

Doctoral Dissertation

Design and Analysis of Algorithms for
Graph Exploration and Resource
Allocation Problems and Their Application
to Energy Management

Naoyuki Morimoto

Graduate School of Informatics
Kyoto University

Abstract

In this dissertation, we discuss the design and analysis of algorithms for graph exploration and resource allocation problems, and their application to energy management.

Efficient use of energy has been a topic of great importance in various fields. For example, in the research fields on sensor networks and robotics, efficient energy control is decisive because available energy resources are generally limited. Therefore, there have been strong needs for energy-efficient control algorithms. Another example is a recent movement to construct new-generation power networks and power management systems, such as the evolving smart grid and frameworks of automated demand response. Especially, new power networks have been proposed based on the concept of Energy-on-Demand (EoD), where an energy management system automatically controls amounts of power supplied to appliances limiting total power consumption below a targeted value. There also have been novel concepts such as “optimal allocation of energy”, where a power network optimally allocates power from various kind of power sources to appliances considering Quality-of-Energy (QoEn), which represents characteristics of power that appliances require and sources supply. For realizing these concepts, we need efficient algorithms, devices and systems for power allocation.

The first topic of this dissertation is the design and analysis of efficient online algorithms for the graph exploration problems. Algorithms for exploration of unknown terrains have been actively studied, and much research work focuses on exploration algorithms for minimizing the total moving distance by a searcher to obtain all the topological information of the terrains. Kalyanasundaram and Pruhs formulated the problem in an online problem on undirected edge-weighted graphs. In Chapter 3, we will discuss the design and analysis of graph exploration algorithms for cycles and unweighted graphs, and give tight bounds on competitive ratios.

Second, the design and analysis of an approximation algorithm for resource allocation problems applicable to QoEn-based power allocation are described in Chapter

4. The multiple knapsack problem with assignment restrictions (MKAR) is an extension of the multiple knapsack problem, where each item (appliance) has a subset of knapsacks (power sources) from which the item is able to be supplied. We consider two variants of the MKAR depending on relationship between capacity of sources and power consumption of appliances, and give an approximation algorithm and an inapproximability result.

The third topic is the design and implementation of devices for policy-based power management including optimal allocation of energy. In Chapter 5, we describe the design and implementation of a smart outlet, a device that measures detailed power consumption of each appliance, communicates with other devices, supplies power to appliances considering allocated amounts of power, and controls appliances in a policy-based manner utilizing collected information of user's behavior and environmental information. We describe the design and implementation of a power router with functions for both QoEn power routing in EoD power networks and policy-based control of power assignment between two inputs and two outputs in a circuit-switching manner, and present its application to a power routing system for policy-based Home-to-Vehicle/Vehicle-to-Home in Chapter 6. The power router outputs power to its relevant port like routers in communication networks.

Lastly, we discuss a power allocation management system to maximize user's Quality-of-Life, keeping total power consumption within a limited amount of available power, which is implemented utilizing the developed smart outlet and a power allocation algorithm. We can treat power allocation for energy-saving and reducing peak load as the knapsack problem, regarding that an appliance as an item, importance of the appliance for user's life as the profit of the item, power consumption of the appliance as the size of the item, a power source as a knapsack, an available power amount of the power source as capacity of the knapsack. In Chapter 7, we describe a power management system utilizing an algorithm for the knapsack problem implemented on the smart outlet. To the best of our knowledge, this is the first literature that describes the implemented system utilizing a knapsack algorithm for power allocation management.

Contents

Abstract	i
Contents	iii
Chapter 1	Introduction
1.1	Background 1
1.2	Outline of this Dissertation 4
Chapter 2	Preliminaries
2.1	Online Algorithms and Competitive Analysis 9
2.2	NP-Hard Problems and Approximation Algorithms 10
Chapter 3	The Online Graph Exploration Problem on Restricted Graphs
3.1	Introduction 13
3.2	Preliminaries 15
3.3	A Tight Bound on Cycles 16
3.3.1	A Lower Bound 16
3.3.2	An Upper Bound 19
3.4	A Tight Bound on Unweighted Graphs 28
3.4.1	An Upper Bound 28
3.4.2	A Lower Bound 29
3.5	Concluding Remarks 31
Chapter 4	Resource Allocation Problems on Bipartite Graphs with Assignment Restrictions
4.1	Introduction 35
4.2	Related Work 37

4.3	The Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints	37
4.3.1	Problem Formulation	37
4.3.2	Algorithm ALG	38
4.3.3	Analysis of the Approximation Ratio of ALG	39
4.3.4	Integrality Gap of the Linear Relaxation Used in ALG	44
4.4	Splittable Resource Allocation Problem with Assignment Restrictions	44
4.4.1	Problem Formulation	45
4.4.2	Inapproximability Result	45
4.5	Concluding Remarks	47
Chapter 5	The Design and Implementation of a Smart Outlet for Policy-based Power Management	49
5.1	Introduction	49
5.2	Related Work	51
5.3	Design and Implementation	52
5.3.1	Hardware Design and Implementation	54
5.3.2	Software Design and Implementation	58
5.3.3	Data Formats	60
5.4	Evaluation of Power Measurement Function	61
5.5	Experimental Implementation of Control Policies	62
5.5.1	Software-based Circuit Breaker	62
5.5.2	Function of Avoiding Overlaps of Inrush Currents	64
5.6	Deployment in Real-life Environments and Control of Appliances Using Various Sensor Information	66
5.6.1	Environmental Sensors and Motion Sensor	67
5.6.2	Communication Network	68
5.6.3	Server	68
5.6.4	Data Formats	68
5.6.5	Deployment of the Developed System in Real-life Environments	69
5.7	Concluding Remarks	75
Chapter 6	Policy-based Power Router with Power Sensors for	

	Optimizing Allocation of Energy	77
6.1	Introduction	77
6.2	Related Work	79
6.3	Design and Implementation	80
	6.3.1 Basic Concepts	80
	6.3.2 Requirements Definition	82
	6.3.3 Implementation	84
6.4	Application to QoEn Routing and Autonomous Local Control	90
	6.4.1 Global Optimization with QoEn Routing	90
	6.4.2 Local Policies for Autonomous Control	95
6.5	Concluding Remarks	96
Chapter 7	A Power Allocation Management System Using an AI- gorithm for the Knapsack Problem	99
7.1	Introduction	99
7.2	Related Work	100
7.3	Basic Concepts and Problem Formulation	101
	7.3.1 Basic Concepts	101
	7.3.2 Problem Formulation	101
7.4	System Implementation	103
	7.4.1 Hardware	103
	7.4.2 Network	104
	7.4.3 Data Formats	105
	7.4.4 Dynamic-programming Based Algorithm for the Knap- sack Problem	107
	7.4.5 Control Software	108
7.5	Experiments and Evaluations	109
	7.5.1 Case with Four Appliances	109
	7.5.2 Case with 20 Appliances	109
7.6	Concluding Remarks	111
Chapter 8	Conclusion	113
	References	119
	Acknowledgements	129

List of Publications by the Author	131
---	------------

List of Figures

3.1	Lower bound construction for cycles (I).	17
3.2	Lower bound construction for cycles (II).	18
3.3	Description of the algorithm.	20
3.4	Proof of Lemma 3.3.	21
3.5	Final step of DIST.	23
3.6	Case 2-(iii) and Case 2-(iv).	25
3.7	Case 3-(v).	27
3.8	Lower bound construction for unweighted graphs.	33
4.1	Case 1-(i): OPT'_j includes the matched item in the knapsack and k_1 is odd.	41
4.2	Case 1-(ii): OPT'_j includes the matched item in the knapsack and k_1 is even.	41
4.3	Case 2-(i): OPT'_j does not include the matched item in the knapsack and k_1 is odd.	42
4.4	Case 2-(ii): OPT'_j does not include the matched item in the knapsack and k_1 is even.	43
4.5	An example of the instance reduction from the maximum independent set problem to SRAAR.	46
5.1	Policy-based power management using smart outlets.	50
5.2	Outward appearance of the developed smart outlet.	56
5.3	Hardware block diagram of the smart outlet.	57
5.4	Left: sensor/control board, Right: relay board.	58
5.5	Interprocess communications on the application board.	59
5.6	Voltage and current waveforms of a fluorescent lamp with an inverter.	62
5.7	Software-based circuit breaker function.	63

5.8	Current waveform of two light bulbs (without avoiding overlaps between inrush currents).	65
5.9	Current waveform of two light bulbs (with avoiding overlaps between inrush currents).	66
5.10	Smart outlets with Wi-Fi interfaces and sensors.	67
5.11	Smart outlet network in a one-room apartment.	70
5.12	Smart outlet network in a multi-room apartment.	71
5.13	Visualization of power consumption data; one-day.	72
5.14	Visualization of power consumption data; one-hour.	72
5.15	Deployment and setting sensors in the one-room apartment.	73
6.1	Policy-based power control in an EoD power network using power routers; QoEn routing for global optimization and autonomous local control.	81
6.2	Developed power router.	85
6.3	Hardware block diagram.	87
6.4	Internal hardware.	88
6.5	Configuring power allocation and parameters of the software-based circuit breaker function via HTTP/CGI.	93
6.6	Establishing an end-to-end power route by using the power router and the EoDresv routing server.	94
6.7	Policy-based autonomous switching between a commercial power source and an EV by the power router.	96
7.1	Power allocation for each appliance utilizing the knapsack algorithm.	102
7.2	An outward appearance of the smart outlet.	105
7.3	System consisting of multiple outlets and a controller.	106
7.4	Implemented system with a single outlet.	109
7.5	Implemented system with five outlets and a controller.	110
7.6	Example of software execution for power allocation and control.	111

List of Tables

5.1	Requirements definition of measurement.	53
5.2	Requirements definition of control, communications and extendibility.	54
5.3	Specifications of the developed smart outlet.	55
6.1	Requirements definition of control, communication and extendibility.	83
6.2	Requirements definition of measurement.	83
6.3	Specifications.	86
7.1	Specifications of the smart outlet.	104
7.2	Specifications of the controller.	105

Chapter 1

Introduction

1.1 Background

Efficient use of energy has been a topic of great importance in various fields. For example, in the research fields on sensor networks and robotics [12][61], efficient control is decisive because available energy resources are generally limited. Therefore, there have been strong needs for energy-efficient control algorithms. Another example is the recent movement to construct new-generation power networks and power management systems, such as the evolving smart grid [92] and frameworks of automated demand response [85]. In conventional power networks, energy-saving activities require user's operation for controlling appliances. Though "energy use visualization", where a system collects and visualizes data on energy usage by users, has been reported to have certain effect on saving energy [91], still users should control appliances by hand-work and should feel troublesome to keep their energy-saving activities. Therefore, new-generation power networks have been proposed based on the concept of Energy-on-Demand (EoD) [56], where an energy management system automatically controls amounts of power supplied to appliances, while keeping total power consumption under some target value, based on management policies suited for user's lifestyles and patterns of energy usage for easier and more effective energy-saving. There also have been novel concepts such as "optimal allocation of energy", where a power network optimally allocates power from various kinds of power sources to appliances considering Quality-of-Energy (QoEn), which represents characteristics of power that appliances require and sources supply [72] [42]. Communication protocols for power routing have been proposed [54] to independently route power from multiple and different power sources to appliances based on QoEn in an end-to-end manner, assuming future power

networks with various kinds of distributed sources such as solar panels or Vehicle-to-Home (V2H). For realizing these concepts, we need efficient algorithms that allocate power to appliances keeping power consumption under the target value, new devices with functions for power measurement/power control/communication, and a power management system utilizing power allocation algorithms.

In this dissertation, we discuss the design and analysis of algorithms for graph exploration and resource allocation, and their application for energy management. First, the design and analysis of efficient online algorithms for the graph exploration problem are described. Second, we present the design and analysis of an approximation algorithm for resource allocation problems applicable to power allocation. Third topic is the design and implementation of devices called *smart outlet* and *power router* for policy-based power management including “optimal allocation of energy”. Then we discuss the power allocation management system for maximizing user’s Quality-of-Life (QoL) under the limitation of available power, which is implemented utilizing smart outlets and a power allocation algorithm.

Backgrounds for each topic of the dissertation are as follows; algorithms for exploration of unknown terrains have been actively studied, and much research work focuses on exploration algorithms for minimizing the total moving distance by a searcher [11][59][70]. Exploration problems are related to various applications including the energy-efficient exploration by mobile robots on unknown terrains, or efficient collection of graph information that change dynamically, such as topology estimation of smart micro grids [25]. The well-known problem formulation on visiting all the nodes in a given graph is traveling salesman problem (TSP) [50], whose objective is to find a route with minimum total cost that visits all the vertices of a graph and returns to the origin. Here the cost of an algorithm is defined as the total traversal distance. This problem is a well-known NP-hard problem, and there have been intensive studies such as heuristics and approximation algorithms. In TSP, all information on the graph is given to the algorithm *in advance*. However, in above mentioned applications, the terrain may be unknown until the searcher visits the place, and the searcher learns the local environment when it actually visits there. This kind of problem is known as the *exploration* or the *map construction* problem and there are several models. Kalyanasundaram and Pruhs [39] formulated the problem in an online problem on undirected edge-weighted graphs, and showed a 16-competitive algorithm. In Chapter 3, we discuss the design and analysis of graph exploration algorithms for cycles and unweighted graphs and improve previous results.

In Chapter 4, the design and analysis of an approximation algorithm for resource allocation problems applicable to power allocation problems are described. Power sources have various characteristics such as cost, stability and CO₂ emission. For example, power from fossil fuel plant is considered to be stable but relatively costly; power from in-home solar panels is low-cost but unstable because it depends on weather conditions. On the other hand, power consuming devices have also characteristics on quality of power that they require. For example, a desktop PC that requires stable power should be supplied from commercial sources. However, a laptop with a battery may accept power from solar panels, because it works with relatively unstable power. Therefore, it is desirable to match power quality of power sources and power consuming devices. The power allocation between appliances and sources can be modeled as a resource allocation problem on bipartite graphs. However, it is hard to obtain an optimal and practical algorithm because it includes the NP-hard knapsack problem [32] as a special case. Hence it is natural to pursue some efficient approximation algorithm [35][93]. Considering application to the QoEn-based power allocation described above, we focus on the multiple knapsack problem with assignment restrictions (MKAR), where each item (appliance) has a limited subset of knapsacks (power sources) from which the item is able to be supplied. There have been studies on heuristics [18] along with studies on approximation algorithms [19][66] for the MKAR. We consider two variants of the MKAR, design approximation algorithms, and analyze approximation ratios.

As introduced above, we need novel devices for automated power management to keep total power consumption under the target value. In order to minimize undesired effect on user's QoL that might be caused by setting a limit of available power, it should be proper to utilize data on user's pattern of life, behavior information and various sensor information obtained by motion sensors or environmental sensors. For realizing these ideas, we need a device that measures detailed power consumption of each appliance, communicates with other devices, and controls supply of power to appliances based on allocated amount of power using gathered information of user's behavior and environmental information. In Chapter 5, we describe the design and implementation of the device called smart outlet with these functions.

With various advanced technologies, a device called *power router* and communication protocols for realizing QoEn power routing have recently been proposed and implemented. Generally, the power router has multiple inputs, and outputs power to its relevant port like routers in communication networks. The implementation meth-

ods of existing power routers are classified as follows; circuit switching of AC 100V [82], Power-over-Ethernet based circuit switching [73], and packet switching based on power packetization [83]. A protocol for QoEn routing has been proposed and implemented applying technologies for Quality-of-Service (QoS) guarantees in computer networks [54]. One of the typical applications of power routing is Vehicle-to-Home (V2H), where an electric vehicle (EV) works not only as transportation but as a huge battery. For example, it is expected that an EV is charged in off-load period such as midnight, and it works as a power source on on-load period to reduce peak load. Assuming these applications, the design and implementation of a power router for both QoEn routing and autonomous local control, and its application to a power routing system for policy-based H2V/V2H are described in Chapter 6.

As discussed in Chapter 4, we can treat optimal power allocation as the knapsack problem, regarding that an appliance as an item, a power source as a knapsack, importance of an appliance as profit of an item, power consumption of an appliance as a size of an item, an available amount of power as capacity of a knapsack. There has been research work on the simulation-based or theoretical evaluations utilizing knapsack algorithms for power allocation assuming reducing peak load, flattening peak-load and supporting automated demand-response [49][67][75]. In Chapter 7, we describe experimental implementation of the power allocation management system utilizing an algorithm for the knapsack problem and the developed smart outlet. To the best of our knowledge, this is the first literature that describes the implemented system utilizing a knapsack algorithm for power allocation management.

1.2 Outline of this Dissertation

In Chapter 2, we will introduce online problems and competitive analysis of online algorithms, and NP-hard problems and approximation algorithms as preliminaries for theoretical analysis of algorithms.

Chapter 3 describes the design and theoretical analysis of graph exploration algorithms. Kalyanasundaram and Pruhs formulated the problem in an online problem on undirected edge-weighted graphs as follows: At the beginning, the searcher is at the starting node o , called the *origin*, and it knows the local information, namely, the labels of the nodes adjacent to o , and the weights of edges incident to o . When the searcher visits a new node v , then it learns the labels of nodes adjacent to v and the weights of edges incident to v . When the searcher moves from u to v along

the edge (u, v) , it costs the weight of (u, v) . The task of the searcher is to visit all the nodes and return to the origin with as small cost as possible. Asahiro et al. [8] proved that if the input graph are cycle, there is a 1.5-competitive online algorithm, while no online algorithm can be $(1.25 - \epsilon)$ -competitive for any positive constant ϵ . In this chapter, we give an optimal online algorithm for this problem; namely, we give a $\frac{1+\sqrt{3}}{2}$ ($\simeq 1.366$)-competitive algorithm, and prove that there is no $(\frac{1+\sqrt{3}}{2} - \epsilon)$ -competitive algorithm for any positive constant ϵ . We also consider the problem on unweighted graphs, and give a tight bound of 2; namely, we prove that algorithm DFS is 2-competitive and that no online algorithm can have the competitive ratio better than 2 (this lower bound holds even when graphs have planarity).

In Chapter 4, we consider the power allocation as resource allocation problems on bipartite graphs. Namely, we analyze two variants of the MKAR with different relationship between capacity of sources and power consumption of appliances, and give an approximation algorithm and an inapproximability result. We first consider the multiple knapsack problem with assignment restrictions with capacity restrictions (MKARCC), where the *capacity constraints* mean that capacity of any knapsack is at least k (k is a natural number) times of the size of the largest item assignable to the knapsack, motivated by a situation where we can assume that capacity of power sources are larger than the amount of power required by power consuming devices at certain ratios. We give a $(1 + \frac{2}{k+1} + \epsilon)$ -approximation algorithm and show $(1 + \frac{1}{k} - \epsilon)$ integrality gap of a linear programming relaxation used in the algorithm, where ϵ is an arbitrary small positive constant. Second, in contrast to the MKARCC, we consider another variant of the MKAR where size of items may exceed the capacity of assignable knapsacks, and multiple knapsacks are able to supply one item cooperatively. We call the problem as the splittable resource allocation problem with assignment restrictions (SRAAR), motivated by a situation where multiple small distributed power sources supply single power consuming device coordinately. We give an inapproximability result; namely, we prove that it is NP-hard to approximate SPAAR within the ratio of $n^{1-o(1)}$ even when all the items have same profit.

Chapter 5 describes the design, implementation and evaluation of a smart outlet, which is developed for realizing “optimal allocation of energy” in a research project “Integration Technology of Information, Communication and Energy (ICE-IT)” [68] based on *i*-Energy concept [57]. In the research project, a *policy-based power management system* [31] is proposed and implemented, where power control is done in a distributed and autonomous manner. Therefore, the smart outlet has been developed

as a device capable of autonomous control, not only used in a centralized system. It has also been designed to have a precise power measurement function, to have sufficient computational resources to execute various policy-based control, to have extendibility and practicalities in both hardware and software aspects, and to be deployed sufficiently safely in real-life environments. The smart outlet has also sufficient specifications to be implemented with functions of appliance recognition and graphical user interfaces with a display for future extension. It has popular communication media (Wi-Fi and Ethernet) and protocols for easy coordination with other devices (e.g. information terminals, servers or other smart outlets.) As examples of power management policies utilizing the smart outlets, we have implemented experimental policies for saving energy and improving QoL; a software-based circuit breaker function, a function for reducing stand-by power utilizing motion sensors, and automated appliance control using environmental sensors.

We have implemented a policy-based power router for optimizing allocation of energy, extending the hardware and software implementation of the smart outlet. The major characteristics of the developed power router are that it has power sensors for power measurement (current, voltage, real-time power consumption, integral power consumption), and is able to control relays autonomously in a circuit-switching manner depending on sensed data and various policies. We have confirmed that an end-to-end power route is able to be established using the developed power router, combined with a routing server utilizing a QoEn routing protocol proposed and implemented by Miyamoto et al. [60]. As experimental implementation of autonomous local control, we have also implemented a control policy on the router that, when the power from one source suddenly becomes unavailable, it autonomously switches power supply with appliances to the other available source. We also have constructed a policy-based H2V/V2H control system in an experimental environment. Chapter 6 presents the design and implementation of the power router and the policy-based H2V/V2H control system.

In Chapter 7, we discuss the design and implementation of a power allocation management system utilizing an algorithm for the knapsack problem. The objective is to maximize the total profit of selected items (appliances), keeping total power consumption under the threshold. We have adopted the well-known dynamic-programming based algorithm for the knapsack problem, and have implemented both a local system where the processing is completed in only single power outlet, and a global centralized optimization system where the processing is across multiple outlets and

a controller. We have also evaluated the time duration needed for computing the optimal allocation, communication among devices and control of relays.

Chapter 8 presents conclusions and future work.

Chapter 2

Preliminaries

This dissertation includes discussions on online algorithms for online problems, and approximation algorithms for computationally hard problems. In this chapter, we describe motivations and backgrounds, definitions of online algorithms and its competitive analysis, NP-hard problems and approximation algorithms, and their evaluation methods as preliminaries for design and theoretical analysis of algorithms in Chapter 3 and 4.

2.1 Online Algorithms and Competitive Analysis

There are many realistic situations where it is natural to consider that the whole information is not given in advance, and we should make our decisions based on only information obtained so far. For example, in the case of stock exchange, we should decide whether we buy or sell stocks using only historical information of stock prices or current market statuses, without knowledge of future price changes. In the field of computer architectures, paging algorithms have to decide which data should be transferred between fast memory and slow storage without knowing future data access. These kind of problems are generally called *online problems* should be considered in a suitable framework. Algorithms for online problems are called *online algorithms*, and online algorithms should work with partial information given so far, not using information provided in future. In contrast, *offline problems* are problems where the whole information is completely given *in advance*, and *offline algorithms* are algorithms for offline problems and work using all the information.

Because an online algorithm should work with limited information, its performance is necessarily not as good as an optimal offline algorithm. In the example of

stock exchange, an online algorithm may fail to buy or sell the optimal amount of proper stocks, while an optimal offline algorithm is always able to obtain the optimal amount of profit using all the information concerned with future price changes of each stock. However, even online algorithms cannot perform as well as optimal offline algorithms in principle, it is natural to pursue online algorithms with the best possible performance.

There have been several schemes for evaluating performance of online algorithms. Today's standard method is called *competitive analysis* [14], where we compare solutions obtained by an online algorithm and an optimal offline algorithm, and try to give performance guarantee of the online algorithm by analyzing a worst-case ratio between their solutions for arbitrary problem instances. Historically, competitive analysis was originally introduced by Sleator and Tarjan and applied to the list access problem [76].

Here we give formal descriptions of online algorithms and competitive analysis. An online algorithm receives a sequence of requests σ one by one, and has no knowledge on future requests. Serving requests requires the algorithm to pay cost, and the objective of the algorithm is to minimize (in the case of minimization problems such as the online graph exploration problem treated in Chapter 3) the total cost required to serve all the requests. In competitive analysis, an online algorithm ALG is compared to an optimal offline algorithm OPT , which knows whole request sequence σ in advance, and pays optimal minimum cost. Given a sequence σ , let $\text{ALG}(\sigma)$ and $\text{OPT}(\sigma)$ denote the costs incurred by ALG and OPT , respectively. Algorithm ALG is said to be α -competitive if $\text{ALG}(\sigma) \leq \alpha \cdot \text{OPT}(\sigma)$ holds for arbitrary sequence σ . An upper bound on the competitive ratio indicates worst-case performance guarantee of an algorithm for arbitrary problem instances, and a lower bound shows performance limitation of an algorithm.

2.2 NP-Hard Problems and Approximation Algorithms

Efficiency is one of the most important factors in designing algorithms and evaluating their performance. Generally, algorithms whose computational time is exponential in the size of an instance do not work efficiently and have poor scalability, because an even small increase of instance size causes a huge amount of computational time.

Algorithms are commonly regarded as efficient if they require only *polynomial time* in the size of an instance. Therefore, a computational problem is generally considered to be easy if it has a polynomial-time algorithm, and to be hard if it requires an exponential-time algorithm. *Computational complexity* is a theoretical measure of difficulty of computational problems, and problems have been categorized into various classes. Typical problem classes are P and NP; P is the set of decision problems that can be solved in polynomial-time by a deterministic Turing machine, and NP is the set of decision problems that can be solved in polynomial-time by a non-deterministic Turing machine.

Though intensive research work has been done on algorithms and computational complexity, polynomial-time algorithms have not been found for problems in *NP-hard*, which is a class of problems that are at least as hard as the hardest problems in NP. It is known that NP-hard problems have no polynomial-time algorithm under the assumption of $P \neq NP$, which is strongly believed to hold though it has not been proved yet [32].

Many optimization problems including the knapsack problem we discuss in Chapter 4 have been proved to be NP-hard, therefore are less likely to be solved optimally with polynomial-time algorithms. However, in realistic situations, we have to deal with the NP-hard problems and obtain some solutions in practical time, even if it is hard to acquire optimal solutions. *Approximation algorithms* [93] are algorithms that work in polynomial time, and give not necessarily optimal but approximate solutions. Performances of an approximation algorithm are commonly evaluated by using a *approximation ratio*, which is a ratio between solutions obtained by an approximation algorithm and an optimal algorithm, analyzed in a worst-case manner for arbitrary problem instances.

Here we give formal definitions of NP-hard problems, approximation algorithms and an approximation ratio. A problem H is called NP-hard if any problem L in NP, which is solved in polynomial time using an oracle for H. In the analysis of an approximation ratio, an approximation algorithm ALG is compared to an optimal algorithm OPT, which always obtains an optimal solution. Given an input σ , let $ALG(\sigma)$ and $OPT(\sigma)$ denote the costs incurred by ALG and OPT. Algorithm ALG is said to be α -approximation if $ALG(\sigma) \leq \alpha \cdot OPT(\sigma)$ holds for arbitrary input σ (in the case of maximization problems such as the resource allocation problems treated in Chapter 4). An upper bound on the competitive ratio means worst-case performance guarantee of an algorithm for arbitrary instances, and a lower bound means perfor-

mance limitation of an algorithm. There are NP-hard problems that have *polynomial time approximation scheme* (PTAS) or *fully polynomial time approximation scheme* (FPTAS), which allows us to obtain near-optimal $(1 + \epsilon)$ -approximation solutions in polynomial-time, where ϵ is an arbitrary positive constant.

Chapter 3

The Online Graph Exploration Problem on Restricted Graphs

3.1 Introduction

In the Traveling Salesperson Problem (TSP) [50], we are given a graph and non-negative weights (lengths) on edges. Our task is to find a tour visiting all the nodes and coming back to the starting node with minimum cost. The cost of a tour is the total length of the tour. This problem is a well-known NP-hard problem, and there have been intensive studies such as heuristics and approximation algorithms. Apparently, TSP has plenty of practical applications, which includes determining a pickup or delivery tour for delivery companies or minimizing the total movement cost of robot arms in LSI wiring. It also has been proposed to utilize TSP algorithms in sensor networks to charge sensor nodes efficiently [7].

In TSP, all information on the graph is given to the algorithm *in advance*. However, in some cases of real applications, the terrain may be unknown until the algorithm visits the place, and the algorithm learns the local environment when it actually visits there. For example, suppose that we wish to gather complete information of an unknown environment using a robot searcher with minimum usage of energy. At the beginning, the robot has no knowledge of the environment. It should decide where to visit next depending only on the partial information of the environment that it has gained through the exploration so far. Another example of real applications is Web crawling by search providers [89], where Web crawlers automatically browse the Web

through links among pages and make indexes used for search engines. In the context of smart grid, it has been pointed out that topology estimation is an important element for future micro grids in many respects, because it provides efficient power routing strategies on grids that dynamically change their topology including node connectivity or resource availability [25]. This kind of problem is known as the *exploration* or the *map construction* problem and there are several models. Kalyanasundaram and Pruhs formulated the problem in an online problem on undirected edge-weighted graphs as follows: At the beginning, the searcher is at the starting node o , called the *origin*, and it knows the local information, namely, the labels of the nodes adjacent to o , and the weights of edges incident to o . When the searcher visits a node v , then it learns the labels of nodes adjacent to v and the weights of edges incident to v . When the searcher moves from u to v along the edge (u, v) , it costs the weight of (u, v) . The task of the searcher is to visit all the nodes and return to the origin with as small cost as possible. The goodness of the algorithm is evaluated by the *competitive analysis* [14, 26].

The most natural algorithm one may consider is the greedy type *Nearest Neighbor* algorithm (NN), which always visits a node nearest to the current node, among those that have not yet been visited. However, it has been shown that NN is not competitive even for planar graphs; there exists a planar graph G with n nodes such that the competitive ratio of NN is $\Omega(\log n)$ [71]. Kalyanasundaram and Pruhs [39] proposed a modified version of NN, called *ShortCut*, and proved that it is 16-competitive for planar graphs.

Note that if input graphs to be explored are restricted to trees, the depth-first search always returns an optimal tour because to visit all nodes and come back to the origin, each edge must be traversed at least twice, and the depth-first search traverses each edge exactly twice. Hence, the simplest non-trivial case is probably cycles. Recently, Asahiro et al. [8] considered the graph exploration on cycles. They proved that NN achieves the competitive ratio of 1.5 and showed that no online algorithm can have the competitive ratio better than 1.25.

Our Results. In this chapter we consider the problems on two classes of graphs and give tight bounds on the competitive ratio for both cases. First we improve both upper and lower bounds of the problem on cycles, and give a tight bound $\frac{1+\sqrt{3}}{2} (\simeq 1.366)$. For improving the upper bound, we propose a new algorithm called DIST, which decides the next node to visit depending on the (weighted) distances between the current node and each of the unvisited two nodes, the total length of

the exploration so far, and the distance from the origin to the current node. We also consider the problem on unweighted graphs, and give a tight bound of 2; namely, we prove that algorithm DFS is 2-competitive and that no online algorithm can have the competitive ratio better than 2 (this lower bound holds even when graphs have planarity).

Related Results. There are several variants of the problem of exploring unknown environment online. Deng and Papadimitriou [21] considered the problem of exploring a directed unweighted graph. This problem requires us to explore not only all nodes but also all edges, and the cost of the searcher is measured by the total number of edges traversed. They gave an online algorithm with $d^{O(d)}m$ edge traversals, where m is the number of edges in the graph and d is the minimum number of edges that have to be added to make the graph Eulerian. Albers and Henzinger [6] presented an algorithm that achieves an upper bound of $d^{O(\log d)}m$, and Fleischer and Trippen [29] gave an algorithm with an upper bound of $O(d^8m)$. Fleischer and Trippen [28] also made an experimental study of major online graph traversal algorithms and evaluated their practical performance on various graph families. In the polygon exploration problem (e.g. [20, 36]), an unknown environment is modeled by a polygon. The task of a searcher is to see all the boundaries of the polygon and come back to the starting point. Ausiello et al. [10] and Ausiello et al. [9] have studied the online traveling salesperson problem in which requests are presented online, and the aim of the searcher is to visit each requested point (not necessarily in the order of requests, unlike the k -server problem). The Canadian traveller problem [69] is an online extension of the shortest path problem where a traveller tries to move from an origin to a goal with minimum total distance of a given graph, however he knows whether an edge is available or blocked (e.g. not available) only when he visits either vertex of the edge.

3.2 Preliminaries

The purpose of the Online Graph Exploration problem is to visit all the nodes of a given graph $G = (V, E)$, where V and E denote the sets of nodes and edges, respectively. For each edge $(u, v) \in E$, a non-negative weight $\ell(u, v)$, sometimes called the *length*, is associated. Initially, the searcher is at the specified node $o \in V$, called the *origin*. It knows only the labels of the nodes adjacent to o , and the length of edges connecting o with those neighborhood nodes. Once the searcher visits a node v , it learns the labels of nodes adjacent to v and the length of edges incident to

v . The searcher has a sufficiently large memory so that it can store all information obtained so far, namely, the labels of nodes, the weights of edges, and the topology of the subgraph consisting of nodes and edges it has already learned. The task of the searcher is to determine the next node to visit, using only the current knowledge. The goal of the searcher is to visit all the nodes and return to the origin. The *cost* of the searcher for the graph G is the total length of the tour made by the searcher on G .

The performance of an online algorithms is evaluated by the *competitive analysis*: Let $\text{ALG}(G)$ denote the cost of an algorithm ALG on G , and let $\text{OPT}(G)$ denote the cost of an optimal *offline* algorithm OPT for G . We say that ALG is c -*competitive* for a class of graphs $\mathcal{G}$ if $\text{ALG}(G)/\text{OPT}(G) \leq c$ for any graph $G \in \mathcal{G}$. We may write ALG and OPT instead of $\text{ALG}(G)$ and $\text{OPT}(G)$, respectively, when G is clear.

3.3 A Tight Bound on Cycles

In this section we consider the problem on cycles and give a tight bound for the competitive ratio. Here is one simple but important fact [8]. Let $\ell_{\max}$ be the maximum length of edges and $L = \sum_{(u,v) \in E} \ell(u,v)$ be the sum of the length of all edges.

Fact 1 For any cycle C , $\text{OPT}(C) = L$ if $\ell_{\max} \leq \frac{L}{2}$, and $\text{OPT}(C) = 2(L - \ell_{\max})$ if $\ell_{\max} > \frac{L}{2}$.

3.3.1 A Lower Bound

In this section, we give a lower bound on the competitive ratio for any online algorithm.

Theorem 3.1 For any positive constant ϵ , there is no $(\frac{1+\sqrt{3}}{2} - \epsilon)$ -competitive online algorithm for cycles.

Proof. We will introduce an adversary giving the above mentioned lower bound. Fix an integer n and a constant μ such that $n > \frac{\sqrt{3}}{\epsilon}$ and $\mu < 1$. First, the adversary reveals two edges (o, u_1) and (o, v) incident to the origin with the equal length one. Without loss of generality, assume that the searcher moves to u_1 . Then, the adversary reveals an edge (u_1, u_2) such that $\ell(u_1, u_2) = 1$. If the searcher visits u_2 , then a new edge (u_2, u_3) with $\ell(u_2, u_3) = 1$ is revealed. Similarly, as long as the searcher visits a

new node u_i ($i \leq n-1$), the adversary gives an edge (u_i, u_{i+1}) with $\ell(u_i, u_{i+1}) = 1$.

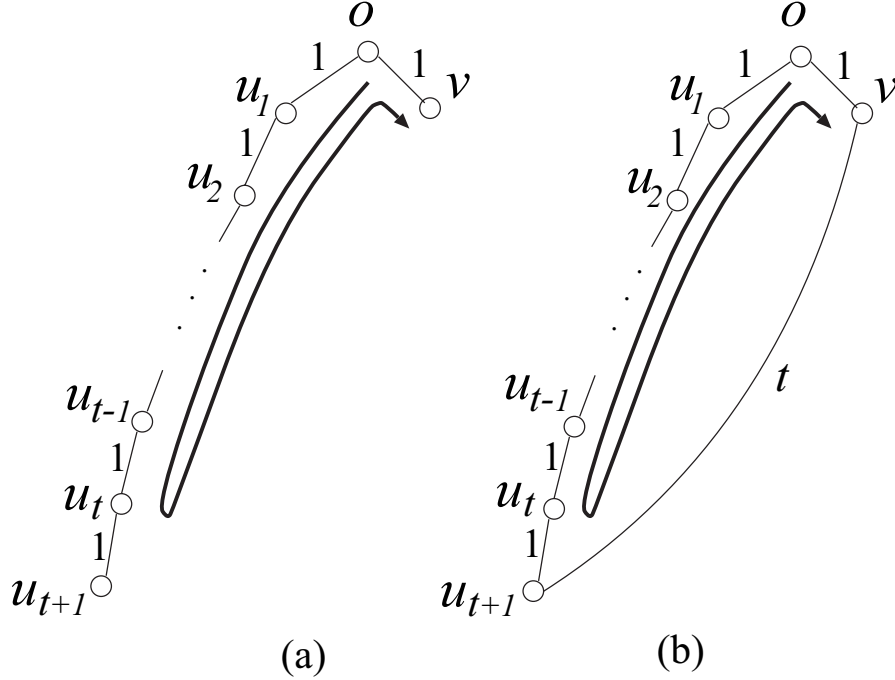


Figure 3.1 Lower bound construction for cycles (I).

Suppose that the searcher visits the node v before visiting u_n , and suppose that this happens just after it visited u_t where $t \leq n-1$ (i.e. it went back to v when it saw the edge (u_t, u_{t+1})) (Figure 3.1 (a)). Then the edge (v, u_{t+1}) with weight t is revealed (Figure 3.1 (b)). The only unvisited node is u_{t+1} , and the best way for the searcher is to go to u_{t+1} directly from v , and go back to the origin by either clockwise or counter-clockwise direction. The cost of the searcher is then $4t + 2$. The optimal tour is to visit all nodes along the cycle, whose cost is $2t + 2$. The competitive ratio in this case is then $(4t + 2)/(2t + 2) \geq 1.5$ since $t \geq 1$.

Next, suppose that the searcher visits u_n before visiting v . Then the adversary gives an edge (u_n, w) with length $\sqrt{3}n$ (Figure 3.2 (a)). We have two cases. First, suppose that the searcher visits w . Then the adversary reveals the edge (w, v) such that $\ell(w, v) = \mu$ (Figure 3.2 (b)). The best way for the searcher is to visit v and o in this order (note that $\mu < 1$). The cost of the searcher is then $n + \sqrt{3}n + \mu + 1 = (1 + \sqrt{3})n + \mu + 1$. Note that the edge (u_n, w) has the length more than half the total length of the whole cycle. Therefore, by Fact 1, the optimal cost is $2(n + \mu + 1)$. The competitive ratio is $\frac{(1+\sqrt{3})n+\mu+1}{2(n+\mu+1)} = \frac{1+\sqrt{3}}{2} - \frac{\sqrt{3}(\mu+1)}{2(n+\mu+1)} > \frac{1+\sqrt{3}}{2} - \frac{\sqrt{3}(\mu+1)}{2n} >$

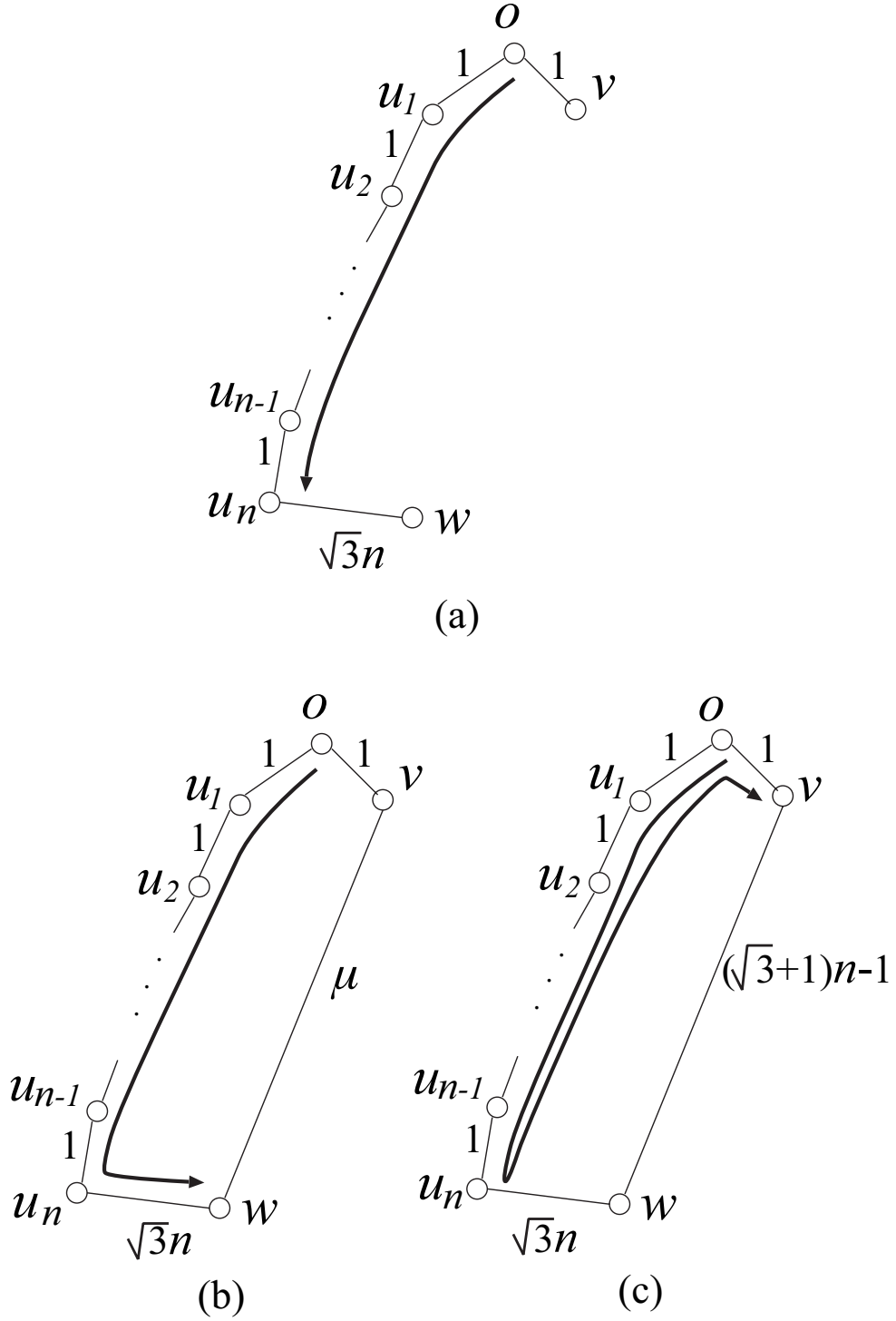


Figure 3.2 Lower bound construction for cycles (II).

$$\frac{1+\sqrt{3}}{2} - \frac{\sqrt{3}(1+1)}{2n} > \frac{1+\sqrt{3}}{2} - \epsilon.$$

Finally, suppose that after the edge (u_n, w) is revealed, the searcher goes back

to the node v . In this case, the adversary reveals the edge (v, w) with $\ell(v, w) = (\sqrt{3} + 1)n - 1$ (Figure 3.2 (c)). Then the only unvisited node is w , and the best way for the searcher is now to visit w directly from v , and then go back to the origin in either clockwise or counter-clockwise direction. The total cost of the tour is $n + n + 1 + (\sqrt{3} + 1)n - 1 + (\sqrt{3} + 1)n = (2\sqrt{3} + 4)n$. The optimal tour is a one along the cycle, whose cost is $(2\sqrt{3} + 2)n$. The competitive ratio in this case is $\frac{(2\sqrt{3}+4)n}{(2\sqrt{3}+2)n} = \frac{1+\sqrt{3}}{2}$. $\square$

3.3.2 An Upper Bound

In this section, we give an online algorithm **DIST** and analyze its competitive ratio.

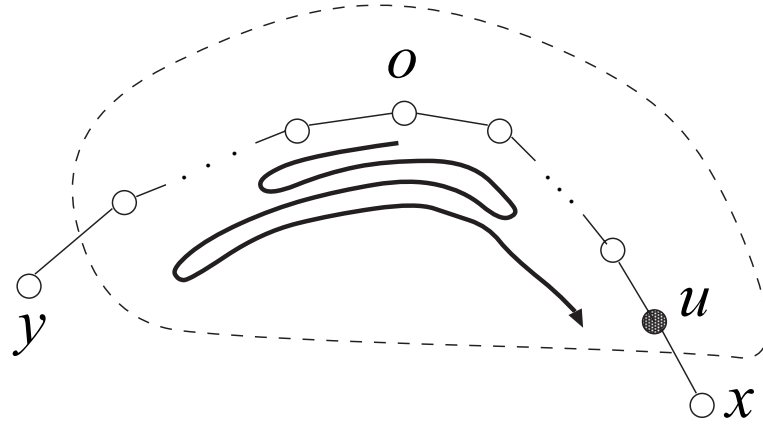
3.3.2.1 Algorithm DIST

Since a given graph is a cycle, there are always two choices for the searcher: (except for the 1st step), either to go forward or to go back. (See Figure 3.3 (a). The visited nodes are surrounded by a dotted curve, and the current position of the searcher is indicated by the black node.) Before presenting the algorithm, we give a few notations. Suppose that as shown in Figure 3.3 (a), the searcher is currently at the node u , and is to determine which of x and y to visit. For any two nodes v_1 and v_2 , let $d(v_1, v_2)$ denote the distance between v_1 and v_2 along the edges already known. Let X be the total length the searcher has traversed so far, and define $W = X - d(o, u)$. The value of W may change as time goes, so it might be appropriate to express it as e.g. W_i for Step i . However, for conciseness, we use W when there is no fear of confusion, or we sometimes say as “ W -value at this moment”. Now, we are ready to give our algorithm **DIST**:

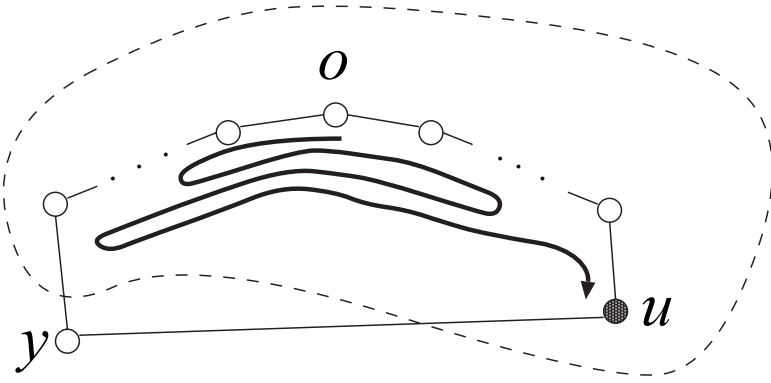
Step 1: The searcher is at the origin o , and there are two nodes adjacent to o . It moves to the node closer to o . If both are in the same distance, it chooses arbitrary one.

Step i ($i \geq 2$): Suppose that the searcher is at a node u as shown in Figure 3.3 (a). If $\ell(u, x) \leq \sqrt{3}d(u, y) - W$, then the searcher moves to x . Otherwise, i.e., if $\ell(u, x) > \sqrt{3}d(u, y) - W$, then the searcher moves to y .

Final step: The current situation is like Figure 3.3 (b). When the searcher visits a node u , it learns that u is connected to the unvisited but known node y (because it has seen y when it was on the node of the other side). Now, it knows the



(a)



(b)

Figure 3.3 Description of the algorithm.

entire graph, and there is only one unvisited node y . The searcher selects the shorter path from u to y , and then the shorter path from y to o .

3.3.2.2 Competitive Analysis

In this subsection, we prove the following theorem:

Theorem 3.2 DIST is $\frac{1+\sqrt{3}}{2}$ -competitive for cycles.

Proof. Consider any time step of the online game, and suppose that the situation is

like Figure 3.3 (a). (In the case just before the final step, x is equal to y as Figure 3.3 (b).) Let W be the current W -value, namely, the total distance the searcher has traversed so far minus $d(o, u)$. The following lemma is crucial in our analysis:

Lemma 3.3 For anytime before the final step, $W \leq (\sqrt{3} - 1)d(o, y)$.

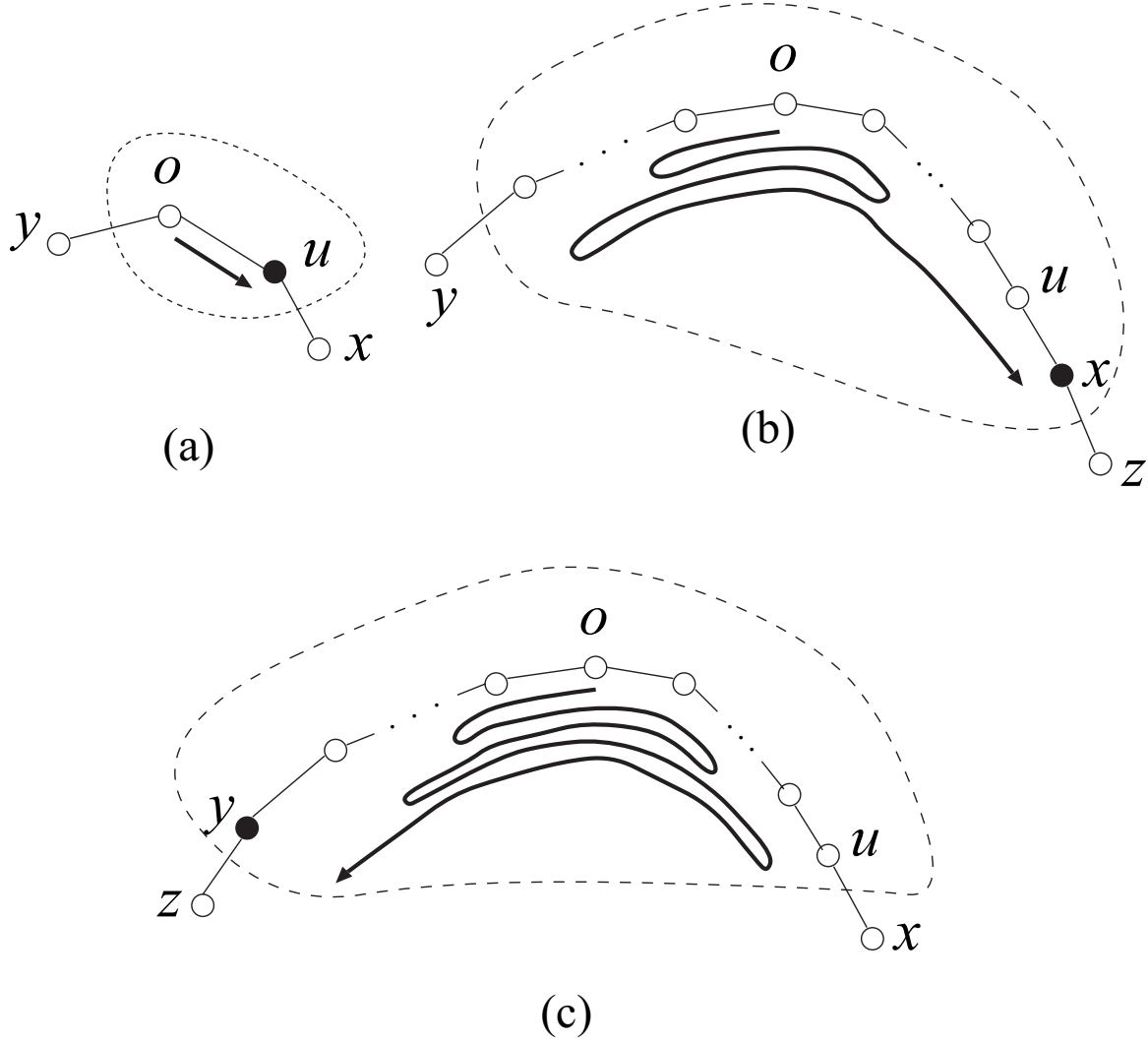


Figure 3.4 Proof of Lemma 3.3.

Note: In the case of Figure 3.3 (b), $d(o, u)$ is the distance from o to u in a clockwise direction, and $d(o, y)$ is the distance from o to y in a counter-clockwise direction.

Proof. The proof is by induction. After Step 1 is performed, the situation is like Figure 3.4 (a). Since $W = \ell(o, u) - \ell(o, u) = 0$, the inequality holds clearly.

Next, we assume that the inequality holds after Step i , and show that it holds after

Step $i + 1$. Suppose that after the execution of Step i , the situation is like Figure 3.3 (a). We denote the W -value at this moment by W_i . By the induction hypothesis, the following inequality holds: $W_i \leq (\sqrt{3} - 1)d(o, y)$. There are two cases to consider depending on whether the searcher moves to x or y in the next step.

Case 1. The searcher moves to x at Step $i + 1$. Then the situation is like Figure 3.4 (b). Note that by definition, the current W -value, denoted by W_{i+1} , is $W_{i+1} = W_i + \ell(u, x) - \ell(u, x) = W_i$. (This means that if the searcher goes forward, then the W -value remains unchanged. This property will be sometimes used hereafter.) Therefore $W_{i+1} \leq (\sqrt{3} - 1)d(o, y)$ by the induction hypothesis, which implies that the inequality holds after Step $i + 1$.

Case 2. The searcher moves to y at Step $i + 1$. Then the situation is like Figure 3.4 (c). The total length of the tour by the searcher increases by $d(y, o) + d(o, u)$ at this step. The distance from the origin was $d(o, u)$ but is now $d(o, y)$. Hence, $W_{i+1} = W_i + d(y, o) + d(o, u) - d(o, y) + d(o, u) = W_i + 2d(o, u)$. Since the searcher selected y rather than x , $\ell(u, x) > \sqrt{3}d(u, y) - W_i$ holds. Also, by the induction hypothesis, $W_i \leq (\sqrt{3} - 1)d(o, y)$. From these two inequalities and the equality $d(u, y) = d(u, o) + d(o, y)$, we have $d(o, y) < \ell(u, x) - \sqrt{3}d(u, o)$. Now using the hypothesis again, we have

$$\begin{aligned}
 W_{i+1} &= W_i + 2d(o, u) \\
 &\leq (\sqrt{3} - 1)d(o, y) + 2d(o, u) \\
 &< (\sqrt{3} - 1)(\ell(u, x) - \sqrt{3}d(u, o)) + 2d(o, u) \\
 &= (\sqrt{3} - 1)(\ell(u, x) + d(o, u)) \\
 &= (\sqrt{3} - 1)d(o, x)
 \end{aligned}$$

as required. □

Now, suppose that we are at the moment just before the final step. Then, the current situation looks like Figure 3.5. The searcher is at the node u , and it just learned that the node y adjacent to u is the same as the one it saw from v before. Since the searcher learned $\ell(u, y)$, it came to know the whole information on the cycle.

For simplicity, let a , b , c , and d denote the lengths of paths (edges) $d(o, u)$, $\ell(u, y)$, $d(o, v)$, and $\ell(v, y)$, respectively, as depicted in Figure 3.5. Let W^* be the W -value at this moment. Then, by Lemma 3.3, $W^* \leq (\sqrt{3} - 1)d(o, y) = (\sqrt{3} - 1)(c + d)$ holds. The only unvisited node is y , and the searcher will visit y in either clockwise or counter-clockwise direction depending on which route is shorter, and then go back

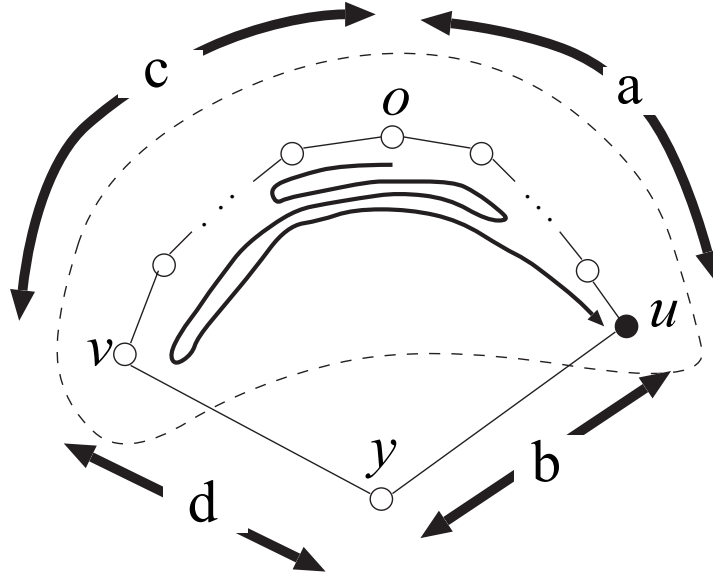


Figure 3.5 Final step of DIST.

to o from y in either clockwise or counter-clockwise direction, again depending on which route is shorter. We will do a case analysis.

Case 1. $b > a + c + d$. In this case, $a + b > c + d$ holds. Therefore, the searcher visits y by way of o , in a counter-clockwise direction, and goes back to o by way of v , in a clockwise direction. Since the cost of the searcher before the final step is $W^* + d(o, u) = W^* + a$ by the definition of the W -value, the final cost is $\text{DIST} = W^* + a + (a + c + d) + (c + d) = W^* + 2(a + c + d)$. Because $b > a + c + d$, $\ell_{\max} = b > L/2$. Therefore, the optimal cost is $\text{OPT} = 2(a + c + d)$ by Fact 1. The competitive ratio is

$$\begin{aligned} \frac{\text{DIST}}{\text{OPT}} &= \frac{W^* + 2(a + c + d)}{2(a + c + d)} \\ &\leq 1 + \frac{(\sqrt{3} - 1)(c + d)}{2(a + c + d)} \\ &\leq 1 + \frac{\sqrt{3} - 1}{2} \\ &= \frac{1 + \sqrt{3}}{2}. \end{aligned}$$

Case 2. $b \leq a + c + d$ and $a + b \leq c + d$. The searcher visits y using the edge (u, y) , and goes back to o in a counter-clockwise direction, i.e., by way of u . Therefore, $\text{DIST} = W^* + a + b + (b + a) = W^* + 2(a + b)$. Let $e_{\max}$ be an edge with maximum length, namely, $\ell(e_{\max}) = \ell_{\max}$. We consider subcases according to the length and

position of e_{max} .

Case 2-(i). $\ell_{max} \leq L/2$. By Fact 1, $\text{OPT} = a + b + c + d$. Thus,

$$\begin{aligned} \frac{\text{DIST}}{\text{OPT}} &= \frac{W^* + 2(a + b)}{a + b + c + d} \\ &\leq \frac{(\sqrt{3} - 1)(c + d) + 2(a + b)}{a + b + c + d} \\ &= \sqrt{3} - 1 + \frac{(3 - \sqrt{3})(a + b)}{a + b + c + d} \\ &\leq \sqrt{3} - 1 + \frac{(3 - \sqrt{3})(a + b)}{2(a + b)} \\ &= \frac{1 + \sqrt{3}}{2}. \end{aligned}$$

Case 2-(ii). $\ell_{max} > L/2$ and $e_{max} = (u, y)$. This does not happen because $b \leq a + c + d$.

Case 2-(iii). $\ell_{max} > L/2$ and $e_{max} = (v, y)$. By Fact 1, $\text{OPT} = 2(a + b + c)$. Consider the time when the searcher was at v (Figure 3.6 (a)), and let W' be the W -value at this moment. Let w be an unvisited node other than y (this notation is used sometimes hereafter). Then, by Lemma 3.3, $W' \leq (\sqrt{3} - 1)d(o, w) \leq (\sqrt{3} - 1)a$. Note that at the next step, the searcher moved to w because y is the last node visited by the searcher. Let W'' be the W -value just after the searcher moved to w . Then, the total length of the tour increased by $c + d(o, w)$, and the distance between the origin and the searcher changed from c to $d(o, w)$. Hence, by a simple calculation, $W'' = W' + c + d(o, w) + c - d(o, w) = W' + 2c$. Note that the searcher does not change the direction hereafter until it reaches u . Therefore, the W -value remains unchanged until the searcher reaches u , namely, $W^* = W'' = W' + 2c \leq (\sqrt{3} - 1)a + 2c$ (recall that W^* is the W -value when the searcher is at u). Now,

$$\begin{aligned} \frac{\text{DIST}}{\text{OPT}} &= \frac{W^* + 2(a + b)}{2(a + b + c)} \\ &\leq \frac{(\sqrt{3} - 1)a + 2c + 2(a + b)}{2(a + b + c)} \\ &= 1 + \frac{(\sqrt{3} - 1)a}{2(a + b + c)} \\ &\leq 1 + \frac{\sqrt{3} - 1}{2} \\ &= \frac{1 + \sqrt{3}}{2}. \end{aligned}$$

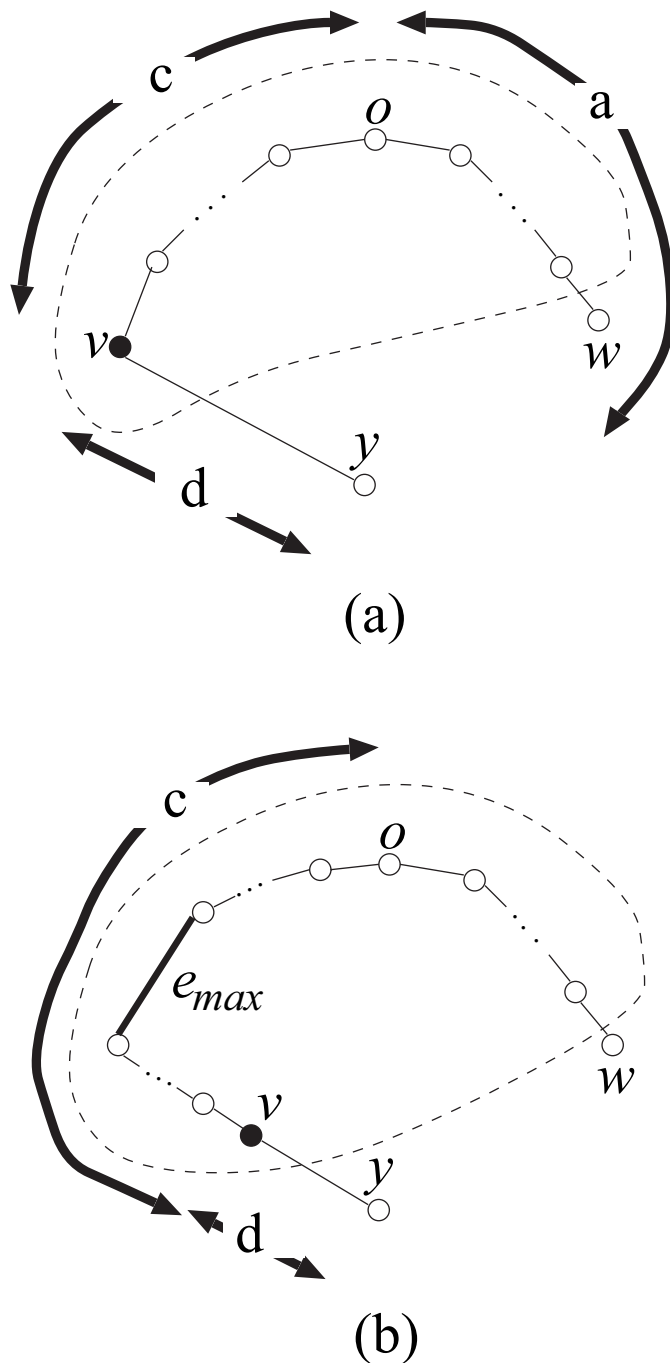


Figure 3.6 Case 2-(iii) and Case 2-(iv).

Case 2-(iv). $\ell_{max} > L/2$ and e_{max} is in the path from o to v (in a counter-clockwise direction). We can show that this case does not happen in the following way: Suppose, on the contrary, that this happens. Consider the time when the searcher was at v (Figure 3.6 (b)). Since the searcher decided to move to w rather than y , it must be the case that $\ell(v, y) > \sqrt{3}d(v, w) - W'$ where W' is the W -value

at this time. Also, by Lemma 3.3, $W' \leq (\sqrt{3} - 1)d(o, w)$. Therefore, $\ell(v, y) > \sqrt{3}d(v, w) - (\sqrt{3} - 1)d(o, w) = \sqrt{3}d(v, o) + d(o, w) \geq \ell_{max}$ because $d(v, o) \geq \ell_{max}$ by assumption. But this is a contradiction. Therefore, we can conclude that this case does not happen.

Case 2-(v). $\ell_{max} > L/2$ and e_{max} is in the path from o to u (in a clockwise direction). This does not happen because $a + b \leq c + d$.

Case 3. $b \leq a + c + d$ and $a + b > c + d$. The searcher visits y using the edge (u, y) , and goes back to o in a clockwise direction, i.e., by way of v . Therefore, $\text{DIST} = W^* + a + b + (c + d)$. Similarly, we consider the following subcases:

Case 3-(i). $\ell_{max} \leq L/2$. By Fact 1, $\text{OPT} = a + b + c + d$. Thus,

$$\begin{aligned} \frac{\text{DIST}}{\text{OPT}} &= \frac{W^* + a + b + c + d}{a + b + c + d} \\ &\leq 1 + \frac{(\sqrt{3} - 1)(c + d)}{a + b + c + d} \\ &< 1 + \frac{(\sqrt{3} - 1)(c + d)}{2(c + d)} \\ &= \frac{1 + \sqrt{3}}{2}. \end{aligned}$$

Case 3-(ii). $\ell_{max} > L/2$ and $e_{max} = (u, y)$. This does not happen because $b \leq a + c + d$.

Case 3-(iii). $\ell_{max} > L/2$ and $e_{max} = (v, y)$. This does not happen because $a + b > c + d$.

Case 3-(iv). $\ell_{max} > L/2$ and e_{max} is in the path from o to v (in a counter-clockwise direction). This does not happen because $a + b > c + d$.

Case 3-(v). $\ell_{max} > L/2$ and e_{max} is in the path from o to u (in a clockwise direction). Consider the time when the searcher was at v . We first show that the searcher had not yet traversed e_{max} at this time. On the contrary, suppose that it had already traversed e_{max} (Figure 3.7(a)). Since the searcher visits w at the next step, $\ell(v, y) > \sqrt{3}d(v, w) - W'$, where W' is the W -value at this moment. Also, by Lemma 3.3, $W' \leq (\sqrt{3} - 1)d(o, w)$. Hence, $\ell(v, y) > \sqrt{3}d(v, w) - (\sqrt{3} - 1)d(o, w) = \sqrt{3}d(o, v) + d(o, w) \geq \ell_{max}$, a contradiction. Therefore, the searcher traversed e_{max} for the first time after it left v .

Now, let $e_{max} = (u', u'')$ and consider the time when the searcher was at u' (Figure

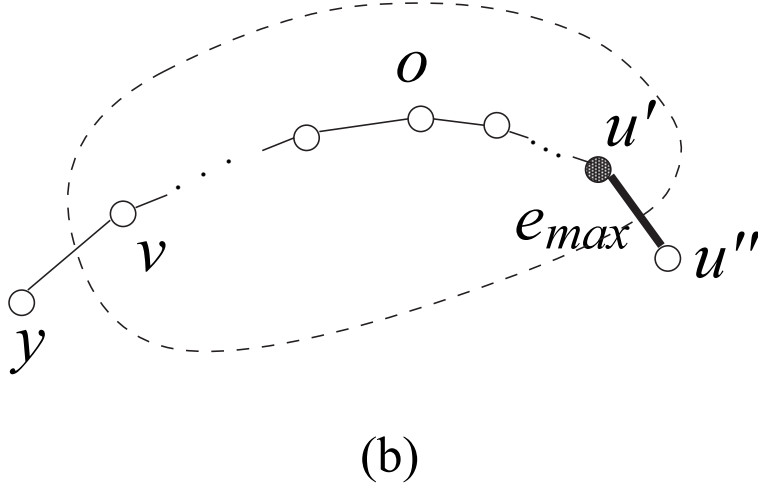
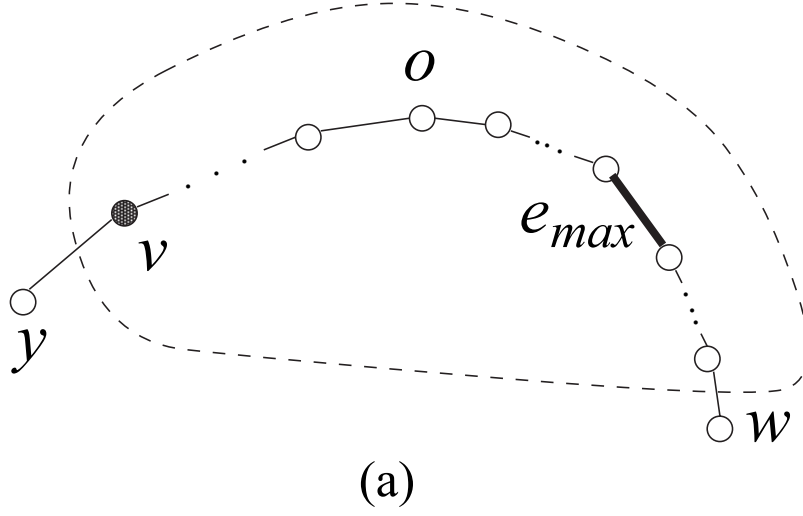


Figure 3.7 Case 3-(v).

3.7 (b)). Let W'' be the W -value at this moment. Since the searcher visited u'' next,

$$\begin{aligned}
 \ell_{max} &\leq \sqrt{3}d(u', y) - W'' \\
 &= \sqrt{3}(d(o, u') + d(o, y)) - W'' \\
 &\leq \sqrt{3}(d(o, u) - \ell_{max} + d(o, y)) - W''.
 \end{aligned}$$

The last inequality follows from the fact that $d(o, u') + \ell_{max} \leq d(o, u)$. From this

inequality,

$$\begin{aligned}
 \ell_{max} &\leq \frac{\sqrt{3}(d(o, u) + d(o, y)) - W''}{1 + \sqrt{3}} \\
 &= \frac{\sqrt{3}(a + c + d) - W''}{1 + \sqrt{3}} \\
 &= \frac{\sqrt{3}(L - b) - W''}{1 + \sqrt{3}}.
 \end{aligned}$$

Here, recall that L is the total length of the cycle. Because y is the last node visited by the searcher, the searcher does not change the direction hereafter, until it gets u . Hence $W^* = W''$ (recall that W^* is the W -value when the searcher is at u). By Fact 1, $\text{OPT} = 2(L - \ell_{max})$. Hence,

$$\begin{aligned}
 \frac{\text{DIST}}{\text{OPT}} &= \frac{W^* + a + b + c + d}{2(L - \ell_{max})} \\
 &\leq \frac{W^* + L}{2(L - \frac{\sqrt{3}(L-b)-W''}{1+\sqrt{3}})} \\
 &= \frac{(1 + \sqrt{3})(W^* + L)}{2(L + W^* + \sqrt{3}b)} \\
 &\leq \frac{1 + \sqrt{3}}{2}.
 \end{aligned}$$

□

3.4 A Tight Bound on Unweighted Graphs

In this section we consider the problem on graphs in which all edges have the same cost 1. Note that we do not restrict the topology of graphs.

3.4.1 An Upper Bound

The Depth-First Search (DFS) gives a good upper bound. When new edges and nodes are revealed, DFS chooses one of the unvisited nodes adjacent to the current node arbitrarily and visits it. If there is no such node, DFS backtracks, i.e., it goes back to the previous node through the edge used to come to the current node for the first time, and does the same procedure there.

To describe the behavior of DFS precisely, we give a recursive procedure **Search**. Inputs of **Search** are a node x and a sequence of nodes p (p could be empty). Intuitively,

First, the adversary reveals two edges (o, u_1) and (o, v_1) incident to the origin. If

the searcher moves to u_1 , a new edge (u_1, u_2) is revealed. As long as the searcher visits a new node u_i ($i \leq n-1$), the adversary gives an edge (u_i, u_{i+1}) . Similarly, if the searcher visits v_i ($i \leq n-1$), a new edge (v_i, v_{i+1}) is revealed. This procedure continues until the searcher reaches u_n or v_n . Without loss of generality we can assume he reaches u_n before v_n .

Now we assume that $v_1, v_2, \dots, v_{t_1}$ have been visited, and v_{t_1+1} has not been visited. Let D_a denote the total length of the exploration so far. Because the searcher visited v_{t_1} before reaching u_n ,

$$\begin{aligned} D_a &\geq 2\langle\langle o, v_1, \dots, v_{t_1} \rangle\rangle + \langle\langle o, u_1, \dots, u_n \rangle\rangle \\ &= n + 2t_1. \end{aligned}$$

Then, new edges (u_n, p_1) and (u_n, q_1) are revealed (Figure 3.8 (a)). When the searcher visits new node p_i ($i \leq n+t_1-1$), (p_i, p_{i+1}) will be revealed, and when the searcher visits q_i ($i \leq n+t_1-1$), (q_i, q_{i+1}) will be revealed (Figure 3.8 (b)). Hereafter, we will do a case analysis depending on the searcher's behavior.

First we consider the case that the searcher reaches v_{t_1+1} before visiting p_{n+t_1} or q_{n+t_1} . Let t_2 ($\leq n+t_1-1$) and t_3 ($\leq n+t_1-1$) be integers such that p_{t_3} and q_{t_2} have been visited, and p_{t_3+1} and q_{t_2+1} are unvisited (Figure 3.8 (c)). The adversary does not reveal new edges anymore. Let D_c denote the total length of the exploration so far. The searcher moved from u_n to v_{t_1+1} after visiting p_{t_3} and q_{t_2} , therefore

$$\begin{aligned} D_c &\geq D_a + 2\langle\langle u_n, p_1, p_2, \dots, p_{t_3} \rangle\rangle + \\ &\quad 2\langle\langle u_n, q_1, q_2, \dots, q_{t_2} \rangle\rangle + \\ &\quad \langle\langle u_n, u_{n-1}, \dots, u_1, o, v_1, \dots, v_{t_1+1} \rangle\rangle \\ &= D_a + 2t_2 + 2t_3 + n + t_1 + 1 \\ &\geq 2n + 3t_1 + 2t_2 + 2t_3 + 1. \end{aligned}$$

Hereafter, the searcher visits q_{t_2+1} and p_{t_3+1} , and returns to o finishing exploration. The total distance is

$$\begin{aligned} \text{ALG} &\geq D_c + \langle\langle v_{t_1+1}, v_{t_1}, \dots, o, u_1, \dots, u_n \rangle\rangle + \\ &\quad 2\langle\langle u_n, q_1, q_2, \dots, q_{t_2+1} \rangle\rangle + \\ &\quad 2\langle\langle u_n, p_1, p_2, \dots, p_{t_3+1} \rangle\rangle + \\ &\quad \langle\langle u_n, u_{n-1}, \dots, o \rangle\rangle \\ &= D_c + (t_1 + 1 + n) + 2(t_2 + 1) + \\ &\quad 2(t_3 + 1) + n \\ &\geq 4(n + t_1 + t_2 + t_3) + 6, \end{aligned}$$

while $\text{OPT} = 2(n + t_1 + t_2 + t_3 + 3)$. Therefore, $\frac{\text{ALG}}{\text{OPT}} > 2 - \epsilon$.

Secondly, we consider the case that the searcher reaches p_{n+t_1} or q_{n+t_1} before visiting v_{t_1+1} . Without loss of generality we can assume that the searcher reaches q_{n+t_1} . Now suppose that p_{t_4} has been visited and p_{t_4+1} is unvisited. Let D_d denote the total length of the exploration so far. By a similar observation as before,

$$\begin{aligned} D_d &\geq D_a + 2\langle\langle u_n, p_1, p_2, \dots, p_{t_4} \rangle\rangle + \\ &\quad \langle\langle u_n, q_1, q_2, \dots, q_{n+t_1} \rangle\rangle \\ &= D_a + 2t_4 + n + t_1 \\ &\geq 2n + 3t_1 + 2t_4. \end{aligned}$$

Then the adversary reveals (q_{n+t_1}, v_{t_1+1}) (Figure 3.8 (d)), and finishes revealing. Hereafter the best way for the searcher is to visit v_{t_1+1} traversing (q_{n+t_1}, v_{t_1+1}) , and to return to o after visiting p_{t_4+1} (the last unvisited node). The total length of the tour is

$$\begin{aligned} \text{ALG} &\geq D_d + \ell(q_{n+t_1}, v_{t_1+1}) + \\ &\quad \langle\langle v_{t_1+1}, q_{n+t_1}, \dots, u_n, p_1, \dots, p_{t_4+1} \rangle\rangle + \\ &\quad \langle\langle p_{t_4+1}, p_{t_4}, \dots, u_n, u_{n-1}, \dots, o \rangle\rangle \\ &= D_d + 1 + (1 + n + t_1 + t_4 + 1) + \\ &\quad (1 + t_4 + n) \\ &\geq 4(n + t_1 + t_4 + 1). \end{aligned}$$

The total length of an optimal offline tour is

$$\begin{aligned} \text{OPT} &= \langle\langle o, v_1, \dots, v_{t_1+1} \rangle\rangle + \ell(v_{t_1+1}, q_{n+t_1}) + \\ &\quad \langle\langle q_{n+t_1}, \dots, q_1, u_n, p_1, \dots, p_{t_4+1} \rangle\rangle + \\ &\quad \langle\langle p_{t_4+1}, \dots, p_1, u_n, \dots, u_1, o \rangle\rangle \\ &= (t_1 + 1) + 1 + (n + t_1 + t_4 + 1) + \\ &\quad (t_4 + 1 + n) \\ &= 2(n + t_1 + t_4 + 2). \end{aligned}$$

Therefore, $\frac{\text{ALG}}{\text{OPT}} > 2 - \epsilon$. □

3.5 Concluding Remarks

In this chapter, we have studied the online graph exploration problem on two graph classes. First, we have given a tight competitive ratio of $\frac{1+\sqrt{3}}{2}$ for the problem on

cycles. We have also studied the problem on unweighted graphs and have given a tight bound of 2.

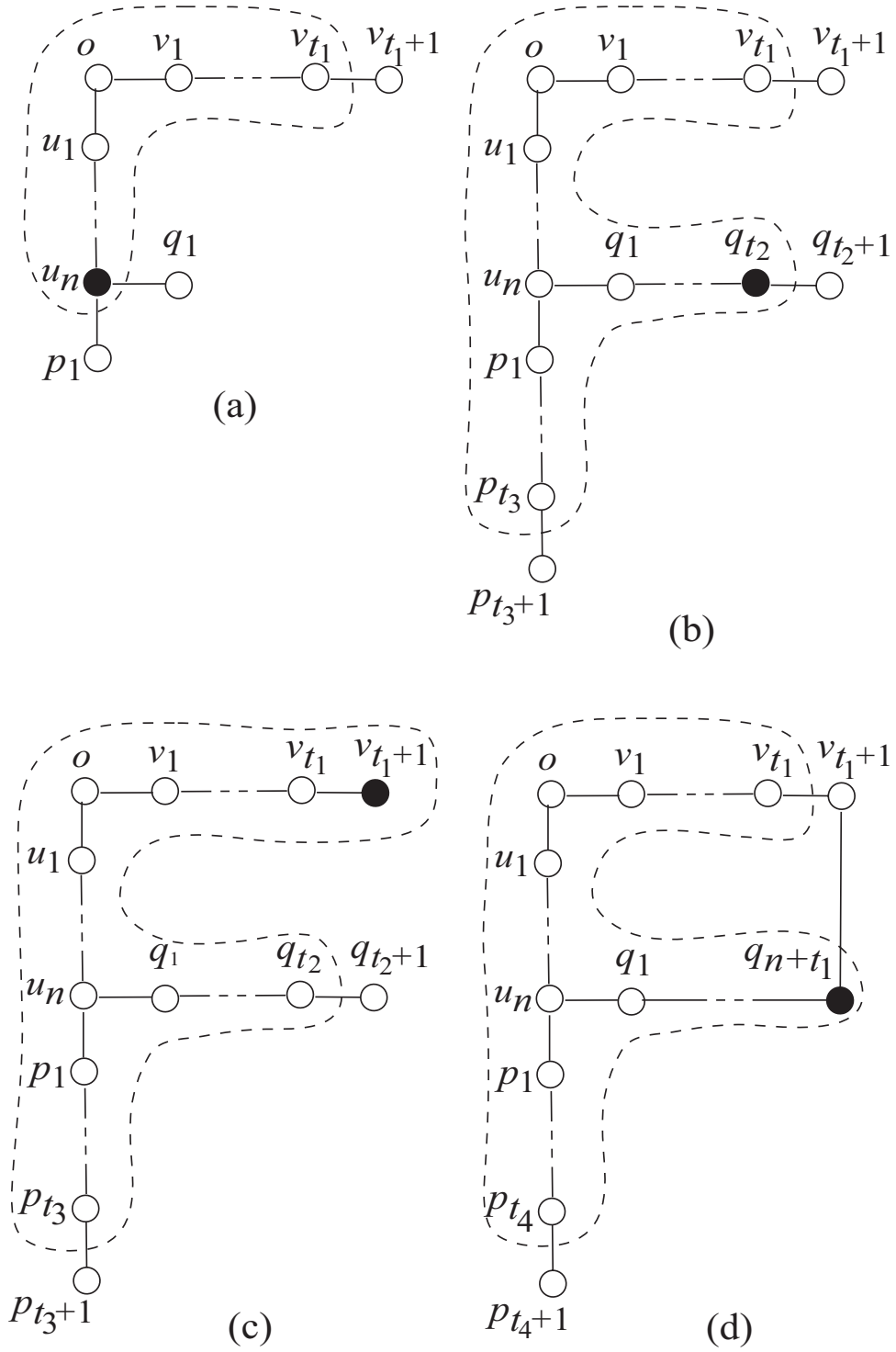


Figure 3.8 Lower bound construction for unweighted graphs.

Chapter 4

Resource Allocation Problems on Bipartite Graphs with Assignment Restrictions

4.1 Introduction

Generally, the resource allocation problem is an optimization problem where the objective is to decide an allocation of *resources* to *activities* that maximizes (or, minimizes) particular objective functions, given a set of resources and activities. In this chapter, we consider the resource allocation problem on bipartite graphs consisting of sets of clients (activities) and resources.

In recent years, efficient usage of natural power sources, such as solar power and wind power, has been studied actively with the spread of in-home power generations such as solar panels. These kinds of relatively small sources are generally called distributed power sources. Future power networks are supposed to include various distributed generations in addition to conventional commercial power sources such as fossil fuel plants and nuclear power plants, and it is strongly desired these power sources are utilized effectively. Power sources have various characteristics such as cost, stability and CO₂ emission. For example, power from fossil fuel plants is considered to be stable but relatively costly, while power from in-home solar panels is low-cost but unstable because it depends on weather conditions. Power consuming devices also have characteristics on quality of power that they require. For example, a desktop PC needs stable power, therefore it should be supplied from commercial sources. On the other hand, a laptop with a battery accept power from solar panels, because it

is able to work with the battery when the solar panels fails to generate stable power due to weather conditions. Therefore, it is desired to match power quality of power sources and power consuming devices in an appropriate manner.

Power allocation systems and power routing protocols considering Quality of Energy (QoEn) have been proposed recently [72][54]. When we consider optimizing power allocation as a mathematical problem, it is hard to obtain an optimal and practical algorithm, because it includes an NP-hard knapsack problem [32] as a special case. Therefore, it is natural to pursue some efficient approximation algorithm [35][93].

In this chapter, we consider the power allocation as resource allocation problems on bipartite graphs, and analyze two different models depending on relation between power capacity of sources and power consumption of devices, and give approximation algorithms and innapproximability results. The Multiple Knapsack Problem with Assignment Restrictions (MKAR), proposed by Dawande et al. [19], is an extension of the multiple knapsack problem where each item has a subset of knapsacks that is able to contain the item. First we treat the Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints (MKARCC), where the capacity constraints mean that any knapsack has at least k (k is a natural number) times of the largest item assignable to the knapsack, and give a $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm and $\left(1 + \frac{1}{k} - \epsilon\right)$ integrality gap of the LP we use in our algorithm where ϵ is an arbitrary small positive constant. The MKARCC models a situation where power sources have large capacity in some extent, compared with required power by a power consuming device. Second, in contrast to MKARCC, we consider the case where the size of items may exceed the capacity of knapsacks and items are able to be split and contained into multiple knapsacks. We call this problem the Splittable Resource Allocation Problem with Assignment Restrictions (SRAAR), and give an inapproximability result; namely, we show that it is NP-hard to obtain a polynomial time $n^{1-o(1)}$ -approximation polynomial algorithm even when all the items have the same profit. The SRAAR models a situation where distributed power generation with small capacity can coordinately supply power consuming devices whose requested amount of power may be larger than capacity of the power sources.

This chapter is organized as follows; Section 4.2 introduces related work. Section 4.3 shows a $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm for MKARCC(k). In Section 4.4 we prove $n^{1-o(1)}$ -approximation hardness for SRAAR. Section 4.5 concludes the

chapter.

4.2 Related Work

The MKAR is a special case of the generalized assignment problems (GAP). Approximation algorithms for the GAP and their variants have been studied actively. Cohen et al. showed a combinatorial translation of any algorithm for the single knapsack problem into an approximation algorithm for GAP, and showed a $(1 + \alpha)$ -approximation algorithm for GAP, where α is an approximation ratio for the single knapsack problem [16]. Shmoys and Tardos showed a 2-approximation algorithm for the minimization version of GAP [74]. Fleischer et al. derived an $\frac{e}{e-1}$ -approximation algorithm for GAP [27]. Dawande et al. showed a 2-approximation algorithm for the restricted case of MKAR, where the size of an item is equal to its profit [19]. Approximation algorithms for other restricted instances of MKAR have been studied as well [2][3]. Later, Nutov et al. [66] considered a 2-approximation algorithm for the general MKAR.

4.3 The Multiple Knapsack Problem with Assignment Restrictions and Capacity Constraints

4.3.1 Problem Formulation

Let I be the set of items and J be the set of knapsacks. For each item $i \in I$, the *profit* of i , denoted by p_i , and the *size* of i , denoted by ℓ_i , are associated. For each knapsack $j \in J$, its *capacity* c_j is associated.

We formulate an instance by a bipartite graph $G = (I, J, E)$, where $(i, j) \in E$ means that item i is assignable to knapsack j . Each edge $e = (i, j)$ has length $\ell_e = \ell_i$ and profit $p_e = p_i$. We define the density of e as $\pi_e = p_e/\ell_e$. For a vertex v of G , $\delta(v)$ denotes the set of edges that are incident on vertex v . We formulate MKAR as an integer program as follows;

$$\begin{aligned}
& \max \sum_{e \in E} \pi_e x_e \\
& \text{s.t. } \sum_{e \in \delta(j)} x_e \leq c_j, \forall j \in J \\
& \quad \sum_{e \in \delta(i)} x_e \leq \ell_i, \forall i \in I \\
& \quad x_e \in \{0, \ell_e\}, \forall e \in E
\end{aligned}$$

In MKARCC(k), any instance has to satisfy capacity constraints, namely $c_j \geq k\ell_{\max}(j), \forall j \in J$ (here $\ell_{\max}(j)$ means the largest size of an item assignable to knapsack j .)

4.3.2 Algorithm ALG

The following corollary by Nutov et al. is crucial for constructing and analyzing our algorithm. For a feasible solution x of the above LP, let $F(x)$ be the graph that consists of the set of fractional edges in x , namely, the set of edges e such that $0 < x_e < \ell_e$, and their endpoint vertices.

Corollary 4.1 (Nutov et al. [66]) Given a feasible solution x to the LP relaxation of MKAR, we can find in $O(|E(G)|^2)$ time a feasible solution z such that (i) $\pi \cdot z \geq \pi \cdot x$, (ii) $F(z)$ is a forest, and (iii) in any connected component of $F(z)$, at most one leaf belongs to I .

Our approximation algorithm **ALG** first obtains an optimal solution x^* of an LP relaxation of MKARCC(k), using a polynomial time algorithm for linear programming [63][40]. It then constructs a solution z from x^* using Corollary 4.1. In Step 3, for each knapsack C_j , we construct a single knapsack problem I_j consisting of knapsack C_j , *full* items, and at most one *matched* item. Full items are those assigned to C_j in x^* by an integral edge (i.e., an edge e such that $x_e = \ell_e$). A matched item is defined as follows. We construct a maximum cardinality matching M in $F(z)$ using the Hungarian method [48]. Then the matched item is the one matched with C_j in M if any. In Step 4, **ALG** obtains a near-optimal solution for each single knapsack problem I_j , using an FPTAS [55][45], and finally in Step 5, **ALG** outputs the union of selected items for each solution of I_j . A formal description of the algorithm is given

in Algorithm 1.

Algorithm 1 Algorithm ALG for MKARCC(k)

Step 1. Obtain an optimal solution x^* of a relaxed instance of MKARCC(k), using a polynomial time algorithm for linear programming.

Step 2. Construct a solution z from x^* as shown in Corollary 4.1.

Step 3. Construct instances of the single knapsack problem by matching fractional items in $F(z)$ to knapsacks.

Step 4. Using an FPTAS, obtain near-optimal solutions for all the instances of the single knapsack problem derived in Step 3.

Step 5. Output the total set of items selected in Step 4 as a solution for the original MKARCC(k) instance.

4.3.3 Analysis of the Approximation Ratio of ALG

Let OPT'_j denote an optimal solution for the instance I' of the single knapsack problem focusing a knapsack j and its adjacent items, which is derived in Step 3 of ALG, and let $\ell_{\text{OPT}'_j}$ denote the total size of all the items in OPT'_j . Let D denote the set of items not selected by OPT'_j . Let $\ell_{I''}$ and $P_{I''}$ denote the total size and profit of an item set I'' . Let SUM_j denote the total profit of a set of all the items adjacent to the knapsack j , namely an item set consisting of the single matched item and other full items, and ℓ_{SUM_j} denote the total size of all the items in SUM_j . Note that $\ell_{\text{SUM}_j} < c + \ell_{\max}$, because the total size of the full items is less than c_j since z is a feasible solution in the LP, and the size of the matched item is at most $\ell_{\max}$ by the capacity constraints.

First, we prove the following lemmas concerning the profit of items in OPT'_j .

Lemma 4.2 For any subset S of the items selected by the OPT'_j such that $\ell_S \geq \ell_D$, $P_S \geq P_D$.

Proof. If we remove all the items in S from the knapsack, we can include D instead of S . Hence, from the optimality of OPT'_j , we have $P_S \geq P_D$. $\square$

Lemma 4.3 When $|D| = 1$, for any subset S of items selected by the OPT'_j such that $\ell_S + (c - \ell_{\text{OPT}'_j}) \geq \ell_{\max}$, $P_S \geq P_D$.

Proof. If we remove all the items in S from the knapsack, we can include D instead of S since $\ell_D \leq \ell_{\max}$. Hence, from the optimality of OPT'_j , $P_S \geq P_D$. $\square$

Lemma 4.4 When $|D| \geq 2$, for any subset S of items selected by OPT'_j such that $\ell_S + (c - \ell_{\text{OPT}'_j}) \geq \ell_{\max}$, $P_S \geq \frac{1}{2}P_D$.

Proof. If we remove all the items in S from the knapsack, we can include $|D| - 1$ items except the item with the lowest profit in D , because the total size of those $|D| - 1$ items is at most $\ell_{\max}$. Hence, from the optimality of OPT'_j , $P_S \geq P_D$. $\square$

Lemma 4.5 For any set S of items selected by OPT'_j such that $\ell_S \geq k'\ell_{\max}$ (k' is a natural number), $P_S \geq \left\lceil \frac{k'}{2} \right\rceil P_D$.

Proof. Since the size of any item is at most $\ell_{\max}$, we can partition S into at least $\left\lceil \frac{\ell_S}{2\ell_{\max}} \right\rceil$ subsets with the size of at least $\ell_{\max}$ and less than $2\ell_{\max}$. By Lemma 4.2, each set has at least the profit of P_D . Hence, $P_S \geq \left\lceil \frac{k'\ell_{\max}}{2\ell_{\max}} \right\rceil P_D = \left\lceil \frac{k'}{2} \right\rceil P_D$. $\square$

The following lemma is crucial to the analysis.

Lemma 4.6 $\frac{\text{OPT}_{\text{LP}_j}}{\text{OPT}'_j} \leq 1 + \frac{2}{k+1}$.

Proof. For proving the lemma, it is sufficient to show $\text{OPT}'_j \geq \frac{k+1}{2}P_D$, because $\frac{\text{OPT}_{\text{LP}_j}}{\text{OPT}'_j} < \frac{\text{SUM}_j}{\text{OPT}'_j} \leq 1 + \frac{P_D}{\text{OPT}'_j}$.

We will do a case analysis depending on whether OPT'_j includes the matched item in the knapsack or not. Let M denote the matched item, and P_M and ℓ_M are the profit and size of M , respectively. Let k_1 denote a natural number such that $k \leq k_1 \leq \frac{c_j}{\ell_{\max}} < k_1 + 1$.

Case 1. OPT'_j includes the matched item in the knapsack.

Here $P_M \geq P_D$ holds, because otherwise OPT'_j included D in the knapsack instead of M . Hereafter we will do a case analysis depending on whether k_1 is odd or even.

Case 1-(i). k_1 is odd.

Let L denote a set of the items selected by OPT'_j other than M , as shown in Figure 4.1. Here $\ell_L \geq (k_1 - 2)\ell_{\max}$ holds, because $\ell_M + \ell_L > c_j - \ell_{\max} \geq (k_1 - 1)\ell_{\max}$ and $\ell_M \leq \ell_{\max}$. Therefore, by Lemma 4.5, $P_L \geq \left\lceil \frac{k_1-2}{2} \right\rceil P_D = \frac{k_1-1}{2}P_D$. Hence,

$$\begin{aligned} \text{OPT}'_j &= P_M + P_L \\ &\geq P_D + \frac{k_1-1}{2}P_D \\ &\geq \frac{k_1+1}{2}P_D \\ &\geq \frac{k+1}{2}P_D. \end{aligned}$$

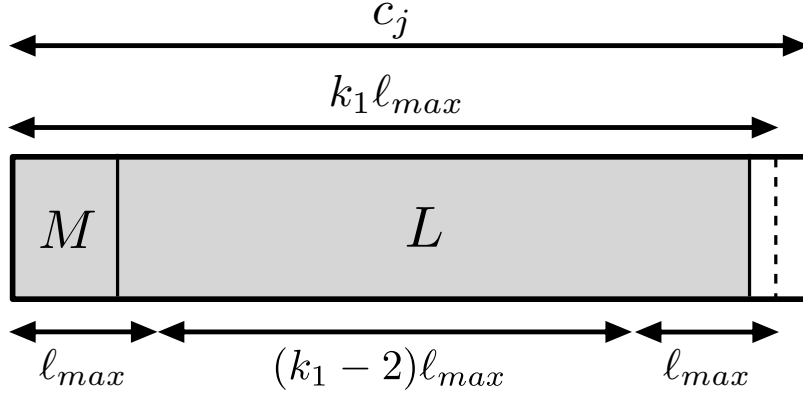


Figure 4.1 Case 1-(i): OPT'_j includes the matched item in the knapsack and k_1 is odd.

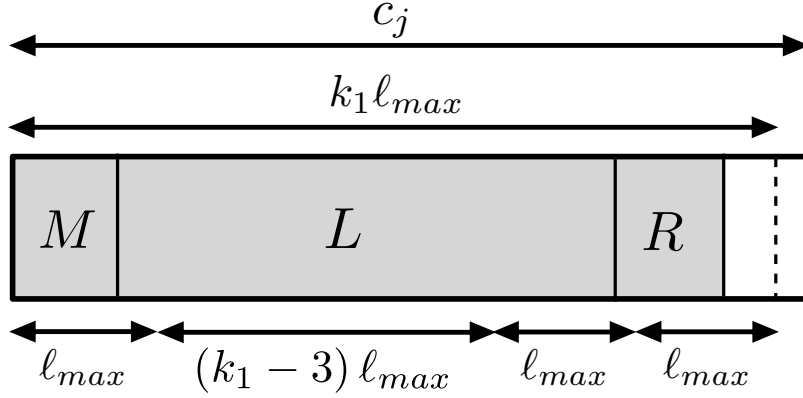


Figure 4.2 Case 1-(ii): OPT'_j includes the matched item in the knapsack and k_1 is even.

Case 1-(ii). k_1 is even.

We partition the items in OPT'_j into three parts of M , L and R , as shown in Figure 4.2. This partitioning is done in a way that satisfies $(k_1 - 2)\ell_{max} \leq \ell_M + \ell_L < (k_1 - 1)\ell_{max}$, and R is a set of the other items than M and L . Here $\ell_L \geq (k_1 - 3)\ell_{max}$ holds since $\ell_M \leq \ell_{max}$. By Lemma 4.5, $P_L \geq \lceil \frac{k_1 - 3}{2} \rceil P_D = (\frac{k_1}{2} - 1)P_D$. We also obtain $P_R \geq \frac{1}{2}P_D$ by Lemma 4.3 and Lemma 4.4, because $\ell_R + (c_j - \ell_{\text{OPT}'_j}) \geq \ell_{max}$.

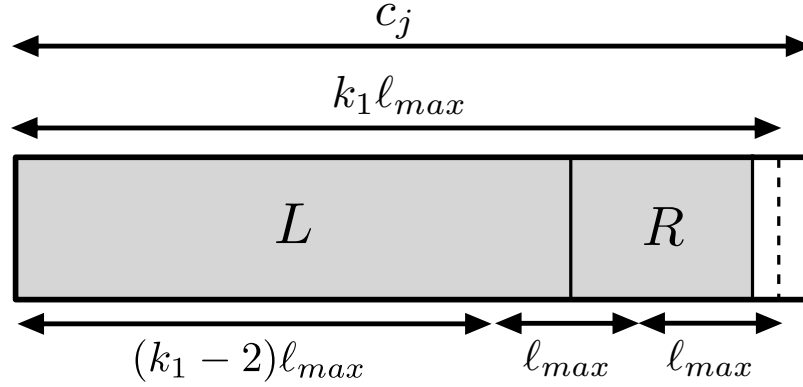


Figure 4.3 Case 2-(i): OPT'_j does not include the matched item in the knapsack and k_1 is odd.

Hence,

$$\begin{aligned}
 \text{OPT}'_j &= P_M + P_L + P_R \\
 &\geq P_D + \left(\frac{k_1}{2} - 1\right) P_D + \frac{1}{2} P_D \\
 &= \frac{k_1 + 1}{2} P_D \\
 &\geq \frac{k + 1}{2} P_D.
 \end{aligned}$$

Case 2. OPT'_j does not include the matched item in the knapsack.

In this case, the matched item is the sole item that OPT'_j discards from the knapsack, because all the full items can be included in the knapsack if the matched item is discarded. Hereafter, we will do a case analysis depending on whether k_1 is odd or even.

Case 2-(i). k_1 is odd.

We partition the items in OPT'_j into two parts of L and R , as shown in Figure 4.3. This partitioning is done in a way that satisfies $(k_1 - 2)\ell_{max} \leq \ell_L < (k_1 - 1)\ell_{max}$, and R is a set of the other items than L . By Lemma 4.5, $P_L \geq \lceil \frac{k_1 - 2}{2} \rceil P_D = \frac{k_1 - 1}{2} P_D$. We also obtain $P_R \geq P_D$ by Lemma 4.3, because $\ell_R + (c_j - \ell_{\text{OPT}'_j}) \geq \ell_{max}$. Hence,

$$\begin{aligned}
 \text{OPT}'_j &= P_L + P_R \\
 &\geq \frac{k_1 - 1}{2} P_D + P_D \\
 &= \frac{k_1 + 1}{2} P_D \\
 &\geq \frac{k + 1}{2} P_D.
 \end{aligned}$$

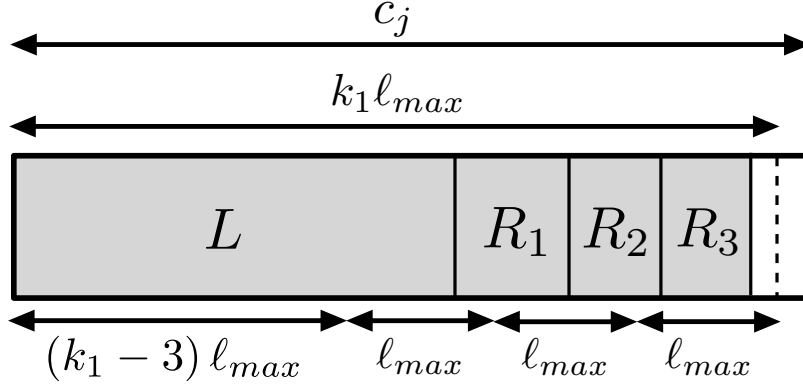


Figure 4.4 Case 2-(ii): OPT'_j does not include the matched item in the knapsack and k_1 is even.

Case 2-(ii). k_1 is even.

We partition the items of OPT'_j into four parts of L , R_1 , R_2 and R_3 , as shown in Figure 4.4. This partitioning is done in a way that satisfies $(k_1 - 3)\ell_{max} \leq \ell_L < (k_1 - 2)\ell_{max}$, $(k_1 - 2)\ell_{max} \leq \ell_L + \ell_{R_1} < (k_1 - 1)\ell_{max}$, $(k_1 - 1)\ell_{max} \leq \ell_L + \ell_{R_1} + \ell_{R_2}$ and $\ell_{R_1} \geq \ell_{R_2}$. By Lemma 4.5, $P_L \geq \lceil \frac{k_1 - 3}{2} \rceil P_D = (\frac{k_1}{2} - 1) P_D$. Since $\ell_{R_1} + \ell_{R_2} \geq \ell_{max} \geq \ell_D$, it follows that $P_{R_1} + P_{R_2} \geq P_D$ by Lemma 4.2. By Lemma 4.3, $P_{R_2} + P_{R_3} \geq P_D$ holds, because $\ell_{R_2} + \ell_{R_3} + (c_j - \ell_{\text{OPT}'_j}) \geq \ell_{max}$. Similarly, by Lemma 4.3, $P_{R_3} + P_{R_1} \geq P_D$ holds because $\ell_{R_3} + \ell_{R_1} + (c_j - \ell_{\text{OPT}'_j}) \geq \ell_{R_3} + \ell_{R_2} + (c_j - \ell_{\text{OPT}'_j}) \geq \ell_{max}$. Therefore, it follows that $P_{R_1} + P_{R_2} + P_{R_3} \geq \frac{3}{2} P_D$. Hence,

$$\begin{aligned}
 \text{OPT}'_j &= P_L + P_{R_1} + P_{R_2} + P_{R_3} \\
 &\geq \left(\frac{k_1}{2} - 1 \right) P_D + \frac{3}{2} P_D \\
 &= \frac{k_1 + 1}{2} P_D \\
 &\geq \frac{k + 1}{2} P_D.
 \end{aligned}$$

□

Theorem 4.7 ALG is a $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm for MKARCC(k), where ϵ is an arbitrary positive constant.

Proof. Let ALG_j denote the total profit of items included in knapsack j by ALG. We define a positive constant ϵ' that satisfies $\epsilon' \leq \frac{\epsilon(k+1)}{k+3+\epsilon(k+1)}$. We can obtain an

approximate solution that satisfies $\text{ALG}_j \geq (1 - \epsilon')\text{OPT}'_j$ using an FPTAS for the knapsack problem in Step 4 of **ALG**. Hence,

$$\begin{aligned}
\frac{\text{OPT}}{\text{ALG}} &\leq \frac{\text{OPT}_{\text{LP}}}{\text{ALG}} \\
&\leq \max_j \left\{ \frac{\text{OPT}_{\text{LP}_j}}{\text{ALG}_j} \right\} \\
&\leq \max_j \left\{ \frac{\text{OPT}_{\text{LP}_j}}{(1 - \epsilon')\text{OPT}'_j} \right\} \\
&\leq \frac{1}{1 - \epsilon'} \left(1 + \frac{2}{k+1} \right) \\
&\leq 1 + \frac{2}{k+1} + \epsilon.
\end{aligned}$$

□

4.3.4 Integrality Gap of the Linear Relaxation Used in **ALG**

Theorem 4.8 The integrality gap of the linear relaxation of $\text{MKARCC}(k)$ used in **ALG** is $1 + \frac{1}{k} - \epsilon$ where ϵ is an arbitrary positive constant.

Proof. Let ϵ' denote a positive small constant that satisfies $\epsilon' \leq \epsilon$. We consider an instance of $\text{MKARCC}(k)$ including $k+1$ items with the profit of 1 and the size of 1, and one knapsack with the capacity of $k+1 - \epsilon'$. The optimal solution for the relaxation problem is $k+1 - \epsilon'$, while the optimal solution for the original $\text{MKARCC}(k)$ is k . Hence, the integrality gap is $1 + \frac{1}{k} - \frac{\epsilon'}{k} \geq 1 + \frac{1}{k} - \epsilon$. □

4.4 Splittable Resource Allocation Problem with Assignment Restrictions

In this section, we consider another resource allocation problem where the size of items may exceed the capacity of the knapsacks and items are able to be split and contained into multiple knapsacks, in contrast to $\text{MKARCC}(k)$.

4.4.1 Problem Formulation

Splittable Resource Allocation Problem with Assignment Restrictions

Instance: A set of items I , a set of knapsacks J and a bipartite graph $G = (I, J, E)$ with a set of edges E between I and J . A vertex $i \in I$ ($i = 1, \dots, n$) has the size ℓ_i and the profit p_i , and a vertex $j \in J$ has the capacity c_j .

Problem: To give non-negative weight on each $e \in E$. Here the total weight of edges incident on j should be at most c_j . If the total weight of edges adjacent to vertex i is at least ℓ_i , it is said that an item i is *satisfied*. The objective is to maximize the total profit obtained by satisfied items by giving weight to all the edges in E .

The proof of the hardness result is done by a reduction from the maximum independent set problem.

The Maximum Independent Set Problem

Instance: A graph $G = (V, E)$.

Problem: An independent set of $G = (V, E)$ is a subset of V which has no edge between an arbitrary pair of two vertices. The objective is to find an independent set of G that has maximum number of vertices.

4.4.2 Inapproximability Result

In this subsection, we will show an inapproximability result of SRAAR.

Theorem 4.9 There is no polynomial time algorithm for SRAAR whose approximation ratio is $n^{1-o(1)}$ under the assumption of $\text{NP} \not\subseteq \text{ZPTIME}(2^{O(\log n (\log \log n)^{3/2})})$, even when all the items have the same profit.

Proof. We will construct an instance Ins_2 of SRAAR from an arbitrary instance Ins_1 of the maximum independent set problem in the following way. $G_1 = (V, E_1)$ is a graph in Ins_1 (without loss of generality, we can assume that G_1 is a connected graph), and we define $V = \{v_1, \dots, v_n\}$. $G_2 = (I, J, E_2)$ is a graph in Ins_2 such that

- An item $i \in I$ corresponds to a vertex $v_i \in V$.
- An knapsack $j \in J$ such that $c_j = 1$, $(i, j) \in E_2$ and $(i', j) \in E_2$, corresponds

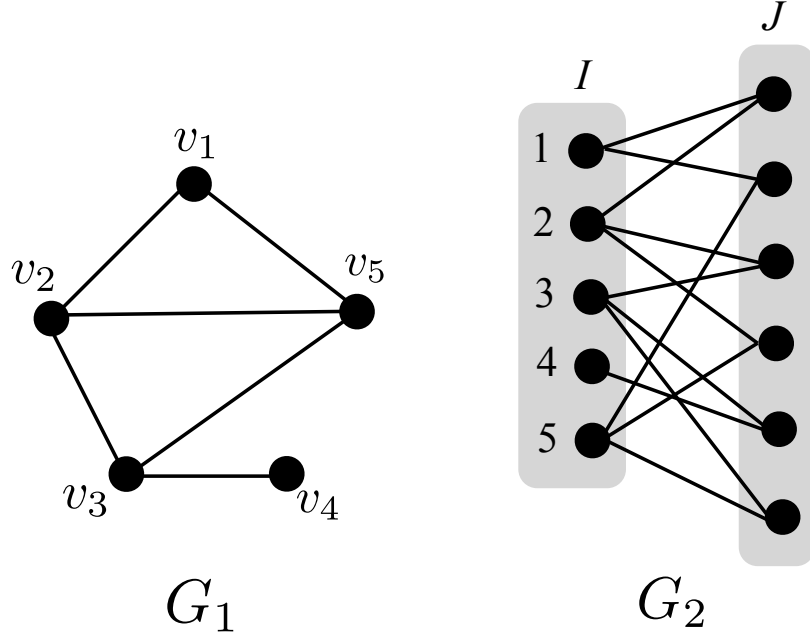


Figure 4.5 An example of the instance reduction from the maximum independent set problem to SRAAR.

to an edge $(v_i, v_{i'}) \in E_1$.

- The profit and size of an item i is $p_i = 1$ and $\ell_i = \deg(v_i)$ where $\deg(v_i)$ denotes a degree of v_i .

Figure 4.5 shows an example of the reduction.

In the above reduction, (i) if vertices v_i and $v_{i'}$ are in a solution V' for Ins_1 , both items i and i' can be satisfied in Ins_2 . Hence, $I' = \{i \mid v_i \in V'\}$ is a set of satisfied items in a solution for Ins_2 , and $|I'| = |V'|$ holds. On the other hand, (ii) if items i and i' are in a set of satisfied items I' in a solution for Ins_2 , vertices v_i and $v_{i'}$ are not adjacent in Ins_1 . Therefore, $V' = \{v_i \mid i \in I'\}$ is a solution for Ins_1 , and $|V'| = |I'|$ holds. From (i) and (ii), we obtain $\text{OPT}(\text{Ins}_1) = \text{OPT}(\text{Ins}_2)$ ($\text{OPT}(\text{Ins})$ denotes an optimal solution for an instance Ins .) Here we assume that there is a polynomial time r -approximation algorithm ALG_2 for SRAAR. Let I' denote a set of items that are satisfied by ALG_2 in Ins_2 . We consider an algorithm ALG_1 that choose $V' = \{v_i \mid i \in I'\}$ as a solution for Ins_1 . Since ALG_2 is an r -approximation algorithm and $|V'| = |I'|$, $\frac{\text{OPT}(\text{Ins}_1)}{|V'|} = \frac{\text{OPT}(\text{Ins}_2)}{|I'|} \leq r$. Therefore, ALG_1 is an r -approximation algorithm for the maximum independent set problem. However, it is known that there is no polynomial $n^{1-o(1)}$ -approximation algorithm for the maximum independent set

problem under the assumption of $\text{NP} \not\subseteq \text{ZPTIME}(2^{O(\log n (\log \log n)^{3/2})})$ [24]. Hence, we obtain the theorem. □

4.5 Concluding Remarks

In this chapter, we have considered two kinds of resource allocation problems on bipartite graphs with assignment restrictions, assuming power allocation problems on power networks with distributed sources. First, we have defined the multiple knapsack problem with assignment restrictions and capacity constraints (MKARCC), have designed the $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm, and have shown that the integrality gap of the LP relaxation used in the algorithm is $\left(1 + \frac{1}{k} - \epsilon\right)$ where ϵ is an arbitrary small positive constant. Second, we have defined the splittable resource allocation problem with assignment restrictions (SRAAR), and have obtained the $n^{1-o(1)}$ -approximation hardness result.

Chapter 5

The Design and Implementation of a Smart Outlet for Policy-based Power Management

5.1 Introduction

To cope with recent global warming and the expected shortage of fossil fuel in the future, it has been required to reduce CO₂ emission, and to realize efficient usage of energy and utilization of natural generations. In Japan, today's total energy consumption in home has increased and been more than twice as much as that in 1973, which is caused by change of lifestyle pursuing usefulness and comfort, and structural change of society such as increase of households [4]. The electricity shortage caused by the Great East-Japan Earthquake in 2011 also highlighted the importance of energy saving in home. Recently, various “energy use visualization” systems have been studied and put into practice, which make the power consumption by appliances visible to users, to help users to carry out energy-saving activities, and it has been reported to be effective [91]. The users still have some troublesomeness, however, even if the system suggests guidelines for energy-saving, because users still have to execute control by handwork.

Several novel in-home power network concepts have been proposed for effective power management with less handwork, where some power management system collects data of power consumption caused by users, the devices work autonomously and

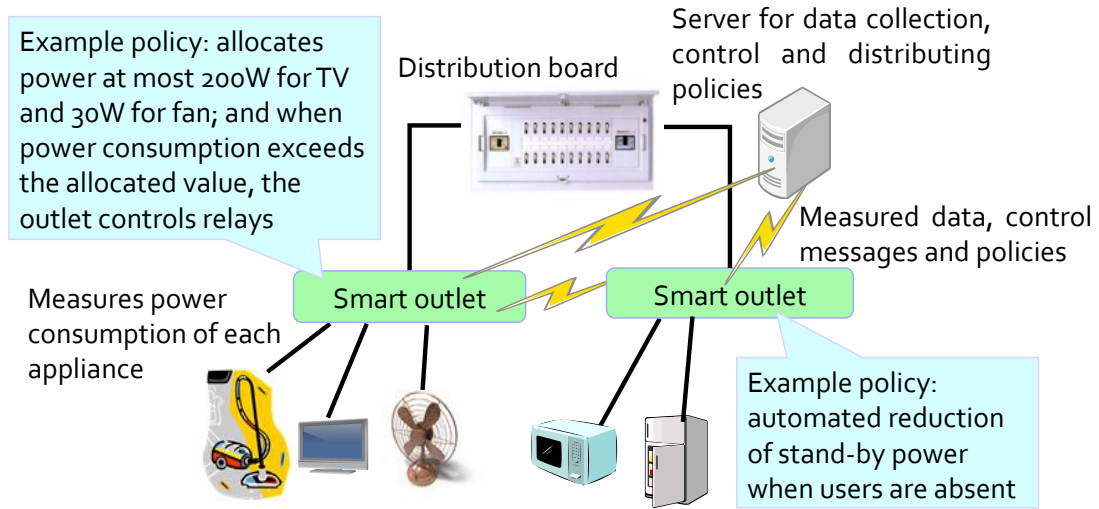


Figure 5.1 Policy-based power management using smart outlets.

cooperatively, and the system proactively optimizes power allocation for appliances based on policies suited for user's lifestyle and patterns of power usage. For example, in *i*-Energy concept [56], "optimal allocation of energy" has been proposed, where the power management system optimizes power allocation (e.g. up to 200W for a TV, 30W for a fan) considering user's QoL, and when total power consumption exceeds a threshold value the system automatically controls power allocation based on control policies suited for users, assuming situations that total usable power is limited in demand-response scheme or for reducing peak load. The system uses data of power consumption, the patterns of user's life, behavior information and environmental information to reduce power allocation to specific appliances. To realize these concepts, we need some device that measures power consumption for each appliance, communicates with other devices, and controls power supply based on user's behavior and environmental information.

In this chapter, the design, implementation and evaluation of a *smart outlet* are described, which is developed for realizing "optimal allocation of energy" in a research project "Integration Technology of Information, Communication and Energy (ICE-IT)" [68] based on the *i*-Energy concept. Figure 5.1 shows an overview of policy-based power management using the smart outlets that we assume in this research work. The smart outlet has been designed to have a precise power measurement function, to have sufficient computational resources for implementation of various power control policies, to have extendibility and practicality in both hardware and software aspects,

and to be deployed sufficiently safely in real-life environments. The smart outlet has sufficient specifications to be implemented with functions of appliance recognition and graphical user interfaces with a display for future extension. We have adopted popular communication media (Wi-Fi and Ethernet) and communication protocols for easy coordination with other devices (information terminals or servers, other smart outlets). As examples of power control policies utilizing the smart outlets, we have implemented several policies for energy saving and improving QoL; a software-based circuit breaker function, a function for reducing stand-by power using motion sensors, and automated control of appliances using environmental sensors.

This chapter is consisted as follows; Section 5.2 refers related work. The following Section 5.3 describes the design and implementation of the hardware and software of the smart outlet. Section 5.4 shows an evaluation of the power measurement function. Section 5.5 describes experimental implementation of power control policies; a software-based circuit breaker and a function for avoiding inrush current, which is followed by descriptions on real-life deployment of the smart outlets with power control policies utilizing various sensor information in Section 5.6. Section 5.7 presents concluding remarks.

5.2 Related Work

There has been research work on methods for finding waste of energy by using smart outlet networks without active power control [1][15][52]. For active power control, Han et al. proposed a system for reducing standby power of appliances. For making power consumption transparent for users, Akselrad et al. [5] proposed a method to utilize power consumption data gathered by a smart meter; since a smart meter is able to measure only a total power consumption of all appliances in a home, their proposed system attempts to decompose total power consumption data into individual data of each appliance using characteristics of appliances. Jiang et al. [38] constructed an IPv6/6LoWPAN-based mesh network that consists of power sensors for gathering power consumption data with high-fidelity. A system implemented by Song et al. [77] consists of smart outlets with ZigBee interfaces and IR nodes, and takes on-off control of appliances on detecting moving behavior of users. Another approach for policy-based power management is centralized ones; a centralized controller decides control of all of the smart outlets, and the outlets simply obey control messages from the controller; for example, there have been the EoD system proposed and implemented

by Kato et al. [43] and the system developed by Mrazovac et al. [62]. In particular, the former system keeps total power consumption under a limited value causing less undesired effect to user's QoL with priority-based control of appliances. "Green tap" [64] developed by NEC System Technologies measures power consumption and controls relays of each socket, and is able to be used in both centralized and distributed systems, with environmental sensor nodes with ZigBee-based communications.

Recently, there have been commercial products alike as smart outlets; WeMo [13] by Belkin is able to be controlled by outer devices via Wi-Fi, and is able to control relays in scheduled time. iRemotap [90] by Ubiquitous has a power measurement function along with on-off control capability from outer devices. It also sends power consumption data to cloud servers, and the cloud service works as an "energy use visualization" system, though it does not control relays automatically in a policy-based manner.

The following functions have been proposed as general functions of smart outlets [88];

- Measurement: Instantaneous current, instantaneous voltage, effective current, effective voltage, effective power, etc.
- Power control: On-off, current limiting, voltage control, etc.
- Communications: Communication with home servers or outer servers, other smart outlets, or smart appliances for sending power consumption data and receiving control messages.
- Other functions: User interfaces such as LED indicators, capabilities of logging power consumption data and sensor data (temperature, humidity, motion etc.).

5.3 Design and Implementation

The smart outlet in this research work is developed as a device for experimental use in a research project "Integration Technology of Information, Communication and Energy", not as a commercial product. A smart outlet by Yoshihisa et al. [96] has been adopted as a reference of our development, because it is implemented for policy-based power management utilizing power consumption data, temperature and motion information, and there has been an experimental environment for simulation of policy-based power management using their outlets. Therefore, the outlet is relevant to be a reference for requirements definition to implement our smart outlet. The smart outlet

Table 5.1 Requirements definition of measurement.

Items	Specs	Notes
Rated voltage	100V	
Rated current	15A	
Rated frequency	50/60Hz	For alternating current.
Measurement	Integral power, instantaneous power, voltage, current	For capturing transient phenomena.
Resolution	12bit	Equivalent with the standard power meters in Japan
Sampling frequency	20kHz	Sufficiently high for capturing transient phenomena.
Error	Under $\pm 2\%$	Automatic two-range switching. (0.5/15A)

by Yoshihisa et al. consists of an independent board for power measurement and power control (with ADE5169) and a Linux board called Armadillo-220 (CPU ARM920T 200MHz, 32MB RAM), and is shaped to be wall-mounted. It has two socket, and one of them has a power measurement function and has a relay for on-off control. It has an Ethernet port and two USB ports for communication and extendibility.

Since the smart outlet is expected to have sufficient specifications for future installation of advances functions such as appliance recognition using current waveform [37] [41] or user interfaces using a display and sounds, It is rather of importance to have sufficient extendibility and computational resources than to be less costly and small. Based on the smart outlet by Yoshihisa et al. introduced above, we have designed the outlet to be able to measure power more precisely, to control each socket independently, and to be used as practically and safely in experimental environments as the conventional power outlets.

Requirements definition for power measurement and for control/communication/extendibility is presented in Table 5.1 and Table 5.2, respectively. Rated voltage is 100V, rated current is 15A and frequency is 50/60Hz, designed to be used with ordinary appliances in Japan. Sampling frequency is designed to be

Table 5.2 Requirements definition of control, communications and extendibility.

Items	Specs	Notes
Power control	Relays	Software-based watching and control functions and control from outer devices.
Communication media	Wi-Fi	The most popular media for easy connection with other devices.
Extendibility	USB	For installing various sensors.

20kHz, based on a report that sampling frequency in A/D conversion needs to be sufficiently higher than 1-2kHz for recent appliances using inverters and switching power sources [88]. The resolution is 12bit based on the standard for power meters in Japan. To improve preciseness in current measurement, we have adopted the standard method of two-range measurement, because it is commonly known that the error tends to be large when we use a single range in A/D conversion, especially on measuring small amounts of current. This achieves $\pm 2\%$ of error rate that satisfies the standard for power meters in Japan. The outlet is designed to be able to control relays autonomously based on various policies assuming the system we presented in Figure 5.1, not only to be controlled from outer devices such as controllers. We have considered that it is realistic to restrict power control function to be simple on/off control, because so-called smart appliances still have not been sufficiently popularized; moreover, even if the outlet limits current supply or controls voltage, ordinary appliances do not necessarily work normally. The outlet has a Wi-Fi and Ethernet interfaces as communication media for general connectivity with other devices and for being used practically in experiments in real-life environments. The specifications of the developed outlets are presented in Table 5.3, and an outward appearance is presented in Figure 5.2. It is shaped to be like a common power strip for being practically used in ordinary environments. We will describe the detail of implementation of both hardware and software in the following sections.

5.3.1 Hardware Design and Implementation

To achieve both precise power measurement in the order of milliseconds and capability of applications such as communication with other devices and policy-based power

Table 5.3 Specifications of the developed smart outlet.

Items	Specs
Power sensing function	Real instantaneous power, integral power consumption, effective current and effective voltage
Sampling frequency	20kHz
Resolution	12bit
Power control	Policy-based on-off control
Error	Under +/- 2%
Number of sockets	Four
Rated voltage	AC 100V
Maximum current	15A (in total of all the sockets)
OS	Debian Linux
CPU	ARM926 400MHz (on the application board) TMS320F28035 60MHz (on the sensor/control board)
Memory and Storage	128MB RAM, 32MB FLASH
Extendibility	Two USB 2.0 ports microSD slot (for expanding internal storage)
Communication media	Wi-Fi, Ethernet
Communication protocols	TCP/IP socket communications, HTTP/CGI
Communication with outer devices	Among a home server, outer servers and other smart outlets
Miscellaneous	LED indicator (for indicating statuses of relays), data logging (using micro SD), motion and environmental sensors (via USB)



Figure 5.2 Outward appearance of the developed smart outlet.

management, we have adopt two individual boards in our design and implementation; one is for power measurement and control, and the other is for applications. The former is named as sensor/control board, and the latter as application board. We have adopted an ARM-based commercial board as the application board, which is capable of handling universal Linux OS and interfaces such as USB and Wi-Fi. Figure 5.3 shows a hardware block diagram. It consists of an application board, a sensor/control board, a relay board, a power source and current transformers. In the following section, we will describe the functions and implementation of the boards, and an evaluation of safety.

- **Application board:** The application board communicates the sensor/control board to gather power consumption data and to send relay control messages. It also communicates with outer devices and executes policy-based power management. As described above, compared with the smart outlet implemented by Yoshihisa et al., our smart outlet has been designed to have more computational resources for managing more outlets based on various policies, to be able to extended with some user interfaces (such as a display), and to be able to log power consumption data on inner storage. Therefore, we have adopt Armadillo-440 (CPU: ARM926 400MHz, Memory: 128MB RAM and 32MB FLASH), which is a commercial microcomputer upper-compatible with Armadillo-220. It has

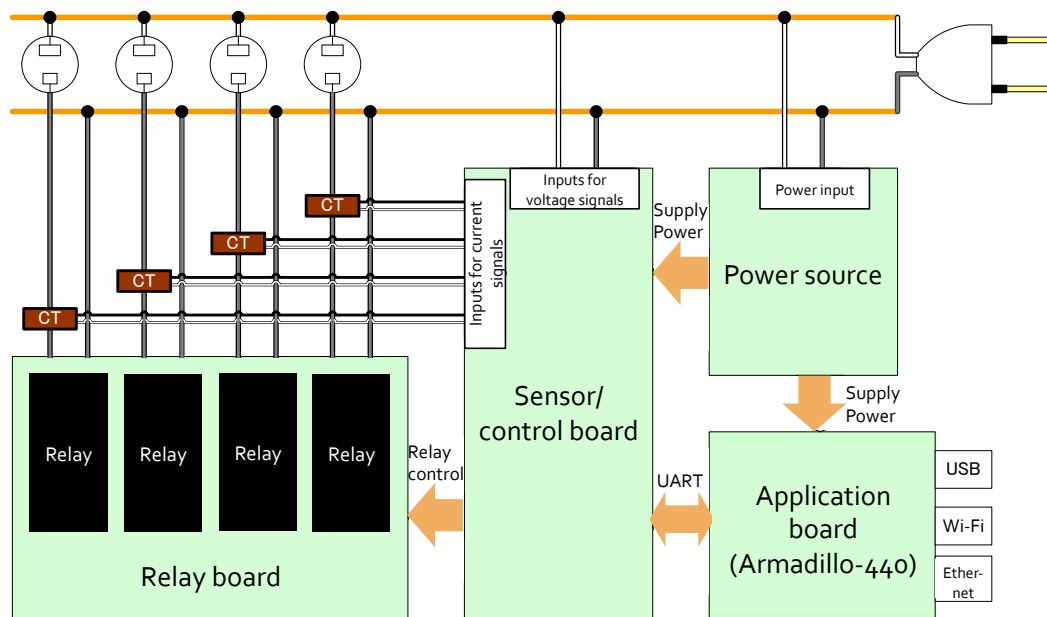


Figure 5.3 Hardware block diagram of the smart outlet.

two USB 2.0 ports for extendibility, one of which is used with a Wi-Fi interface and the other is assumed to be used with extensional devices such as various sensors.

- **Sensor/control board:** The sensor/control board measures current and voltage, and computes instantaneous and integral power consumption, and sends on-off messages to the relay board. In A/D conversion of current, it automatically switches its measurement range in a two-step manner (0.5/15A) for accurate measurement in the whole range. In calculation of power consumption, it samples current and voltage simultaneously using two A/D converters for improving preciseness. We have adopted TMS320F28035 (by Texas Instruments) with 16 inputs for processing in total eight current signals of four sockets and eight voltage signals. The left side of Figure 5.4 shows an image of the board.
- **Relay board:** The relay board has four mechanical relays for on-off control of each socket, and it controls relays obeying control messages from the sensor/control board. The right side of Figure 5.4 shows an image of the board.
- **Power source:** The power source converts AC 100V input to DC 5V outputs, and supplies power to the application board and the sensor/control board.

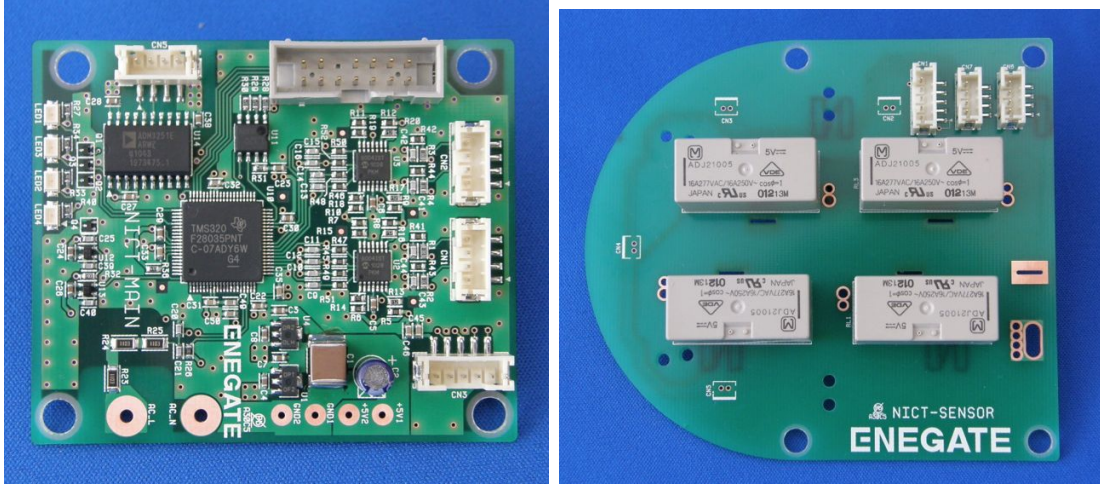


Figure 5.4 Left: sensor/control board, Right: relay board.

- **Safeness:** We have verified that the smart outlet meets the standard of resistance and commercial voltage defined in JIS C 1271-1, and have verified the standard of clearance and creepage distance referencing the Product Safety Electrical Appliance & Material (PSE) in Japan. The enclosure is made of material that meets the UL94-V0 standard of incombustibility.

5.3.2 Software Design and Implementation

In software implementation, we have designed it to be able to execute various policies installed on the Linux OS. The sensed data and control messages are formatted in an XML-like manner for extendibility and readability. It uses standard TCP/IP socket communications and HTTP/CGI to communicate easily with other devices.

First, we will describe the software processing on the sensor/control board. It executes A/D conversion of current and voltage (AC 100V) values on each socket, which is measured with current transformers with 20kHz of sampling frequency and 12bit of resolution, and computes effective current, effective voltage, instantaneous power and integral power consumption. The process sends measured and computed data (current, voltage, instantaneous power and integrated power) based on polling from a power sensor interface process (described later) on the application board, and receives relay control messages from the power sensor interface process, and sends control signals to the relay board.

The software on the application board is as follows; we have adopted Debian

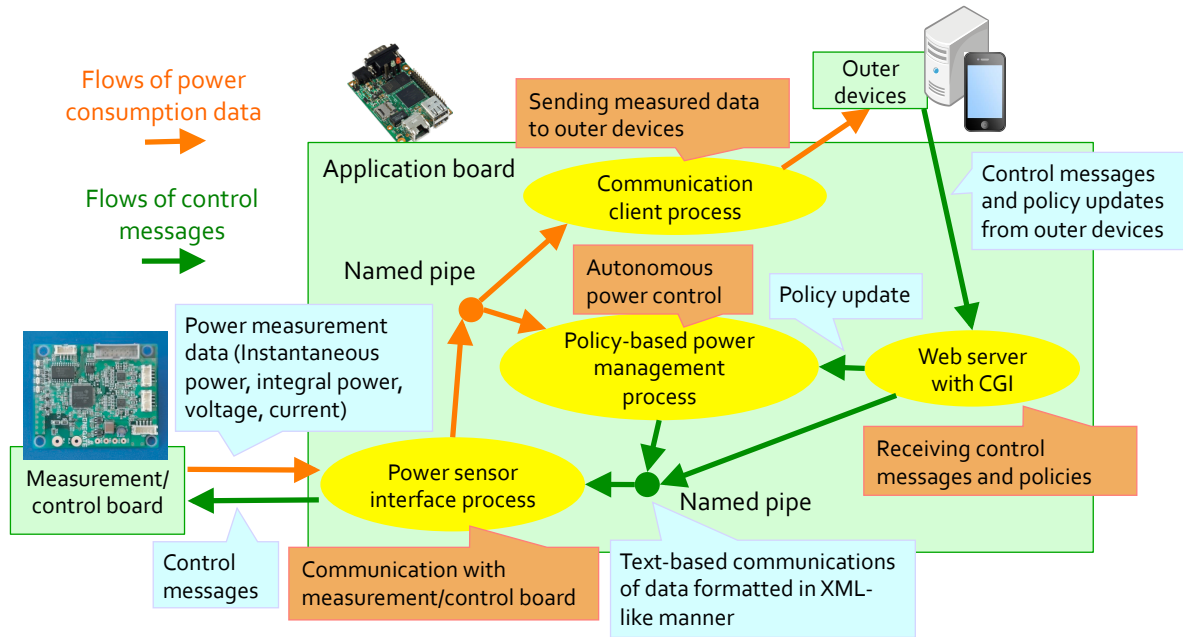


Figure 5.5 Interprocess communications on the application board.

Linux OS for embedded computers, on which four main software processes run; a power sensor interface process, a communication client process, a Web server with CGI, a policy-based power management process. They communicate in text-based messages via named pipes. The interprocess communication is designed to be suited for both centralized control where a controller decides all the behaviors of outlets, and distributed and autonomous policy-based control, for improving flexibility of system configuration. Figure 5.5 presents an entire overview of processes and interprocess communication. We will describe the functions of each process and the data formats.

- **Power sensor interface process:** The power sensor interface process communicates with the sensor/control board via serial (UART) communication; it receives measured data sent from the sensor/control board in every 50 milliseconds, transforms the received data in the XML-like format (described later) and sends data to the communication client process via a named pipe. It also parses XML-like control messages from a Web server with CGI, transforms it to control messages to relays, and sends them to the sensor/control board.
- **Communication client process:** The communication client process sends the measured power data received from the power sensor interface process to the outer devices in every 0.5 seconds. The standard TCP/IP socket communica-

tion is used as a communication protocol.

- Web server with CGI: The web server with CGI receives control messages from outer devices (e.g. smartphones, server and other smart outlets), and sends the received messages to the power sensor interface process via a named pipe.
- Policy-based power management process: The policy-based power management process executes autonomous control based on defined policies. Experimental implementation of policies is based on power-consumption data and environmental sensor data, which will be described in Section 5.5.

5.3.3 Data Formats

We have defined XML-like messages for relay control and data transmission, intending future extendibility for adding various kind of tags (e.g. environmental sensor information) and readability.

- Relay control message: The below is an example of `command_socket`, a message for controlling relays. In this example, relay #1 should be turned on and #3 should be off.

```
<root>
  <info>
    <kind>command_socket</kind>
  </info>
  <data>
    <socket1><state>ON</state></socket1>
    <socket3><state>OFF</state></socket3>
  </data>
</root>
```

- Measured power consumption data: The below is an example of `notice_wattmeter`, which is used for sending measured power consumption data. This example means that integrated power on socket #1 is 20954Wh, instantaneous voltage is 102.070V, current is 1.551A, instantaneous power is 84.6W, and the relay has been turned on.

```
<root>
  <info>
    <kind>notice_wattmeter</kind>
```

```

    <time>20120909104854039</time>
  </info>
  <data>
    <socket1><wh>20954</wh><volt>102.070</volt><current>1.551</
      current><watt>84.6</watt><state>ON</state></socket1>
    <socket2><wh>2536</wh><volt>102.085</volt><current>0.013</
      current><watt>4.1</watt><state>ON</state></socket2>
    <socket3><wh>2499</wh><volt>102.073</volt><current>0.242</
      current><watt>14.2</watt><state>ON</state></socket3>
    <socket4><wh>1219</wh><volt>102.080</volt><current>0.081</
      current><watt>5.1</watt><state>ON</state></socket4>
  </data>
</root>

```

5.4 Evaluation of Power Measurement Function

We have evaluated error rates of power measurement, voltage measurement and current measurement. Power measurement error is tested in a range of from 100% to 3.3%, which is standardized in JIS C 1271-1: power meter. Voltage measurement error is tested in a range of 80V to 120V. Current measurement error is tested in a range of 100% (15A) to 0.0033% (5mA). The results show that power error is under 2%, voltage error is under 0.1% and current error is under 0.2%. These results mean that the developed outlet measures power precisely even on measuring standby power as small as 0.5W, because 0.033% current compared to rated values (100V, 15A) corresponds to approximately 5mA.

Figure 5.6 shows voltage and current waveforms of a fluorescent lamp with an inverter, which is measured using the smart outlet. It clearly shows distortions of waveforms unique for inverters, and shows instantaneous variation of current and voltage. This example shows that instantaneous power should be huge because peaks of current and voltage are overlapped. Therefore we have evaluated that the outlet has sufficient measuring preciseness for applications such as appliance recognition from current waveforms.

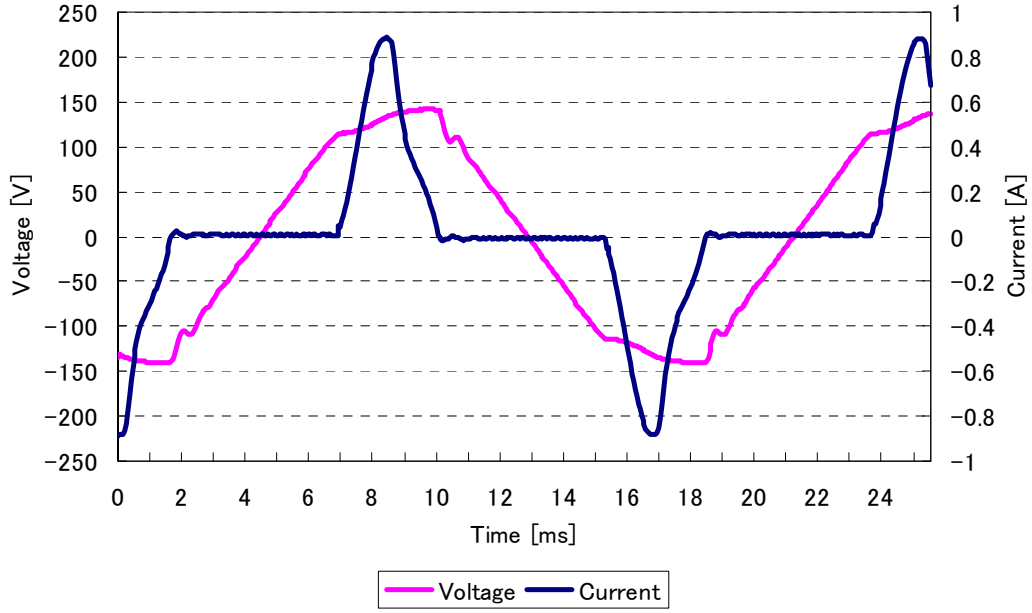


Figure 5.6 Voltage and current waveforms of a fluorescent lamp with an inverter.

5.5 Experimental Implementation of Control Policies

In this section, we describe a software-based circuit breaker function for the optimal allocation of energy, and a function for avoiding overlaps between inrush currents as experimental implementation of control policies.

5.5.1 Software-based Circuit Breaker

The conventional breaker is implemented in a home distribution board, and it is not able to limit current supply for each individual socket in a home. When there happens over current, the breaker unnecessarily has to shut down the whole home power network that causes inconvenience for users. Also, the conventional heat-driven or magnetic-driven breakers work differently depending on various situations. Therefore, it is difficult to control them to work based on flexible conditions for each independent socket, which implies it is not sufficiently suited for optimal allocation of energy.

An advantage of the software-based breaker is that we can set flexible and exact

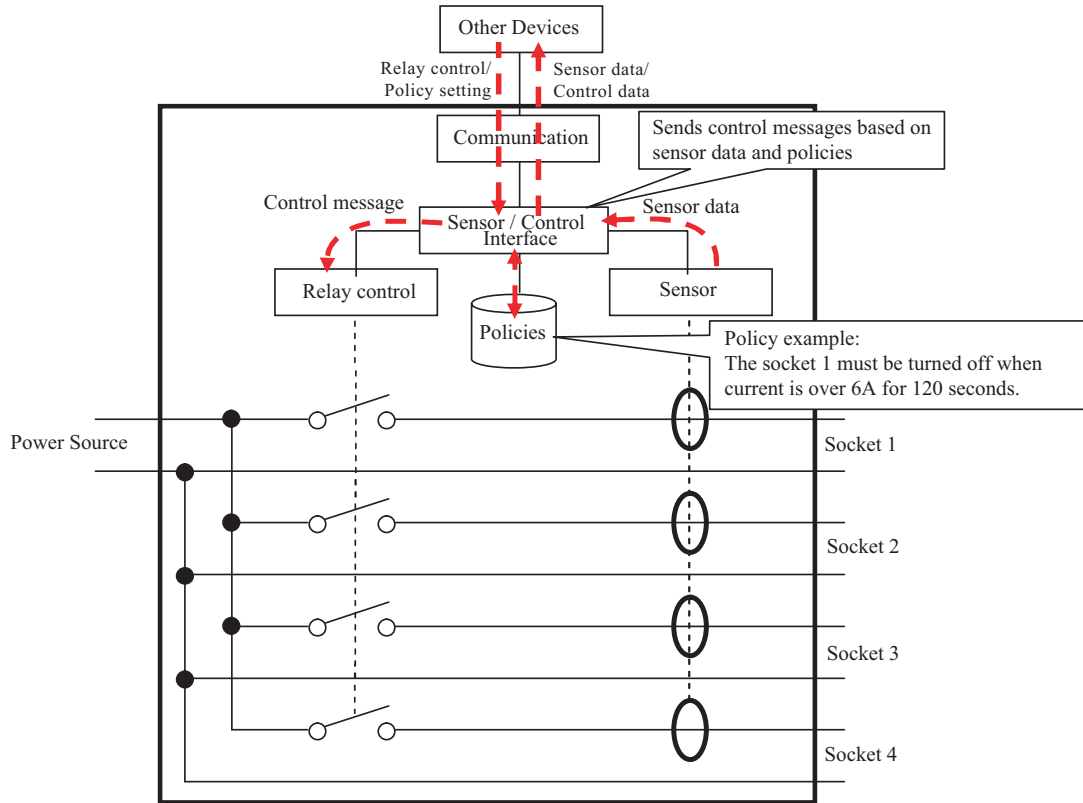


Figure 5.7 Software-based circuit breaker function.

policies, which enables more intelligent and convenient limiting than the conventional breaker. Figure 5.7 shows overview of the software-based breaker. The sensor modules measures values of current in each individual socket, and make on-off control under the policies set by users. The policy is, for example, “The socket #1 must be turned off when current is over 6A for 120 seconds, or 9A for three seconds.”

5.5.1.1 Implementation

The circuit breaker watches current values of each socket, and shuts off a socket where current value exceeds threshold values for certain duration set by policies. The threshold value is computed using a base value of current and excess percentage. For example, when the base value of current is 5A and excess percentage is 120%, the threshold is computed as $5A \times 1.2 = 6A$. Excess percentage and duration time can be set in a two-step manner for flexible control. Similarly, the circuit breaker based on a total current value shuts down all the sockets when total current consumption exceeds a threshold value for a certain duration. These parameters can be set and

modified from outer devices via HTTP/CGI. The LED indicator blinks when current exceeds a threshold value to notify users before it actually shuts down the socket.

5.5.1.2 Experiments and Evaluations

We have made experiments under the following two policies set in the implemented breaker function, and have confirmed that the policies properly work.

- Limiting based on the individual socket: “Each socket must be turned off when current exceeds 120% of 3A for one second or 150% for three seconds.”
- Limiting based on the total current of all the sockets: “All sockets must be turned off when the total current of all the socket exceeds 120% of 15A for one second.”

Therefore, we have evaluated that the developed outlet has sufficient functions to watch and control for the optimal allocation of energy, because it properly controls sockets based on the conditions independently defined for each socket.

5.5.2 Function of Avoiding Overlaps of Inrush Currents

Many kinds of appliances cause an inrush current, which is a huge amount of current that occurs immediately after they are turned on and exceeds their rated current values. In case of recovering from blackouts, overlapped inrush currents caused by multiple appliances should be extraordinarily huge, forcing a circuit breaker at a distribution board to shut down all of the appliances, and causing unstability of voltage at power sources followed by undesired effects to other appliances. Especially, in case of distributed power sources such as solar power, voltage drop tends to be caused easily because its power capacity is generally limited.

We have implemented a function to avoid overlaps of inrush currents that interposes some delays when the outlet turn on multiple relays. We will describe the implementation, experiments and evaluations of the function.

5.5.2.1 Implementation

When the outlet boots up and multiple socket should be turned on simultaneously, this function inserts some delay between messages sent to the sensor/control board to turn each socket one by one, instead of sending multiple control messages at the same time. The delay amount is set suited to characteristics of inrush currents of the

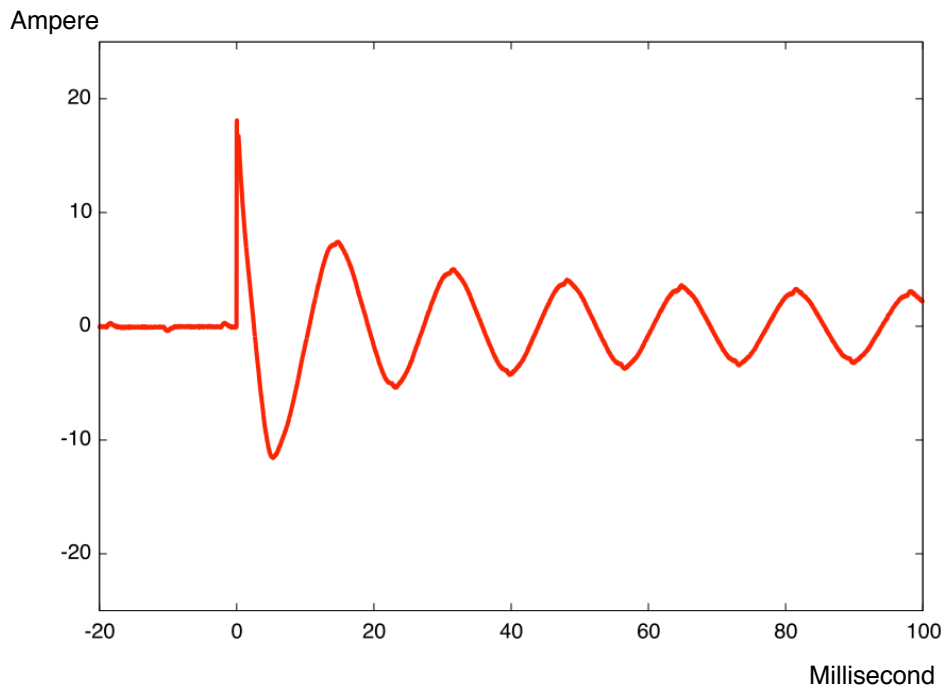


Figure 5.8 Current waveform of two light bulbs (without avoiding overlaps between inrush currents).

appliances connected to the sockets.

5.5.2.2 Experiments and Evaluation

The experiments are executed using two incandescent lamps. Generally, an incandescent lamp causes inrush current when it is turned on until temperature of its filament is stabilized. The incandescent lamp's rated power is 100W, and inrush current caused by a single lamp is 9.528A (average of 10 times). The inrush current caused after 20 milliseconds from being turned on, therefore we have set delay amount as 50 milliseconds so that the amount of delay is sufficient but does not cause inconvenience.

The overlapped inrush current of the two lamps is 19.296A (average of 10 times) when the delay is not inserted. Figure 5.8 shows an example of waveforms. In this example, the peak of the overlapped currents is 18.10A. On the other hand, when 50 milliseconds of delay is inserted, the total inrush current is 13.238A (average of 10 times). Figure 5.9 shows an example, where the peak current is 11.26A.

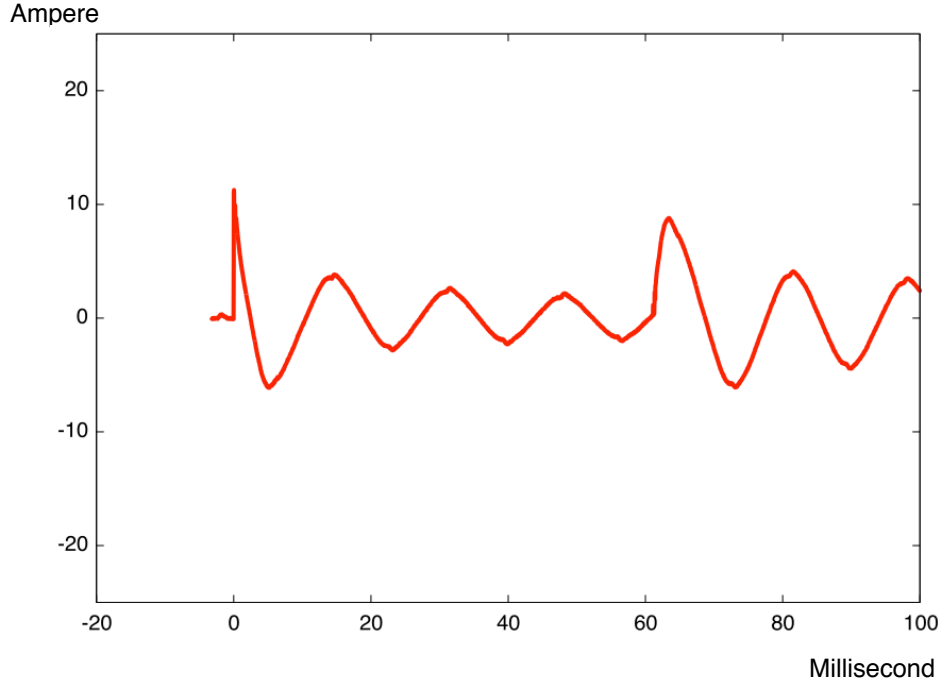


Figure 5.9 Current waveform of two light bulbs (with avoiding overlaps between inrush currents).

5.6 Deployment in Real-life Environments and Control of Appliances Using Various Sensor Information

In this section, we describe the networking of multiple smart outlets for providing an energy-aware service platform and home automation of multiple and different appliances, utilizing various sensor information. The network consists of the smart outlets, various sensors (light, temperature, humidity and motion) connected to the outlets, and a server for accumulating data. We have deployed the network in real-life environments. With highly-frequent power sensing, environmental sensors and motion sensors, we have implemented power control policies for realizing energy-aware coordination of appliances. There should be two kinds of system architectures; one is a centralized architecture, in which a center server takes all the control of smart outlets in the network. It is rather simple architecture. The other is distributed architecture, where outlets work independently based on policies defined in each one. It takes

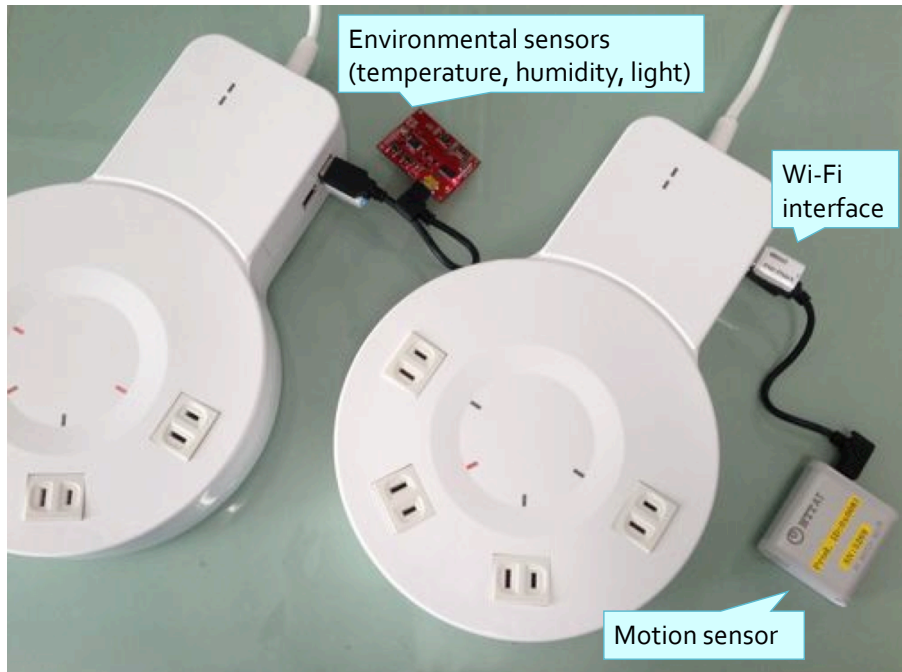


Figure 5.10 Smart outlets with Wi-Fi interfaces and sensors.

advantage of having no single-point-of-failure. The system is able to work in both centralized and distributed manners. We have implemented, in real-life environments, several policies which realize the centralized control based on power consuming information, and the distributed control based on environmental sensor information (to the best of authors knowledge, this is the first literature report describing deployment and experiments of distributed policy-based power management, especially in real-life environments.)

5.6.1 Environmental Sensors and Motion Sensor

We have installed environmental sensors and a motion sensor to the smart outlet, as shown in Figure 5.10. The environmental sensor is USB Weather Board [79], which senses temperature, humidity, light and pressure with a frequency of 1 per second. The motion sensor is AT WATCH NET IR_mini [65], which detects human motion using infrared.

5.6.2 Communication Network

Various communication media could be deployed in a smart outlet network; Wi-Fi, ZigBee, Ethernet, etc. Among them we have chosen Wi-Fi. Since smart outlets are usually placed on some narrow space in a room, wireless communications are more preferable than wired ones. Wi-Fi has rich bandwidth for highly-frequent data gathering and sending control messages, and it has been already well spread in ordinary homes, which means that additional construction of a network infrastructure is less required. The Wi-Fi standard used in the network is IEEE 802.11n with maximum bandwidth of 150Mbps using 2.4GHz of frequency, and it is protected with WPA2-AES encryption.

As a transport layer protocol, we have chosen TCP for reliability, dependability and safety, because the system does control electricity actively, not only gathering data.

The smart outlet runs an HTTP/CGI server via which other devices (a center server, a smartphone, etc.) control it.

5.6.3 Server

The network has a server (Core 2 Duo 1.4GHz, 4GB RAM, SSD, OS X 10.8 Mountain Lion) for accumulating data from the smart outlets. It also has an HTTP service for the visualization of power consumption. The server periodically generates a graph showing power usage based on gathered data. A user can see the data and the graph through accessing the server with web browsers.

The network can be accessed from outside of a home, and via the Internet a user can control the smart outlets and appliances connected with them.

5.6.4 Data Formats

Control messages, power sensing data, environmental sensor data and motion sensor data are formatted in XML-like manners as follows, considering the extendibility, versatility and easier handling.

A control message is formatted as shown below. When the smart outlet receives this message, it turns on socket 1 and turns off socket 3.

```
<root>
```

```

<info>
  <kind>command_socket</kind>
</info>
<data>
  <socket1><state>ON</state></socket1>
  <socket3><state>OFF</state></socket3>
</data>
</root>

```

Environmental sensor data consist of time, temperature (Celsius), humidity (percent) and light (0 is the brightest and 1023 is the darkest).

```

<root>
  <info>
    <kind>env_sensor</kind>
    <time>20120905143743429</time>
  </info>
  <data>
    <temp>28.4</temp><humid>58.5</humid><light>950</light>
  </data>
</root>

```

Motion sensor data consist of time and whether it has detected a motion or not.

```

<root>
  <info>
    <kind>motion_sensor</kind>
    <time>20120905143743429</time>
  </info>
  <data>
    <motion>detected</motion>
  </data>
</root>

```

5.6.5 Deployment of the Developed System in Real-life Environments

We have deployed the system in two different places for experiment; a one-room apartment and a multiple-room apartment. In particular, in the one-room apartment

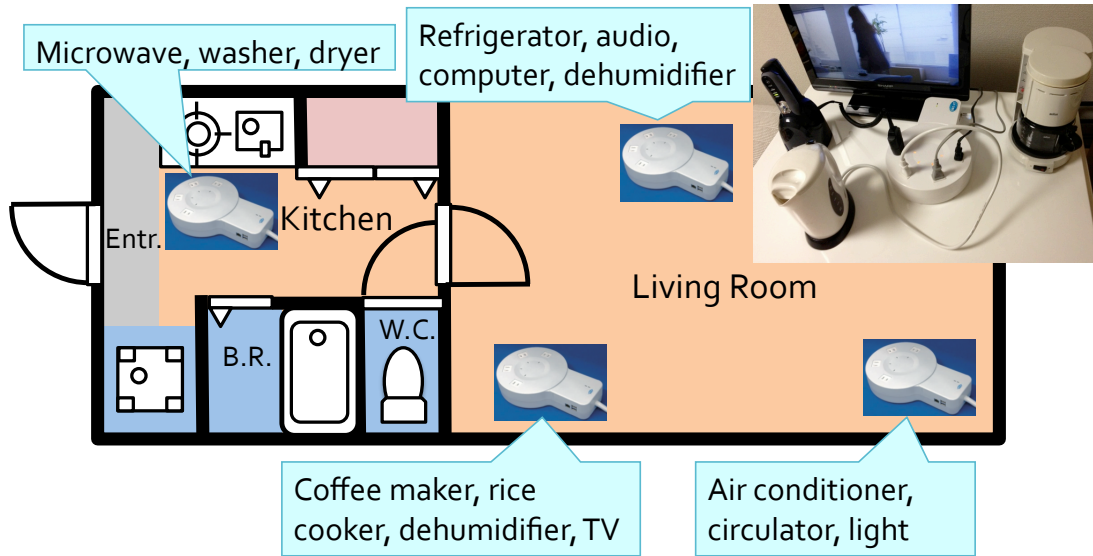


Figure 5.11 Smart outlet network in a one-room apartment.

we have been continuously running the system in real-life of a resident, gathering power consumption data of all appliances (except some built-in lighting), and have implemented several policies for realizing energy-aware services utilizing environmental sensors and motion sensors. The appliances used in the one-room apartment are as follows; an air conditioner, a fan, a circulator, a microwave, a refrigerator, a coffee maker, a rice cooker, computers, an audio system, a TV, an electric kettle, a washer, an LED light and two dehumidifiers. The smart outlets are on the same Wi-Fi ESSID and on the same IP network in which the resident's daily-use computers (laptops, tablets and a smartphone) participate. Figure 5.11 and Figure 5.12 show the deployment of the outlets in the one-room apartment and the multi-room apartment, respectively.

5.6.5.1 Gathering Power Consumption Data of Each Appliance

In order to sense power consumption and control all appliances in the experimental environment, we have placed four outlets in the one-room apartment, and seven outlets in the multi-room apartment. In both environments, power consumption data have been successfully gathered in every 0.5 seconds without causing particular data losses, even when the resident uses a microwave or an IH cooking heater which make noise with frequency of around 2.4GHz, and when the resident uses computers for daily-activities (watching video streaming, browsing web or using syncing services

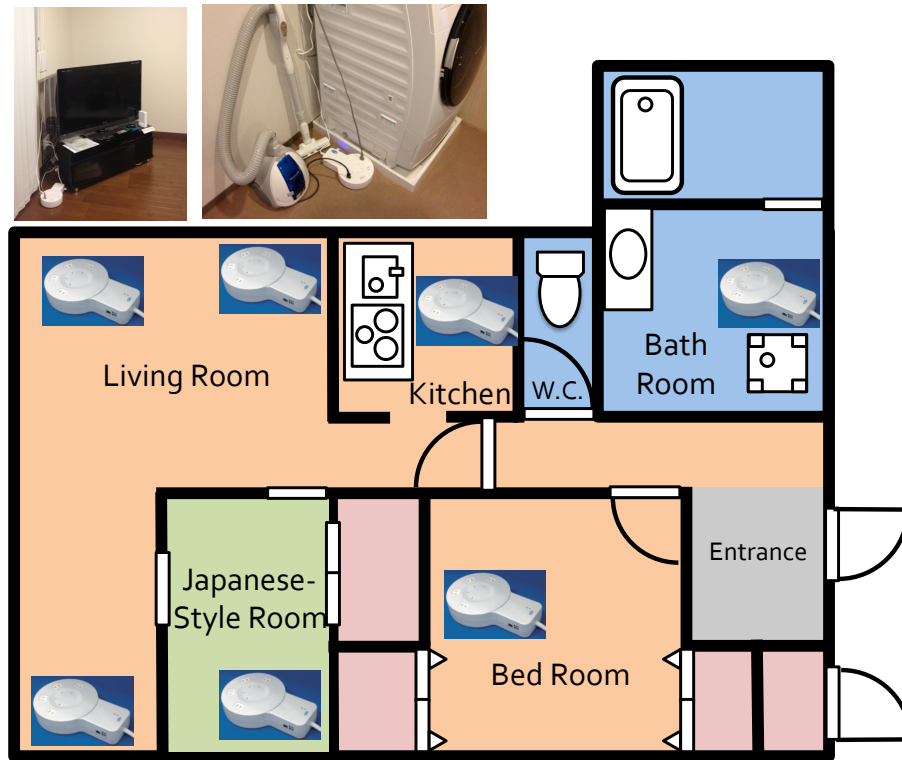


Figure 5.12 Smart outlet network in a multi-room apartment.

with cloud servers). The time needed for transmitting data from an outlet to the server is around 10-200 milliseconds.

In the one-room apartment, power consumption data have been continuously gathered for several months. Figure 5.13 shows one-day (Sep. 9, 2012) power consumption data plotted in every one minute. By combining power consumption records of multiple and different appliances, it is easy to grasp resident's behaviors of the day clearly. The usage of the computers and the fan mean that the resident was awake until 3:30 last night, and the usage of the TV implies that the resident woke up on approximately 10:00. The microwave was used for cooking at 12:30 for lunch, and 18:30 for dinner. The computers were turned 15:00-17:00 and around 23:00, while the rice cooker worked after 23:00.

Figure 5.14 represents an hourly power consumption data on another day (from 8:00 to 9:00, Nov. 6, 2012). Data gathering with high frequency (period of 0.5 second) enables us to grasp detailed information of power consumption of each appliance and behaviors of the resident; For example, periodic impulse-like peaks of the refrigerator should mean rush current it causes (widths of the peaks are about only one second).

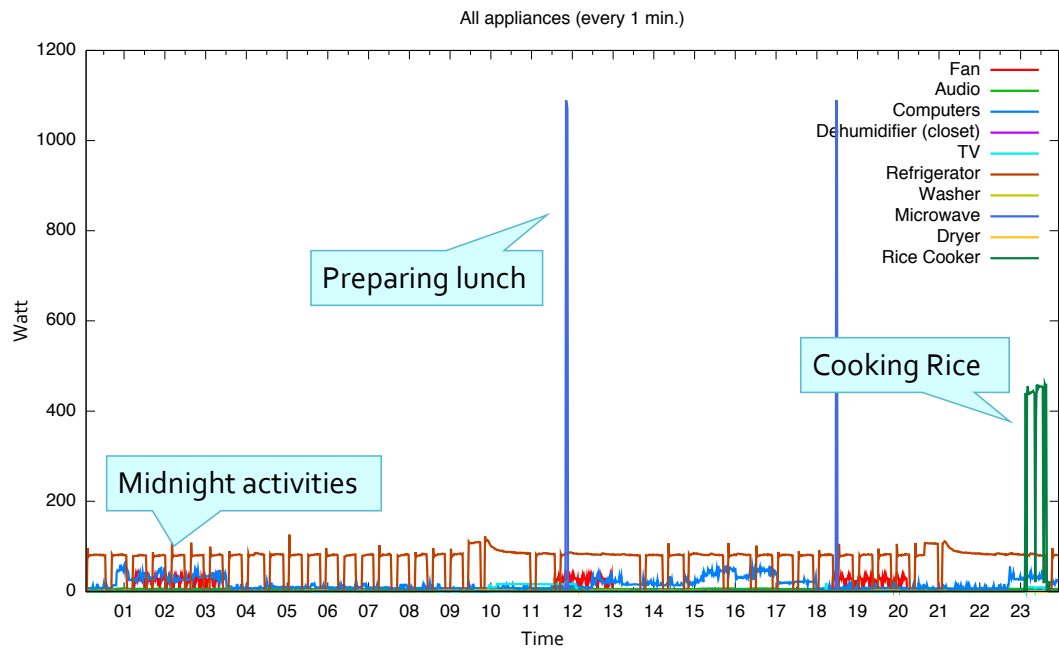


Figure 5.13 Visualization of power consumption data; one-day.

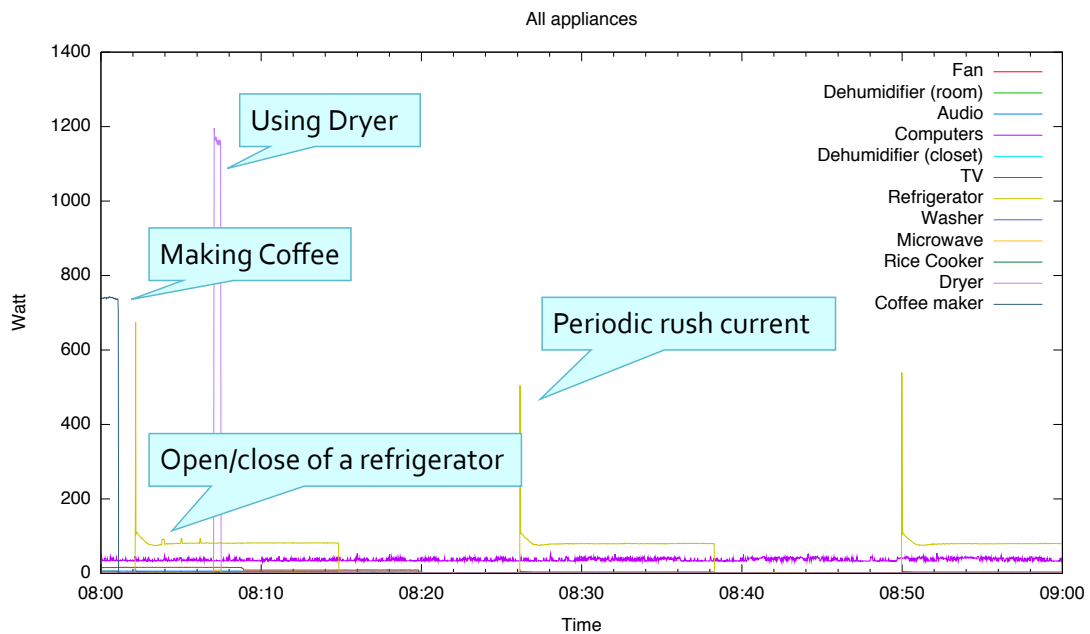


Figure 5.14 Visualization of power consumption data; one-hour.

On the other hand, irregular tiny peaks mean that the resident opened and closed the refrigerator, with widths of several seconds.

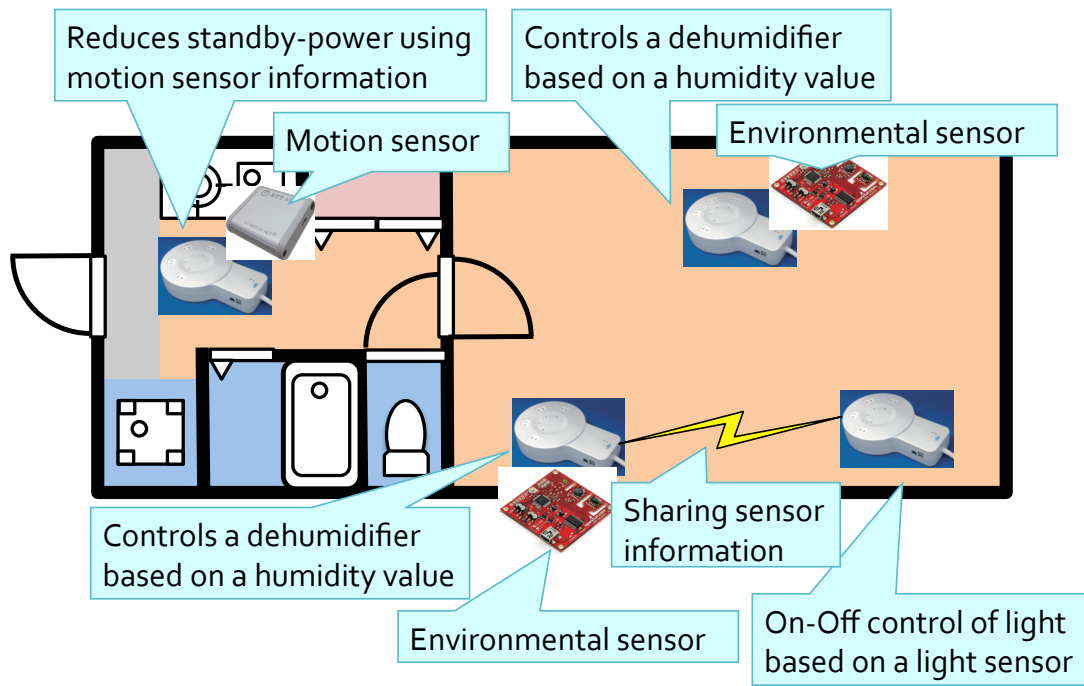


Figure 5.15 Deployment and setting sensors in the one-room apartment.

5.6.5.2 Scheduled On-Off Control of an Audio System

The smart outlet has a capability of scheduled on-off control using the standard cron daemon installed in Debian Linux OS. In the one-room apartment, we have set a schedule for reducing standby-power of an audio system, which is suitable for daily habits of the resident; during 8:30-17:30 on weekday (when the resident goes out for work and should be absent) the outlet automatically turns off the relay of the socket with which the audio system is plugged, and it does no control in weekend. This schedule-based control reduces approximately 6W.

5.6.5.3 Reduction of Stand-by Power Using a Motion Sensor

Another natural way to reduce standby-power reasonably is to utilize motion sensor information. As shown in Figure 5.15, we have installed the motion sensor to the smart outlet placed in the kitchen, and have set the following policy; when the motion sensor does not detect the resident for two hours, the smart outlet turns off the relay with

which the microwave is plugged. This control saves approximately 2W.

5.6.5.4 On-Off Control of an LED Light Using Environmental Sensors

Figure 5.15 also represents environmental sensors set in the living room. The two smart outlets located below on the figure share environmental sensor information by communicating above-mentioned XML-like messages. The outlet located on right side takes on-off control of the LED light, based on shared light information; when the room is dark, the LED light automatically turned on and works as a nightlight.

5.6.5.5 Coordination of Multiple or Different Appliances

As noted above, two different architectures can be considered for realizing coordination of multiple or different appliances; a centralized architecture and a distributed architecture. In the centralized architecture, a center server takes control of all the smart outlets based on policies. On the other hand, in the distributed architecture with no center server, each smart outlet works independently based on their own policies.

- Coordination of appliances based on power consumption data: It is commonly known that simultaneous usage of an air conditioner and a circulator makes cooling/heating efficiency better. Therefore, as an example of a coordination in the centralized manner, we have implemented an automatic coordination of the air conditioner and the circulator. In the system, the data gathering server also works as the center server. The server monitors power consumption of the air conditioner, and when the power consumption of the air conditioner suddenly arises from 0W to over 100W, the server interprets that the air conditioner is turned on by the user and it sends a control message to the targeted smart outlet with which the circulator is plugged.
- Coordination of appliances based on sensor information: We have implemented a policy for coordination of two dehumidifiers based on humidity sensor information in the distributed manner. Two different dehumidifiers are plugged with different outlets, and each outlet has environmental sensors and its own control policy. The implemented policies are as follows; the outlet located left side in the figure turns on the dehumidifier when a value of humidity is over 60%, and turns off when the value goes under 55%; the other outlet turns on the dehumidifier when humidity is over 65%, and turns off when under 60%.

The policies make the dehumidifiers work only when needed, not be always turned on consuming approximately 50W in total.

5.7 Concluding Remarks

The design and implementation of a smart outlet for policy-based power management have been described, which aims to reduce user's handwork required for power-saving activities and to realize efficient power control to realize optimal allocation of energy. The smart outlet has been designed to have an accurate power-sensing function, to have sufficient computational resources to execute various policy-based power management, and to be used safely in real-life situations. As applications of policy-based power management, the authors have implemented and evaluated a software-based circuit breaker, and a function of reducing overlaps of inrush current caused by multiple appliances. We have also implemented the smart outlet network as a platform for energy-aware services utilizing various sensor information. The smart outlet network has been deployed in the real-life environments; the one-room apartment and the multi-room apartment. Especially, in the one-room apartment the developed system continuously gathers power consumption data of all the appliances. It has been found that highly frequent gathering (0.5 seconds) of power consumption data helps us to analyze characteristics of appliances in detail, and to grasp daily behavior of the resident. The following energy-aware services have been implemented in the system; scheduled reduction of stand-by power, on-off control based on a motion sensor, coordination between the air conditioner and the air circulator based on power consumption data in a centralized manner, and coordination between two different humidifiers based on environmental information in a distributed manner. Therefore, we conclude that the developed smart outlet has sufficient functions for policy-based power management utilizing behavior information and environmental information.

Chapter 6

Policy-based Power Router with Power Sensors for Optimizing Allocation of Energy

6.1 Introduction

While the global warming and energy saving have been critical issues, power consumption in homes and offices is in increasing tendency because of the ongoing spread of various electric appliances, especially compared with that in 1970's [4]. As assumed in the concept of smart grid [92], future power networks are supposed to contain different power sources in a mesh-like structure, including distributed generations such as solar panels, wind power generations, and electric vehicles (EV) that works not only as transportation but as a huge battery. Therefore, efficient use of various power sources has been crucial for energy saving. These power sources have characteristics such as cost, stability and amounts of CO₂ emission. Hence it is desired to use them in a way suited for various characteristics of power consuming devices; for example, it is preferable that a laptop computer is supplied from natural power sources because it has a battery and accepts less stable power, while a TV is desired to be supplied with power from sufficiently stable power sources such as a commercial power source.

With the recent advance on information technology, concepts of next-generation power networks have been proposed that automatically optimize power supply and consumption utilizing information technology for realizing efficient and flexible power

control. In the Open-Electric-Energy-Network (OEEN) [87], the concept of a *power router* was proposed that controls power flows based on information tagged with power to optimize power allocation. Recently, Matsuyama proposed a concept of Energy-on-Demand (EoD) [56], where power consuming devices explicitly request power based on *Quality-of-Energy* (QoEn) parameters such as the amount of power, stability, cost etc., and they would be supplied with power only when some mediation system in the network accepts their requests. The mediation system computes the optimal matching between power sources and power consuming devices based on QoEn parameters of both devices and sources, depending on power control policies best suited for the users. Moreover, QoEn power routing [72] has been proposed to realize *end-to-end power routing* between power sources and power consuming devices through multiple power routers in the network, applying schemes of end-to-end Quality-of-Service (QoS) routing and resource reservation in computer networks [17]. In addition to that, while the end-to-end QoEn routing is important as global optimization, another crucial factor is autonomous local control; for example, in a situation where distributed power sources appear and disappear dynamically, establishing the optimal end-to-end power route is not always completed sufficiently fast to immediately react to sudden or unexpected state change of power sources. Hence power consuming devices might be temporally unavailable until the QoEn routing is completed. In such dynamic situations, it is better that the appliances temporally be supplied with other available power sources until the unavailable power source is restored or some global optimization schemes reallocate other power sources to the power consuming devices. Therefore, the power router is desirable to be capable of autonomous local control as well as QoEn routing.

Based on these concepts and ideas, we have designed and implemented a *policy-based power router* as a prototype for experimental use in a research project “Integration Technology of Information, Communication and Energy” [68]. It has mechanical relays for power allocation between two power inputs and two outputs in a circuit-switching manner. The major characteristics of the developed power router are that it has power sensors at the outputs, and has sufficient computational resources for autonomous policy-based power control. It has also communication interfaces for sending and receiving control messages and sensed data between outer devices such as other routers or routing servers. We have confirmed that an end-to-end power route is able to be established using the developed power router, combined with a routing server utilizing a QoEn routing protocol proposed and implemented by Miyamoto et

al. [60]. As experimental implementation of autonomous local control, we have also implemented a control policy on the router that, when the power from one source suddenly becomes unavailable, the power router autonomously switches power supply to the other available source. This policy reduces temporal unavailability of appliances and works as a complement to the global optimization with QoEn routing.

The rest of this chapter is composed as follows. In Section 6.2, we refer related work. Section 6.3 presents basic concepts, gives requirements definition, and describes the design and implementation of the power router. The experimental implementation of QoEn routing and a local control policy are described in Section 6.4. Finally, Section 6.5 presents future work and concludes this chapter.

6.2 Related Work

Several literatures discussed systems for power routing and efficient allocation of energy. Toyoda et al. originally proposed a fundamental concept of the power router [87]. Toward more advanced integration of information and energy, Takuno et al. proposed and implemented a PLC-based power routing system with direct information tagging to power waveform [82] toward realizing the concept of power packetization utilizing silicon carbide semiconductors. Shibata et al. designed and implemented a PoE-based power routing system as a prototype of an EoD power network [73]. In an EoD network proposed and implemented by Kato et al. [43], multiple smart outlets work coordinately, and the entire smart outlet network can be regarded as a virtual power router that optimizes power allocation to appliances. Universal Power Router [80] is a device that controls power switch in a policy-based manner depending on Time-of-Use, a state of battery, states of power generations, etc. It does not have capability of allocating different power sources to difference power consuming devices; all the outputs should be supplied from single input. Intelligent Power Switch [33] [53] has been developed for providing a testbed for research work on smart grid, has six ports that work as both inputs and an outputs, and does not have power sensors and functions for autonomous control. Takahashi et al. constructed and evaluated a power network with multiple power routers, especially focusing physical feasibility of connecting multiple routers in both serial and parallel manners [81].

6.3 Design and Implementation

This section describes basic concepts of an assumed power network and a power router, gives requirements definition, and presents the design and implementation of the router in both hardware and software aspects.

6.3.1 Basic Concepts

The power router has been developed as a prototype for experimental use in a research project “Integration Technology of Information, Communication and Energy,” based on the concepts of EoD and QoEn routing. The policy-based power controls considered in this chapter are classified into global optimization with QoEn routing and autonomous local control.

Figure 6.1 shows an entire image of the EoD power network with QoEn routing and autonomous local control. In this example, there are three power sources with different characteristics; the commercial power supply with capacity of 15A is stable, power from the solar power panel with capacity of 0.5A is less stable because it depends on whether conditions, and the electric vehicle with capacity of 10A is stable but possibly temporal because it is disconnected from the network when used as transportation. On the side of appliances, the laptop requires 0.3A and accepts relatively unstable power because it has a battery, while the TV requires 4A and works with only stable power. In the following subsections, we describe both global optimization with QoEn routing and autonomous local control based on example scenarios in Figure 6.1.

6.3.1.1 Global Optimization with QoEn Routing

The power routing system decides the best matching and end-to-end routes between the power sources and the appliances based on their QoEn parameters. For example, in Figure 6.1, the laptop computer can be supplied with power from the solar panels. The TV can be supplied from the stable EV and commercial source with sufficient capacity of 10A and 15A, however it is not able to work with less stable power from the solar panel. The power router configures assignment between their inputs to proper outputs, depending on control messages from some routing scheme such as a routing server.

The three-layer protocol suite in EoD networks has been proposed [47] for inter-

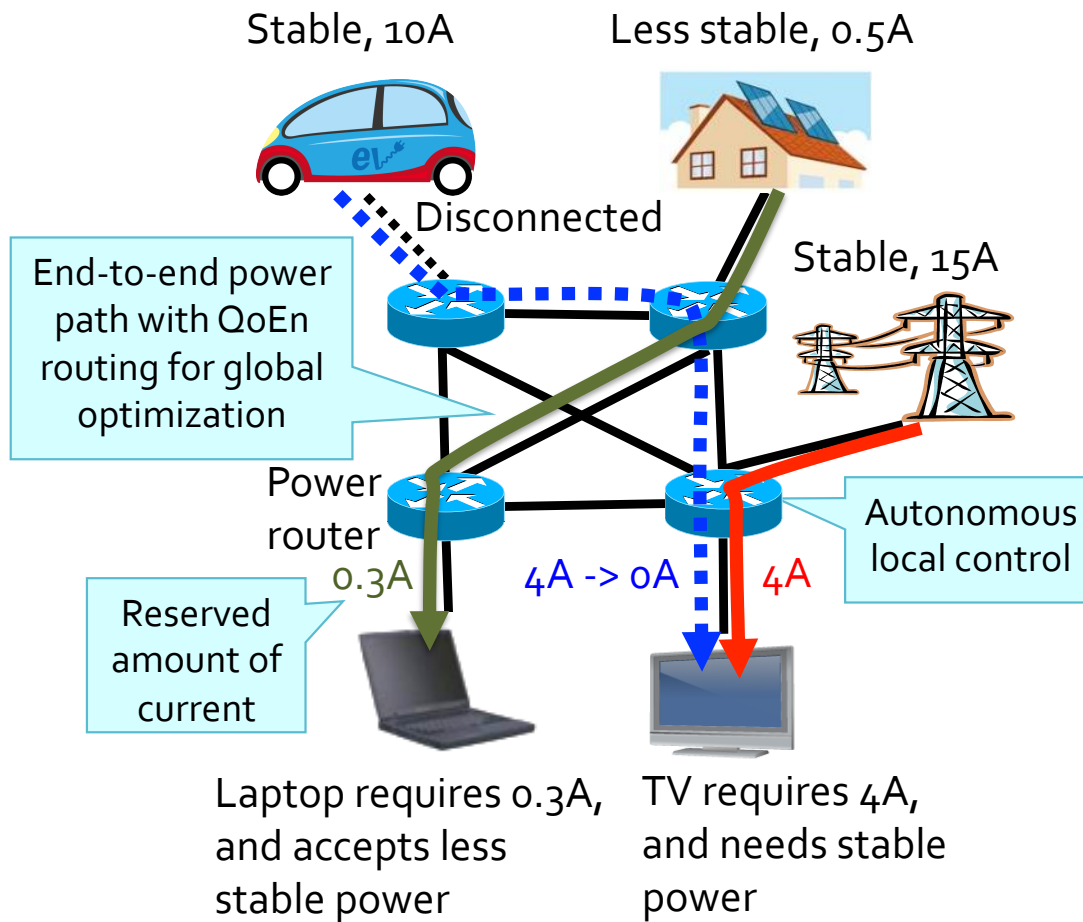


Figure 6.1 Policy-based power control in an EoD power network using power routers; QoEn routing for global optimization and autonomous local control.

operability, because there have been various implementation of the power routers and it is reasonable to design protocols that are able to handle differently implemented power routers in integrated manners. The concept of the three-layer protocol suite is based on Quality of Service routing technology in computer networks that is used for guaranteeing throughput, delay and other parameters for particular services such as real-time video streaming. In the EoD protocol suite, the upper layer, a *require/response layer*, determines QoEn parameters of both power sources and appliances, based on various factors such as their characteristics, preferences of users or environmental information. The middle layer, a *routing layer*, computes the best matching and routes between appliances and power sources based on QoEn parameters in an end-to-end manner using some optimization algorithms (e.g. resource allocation algorithms or shortest path algorithms), and reserves resources at each router located

on the computed power route. The bottom layer, a *physical layer*, works to establish physical power routes that the power routing protocols defines, and maintain the power consumption under the reserved value. Our power router has been developed as an experimental device for implementing the physical layer protocols. It also monitors power consumption caused by each power flow, and maintains each flow not to exceed the amount decided by the routing protocols.

6.3.1.2 Autonomous Local Control

While the global optimization with QoEn routing is considered to be effective for energy efficiency, it is not necessarily able to immediately react to unexpected variations in a local power situation, because the global optimization probably requires some time duration for processing across multiple devices. Hence it is considered that autonomous local control adequately complements the global optimization, especially in power networks where states of power sources change dynamically. For example, in Figure 6.1, the TV has been supplied with power from the EV; however, when the EV is disconnected to be used as transportation, power from the EV has become unavailable. Then the power router located next to the TV autonomously switches power assignment to the commercial source, reducing unavailable time duration of the TV. Therefore, it is considered to be useful and practical that the router has sufficient functions for autonomous local control as well as QoEn routing.

6.3.2 Requirements Definition

Based on the above concept, the power router is required to satisfy the functions of the physical layer in the EoD protocols and for autonomous local control. The requirements are classified into the following; policy-based power control of assignment between power inputs and outputs, communication interfaces for coordination with other devices, sending and receiving control messages and sensed data, power sensors for grasping local power states including power consumption, and future extendibility.

Requirements definition for power measurement and for control/communication/extendibility are presented in Table 6.1 and Table 6.2, respectively.

- Control: The router should have functions for switching between power inputs and outputs in arbitrary combinations. The circuit switching of power is implemented using mechanical relays for easier implementation as the prototype.

Table 6.1 Requirements definition of control, communication and extendibility.

Items	Specs	Notes
Control	Mechanical relays	Switching between multiple inputs and outputs. Enables arbitrary allocation between power inputs and outputs, including off control.
Communication	Wi-Fi and Ethernet	Communicates with routing servers or other power routers, policy distributing servers etc. Universal media for easier connectivity
Extendibility	USB	For other communication media and various sensors

Table 6.2 Requirements definition of measurement.

Items	Specs	Notes
Rated voltage	100V	Alternating current For measuring local power information and control for policy-based power management using sensed data of power consumption. 0.5/15A, Equivalent with the standard power meters in Japan
Rated current	15A	
Rated frequency	50/60Hz	
Measurement	Integral power, instantaneous power, voltage and current	
Error	Under +/- 2%	

Rated voltage and current is designed to be 100V and 15A respectively, for experimental use with ordinary power equipment or power consuming devices. Time duration needed for control should be sufficiently short compared with

that required in the global optimization with QoEn routing.

- **Communication:** The router should have communication interfaces for accepting control from outer devices via control messages, exchanging information needed for routing, sending measured data of power consumption, updating policies and configuring parameters. Popular communication media (Wi-Fi and Ethernet) are used for easier coordination with information devices and smart outlets.
- **Measurement:** With implemented power sensors, the power router is able to measure physical information of power. The measured information can be used for policy-based autonomous control based on the amount of power consumption, current and voltage data, including watching and controlling power flows based on the reserved amount of power by the routing layer. This enables the router to grasp the local states of power, and to control autonomously based on variations of local situations, and to monitor current value and maintain reserved values. The error rate for measuring power consumption is designed to be less than $\pm 2\%$, referencing the standard power meters in Japan.
- **Others:** Indicators that presents the states of power inputs, outputs and their assignments. Extendibility is also important for future implementation of other communication media or more advanced policy-based control based on environmental sensors, etc.

6.3.3 Implementation

The implementation is done referring the smart outlet described in Chapter 5. The smart outlet was designed and implemented for policy-based power management, and has power sensors, relays and communication capability, and there has been implemented policy-based power management system using the smart outlets [44]. However, the smart outlet is able to treat only single input, hence it has no functionality for assigning multiple inputs to outputs. Therefore, we have designed and implemented the power router extending the smart outlet to have multiple inputs, and to be able to realize arbitrary assignment between power outputs and inputs.

Figure 6.2 is an outward appearance of the power router presenting the inputs, outputs, LEDs and interfaces of USB and Ethernet. Table 6.3 shows a summary of the specifications. We describe the implementation of hardware and software in the

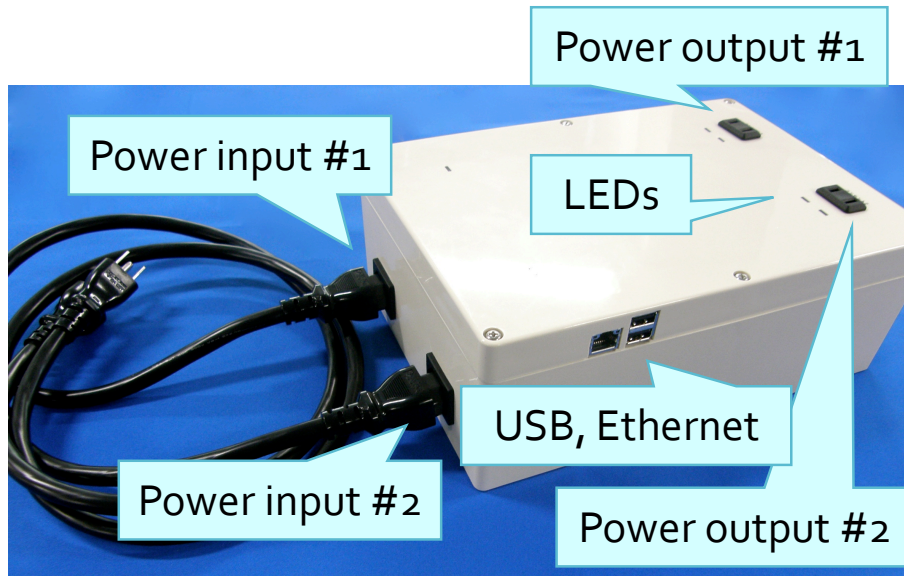


Figure 6.2 Developed power router.

following subsections, depending on the requirements definition shown above.

6.3.3.1 Implementation of Hardware

The major function of the physical layer is control of assignment between inputs and outputs depending on control messages from routing layer protocols. It also controls power maintaining the actual amount of consumed current on each power flow under the reserved value. Therefore it maintains control routes and current values that are decided by the upper layer.

The power router has two power inputs and two outputs. Some indicators are needed that present the current state of power inputs and power assignment for outputs, not only on-off states of outputs. Therefore, at each input and output, LED indicators are implemented to represent current availability of each power source and the current state of power assignment.

The router also has interfaces of USB 2.0 and Ethernet for communication capability and future extendibility. As a communication media, we have adopted Wi-Fi (IEEE 802.11n) via USB and 100Mbit/s Ethernet, while other media such as ZigBee can also be installed. The USB ports can also be used for other kinds of extension, such as adding environmental sensors or motion sensors for more advanced policy-based power control.

The power router has current transformers (CT), voltage sensors for measuring

Table 6.3 Specifications.

Items	Specs
Power sensing function	Real-time power consumption, Integral power consumption, current and voltage for each socket
Sampling frequency	20kHz
Resolution	12bit
Power control	Policy-based relay switching
Rated voltage	AC 100V (standard voltage in Japan)
Rated current	15A
Rated frequency	50/60Hz
Error	Under +/- 2%
Interface for extendibility	USB 2.0 (two ports)
Communication media	Wi-Fi 802.11n (via USB), 100Mbit/s Ethernet
Number of input/output	Two inputs / two outputs
Others	LED indicators (for representing the state of power allocation)
Operating System	Debian-based Linux

the amount of current and voltage, respectively. It computes instantaneous power consumption and integral power consumption by using measured values of current and voltage. Two-step auto-ranging (1A and 15A) achieves the low error rate of less than 2%.

Internal boards of the router are designed to be able to work properly if either one of the two inputs can be used, because both sources are not always available.

Figure 6.3 shows the hardware diagram of the power router. Mainly, the router consists of four relays, two CTs, voltage sensors, two sockets for outputs, two plugs for inputs, and six kinds of internal boards. The major functions of internal boards are as follows;

- Application board: this board is a commercial microcomputer board for embedded computing with an ARM microprocessor (ARM926, 400MHz), 128MB

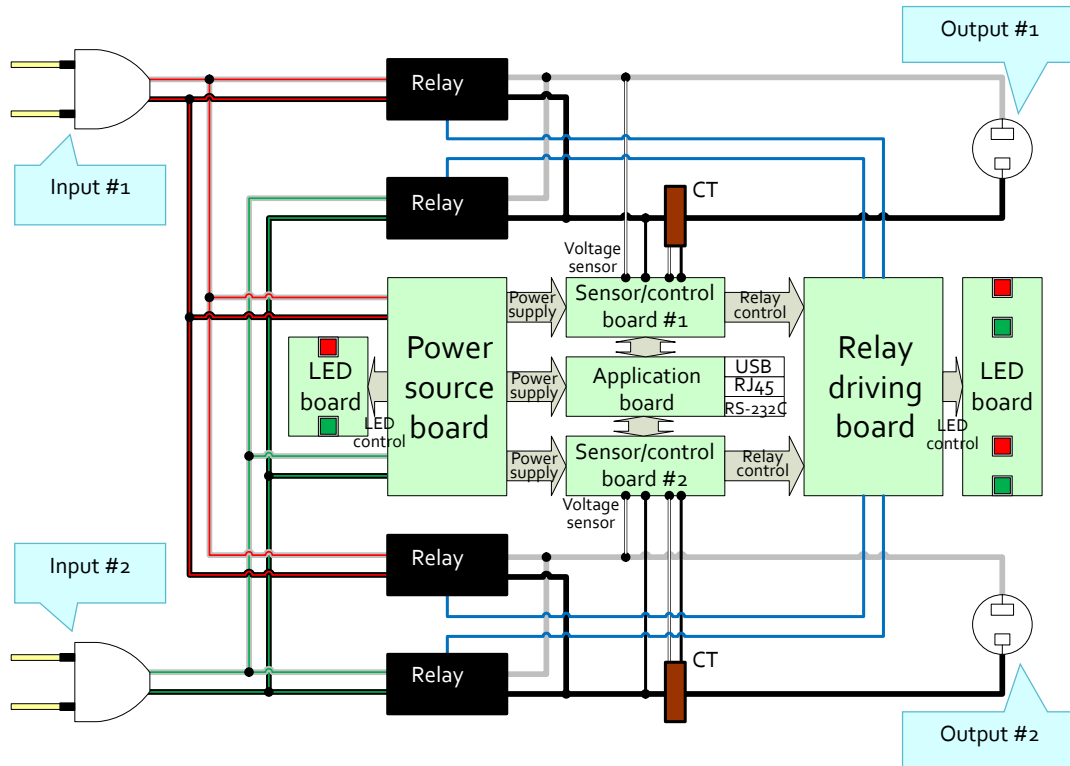


Figure 6.3 Hardware block diagram.

RAM and 32MB flash. It communicates with the sensor/control board to receive measured data and send relay control messages.

- **Sensor/control board:** This board measures power consumption, send relay control messages to the relay driver board, and communicates with the application board. It has a microprocessor of TM S320F28035 60MHz, and computes instantaneous power consumption and integrated power consumption from the measured current and voltage values. The router has two sensor/control boards in total for sensing and controlling of each output. It detects availability or unavailability of the two power sources from DC 5V input signals from the power source board that is converted from AC 100V of each input.
- **Relay driver board:** This board drives relays depending control messages from the sensor/control board. It has a mechanical relay on each socket for on-off control, and controls relays depending on the relay control messages from the sensor/control board.
- **LED board (inputs):** This board controls LEDs for indicating states of power sources. An LED on power source #1 is turned on green if source #1 is avail-

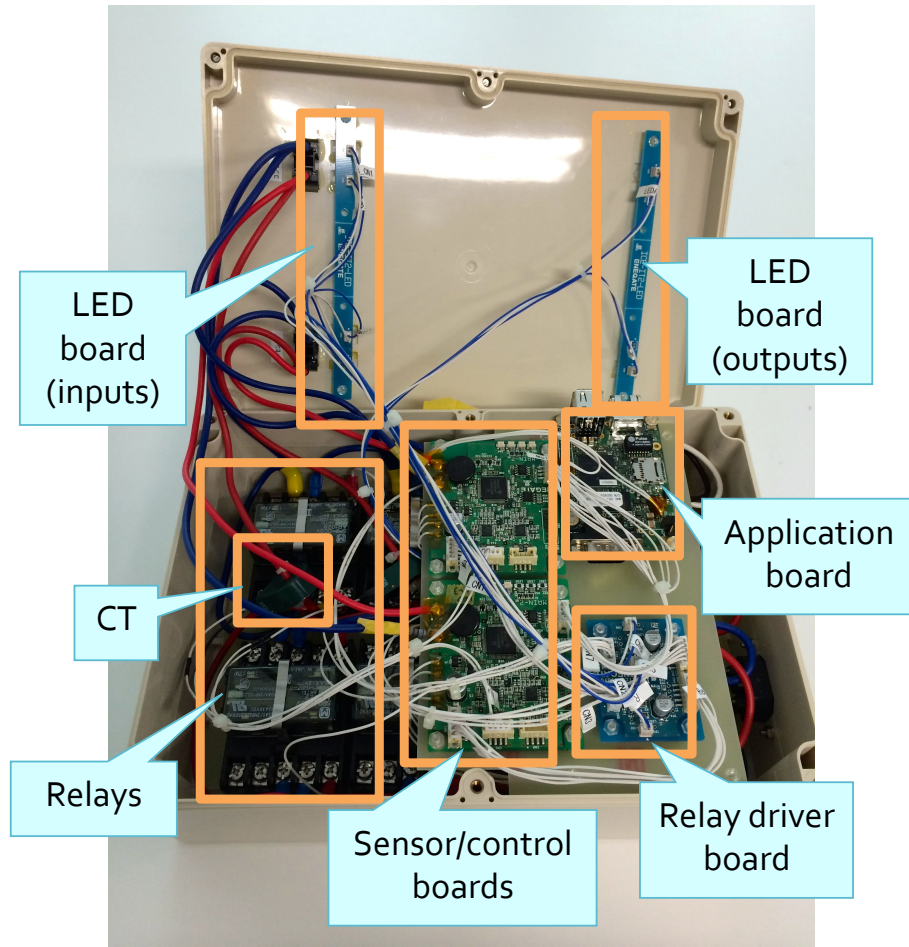


Figure 6.4 Internal hardware.

able, and turned off when unavailable. Similarly, an LED on power source #2 is turned on red when available, and turned off when unavailable.

- LED board (outputs): This board controls LEDs for indicating states of power assignment to each output. Each output has two LEDs that corresponding to input #1 and #2. The LED corresponding to input #1 is turned green when supplied from the input #1, while the LED corresponding to input #2 is turned red when supplied from the input #2, and both are turned off when the output is not assigned with sources.
- Power source board: This board generates power (DC 5V) from AC 100V of the two inputs, and supplies it to the other boards. The power circuit is implemented to be able to supply DC 5V to the other board is either input is available, because the both inputs are not always available.

The internal hardware of the router including these boards with wires is presented as Figure 6.4.

6.3.3.2 Implementation of Software

Upon the Debian-based Linux OS for embedded computing running on the application board, we have implemented software of a sensor/control interface process, a communication process with outer devices and a policy-based control process.

- Sensor/control interface process: The sensor/control interface communicates with the sensor/control board via the standard serial UART (Universal Asynchronous Receiver Transmitter) communication. It consists of mainly two functions.
 - Collection of sensed data: It collects sensed data from power sensors by polling from sensor/control boards. For future extendibility and easier cooperation with other devices, the sensed data are formatted in an XML-like manner; for example, the message shown below is sensed data including the amounts of integral power consumption, voltage, current and instantaneous power consumption, which means that output (load) #1 is supplied by source #2 and load #2 by source #1.

```
<root>
  <info>
    <kind>notice_wattmeter</kind>
    <time>201402201147323593</time>
  </info>
  <data>
    <load1>
      <wh>3727</wh>
      <volt>101.237</volt>
      <current>1.531</current>
      <watt>82.7</watt>
      <state>route2_ON</state>
    </load1>
    <load2>
      <wh>1274</wh>
      <volt>101.159</volt>
      <current>0.315</current>
```

```

        <watt>26.9</watt>
        <state>route1_ON</state>
    </load2>
</data>
</root>

```

- Relay control: It sends control messages to the sensor/control board for assigning power inputs to outputs based on the instruction messages via HTTP/CGI.
- Communication process with outer devices: The communication software works as an interface with other devices (power routers, smart outlets or servers) via HTTP/CGI, receiving policy updates and sending data from power sensors.
- Policy-based control process: This process decides control of relays depending on sensed data and defined policies. This process includes the circuit breaker function for maintaining the amount of consumed current under an upper limit, which is described Sec. 6.4.1, and an autonomous power control function for reacting sudden unavailability of power sources, which is described in Sec. 6.4.2.

6.4 Application to QoEn Routing and Autonomous Local Control

This section describes experimental implementation of global optimization realized by multiple power routers using a QoEn power routing protocol, and policy-based autonomous local control.

6.4.1 Global Optimization with QoEn Routing

To confirm that the developed power router works properly as the physical layer in the EoD protocol layers, we have implemented a prototype end-to-end power routing system with a software-based circuit breaker function for maintaining current consumption under the reserved amount by watching current consumption of each power flow.

As described in Section 6.3.1.1, a routing protocol computes the best matching and routes between appliances and power sources based on QoEn parameters in an end-to-end manner using some optimization algorithms (e.g. resource allocation al-

gorithms or shortest path algorithms), and reserves resources at each router located on the computed power route. Then the protocol informs to the physical layer of assignment between inputs and outputs that physically realizes the optimized power route; namely, relay configurations that assign inputs and outputs properly. The power router appropriately controls relays based on informed configurations from the routing protocol to realize QoEn routing.

One of the desired functions of the physical layer is maintaining the power route as decided by the routing protocol. We have considered that the software-based circuit breaker function alike to the one in the smart outlet described in Chapter 5 is applicable to maintain the consumed amount of current under the reserved value by the routing protocols. The power router watches current consumption on each power flow and compares the measured value with the reserved value of each flow. When the measured value exceeds baselines for configured time duration, it decides that the flow does not obey the reserved configuration and controls relays to shut off the flow.

The following is an instruction message used to control relays for allocating the input (power source) #2 to the output (load) #1 and the input #1 to the output #2, and to configure parameters of the circuit breaker function.

```
<root>
  <config>
    <load1>
      <state>route2_ON</state>
      <register_current>5</register_current>
      <watch_control1>
        <watch_percent>110</watch_percent>
        <watch_time>6000000</watch_time>
      </watch_control1>
      <watch_control2>
        <watch_percent>150</watch_percent>
        <watch_time>10000</watch_time>
      </watch_control2>
    </load1>
    <load2>
      <state>route1_ON</state>
      <register_current>7</register_current>
      <watch_control1>
```

```

        <watch_percent>130</watch_percent>
        <watch_time>7000000</watch_time>
    </watch_control1>
    <watch_control2>
        <watch_percent>160</watch_percent>
        <watch_time>8000</watch_time>
    </watch_control2>
</load2>
</config>
</root>

```

The **state** tag indicates states of relays. The **register_current** tag means the baseline amount of current. The **watch_percent** tag means a ratio compared with the baseline, and **watch_time** means a upper limit of time duration of exceeding. The parameters can be set in a two-step manner for flexible configuration suited for characteristics of current consumption of power routes and appliances. By setting the baselines as decided values by routing protocols via HTTP/CGI, the power router physically establishes the optimized power path and maintains the current consumption as a reserved value. This example means to assign power source #2 to output #1 and source #1 to output #2. The configuration of parameters means that the baseline for output #1 is 5A, and that the router shuts down the output when the current value exceeds 110% for 10 minutes or 150% for 10 seconds, regarding that the power route does not obey the reserved value. Therefore, the output #1 must be turned off when current is over 5.5A for 10 minutes, or over 7.5A for 10 seconds.

The relay states and parameters can also be manually configured. Figure 6.5 shows a prototype user interface for manually configuring power allocation and modifying parameters, implemented on the HTTP/CGI server. In this example, the configuration is same as the one in the control message described above.

To evaluate that the developed router has sufficient functions to work properly as the physical layer on an EoD system, we have confirmed that end-to-end power routing is realized using multiple routers. A QoEn routing protocol used in this implementation is *EoDresv*, proposed and implemented by Miyamoto et al. [60]. The EoDresv is designed as an extension and application of QoS routing protocols in computer networks; namely, RSVP-TE (Resource reSerVation Protocol - Traffic Engineering) and OSPF-TE (Open Shortest Path First - Traffic Engineering). In the implementation of EoDresv, QoEn parameters contain stability and capacity of

Input #2 is allocated to output #1

Parameters of the soft-based circuit breaker

Input #1 is allocated to output #2

Parameters of the soft-based circuit breaker

Settings for output #1	
Input selection	Input #1
Current (A)	5.000
1st threshold (%)	110
1st threshold time (ms)	6000000
2nd threshold (%)	150
2nd threshold time (ms)	10000

Submit

Settings for output #2	
Input selection	Input #2
Current (A)	7.000
1st threshold (%)	130
1st threshold time (ms)	7000000
2nd threshold (%)	160
2nd threshold time (ms)	8000

Submit

Figure 6.5 Configuring power allocation and parameters of the software-based circuit breaker function via HTTP/CGI.

power sources, and stability, required power amount and priority of power consuming device. EoDresv computes the best matching and route with the reserved amount of current, based on these QoEn parameters, and sends routing messages to the power routers. By this process, the reserved power route is established. EoDresv computes the best matching between the sources and alliances, and decides the best route to connect the matched source and appliance using optimization algorithms. In this experimental configuration, the topology of the graph is rather simple because it

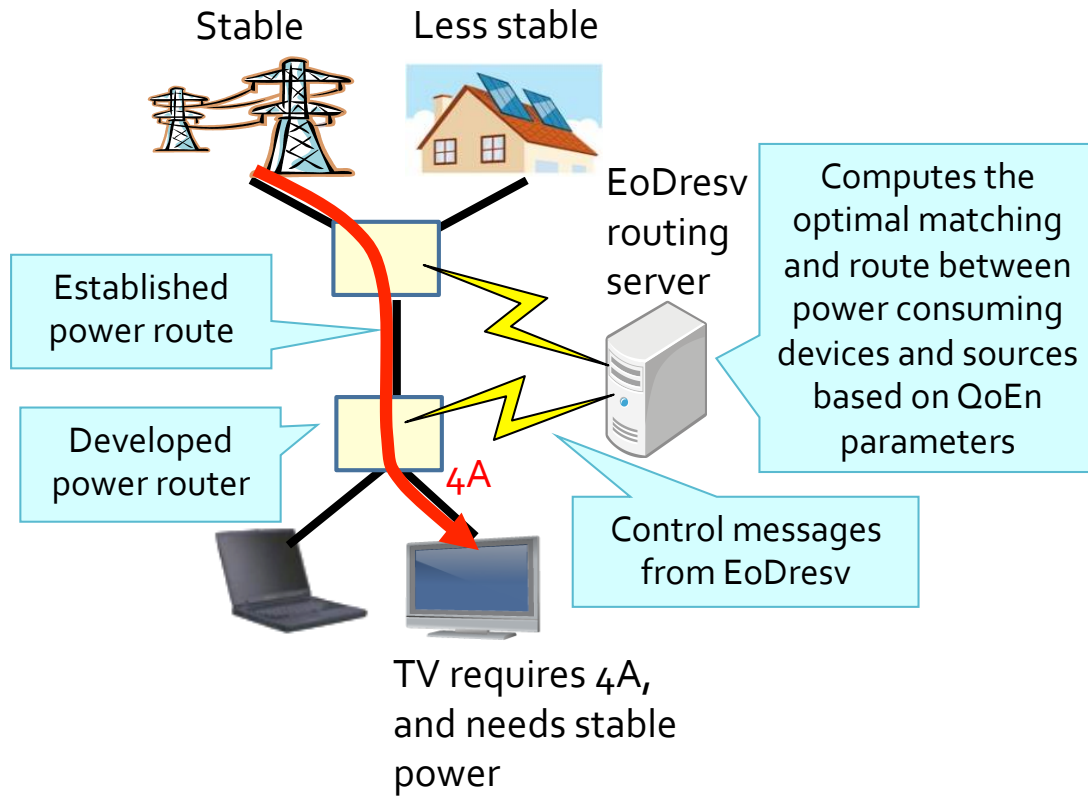


Figure 6.6 Establishing an end-to-end power route by using the power router and the EoDresv routing server.

contains no cycles, however the EoDresv protocol can be applied in networks with more complicated topology.

We have combined the simulation system with the developed power router to physically establish the end-to-end power route. Figure 6.6 shows an entire image of the experiment. The routing server used in the experiment is the extended version of the one in the simulation of the EoDresv. The routing server has an entire topology including costs of each edge in the graph, and QoEn parameters of both the power consuming devices and power sources, that includes power consumption, power stability, priority among the appliances, and power capacity. EoDresv computes the best matching and routes between power sources and power consuming devices, and sends control messages to the power router via HTTP/CGI, with `state` value that corresponding to the configuration of the relays that realizes the power route that is computed by a routing protocol. The routers control relays depending on received control messages. By configuring baseline values of the soft-based circuit breaker

function as reserved value of current by EoDresv, the power router keeps and controls each power path does not consume more current than the reserved amount. In this example scenario, the TV that requires stable power with 4A is properly supplied with power from the stable commercial power source; and by configuring the baseline as 4A at the power router on the established power path, the amount of current consumed by the power path is kept under the reserved value.

6.4.2 Local Policies for Autonomous Control

One of the major application of the power router is policy-based control of Vehicle-to-Home (V2H), where an electric vehicle works not only as transportation but as a huge battery. For example, it is expected that an EV is charged in off-load period such as midnight, and it works as a power source on on-load period to reduce peak load. As experimental implementation of autonomous local control, we have assumed the following scenario; an appliance has been supplied with power from an electric vehicle. when the power from the EV suddenly becomes unavailable with disconnection of the EV from the power network, the power router autonomously allocates another power source to the appliance for reducing unavailability of the appliance.

As described in Section 6.3.3.1, the power source board divides each input of AC 100V to two power lines with DC 5V, and the sensor/control boards detects availability and unavailability of each power source using a hardware logic circuit, and communicates obtained information to the application board. On detecting voltage drop and the unavailability of the EV, the local control software implemented on the application board immediately sends an instruction message to the relays via the sensor/control interface, to switch power assignment to commercial power source to the appliance, if the other commercial power source is available. Using this policy, the power consuming devices connected to this router temporarily be supplied with power from available power sources, until the QoEn routing system recomputes a new end-to-end power route that is optimized for the device and available sources, and establishment of the computed power route is physically completed.

We have made an experiment of the above autonomous local control of switching between power from an EV and power from the commercial power source, using a incandescent lamp and a V2H-ready EV (i-MiEV by MITSUBISHI Motors.) The experiment is done as shown in Figure 6.7; in the scenario of the experiment, first the light is supplied with power from the EV. When power from the EV is suddenly

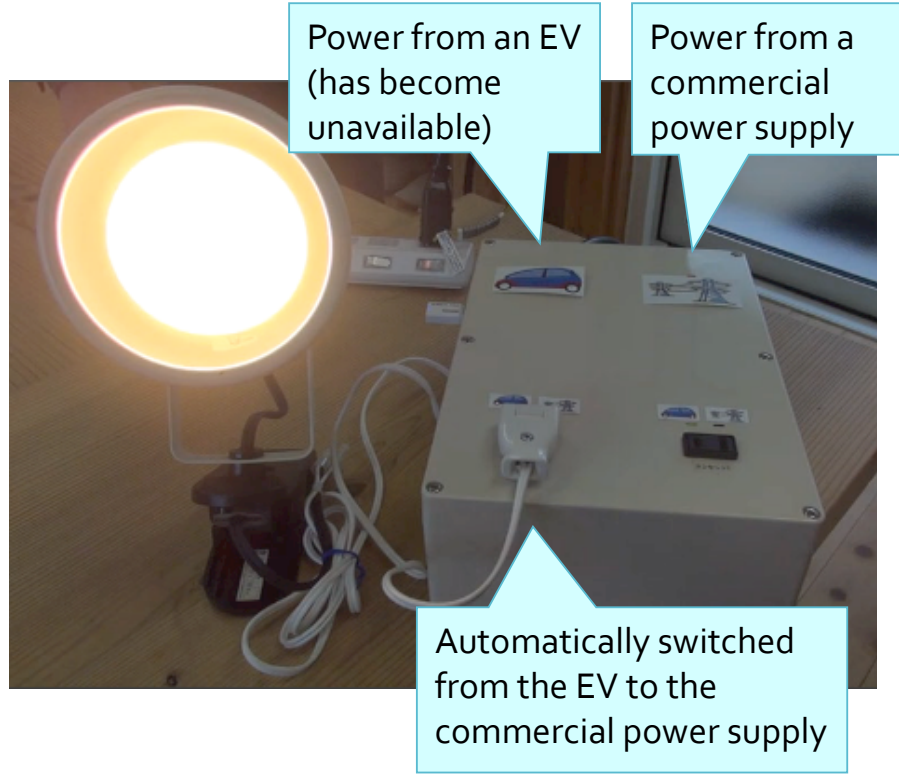


Figure 6.7 Policy-based autonomous switching between a commercial power source and an EV by the power router.

become unavailable, the router detects voltage drop of the EV and autonomously alternates power allocation for the light to the commercial power source for reducing unavailability of the lamp. We have confirmed that the policy works as intended.

6.5 Concluding Remarks

In this research work, we have designed and implemented a policy-based power router with power sensors for both end-to-end QoEn routing and autonomous local control. As experimental implementation, we have confirmed that an end-to-end power route through two power routers is established as intended, combined with a routing server with the EoDresv routing protocol. The software-based circuit breaker function implemented on the router is able to maintain the amount of consumed current under a reserved value. We have also implemented an experimental policy for autonomous local control that switches power allocation between two power sources when detecting sudden unavailability of sources. While it was shown that it requires approximately

four or five seconds to establish an end-to-end path in a power network with nine power routers in the simulation of the EoDresv [60], the local policy-based switching executed on the router requires approximately 0.2 seconds after detecting the unavailability of sources; therefore, this local optimization policy suitably complements the global optimization as intended. Hence we conclude that the developed power router has sufficient functions as a prototype for policy-based power management for both QoEn routing and autonomous local control.

Chapter 7

A Power Allocation Management System Using an Algorithm for the Knapsack Problem

7.1 Introduction

The importance of intelligent power management has been widely discussed. Today's power management is done mainly by handwork, therefore users feel troublesome to manually control appliances, even if they understand that their power-saving activities contribute to save electricity bills. It is desirable for users that some power management system in home automatically controls appliances under power management policies, minimizing harmful effect to QoL of users; for example, when power utilities cannot supply sufficient power because of some power failure and total power usage in home is limited, the power management system optimizes usage of appliances and controls appliances to keep total power consumption under the limit value. The major issue in such control is that the system has to decide which appliance should be controlled, because all the appliances are not necessarily usable with the limit of available power. Some architectures are proposed that keeps total power consumption under a certain value in a way where the undesired effect to users' QoL is minimized, optimizing power allocation to appliances (e.g. *i-Energy* [56] and a concept of *optimal allocation of energy* [68]).

Such optimization can be treated as the knapsack problem [46]. In the case of

power allocation, an appliance is regarded as an item with *profit* (namely, how much the appliance contributes for user's QoL) and *size* (power consumption of the appliance). The threshold of total power consumption is *capacity* of a knapsack. The objective is to maximize the total profit of selected items, keeping total power consumption under the threshold.

We have implemented the dynamic-programming algorithm for the knapsack problem on the smart outlet and a controller, and have evaluated time needed for computation of optimal allocation, communication among the outlets and the controller, and controlling relays, as fundamental research work for realizing the optimal allocation of energy. The implementation and evaluation are done with a system with single outlet, and a system with five outlets and a single controller. The controller collects power consumption data of appliances from the smart outlets, computes the optimal power allocation using the dynamic-programming based algorithm for the knapsack problem, and sends relay control messages to the smart outlets that realize the optimal power allocation obtained by the knapsack algorithm. To the best of our knowledge, this is the first literature that describes the implemented system utilizing a knapsack algorithm for power allocation management.

This chapter is consisted as follows; Section 7.2 refers related work. Section 7.3 presents basic concepts and a problem formulation of power allocation as the knapsack problem. In Section 7.4, we discuss our implementation of the developed power allocation system. Section 7.5 describes evaluations of the developed system with five outlets and single controller. Section 7.6 concludes this chapter.

7.2 Related Work

There has been simulation-based or theoretical research work regarding power allocation as optimization problems [67]. For example, Kumaraguruparan et al. considered residential task scheduling problems under dynamic pricing as the multiple knapsack problem, where task is regarded an item and time periods as knapsacks [49]. Sianaki et al. formulated the optimal power allocation as the knapsack problem, and presented results obtained by numerical simulations [75]. Yu and Chau considered complex-demand knapsack problems assuming AC electrical systems [98]. Another approach is priority-based; Kato et al. [43] proposed a priority-based system for efficient power allocation in EoD power networks, and evaluated it in real-life situations. Yokohata et al. proposed a priority-based power allocation algorithm in EoD power networks,

and implemented a power allocation system extending Power-over-Ethernet protocols [95]. The major difference between the knapsack-based approach and the priority-based approach is that the former explicitly defines user's profit (Quality-of-Life) as an additive value. There have been simulation-based studies [51] that models power allocation as a mixed integer programming [78]. Fujita et al. proposed a system that allocates power hierarchically in layered-tree power networks [30]. As for commercial products, there has been a system that sequentially turns off appliances with lower priority, based on pre-set priority [23].

7.3 Basic Concepts and Problem Formulation

7.3.1 Basic Concepts

Energy-on-Demand (EoD) [56] is a new-generation power network proposed recently, where power consuming devices explicitly send request messages to power management systems for being supplied with power, based on *Quality-of-Energy* (QoEn) parameters such as the amount of power, cost, etc., and they would be supplied with power only when some mediation system in the network accepts their requests. Our developed system has been designed and implemented as a centralized optimization system where processing is across multiple outlets and a controller, and can be regarded as an EoD system that uses the knapsack algorithm as the mediation scheme. In this prototype implementation, we have focused on control of instantaneous power assuming to reduce peak load.

7.3.2 Problem Formulation

The power allocation we assume can be treated as the knapsack problem in the following way; here the number of appliances is n . An appliance is regarded as an item i ($1 \leq i \leq n$) with *profit* p_i (contribution of the appliance for user's QoL), and *size* w_i (power consumption of the appliance). The threshold of total power consumption is *capacity* c of a knapsack. A variable $x_i \in \{0, 1\}$ means that if $x_i = 1$ the appliance i is turned on, while $x_i = 0$ means off. The objective is to maximize the total profit of selected items, keeping total power consumption under the threshold. The below is a formulation of the knapsack problem;

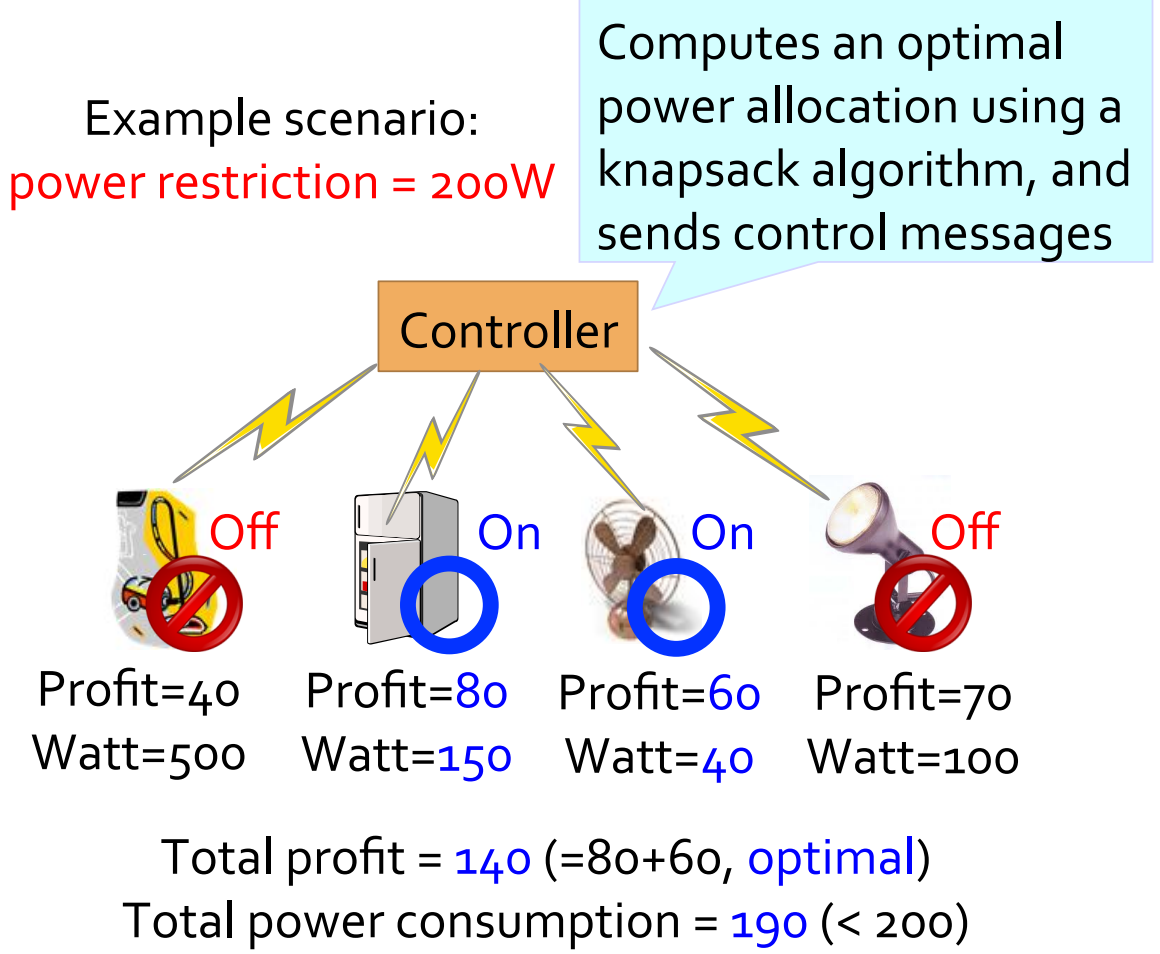


Figure 7.1 Power allocation for each appliance utilizing the knapsack algorithm.

$$\begin{aligned} & \max \sum_{i=1}^n p_i x_i \\ & \text{s.t. } \sum_{i=1}^n w_i x_i \leq c, x_i \in \{0, 1\} \end{aligned}$$

Though the knapsack problem is NP-hard [32], it has been shown that the problem can be solved in $O(cn)$ using the well-known dynamic-programming based algorithm [46].

Figure 7.1 shows an overview of the power control as the knapsack problem including a controller and appliances. The controller decides on-off control using an algorithm for the knapsack problem, and controls power supply to each appliance to keep total power consumption under the upper threshold; namely, the controller

turns on an appliance i if $x_i = 1$ in the obtained optimal solution, and turns off if $x_i = 0$, respectively. In the example shown, the upper limit is set to be 200W, and the optimal profit is 140 obtained by turning on a refrigerator and a fan, and turning off a cleaner and a light. Total power consumption is 190W, which is kept less than 200W of the upper limit.

7.4 System Implementation

In this section, we will describe system implementation focusing on hardware, a communication network, data formats, an algorithm for the knapsack problem and control software.

7.4.1 Hardware

7.4.1.1 Smart Outlet

The smart outlet used on the developed system is the extended version of the one described in Chapter 5; we have improved the outlet to be more compact and practically designed, and have extended computational resources. It has four sockets, and has functions of power sensing (voltage, current and power) and controlling relays for each socket. The Raspberry Pi [86] is used as the application board; an ARM-based microcomputer with CPU (ARM1176JZF-S 700MHz) and RAM (256MB or 512MB), enabling implementation of various software and control policies on the outlet. It has a built-in Ethernet interface and a Wi-Fi interface installed via USB. It communicates power consumption data and control messages via Wi-Fi and TCP/IP communications with outer devices such as a controller, a server or other smart outlets. Rated voltage and current is 100V and 15A, respectively. A/D conversion of measured amount of power is done with 20kHz sampling frequency and 12bit resolution. Table 7.1 shows specifications. Figure 7.2 is an outward appearance of the smart outlet.

7.4.1.2 Controller

In our developed system with multiple smart outlets, a centralized controller gathers power consumption data from the outlets, computes the optimal allocation and sends control messages to the outlets. The controller has the similar computational resources as the outlet (except the amount of memory). Table 7.2 shows specifications of the

Table 7.1 Specifications of the smart outlet.

Items	Specs
Measurement	Instantaneous power, integrated power, current and voltage
Sampling frequency	20kHz
Resolution	12bit
Error	Under 2%
Power control	On-off control for each socket
Number of sockets	Four
Rated voltage	AC 100V
Rated current	15A (in total of all the four sockets)
CPU	ARM1176JZF-S 700MHz
RAM	512MB
Storage	8GB
Communication media	Wi-Fi and Ethernet
Interface	USB 2.0
OS	Debian-based Linux

controller. Figure 7.3 shows an entire image of the system consisting of multiple outlets and a controller.

7.4.2 Network

The Wi-Fi used in the network is IEEE 802.11n with maximum bandwidth of 300Mbps using 2.4GHz of frequency and is protected with WPA2-AES encryption for easier coordination with ordinary information devices such as PC, tablets or smartphones. As a transport layer protocol, we have chosen TCP for reliability, dependability and safety, because the system does control electricity actively, not only gathering data of power consumption.

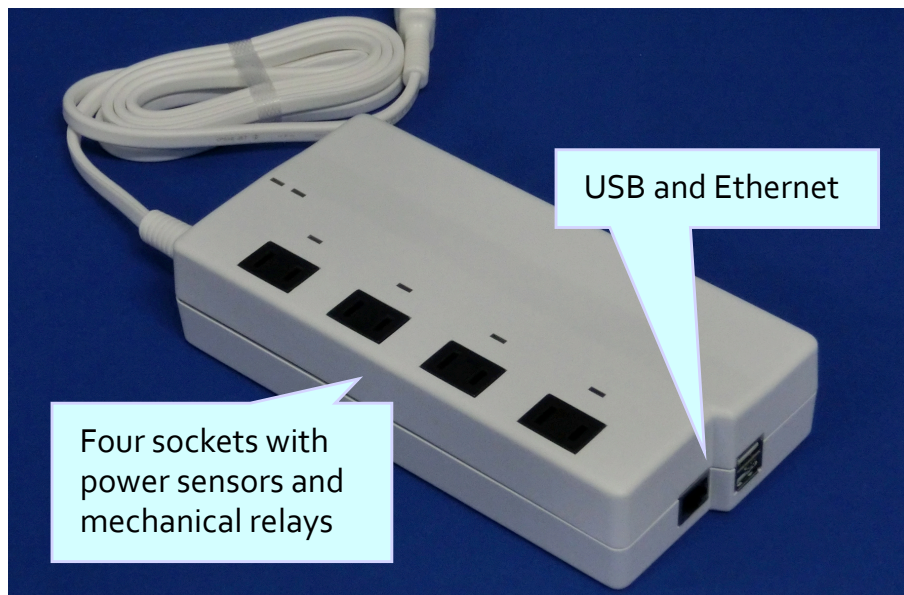


Figure 7.2 An outward appearance of the smart outlet.

Table 7.2 Specifications of the controller.

Items	Specs
CPU	ARM1176JZF-S 700MHz
RAM	256MB
Storage	8GB
Communication media	Wi-Fi
Interface	USB 2.0
OS	Debian-based Linux

7.4.3 Data Formats

- Relay Control Message: The below is an example of `command_socket`, a message for controlling relays. In this example, relay #1 and #4 should be turned on and #2 and #3 should be off.

```
<root>
  <info>
    <kind>command_socket</kind>
  </info>
```

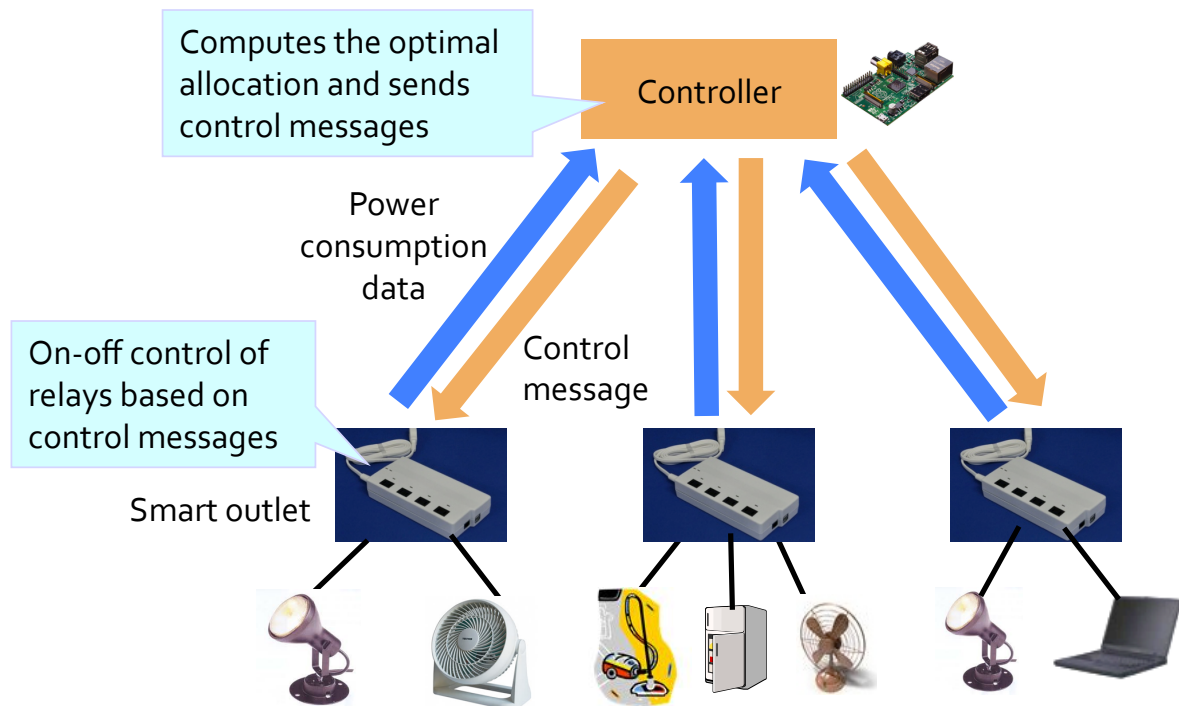


Figure 7.3 System consisting of multiple outlets and a controller.

```

<data>
  <socket1><state>ON</state></socket1>
  <socket2><state>OFF</state></socket2>
  <socket3><state>OFF</state></socket3>
  <socket4><state>ON</state></socket4>
</data>
</root>

```

- **Measured Power Consumption Data:** The below is an example of `notice_wattmeter`, which is used for sending measured power consumption data. This example means that integrated power on socket #1 is 20954Wh, instantaneous voltage is 102.070V, current is 1.551A, instantaneous effective power is 84.6W, and the relay at the socket has been turned on.

```

<root>
  <info>
    <kind>notice_wattmeter</kind>
    <time>20120909104854039</time>
  </info>
  <data>

```

```
<socket1>
  <wh>20954</wh>
  <volt>102.070</volt>
  <current>1.551</current>
  <watt>84.6</watt>
  <state>ON</state>
</socket1>
<socket2>
  <wh>2536</wh>
  <volt>102.085</volt>
  <current>0.013</current>
  <watt>4.1</watt>
  <state>ON</state>
</socket2>
<socket3>
  <wh>2499</wh>
  <volt>102.073</volt>
  <current>0.242</current>
  <watt>14.2</watt>
  <state>ON</state>
</socket3>
<socket4>
  <wh>1219</wh>
  <volt>102.080</volt>
  <current>0.081</current>
  <watt>5.1</watt>
  <state>ON</state>
</socket4>
</data>
</root>
```

7.4.4 Dynamic-programming Based Algorithm for the Knapsack Problem

Algorithm 2 shows the well-known dynamic-programming based algorithm for the knapsack problem. As mentioned above, each appliance corresponds to an item i ,

Algorithm 2 Dynamic-programming Based Algorithm for the Knapsack Problem [46]

```

for  $j = 0$  to  $c$  do
     $P(0, j) = 0$ 
end for
for  $i = 1$  to  $n$  do
    for  $j = 0$  to  $c$  do
        if  $j \geq w_i$  then
             $P(i, j) = \max\{P(i - 1, j - w_i) + p_i, P(i - 1, j)\}$ 
        else
             $P(i, j) = P(i - 1, j)$ 
        end if
    end for
end for

```

measured power consumption to size w_i , and an importance of the item for users to profit p_i . Here profit is set to the appliance by users. $P(i, j)$ means the maximum total profit that can be achieved using item 1, ..., i and a knapsack with capacity j . The optimal solution is $P(n, c)$, and a set of items that achieves $P(n, c)$ corresponds to alliances that should be turned on. The time complexity of this algorithms is $O(cn)$. Therefore, if c does not depend on n the algorithm is sufficiently fast even when n is large.

7.4.5 Control Software

The developed software, written with the C-language, first measures power consumption of each appliance, and then computes the optimal power allocation that maximizes a total profit using the dynamic-programming based algorithm. The outlet and controller have a database of “profit” parameters of the items. The control software parses the XML-like data of power consumption collected from the smart outlets, and uses the values of power consumption as “size” of items. Here users can set another parameter, “necessity”, which means whether the appliances is needed or not by the user; when the parameter is set to be zero (namely, the appliance i is not needed at that time by the user), the profit of the item is considered to be zero, and the item is excluded from the set of the items. Therefore, the dynamic-programming based



Figure 7.4 Implemented system with a single outlet.

algorithm computes an optimal power allocation without the excluded item. Hence the parameter “necessity” can be regarded as “demand” from the appliance in the concept of the EoD power network.

7.5 Experiments and Evaluations

7.5.1 Case with Four Appliances

Figure 7.4 shows an experimental system with the single smart outlet and appliances. We have measured that computation of an optimal solution needs around 2.914 milliseconds, and on-off controls of four relays require at most 29.3 milliseconds. In total, time required is $2.914 + 29.3 = 32.214$ milliseconds. These results suggest that computation and relay control are sufficiently fast even if done with an embedded computer.

7.5.2 Case with 20 Appliances

We have experimented and evaluated in the situation including five outlets and a controller, assuming a situation in ordinary homes. In this situation, the controller

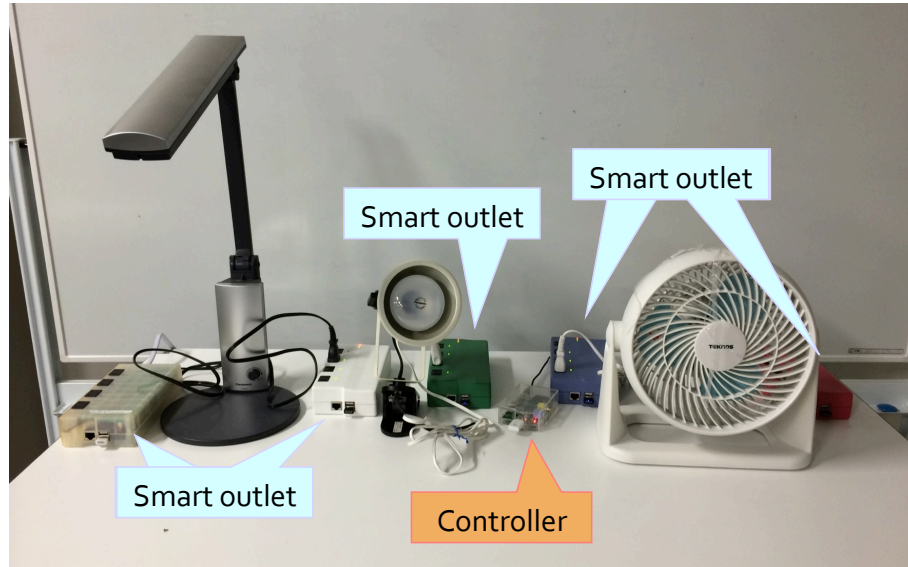


Figure 7.5 Implemented system with five outlets and a controller.

with computational resource as the outlets gathers power consumption data from the outlets, computes optimal power allocation and send control messages to each outlet via Wi-Fi (802.11n). Each outlet completes relay control based on the received messages. We have measured time needed to execute the dynamic-programming based algorithm, sending control messages to smart outlets and control relays on each outlet. The parameter c is set to be 2000 (that is, an upper limit of total power consumption is set to be 2000W). Figure 7.5 shows an experimental system with five smart outlets that is able to handle up to 20 appliances.

We have measured that computational time of the dynamic-programming based algorithm with 20 items is around 10 milliseconds. Using Wi-Fi 802.11n as a communication media, 330.5 milliseconds is measured to be averagely required to send a control message from a home controller to all the smart outlets. As described above, relay control requires at most 29.3 milliseconds on each smart outlet. Therefore, time required to complete computation of optimal allocation and control is about $10 + 330.5 + 29.3 = 369.8$ milliseconds. These measurements and estimations also suggest that computation and control can be done practically fast on an embedded microcomputer such as the home controller.

Figure 7.6 presents an experimental result of software execution. In this example, the total power consumption is 1995W, obeying the upper limit of 2000W. Note that the algorithm turns off an air conditioner as a result of the computation though the

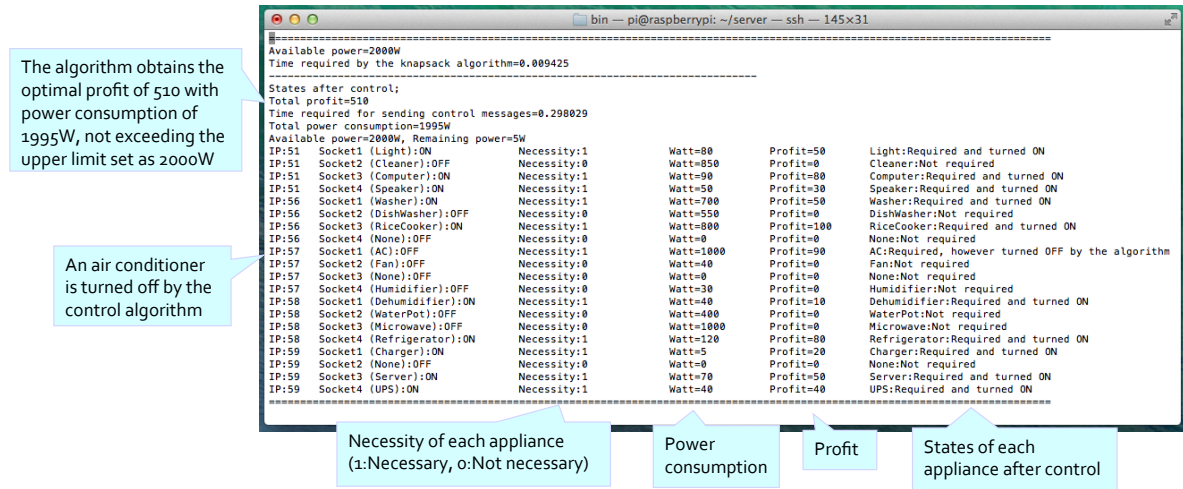


Figure 7.6 Example of software execution for power allocation and control.

“necessity” parameter of the air conditioner is set to be one. This can be regarded that the power request from the air conditioner was rejected by the allocation algorithm. Computation of optimal allocation required approximately 9.4 milliseconds, and communicating control messages required approximately 298 milliseconds.

7.6 Concluding Remarks

To optimize power allocation to appliances in a way that maximizes QoL of users with limited total power consumption, we have implemented a prototype power management system using smart outlets and a controller, which measures power consumption of appliances, computes optimal power allocation to appliances, and controls relays. The optimal power allocation is computed with the dynamic-programming based algorithm for the knapsack problem that maximizes total profit under the limit of total power consumption. We have experimented and evaluated time duration required for computation of an optimal power allocation by the algorithm, sending control messages to the smart outlets and control relays. The result shows that computation, communication and controls can be done in practical time with embedded microcomputers using the smart outlets and the controller.

Chapter 8

Conclusion

In this dissertation, we have discussed the design and analysis of algorithms for graph exploration and resource allocation problems, and their application to energy management. This chapter describes summary and future work.

In Chapter 3, we have studied the online graph exploration problem on two graph classes, related to various applications including the energy-efficient exploration by mobile robots on unknown terrains, Web crawling by search providers, or efficient topology estimation of smart micro grids where node connectivity or resource availability change dynamically. First, we have given a tight competitive ratio of $\frac{1+\sqrt{3}}{2}$ for the problem on cycles. We have also studied the problem on unweighted graphs and have given a tight bound of 2. For planar graphs, the best known upper bound is 16, as mentioned in Chapter 1. Though several further studies have been done recently [58] and the lower bound for planar graphs has been improved to $\frac{5}{2}$ [22], there still remains a large gap between these upper and lower bounds. Narrowing the gap is a challenging problem. There also have been studies on an extended version of the graph exploration problem, where multiple searchers explore a graph, not only a single searcher [34]. Another future work is to consider randomized algorithms to break the deterministic lower bound.

We have considered two kinds of resource allocation problems on bipartite graphs with assignment restrictions in Chapter 4, aiming for efficient power allocation on power networks with distributed sources. First, we have defined the multiple knapsack problem with assignment restrictions and capacity constraints, have designed a $\left(1 + \frac{2}{k+1} + \epsilon\right)$ -approximation algorithm, and have shown that an integrality gap of the LP formulation used in the algorithm is $\left(1 + \frac{1}{k} - \epsilon\right)$ where ϵ is an arbitrary small positive constant. Second, we have defined the splittable resource allocation

problem with assignment restrictions, and have gained the $n^{1-o(1)}$ -inapproximability, not only it is strongly NP-hard. As future work, it would be a challenging problem to fill the gap between upper bounds and lower bounds. Another directions are pursuing efficient algorithms for the problems with more realistic conditions such as edge capacities, or the problems on extended graph classes. It also should be interesting to formulate the problem in an online manner, assuming the situation where power consuming devices and power sources appear and disappear dynamically .

The design and implementation of a smart outlet for policy-based power management have been described in Chapter 5, which aims to reduce user's handwork required for power-saving activities and for efficient power control to realize optimal allocation of energy. The smart outlet has been designed to have an accurate power-sensing function, to have sufficient computational resources to execute various policy-based power management, and to be used safely in a real-life situation. As applications of policy-based power management, the authors have implemented and evaluated a software-based circuit breaker, and a function of reducing overlaps of inrush currents caused by multiple appliances. We also have implemented a smart outlet network as a platform for energy-aware services utilizing various sensor information. The smart outlet network has been deployed in the real-life environments; the one-room apartment and the multi-room apartment. Especially, in the one-room apartment the developed system continuously gathers power consumption data of all the appliances in the real-life environment. We have learned that highly frequent gathering (0.5 seconds) of power consumption data helps us to analyze characteristics of appliances in detail, and to grasp daily behavior of the resident. The following energy-aware services have been implemented in the system; scheduled reduction of stand-by power, on-off control using a motion sensor, coordination between an air conditioner and an air circulator based on power consumption data in a centralized manner, coordination between two different humidifiers based on environmental information in a distributed manner. Therefore, we conclude that the developed smart outlet has sufficient functions for policy-based power management utilizing behavior information and environmental information. Major future work is as follows; the developed outlet is capable of only on-off control, therefore Koyama et al. implemented a system for controlling appliances using a universal IR controller via its USB port [47] for more flexible control. When so-called smart appliances become commonly available in future, it is expected to control appliances in more straightforward and flexible ways using protocols for controlling appliances such as ECHONET Lite [84] or

ZigBee Smart Energy Profile [99]. The software-based circuit breaker function is able to watch the current value comparing to the allocated value; however, their parameters need to be determined by other schemes. Power allocation should be dynamically done considering effect to QoL of users, and we are studying methods for deciding the allocation. We are also considering to extend the system for multiple residents; when only a single resident lives in a home, it is rather simple to design various policies, and the preference-based control is relatively easy. However, different users should prefer different policies. We are considering that some policy generation method should be helpful, such as the one proposed by Yoshihisa et al. [97]. Also it would be useful and interesting to implement policies based on the combination of different kinds of sensor information; for example, combination of a motion sensor and a light sensor would provide much more accurate detection of presence and absence of a resident. Another research topic is to test and evaluate different communication media. In this system we have chosen Wi-Fi, however there are several other possible candidates. If we choose ZigBee, it should surely require less power consumption than Wi-Fi, however we might not obtain the highly-frequent data collecting capability as trade-off.

In Chapter 6, we have designed and implemented a policy-based power router with power sensors for both end-to-end QoEn routing and local autonomous control. As experimental implementation, we have confirmed that end-to-end power routes through two power routers are established as intended, combined with a routing server with an EoDresv routing protocol. The software-based circuit breaker function on the router is able to maintain an amount of consumed current under a reserved value. We have implemented an experimental policy for local autonomous control that switches power allocation between two power sources when detecting sudden unavailability of sources. We have also considered and constructed a system that controls an EoD-ready EV charger (H2V) and a discharger for QoEn routing (V2H) in a policy-based manner using the developed router, and have experimented with an autonomous policy that switches between power from an EV and commercial power source depending on availability of the sources. One of the future research topics is to evaluate the developed router in a network with more general topology such that includes cycles, not only with two power routers connected serially. In the prototype policy implementation, we have evaluated QoEn routing and local autonomous control independently, and have not yet considered the system with both global and local controls. We are considering to implement and evaluate the QoEn power network where both global and local policies work coordinately, based on the generation method of global and

local policies in EoD systems [31]. Fully distributed and autonomous power routing systems are expected to be constructed by implementing EoDresv protocol on each power router in networks, similar to the routing protocols on the routers in communication networks. It would be also interesting to use local power information sensed by the power sensor of the power router as cost parameters in computation of the optimal routes between the sources and appliances. In the current implementation, switching between power inputs requires a momentary power interruption caused by mechanical relays. We are considering to optimize both hardware and software, or to install a battery as a power buffer. Another research direction is to implement various kinds of policies; for example, the policy that switches charging and discharging an EV with an estimated remaining amount of a battery from measured data at the charger and the router.

In order to maximize QoL of users with limited total power consumption, we have implemented a power allocation system that measures power consumption of appliances, computes optimal power allocation to appliances, and controls relays utilizing a smart outlet, as described in Chapter 7. The optimal power allocation is computed with the dynamic-programming based algorithm for the knapsack problem that maximizes total profit under the threshold of total power consumption. We have made experiments and evaluated time and memory required for computation, communication and relay control. The result shows that all of the operations can be done in practical time with embedded microcomputers in the smart outlets and the controller. As future work, we are considering to implement the system on the smart outlet network in real-life environments described in Chapter 5, and evaluate effects of the system for QoL of users, comparing with some other approach such as priority-based one. In order to do that, one of the most challenging problems is that we have to decide the profit values of appliances in dynamic situations. We are considering to decide them based on patterns of user's activities and power consumption data of appliances collected in the real-life environments, to apply a probabilistic method for human behavior estimation from and power consumption patterns of appliances [94], or to adopt the analytic hierarchy process as proposed by Sianaki et al. Another essential problem is that there are some dependencies among appliances, for example it should not be practical to use a cleaner without a light. The power allocation problem with such constraints can be modeled as the precedence constraint knapsack problem, another variant of the knapsack problem, and we are considering to implement an algorithm for the problem [46] in our developed system.

Throughout this dissertation, we have discussed the design, analysis and application of algorithms for energy management in both theoretical and practical approaches. We hope that this research work will contribute to easier and more efficient energy management, especially in future daily lives of people.

References

- [1] Keiichi Abe, Hiroshi Mineno, and Tadanori Mizuno. Development and evaluation of smart tap type home energy management system using sensor networks. In *Proceedings of 2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1050–1054. IEEE, 2011.
- [2] Joep Aerts, Jan Korst, and Frits Spieksma. Approximation of a retrieval problem for parallel disks. In *Proceedings of the 5th Italian Conference on Algorithms and Complexity (CIAC)*, pp. 178–188, 2003.
- [3] Joep Aerts, Jan Korst, Frits Spieksma, Wim Verhaegh, and Gerhard Woeginger. Random redundant storage in disk arrays: complexity of retrieval problems. *IEEE Transactions on Computers*, Vol. 52, No. 9, pp. 1210–1214, 2003.
- [4] Agency for Natural Resources and Energy. Energy White Paper 2012. <http://www.enecho.meti.go.jp/topics/hakusho/2012energyhtml/>. (In Japanese).
- [5] Daniel Akselrad, Victor Petcu, B Romer, Andreas Schmid, Denis Bytschkow, and Maximilian Engelken. Making home energy usage transparent for households using smart meters. In *Proceedings of IEEE the 1st International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pp. 150–153. IEEE, 2011.
- [6] Susanne Albers and Monika R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, Vol. 29, No. 4, pp. 1164–1188, 2000.
- [7] Constantinos Marios Angelopoulos, Sotiris Nikolettseas, Theofanis P Raptis, Christoforos Raptopoulos, and Filippas Vasilakis. Efficient energy management in wireless rechargeable sensor networks. In *Proceedings of the 15th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pp. 309–316. ACM, 2012.
- [8] Yuichi Asahiro, Eiji Miyano, Shuichi Miyazaki, and Takuro Yoshimuta. Weighted nearest neighbor algorithms for the graph exploration problem on cycles. *Information Processing Letters*, Vol. 110, No. 3, pp. 93–98, 2010.

- [9] Giorgio Ausiello, Vincenzo Bonifaci, and Luigi Laura. The on-line asymmetric traveling salesman problem. In *Proceedings of the 9th International Workshop on Algorithms and Data Structures (WADS)*, pp. 306–317. Springer, 2005.
- [10] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. Algorithms for the on-line travelling salesman. *Algorithmica*, Vol. 29, No. 4, pp. 560–581, 2001.
- [11] Baruch Awerbuch, Margrit Betke, Ronald L. Rivest, and Mona Singh. Piecemeal graph exploration by a mobile robot. *Information and Computation*, Vol. 152, No. 2, pp. 155–172, 1999.
- [12] Maxim A. Batalin and Gaurav S. Sukhatme. The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment. *IEEE Transactions on Robotics*, Vol. 23, No. 4, pp. 661–675, 2007.
- [13] Belkin. Wemo. <http://www.belkin.com/us/wemo>.
- [14] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*, Vol. 53. Cambridge University Press Cambridge, 1998.
- [15] Chang-Sic Choi, Wan-Ki Park, Jin-Soo Han, and Il-Woo Lee. The architecture and implementation of proactive green home energy management system. In *Proceedings of 2010 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 457–458, 2010.
- [16] Reuven Cohen, Liran Katzir, and Danny Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, Vol. 100, No. 4, pp. 162–166, 2006.
- [17] Eric S. Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A framework for QoS-based routing in the internet. RFC 2386 (Informational), August 1998.
- [18] Geir Dahl and Njl Foldnes. LP based heuristics for the multiple knapsack problem with assignment restrictions. *Annals of Operations Research*, Vol. 146, No. 1, pp. 91–104, 2006.
- [19] Milind Dawande, Jayant Kalagnanam, Pinar Keskinocak, F. Sibel Salman, and R. Ravi. Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of Combinatorial Optimization*, Vol. 4, No. 2, pp. 171–186, 2000.
- [20] Xiaotie Deng, Tiko Kameda, and Christos Papadimitriou. How to learn an unknown environment. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 298–303. IEEE, 1991.
- [21] Xiaotie Deng and Christos H. Papadimitriou. Exploring an unknown graph. In

-
- Proceedings of the 31st Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 355–361. IEEE, 1990.
- [22] Stefan Dobrev, Rastislav Kráľovič, and Euripides Markou. Online graph exploration with advice. In *Proceedings of the 19th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pp. 267–278. Springer, 2012.
 - [23] EDION. EDISMA-H001. <http://www.edion.co.jp/news/release/release-20121003.html>.
 - [24] Lars Engebretsen and Jonas Holmerin. Towards optimal lower bounds for clique and chromatic number. *Theoretical Computer Science*, Vol. 299, No. 1, pp. 537–584, 2003.
 - [25] Tomaso Erseghe, Stefano Tomasin, and Alberto Vigato. Topology estimation for smart micro grids via powerline communications. *Signal Processing, IEEE Transactions on*, Vol. 61, No. 13, pp. 3368–3377, 2013.
 - [26] Amos Fiat and Gerhard J. Woeginger. Competitive analysis of algorithms. In *Online Algorithms*, pp. 1–12. Springer, 1998.
 - [27] Lisa Fleischer, Michel X. Goemans, Vahab S. Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pp. 611–620, 2006.
 - [28] Rudolf Fleischer and Gerhard Trippen. Experimental studies of graph traversal algorithms. In *Proceedings of Second International Workshop on Experimental and Efficient Algorithms (WEA)*, pp. 120–133. Springer, 2003.
 - [29] Rudolf Fleischer and Gerhard Trippen. Exploring an unknown graph efficiently. In *Proceedings of the 13th annual European conference on Algorithms (ESA)*, pp. 11–22. Springer, 2005.
 - [30] Naotaka Fujita, Tomoki Yoshihisa, and Masahiko Tsukamoto. Electrical power assignment method for hierarchical supply control system with energy on demand. *Proceedings of the IEICE General Conference*, pp. S160–161, 2013. (In Japanese).
 - [31] Naotaka Fujita, Tomoki Yoshihisa, and Masahiko Tsukamoto. A rule-based energy-on-demand system for reducing energy consumption. *IPSJ Journal*, Vol. 55, No. 2, pp. 1–9, 2014.
 - [32] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA,

- 1990.
- [33] Mike M. He, Evan M. Reutzel, Xiaofan Jiang, Randy H. Katz, Seth R. Sanders, David E. Culler, and Ken Lutz. An architecture for local energy generation, distribution, and sharing. In *Proceedings of 2008 IEEE Energy 2030 Conference*, pp. 1–6. IEEE, 2008.
 - [34] Yuya Higashikawa, Naoki Katoh, Stefan Langerman, and Shin-ichi Tanigawa. Online graph exploration algorithms for cycles and trees by multiple searchers. *Journal of Combinatorial Optimization*, pp. 1–16, 2012.
 - [35] Dorit S. Hochbaum. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Co., 1996.
 - [36] Frank Hoffmann, Christian Icking, Rolf Klein, and Klaus Kriegel. The polygon exploration problem. *SIAM Journal on Computing*, Vol. 31, No. 2, pp. 577–600, 2001.
 - [37] Masahito Ito, Ryuya Uda, Satoshi Ichimura, Kazuya Tago, Tohru Hoshi, and Yutaka Matsushita. A method of appliance detection based on features of power waveform. In *Proceedings of 2004 IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)*, pp. 291–294. IEEE, 2004.
 - [38] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. Design and implementation of a high-fidelity ac metering network. In *Proceedings of the 8th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 253–264. IEEE, 2009.
 - [39] Bala Kalyanasundaram and Kirk R. Pruhs. Constructing competitive tours from local information. *Theoretical Computer Science*, Vol. 130, No. 1, pp. 125–138, 1994.
 - [40] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pp. 302–311. ACM, 1984.
 - [41] Takekazu Kato, Hyun Sang Cho, Dongwook Lee, Tetsuo Toyomura, and Tatsuya Yamazaki. Appliance recognition from electric current signals for information-energy integrated network in home environments. *International Journal of Assistive Robotics and Systems*, Vol. 10, No. 4, pp. 51–60, 2009.
 - [42] Takekazu Kato and Takashi Matsuyama. *i*-Energy profile: Energy informationization profile by smart-tap network. *The transactions of the Institute of Electronics, Information and Communication Engineers. B*, Vol. 94, No. 10, pp. 1232–1245, 2011. (In Japanese).

-
- [43] Takekazu Kato, Kenji Yuasa, and Takashi Matsuyama. Energy on demand. *IPSSJ Journal*, Vol. 54, No. 3, pp. 1185–1198, 2013. (In Japanese).
 - [44] Tomoya Kawakami, Tomoki Yoshihisa, Naotaka Fujita, and Masahiko Tsukamoto. A rule-based home energy management system using the rete algorithm. In *Proceedings of the 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, pp. 162–163. IEEE, 2013.
 - [45] Hans Kellerer and Ulrich Pferschy. A new fully polynomial time approximation scheme for the knapsack problem. *Journal of Combinatorial Optimization*, Vol. 3, No. 1, pp. 59–71, 1999.
 - [46] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.
 - [47] Youichi Koyama, Akihiro Okamoto, and Hiroki Nakano. The design of the layers for power management technology in home power networks. *Proceedings of the IEICE General Conference 2012*, pp. S115–116, 2012. (In Japanese).
 - [48] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, Vol. 2, No. 1-2, pp. 83–97, 1955.
 - [49] Nallaperumal Kumaraguruparan, Harihara Sivaramakrishnan, and Sachin Sapatnekar Sapatnekar. Residential task scheduling under dynamic pricing using the multiple knapsack method. In *Proceedings of the 2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pp. 1–6. IEEE, 2012.
 - [50] Eugene L. Lawler, Jan Karel Lenstra, AHG Rinnooy Kan, and David B. Shmoys. *The traveling salesman problem: a guided tour of combinatorial optimization*, Vol. 3. Wiley Chichester, 1985.
 - [51] Junghoon Lee, Hye-Jin Kim, Gyung-Leen Park, and Mikyung Kang. Energy consumption scheduler for demand response systems in the smart grid. *Journal of Information Science and Engineering*, Vol. 28, No. 5, pp. 955–969, 2012.
 - [52] Joshua Lifton, Mark Feldmeier, Yasuhiro Ono, Cameron Lewis, and Joseph A. Paradiso. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *Proceedings of the Sixth International Symposium on Information Processing in Sensor Networks*, pp. 119–127. IEEE, 2007.
 - [53] Gang Lu, Debraj De, and Wen-Zhan Song. Smartgridlab: A laboratory-based smart grid testbed. In *Proceedings of the 2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 143–148. IEEE, 2010.
 - [54] Tomotaka Maeda, Akihiro Okamoto, Youichi Koyama, Hiroki Nakano, Hiroshi

- Hoshino, Shintaro Murakami, Naoyuki Morimoto, Kazumi Sakai, and Yasuo Okabe. Design and implementation of an on-demand home power management system with the protocol hierarchy. *IPSS SIG Notes*, 2014. (In Japanese).
- [55] M. J. Magazine and Osman Oguz. A fully polynomial approximation algorithm for the 0–1 knapsack problem. *European Journal of Operational Research*, Vol. 8, No. 3, pp. 270–273, 1981.
- [56] Takashi Matsuyama. Creating safe, secure, and environment-friendly lifestyles through i-Energy. *New Breeze*, Vol. 21, No. 2, pp. 1–8, 2009.
- [57] Takashi Matsuyama. i-Energy: background, objective, ideas, and realization methods. *IPSS Magazine*, Vol. 51, No. 8, pp. 926–933, 2010. (In Japanese).
- [58] Nicole Megow, Kurt Mehlhorn, and Pascal Schweitzer. Online graph exploration: New results on old and new algorithms. *Theoretical Computer Science*, 2012.
- [59] Yongguo Mei, Yung-Hsiang Lu, CS George Lee, and Y Charlie Hu. Energy-efficient mobile robot exploration. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pp. 505–511. IEEE, 2006.
- [60] Takuya Miyamoto, Youichi Koyama, Kazumi Sakai, and Yasuo Okabe. A GMPLS-based power resource reservation system toward energy-on-demand home networking. In *Proceedings of 2012 IEEE/IPSS 12th International Symposium on Applications and the Internet (SAINT)*, pp. 138–147. IEEE, 2012.
- [61] Laurent Mounier, Ludovic Samper, and Wassim Znaidi. Worst-case lifetime computation of a wireless sensor network by model-checking. In *Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pp. 1–8. ACM, 2007.
- [62] Bojan Mrazovac, Milan Z Bjelica, Nikola Teslic, and Istvan Papp. Towards ubiquitous smart outlets for safety and energetic efficiency of home electric appliances. In *Proceedings of IEEE the 1st International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pp. 322–326. IEEE, 2011.
- [63] Katta G. Murty. *Network Programming*. Prentice-Hall, Inc., 1992.
- [64] NEC System Technologies, Ltd. Green Tap. <http://www.nec.co.jp/press/ja/0911/0405.html>. (In Japanese).
- [65] NTT Advanced Technology. AT WATCH NET IR_mini. http://www.ntt-at.co.jp/page.jsp?id=1793&content_id=886. In Japanese.
- [66] Zeev Nutov, Israel Beniaminy, and Raphael Yuster. A $(1-1/e)$ -approximation algorithm for the generalized assignment problem. *Operations Research Letters*, Vol. 34, No. 3, pp. 283–288, 2006.

-
- [67] Seog-Chan Oh and Alfred J. Hildreth. Decisions on energy demand response option contracts in smart grids based on activity-based costing and stochastic programming. *Energies*, Vol. 6, No. 1, pp. 425–443, 2013.
 - [68] Yasuo Okabe. ICE-IT: Integration Technology of Information, Communication and Energy. *Systems, Control and Information*, Vol. 55, No. 6, pp. 221–226, 2011. (In Japanese).
 - [69] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, Vol. 84, No. 1, pp. 127–150, 1991.
 - [70] Sonali Patel, Ritu Tiwari, and Anupam Shukla. An energy efficient platform for mobile robot. In *Proceedings of the Sixth Global Conference on Power Control and Optimization*, pp. 90–95. AIP Conference Proceedings, 2012.
 - [71] Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, Vol. 6, No. 3, pp. 563–581, 1977.
 - [72] Kazumi Sakai and Yasuo Okabe. Quality-aware energy routing toward on-demand home energy networking. In *Proceeding of the 2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 1041–1044. IEEE, 2011.
 - [73] Tomoki Shibata, Kazumi Sakai, and Yasuo Okabe. The design and implementation of an on-demand DC grid in home. In *Proceedings of 2011 IEEE/IPSJ 11th International Symposium on Applications and the Internet (SAINT)*, pp. 152–159. IEEE, 2011.
 - [74] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, Vol. 62, No. 1-3, pp. 461–474, 1993.
 - [75] Omid Ameri Sianaki, Omar Hussain, and Azadeh Rajabian Tabesh. A knapsack problem approach for achieving efficient energy consumption in smart grid for endusers’ life style. In *Proceedings of 2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, pp. 159–164. IEEE, 2010.
 - [76] Daniel D. Sleator and Robert E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, Vol. 28, No. 2, pp. 202–208, 1985.
 - [77] Guangming Song, Fei Ding, Weijuan Zhang, and Aiguo Song. A wireless power outlet system for smart homes. *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 4, pp. 1688–1691, 2008.

- [78] Kin Cheong Sou, James Weimer, Henrik Sandberg, and Karl Henrik Johansson. Scheduling smart home appliances using mixed integer linear programming. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 5144–5149. IEEE, 2011.
- [79] Sparkfun Electronics. USB Weather Board. <https://www.sparkfun.com/products/9800>.
- [80] Brian P. Stalling, Tavis Clemmer, H. Alan Mantooth, Robert Motte, Hang Xu, Tyler Price, and Roger Dougal. Design and evaluation of a universal power router for residential applications. In *Proceedings of 2012 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 587–594. IEEE, 2012.
- [81] Ryo Takahashi, Yutaro Kitamori, and Takashi Hikihara. AC power local network with multiple power routers. *Energies*, Vol. 6, No. 12, pp. 6293–6303, 2013.
- [82] Tsuguhiro Takuno, Yutaro Kitamori, Ryo Takahashi, and Takashi Hikihara. AC power routing system in home based on demand and supply utilizing distributed power sources. *Energies*, Vol. 4, No. 5, pp. 717–726, 2011.
- [83] Keiji Tashiro, Ryo Takahashi, and Takashi Hikihara. Feasibility of power packet dispatching at in-home dc distribution network. In *Proceedings of the 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 401–405. IEEE, 2012.
- [84] The ECHONET Consortium. ECHONET Lite. <http://www.echonet.gr.jp/english/index.htm>.
- [85] The OpenADR Alliance. OpenADR. <http://www.openadr.org>.
- [86] The Raspberry Pi Foundation. Raspberry Pi. <http://www.raspberrypi.org>.
- [87] Junichi Toyoda and Hiroumi Saitoh. Proposal of an Open-Electric-Energy-Network (OEEN) to realize cooperative operations of IOU and IPP. In *Proceedings of the 1998 International Conference on Energy Management and Power Delivery*, pp. 218–222. IEEE, 1998.
- [88] Masahiko Tsukamoto and Takekazu Kato. Developing common specifications of smarttaps. *IPSJ Magazine*, Vol. 51, No. 8, pp. 934–942, 2010. (In Japanese).
- [89] Michal Tvarozek and Mária Bieliková. Bridging semantic and legacy web exploration: Orientation, revisitation and result exploration support. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, Vol. 3, pp. 326–329. IEEE, 2010.
- [90] Ubiquitous Corporation. iRemotap. <http://www.ubiquitous.co.jp/products/conceptlaboratory/iremotap/>. (In Japanese).

-
- [91] Tsuyoshi Ueno, Ryo Inada, Osamu Saeki, and Kiichiro Tsuji. Effectiveness of an energy-consumption information system for residential buildings. *Applied Energy*, Vol. 83, No. 8, pp. 868–883, 2006.
 - [92] U.S. Department of Energy. SmartGrid.gov. <http://www.smartgrid.gov>.
 - [93] Vijay V. Vazirani. *Approximation Algorithms*. Springer, 2001.
 - [94] Yusuke Yamada, Takekazu Kato, and Takashi Matsuyama. Human behavior estimation from power consumption patterns of appliances over smart tap network. *IEICE Technical Report*, Vol. 111, No. 134, pp. 25–30, 2011. (In Japanese).
 - [95] Masaya Yokohata, Tomotaka Maeda, and Yasuo Okabe. Power allocation algorithms of poe for on-demand power supply. In *2013 IEEE 37th Annual IEEE International Computers, Software, and Applications Conference Workshops (COMPSACW)*, pp. 517–522. IEEE, 2013.
 - [96] Tomoki Yoshihisa, Naotaka Fujita, and Masahiko Tsukamoto. HEMS toolkit: A toolkit for constructing a home energy management system. In *Proceedings of 2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 822–823. IEEE, 2011.
 - [97] Tomoki Yoshihisa, Naotaka Fujita, and Mashiko Tsukamoto. A rule generation method for electrical appliances management systems with home EoD. In *Proceedings of IEEE 1st Global Conference on Consumer Electronics (GCCE)*, pp. 248–250. IEEE, 2012.
 - [98] Lan Yu and Chi-Kin Chau. Complex-demand knapsack problems and incentives in ac power systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pp. 973–980. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
 - [99] ZigBee Alliance. Smart Energy Profile 2. <http://www.zigbee.org/Standards/ZigBeeSmartEnergy/SmartEnergyProfile2.aspx>.

Acknowledgements

I would like to express my sincere gratitude to Professor Yasuo Okabe for supervising me on research activities, for encouraging me to have broad and focused perspectives, and for providing me invaluable opportunities for growth as a researcher, during eight years since I joined the laboratory. My deep appreciation goes to Professor Takashi Matsuyama and Professor Tatsuya Akutsu, co-supervisors of this dissertation, who have given me invaluable comments and suggestions and have provided me opportunities to improve this dissertation. I owe my deep gratitude to Associate Professor Shuichi Miyazaki, who has supervised me on research activities on algorithms, has encouraged me to publish papers, and has kindly supported me when I was facing difficulties. Professor Kazuo Iwama has given me sharp comments and suggestions on my basic research results, without which I am not sure whether I was able to strengthen the results as described in Chapter 3.

I would like to express my deep thankfulness to Mr. Hirofumi Tayama, the president of Enegate Co., Ltd., who has given me invaluable opportunities for continuing my research activities, and has strongly motivated me to make contributions to both the company and society. During three years, my research activities at Department of Research & Development was strongly encouraged and supported by Mr. Mitsuaki Fukushima, Mr. Masami Tanaka, Mr. Terukazu Akehi, Mr. Masatoshi Yamashita, Mr. Kenichi Matsuda and Mr. Eijiroh Ohnishi, with their steady management of the department and groups. I was able to carry out my job in the company with sincere helps and suggestions by many people, especially Mr. Masaaki Yoshida, Mr. Hiroyuki Yoshimizu, Mr. Masafumi Takiyamada, Mr. Yuu Fujita, Mr. Yohei Kamimura, Mr. Tsuyoshi Kurano, Mr. Akinari Marui and Mr. Shingo Morita.

I would like to thank deeply both the current and former members of Okabe Laboratory. Professor Hiroki Takakura at Nagoya University, a former associate professor at Okabe Laboratory, has given me invaluable comments and suggestions, especially

at annual research meetings in the laboratory. Visiting Professor Dae Young Kim (a professor at Chungnam National University) has given me strong encouragements and a lot of precious time for interacting with foreign researchers. Ms. Kazumi Sakai has kindly supported my research activities throughout the eight years since I was a master course student. Dr. Shin Maruyama, Dr. Kenji Ohira (currently is an assistant professor at NAIST) and Dr. Mitsuo Okada have kindly provided me advice on research activities as a doctoral course student. Dr. Koji Kobayashi has supported me to start research work on online algorithms. Dr. Jungsuk Song (currently at Korea Institute of Science and Technology Information) and Mr. Shoichiro Fujiwara have given valuable suggestions and comments to my research work. Mr. Tomotaka Maeda and Mr. Toshihiro Takagi have given me fruitful time through discussions and recreation. Mr. Chia-Lung Hsieh has kindly helped and advised me to improve my English skill. Mr. Takeshi Fukuda has helped me to study algorithms though discussions and laboratory seminars. Mr. Tomoki Shibata has given me opportunities to experience the importance of prototyping ideas. Mr. Takuya Miyamoto and Mr. Masaya Yokohata have sincerely helped me to survey references. Mr. Teruhisa Tajima has given valuable suggestions and comments to my presentation.

I would like to express my appreciation for people associated with the research project “Integration Technology for Information, Communication and Energy (ICE-IT).” Especially, I have been able to deepen my consideration and understandings on the research work through the discussions with Dr. Hiroshi Hoshino, Mr. Youichi Koyama, Mr. Akihiro Okamoto, Mr. Shintaro Murakami and Dr. Tomoya Kawakami. Associate Professor Tomoki Yoshihisa at Osaka University and Mr. Naotaka Fujita have kindly provided useful information related to the content described in Chapter 5. This dissertation includes research work supported by NICT (National Institute of Information and Communication Technology) FY2009-FY2013 Advanced Telecommunication Research Fund.

I also would like acknowledge invaluable supports by people associated with Institute for Integrated Cell-Material Sciences (iCeMS) of Kyoto University, especially Professor Shinji Tomita and Assistant Professor Yoshitaka Morimura, to carry out my current job in iCeMS.

Finally, I would like to gratefully thank all the precious people in my life. Especially, my deep gratitude goes to my parents Chuzo and Hiroko Morimoto, who have always given me amazing opportunities and strong encouragements.

List of Publications by the Author

Journal Papers

- Shuichi Miyazaki, Naoyuki Morimoto, and Yasuo Okabe. The online graph exploration problem on restricted graphs. *IEICE Transactions on Information and Systems*, Vol. E92-D, No. 9, pp. 1620–1627, Sep. 2009.
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. The design and implementation of a smart outlet for policy-based power management. *IPSJ Transactions on Consumer Devices & Systems*, Vol. 3, No. 4, pp. 1–11, Dec. 2013. (In Japanese)

Refereed International Conference Papers

- Naoyuki Morimoto, Masami Tanaka, Terukazu Akehi, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, and Yohei Kamimura. The design and implementation of a smart tap for policy-based power management. In *Proceedings of 2012 IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 296–300, Jan. 2012.
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Policy-based power router with power sensors for efficient allocation of energy. In *Proceedings of 2012 IEEE 1st Global Conference on Consumer Electronics (GCCE)*, pp. 477–480, Oct. 2012.
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masa-

- fumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Smart outlet network for energy-aware services utilizing various sensor information. In *Proceedings of First International WorkShop on Energy-Aware Systems, Communications and Security (EASyCoSe) (In conjunction with IEEE AINA-2013)*, pp. 1630–1635, March 2013.
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Optimizing power allocation to electrical appliances with an algorithm for the knapsack problem. In *Proceedings of 2013 IEEE 17th International Symposium on Consumer Electronics (ISCE)*, pp. 155–156, June 2013.
 - Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. A power allocation management system using an algorithm for the knapsack problem. In *Proceedings of the 2nd IEEE International Workshop on Consumer Devices and Systems (CDS) (In conjunction with IEEE COMPSAC)*, 2014 (To appear).

Technical Reports

- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. An optimal online algorithm for the map construction problem on cycle graphs. *LA Symposium Summer 2007*, [9], July 2007. (In Japanese)
- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. An optimal online algorithm for the graph exploration problem on cycles. *IEICE Technical Report*, Vol. 107, No. 219, COMP2007-39, pp. 51–57, Sep. 2007. (In Japanese)
- Shuichi Miyazaki, Naoyuki Morimoto, and Yasuo Okabe. An optimal online algorithm for the graph exploration problem on cycles. *Proc. Annual Meeting of Asian Association for Algorithms and Computation (AAAC)*, p. 42, April 2008.
- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. The online graph exploration problem on restricted graphs. *IPSJ Kansai-Branch Convention*, pp. 51–54, Oct. 2008. (In Japanese)
- Shuichi Miyazaki, Naoyuki Morimoto, and Yasuo Okabe. An optimal online algorithm for the graph exploration problem on unweighted graphs. *Proc. Annual Meeting of Asian Association for Algorithms and Computation (AAAC)*,

p. 45, April 2009.

- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. Splittable resource allocation problems on bipartite graphs with assignment restrictions. *IEICE General Conference 2010*, BS-8-11, March 2010. (In Japanese)
- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. Resource allocation problems on power networks with distributed sources. *IPSJ Kansai-Branch Convention*, F-14, Sep. 2011. (In Japanese)
- Naoyuki Morimoto, Masami Tanaka, Terukazu Akehi, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, and Yohei Kamimura. A smart tap for intelligent and active power control. *IPSJ Kansai-Branch Convention*, F-101, Sep. 2011. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. A circuit-switching power router for optimal allocation of energy. *IEICE General Conference 2012*, BS-7-2, March 2012. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Smart outlet network for energy-aware services utilizing various sensor information. *The 75th National Convention of IPSJ*, 4E-3, March 2013. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Design consideration and implementation of a policy-based H2V/V2H management system. *IEICE General Conference 2013*, BS-6-3, March 2013. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. The design and implementation of a smart outlet for policy-based power management. *IPSJ SIG Consumer Device & Systems*, 2013-CDS-7 (6), May 2013. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. Controlling power allocation to electrical appliances using an algorithm for the knapsack problem. *IPSJ Kansai-Branch Convention*, F-102, Sep. 2013. (In Japanese)
- Naoyuki Morimoto, Yuu Fujita, Masaaki Yoshida, Hiroyuki Yoshimizu, Masafumi Takiyamada, Terukazu Akehi, and Masami Tanaka. A power allocation management system using an algorithm for the knapsack problem. *IEICE Technical Report*, Vol. 113, No. 371, COMP2013-44, pp. 39–43, Dec. 2013. (In

Japanese)

- Naoyuki Morimoto, Shuichi Miyazaki, and Yasuo Okabe. The multiple knapsack problem with assignment restrictions and capacity constraints. *LA Symposium Winter* [17], Jan. 2014.

Oral Presentations

- Naoyuki Morimoto. Resource allocation problems on bipartite graphs with assignment restrictions. *AsiaFI Summer School 2010*, Aug. 2010.
- Naoyuki Morimoto. Resource allocation problems on future power networks. *The 15th Core University Program Seminar on Next Generation Internet Technologies*, Sep. 2010.