "regular" (i.e., non-real-time) DLT. *What explains these (previously-identified) apparent anomalies in RT-DLT?*

ii. When the processors in a multiprocessor platform all become available at the same instant in time, the issue of scheduling a real-time divisible workload on such platforms is pretty well understood. However, the reality in many multiprocessor environments is that all the processors do not become available to a given workload at the same instant (perhaps because some of the processors are also being used for other purposes). *How does one extend RT-DLT to render it applicable to the scheduling of real-time workloads upon platforms in which all the processors are not made available simultaneously? Specifically we address two important problems:*

   - Given a divisible job $\tau_i = (a_i, \sigma_i, d_i)$ and varying processor ready-times $r_1, r_2, r_3, ...$ what is the minimum number of processors needed to meet a job's deadline?

   - Given a divisible job $\tau_i = (a_i, \sigma_i, d_i)$ and ***n*** (identical) processors with varying ready-times $r_1, r_2, ..., r_n$ upon which to execute it, what is the earliest time at which the job τ_i can complete execution?

1.3 Research Objectives

As stated above, the goal of this thesis is to develop strong formal foundations that enable the application of RT-DLT for the design and analysis of multiprocessor hard real-time systems. To achieve this goal, we must build theoretical foundations and accurate simulation environments for experimenting with, and explaining the behavior of, hard real-time DLT systems. Some of the specific objectives that we have identified as needing to be accomplished in order to achieve this goal are as follows:

i. To investigate the application of Divisible Load Theory (DLT) models to real-time workloads, in order to obtain a deep and detailed understanding of the behavior of such systems.

ii. To theoretically explain the apparent anomalies of Real-time Divisible Load Theory (RT-DLT).

iii. To extend RT-DLT so that they are able to handle cluster and workload models that are as general as possible. Specifically, we hope that these extensions will be applicable to platforms in which all processors do not become available simultaneously.

iv. To build efficient scheduling algorithms that will compute the exact minimum number of processors that must be assigned to a job in order to guarantee that it meets its deadline — on clusters in which all processors are not simultaneously available.

v. To develop efficient scheduling algorithms that minimize the completion time of a given divisible job upon a specified number of processors — on clusters in which all processors are not simultaneously available.

1.4 Scope of Research

In this thesis, we focus upon a particular formal model of real-time workloads that is very widely used in real-time and embedded systems design and implementation. In this model, it is assumed that there are certain basic units of work, known as *jobs* that need to be executed. Such jobs are generated by recurring processes known as periodic or sporadic tasks – each such task represents a piece of straight-line code embedded within a potentially infinite loop. This workload model is described in greater detail in Chapter 2.

There are several kinds of timing constraints considered in the real-time scheduling literature; in this thesis, we restrict our attention for the most part to just one of these kinds of constraints – meeting *deadlines* of jobs.

With respect to system resources, we will focus for the most part on *minimizing the number of processors* used. (Although other system resources, such as network bandwidth, energy, etc. are also important, optimization with respect to these resources does not lie within the scope of this thesis.)

Several different network topologies, such as stars, meshes, and trees, have been studied in DLT. We restrict our attention to the *single-level tree* topology, since this is one of the simpler models but nevertheless appears to contain most of the important issues that arise when DLT is extended to apply to real-time workloads.

1.5 Research Methodology

We conducted this research in six major phases, as shown in Figure 1.1. The six phases are: Literature Review, Analysis and Problem Formulations, Algorithms Design, Algorithms Implementation, Algorithms Evaluations and Documentation. Each of these phases will be described in greater detail in the following pages.

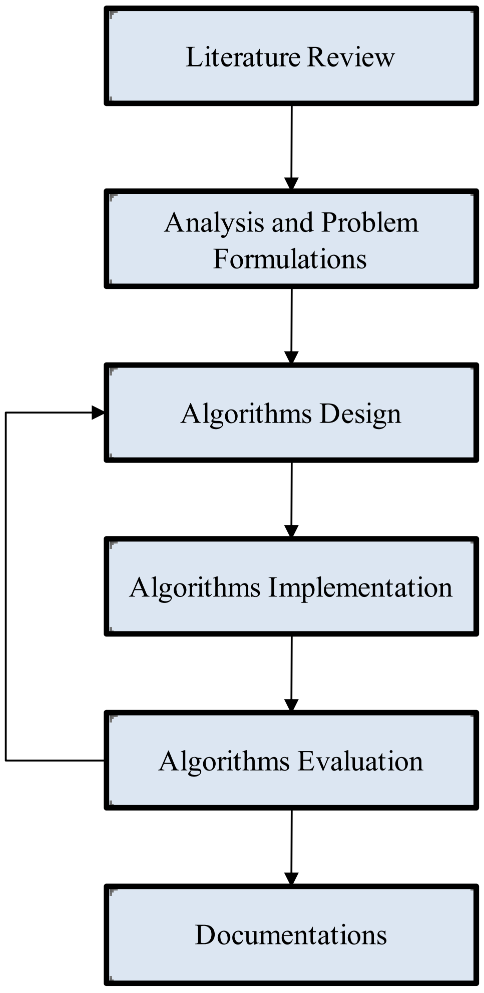

Figure 1.1 Conducted phases in this research

i. **<u>Literature Review</u>**

We performed literature review on various topics related to the research conducted in this thesis. The topic includes:

- State of the art of Real-time Systems
- State of the art of Real-time scheduling theory
- Current findings on Divisible Load Theory (DLT)
- Current findings on Real-time Divisible Load Theory (RT-DLT)

ii. **<u>Analysis and Problem Formulations</u>**

In this phase, we studied the applicability of DLT to multiprocessor scheduling of real-time systems. Specifically we analyzed series of work on RT-DLT (Lin et al., 2006a, 2006b, 2007a, 2007b, 2007c) and formulated three important problems arises upon these works. We explain these formulations in Chapter 3, 4 and 5 accordingly.

iii. **<u>Algorithms Design</u>**

As stated earlier, we formulated three significant problems detected from the work of Lin et al. (2006a, 2006b, 2007a, 2007b, and 2007c). For the first problem, we used existing scheduling theory to explain an anomalous observation of Lin et al. (2006a, 2006b, 2007a) when they first applied DLT to real-time multiprocessor scheduling. For the second problem, we designed an efficient algorithm to compute the minimum number of processors needed for a job to meet its deadline. To develop this algorithm, we used the first principle of RT-DLT found in Lin et al. (2006a, 2006b, and 2007a). And for the third problem, we formed a Linear Programming-based algorithm to compute the minimum completion time of a job execution. We present each detail design in Chapter 3, 4 and 5 respectively.

iv. **Algorithms Implementation**

In this phase, we developed series of simulations to compare the degree of improvement of our proposed algorithms to prior existing ones. For the second problem, we implemented the algorithm using C++ and for the third problem we developed the simulation programs using MATLAB.

v. **Algorithms Evaluation**

We evaluated our proposed algorithms by analyzing the results produced by our simulation programs. We compared the results produced by our algorithm with the ones produced by previous algorithms. In all comparisons, our algorithms showed significant improvement over pre-existing ones. We also provide lemmas and proofs to support our results and discussion in this thesis.

We conducted phase 3, 4 and 5 in three cycles for the three problems formulated.

vi. **Documentations**

Finally each contribution reported in this thesis was documented in technical publications. A list of papers published in the proceedings of conferences and journals are listed in Appendix A. The final and complete documentation is compiled in this thesis.

1.6 Thesis Organization

This thesis is organized into six chapters. Figure 1.2 shows the flow of the thesis organization; descriptions are given in the following pages.

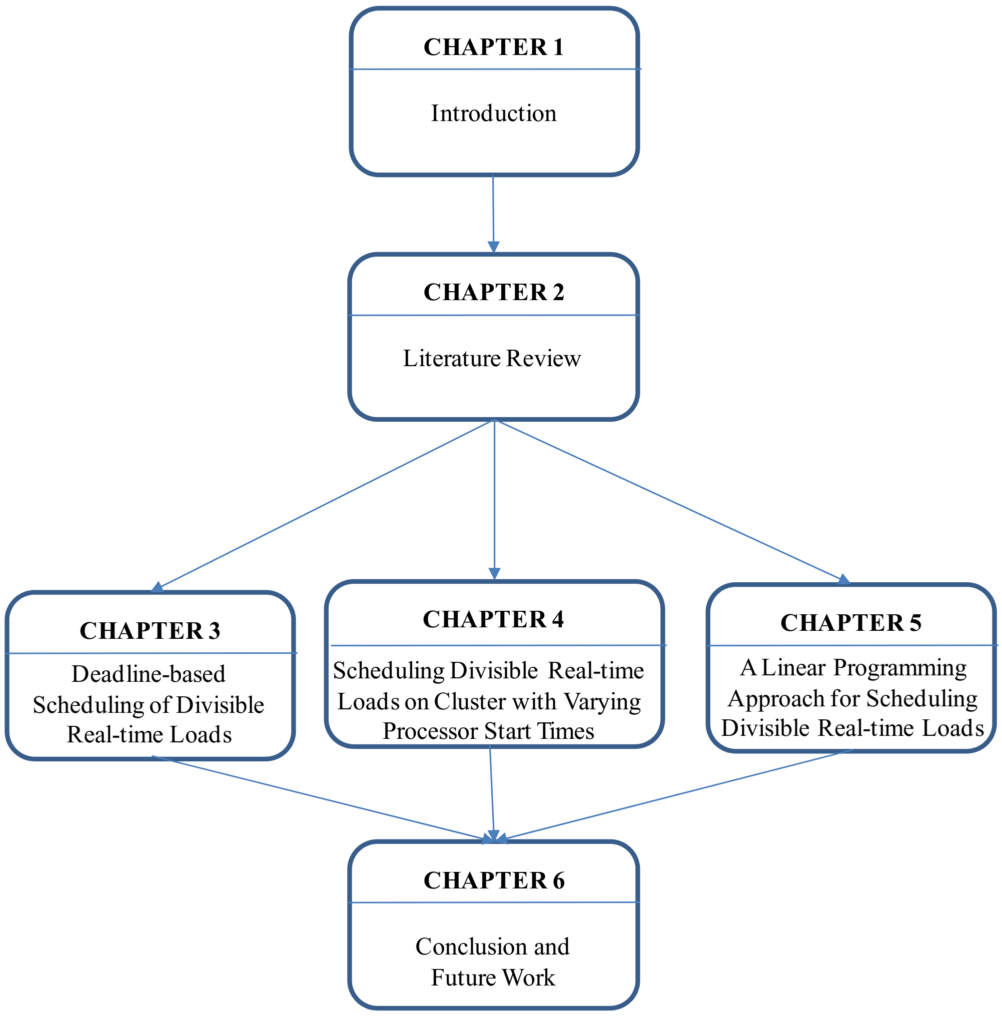
CHAPTER 1
Introduction
CHAPTER 2
Literature Review
CHAPTER 3
Deadline-based Scheduling of Divisible Real-time Loads
CHAPTER 4
Scheduling Divisible Real-time Loads on Cluster with Varying Processor Start Times
CHAPTER 5
A Linear Programming Approach for Scheduling Divisible Real-time Loads
CHAPTER 6
Conclusion and Future Work

Figure 1.2 Thesis organization

This thesis explores two important research areas: Real-time Systems and Divisible Load Theory. In Chapter 2, we present some background information and review some of the prior results on real-time systems. The first part describes the basic concepts of real-time systems. We then briefly review some fundamental results concerning real-time multiprocessor scheduling. The discussion mainly focuses on global multiprocessor scheduling with the Earliest Deadline First (EDF) scheduling algorithm. This chapter also discusses in greater detail the concept of Divisible Load Theory (DLT) and the application of this theory to multiprocessor scheduling of real-time systems, referred to as RT-DLT. We review some of the prior work done in RT-DLT, which we extend as part of this thesis.

In Chapter 3, we will report our first contribution presented in this thesis. We describe the initial work of Lin et al. (2006a, 2006b and 2007a) and their apparently anomalous findings with respect to a scheduling framework integrating DLT and EDF. We then present our results that provide a theoretical analysis to some of these anomalies.

In Chapter 4, we describe our study on scheduling problems in RT-DLT when applied to clusters in which different processors become available at different time-instants. We present an algorithm that efficiently determines the minimum number of processors that are required to meet a job's deadline. We then describe and discuss simulation results evaluating the proposed algorithm, and comparing it to previously-proposed heuristics for solving the same problem.

We have proposed a Linear Programming (LP) based approach to efficiently determine the earliest completion time for the job on a given processors which may become available at different times. This LP based approach is described in Chapter 5. We then present extensive experimental simulations to evaluate this LP based approach and consequently show how this approach significantly improves on the heuristic approximations that were the only techniques previously known for solving these problems.

Finally, we conclude our work and suggest directions for future research in Chapter 6.