Studies on Optimal Control Problems in Communication Networks with Multiple Users

Department of Computer Science

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2006

Atsushi Inoie

Studies on Optimal Control Problems in Communication Networks with Multiple Users

Atsushi Inoie

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of ENGINEERING

Department of Computer Science
Graduate School of Systems and Information Engineering
University of Tsukuba

March 2006

複数のユーザを持つ通信ネットワークにおける最適制御問題に関する研究

井家　敦

Internet に代表されるような大規模なネットワークシステムを考える。このようなネットワーク内には不特定多数のネットワーク構成要素（コンピュータ、ルータなど）および利用者が存在する。通常、これらのネットワーク全体を中央集権的に管理するのは非常に困難であると考えられる。このような場合、個々のユーザ（あるいはインターネットプロバイダ）が、他のユーザとの協力を求めずに自身の性能を追求していく、すなわち独立分散的な管理が行われる。

独立分散的に性能を最適化する方法を、非協力的最適化と呼ぶ。これは大きく分けて、次に示すような 2 つの場合が考えられる。一方は、ユーザを有限個の独立な組織に分割し、各組織がそれぞれ自身の性能を最適化するという場合である。この場合に、システムがある均衡状態に達した（すなわち、解が得られた）とき、それは Nash 均衡と呼ばれる。もう一方は、無限に多くのユーザがそれぞれ自身の性能を最適化する場合であり、その場合の均衡状態は Wardrop 均衡と呼ばれる。しかしながら、個々のユーザが自身の努力により性能向上を行う状況において、ネットワークにおける Braess のパラドックスのように、例えば、ネットワークシステムのリンクを追加したり、あるいはその容量を増加したりすることが、かえって全ユーザの性能を劣化させてしまう現象が現在までにいくつか報告されている。

一方で、仮に利用者間でネットワークの利用が協力的に行われるならば、ネットワーク全体での最適性は保証される。このように最適化される方法を協力的最適化とよぶ。しかしながら、最適解となりうる解の候補が複数存在することがあり、それらの中での選択が困難になってしまう場合がある。それを決める方法の 1 つとして、本論文では公平性という概念を用いる。

本論文では、ある通信ネットワークを複数の利用者が共有する場合において考えられる、以下の 3 つの最適制御問題について述べる。

1 つ目は、M/M/m 待ち行列のフロー制御問題における逆説的性能劣化である。フロー制御とは、システムへの過負荷を防ぎかつ性能を十分に発揮できるよう、システムへのフ

ローを調節することであり、ネットワークシステムの性能向上方策としてルーティングとともに重要視されている問題である。ここで、複数のユーザが１つのシステムを共有している場合を考える。このとき、ユーザに対する最適化方策として、システム全体での性能の平均を最適化する全体最適化と個々のユーザがそれぞれ自身の性能のみを最適化する非協力最適化が考えられる。後者の非協力最適化は経済学における非協力ゲーム（ゲーム理論）の考えに基づいたもので、それにより与えられる解は Nash 均衡と呼ばれている。

複数の独立した待ち行列システムとそれを１つに多重化した待ち行列システムを考えた場合、意思決定の範囲が広まることから後者がより高い性能を与えると考えられる。本研究では、M/M/m 待ち行列に対し、フロー制御特有の性能指標である、スループットをレスポンスタイムで割った値で得られるパワーを最大化する最適フロー制御問題を考える。まず、全体および非協力最適フロー制御問題の定式化を行い、その最適性条件およびアルゴリズムを導出する。また、数値的に非協力最適フロー制御において、各意思決定者が自身のパワーをそれぞれ最大化した場合に、待ち行列を多重化することが逆にすべてのユーザの性能劣化を導いてしまうことを示す。

２つ目は、Cohen-Kelly ネットワークの動的ルーティングにおける逆説的性能劣化である。Cohen-Kelly ネットワークとは、4 つのノードから構成される待ち行列ネットワークである。Cohen と Kelly はこのネットワークに対し、独立分散型（すなわち、個々のユーザが自身の性能のみを追及する）の静的ルーティング問題を考え、ノード間のコネクションを追加したときに、追加前より性能が悪化してしまう可能性があることを発見した。本研究では Cohen-Kelly ネットワークの動的ルーティング問題を考え、同様の性能劣化が生じうることを示した。

ただし、Cohen-Kelly ネットワークの動的ルーティングの性能劣化に関しては、Calvert 等によってすでに研究が行われている。しかしながら、彼らの研究では、シミュレーション結果と解析的結果との間に多少の相違点が見られ、また性能劣化の評価方法も適切でないように見受けられた。ゆえに、本研究では彼らの研究について不十分であると考えられる点について指摘し、解析的結果に基づいたシミュレーションを行う。結果として、Calvert 等と同様に我々の方法でも同様の性能劣化が発生することが明らかになった。

３つ目は、公平性を考慮した分散コンピュータシステムの最適負荷分散である。負荷

分散とは、与えられたシステム構成のもとで性能を最大限に発揮できるように、各コンピュータに負荷を割り当てることであり、現在、情報・通信ネットワークにおいて重要視されている問題の 1 つである。一般に、複数のユーザがシステムを利用している場合、負荷分散政策が有効であるかを判断するには、それが Pareto 最適性および公平性を満たしているかが重要な要素として挙げられる。負荷分散政策が Pareto 最適性を満たすとは、複数のユーザがいる場合に、任意の 1 人のユーザの性能を向上させるためには、少なくとも他の 1 人のユーザの性能を犠牲にしなければならないような性質を持つことを指す。一方、公平性はすべてのユーザに対し不利益が生じないことを指す。Pareto 最適性と公平性の双方を満たす代表的なものとして、例えば、Max-min Fairness、Proportional Fairness といったものが挙げられる。さらに、これらを一般化した公平性指標が Mo と Walrand によって提案されている。

一方で、Nash 均衡もまた、各ユーザが自身の性能を追及するという意味で公平性を持つと考えられる。しかしながら、時に Nash 均衡は Pareto 最適性を満たさないことがある。従って、本研究では、Nash 均衡を Pareto 最適解に拡張した、Nash-Proportionate-Fairness を考え、単純な分散コンピュータシステムに対し適用し、その特徴を観測する。

Abstract

Consider a network system such as the Internet or Grid. The network system consists of enormous number of network components (e.g., computers, routers and communication media) and users who require services from the network system. Each individual user may have different requests to the network. Consequently, there exist various kinds of packet flows (e.g., voice, video and data) required by independent users in the network. Owing to the diversified needs of the users, the service providers (or the network administrators) often face difficulty in guaranteeing Quality of Service (QoS) for all users.

Considering optimal control in communication networks with multiple users, we can think of the following two optimization schemes: the noncooperative and cooperative optimization schemes. In the noncooperative optimization scheme, each user strives to optimize its own performance measure (the utilities of the user) unilaterally. On the other hand, in the cooperative optimization scheme, a single agent optimizes a single performance measure (e.g., the sum of utilities of all users).

In this thesis, we study the following three problems on optimal control in communication networks where multiple users share the system resources.

First, optimal flow control problems of multiple-server (M/M/m) queueing systems are studied. Due to enhanced flexibility of the decision making, intuitively, we expect that grouping together separated systems into one system provides improved performance over the previously separated systems. In this thesis, we present a counter-intuitive result against such an expectation. More precisely, we consider a noncooperative optimal flow control scheme of M/M/m queueing systems where each of multiple players strives to

optimize unilaterally its own power where the power of a player is the quotient of the throughput divided by the mean response time for the player. We report a counter-intuitive case where the power of every user degrades after grouping together $K(>1)$ separated M/M/N systems into a single M/M/$(K\times N)$ system.

Second, a paradox in dynamic routing in the Cohen-Kelly network is studied. Intuitively, we expect that adding capacity to a network improves the performance of the users. Braess, however, showed an example where the opposite occured. We refer to such a situation as a paradox. In this thesis, we deal with a class of queueing networks where Cohen and Kelly discovered a paradox in static routing by individual users. We consider the dynamic routing problems in the above mentioned class of networks, and show the existence of a paradox in dynamic routing of the above mentioned class of networks through simulation experiments analogous to what Cohen and Kelly showed.

Third, fair and Pareto optimal solutions to a load balancing problem are studied. Various fairness objectives are studied in relation to Pareto optimal sets and Nash equilibria. We examine an already discussed general parameterized fairness objective that covers a variety of fairness criteria and the newly introduced Nash-proportionate-fairness objective. We mainly study them numerically on a simple static load balancing model with two identical servers (computers) each of which has an independent arrival process and its own queue. Through numerical results, several counter-intuitive results are shown. For example, we observe that the points that achieve the general parameterized fairness objectives may cover a part but not all of the Pareto set, and at times, do not cover the Nash-proportionate-fair Pareto optimal point.

Acknowledgements

First of all, I would like to express my gratitude to Professor Hisao Kameda, my supervisor, for his many suggestions and comments to lead me the completion of this thesis. His insightful instruction inspired me to make further efforts. Without his support, this work could not have been completed.

I am also deeply grateful to Associate Professor Jie Li for his continueous encouragement. Although I had few opportunities to receive directly his instruction, he was very helpful in many ways.

I would like to thank Corinne Touati who has given me many advices. I deeply appreciate her helpful comments on my research.

My deep gratitude also goes to Professor Hiroyuki Kitagawa, Professor Koichi Wada, Associate Professor Shigetomo Kimura of the Institute of Information Sciences at University of Tsukuba for their constructive comments and suggestions.

I would like to thank all members of our laboratory, especially, Atsushi Kubota, Vasilache Adrian and Chang Woo Pyo who have greatly contributed to my research through their system administration work and comments on English writing.

Finally, I am also indebted to Professor Katsuhisa Ohno, who was my previous professor when I has been a student at Nagoya Institute of Technology. He had supported me so that I could enter University of Tsukuba.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

For a number of years, the computer and communication network environments around us have been changing dramatically. Personal computers and communication equipments have been improving their cost-performance, and they have been downsized. Although such rapid growth and development of the networks are great satisfactions to us, they yield a need for next-generation network control technologies.

Consider a wide-scale network system such as the Internet or Grid. The network system consists of enormous numbers of network components (e.g., computers, routers and communication media) and of users each of who requires various kinds of services for the network system. Each individual user may have different requests to the network. Consequently, there exist various kinds of packet flows (e.g., voice, video and data) required by the independent users in the network. The users also require a satisfactory level of the network performance and the cost (e.g., the communication delay and the network pricing). Owing to the diversified needs of users, the service providers (network administrators or agents) often face with difficulty in guaranteeing Quality of Service (QoS) for all the users.

In this situation, we can consider the following two network control schemes: In the first one, there exist an infinitely large number of users in the network where each user belongs to a group of providers. Each provider optimizes unilaterally its own performance objective without cooperation with the other providers. It is called the *noncooperative optimization scheme*. If the number of providers is finite, then the corresponding equilibrium is referred to as a Nash equilibrium [165]. If infinitely large number of users optimize unilaterlly their own performance objective, then the corresponding equilibrium is referred to as a Wardrop equilibrium [202].

The other is that a single provider on behalf of the entire users optimizes a performance objective in order to satisfy all the users. It is called the *cooperative optimization scheme*. In this scheme, satisfying (Pareto) efficiency of network performance may be comparatively easier than in the noncooperative optimization scheme. The solutions which we can regard as efficient, however, are not unique in the cooperative optimization scheme, and we cannot find any absolute preference among these solutions. Consequently, it is difficult to provide satisfactory performance for all users. In this thesis, we consider a fairness concept [41, 59, 153] as a kind of criteria deciding the preference.

The rest of this chapter is organized as follows: In Sections 1.2 and 1.3, we briefly describe the noncooperative and cooperative optimization schemes, respectively. In Section 1.4, we introduce our problems and in Section 1.5, we show the organization of the remaining chapters in this thesis.

1.2 Noncooperative Optimization

As described in the previous section, in the noncooperative optimization scheme, multiple providers strive to optimize their own performance unilaterlly. This scheme has been developed in the noncooperative game theory [18, 57], and has been applied to various kinds of problems in many fields, such as management sciences, communication networks, and transportation systems.

Nash Equilibrium: Considering a network shared by a finite number of groups each of which has a single provider (i.e., decision maker). Each provider optimizes unilaterally its own performance objective, that is, there is no coordination among providers. In this situation, if no provider can improve its own performance by changing its strategy, then the achieved equilibrium is called a *Nash equilibrium* [165].

A feature of a Nash equilibrium is that there exist several helpful mathematical properties found by many researchers. Especially, if the strategy space is convex and compact, and the utility functions are semi-continuous, then a Nash equilibrium exists (see, e.g., [57]). Rosen also [177] showed some conditions for existence, uniqueness, and stability of Nash equilibria in convex problems.

Nash equilibria are, however, not always Pareto optimal. A famous example is the prisoner's dilemma, that is, making decision without coalition among providers sometimes leads to the worst result.

Wardrop Equilibrium: Consider a case where a network is shared by an infinitely large number of users and where each individual user optimizes unilaterally its own performance objective. In this situation, it is assumed that the behavior of each individual user has little impact on the entire system since the effect of the decision of each user is negligibly small. Wardrop [202] first studied this situation on a routing problem in a transportation network, and therefore the achieved equilibrium is called a *Wardrop equilibrium*.

The definition of the Wardrop equilibrium is different from that of the Nash equilibrium. The Wardrop and Nash equilibria, however, have the following relation: A behavior in the Nash equilibrium approaches to a behavior in the Wardrop Equilibrium when the number of users is infinitely large. Haurie and Marcotte [72] observed this relation in state-independent noncooperative optimization problems on two transportation networks that had finite number and infinitely large number of users. Wie and Tobin [204] also showed this relation in a dynamic traffic assignment problem.

1.3 Cooperative Optimization

In this section, we describe Pareto optimality and fairness. Both Pareto optimality and fairness are important concepts of efficiency for the network and for users, respectively.

1.3.1 Pareto Optimality

In a network shared by multiple users, a state of the network determines performance (i.e., the utilities or the costs) of the users. Then, we need to determine whether the performance of the users in a state is absolutely superior to that given by any other state or not. We may explain the absolute superiority by using two concepts called *Pareto-inefficiency* and *Pareto-optimality*, which have been introduced by Vilfredo Pareto [170].

Consider a network shared by n users. Let U_i denote the utility of user i where U_i should be maximized. Then, the definition of Pareto-inefficient is as follows: A utility vector $\boldsymbol{U} = (U_1, \ldots, U_n)$ is called *Pareto-inefficient* if there exists a utility vector $\boldsymbol{U}'$ such that $U_i \leq U'_i$ for all $i = 1, 2, \ldots, n$, and $U_j < U'_j$ for some $j,\ j = 1, 2, \ldots, n$. If a utility vector is Pareto-inefficient, then it is possible to improve the performance for some users without the degrading performance for any other user.

Also, the definition of Pareto-optimal is as follows: A utility vector $\boldsymbol{U}$ is called *Pareto-optimal* if there exists no utility vector $\boldsymbol{U}'$ that satisfies $U_i \leq U'_i$ for all $i = 1, 2, \ldots, n$, and $U_j < U'_j$ for some $j,\ j = 1, 2, \ldots, n$. In other words, a utility vector is Pareto-optimal if it is not Pareto-inefficient.

We next define *strong Pareto superiority* and *strong Pareto-inefficiency*. Consider two utility vectors $\boldsymbol{U}$ and $\boldsymbol{U}'$. If a utility vector $\boldsymbol{U}'$ satisfies $U_i < U'_i$ for all $i = 1, 2, \ldots, n$, a utility vector $\boldsymbol{U}$ is *strongly Pareto-inefficient*, and a utility vector $\boldsymbol{U}'$ is *strongly Pareto-superior* to $\boldsymbol{U}$.

In general, there can exist an innumerably large number of Pareto optima in a network control problem. For example, let us consider a two-user network control problem where the objective is to maximize the utilities of both users simultaneously. Then, the Pareto

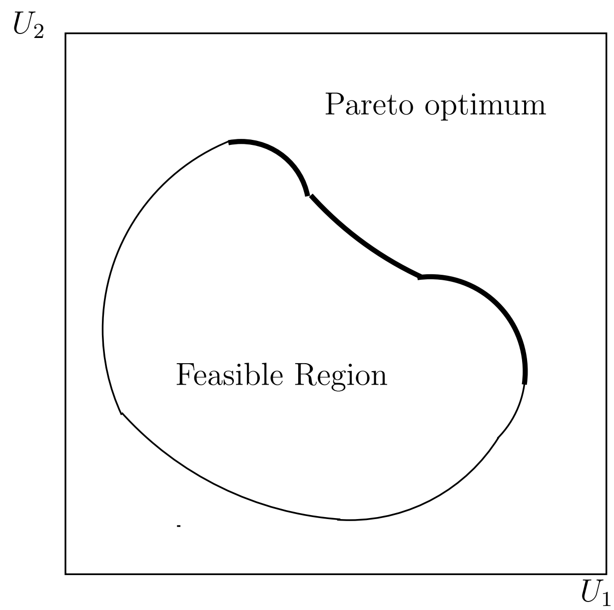

Figure 1.1: Pareto border and feasible region in a two-user network control problem where U_i is the utility of user i.

optima cover the upper right part of the feasible region in this problem (see Figure 1.1).

1.3.2 Fairness

From the viewpoint of users, fair use of the network is important. Defining fairness is, however, not simple. In fact, fairness is inherently subjective, and depends heavily on the network and on the traffics of users that should be optimized. In [59], the fairness scheme in computer and communication networks is classified into the following three classes: fair resource utilization, equal performance, and balanced interference among user. Jain [81] also suggested a fairness index.

Here, we must note that usually achieving simultaneously both fairness and effective use of resources in the network is difficult since maximizing the performance in the whole network is often incompatible with achieving fairness for all users[1]. We therefore need to

[1]Tang et al. [193], however, found a counter-example, that is, a fairer strategy is *always* more efficient.

find a tradeoff between the fairness and the network performance.

There exist various types of fairness concepts that have been already proposed. In this section, therefore, we introduce some fairness objectives. Note that these fairness objectives are Pareto optimal.

Max-Min Fairness: *Max-min fairness* [21, 68, 79, 82, 152, 154, 158, 176] is one of the most common concepts of fairness. The basic idea of the Max-min fairness is to optimize the performance for the user whose performance is the worst (see e.g., [21]). In other words, the Max-min fairness makes performance for all users as equal as possible.

Note that the fairest utility vectors in the concept of fairness by Bonald and Massoulie [22] corresponds to a *max-min fair* point. Note also that a max-min fair point always satisfies Pareto optimality. It is known that there exists a unique max-min fair point if the numbers of users and of network resources shared by users are finite [154].

Proportional Fairness: Although a max-min fair point satisfies Pareto optimality, it may sometimes utilize resources in a network inefficiently for all users. In fact, this question was discussed in the literature (e.g., [22, 97, 154, 176, 189]). For example, in [97] the max-min fairness does not make efficient use of the resources in a network since it provides users whose performances are comparatively worse with better performances.

Kelly et al. [97, 98] then proposed an alternative concept, called *proportional fairness*. Let $\boldsymbol{U}$ denote the utility vector in a feasible state. If $\boldsymbol{U}$ is the proportionally fair point, then for the utility vector $\boldsymbol{U}^*$ in any other feasible state, we have

$$\sum_i \frac{U_i^* - U_i}{U_i} \leq 0, \tag{1.1}$$

where U_i and U_i^* are i-th elements of $\boldsymbol{U}$ and $\boldsymbol{U}^*$, respectively.

In [97], it has been shown that the logarithm of utility function is intimately associated with the concept of proportional fairness. In other words, the maximization of the sum of logarithms of utilities for the users leads to the proportionally fair point. It is well-known that the proportionally fair point corresponds to a kind of Nash bargaining solutions [164] (see Remark 2.3 in [206]).

Generalized Fairness Objective: Recently, in the context of congestion control, Mo and Walrand [162] proposed a general and simple uniform description (parameterized by α) of a wide family of fairness objectives, including in particular, the proportional fairness and the max-min fairness. In their article, it is called *α-proportional fairness*. Touati et al. [196, 197, 198] further generalized the fairness objective by taking into account the utilities for the users. Some applications to this fairness objective appear in, for example, La and Anantharam [128, 130] in window-based congestion control, and Fang and Bensaou [52] in wireless ad-hoc networks.

1.4 Our Problems

In this thesis, we focuses on the various types of optimal control problems of computer and communication networks where multiple users share the network noncooperatively or cooperatively. Especially, we deal with the following three problems:

1) A paradox in optimal flow control of M/M/m queues [75, 78].

2) Braess paradox in optimal routing for the Cohen-Kelly network [77].

3) Case study of Pareto set, fairness, and Nash equilibrium in load balancing [76, 199, 200].

Note that 1) and 2) deal with noncooperative optimization, and 3) deals with cooperative optimization.

First, optimal flow control problems of multiple-server (M/M/m) queueing systems are studied. Flow control is a means to utilize limited system resources effectively and to guarantee proper quality of service (QoS) in computer and communication networks. Consider a network that is shared by multiple users. Due to enhanced flexibility of the decision making, intuitively, we expect that grouping together separated systems into one system provides improved performance over the previously separated systems. In fact, it has been shown analytically that grouping together separated systems improves the average response time although the utilization factor of each server remains the same (see [106]). This thesis, however, presents a result counter-intuitive against such an expectation. We consider a noncooperative optimal flow control scheme of M/M/m queueing systems where each of multiple players strives to optimize unilaterally its own power where the power of a player is the quotient of the throughput divided by the mean response time for the player. We show numerically a counter-intuitive case where the power of every user degrades after grouping together $K(>1)$ separated M/M/N systems into a single M/M/$(K\times N)$ system in noncooperative optimal flow control scheme. Especially, we show that grouping together the systems decreases the power of every user about 10% in the worst case. On the other hand, we see the opposite behavior (that is, it agrees to our natural intuition) in the overall flow control scheme. Also, in this thesis, we prove the existence and uniqueness of solutions to the overall and noncooperative optimal flow control problems, and present an algorithm that computes the solutions.

Second, Braess paradox in dynamic routing for the Cohen-Kelly network is studied. Cohen and Kelly [36] studied static routing in two queueing networks, called the *initial* and *augmented* networks. The initial network has two paths, and the augmented network has an additional path. In both networks, an individual user arriving at the network only knows the expected transit time of each path in the network, and the user is routed by using that information. Intuitively, we expect that adding capacity to a network improves the performance of the users. In that situation, Cohen and Kelly [36], however, showed that the opposite occurs, that is, they found a case where the augmented network gives worse performance than the initial network. We refer to such a situation as a (Braess) paradox. In this thesis, we consider the dynamic routing problems in the above mentioned class of networks. Here, note that Calvert et al. [36] have already shown the existence of a paradox in dynamic routing for the networks. We may, however, think that some of their results are improper. For example, their simulation study is not based exactly on their analytical study, and they just compared between overall mean transit times of jobs in the initial and augmented networks. We therefore pay attention to some confusions of their study, and show that a paradox also occurs in dynamic routing of the above mentioned class of networks analogous to what Cohen and Kelly showed.

Third, fair and Pareto optimal solutions to a load balancing problem are studied. Load balancing among computers in a distributed system is a means of efficiently sharing resources among users. An important objectives of a load balancing system is the Pareto optimality. That is, there exist no other states where all users have better benefits simultaneously in a Pareto optimal situation. In general, there exist a number of Pareto optima in a load balancing problem. Therefore, we may face to choosing an appropriate preference among them. Fairness objective is a criterion for this requirement. Various fairness objectives have been already proposed, for example, max-min fairness [21], proportional fairness [97]. Furthermore these objectives have been generalized by Mo and Walrand [162]. In this thesis, we also consider a Nash equilibrium based Pareto optimum, called *Nash-proportionate-fairness*. We examine an already discussed general parameterized fairness objective that covers a variety of fairness criteria. We study them mainly numerically on a simple static load balancing model with two identical servers (computers) each of which has an independent arrival process and its own queue. Through numerical results, several intuitive results are shown.

1.5 Organization of the Thesis

This thesis is organized as follows: In Chapter 2, we describe the previous and current work on network routing, flow control, load balancing and Braess paradox. In Chapter 3, we present a paradox in optimal flow control of M/M/m queues. In Chapter 4, we consider a dynamic routing problem in the Cohen-Kelly network, and show a paradox. In Chapter 5, we consider load balancing in a distributed computer system, and characterize the Pareto set, the Pareto solution based on Nash equilibrium and the fairness objective. In Chapter 6, we conclude this thesis. In Appendix A, we show the existence and the uniqueness of solutions to the optimization problems, and present algorithms that obtain the solutions.

Chapter 2

Related Work

In this chapter, we present a survey of control problems in some communication networks and distributed computer systems, that is, network routing, flow control, and load balancing, which are intimately relevant to our work. We further describe a counter-intuitive phenomenon observed in network control for some classes of networks, called the Braess paradox, and introduce some network examples where the Braess paradoxes occur. Some related surveys are given, for example, in [7, 12, 60, 91, 95, 96, 109, 111].

2.1 Network Routing

Network routing is deciding paths and the amount of traffic flows for users in a network in order to maximize the performance for the users or the network. In fact, routing is regarded as a special case of *resource sharing* [140]. Methods and technologies of routing are developed in transportation sciences as well as in computer and communication engineering. In this section, we describe a survey of network routing.

When we consider a network routing problem, the goal is to optimize one or several objectives. The objectives that we can think of are, for example, average delays, communication costs, and network prices. If the capacities of network resources are finite, then we also must consider the loss probabilities of network traffic.

Lee and Cohen [134] dealt with a set of parallel M/M/m queues where users could adjust the amount of flow in each queue. They considered both the mean queue length and the delay of each user as performance criteria, and showed that there exists at most one Nash equilibrium. Economides and Silvester [46] found the existence and uniqueness of the Nash equilibrium in routing among two parallel exponential servers with two users where the average delay and blocking probability are used as performance criteria.

Orda et al. [166] dealt with routing in a communication network shared by multiple users each of who had a flow of demand. Each user distributed its own flow to each link in order to minimize a sum of the cost functions of links. They first considered a network with parallel links, and they showed the uniqueness of a Nash equilibrium under some assumptions on the cost functions, and showed the convergence to a Nash equilibrium in the simple model that consisted of two users and two links. They further studied a network of general topology that consisted of a finite set of nodes and a set of direct links between two nodes. They showed that Nash equilibrium in this network was unique if the network model satisfied diagonal strict convexity [177].

With both network models studied by [166], La and Anantharam [125, 129] studied repeated games, and showed some additional properties. They further considered more general networks where the networks have a single source-destination pair [125]. That is, every user has the same source and destination nodes. It is shown that there exists a Nash equilibrium that corresponds to the minimization of the overall cost of the network.

Libman and Orda [140, 141] also studied noncooperative routing in parallel links as studied in [166]. They claimed, however, that splitting the flow of each user was often impractical, and considered noncooperative networks with atomic resource sharing in which each demand cannot be split among two or many resources. They showed the convergence of a Nash equilibrium in the case where the costs of the network were given by the remainder of the link capacities [141].

Altman et al. [3] considered network routing in which a number of users share two parallel links with linear holding costs. They obtained the properties of the convergence to the unique Nash equilibria under two dynamic routing schemes: round-robin and random polling schemes. Liu and Simaan [147, 148] also studied two node parallel links shared by competitive multiple users. They applied a game theoretic approach, called *Non-inferior Nash strategy* [146] to their model.

Korilis et al. [114] studied a Stackelberg equilibrium in routing for the parallel link model of [166]. More specifically, an agent (leader) wishes to optimize the overall network performance. The agent, however, also understands that the users (followers) behave noncooperatively to him. More specifically, the agent needs to predict the behaviors of the users. This situation is called leader-follower problem [7, 190], and its solution is a Stackelberg equilibrium. They [114] derived a necessary and sufficient condition of the existence of a maximally efficient strategy in the Stackelberg routing policy. Roughgarden [181] also studied Stackelberg equilibrium routing in a class of networks. They [115] also studied optimal capacity allocation under noncooperative routing. They considered a situation where a network designer endeavored to allocate link capacities in the network in order to make a Nash equilibrium efficient (i.e., Pareto optimal), and showed that the performance degradation did not occur in noncooperative routing of their network when adding the capacity to the network. It is a similar behavior to that of cooperative routing.

Korilis et al. [118] studied pricing in a noncooperative network using routing in the model of [166]. Indeed, in the last few decades, pricing have attracted much attention in computer and communication networks (see e.g., [29, 30]). In their study, the cost function for a user is the sum of the average delay of the user and the monetary cost of the user required by using network links. Considering pricing in parallel links, they introduced link weights referred to as discount factors. The agent seeks a set of discount factors such that a Nash equilibrium coincides with a Pareto optimal solution, which is called *incentive compatible*. Assuming that the monetary cost of any network link is proportional to the average delay of the link, they showed that the agent can always determine the discount factors. Korilis and Orda [117] considered pricing in routing for a class of general topology networks with multiple users where their cost function is based on the delay of a generalized processor sharing system [58, 168, 169]. They showed the existence of incentive compatible price functions in this network. Park et al. [171] studied a QoS provision queueing system model where m service classes and n users. In fact, their formulation is very close to that in [166] although the utility of each user takes the value of either 0 or 1 (that is, if the system satisfies the requirement of user i, then the utility is 1, otherwise it is 0).

Altman et al. [4] studied a class of general topology networks with multiple users. In their study, they showed that when the costs of the links are polynomial, there exists a unique Nash equilibrium. Furthermore, they showed some explicit results when the costs of the links were affine. Boulogne et al. [24] studied mixed equilibrium [71], that is, mixing of Nash and Wardrop equilibria in multiclass routing games. They showed the existence and uniqueness of an equilibrium under several assumptions. Azouzi and Altman [17] studied a coupled Nash equilibrium [177] in optimal routing for the same class of networks as [4]. They showed that there exists an equilibrium under some assumptions on the cost functions. Boulogne and Altman [23] studied competitive routing in multicast communications where multicast communication is a process to send the same message from a source node to an arbitrary number of destination nodes. They dealt with two specific networks: three-node and four-node networks, and discussed the uniqueness of Nash equilibria and their convergence in these networks.

Altman and Kameda [9] dealt with static routing in open BCMP queueing systems [19] with multiple users. They showed some properties of the existence and uniqueness of overall, Wardrop and Nash equilibria. Kameda et al. [90] studied overall and individual routing for networks of general topology with multiple users and general costs of links. They showed that there exist a link-traffic loop-free property[1] in these problems.

Some significant results in routing of loss networks are described in [95]. In this study, the goal is to minimize the average blocking probability of each link in a network. Altman et al. [2] studied noncooperative routing in loss networks with multiple users. They

[1]Loop-free property is that there is no link-traffic loop within each user.

considered a network consisting of J parallel links, and adopted two concepts of solutions: Nash and Wardrop equilibria. They showed the existence of a Wardrop equilibrium using a result by Patriksson [172] although it is not unique. They also showed that there exists a unique Nash equilibrium if the network has some properties. Mitra and Gibbens [160], and Chung et al. [184] considered state-dependent routing in symmetric loss networks.

Altman [1] considered dynamic routing for queueing systems in a non-zero game theoretic framework. Altman et al. [6] studies a Markov decision process (MDP) [173] model of dynamic routing in M parallel servers.

Roughgarden and Tardos [178, 180, 182] studied a general topological network with general link costs. They used the price of anarchy (or coordination ratio) [119, 167], which is the ratio between the worst-case Nash equilibrium and the overall optimal solution. They concentrated on selfish behaviors of users, and proved that if the cost function of each link is linear, then the total cost by selfish routing (Nash equilibrium) is at most 4/3 times as the total cost by overall routing. They also proved that if the cost function of each link is continuous and nondecreasing in the flow, then the total cost by selfish routing is at most twice as large as the total cost by overall routing. Further studies in selfish routing can be found, for example, in [37, 53, 54, 157, 183].

A recent trend in routing applications is noncooperative routing in overlay networks [13, 186]. In an overlay network, end hosts choose their routes individually. Hence, the behaviors of users can be expressed by using concepts of noncooperative games. In fact, some publications related to overlay networks appear [33, 83, 149, 174].

Chun et al. [33] applied an approach of the noncooperative game to their network where their work was inspired by [51]. Liu et al. [149] studied the interaction between overlay routing and Traffic Engineering (TE). They formulated it as a two-player noncooperative non-zero sum game where the goal of the overlay network is to minimize the average delay, and the goal of the TE is to minimize the network cost. They showed that a noncooperative behavior of the hosts in the overlay network causes increase of the cost. Furthermore, Jiang et al. [83] considered the interaction of multiple overlay routing.

They proposed a new concept, called overlay optimal routing and formulated their model as a noncooperative optimization problem. They showed that there always exists a Nash equilibrium, and the equilibrium is not Pareto optimal.

2.2 Flow Control

Flow control is one of the most important technologies in communication networks to use the limited resources efficiently. In fact, flow control adjusts the input traffic flow (i.e., the instantaneous throughput of the network) in order to avoid the waste of resources in the network and to prevent the performance degradation by traffic congestion. The term *congestion*[2] is used commonly in the sense of a phenomenon, that is, a heavy traffic in the network leads to the higher delay and the lower throughput for all users. This phenomenon is similar to a traffic jam in a highway network.

Here, we must note that flow control is not a direct approach decreasing the delay for users of a network. More specifically, decreasing the delay of the network causes decreasing the throughput of the network, and vice versa. A primal advantage achieved by flow control is prevention of a disastrous congestion in the network that would cause inconvenience to many users. If a user wishes to decrease the network delay without decreasing the throughput, he will just increase the communication resources of the network, or improve the routing algorithm.

Thus, an important problem involved in flow control is how to determine a trade-off between the average delay and throughput of the network. In studies of computer and communication networks, several performance measures satisfying the above requirement have been considered. Lazar [132] has used the maximization of the average throughput subject to a bound on the average delay as a performance requirement for flow control. Giessler et al. [62] have introduced the notion of *power*, which is defined as the quotient of the average throughput over the average delay. Kleinrock [108], Kumar and Jaffe [120],

[2] Indeed, in some articles, *congestion control* is used interchangeably with flow control.

and Selga [187] later generalized the power as follows: Denote the average delay and the average throughput of the network by T and λ, respectively. Then the power P of the network is given as

$$P = \frac{\lambda^{\beta}}{T}, \tag{2.1}$$

where $\beta > 0$ is the trade-off parameter (i.e., The definition of Giessler et al. [62] is the case where $\beta = 1$). Fortunately, in the concavity of power is shown in some queueing system models (for example, see [75, 78, 109]). Kolias and Kleinrock [110] applied the power to ATM switching systems.

Jaffe [80] first studied flow control of an M/M/1 queue that has multiple users. He gave a class of decentralized (i.e., each user strives to optimize its own performance measure) flow control algorithms, and found that a solution obtained by the algorithms were inefficient even in an optimization problem with any considerable performance objective. Kumar and Jaffe [120] also dealt with the same problem as [80]. In their study, three types of powers are considered and compared. After their study, Douligeris and Mazumdar [43] studied a Nash equilibrium in noncooperative flow control of an M/M/1 queue with multiple users. They further studied a Stackelberg equilibrium in a two user case. They showed that increasing the number of users caused the degradation of the overall throughput of the queue. Shenker [188] also considered Nash equilibria in another class of flow control problems of M/M/1 queues, and discussed their Pareto inefficiency. Pareto optimal (efficient) flow control in a telecommunication network with multiple users was studied by Douligeris and Mazumdar [42], and Kumar et al. [121]. Douligeris and Mazumdar [42] showed a necessary and sufficient condition for a solution in flow control for the same model as [43] to be Pareto optimal. They further suggested several algorithms that achieved a Pareto optimal solution, and they compared the behaviors of the convergences. Kumar et al. [121] proposed another iterative algorithm.

Zhang and Douligeris [209] proposed the synchronous and asynchronous greedy algorithms in order to obtain a Nash equilibrium in flow control for the same model as [43]. The synchronous greedy algorithms were based on the Gauss-Seidel method, and the convergence of a decentralized synchronous greedy flow control algorithm was shown. Also, in the asynchronous greedy algorithm based on the Jacobi method, they showed a necessary and sufficient condition for the convergence of the algorithm. As a note on study of Zhang and Douligeris [209], Ching [31] studied the convergence of the asynchronous algorithm with relaxation, and suggested some relaxation parameters such that the algorithm converges fast.

Another algorithmic studies in optimal flow control were studied by Gibbens and Kelly [61] Kar et al. [94], Kunniyur and Srikant [122, 123, 124], La and Anantharam [127, 128, 129], Liu et al. [145], Low and Lapsley [150], and Wang et al [201]. Their model is close to the model by Kelly et al. [97, 98] where the objective is to maximize an aggregation of utilities for users. They presented algorithms to achieve overall optimal solutions, and proved their convergence and stability. Furthermore, Athuraliya and Low [15] developed a practical implementation of their algorithms. Low [151] also studied flow control algorithms in several transmission control protocols. La and Anantharam [129] extended the algorithm introduced in [162].

Lazar [131, 132] studied dynamic flow control of M/M/1 queues, where the meaning of “dynamic” is that the decisions for users depend on the state of the network that the users see. He formulated a throughput maximization problem subject to a delay constraint. He showed that the problem was transformed to a linear programming one, and analytically derived an optimal dynamic flow control decision. Furthermore, he [133] expanded the problem of M/M/m queues. Hsiao and Lazar [73], and Korilis and Lazar [112] studied more general model. Hsiao and Lazar [73] obtained a threshold equilibrium for the network by the product form solution of the queue network. The existence of an equilibrium was shown in [112]. Korilis and Lazar [112] studied the existence of Nash Equilibria of the flow control model introduced in [73] for a general product-form network where the monotonicity assumptions of that reference did not necessary hold. Douligeris [40] considered multi-objective flow control in the model as [131, 132]. He formulated the model as a weighted-sum optimization problem, and showed some numerical results.

Libman and Orda [142] studied sliding-window strategies in a noncooperative flow control problem. Their goal is to minimize the expected cost/throughput ratio, and they proposed an algorithm for calculating optimal solution. Another dynamic flow control is studied by Imer and Başer [74].

Combined flow control and routing problems in a network of parallel links with multiple users have been studied by Haurie and Marcotte [72], La and Anantharam [126], Patriksson [172], Rhee and Konstantopoulos [175], Altman et al. [5], and Wang et al. [201]. They assumed that each link in the network was an M/M/1 queue with a determined capacity. In particular, Altman et al. [5] showed an asymptotic behavior in the case where the number of users was close to infinity. Dynamic flow control combined with network routing is considered in Wie [203]. In his work, a simple traffic network model is formulated as a noncooperative N-person game, and a dynamic mixed equilibrium is computed. Sahin and Simman [185] especially brought up a two-node parallel link network where their objective is to maximize a class of the values given by the throughputs minus the delays.

Fair flow control is also significant problem to guarantee QoS. In general, maximizing the total system performance is often incompatible with guaranteeing fairness [41]. Hence, a trade-off between fairness and the total system performance is also required. Mazumdar et al. [159] studied a fair flow control problem of a Jackson network model of M/M/1 queues with multiple users. They applied a cooperative game theoretical framework to the models and derive Nash bargaining solutions (it is referred to as the Nash arbitrated scheme in [159]). They showed that maximizing the product of the powers for all users gave a unique Nash bargaining solution. They also showed that maximizing the product of the average throughputs subject to bounds on the average delays for all users gave a unique Nash bargaining solution. Wong and deMarca [205] described fairness in window flow control in a network where a new performance measure was proposed. Chang [28] investigated the interrelationships of there conflicting performance criteria, delay, throughput and fairness in routing and flow control problems for a number of networks.

2.3 Load Balancing

A distributed computer system is a set of computers connected by a communication network. *Load balancing* is a technique to improve performance of a distributed computer system. It is to dispatch jobs for users among the resources (e.g., CPU, memory and disks) among the users, and is to avoid that particular set of resources are overloaded. Indeed, load balancing is similar to network routing (or resource sharing). A recent trend application in load balancing is GRID computing (clustering) [55].

Tantawi and Towsley [195] dealt with overall optimal load balancing in a distributed computer system with single class[3] job, where the system consists of heterogeneous host computers connected by a single channel communication network. More specifically, each node has a number of resources, and all nodes may have different configurations, that is, different processing rates. An external job arrives at each node according to a Poisson process. They formulated this distributed computer system as a class of product form queueing network models, and considered an optimal static load balancing policy which determines the optimal load at each node in order to minimize the average response time in the whole system. They proposed an algorithm (called a single-point algorithm) that determines the optimal load at each node for given system parameters. Kim and Kameda [102] considered the model in Tantawi and Towsley [195], and proposed another single-point algorithm. It is shown that the algorithm is simpler and its convergence is faster than the algorithm of Tantawi and Towsley [195]

Tantawi and Towsley [194] also studied a distributed computer system that consists of a set of heterogeneous host computers connected to a central host. They derived the optimality condition in this problem, and show a load balancing algorithm to solve the problem. Based on the their work, Kim and Kameda [102] proposed an improved load balancing algorithm to obtain the solution to the problem in [194]. In Tantawi and Towsley's model [194], however, there is only one-way traffic from the external nodes to

[3]In this section, we refer to a *user* as a *class*.

the central node in the sense that jobs can be forwarded for remote processing only from the external nodes to the central node. As an extension of this work, Li and Kameda [138] proposed an algorithm for optimal static load balancing in star network configurations with two-way traffic.

In [136, 137], Li and Kameda proposed an algorithm for optimal static load balancing in tree hierarchy network configurations. Kameda and Zhang [93] studied the uniqueness of solutions in optimal static load balancing of open BCMP [19] queuing networks. They obtained the linear relations that characterize the set of the optimal solutions.

The models presented above deal only with single job class environment. In [100, 101, 103], Kim and Kameda extended the single job class model by Tantawi and Towsley [195] to multiple job class environment where their assumptions were almost the same as Tantawi and Towsley, and they proposed an optimal static load balancing algorithm for multiple job classes. As a generalization, Li and Kameda [139] proposed an optimal static load balancing algorithm in a multi-class jobs distributed/parallel computer system with general network configurations. In fact, we can express any configuration with this models except for a single channel model [100, 101, 102, 103, 195]. Furthermore, Zeng and Veeravalli [207] proposed a more efficient algorithm to obtain a solution to the problem in [139], which was based on the Newton's method.

Noncooperative static load balancing is studied in [10, 25, 89, 91]. In [91], Kameda et al. studied Wardrop equilibria in static load balancing for single channel and star networks. They analytically derived the effects of the system parameters on the behavior of the system by overall optimum and Wardrop equilibrium. Altman et al. [10] focused on load balancing in two-node distributed systems, and showed the uniqueness and existence of the Nash equilibrium. Boulogne et al. [25] studied load balancing problems in distributed computing. They considered greedy algorithms, and showed the convergence of a Nash equilibrium.

Kleinberg et al. [104, 105] introduced fairness in load balancing. They dealt with homogeneous and heterogeneous load balancing and applied the Max-min fairness to these problems. Grosu and Chronopoulos [64, 65] studied noncooperative load balancing in distributed systems. They dealt with a system consisting of n heterogeneous M/M/1 queues shared by m users. They compared the fairness indices [81] of proportional [32], overall, noncooperative (Nash and Wardrop equilibria) schemes. Grosu et al. [66] studied static cooperative load balancing in the same model as [64, 65]. They formulated it as a Nash bargaining problem, and derived distributed load balancing algorithms to obtain Nash equilibria and Nash bargaining solutions. They compared the fairness index of cooperative with those of proportional, overall, noncooperative (Wardrop equilibrium) and shortest expected delay schemes. Touati et al. [200] applied the fairness objective proposed by Mo and Walrand [162] to a two-node load balancing system.

Dynamic (adaptive) load balancing is also an important problem in distributed computer systems. Eager et al. [45] studied dynamic load balancing policies in parallel M/M/1 queueing systems. They analytically derive the average queue lengths under random, threshold, and shortest policies. Dynamic load balancing in parallel queueing systems was studied by Down and Lewis [44]. They formulated his model as a Markov decision process [173] to minimize the average total costs consisting of the holding cost of jobs for each queue per unit time and the transfer cost of jobs. Zhou [210] studied dynamic load balancing in homogeneous distributed systems. Zhou compared seven load balancing algorithms with simulation. Kencl and Boudec [99] studied dynamic load balancing in a router model where different processors are dedicated to the data and to the control plane. They presented a novel packet processing scheme.

Mitzenmacher [161] studied dynamic load balancing in parallel M/M/1 queueing systems where arriving jobs made use of old information on the loads of the servers when it decided a route. In fact, if information on the loads of the servers is old, a system may behave badly. He investigated dynamic load balancing system models where load information was updated periodically or continuity. Dahlin [39] dealt with the same problem as [161]. He proposed load balancing algorithms that utilized old information on the loads of the servers, and examined the algorithms in large-scale parallel queueing systems through simulation experiments.

Altman and Shimkin [11] considered dynamic load balancing in a system consisting of a mainframe (MF) node and infinitely many personal computer (PC) nodes. The MF node is a processor sharing queueing system, and each PC node is identical and has a fixed service rate. In their model, each job sees the current system state, and noncooperatively chooses between the MF node and a PC node. They showed an optimal threshold policy corresponding to a Nash equilibrium. El-Zoghdy et al. [47] compared numerically static and dynamic load balancing in the same model as [11] where they considered overall and noncooperative (Nash equilibrium) situations.

2.4 Braess Paradox

The service providers (network administrators or agents) often face a necessity to improve the network performance. In this situation, they may try to add server capacities or network links in the network. Intuitively, we expect that the whole system performance improves when the capacity of a part of the system increases. In noncooperative optimization, however, it is well-known that the following phenomenon can occur: *Increased capacity of a part of the system may sometimes lead to the degradation in the benefits of all users*. Braess [26] first exhibited this *paradoxical* phenomenon, and Murchland [163], Stewart [191], Frank [56], Taguchi [192], Dafermos and Nagurney [38], and Cohen [34] evolved it. It is called the *Braess paradox*, and it attracted the interest of many researchers. A survey of the Braess paradox appears in [86], and many publications appear on the Braess's homepage (http://homepage.ruhr-uni-bochum.de/Dietrich.Braess/).

Braess [26] discovered a deterministic mathematical model of a congested network in a *Wardrop equilibrium*. That is, when a link (path) is added in the network and each user determines a path in order to minimize the mean response time unilaterally, the mean response times for all users in the network are worse than before adding the link. Later on, Frank [56], and Hagstrom and Abrams [67] gave mathematical properties of this model.

We show an example of a network where the Braess paradox occurs: Consider two networks as shown in Fig 2.1. Each network has one origin, four nodes, and one destination. The left hand side in Fig 2.1 has four links $(1, 2)$, $(1, 3)$, $(2, 4)$ and $(3, 4)$, and the right hand side has a link $(2, 3)$ in addition to the links that the former has. That is, the former is before adding a link and the latter is after adding link.

We consider a routing problem in these networks where each individual user determines the best path unilaterally. Each network has a fixed traffic flow rate λ from the origin to the destination. A traffic needs the time $C_{ij}(x)$ when it moves from node i to node j where x is the traffic flow rate in link (i, j). Now, we set $C_{12}(x) = C_{34}(x) = 10x$, $C_{13}(x) = C_{24}(x) = 50 + x$, $C_{23}(x) = 10 + x$ and $\lambda = 6$. Then, the flow of each path in the former is 3, and the total time of traffic in the former is 83. On the other hand, the flow of each path in the latter is 2, and the total time of traffic in the latter is 92. Hence, it shows that adding a link leads the performance degradation for all users.

Cohen and Kelly [36] reported the Braess paradox in a queueing network model based on the model described in [26]. Their network consists of an entrance, four queues (two First-Come-First-Served (FCFS) queues and two Infinite-Server (IS) queues), and an exit, and they assume that arrival streams are according to a Poisson process. They mathematically showed a Braess paradox in optimal static routing by individual users for the network. Calvert et al. [27] found a paradox in dynamic (state-dependent) routing for the same network as that studied by Cohen and Kelly [36]. In their study, individual users know full information of all the instantaneous queue lengths of the network servers and they can make use of that information in their dynamic routing. They analytically derived the recursive equations that lead a dynamic routing decision, and showed that a Braess paradox occurred in dynamic routing for the network analogous to what Cohen and Kelly showed.

Cohen and Jeffries [35] reported examples of single-server queueing networks in which adding servers or increasing the processing capacity of existing servers lead to degrading the network performance. Bean et al. [20] found a paradox occurred even in a loss

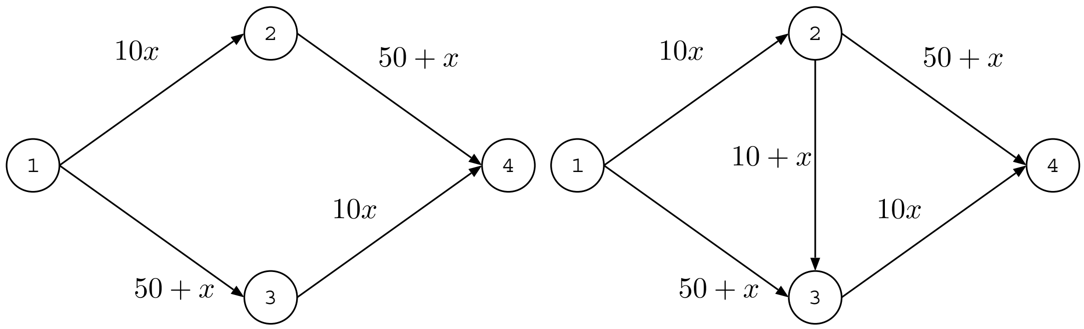

Figure 2.1: The Braess networks: **Left:** Before adding link $(1,2)$. **Right:** After adding link $(1,2)$.

network. Arora and Sen [14] studied a Braess paradox in software multi-agent systems where a genetic algorithm (GA) was used to evolve agent societies that are faced with the Braess paradox. Masuda and Whang [155] considered a capacity expansion/reduction problem for a network in individual routing. They further developed a mechanism to avoid Braess paradoxes.

As described above, there are many studies about paradoxes in optimal routing by Wardrop equilibria. Similarly to them, it had been expected that a paradox could also occur in Nash equilibria. Indeed, a Nash equilibrium corresponds to a Wardrop equilibrium when the number of users is infinitely large [72]. Hence, we can also expect that in a Nash equilibrium, a similar type of paradox occurs. In [87], this paradox is called the *Braess-like paradox*. Korilis et al. [113] found some examples wherein the Braess-like paradox appears in a class optimum where all user classes are identical in the same topology for which the original Braess-like paradox (for the individual optimum) was in fact obtained. Furthermore, Korilis et al. [115, 116], and Altman et al. [8] derived a sufficient condition for avoiding the paradox.

Kameda et al. [87, 88, 89, 92] found a paradox in load balancing for distributed computer systems. In [89, 91, 208], they showed that increased capacity of a part of a system might lead to the degradation of the overall performance measure in distributed computer system models in Wardrop and Nash equilibria. They called it the *weaker Braess-like paradox*. Furthermore, Kameda et al. [89] noticed that in a Nash equilibrium, mutual forwarding between two processing nodes (servers) occurred although that should never occurred in the overall optimum and the Wardrop equilibrium in the same system model. Finally, Kameda et al. [87] discovered some cases where a Braess-like paradox appeared in the Nash equilibrium. They dealt with a distributed computer system model consisting of two nodes and a communication network that connects both nodes each of which has a single exponential server and a Poisson arrival stream. They considered two types of communication networks: a single-channel communication network and a network consisting of two-way communication lines, and showed that a paradox occurred in the system with either type of the network in Nash equilibria. Furthermore, Kameda et al. [88] investigated the model in [87] numerically, and show that a paradox appears most strongly in the case where arrival and service rates of nodes are identical.

Later on, Kameda and Pourtallier [92] considered a system with homogeneous nodes (i.e., both job arrival and server processing rates of each node are identical). Considering static load balancing for such a system, we intuitively may expect that no mutual forwarding among nodes since it leads to the degradation of the system performance for every user. They, however, showed that such paradoxical behavior occurred in a Nash equilibrium. Also, they found that the performance degraded without bound in their distributed computer system.

As described above, there are many publications about the Braess paradox. Unfortunately, it seems that finding the existence of a paradox in a given network effectively is impossible [178].

The Degree of Paradox: Other interest in Braess Paradox is to investigate, "How system performance is degraded." *The degree of paradoxes* [47, 48, 49, 50, 86] is a measure to show the degradation.

It is defined as follows: We consider a system shared by n users, and two states $\boldsymbol{x}$ and

$\boldsymbol{x}'$ in the system. $\boldsymbol{x}$ is a state before adding a link or capacity, and $\boldsymbol{x}'$ is a state after adding. Let $C_i(\boldsymbol{x})$ and $C_i(\boldsymbol{x}')$ denote the costs of user i, $i = 1, 2, \ldots, n$ at the system states are $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively. Then,

$$k^* = \min\{k_i,\ i = 1, 2, \ldots, n\},\ k_i = \frac{C(\boldsymbol{x}')}{C(\boldsymbol{x})}. \tag{2.2}$$

is called *the degree of Paradox.* If k^* is greater than 1, then we see that a Braess paradox occurs in the system.

Kameda [84] investigated a paradox in general networks that had been studied by Cohen, Kelly and Jeffries [35, 36]. He considered to use the measure of cost degradation as the ratio of the cost for each user of a network before adding capacity (a link) to that after adding capacity. That is, performance degradation occurs in a network if the measure is larger than 1. He showed a value of the measure was at most 2 for any general Braess network. Furthermore, Roughgarden and Tardos [178, 179, 182] showed more general results. They dealt with a direct network with N nodes including one source node and one sink node. Then, they showed that the degree of a paradox was not over $n/2$ in the network. In particular, the degree of a paradox is not over 4/3 when the cost functions are linear. A stronger bound on Braess's paradox is showed in [143]. Kameda [85] studied the bounds on the degrees of coincident cost improvement and degradation of Wardrop and Nash equilibria in Braess network and distributed computer system models.

El-Zoghdy et al. [47, 48, 49, 50] studied the impact of the worst-case degree of the paradox (WCDP) on the values of system parameter through a number of numerical studies in a distributed computer system. They showed that WCDP was the largest in the case where the parameter values for each node were completely symmetric, and job arrival rate approached the server processing rate. Lin et al. [144] studied the worst-case severity of Braess' paradox in multi-commodity networks.

Chapter 3

A Paradox in Flow Control of M/M/m queues

3.1 Introduction

In computer and communication networks, flow control is one of the important means to utilize limited system resources effectively and to guarantee proper Quality of Service (QoS). Flow control adjusts the input flow (throughput) in order to provide good performance. Considering an optimal flow control problem, one may face the trade-off between the throughput and the response time. These two performance measures are mutually contradictory, that is, if one improves the system throughput, then the system response time degrades, and vice versa. Therefore, as the utility of each user (player), we use the power that is the quotient of the throughput divided by the average response time for the user (see, e.g., Giessler et al., [62], Kleinrock [107, 108]).

We consider a system where multiple users (players) share an M/M/m queue and where the utility of a player is the power. We can think of two typical performance optimization schemes: the *noncooperative optimization* scheme and the *overall optimization* scheme. In the noncooperative optimization scheme, each player strives to optimize unilaterally its own power given the decisions by others. In the overall optimization scheme, a single agent optimizes a single performance measure that is the total sum of the powers of all players. The former is regarded as a noncooperative game, and the equilibrium is called a *Nash equilibrium* [165]. In this chapter, we study the *Nash equilibrium*.

Douligeris and Mazumdar [43], and Zhang and Douligeris [209] studied algorithms to obtain Nash equilibria in flow control of M/M/1 queues with multiple users. Their performance objective was to maximize the powers of all players. They proposed greedy algorithms, and showed convergence properties of them. State-dependent flow control was analyzed by Hsiao and Lazar [73], and Korilis and Lazar [112]. They considered a closed queueing network model, and maximized the average throughput subject to an upper bound on the average response time. In particular, Korilis and Lazar [112] derived the existence of equilibria using fixed-point theorems. Altman et al. [5] combined flow control and routing in a network model with several parallel links. Lazar [133] studied optimal flow control problems of an M/M/m queue where one player maximizes the throughput subject to the constraint that the average time delay should not exceed a specified value. In this chapter, we deal with flow control problems of M/M/m queues that have multiple players, where each player strives to optimize unilaterally its own power.

Consider the operation of grouping together separated systems into one system. Due to enhanced flexibility in resource utilization, we expect that system grouping improves the system performance. For example, it has been shown analytically that grouping together separated M/M/m queues improves the average response time although the utilization factor of each server remains the same (see [106]). We therefore expect performance improvement in term of power by grouping together separate systems.

In noncooperative optimization of routing in networks and load balancing in distributed systems, however, it is known that the following phenomenon can occur, that is, grouping separated systems together and/or adding connections to the system may sometimes degrade the utilities for all users. The first example is the Braess paradox [26]. Other examples have been presented, for example, by Bean et al. [20], Calvert et al. [27], Cohen and Jeffries [35], Cohen and Kelly [36], Kameda et al. [87], Kameda and Pourtallier [92], Korilis et al. [113] [116], Roughgarden and Tardos [182]. However, most of the previous results are on routing (or load balancing) problems, and the paradox is observed in very limited models. Flow control is essentially different from them. In optimal flow control, it seems that no such paradoxes have been reported yet.

In this chapter, we show that a paradox similar to the above ones may occur in noncooperative optimal flow control of M/M/m queues. More specifically, we show a case where grouping together separated M/M/m queues leads to the degradation of the power to all players in noncooperative optimal flow control.

The rest of this chapter is organized as follows. In Section 3.2, we describe a queueing system model and formulate overall and noncooperative optimal flow control schemes as nonlinear programming problems. In Section 3.3, we show the case of a paradox in noncooperative flow control of M/M/m queues. Finally, in Section 3.4, we conclude this chapter. In Appendix A, we show the existence and the uniqueness of solutions to the optimization problems, and present algorithms that obtain the solutions.

3.2 The Model and Problem Formulation

We first consider an M/M/m queueing system. The system has an arrival stream of jobs that forms a Poisson process with rate Λ (jobs per unit time). Also, the system has n servers. The processing time of a job at each server is independent, identically distributed and according to an exponential distribution with mean $1/\mu$. Then, the mean response time T of an arbitrary job is given by

$$T = T(\Lambda) = \begin{cases} \dfrac{B_m(\Lambda)}{m\mu - \Lambda} + \dfrac{1}{\mu}, & \text{if } 0 \le \Lambda < m\mu \\ \infty, & \text{if } \Lambda \ge m\mu \end{cases} \tag{3.1}$$

for $l = 1, 2, \ldots, r$, where

$$B_m(\Lambda) = \left[1 + \sum_{k=0}^{m-1} \frac{m!(1 - \frac{\Lambda}{m\mu})}{k!(\frac{\Lambda}{\mu})^{m-k}} \right]^{-1} \tag{3.2}$$

is the probability that all servers are busy (called *the Erlang delay formula*).

We next consider the M/M/m queueing system with r players. Each player sends the multiple-server an arrival stream of jobs that is mutually independent and forms a Poisson arrival process with rate λ_l, $l = 1, 2, \ldots, r$. Note that $\Lambda = \sum_l \lambda_l$ and that the processing time is independent of the player that sends the job. Then the mean response time T of an arbitrary job in this model is given by (3.1).

As the utility for each user in the flow control problems, we use the power for the user. Since the throughput is equivalent to the rate of the arrival flow (i.e., the arrival rate), from [108], the overall power P of the system and the power p_l of player l are given as:

$$P = P(\Lambda) = \begin{cases} \dfrac{\Lambda}{T(\Lambda)}, & \text{if } 0 \le \Lambda \le m\mu, \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

and

$$p_l = p_l(\lambda_1, \lambda_2, \ldots, \lambda_l, \ldots, \lambda_r) = \begin{cases} \dfrac{\lambda_l}{T(\Lambda)}, & \text{if } 0 \le \lambda_l \le m\mu - \sum_{j \ne l} \lambda_j, \\ 0, & \text{otherwise,} \end{cases} \tag{3.4}$$

for $l = 1, 2, \ldots, r$, respectively, where note that $\Lambda = \sum_l \lambda_l$. From (3.3) and (3.4), we have $P = \sum_l p_l$. P and p_l are positive for $\Lambda < m\mu$, and zero for $\Lambda = 0$ and $\Lambda = m\mu$.

We formulate, for the system described above, two typical optimal flow control schemes: the noncooperative optimization scheme (I) and the overall optimization scheme (II). The noncooperative and overall optimization schemes in an M/M/m queueing system with r players are presented as follows:

(I) The noncooperative optimization scheme: Each player strives to maximize unilaterally its own power, that is, the noncooperative optimization is to find $\widehat{\lambda}_l$ for each $l = 1, 2, \ldots, r$, that satisfies:

$$\widehat{p}_l = \max_{\lambda_l \ge 0} p_l(\widehat{\lambda}_1, \widehat{\lambda}_2, \ldots, \lambda_l, \ldots, \widehat{\lambda}_r).$$

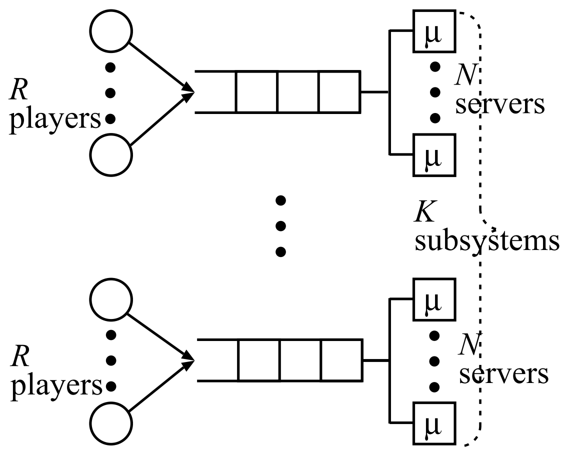

Figure 3.1: A system consisting of K separated M/M/N queues (S_I).

As shown in our numerous experiments we always observed symmetric solution to $\widehat{\lambda}_l$ and $\widehat{p}_l$, that is

$$\widehat{\lambda}_l = \widehat{\lambda} = \frac{\widehat{\Lambda}}{r}, \text{ and } \widehat{p}_l = \widehat{p}, \; k = 1, 2, \ldots, r. \tag{3.5}$$

(II) The overall optimization scheme: A single agent maximizes the overall power of the system, that is, strives to find $\widetilde{\Lambda}$ that satisfies:

$$\widetilde{P} = P(\widetilde{\Lambda}) = \max_{\Lambda \geq 0} P(\Lambda).$$

Then, the agent distributes equally to each player l the power $\widetilde{p}_l$, and thus the throughput $\widetilde{\lambda}_l$. Thus,

$$\widetilde{p}_l = \widetilde{p} = \frac{\widetilde{P}}{r}, \text{ and } \widetilde{\lambda}_l = \widetilde{\lambda} = \frac{\widetilde{\Lambda}}{r}. \tag{3.6}$$

In Appendix A, we show exsitence and uniqueness of noncooperative and overall optimal solutions, and provide algorithms that obtain them.

Based on the above definitions, we consider two queueing system models as shown in Figures 3.1 and 3.2:

1) One system, called S_I, consists of K subsystems each of which consists of a separated

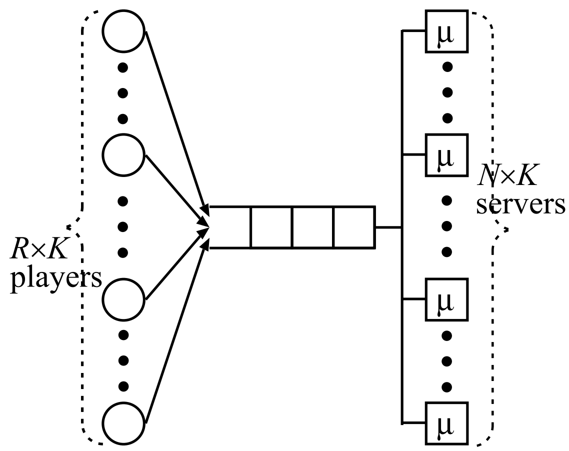

Figure 3.2: An M/M/($N\times K$) queue (S_U).

M/M/N queue (N exponential servers with Poisson arrivals of jobs) and R independent players (Figure 3.1).

2) The other system, called S_U, results from grouping together all the above separated M/M/N queues. That is, the system consists of one M/M/($K \times N$) queue and $K \times R$ independent players (Figure 3.2).

Obviously, we can formulate each subsystem of S_I and the system S_U using M/M/m queues where $m = N$ and $r = R$ in S_I, and $m = K \times N$ and $r = K \times R$ in S_U, respectively. We express the parameters of a system by a quadruplet. That is, S_I given in 1) corresponds to (N, R, K, μ), and S_U given in 2) corresponds to $(N \times K, R \times K, 1, \mu)$. We note that in S_I the power of users is independent of the value of K. Therefore, the solution of S_I for a given (N, R, K, μ) is identical to that of a system with parameters $(N, R, 1, \mu)$.

For both S_I and S_U, there are $K \times R$ players numbered by $(1, 1)$, $(1, 2)$, $\ldots$, $(K, 1)$, $\ldots$, (K, R). Denote the throughput of player (i, l) by λ_{il}. A subsystem k of S_I, has R players $(k, 1)$, $(k, 2)$, $\ldots$, (k, R). Note that for system S_U, we associate the symbols with the mark $'$.

From (3.5), in the noncooperative optimization scheme, we have for each subsystem k of S_I

$$\widehat{\lambda}_{kl} = \widehat{\lambda} = \frac{\widehat{\Lambda}}{R}, \text{ and } \widehat{p}_{kl} = \widehat{p}, \; l = 1, 2, \ldots, R, \tag{3.7}$$

and for system S_U

$$\widehat{\lambda}'_{il} = \widehat{\lambda}' = \frac{\widehat{\Lambda}'}{K \times R}, \text{ and } \widehat{p}'_{il} = \widehat{p}', \tag{3.8}$$

for $i = 1, 2, \ldots, K, \; l = 1, 2, \ldots, R$.

We define the degree of the paradox, δ, as follows:

$$\delta = \begin{cases} \dfrac{\widehat{p} - \widehat{p}'}{\widehat{p}}, & \text{if } \widehat{p} > \widehat{p}', \\ 0, & \text{otherwise.} \end{cases} \tag{3.9}$$

Note that a paradox occurs when $\delta > 0$.

From (3.6), in the overall optimization scheme, we have for each subsystem k of S_I

$$\widetilde{\lambda}_{kl} = \widetilde{\lambda} = \frac{\widetilde{\Lambda}}{R}, \text{ and } \widetilde{p}_{kl} = \widetilde{p} = \frac{\widetilde{P}}{R}, \; l = 1, 2, \ldots, R, \tag{3.10}$$

and for system S_U

$$\widetilde{\lambda}'_{il} = \widetilde{\lambda}' = \frac{\widetilde{\Lambda}'}{K \times R}, \text{ and } \widetilde{p}'_{il} = \widetilde{p}' = \frac{\widetilde{P}'}{K \times R}. \tag{3.11}$$

for $i = 1, 2, \ldots, K, \; l = 1, 2, \ldots, R$.

3.3 A Case of a Paradox

In this section, we show a case where a paradox occurs. Without loss of generality, we assume a time scale such that $\mu = 1$. We compare the two systems presented in Section 3.2: S_I – a system consisting of K separated M/M/N queues (Figure 3.1) and S_U – an M/M/($N\times K$) queue (Figures 3.2). In the former, the flow control schemes are concerned with each separated M/M/N queue. On the other hand, in the latter, the flow control schemes are concerned with the M/M/($K \times N$) queue. We recall that the latter is the result of grouping together the former.

3.3.1 The Results

Table 3.1: Power of each player and degree of paradox in the noncooperative optimization scheme, for $N = R = 11$ and various values of K.

K	1	2	4	8	16
power of each player	0.4248	0.4208	0.4135	0.4061	0.3999
degree of paradox δ	0.000	0.009	0.027	0.044	0.059
K	32	64	128	256	512
power of each player	0.3951	0.3915	0.3888	0.3868	0.3854
degree of paradox δ	0.070	0.078	0.085	0.089	0.093

In Table 3.1 and Figures 3.3 and 3.4, we show the cases where $N = R = 11$ with the numbers N of servers and R of players in each subsystem.

Table 3.1 and Figure 3.3 illustrate how the noncooperative optimal powers of each player in S_I and S_U depend on the number K of the subsystems grouped together. Figure 3.3 shows a trend observed for the noncooperative optimization scheme. That is, in the scheme, the power of each player in S_U *decreases* as the number K of subsystems grouped together *increases*, for $N = 11$ and $R = 11$, as shown by the curve associated with the symbol S_U. On the other hand, naturally, the power of each player of each separated subsystem S_I remains constant regardless of the value of K as shown by the line associated with the symbol S_I. The trend looks counter-intuitive. Thus, we should not always expect that system grouping improves the system performance in the noncooperative flow control although we anticipate the opposite. As shown in Table 3.1, the degree of

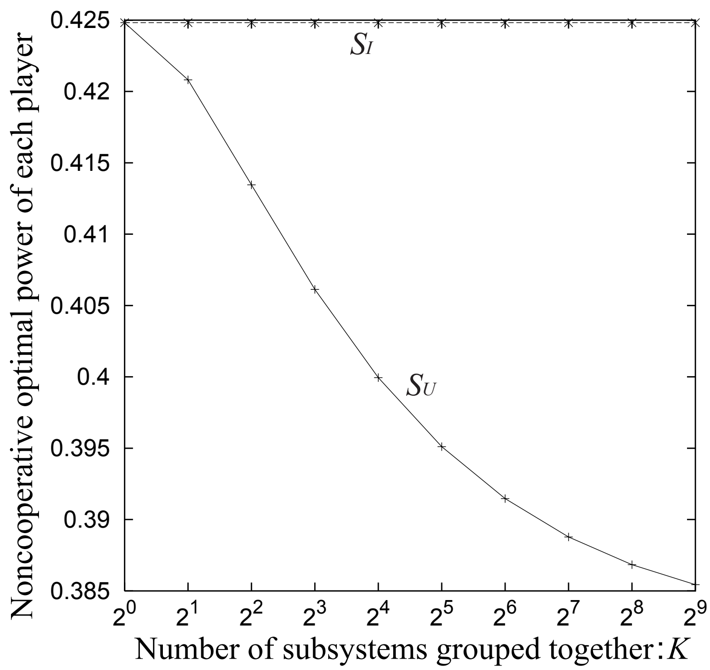

Figure 3.3: Noncooperative optimization scheme

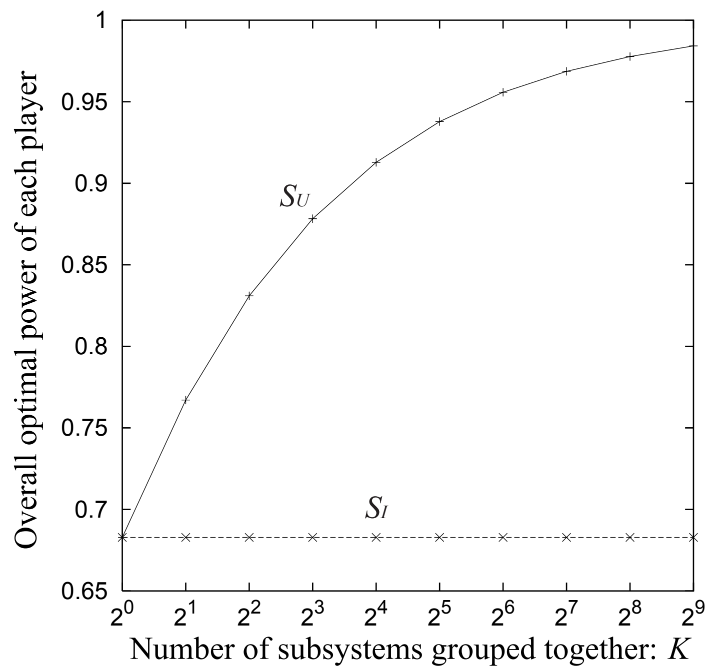

Figure 3.4: Overall optimization scheme

paradox increases in K.

On the other hand, Figure 3.4 shows a naturally-expected trend observed for the overall optimization scheme. That is, in the scheme, the power of each player in S_U increases as the number K of subsystems grouped together increases, for $N = 11$ and $R = 11$, as shown by the curve associated with the symbol S_U. On the other hand, naturally, the power of each player of each separated subsystem S_I remains constant regardless of the value of K as shown by the line associated with the symbol S_I.

3.3.2 Discussion of the Results

To give some idea on the above-mentioned results, we present Figure 3.5 for the case where $N = R = 1$. In the overall optimization scheme, each player always enjoys the increase in the power as the number K of subsystems grouped together increases. On the other hand, in the noncooperative optimization scheme, for the values of K from 1 till about 10, the power of each player increases although the improvement is smaller than that of the overall optimization scheme. For the values of K larger than 11, the power of each player decreases as K increases, which shows the same trend as the above-mentioned paradoxical result. Note, from the definition, that the system S_U with parameters $(1, 1, \kappa, \mu)$ is identical with each subsystem of the system S_I with parameters $(\kappa, \kappa, 1, \mu)$ for any $\kappa \geq 1$. Therefore, each curve associated with the symbol S_U in the Figures 3.3 and 3.4 that starts at $K = 1$ with $R = N = 11$ shows the part of each corresponding curve in the Figure 3.5 for the domain of $K \geq 11$ with $R = N = 1$. In Figure 3.5 ($R = N = 1$), we see that, in the noncooperative scheme, the power of each user is the largest for $K = 11$ (which is identical to that of a subsystem of S_I with $R = N = 11$) and decreases as K increases. Therefore, we see that if $R = N$ the paradox is the largest for the case with S_I with $R = N = 11$ and the value of K as large as possible.

In the following, we seek the underlying structure of the system that may lead to the fact that the paradox is the largest for the case with S_I with $R = N = 11$ and the value

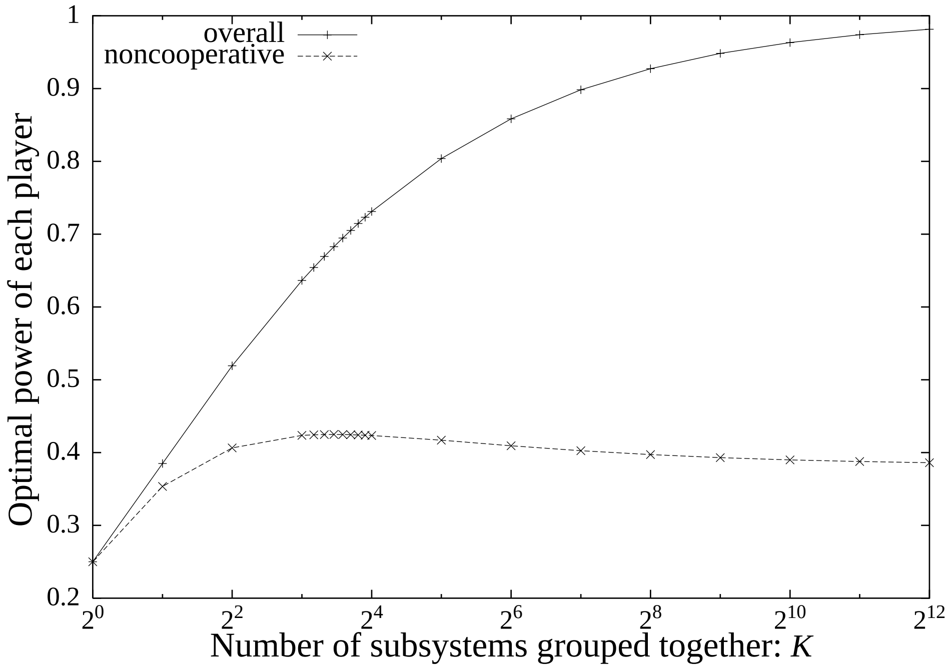

Figure 3.5: Powers of each player in the noncooperative ($\widehat{p}$) and overall optimization ($\widetilde{p}$) schemes as functions of the number K of subsystems grouped together, for $N = R = 1$.

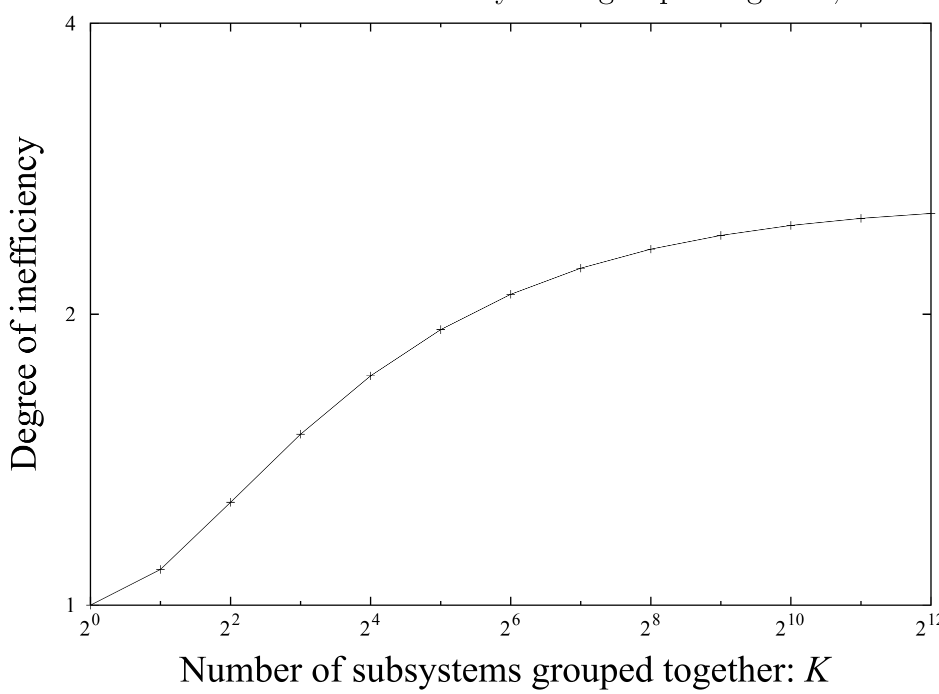

Figure 3.6: Degree of inefficiency of the Nash equilibrium ($\widetilde{p}/\widehat{p}$), for $N = R = 1$.

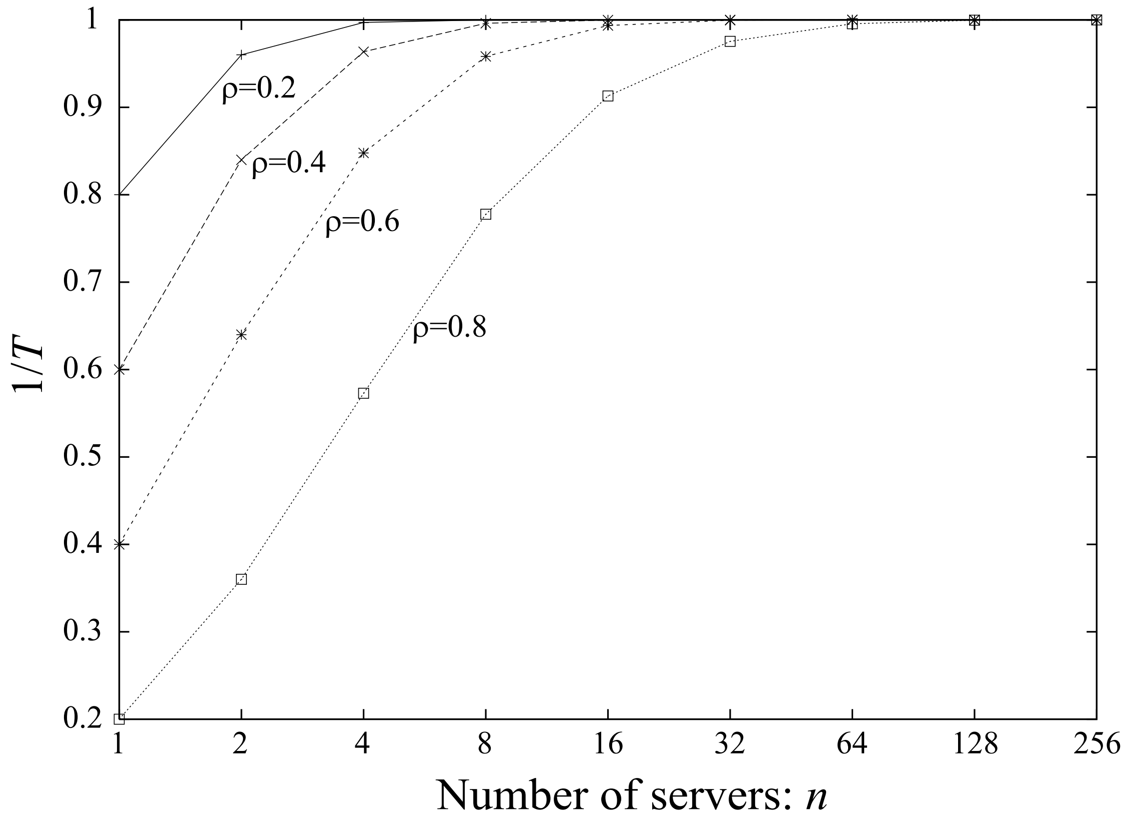

Figure 3.7: [**Effect of the number of servers on the mean response time of M/M/n**]: The graph shows how $1/T$ (T is the mean response time given by (3.1)) increases as the number of servers increases with the traffic intensity ρ at each server remaining the same.

of K as large as possible. We see in Figure 3.6 that the degree of the inefficiency of the Nash equilibrium ($\widetilde{p}/\widehat{p}$) with $N = R = 1$ increases as K increases. That is, the possibility of paradox may start at some small value of K. We call this the effect A. On the other hand, we see in Figure 3.7 that for small values of K, the grouping of servers improves the performance (responsiveness) of the system although the traffic intensity ρ of each server is kept identical, while this improvement is very small for large values of K. We call this the effect B. We, therefore, see that the effects A and B eliminate each other for small values of K whereas only the effect A is remarkable for the large values of K. Thus, as shown in Figure 3.5, in the noncooperative optimization scheme with $N = R = 1$, the power is the largest for some intermediate value of K, that is in fact 11, and decreases as K increases. Therefore, we see that the paradox is the largest for the case with S_I with $R = N = 11$ and the value of K as large as possible.

Furthermore, we have investigated the paradoxical behaviors exhaustively regarding the various combinations of the values of the number, N, of servers and the number, R, of players in each independent subsystem and the number, K, of subsystems to be grouped together. Figures 3.8 and 3.9 show the degree of paradox for various combinations of the values of N and R in the two cases where $K = 16$ and 512, respectively. Also, we observe that the degree of paradox tends to be larger in the cases where the values of N is nearly equal to R than the other cases relevant to them. Thus, we observe again that the worst case of paradox may occur in the case where $N = R = 11$ and K has a large value. We have seen so far, however, no simple underlying structure that would support the fact that the worst case of paradox occurs in the case where $N = R$.

3.4 Conclusion

In this chapter, we have studied the existence of a paradox in noncooperative flow control in M/M/m queues. We have formulated the overall and noncooperative optimal flow control schemes of maximizing the power and have shown the existence and uniqueness of

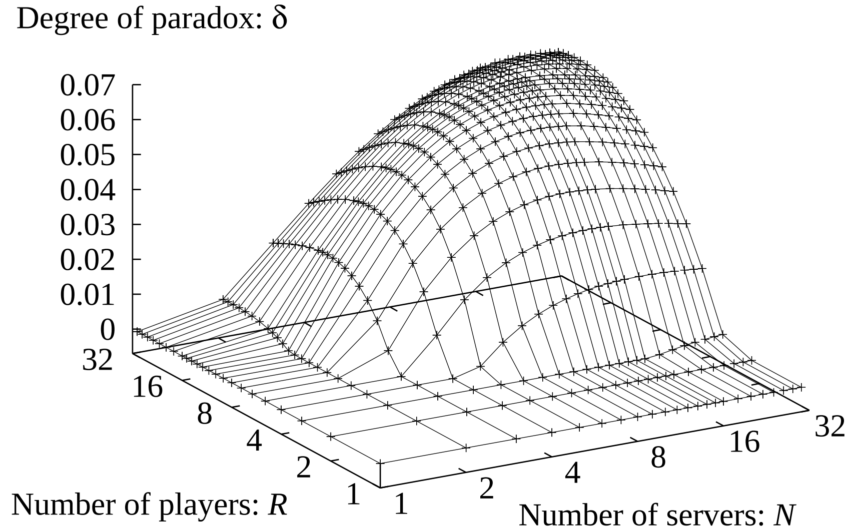

Figure 3.8: Degree of paradox δ in noncooperative flow control for various combinations of N and R in the case where $K = 16$.

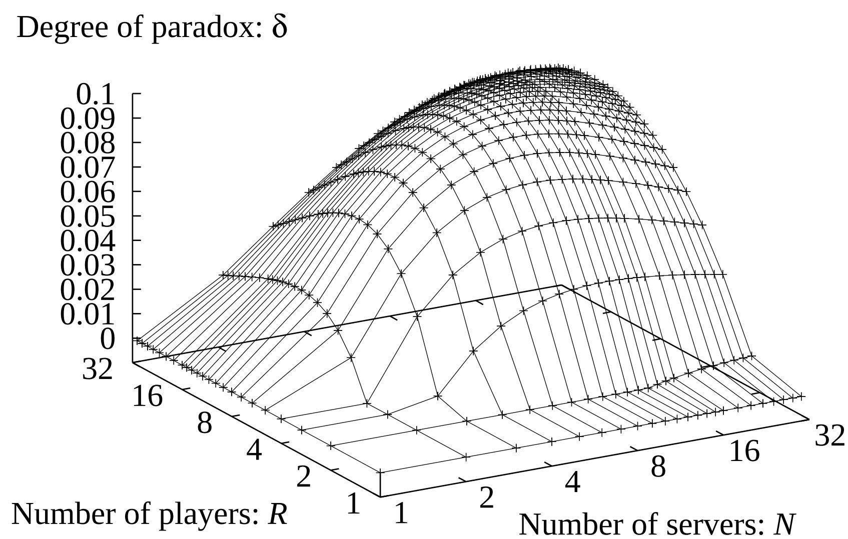

Figure 3.9: Degree of paradox δ in noncooperative flow control for various combinations of N and R in the case where $K = 512$.

solutions. We have found a paradoxical behavior in noncooperative optimal flow control similar to the Braess paradox of noncooperative optimal routing.

We have been interested in this research because optimal flow control may essentially be different from optimal routing and also because the paradox has been found to occur only in limited types of problems except optimal flow control. Our example of the paradox found in optimal flow control would suggest that the paradox could occur also in various other types of problems.

Chapter 4

Braess Paradox in Dynamic Routing for the Cohen-Kelly Network

4.1 Introduction

An important problem in current high-speed and large-scale computer and communication networks (e.g., GRID, Internet) is to provide all users with satisfactory network performance. Intuitively, we expect that the total performance of a network will increase in its capacity. In noncooperative optimization (which results in a Nash [165] or Wardrop [202] equilibrium), however, adding capacity or a link to a network may sometimes degrade the performance for all users as exemplified by the Braess paradox [26]. That is, Braess [26] considered a network routing problem in which all the costs of the links are linear. In a Wardrop equilibrium, he discovered a paradoxical phenomenon as described above. The Braess paradox attracted attention of many researchers, and was found in various types of network optimization problems, for example, Cohen and Kelly [36], Bean et al. [20], Calvert et al. [27], Cohen and Jeffries [35], El-Zoghdy et al. [50], Korilis et al. [113, 116], Kameda [84], Kameda et al. [87, 92], Inoie et al. [78], and Roughgarden and Tardos [182].

Cohen and Kelly [36] studied static routing in two queueing networks, called the *initial* and *augmented* networks. The initial network has two paths, and the augmented network has an additional path. An individual user arriving at the network only knows the expected transit time of each path in the network. In that situation, they showed the existence of a paradox. Also, in [36], the authors raised the question of whether the paradox also occurs in networks where users have knowledge not only on the expected transit time of each path in the network, but also on the instantaneous length of each queue.

Calvert et al. [27] considered dynamic routing in the same networks as introduced by Cohen and Kelly [36]. In their study, individual users have full knowledge of all the instantaneous queue lengths of the network servers and they can make use of that knowledge in their dynamic routing decisions. They analytically derived the recursive equations that provide the dynamic routing decisions. Through some simulation experiments, they contended that they showed that a Braess paradox occurred in dynamic routing analogous to what Cohen and Kelly showed.

To show the existence of the paradox, however, they only compared between the overall mean transit times of jobs in the initial and augmented networks. That is, they did not show that the mean transit time of jobs routed through *any* path was higher in the augmented network than in the initial network.

Furthermore, the system used in the simulation study performed by Calvert et al.[27] does not seem to be identical to the one they used in their analytical study. We particularly note that, in the analytical study, they assumed that the total number of jobs entering the system is finite while in their simulation study, the total number of jobs entering the system is infinite.

In this chapter, we therefore deal with dynamic routing for the Cohen-Kelly network. Based on the assumptions and formulations of the network in [27], we derive dynamic routing decisions in the initial and augmented networks. We make experiments on the simulation system that reflects exactly the analytical model given in [27]. Through the simulation experiments, we show a case where the paradox occurs for all users. Also, we compare our results with the results by Calvert et al. [27], and point out that their results are not sufficient to show that a paradox occurs in the network, in our sense.

This chapter is organized as follows. In section 4.2, we describe the Cohen-Kelly network and derive the dynamic routing decision. In section 4.3, we describe our simulation method, and show some simulation results. In section 4.4, we conclude this chapter.

4.2 The Model and Assumptions

We consider the two networks as shown in Figures 4.1 and 4.2. Note that the model and assumption described in this section are based on the analytical study of Calvert et al. (see Section 2 in [27]). We call the former the *initial network* and the latter the *augmented network*. Both networks consist of an entrance (node 0), four queues (nodes 1, 2, 3 and 4), and an exit (node 5). Nodes 1 and 4 are first-come-first-served (FCFS) queueing systems, each of which has an exponential server with mean $1/\mu_1$ and $1/\mu_4$, respectively. Nodes 2 and 3 are queueing systems with infinite servers (IS), each of which service time is exponentially distributed with mean $1/\mu_2$ and $1/\mu_3$, respectively. Let $x_1(t)$, $x_2(t)$, $x_3(t)$ and $x_4(t)$ denote the numbers of jobs in nodes 1, 2, 3 and 4 at time t, respectively.

A flow of jobs arrives at the system according to a Poisson process with rate λ. As assumed in [27], the total number of jobs that enter the network is finite, and we denote it by N. Also, we denote the number of jobs before arriving at Q_0 by x_0.

Let D denote the set of states in the whole system using dynamic routing, that is,

$$D = \{\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{N}^4 \text{ and } x_0 + x_1 + x_3 + x_4 \leq N\}, \tag{4.1}$$

where $\boldsymbol{x} = (x_0, x_1, x_3, x_4)$ is the state vector for dynamic routing decision, and $\mathbb{N} = \{0, 1, 2, \ldots\}$. Note that we can ignore x_2 in any routing decisions since node 2 is an IS queueing system, and hence the expected transit time of jobs routed through any path is not affected by x_2.

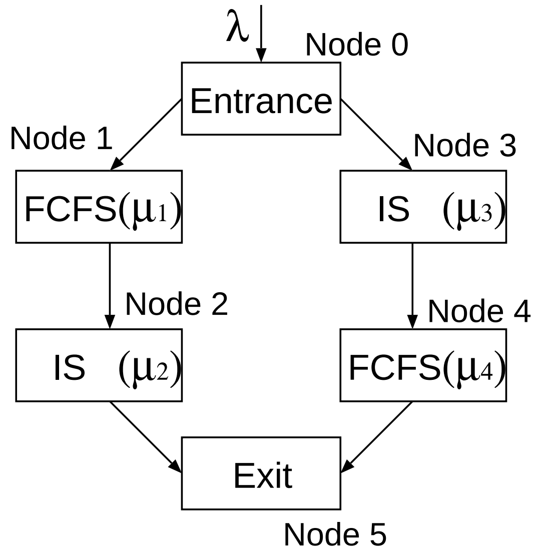

Figure 4.1: Initial Network: The network has two paths (0-1-2-5 and 0-3-4-5).

4.2.1 Initial Network

The initial network (Figure 4.1) has two paths, $0-1-2-5$ and $0-3-4-5$, and a job dynamically chooses one of these paths in order to minimize its own expected transit time from the entrance (node 0) to the exit (node 5). Let $T_1^I(\boldsymbol{x})$ and $T_3^I(\boldsymbol{x})$ denote the expected transit times of jobs routed through nodes 1 and 3, respectively, when the job sees at node 0 that the system state is $\boldsymbol{x}$ in the initial network.

We denote the two subsets of states in the whole system in the initial network by D_1^I and D_3^I where we use them for dynamic routing in the initial network. D_1^I and D_3^I are given as follows:

$$D_1^I = \{\boldsymbol{x} \mid \boldsymbol{x} \in D \text{ and } T_1^I(\boldsymbol{x}) \leq T_3^I(\boldsymbol{x})\}, \tag{4.2}$$

and

$$D_3^I = \{\boldsymbol{x} \mid \boldsymbol{x} \in D \text{ and } T_1^I(\boldsymbol{x}) > T_3^I(\boldsymbol{x})\}. \tag{4.3}$$

Note that $D_1^I \cap D_3^I = \phi$ and $D_1^I \cup D_3^I = D$. If a job sees at node 0 that the system state is in D_1^I, then the job is routed through node 1, and otherwise (i.e., the system state is in

D_3^I) it is routed through node 3.

To obtain D_1^I and D_3^I, we need to give expressions of $T_1^I(\boldsymbol{x})$ and $T_3^I(\boldsymbol{x})$. When a job at node 0 sees that the system state is $\boldsymbol{x}$, then the expected delay of the job at node 1 is $(x_1+1)/\mu_1$. Noting that the expected delay of any job at node 2 is $1/\mu_2$, then T_1^I is given by

$$T_1^I(\boldsymbol{x}) = \frac{x_1+1}{\mu_1} + \frac{1}{\mu_2}, \;\; \boldsymbol{x} \in D. \tag{4.4}$$

To express T_3^I we suppose that a job (called the *marked job*) sees at node 0 that the system state is $\boldsymbol{x}$, and that it is routed through node 3. Then, the network has the state-transitions depicted in Table 4.1. Note that the inter-event time is exponentially distributed with mean $1/(I_{X_0}\lambda + I_{X_1}\mu_1 + (x_3+1)\mu_3 + I_{X_4}\mu_4)$ where $X_i = \{x_i > 0\}$, and

$$I_X = \begin{cases} 1, & \text{if } X \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

Then, $T_3^I(\boldsymbol{x})$ is given by the following equation:

$$\begin{aligned} T_3^I(\boldsymbol{x}) &= \frac{1}{I_{X_0}\lambda + I_{X_1}\mu_1 + (x_3+1)\mu_3 + I_{X_4}\mu_4}\Big[1 \\ &+ I_{X_0}\lambda T_3^I(x_0-1, x_1+I_{C_1}, x_3+1-I_{C_1}, x_4) \\ &+ I_{X_1}\mu_1 T_3^I(x_0, x_1-1, x_3, x_4) \\ &+ \frac{\mu_3(x_4+1)}{\mu_4} \\ &+ x_3\mu_3 T_3^I(x_0, x_1, x_3-1, x_4+1) \\ &+ I_{X_4}\mu_4 T_3^I(x_0, x_1, x_3, x_4-1)\Big], \;\; \boldsymbol{x} \in D, \end{aligned} \tag{4.6}$$

where $C_1 = \{(x_0-1, x_1, x_3+1, x_4) \in D_1^I\}$.

4.2.2 Augmented Network

The augmented network (Figure 4.2) has a path $0-1-4-5$ in addition to the paths that the initial network has. Therefore, there exist two decision-making points (nodes 0

Table 4.1: State-transitions for T_3^I from $\boldsymbol{x} = (x_0, x_1, x_3, x_4)$ to another state.

state transition	condition	transition rate	next state	remaining time
Next arriving job is routed through node 1	$x_0 > 0$ $(x_0 - 1, x_1, x_3 + 1, x_4) \in D_1^I$	λ	$(x_0 - 1, x_1 + 1, x_3, x_4)$	$T_3^I(x_0 - 1, x_1 + 1, x_3, x_4)$
Next arriving job is routed through node 3	$x_0 > 0$ $(x_0 - 1, x_1, x_3 + 1, x_4) \in D_3^I$	λ	$(x_0 - 1, x_1, x_3 + 1, x_4)$	$T_3^I(x_0 - 1, x_1, x_3 + 1, x_4)$
Node 1 serves a job	$x_1 > 0$	μ_1	$(x_0, x_1 - 1, x_3, x_4)$	$T_3^I(x_0, x_1 - 1, x_3, x_4)$
Node 3 serves the marked job	(none)	μ_3	$(x_0, x_1, x_3 - 1, x_4 + 1)$	$(x_4 + 1)/\mu_4$
Node 3 serves a job (not the marked job)	$x_3 > 0$	$x_3\mu_3$	$(x_0, x_1, x_3 - 1, x_4 + 1)$	$T_3^I(x_0, x_1, x_3 - 1, x_4 + 1)$
Node 4 serves a job	$x_4 > 0$	μ_4	$(x_0, x_1, x_3, x_4 - 1)$	$T_3^I(x_0, x_1, x_3, x_4 - 1)$

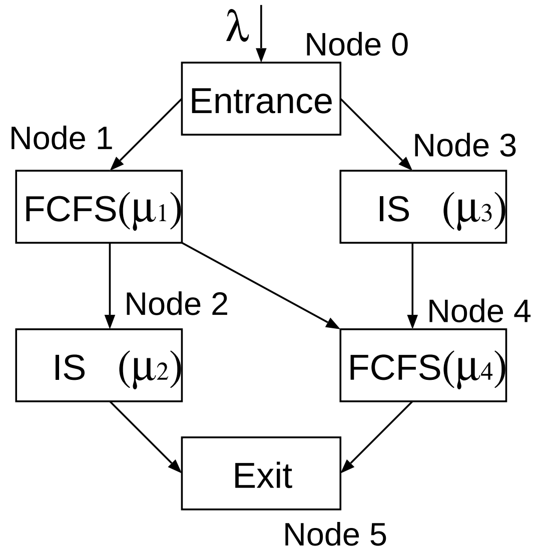

Figure 4.2: Augmented Network: The network has three paths (0-1-2-5, 0-3-4-5 and 0-1-4-5).

and 1) in the augmented network. At node 0, a job chooses a path by using dynamic routing decision. If the job is routed through node 1, then it needs to choose between the two paths (nodes 2 or 4). Note that if a job at node 1 sees that the current state is (x_0, x_1, x_3, x_4), then the expected delays at nodes 2 and 4 are $1/\mu_2$ and $(x_4+1)/\mu_4$, respectively. Therefore, if $1/\mu_2 \leq (x_4+1)/\mu_4$, then the job is routed through node 2. Otherwise, it is routed through node 4.

Let $T_1^A(\boldsymbol{x})$ and $T_3^A(\boldsymbol{x})$ denote the expected transit times of jobs routed through nodes 1 and 3, respectively, when the job sees at node 0 that the system state is $\boldsymbol{x}$ in the augmented network. Then, similarly to the initial network, we define two subsets of D, D_1^A and D_3^A as follows:

$$D_1^A = \{\boldsymbol{x} |\ \boldsymbol{x} \in D \text{ and } T_1^A(\boldsymbol{x}) \leq T_3^A(\boldsymbol{x})\}, \tag{4.7}$$

and

$$D_3^A = \{\boldsymbol{x} |\ \boldsymbol{x} \in D \text{ and } T_1^A(\boldsymbol{x}) > T_3^A(\boldsymbol{x})\}. \tag{4.8}$$

Note also that $D_1^A \cap D_3^A = \phi$ and $D_1^A \cup D_3^A = D$.

As to $T_1^A(\boldsymbol{x})$, we note that the expected transit time of a job at node 1 is affected by the number of jobs waiting behind. We denote the number of jobs behind the job at node 1 by x_1', the state vector including x_1' by $\boldsymbol{x}' = (x_0, x_1, x_1', x_3, x_4)$, and the set D' of $\boldsymbol{x}'$ by

$$D' = \{\boldsymbol{x}' \mid \boldsymbol{x}' \in \mathbb{N}^5 \text{ and } x_0 + x_1 + x_1' + x_3 + x_4 \leq N\}. \tag{4.9}$$

Suppose that a marked job sees at node 1 (that is, routed through node 1) that the system state is $\boldsymbol{x}'$. Then, the network has the state-transitions depicted in Tables 4.2 and 4.3. Note that the interevent time is exponentially distributed with mean $1/(I_{X_0}\lambda + \mu_1 + x_3\mu_3 + I_{X_4}\mu_4)$. Then the expected transit time of the marked job, $T_1'^A(\boldsymbol{x}')$, is given by the following equation:

$$\begin{aligned} T_1'^A(\boldsymbol{x}') &= \frac{1}{I_{X_0}\lambda + \mu_1 + x_3\mu_3 + I_{X_4>0}\mu_4}\Big[1 \\ &+ I_{X_0}\lambda T_1'^A(x_0 - 1, x_1, x_1' + I_{C_3}, x_3 + 1 - I_{C_3}, x_4) \\ &+ \mu_1\big(I_{X_1}T_1'^A(x_0, x_1 - 1, x_1', x_3, x_4 + I_{C_2}) \\ &+ I_{x_1=0}\min(\frac{1}{\mu_2}, \frac{x_4+1}{\mu_4})\big) \\ &+ x_3\mu_3 T_1'^A(x_0, x_1, x_1', x_3 - 1, x_4 + 1) \\ &+ I_{X_4}\mu_4 T_1'^A(x_0, x_1, x_1', x_3, x_4 - 1)\Big], \ \boldsymbol{x}' \in D' \end{aligned} \tag{4.10}$$

where $C_2 = \{1/\mu_2 > (x_4 + 1)/\mu_4\}$ and $C_3 = \{(x_0 - 1, x_1 + 1, x_3, x_4) \in D_1\}$. Note that $T_1^A(x_0, x_1, x_3, x_4) = T_1'^A(x_0, x_1, 0, x_3, x_4)$.

As to T_3^A, the network has the state-transitions as shown in Table 4.4. Therefore, similarly to the initial network, $T_3^A(\boldsymbol{x})$ is given by the following equation:

$$\begin{aligned} T_3^A(\boldsymbol{x}) &= \frac{1}{I_{X_0}\lambda + \mu_1 I_{X_1} + (x_3+1)\mu_3 + I_{X_4}\mu_4}\Big[1 \\ &+ \frac{\mu_3(x_4+1)}{\mu_4} \\ &+ I_{X_0}\lambda T_3^A(x_0-1, x_1+I_{C_1}, x_3+1-I_{C_1}, x_4) \\ &+ I_{X_1}\mu_1 T_3^A(x_0, x_1-I_{x_1>0}, x_3, x_4+I_{C_2}) \\ &+ x_3\mu_3 T_3^A(x_0, x_1, x_3-1, x_4+1) \\ &+ I_{X_4}\mu_4 T_3^A(x_0, x_1, x_3, x_4-1)\Big],\ \boldsymbol{x} \in D. \end{aligned} \tag{4.11}$$

4.3 Simulation Experiments

4.3.1 The Method

In this section, we describe our simulation method. Note that it is different from the simulation method of Calvert et al. [27].

We programmed the simulator using Microsoft Visual Studio 2003 (C++ Language), and used Mersenne Twister [156] as a pseudo-random number generator. We ran simulator on several personal computers, each of which has an AMD Athlon64 FX-51 (2.2GHz) CPU, and 2GB memory.

We can obtain dynamic routing decisions in the initial (D_1^I, D_3^I) and augmented (D_1^A, D_3^A) networks by solving (4.4) and (4.6) for $\boldsymbol{x} \in D$ (in the augmented network, (4.10) for $\boldsymbol{x}' \in D'$ and (4.11) for $\boldsymbol{x} \in D$) recursively, and comparing between $T_1^I(\boldsymbol{x})$ and $T_3^I(\boldsymbol{x})$ (in the augmented network, $T_1^A(\boldsymbol{x})$ and $T_3^A(\boldsymbol{x})$) for $\boldsymbol{x} \in D$, respectively.

We perform the following simulation in both the initial and augmented networks. First, we set the initial state of the network.

$$(x_0(0), x_1(0), x_2(0), x_3(0), x_4(0)) = (N, 0, 0, 0, 0),$$

Table 4.2: State-transitions for T_1^A from $\boldsymbol{x}' = (x_0, x_1, x_1', x_3, x_4)$ to another state (1).

state transition	condition	transition rate	next state	remaining time
Next arriving job is routed through node 1	$x_0 > 0$ $(x_0 - 1, x_1 + 1, x_1', x_3, x_4) \in D_1^A$	λ	$(x_0 - 1, x_1, x_1' + 1, x_3, x_4)$	$T_1'^A(x_0 - 1, x_1, x_1' + 1, x_3, x_4)$
Next arriving job is routed through node 3	$x_0 > 0$ $(x_0 - 1, x_1 + 1, x_1', x_3, x_4) \in D_3^A$	λ	$(x_0 - 1, x_1, x_1', x_3 + 1, x_4)$	$T_1'^A(x_0 - 1, x_1, x_1', x_3 + 1, x_4)$
The marked job served at node 1 is routed through node 2	$x_1 = 0$ $1/\mu_2 \leq (x_4 + 1)/\mu_4$	μ_1	$(x_0, x_1 - 1, x_1', x_3, x_4)$	μ_2
The marked job served at node 1 is routed through node 4	$x_1 = 0$ $1/\mu_2 > (x_4 + 1)/\mu_4$	μ_1	$(x_0, x_1 - 1, x_1', x_3, x_4 + 1)$	$(x_4 + 1)/\mu_4$

Table 4.3: State-transitions for T_1^A from $\boldsymbol{x}' = (x_0, x_1, x_1', x_3, x_4)$ to another state (2).

state transition	condition	transition rate	next state	remaining time
A job (not the marked job) served at node 1 is routed through node 2	$x_1 > 0$ $1/\mu_2 \leq (x_4+1)/\mu_4$	μ_1	$(x_0, x_1-1, x_1', x_3, x_4)$	$T_1'^A(x_0, x_1-1, x_1', x_3, x_4)$
A job (not the marked job) served at node 1 is routed through node 4	$x_1 > 0$ $1/\mu_2 > (x_4+1)/\mu_4$	μ_1	$(x_0, x_1-1, x_1', x_3, x_4+1)$	$T_1'^A(x_0, x_1-1, x_1', x_3, x_4+1)$
Node 3 serves a job	$x_3 > 0$	$x_3\mu_3$	$(x_0, x_1, x_1', x_3-1, x_4+1)$	$T_1'^A(x_0, x_1, x_1', x_3-1, x_4+1)$
Node 4 serves a job	$x_4 > 0$	μ_4	$(x_0, x_1, x_1', x_3, x_4-1)$	$T_1'^A(x_0, x_1, x_1', x_3, x_4-1)$

Table 4.4: State-transitions for T_3^A from $\boldsymbol{x} = (x_0, x_1, x_3, x_4)$ to another state.

state transition	condition	transition rate	next state	remaining time
Next arriving job is routed through node 1	$x_0 > 0$ $(x_0 - 1, x_1, x_3 + 1, x_4) \in D_1^A$	λ	$(x_0 - 1, x_1 + 1, x_3, x_4)$	$T_3^A(x_0 - 1, x_1 + 1, x_3, x_4)$
Next arriving job is routed through node 3	$x_0 > 0$ $(x_0 - 1, x_1, x_3 + 1, x_4) \in D_3^A$	λ	$(x_0 - 1, x_1, x_3 + 1, x_4)$	$T_3^A(x_0 - 1, x_1, x_3 + 1, x_4)$
A job served at node 1 is routed through node 2	$x_1 > 0$ $1/\mu_2 \leq (x_4 + 1)/\mu_4$	μ_1	$(x_0, x_1 - 1, x_3, x_4)$	$T_3^A(x_0, x_1 - 1, x_3, x_4)$
A job served at node 1 is routed through node 4	$x_1 > 0$ $1/\mu_2 > (x_4 + 1)/\mu_4$	μ_1	$(x_0, x_1 - 1, x_3, x_4 + 1)$	$T_3^A(x_0, x_1 - 1, x_3, x_4 + 1)$
Node 3 serves the marked job	(none)	μ_3	$(x_0, x_1, x_3 - 1, x_4 + 1)$	$(x_4 + 1)/\mu_4$
Node 3 serves a job (not the marked job)	$x_3 > 0$	$x_3\mu_3$	$(x_0, x_1, x_3 - 1, x_4 + 1)$	$T_3^A(x_0, x_1, x_3 - 1, x_4 + 1)$
Node 4 serves a job	$x_4 > 0$	μ_4	$(x_0, x_1, x_3, x_4 - 1)$	$T_3^A(x_0, x_1, x_3, x_4 - 1)$

at time 0. We start each simulation at the initial state, and the simulation continues until the state becomes $(0,0,0,0,0)$. We regard it as one simulation cycle. In all the results shown in next section, we set the parameter value as follows: $N = 100$. We then discard the first 20 jobs since initial states are not always steady-states. We obtain the mean value of transit times of the remaining 80 jobs. We then compute the expectation of the mean transit times obtained by 50000 simulation cycles.

We repeat the above procedure 20 times using random numbers of different seeds, and calculate the 95% confidence interval for each value of arrival rate λ. Note, however, that the lengths of the confidence intervals are so short that the errorbars in the graphs are invisible. Also, note that the capacity of the initial and augmented networks is $(\mu_1 + \mu_4)$.

Because of the recursive nature of the equations giving the expected transit time, the size of memory needed for the computation increases rapidly in the total number of jobs. More precisely, from (10), for a given N, the program stores in memory the different values of T_1^A corresponding to the different cases of $\boldsymbol{x}'$, which is a subset of $\mathbb{N}^5$. We therefore limit the parameter value N to be 100 in our simulations.

4.3.2 The Results

An example in which the paradox occurs: Figure 4.3 illustrates the mean transit times of jobs routed through the two paths, 0-1-2-5 and 0-3-4-5 in the initial network, and three paths, 0-1-2-5, 0-3-4-5 and 0-1-4-5 in the augmented network with dynamic routing for each value of arrival rate λ where the parameter values are given as follows: $\mu_1 = \mu_4 = 5.0$, and $\mu_2 = \mu_3 = 0.5$. Note that these networks are symmetric. For static routing, the worst-case paradox has been shown to occur in a case where both the initial and augmented networks are symmetric [84].

We say that "the Braess paradox occurs" if the mean transit time of the jobs routed through *any* path in the augmented network is larger than the mean transit time of the jobs routed through *any* path in the initial network. In Figure 4.3 we observe that the

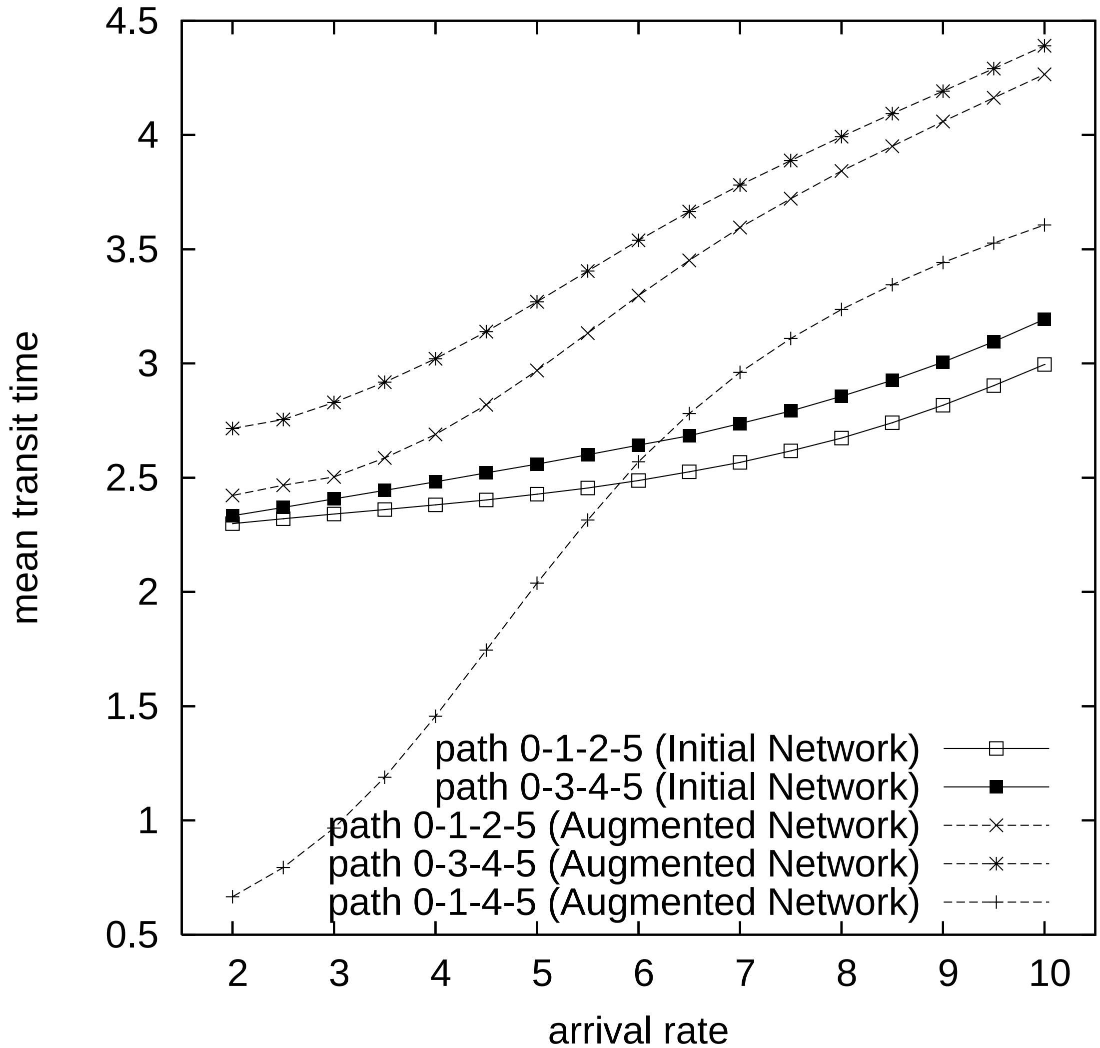

Figure 4.3: Mean transit time of jobs routed through each path in the initial and augmented networks with dynamic routing for each value of arrival rate λ ($\mu_1 = \mu_4 = 5.0$, and $\mu_2 = \mu_3 = 0.5$).

Braess paradox occurs in the case where the range of the value of the arrival rate is $6.5 \leq \lambda \leq 10$.

Note that if we consider static routing in the initial and augmented networks described in [36], the mean transit time of jobs routed through any path may have the same value. On the other hand, in Figure 4.3, we observe that the mean transit time of jobs routed through 0-1-4-5 is different from the mean transit times of jobs routed through 0-1-2-5 and 0-3-4-5 in the augmented network. Therefore, the behavior of individual users in dynamic routing may not be similar to that of individual users in static routing.

Discussion of a result by Calvert et al. [27]: Figure 4.4 illustrates the overall mean transit times of jobs in the initial and augmented networks with dynamic routing for each value of arrival rate λ where the parameter values are given as follows: $\mu_1 = \mu_4 = 2.5$, and $\mu_2 = \mu_3 = 0.5$.

Note that the values of λ and μ_i are the same as the values in [27], and the simulation method is different from in [27]. When λ is larger than 2.5, the overall mean transit time of jobs in the augmented network is larger than the overall mean transit time of jobs in the initial network. This behavior is very similar to the result shown in [27]. Calvert et al. [27] said "Braess paradox appears for λ greater than the crossover value, i.e. about $\lambda = 2.65$." Figure 4.5 shows that the mean transit time of jobs routed through each path in the initial and augmented networks with dynamic routing. The parameter values are same as those of Figure 4.4.

In Figure 4.5, we observe that the mean transit time of jobs routed through the path 0-1-4-5 in the augmented network is smaller than the mean transit time of jobs routed through the path 0-3-4-5 in the initial network. We think that the Braess paradox occurs when by "adding capacity (a link) to a network degrades the costs for *all* users." Therefore, Figure 4.4 does not show any paradox in our sense.

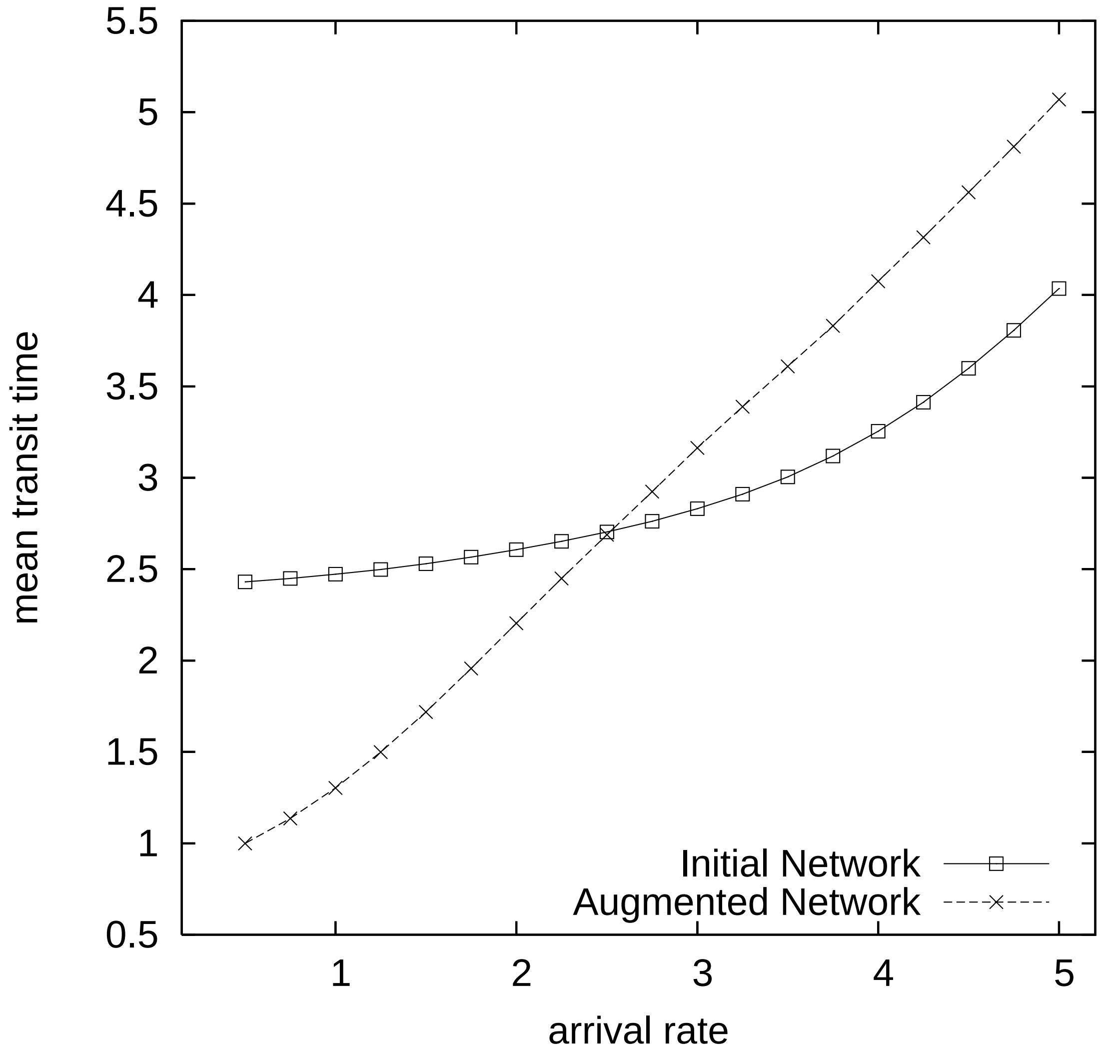

Figure 4.4: Overall mean transit time of jobs routed through all paths in the initial and augmented networks with dynamic routing for each value of arrival rate λ ($\mu_1 = \mu_4 = 2.5$, and $\mu_2 = \mu_3 = 0.5$).

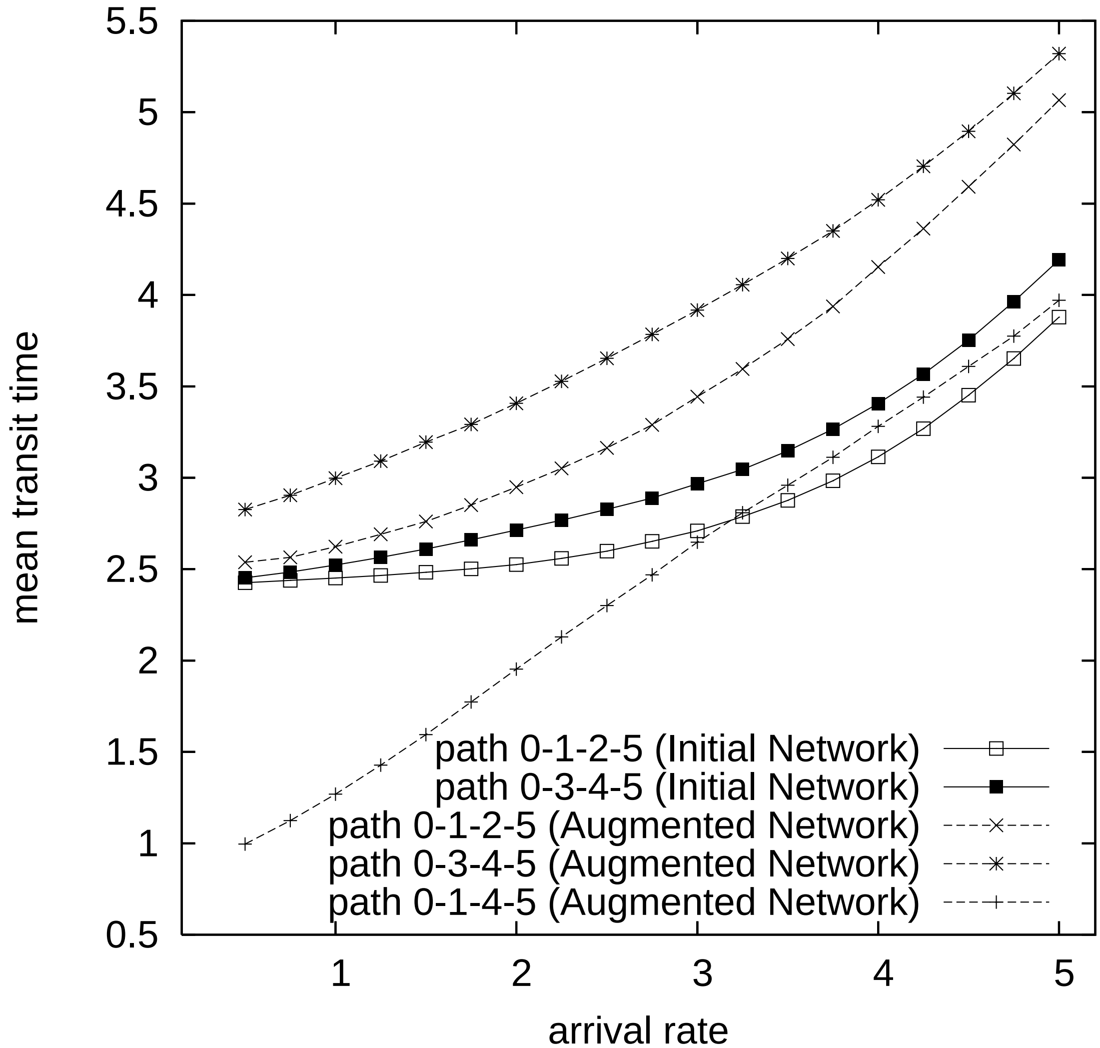

Figure 4.5: Mean transit time of jobs routed through each path in the initial and augmented networks with dynamic routing for each value of arrival rate λ ($\mu_1 = \mu_4 = 2.5$, and $\mu_2 = \mu_3 = 0.5$).

4.4 Conclusion

In this chapter, we have studied the existence of the Braess paradox in dynamic routing for the Cohen-Kelly network. Based on the analytical study by Calvert et al., we have obtained the dynamic routing decisions in the initial and augmented networks. We have compared the performance of the initial and augmented networks by simulation, and have found a paradoxical phenomenon similar to the one that Cohen-Kelly showed under static routing. We have touched on the results of Calvert et al.

Chapter 5

Case Study of Pareto Set, Fairness, and Nash Equilibrium in Load Balancing

5.1 Introduction

There exist many systems where multiple independent users, or players, may strive to optimize their own utility or cost unilaterally, which can be regarded as noncooperative games. The situation where each user attains its own optimum coincidently is a Nash equilibrium. Nash equilibria may, however, be Pareto inefficient. In particular, we call a situation of a system *strongly Pareto inefficient* if all users have more benefits in another situation than the considered situation. As for the communication and transportation networks, examples of such strong Pareto inefficiency have been shown with respect to noncooperative routing, first by Braess [26], and a number of related studies followed [35, 36, 56, 96, 113, 116, 163, 182]. As for the non-cooperative load balancing in distributed computer systems, the existence of paradoxes that appear only in the case of a finite number of players but not in the case of infinitesimal players has been shown [87, 92]. Note that load balancing and routing have mutually similar logical structures [8, 91, 139, 195].

On the other hand, there can exist innumerably many Pareto-optimal situations. The choice of one to achieve can be controversial among users. One selection criterion is fairness among users. Various fairness concepts that achieve Pareto optima but are not directly related to Nash equilibria have been already proposed [21, 97, 98, 159, 162].

In contrast, each Nash equilibrium is fair on all users in the sense that it is achieved by the fair competition (with no coalition) among users. Then, among the Pareto-optima, only those that are strongly Pareto-superior to the Nash equilibrium could satisfy all users. In particular, as the situations that would make all users to feel fairness similar to that of the Nash equilibrium, we consider a group of situations where each user's utility is proportionately larger than that of the Nash equilibrium. We say that such situations are *Nash proportionately fair* to the Nash equilibrium. If we identify a Nash-proportionately-fair Pareto optimum, the resulting situation will satisfy all users since it reflects the competitive fairness given by the Nash equilibrium and is Pareto optimal, at the same time.

By the Pareto set of a system, we mean the set of all Pareto optima of the system. We are quite interested in the positions that the already proposed and Nash proportionate fairness objectives occupy in the Pareto set. Since it may seem difficult to study this problem in a general framework from this beginning stage, in this chapter, we use a simple model of load balancing in distributed computer systems as the platform of the present research. We numerically obtain the cost (the mean response time) of each user at the points in the Pareto set, at the solutions that achieve various fairness objectives, and at the Nash equilibrium (which is unique in this case [10]).

This chapter characterizes these fairness objectives through numerical results in simple static load balancing model with two identical servers (computers) each of which has an identical arrival and its own queue. In numerical results, we compare the fairness objectives. For example, we observe that the points that achieve the general parameterized fairness objectives generally cover a part of the Pareto set, and at times, do not cover the

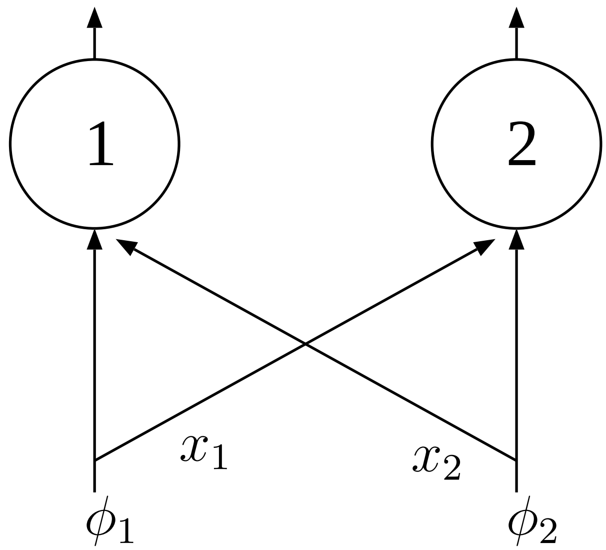

Figure 5.1: Load balancing in a distributed system consisting of two servers

Nash-proportionate-fair Pareto optimal points.

The rest of this chapter is organized as follows. Section 5.2 describes our model and formulates as various types of fair and optimal load balancing problems. Section 5.3 shows some numerical results. Section 5.4 concludes this article.

5.2 The Model and Assumptions

We consider a distributed computer system consisting of two servers (computers), numbered 1 and 2, with two flows of demands ϕ_1 and ϕ_2 arriving from users 1 and 2 at servers 1 and 2, respectively, as shown in Figure 5.1. This model is similar to the system studied in Kameda et al. [87]. Let a fraction x_i $(0 \leq x_i \leq \phi_i)$ of a flow of jobs be forwarded from server i to the other server j $(\neq i)$. Denote by $\boldsymbol{x}$ the vector (x_1, x_2). Denote further by β_1 and β_2, respectively, the resulting loads on nodes 1 and 2. Then,

$$\beta_i = \phi_i - x_i + x_j, \;\; i, j = 1, 2 \; (i \neq j).$$

We assume that node i has an exponential server with mean $1/\mu_i$, $i = 1, 2$. Then, the expected delay at server i under the load of rate β_i is given by $(\mu_i - \beta_i)^{-1}$. For simplicity, we assume that forwarding a job requires a fixed delay t. Therefore, the cost of user i, that is, the delay of each flow arriving from the user i, can be written:

$$T_i(\boldsymbol{x}) = \frac{1}{\phi_i}\left[\frac{\phi_i - x_i}{\mu_i - \phi_i + x_i - x_j} + x_i(t + \frac{1}{\mu_j - \phi_j + x_j - x_i})\right], \tag{5.1}$$

for $i, j = 1, 2 (j \neq i)$.

We denote by $\boldsymbol{C}$ the feasible region of $\boldsymbol{x}$. Note that the forwarding rate $x_i,\ i = 1, 2$ is non-negative, bounded by arrival flow, and the load β_i is positive. Then, we have

$$\boldsymbol{C} = \{\boldsymbol{x} \mid 0 \leq x_i \leq \phi_i,\ i = 1, 2 \ \text{ and } \ \mu_i - \phi_i + x_i - x_j > 0,\ i, j = 1, 2\ (i \neq j)\}. \tag{5.2}$$

Clearly, $\boldsymbol{C}$ is a convex set. Also, we call the set of points $(T_1(\boldsymbol{x}), T_2(\boldsymbol{x}))$ for $\boldsymbol{x} \in \boldsymbol{C}$ *achievable set*.

On the other hand, $T_i,\ i = 1, 2$ satisfies the following proposition [199]:

Proposition 5.1. *The function $T_i,\ i = 1, 2$ is non-convex in $\boldsymbol{x} \in \boldsymbol{C}$.*

5.2.1 Pareto set and weighted-sum optimization

Denote by $\boldsymbol{\Pi}$ the Pareto set defined as follows:

$$\boldsymbol{\Pi} = \left\{ (T_1(\boldsymbol{x}), T_2(\boldsymbol{x})) \;\middle|\; \begin{array}{l} \boldsymbol{x} \in \boldsymbol{C}, \forall \boldsymbol{x}' \in \boldsymbol{C}, \exists i = 1, 2, T_i(\boldsymbol{x}') < T_i(\boldsymbol{x}) \\ \Rightarrow \exists j \neq i, T_j(\boldsymbol{x}') > T_j(\boldsymbol{x}) \end{array} \right\}. \tag{5.3}$$

The Pareto set $\boldsymbol{\Pi}$ is the lower left border in the achievable set. We therefore also refer to it as *Pareto border*.

There may exist a number of Pareto optima in a distributed computer system. Then, one of our interests is how to obtain all of them. Consider the following objective:

$$\Omega(\boldsymbol{x}) = \xi_1 U_1(\boldsymbol{x}) + \xi_2 U_2(\boldsymbol{x}) \tag{5.4}$$

where $\xi_i,\ i = 1, 2$ is the weighting factors that satisfy

$$\xi_i \geq 0,\ i = 1, 2, \text{ and, } \xi_1 + \xi_2 > 0.$$

Then, any given solution by maximization of (5.4) with respect to $\boldsymbol{x} \in \boldsymbol{C}$ is Pareto optimal. Also, under the assumption of convexity, the maximization problem has the following property [16]:

Proposition 5.2. *If the feasible region* $\boldsymbol{C}$ *is convex and the functions* $T_i,\ i = 1, 2$ *are convex in* $\boldsymbol{x} \in \boldsymbol{C}$*, then (5.4) gives all the Pareto solutions for the different combinations of the weighting factors* $\xi_i,\ i = 1, 2$.

Remark 5.1. *In fact* $T_i,\ i = 1, 2,$ *are not convex in* $\boldsymbol{x}$ *as noted in proposition 5.1. Therefore, there may exist Pareto optimal points that are not a solution of minimizing a weighted sum of costs as shown in the numerical examples given in Section 5.3.*

5.2.2 General Fairness Objective

Recently, in the context of congestion control, Mo and Warland [162] proposed a general and simple uniform description (parameterized by parameter α) of a wide family of fair criteria, including in particular proportional fairness and max-min fairness. With the general parameterized fairness objective of Mo and Warland [162], our load balancing problem is given as follows:

$$F_\alpha(\hat{\boldsymbol{x}}) = \min_{\boldsymbol{x} \in C} F_\alpha(\boldsymbol{x}), \tag{5.5}$$

where

$$F_\alpha(\boldsymbol{x}) = \begin{cases} \dfrac{1}{1-\alpha} \displaystyle\sum_{i=1}^{2} (\beta_i T_i(\boldsymbol{x}))^{1-\alpha}, & \text{if } \alpha \geq 0,\ \alpha \neq 1, \\ \displaystyle\sum_{i=1}^{2} \log T_i(\boldsymbol{x}), & \text{if } \alpha = 1. \end{cases} \tag{5.6}$$

where α is the fairness parameter and β_i (> 0) is the weighting factor associated to the utility of user i. Note that when $\alpha = 0$, Eq. (5.8) corresponds to the weighted sum of the costs of the users. When $\alpha = 1$ the criterion converges to a Nash bargaining solution or weighted proportional fair point. Finally, as α grows to infinity the solution converges to the one given by the max-min fair criterion (see Corollary 2 in [162]).

Similarly as (5.5), we consider maximization of the utilities of the users, and formulate the following fairness objectives:

$$\grave{F}_\alpha(\hat{\boldsymbol{x}}) = \max_{\boldsymbol{x}\in\boldsymbol{C}} \grave{F}_\alpha(\boldsymbol{x}) \tag{5.7}$$

where

$$\grave{F}_\alpha(\boldsymbol{x}) = \begin{cases} \dfrac{1}{1-\alpha}\displaystyle\sum_{i=1}^{2} (\beta_i U_i(\boldsymbol{x}))^{1-\alpha}, & \text{if } \alpha \geq 0,\ \alpha \neq 1, \\ \displaystyle\sum_{i=1}^{2} \log U_i(\boldsymbol{x}), & \text{if } \alpha = 1. \end{cases} \tag{5.8}$$

and $U_i(\boldsymbol{x}) = 1/T_i(\boldsymbol{x}),\ i = 1, 2.$

Obviously, (5.7) also follows the lines of already proposed general parameterized fairness objectives.

Remark 5.2. *Denote*

$$C_i(\boldsymbol{x}) = \frac{\{T_i(\boldsymbol{x})\}^{1-\alpha}}{1-\alpha}$$

for $\alpha \geq 0, \neq 1$ *while* $C_i(\boldsymbol{x}) = \log T_i(\boldsymbol{x})$ *for* $\alpha = 1$. *Then, we have*

$$C_i(\boldsymbol{x}) \gtreqless C_i(\boldsymbol{x}') \Leftrightarrow T_i(\boldsymbol{x}) \gtreqless T_i(\boldsymbol{x}'),\ \textit{for } i = 1, 2.$$

That is, a Pareto optimum for user cost C_i, $i = 1, 2$, *is also a Pareto optimum for user cost* T_i, $i = 1, 2$, *and vice versa.*

Similarly as (5.4) gives a Pareto optimum for user cost T_i, $i = 1, 2$, *(5.5) will give also a Pareto optimum for user cost* T_i, $i = 1, 2$. *The same arguments hold for (5.7), and the objective (5.7) will also give a Pareto optimum for user cost* T_i, $i = 1, 2$.

5.2.3 Nash proportionate fairness

In this system, a Nash equilibrium $\tilde{\boldsymbol{x}}$ is given as follows:

$$T_i(\tilde{x}) = \min_{x_i} T_i(x_i, \tilde{x}_j), \quad \text{s.t.} \ \ (x_i, \tilde{x}_j) \in \boldsymbol{C},\ i, j = 1, 2\ (i \neq j). \tag{5.9}$$

For the model in question, there exists a unique Nash equilibrium [10]. The Nash equilibrium, $T_i(\tilde{x})$, $i = 1, 2$ may, however, sometimes be Pareto inefficient [87]. Consider

$$P_i = \eta T_i(\tilde{x}), \text{ for } i = 1, 2. \tag{5.10}$$

By decreasing η (> 0), if (P_1, P_2) hits the Pareto border $\mathbf{\Pi}$, and reaches a Pareto optimal point, $(\bar{P}_1, \bar{P}_2)$, it is called *Nash-proportionate-fair* point. Note that a Nash equilibrium corresponds to exactly one Nash-proportionate-fair point if it exists.

5.3 Numerical Results

We characterize fair and optimal load balancing problem through some numerical results. For convenience, we add the constraint $\xi_1 + \xi_2 = 1$ in minimization of weighted-sums without losing generality. Also, we set the parameter value as follows: $\beta_1 = \beta_2 = 1$. We have paid special attention not to catch local optima that are not global optima since the functions to be optimized are not convex in general.

5.3.1 A case where weighted-sum objective does not cover all the Pareto optima

Figures 5.2-5.5 show the Nash equilibrium, the part of Pareto set obtained by the weighted-sum optimization and the fairness solutions that achieve (5.5) and (5.7). The values of the system parameters are $\phi_1 = 2.1,\ \mu_1 = 3,\ \phi_2 = 2.7,\ \mu_2 = 3.7,$ and $t = 0.001$.

In Figure 5.3, we observe that the part of the Pareto set obtained by the weighted-sum objectives are divided into two parts and do not cover all the Pareto set. We note that T_1 and T_2 are nonconvex in $\boldsymbol{x}$, and, therefore, the optimal solutions to the weighted-sum objectives may not cover the Pareto set since the conditions of the Aubin's theorem are not satisfied. Thus, the above result presents a counter example that shows that if a condition of the Aubin's theorem is not satisfied, the theorem does not hold.

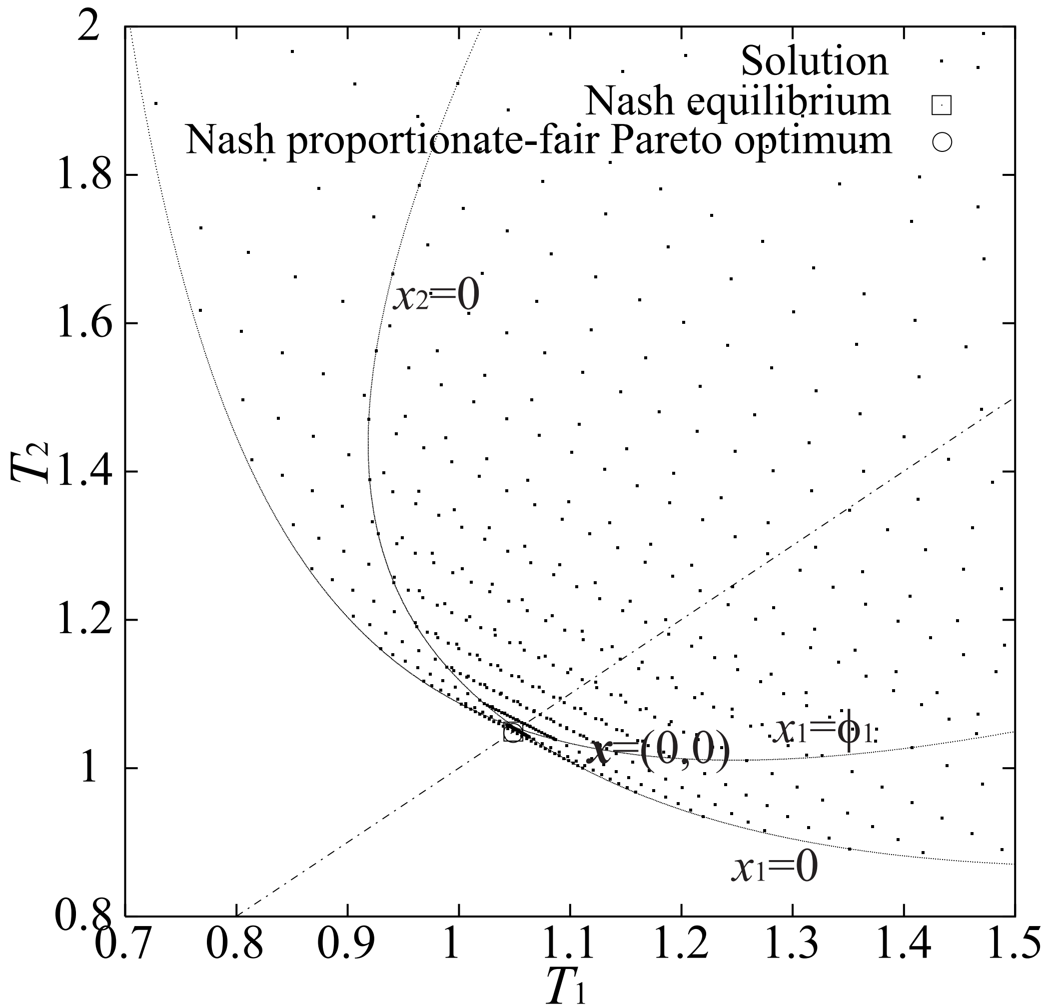

Figure 5.2: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Nash equilibrium (Nash proportionate fairness). The values of system parameters are $\phi_1 = 2.1,\ \mu_1 = 3,\ \phi_2 = 2.7,\ \mu_2 = 3.7,$ and $t = 0.001$.

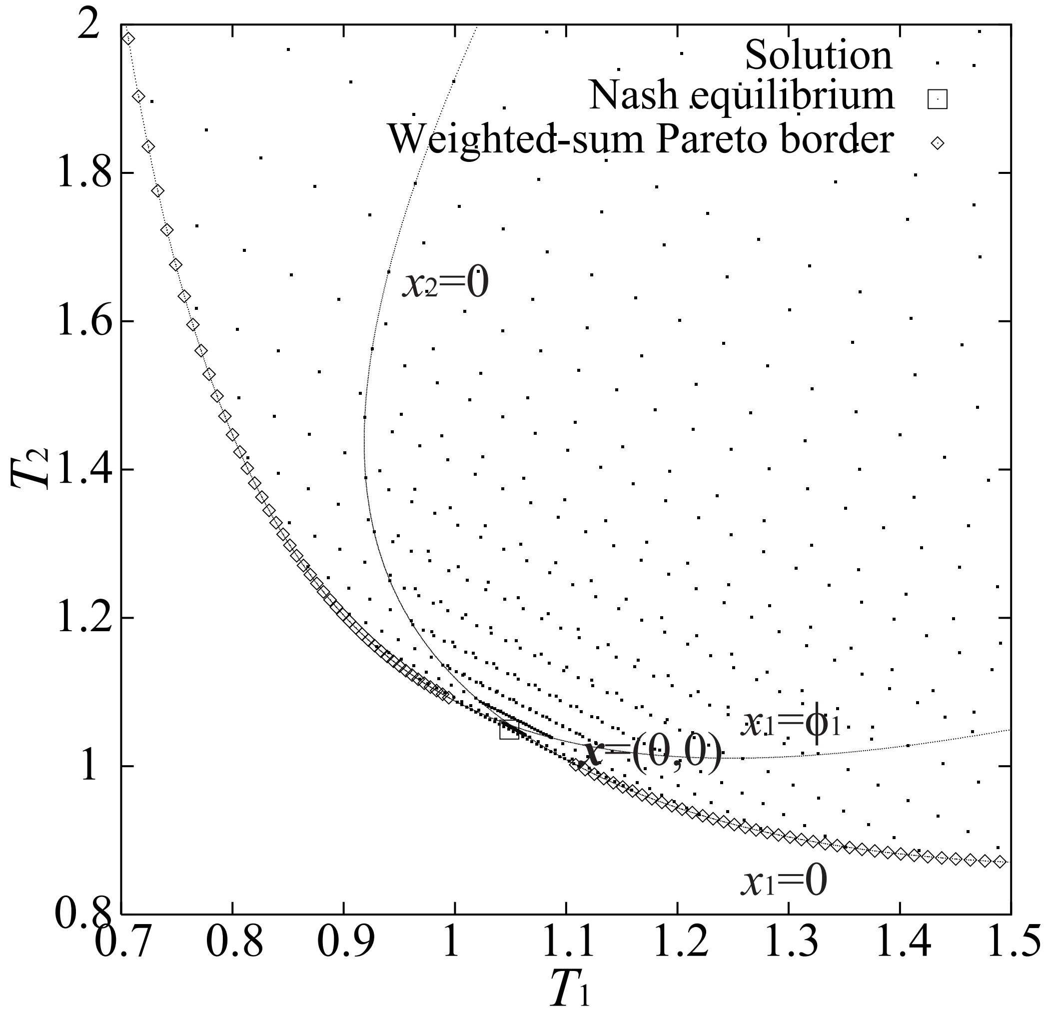

Figure 5.3: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and minimization of weighted sums of costs. The values of system parameters are $\phi_1 = 2.1,\ \mu_1 = 3,\ \phi_2 = 2.7,\ \mu_2 = 3.7,$ and $t = 0.001$.

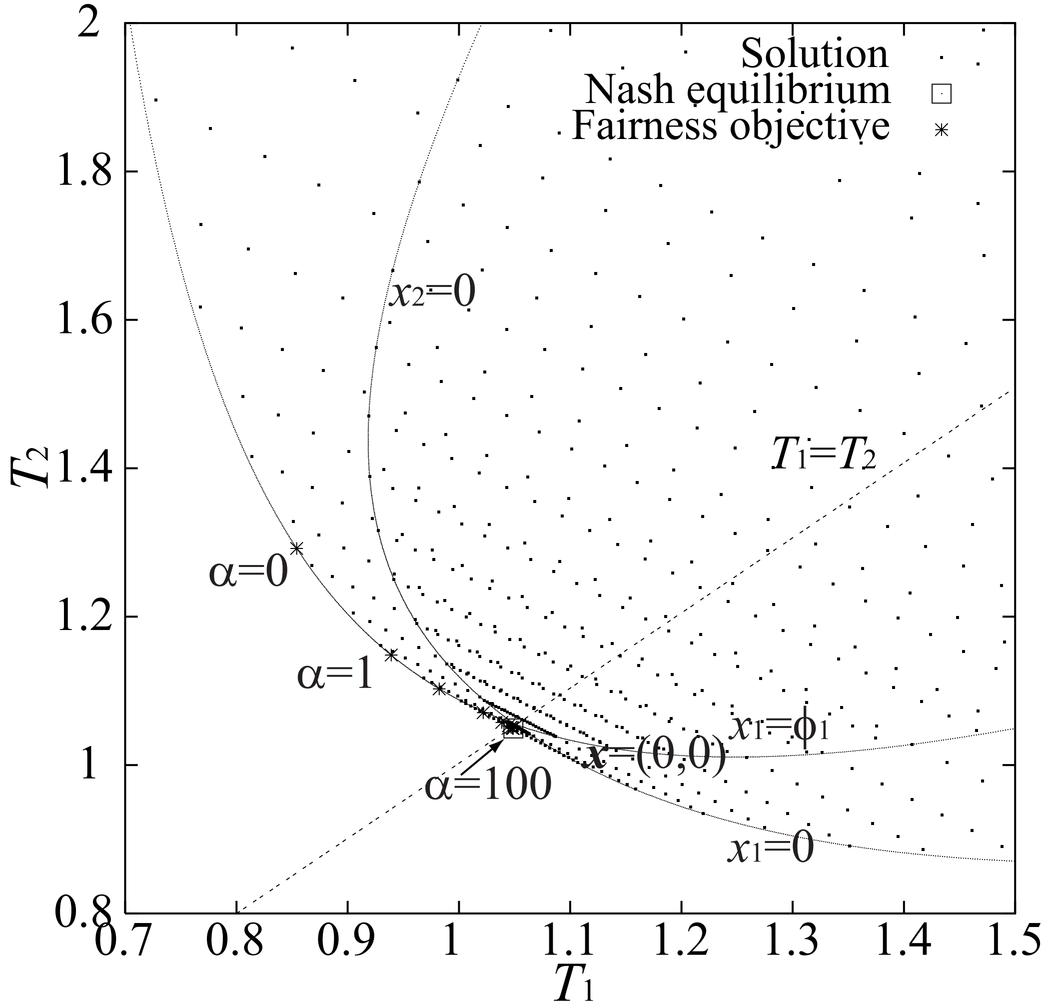

Figure 5.4: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.7). The values of system parameters are $\phi_1 = 2.1,\ \mu_1 = 3,\ \phi_2 = 2.7,\ \mu_2 = 3.7,$ and $t = 0.001$.

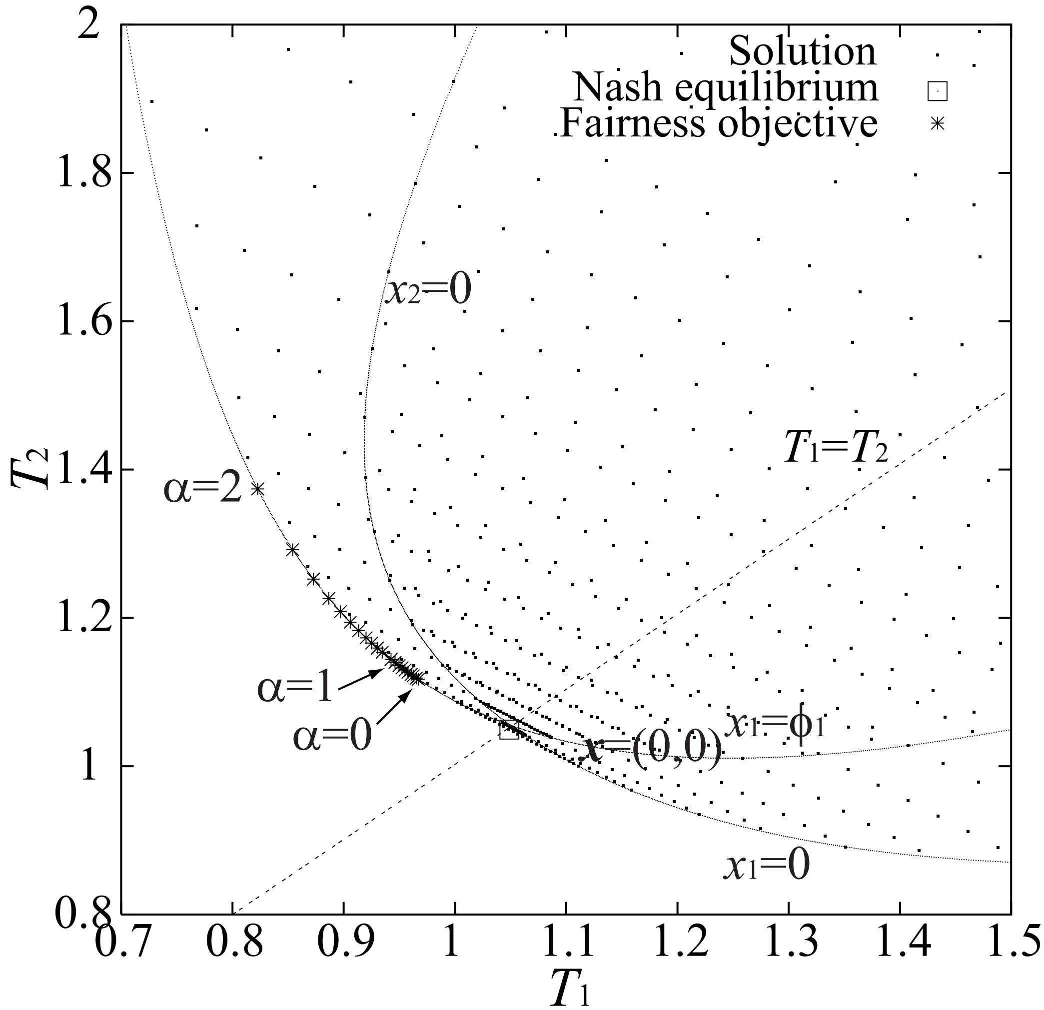

Figure 5.5: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.5). The values of system parameters are $\phi_1 = 2.1,\ \mu_1 = 3,\ \phi_2 = 2.7,\ \mu_2 = 3.7,$ and $t = 0.001$.

In Figures 5.4-5.5, we observe that, as to the solutions obtained by the already proposed fairness objectives (5.7), the optimal points converges to the point of the Pareto set satisfying $T_1 = T_2$ as the value of α increases. (Note, however, that, in this case, we were unable to obtain numerically the optimal values for very large values of α (> 100). This is perhaps because of accumulation of round-off errors.) In particular, some such optimal points are in the part of the Pareto set which the weighted-sum objective cannot cover. On the other hand, the points optimal for the objectives (5.5) diverge from the Max-Min fair point as the value of α increases. Superficially thinking, both the objectives (5.7) and (5.5) would be anticipated to show similar behaves, but, in fact, the objective (5.5) is not good as a general fairness objective.

It is seen in Figure 5.2 that the Nash equilibrium is almost Pareto optimal, and almost identical to the Nash proportionate-fair Pareto optimum. In this case, however, the Nash proportionate-fair Pareto-optimal point is not in the part of the Pareto set obtained by weighted-sum objectives and any fairness objectives.

5.3.2 A case where the Nash equilibrium is not Pareto optimal

Figures 5.6-5.9 show a case where the values of system parameters are $\phi_1 = 0.9,\ \mu_1 = 1.5,\ \phi_2 = 0.8,\ \mu_2 = 2,$ and $t = 0.35$.

In Figure 5.6, we observe that the Nash equilibrium is not on the Pareto set. The Pareto set and the straight line passing through the origin (0,0) and the Nash equilibrium intersect at a point, which is the Nash proportionate-fair Pareto-optimal point.

In this case, Pareto optimal points that achieve the weighted-sum optimization cover all the Pareto set. The Pareto optimum corresponding to the Nash proportionate fairness is given by $\xi_1 \simeq 0.934$ and $\xi_2 \simeq 0.066$. In this case, the Nash-proportionate-fair optimal point happens to be the point that achieves the fairness objective (5.7) for $\alpha \simeq 26.4$. On the other hand, in this case, no points that achieve the fairness objectives (5.5) with any values of α can be identical with it.

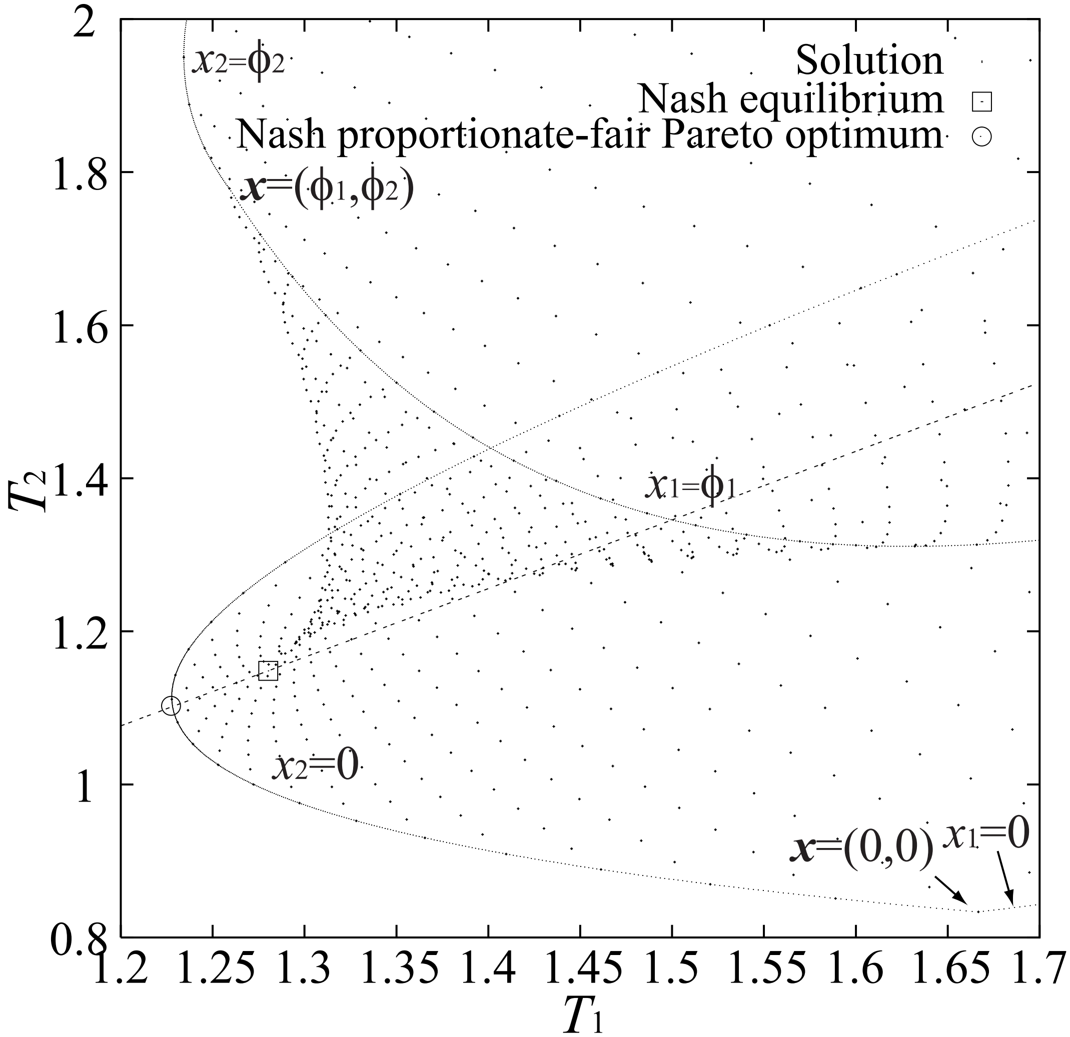

Figure 5.6: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Nash equilibrium (Nash proportionate fairness). The values of system parameters are $\phi_1 = 0.9,\ \mu_1 = 1.5,\ \phi_2 = 0.8,\ \mu_2 = 2,$ and $t = 0.35$.

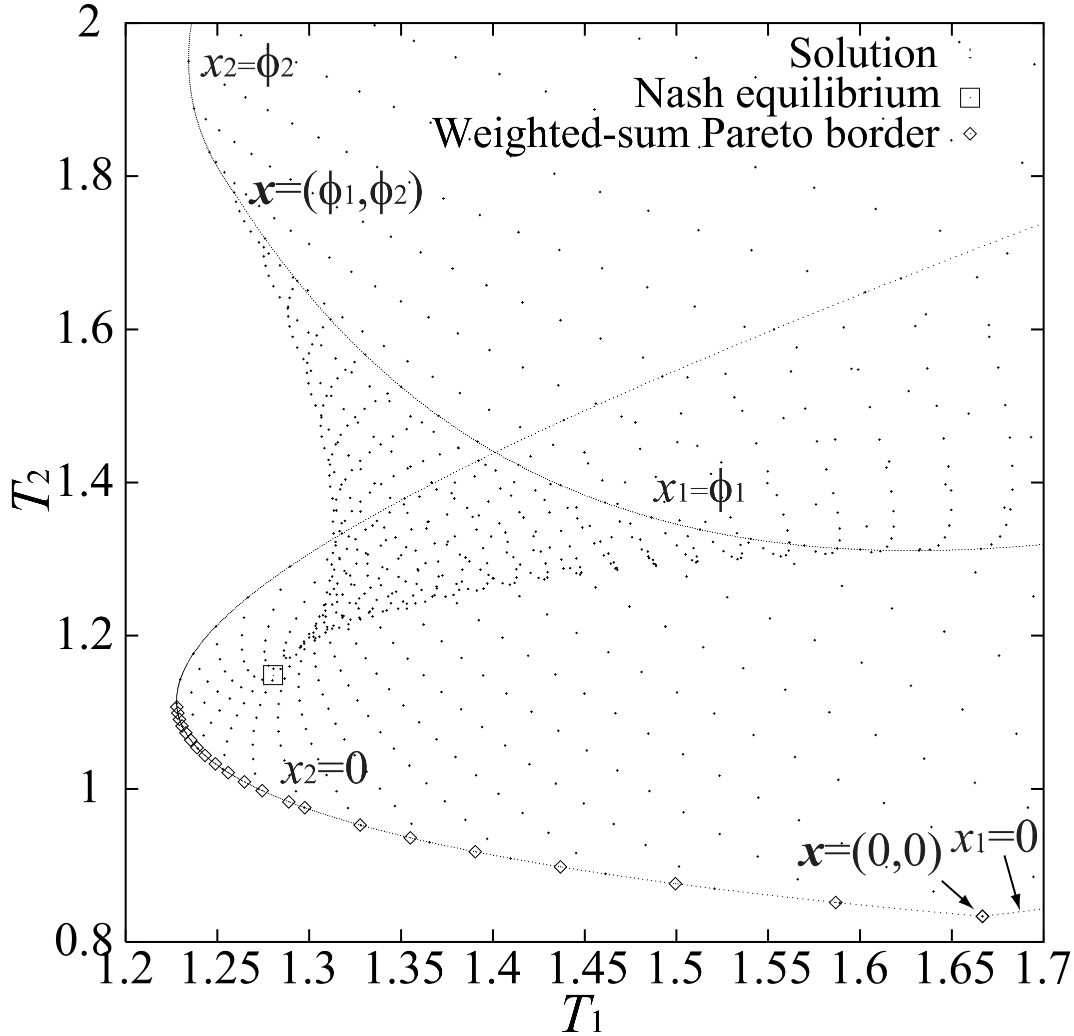

Figure 5.7: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and minimization of weighted sums of costs. The values of system parameters are $\phi_1 = 0.9,\ \mu_1 = 1.5,\ \phi_2 = 0.8,\ \mu_2 = 2,$ and $t = 0.35$.

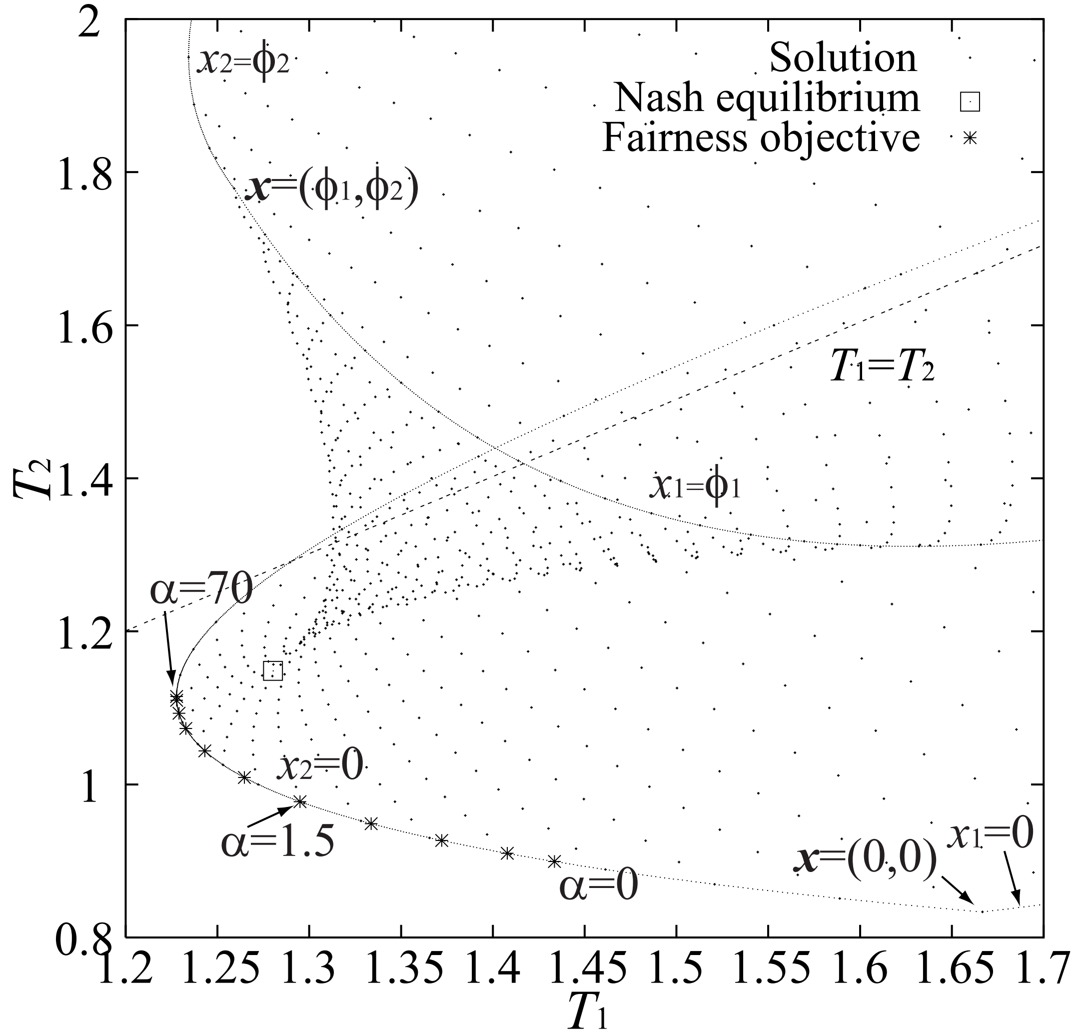

Figure 5.8: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.7). The values of system parameters are $\phi_1 = 0.9,\ \mu_1 = 1.5,\ \phi_2 = 0.8,\ \mu_2 = 2,$ and $t = 0.35$.

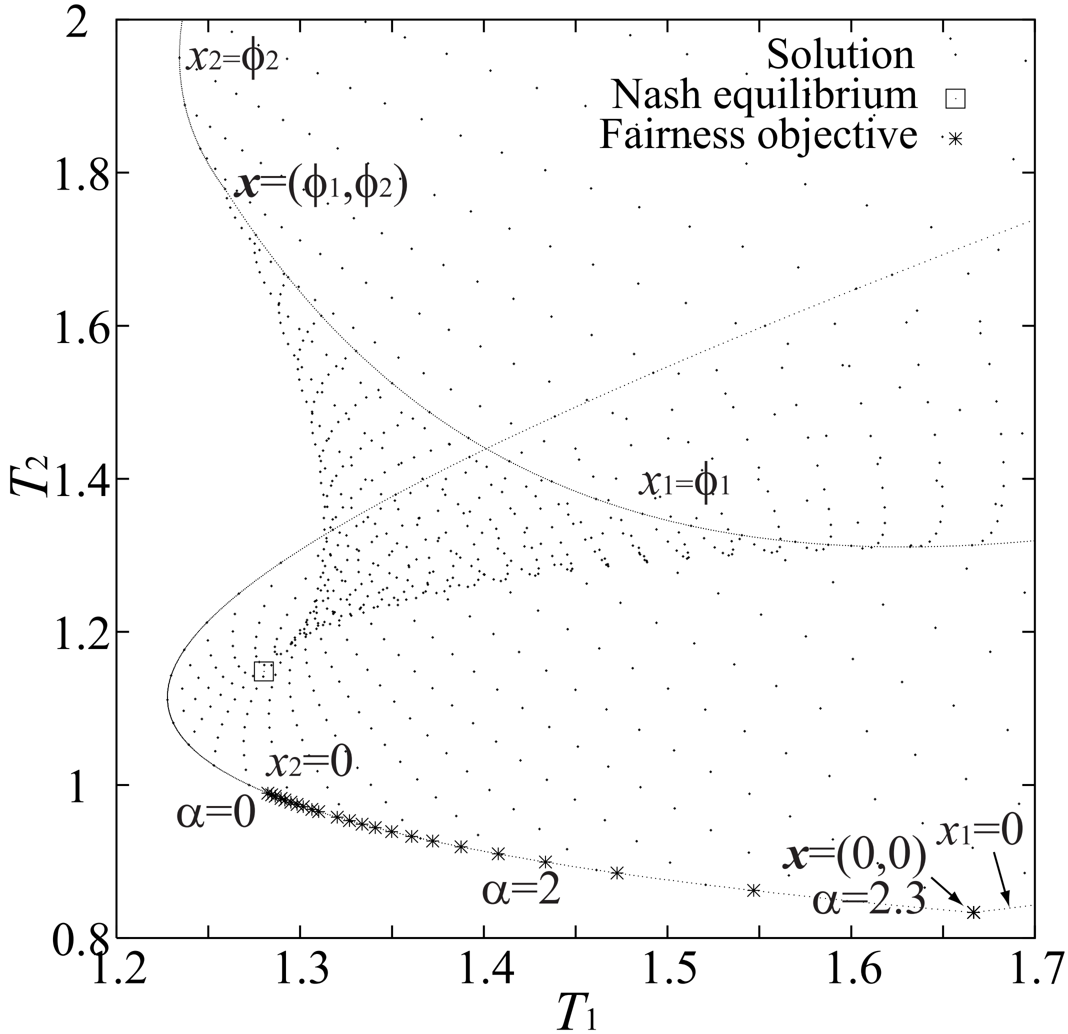

Figure 5.9: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.5). The values of system parameters are $\phi_1 = 0.9,\ \mu_1 = 1.5,\ \phi_2 = 0.8,\ \mu_2 = 2,$ and $t = 0.35$.

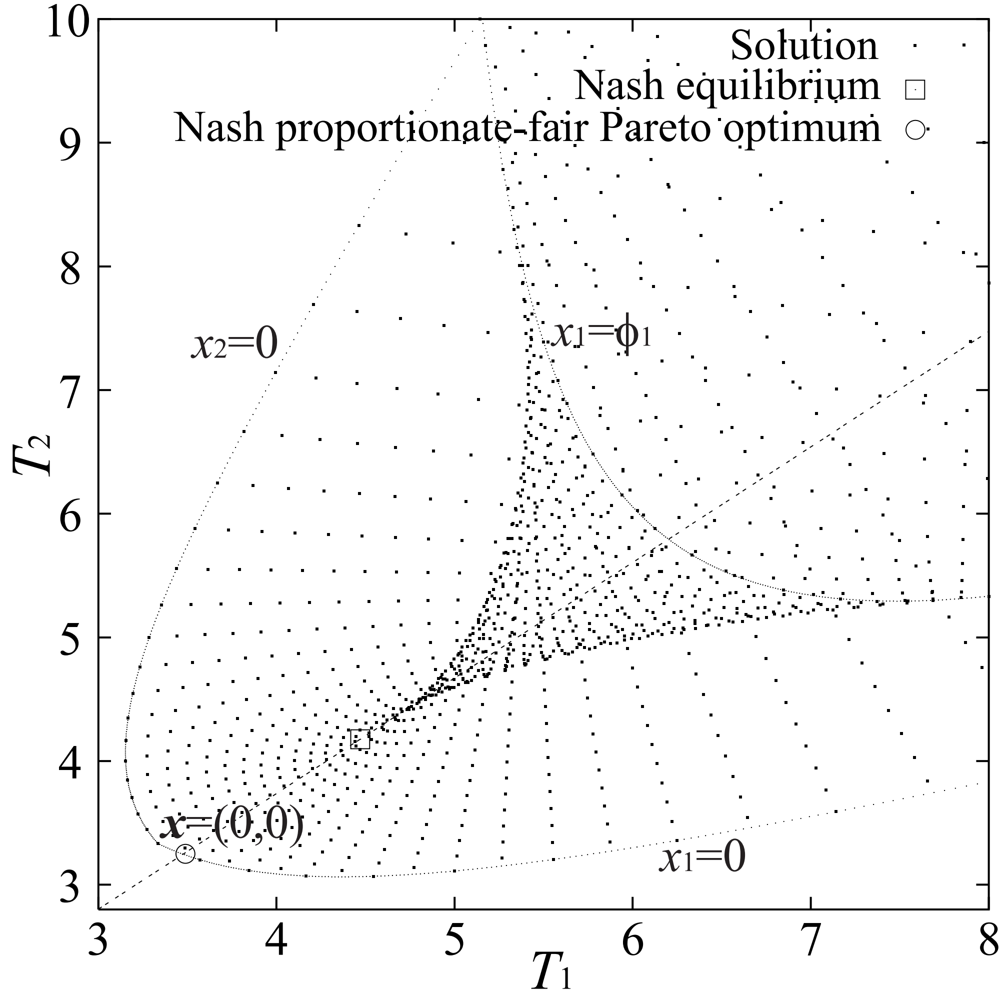

Figure 5.10: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Nash equilibrium (Nash proportionate fairness). The values of system parameters are $\phi_1 = 0.7,\ \mu_1 = 1.0,\ \phi_2 = 0.9,\ \mu_2 = 1.2,$ and $t = 3$.

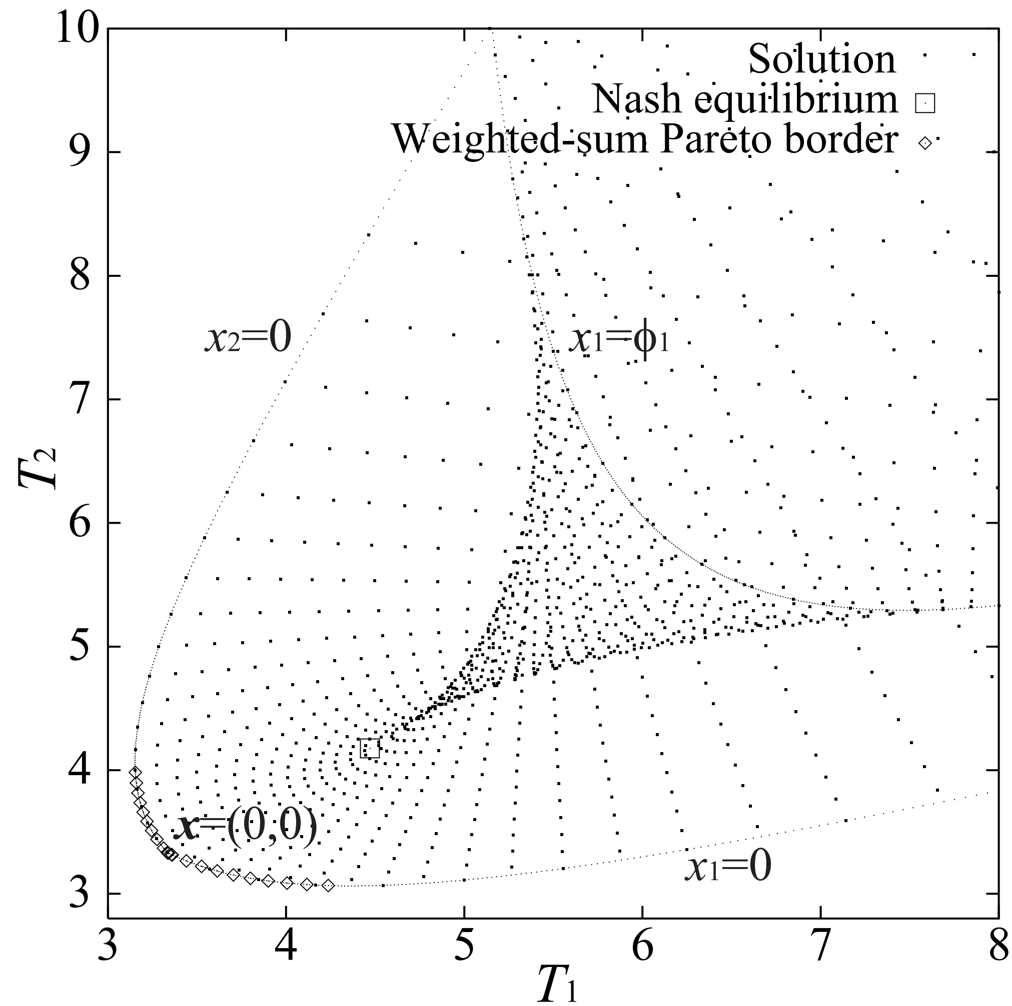

Figure 5.11: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and minimization of weighted sums of costs. The values of system parameters are $\phi_1 = 0.7,\ \mu_1 = 1.0,\ \phi_2 = 0.9,\ \mu_2 = 1.2,$ and $t = 3$.

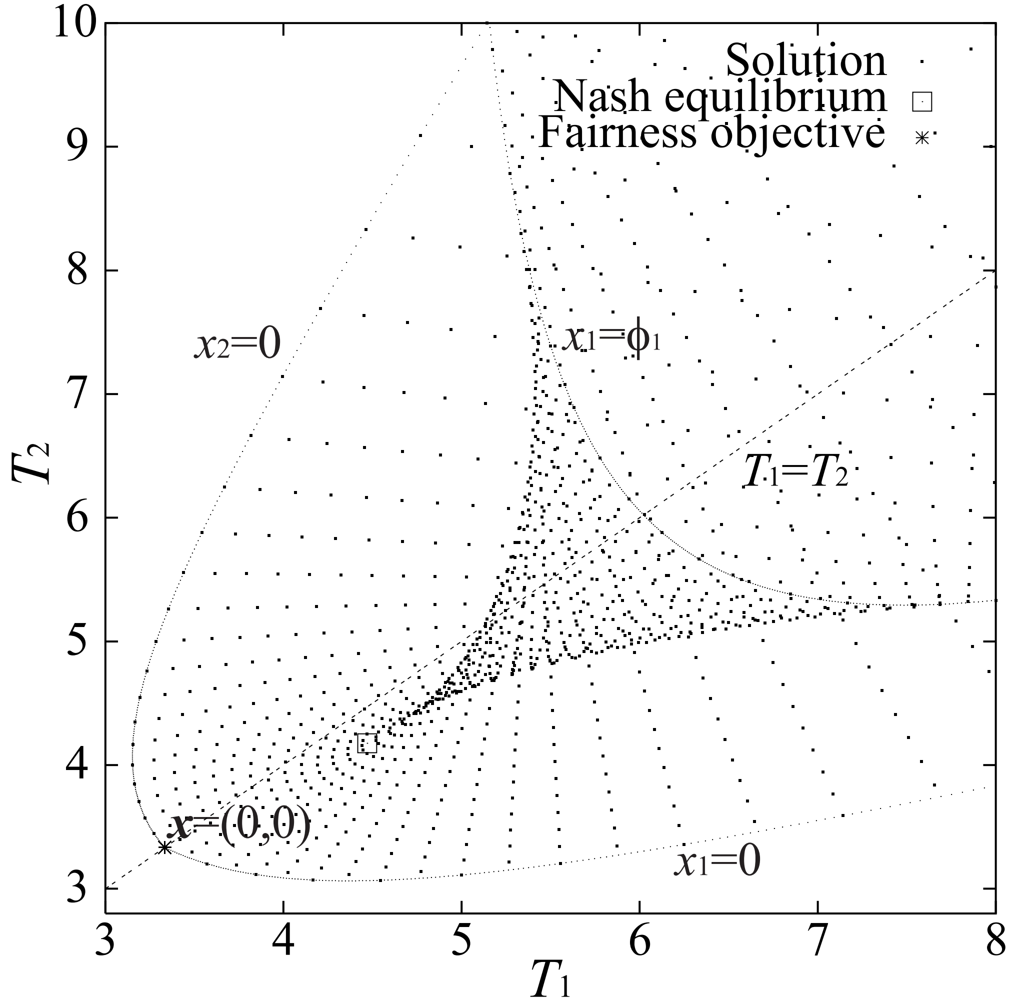

Figure 5.12: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.7). The values of system parameters are $\phi_1 = 0.7,\ \mu_1 = 1.0,\ \phi_2 = 0.9,\ \mu_2 = 1.2,$ and $t = 3$.

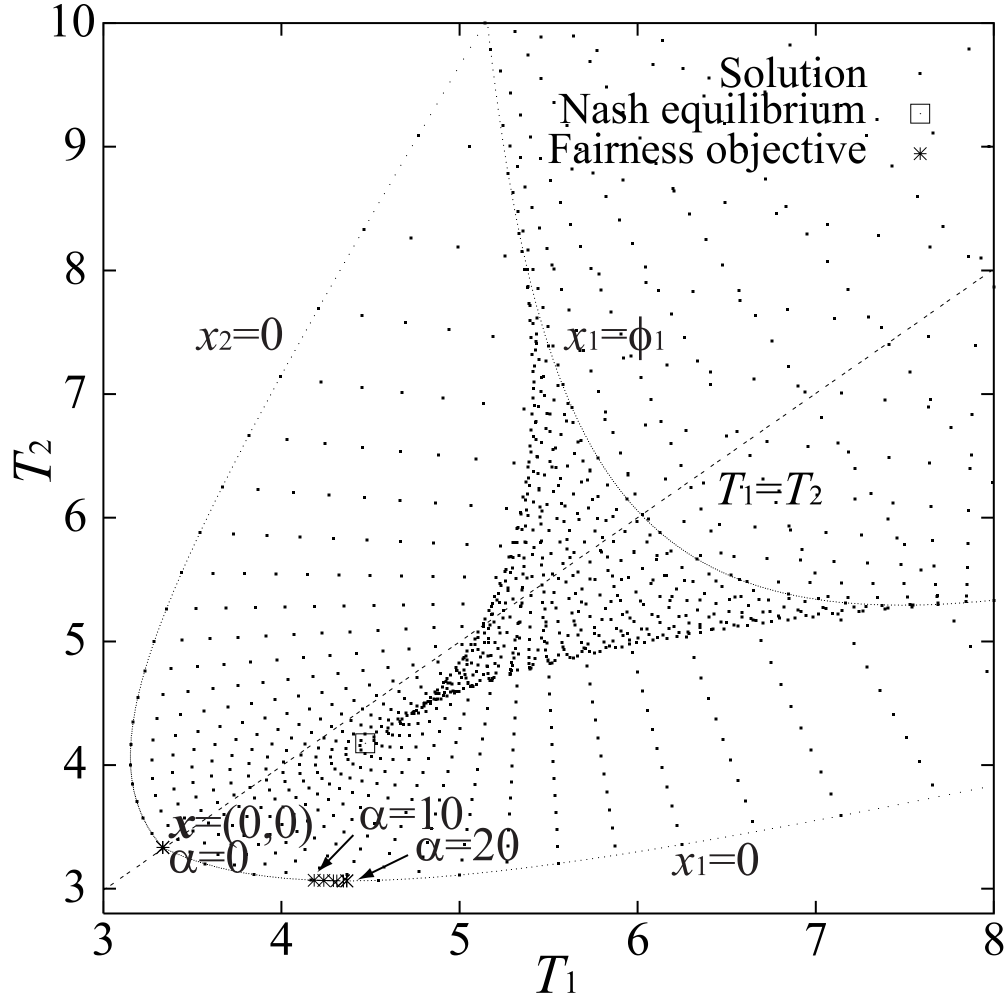

Figure 5.13: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that achieve all solutions and Fairness Objective (5.5). The values of system parameters are $\phi_1 = 0.7,\ \mu_1 = 1.0,\ \phi_2 = 0.9,\ \mu_2 = 1.2,$ and $t = 3$.

5.3.3 A case where only one Pareto optimum point achieves the fairness objectives (5.7) with various values of α

Figures 5.10-5.13 show a case where only one Pareto optimum point achieves the fairness objective (5.7) at $T_1 \simeq 3.333$ and $T_2 \simeq 3.333$, that is, the case of no load balancing ($x_1 = 0$ and $x_2 = 0$). The values of the system parameters are $\phi_1 = 0.7,\ \mu_1 = 1.0,\ \phi_2 = 0.9,\ \mu_2 = 1.2,$ and $t = 3$. Note that in this case, the following relation is satisfied: $\phi_1 - \mu_1 = \phi_2 - \mu_2$. Note that, in this case also, the Nash proportionate-fair Pareto-optimal point is different from the Pareto optimum that achieves the fairness objective (5.7).

5.3.4 A case where only one Pareto optimum point exists

Figure 5.14 shows a case where only one Pareto optimum point exists. The value of the system parameters are $\phi_1 = 0.5,\ \mu_1 = 0.7,\ \phi_2 = 0.4,\ \mu_2 = 0.7,$ and $t = 20$. We note that load balancing must be ineffective when job forwarding time t has a large value. In Figure 5.14, all optimal points that achieve the weighted-sum optimization for any combinations of the values of ξ_1 and ξ_2, the points that achieve both fairness objectives (5.5) and (5.7) with any values of α, and the Nash equilibrium point happens to be the Pareto optimal point.

5.4 Concluding Remarks

We have numerically examined the generally parameterized fairness objectives and the Nash-proportionate fairness recently introduced. The platform of this research has been simple static load balancing model with two identical servers (computers) each of which has an identical arrival and its own queue.

The points that achieve the general parameterized fairness objectives generally cover a part of the Pareto set, and at times, do not covered the Nash-proportionate-fair Pareto optimal point. Since each Pareto optimum may have its own significance, we may wish to

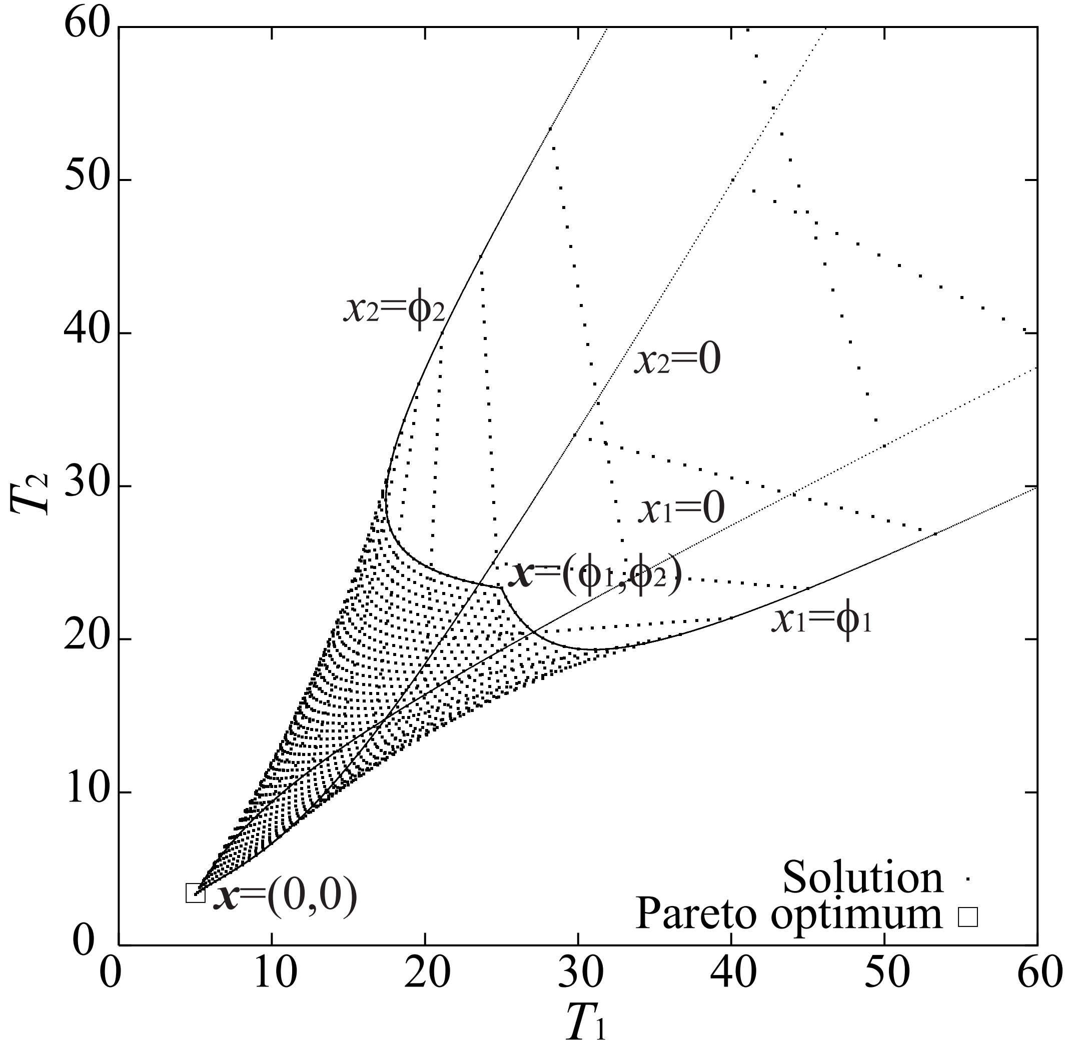

Figure 5.14: Combinations of response times, respectively, T_1 and T_2, of users 1 and 2 that acheive all solution and only one Pareto optimum. The parameter values are $\phi_1 = 0.5,\ \mu_1 = 0.7,\ \phi_2 = 0.4,\ \mu_2 = 0.7,$ and $t = 20$.

have a more generally parameterized fairness objective that all the Pareto optimal points may be achieved with a certain choice of the values of the parameters.

We have observed that careful consideration is needed in establishing the concrete form of the fairness objective along the lines of the generally parameterized fairness objectives. Otherwise, we may have an inappropriate objective that would not give us truly fair assignment of resources to users.

Future problems that remain to be solved are numerous. For example, we may need to examine other categories of models, and the analytical investigations to reveal the underlying logical structures of the problems.

Chapter 6

Conclusion

In recent large-scale networks such as Internet, one important issue is to guarantee Quality of Service (QoS) for all users. In this thesis, we have studied various types of optimal control problems in communication networks where multiple users share the system resources.

We have first introduced a survey of related work in Chapter 2. An overview and several problems on network routing, flow control and load balancing, have been presented. We have also described the Braess paradox, and its examples.

In Chapter 3, we have studied a paradox in noncooperative flow control of multiple-server (M/M/m) queues. We have considered the overall and noncooperative optimal flow control schemes, and have formulated those schemes as nonlinear programming problems to maximize the power, that is, the quotient of the throughput divided by the mean response time. To observe a paradox, two queueing systems have been considered, which are K separated M/M/N queues, and an M/M/$(K \times N)$ queue, respectively. We have reported a counter-intuitive case where the power of every user degrades after grouping together $K(> 1)$ separated M/M/N systems into a single M/M/$(K \times N)$ system. Especially, it have been numerically shown that grouping together decreases the power of every user about 10% in the worst case. We have also described our interpretation of the worst case.

In Chapter 4, we have studied Braess paradox in dynamic routing for the Cohen-Kelly network. We have dealt with two queueing networks, called the initial and augmented networks, where the augmented network is a result of adding a link to the initial network. Intuitively, we expect that adding a link to a network improved the performance of the users. Cohen and Kelly discovered that the opposite occured in static routing for those networks. We have first introduced a previous study on dynamic routing in the Cohen-Kelly network by Calvert et al, which may, however, include some confusions. We therefore have retried the same problem as Calvert et al. to avoid those confusions. We have derived dynamic routing decisions in above mentioned two networks based on the analytical study by Calvert et al. Through simulation experiments, we finally have shown that a paradox occurs in dynamic routing for the networks, that is, adding a link to a network degrades the performance of all users analogous to what Cohen and Kelly showed.

In Chapter 5, we have studied fair and Pareto optimal solutions to a load balancing problem. There exists various fairness objectives studied in relation to Pareto optimal sets and Nash equilibria. We have examined the general parameterized fairness objective proposed by Mo and Warland, which can achieve max-min or proportional fairness when the parameter of the objective has a certain value. We have also introduced Nash-proportionate-fairness objective where a Nash-proportionate-fair point is Pareto optimal, and is proportional to a Nash equilibrium solution. We have dealt with a simple static load balancing model with two identical servers (computers) each of which has an independent arrival process and its own queue. Although our load balancing model may look simple, the analysis is complicated due to the non-convexity of the delay functions. We therefore have applied those fairness objectives mainly numerically on the load balancing model. We have further studied the Pareto border of the load balancing model that are achieved by the weighted-sum optimization of the delay functions. Through numerical studies, several counter examples have been found: Points achieved by the weighted-sum optimization sometimes do not cover all the Pareto border, a delay minimization problem with the fairness obective by Mo and Warland always show a counter-intuitive behavior while the utility maximization problems always show a natural behavior.

As future work, we can consider many extensions of the above studies. Although many publications in the context of Braess paradox have appeared, the paradox is observed in very limited models. Especially, there exists few examples of the Braess paradox in dynamic networks. We can also find some computer and communication networks where users mutually compete to satisfy their requirements. As described in Section 2, overlay networks may be a possible target of research. In mobile ad-hoc networks, each mobile terminal often moves and communicates with other terminals unilaterally. We can therefore study noncooperative and cooperative (fairness) control problems in those networks.

Bibliography

[1] E. Altman. Non zero-sum stochastic games in admission, service and routing control in queueing systems. *Queueing Systems*, 23:259–279, 1996.

[2] E. Altman, R. E. Azouzi, and V. Abramov. Non-cooperative routing in loss networks. *Performance Evaluation*, 49:257–272, 2002.

[3] E. Altman, T. Başer, T. Jiménez, and N. Shimkin. Routing into two parallel links: Game-theoretic distributed algorithms. *Journal of Parallel and Distributed Computing*, 61(9):1367–1381, 2001.

[4] E. Altman, T. Başer, T. Jiménez, and N. Shimkin. Competitive routing in networks with polynomial costs. *IEEE Transactions on Automatic Control*, 47(1):92–96, Jan 2002.

[5] E. Altman, T. Başer, and R. Srikant. Nash equilibria for combined flow control and routing in networks: Asymptotic behavior for a large number of users. *IEEE Transactions on Automatic Control*, 47(6):917–930, Jun 2002.

[6] E. Altman, S. Bhulai, B. Gaujal, and A. Hordijk. Open-loop routeing to M parallel servers with no buffers. *Journal of Applied Probability*, 37:668–684, 2000.

[7] E. Altman, T. Boulogne, R. El-Azouzi, T. Jimenez, and L. Wynter. A survey on networking games in telecommunications. *Computers and Operations Research*, 33(2):286–311, 2006.

[8] E. Altman, R. El-Azouzi, and O. Pourtallier. Avoiding paradoxes in routing games. In *Proceeding of the 17th International Teletraffic Congress*, Sep 2001.

[9] E. Altman and H. Kameda. Equilibria for multiclass routing problems in multi-agent networks. *Advances in Dynamic Games*, 7:343–368, 2005.

[10] E. Altman, H. Kameda, and Y. Hosokawa. Nash equilibria in load balancing in distributed computer systems. *International Game Theory Review*, 4(2):91–100, 2002.

[11] E. Altman and N. Shimkin. Individual equilibrium and learning in processor sharing systems. *Operations Research*, 46:776–784, Nov-Dec 1998.

[12] E. Altman and L. Wynter. Equilibrium, games, and pricing in transportation and telecommunication networks. *Networks and Spatial Economics*, 4:7–21, 2004.

[13] D. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceeding of the 18th ACM Symposium on Operating System Principles*, Oct 2001.

[14] N. Arora and S. Sen. Resolving social dilemmas using genetic algorithms: Initial results. In *Proceeding of the 7th International Conference on Genetic Algorithms*, pages 19–23, Jul 1997.

[15] S. Athuraliya and S. H. Low. Optimization flow control – II: Implementation. preprint, http://netlab.caltech.edu, May, 2000.

[16] J.-P. Aubin. *Optima and Equilibria: An Introduction to Nonlinear Analysis, 2nd Ed.* Springer-Verlag, Berlin, 1998.

[17] R. E. Azouzi and E. Altman. Constrained traffic equilibrium in routing. *IEEE Transactions on Automatic Control*, 48(9):1656–1660, Sep 2003.

[18] T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Academic Press, London, 1982.

[19] F. Baskett, K. M. Chandy, R. Muntz, and F. G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22:248–260, Apr 1975.

[20] N. G. Bean, F. P. Kelly, and P. G. Taylor. Braess' paradox in a loss network. *Journal of Applied Probability*, 34:155–159, 1997.

[21] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall, Englewood Cliffs, 1992.

[22] T. Bonald and L. Massoulié. Impact of fairness on internet performance. In *Proceeding of the ACM SIGMETRICS*, pages 82–91, 2001.

[23] T. Boulogne and E. Altman. Competitive routing in multicast communications. *Networks*, 46(1):22–35, 2005.

[24] T. Boulogne, E. Altman, H. Kameda, and O. Pourtallier. Mixed equilibrium in multiclass routing games. *IEEE Transactions on Automatic Control*, 47(6):903–916, Jun 2002.

[25] T. Boulogne, E. Altman, and O. Pourtallier. On the convergence to nash equilibrium in problems of distributed computing. *Annals of Operations Research*, 109(1):279–291, Jan 2002.

[26] D. Braess. Über ein Paradoxien aus der Verkehrsplanung. *Unternehmensforschung*, 12:258–268, 1968.

[27] B. Calvert, W. Solomon, and I. Ziedins. Braess's paradox in a queueing network with state dependent routing. *Journal of Applied Probability*, 34:134–154, 1997.

[28] S. G. Chang. Fair integration of routing and flow control in communication networks. *IEEE Transactions on Communications*, 40(4):821–834, Apr 1992.

[29] S. Chen and K. Park. A distributed protocol for multi-class QoS provision in noncooperative many-switch systems. In *Proceeding of the IEEE International Conference on Network Protocols*, pages 98–107, Oct 1998.

[30] S. Chen and K. Park. An architecture for noncooperative QoS provision in many-switch systems. In *Proceeding of IEEE INFOCOM 1999*, pages 864–872, Mar 1999.

[31] W. K. Ching. A note on the convergence of asynchronous greedy algorithm with relaxation in a multiclass queueing environment. *IEEE Communication Letters*, 3(2):34–36, Feb 1999.

[32] Y. C. Chow and W. H. Kohler. Models for dynamic load balancing in a heterogeneous multiple processor system. *IEEE Transactions on Computers*, 28(5):354–361, May 1979.

[33] B. G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz. Characterizing selfishly constructed overlay routing networks. In *Proceeding of IEEE INFOCOM 2004*, 2004.

[34] J. E. Cohen. The counterintuitive in conflict and cooperation. *American Scientist*, 76:576–584, Nov-Dec 1988.

[35] J. E. Cohen and C. Jeffries. Congestion resulting from increased capacity in single-server queueing networks. *IEEE/ACM Transactions on Networking*, 5(2):305–310, 1997.

[36] J. E. Cohen and F. P. Kelly. A paradox of congestion in a queueing network. *Journal of Applied Probability*, 27:730–734, 1990.

[37] J. R. Correa, A. S. Schulz, and N. E. S. Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[38] S. Dafermos and A. Nagurney. On some traffic equilibrium theory paradoxes. *Transportation Research Part B*, 18:101–110, 1984.

[39] M. Dahlin. Interpreting stale load information. *IEEE Transactions on Parallel and Distributed Systems*, 11(10):1033–1047, Oct 2000.

[40] C. Douligeris. Multiobjective flow control in delay constrained telecommunication networks. *Journal of the Franklin Institute*, 331B(1):77–100, 1994.

[41] C. Douligeris and L. N. Kumar. Fairness issues in networking environment. *Computer Communications*, 18(4):288–299, Apr 1995.

[42] C. Douligeris and R. Mazumdar. Efficient flow control in a multiclass telecommunications environment. *IEE Proceedings-I, Communications, Speech and Vision*, 138(6):494–502, Dec 1991.

[43] C. Douligeris and R. Mazumdar. A game theoretic perspective to flow control in telecommunication networks. *Journal of the Franklin Institute*, 329:383–402, 1992.

[44] D. G. Down and M. E. Lewis. Dynamic load balancing in parallel queueing systems: Stability and optimal control. *European Journal of Operational Research*, 168:509–519, 2006.

[45] D. L. Eager, E. D. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, 12(5):662–675, May 1986.

[46] A. A. Economides and J. A. Silvester. Multi-objective routing in integrated services networks: A game theory approach. In *Proceeding of IEEE INFOCOM 1991*, pages 1220–1225, 1991.

[47] S. F. El-Zoghdy. *Certain Performance Aspects of Optimal Load Balancing*. PhD thesis, University of Tsukuba, Japan, Mar 2004.

[48] S. F. El-Zoghdy, H. Kameda, and J. Li. Numerical studies on Braess-like paradoxes for non-cooperative load balancing in distributed computer systems. In *International Workshop (Network Games and Resource Allocation)*, 2002.

[49] S. F. El-Zoghdy, H. Kameda, and J. Li. Numerical studies on paradoxes in non-cooperative load balancing in distributed computer systems. *Game Theory and Application*, 9:1–16, 2003.

[50] S. F. El-Zoghdy, H. Kameda, and J. Li. Numerical studies on a paradox for non-cooperative static load balancing in distributed computer systems. *Computers and Operations Research*, 33(2):345–355, 2006.

[51] A. Fabrikant, A. Luthra, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *Proceeding of the 23nd ACM Symposium on Principles of Distributed Computing*, 2003.

[52] Z. Fang and B. Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks. In *Proceeding of the IEEE INFOCOM 2004*, Mar 2004.

[53] R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Selfish routing in non-cooperative networks: A survey. In *Proceeding of the Conference on Mathematical Foundations of Computer Science*, pages 22–45, 2003.

[54] S. Fischer and B. Vocking. The evolution of selfish routing. In *Proceeding of the 12th European Symposium on Algorithms*, pages 323–334, 2004.

[55] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.

[56] M. Frank. The Braess paradox. *Mathematical Porogramming*, 20:283–302, 1981.

[57] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, MIT, 1991.

[58] L. Georgiadis, R. Gueŕin, V. Peris, and K. N. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, 4(4):482–501, Aug 1996.

[59] M. Gerla, H. W. Chan, and J. R. B. deMarca. Fairness in computer networks. In *Proceeding of the IEEE Internationl Conference on Communications*, pages 1384–1389, 1985.

[60] M. Gerla and L. Kleinrock. Flow control: a comparative survey. *IEEE Transactions on Communications*, 28(4):553–574, Apr 1980.

[61] R. J. Gibbens and F. P. Kelly. Resource pricing and the evolution of congestion control. *Automatica*, 35(12):1969–1985, Dec 1999.

[62] A. Giessler, J. Hänle, A. König, and E. Pade. Free buffer allocation—an investigation by simulation. *Computer Networks*, 1:199–204, 1978.

[63] W. Grassmann. The convexity of the mean queue size of the M/M/c queue with respect to the traffic intensity. *Journal of Applied Probability*, 20:916–919, 1983.

[64] D. Grosu and A. T. Chronopoulos. A game-theoretic model and algorithm for load balancing in distributed systems. In *IEEE International Parallel and Distributed Processing Symposium*, pages 146–153, Apr 2002.

[65] D. Grosu and A. T. Chronopoulos. Noncooperative load balancing in distributed systems. *Journal of Parallel and Distributed Computing*, 65:1022–1034, 2005.

[66] D. Grosu, A. T. Chronopoulos, and M.-Y. Leung. Load balancing in distributed systems: An approach using cooperative games. In *IEEE International Parallel and Distributed Processing Symposium*, pages 501–510, Apr 2002.

[67] J. N. Hagstrom and R. Abrams. Characterizing Braess's paradox for traffic networks. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pages 837–842, 2001.

[68] E. Hahne. Round-robin scheduling for max-min fairness in data network. *IEEE Journal on Selected Areas in Communications*, 9:1024–1039, Sep 1991.

[69] A. Harel and P. Zipkin. The convexity of a general performance measure for multi-server queues. *Journal of Applied Probability*, 24:725–736, 1987.

[70] A. Harel and P. Zipkin. Strong convexity result for queueing systems. *Operations Research*, 35:405–418, 1987.

[71] P. Harker. Multiple equilibrium behavior on networks. *Transportation Research Part B*, 22:39–46, 1988.

[72] A. Haurie and P. Marcotte. On the relationship between Nash-Cournot and Wardrop equilibria. *Networks*, 15:295–308, 1985.

[73] M. T. Hsiao and A. A. Lazar. Optimal decentralized flow control of Markovian queueing networks with multiple controllers. *Performance Evaluation*, 13(3):181–204, 1991.

[74] O. Imer and T. Başer. Dynamic optimization flow control. In *Proceeding of the 42th IEEE Conference on Decision and Control*, pages 2082–2087, Dec 2003.

[75] A. Inoie, H. Kameda, and C. Touati. A paradox in optimal flow control of M/M/m queues. In *Proceeding of the 43th IEEE Conference on Decision and Control*, pages 2768–2773, Dec 2004.

[76] A. Inoie, H. Kameda, and C. Touati. Pareto set, fairness, and nash equilibrium: A case study on load balancing. In *Proceeding of the 11th International Symposium on Dynamic Games and Applications*, pages 386–393, Dec 2004.

[77] A. Inoie, H. Kameda, and C. Touati. Braess paradox in dynamic routing for the cohen-kelly network. In *Proceeding of the 4th IASTED International Conference on Communications, Internet and Information Technology*, pages 335–339, Oct-Nov 2005.

[78] A. Inoie, H. Kameda, and C. Touati. A paradox in optimal flow control of M/M/n queues. *Computers and Operations Research*, 33(2):356–368, 2006.

[79] J. M. Jaffe. Bottleneck flow control. *IEEE Transactions on Commumications*, 29:955–962, Jul 1981.

[80] J. M. Jaffe. Flow control power is nondecentralizable. *IEEE Transactions on Commumications*, 29:1301–1306, Sep 1981.

[81] R. Jain. *The Art of Computer Systems Performance Analysis: Technique for Experimental Design, Measurement, Simulation, and Modeling.* Wiley-Interscience, 1991.

[82] R. Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, 28(13):1723–1738, Oct 1996.

[83] W. Jiang, D. M. Chiu, and J. C. S. Lui. On the interaction of multiple overlay routing. *Performance Evaluation*, 62, 2005.

[84] H. Kameda. How harmful the paradox can be in the Braess/Cohen-Kelly-Jeffries networks. In *Proceeding of the IEEE INFOCOM 2002*, 2002.

[85] H. Kameda. Bounds on benefits and harms of adding connections to noncooperative networks. In *Proceeding of the NETWORKING 2004*, pages 405–417, May 2004. Series: Lecture Notes in Computer Science, Vol. 3042, Mitrou, N. et al. (Eds.).

[86] H. Kameda. Bounds on the degree of paradoxical performance degradation in non-cooperative networks. *Transactions of the Operations Research Society of Japan*, 48:26–47, 2005. (in Japanese).

[87] H. Kameda, E. Altman, T. Kozawa, and Y. Hosokawa. Braess-like paradoxes in distributed computer systems. *IEEE Transactions on Automatic Control*, 45:1687–1691, 2000.

[88] H. Kameda, Y. Hosokawa, and O. Pourtallier. Effects of symmetry on Braess-like paradoxes in distributed computer systems - a numerical study -. In *Proceeding of the 40th IEEE Conference on Decision and Control*, pages 831–836, Dec 2001.

[89] H. Kameda, T. Kozawa, and J. Li. Anomalous relations among various performance objectives in distributed computer systems. In *Proceeding of the 1st World Congr. Syst. Simulation*, pages 459–465, 1997.

[90] H. Kameda, J. Li, and E. Altman. Link traffic loop free property in optimal routing for multi-class networks. In *Proceeding of the 43rd IEEE Conference on Decision and Control*, pages 4217–4224, Dec 2004.

[91] H. Kameda, J. Li, C. Kim, and Y. Zhang. *Optimal Load Balancing in Distributed Computer Systems*. Springer, 1997.

[92] H. Kameda and O. Pourtallier. Paradoxes in distributed decisions on optimal load balancing. *Journal of the ACM*, 32:407–433, 2002.

[93] H. Kameda and Y. Zhang. Uniqueness of the solution for optimal static routing in open BCMP queueing networks. *Mathematical and Computer Modeling*, 22(10-12):119–130, 1995.

[94] K. Kar, S. Sarkar, and L. Tassiulas. A simple rate control algorithm for maximizing total user utility. In *Proceeding of IEEE INFOCOM 2001*, pages 133–141, Apr 2001.

[95] F. P. Kelly. Loss networks. *Annals of Applied Probability*, 1:319–378, 1991.

[96] F. P. Kelly. Network routing. *Philosophical Transactions of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 337:343–367, 1991.

[97] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[98] F. P. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[99] L. Kencl and J.-Y. L. Boudec. Adaptive load sharing for network processors. In *Proceedings of IEEE INFOCOM 2002*, pages 545–554, Jun 2002.

[100] C. Kim and H. Kameda. Optimal static load balancing of multi-class jobs in a distributed computer system. In *In 10th IEEE International Conference on Distributed Computing Systems*, pages 562–569, Jun 1990.

[101] C. Kim and H. Kameda. Optimal static load balancing of multi-class jobs in a distributed computer system. *The Transactions of the IEICE*, E73(7):1207–1214, Jul 1990.

[102] C. Kim and H. Kameda. An algorithm for optimal static load balancing in distributed computer systems. *IEEE Transactions on Computers*, 41(3):381–384, Mar 1992.

[103] C. Kim and H. Kameda. Parametric analysis of static load balancing of multi-class jobs in a distributed computer system. *IEICE Transactions on Information and Systems*, E75-D(4):527–534, Jul 1992.

[104] J. Kleinberg, Y. Rabani, and É. Tardos. Fairness in routing and load balancing. In *Proceeding of the 40th Annual IEEE Symposium on the Foundations of Computer Science*, pages 568–578, Oct 1999.

[105] J. Kleinberg, Y. Rabani, and É. Tardos. Fairness in routing and load balancing. *Journal of Computer and System Sciences*, 63:2–20, 2001.

[106] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications*. John Wiley, New York, 1976.

[107] L. Kleinrock. On flow control in computer networks. In *Proceeding of the IEEE Internationl Conference on Communications*, pages 27.2.1–27.2.5, 1978.

[108] L. Kleinrock. Power and deterministic rules of thumb for probabilistic problems in computer communications. In *Proceeding of the IEEE Internationl Conference on Communications*, pages 43.1.1–43.1.10, 1979.

[109] L. Kleinrock. On the modeling and analysis of computer networks. *Proceedings of the IEEE*, pages 1179–1191, 1993.

[110] C. Kolias and L. Kleinrock. The power function as a performance and comparison measure for ATM switches. In *Proceeding of the IEEE Globecom'98*, pages 381–386, 1998.

[111] Y. Korilis and A. Lazar. Why is flow control hard: Optimality, fairness, partial and delayed information. In *Proceeding of the 2nd ORSA Telecommunications Conference*, Mar 1992.

[112] Y. A. Korilis and A. A. Lazar. On the existence of equilibria in noncooperative optimal flow control. *Journal of the ACM*, 42(3):584–613, May 1995.

[113] Y. A. Korilis, A. A. Lazar, and A. Orda. Architecting noncooperative networks. *Journal on Selected Areas in Communications*, 15(7):1241–1251, Sep 1995.

[114] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1):161–173, Feb 1997.

[115] Y. A. Korilis, A. A. Lazar, and A. Orda. Capacity allocation under non-cooperative routing. *IEEE Transactions on Automatic Control*, 42(3):309–325, Mar 1997.

[116] Y. A. Korilis, A. A. Lazar, and A. Orda. Avoiding the Braess paradox in non-cooperative networks. *Journal of Applied Probability*, 36:211–222, 1999.

[117] Y. A. Korilis and A. Orda. Incentive compatible pricing strategies for QoS routing. In *Proceeding of IEEE INFOCOM 1999*, Mar 1999.

[118] Y. A. Korilis, T. A. Varvarigou, and S. R. Ahuja. Incentive-compatible pricing strategies in noncooperative networks. In *Proceeding of IEEE INFOCOM 1998*, pages 439–446, Apr 1998.

[119] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[120] K. B. Kumar and J. M. Jaffe. A new approach to performance-oriented flow control. *IEEE Transactions on Communications*, 29(4):427–435, Apr 1981.

[121] L. N. Kumar, C. Douligeris, and G. Develekos. Implemention of a decentralized Pareto optimal algorithm. *Computer Communications*, 17(8), Aug 1994.

[122] S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ECN marks. In *Proceeding of IEEE INFOCOM 2000*, pages 1323–1332, Mar 2000.

[123] S. Kunniyur and R. Srikant. A time scale decomposition approach to adaptive ECN marking. In *Proceeding of IEEE INFOCOM 2001*, pages 1331–1339, Apr 2001.

[124] S. Kunniyur and R. Srikant. A time scale decomposition approach to adaptive ECN marking. *IEEE Transactions on Automatic Control*, 47(6):882–894, Jun 2002.

[125] R. J. La and V. Anantharam. Optimal routing control: Game theoretic approach. In *Proceeding of the 36th IEEE Conference on Decision and Control*, pages 2910–2915, Dec 1997.

[126] R. J. La and V. Anantharam. Network princing using game theoretic approach. In *Proceeding of the 38th IEEE Conference on Decision and Control*, pages 4008–4013, Dec 1999.

[127] R. J. La and V. Anantharam. Charge-sensitive tcp and rate control in the internet. In *Proceeding of IEEE INFOCOM 2000*, pages 1166–1175, Mar 2000.

[128] R. J. La and V. Anantharam. Window-based congestion control with heterogeneous users. In *Proceeding of the IEEE INFOCOM 2001*, pages 22–26, Apr 2001.

[129] R. J. La and V. Anantharam. Optimal routing control: Repeat game approach. *IEEE Transactions on Automatic Control*, 47(3):437–450, Mar 2002.

[130] R. J. La and V. Anantharam. Utility-based rate control in the internet for elastic traffic. *IEEE Transactions on Networking*, 10(2):272–286, Apr 2002.

[131] A. A. Lazar. Optimal flow control of a class of queueing networks in equilibrium. *IEEE Transactions on Automatic Control*, 28(11):1001–1007, Nov 1983.

[132] A. A. Lazar. The throughput time delay function of an M/M/1 queue. *IEEE Transactions on Information Theory*, 29:914–918, Nov 1983.

[133] A. A. Lazar. Optimal flow control of an M/M/m queue. *Journal of the ACM*, 32(1):86–98, Jan 1984.

[134] H. Lee and M.Cohen. Multi-agent customer allocation in a stochastic service system. *Management Science*, 31(6):752–763, Jun 1985.

[135] H. L. Lee and M. A. Cohen. A note on the convexity of performance measures of M/M/c queueing systems. *Journal of Applied Probability*, 20:920–923, 1983.

[136] J. Li and H. Kameda. Optimal load balancing in tree networks with two-way traffic. *The International Journal of Distributed Informatique*, 25(12):1335–1348, 1993.

[137] J. Li and H. Kameda. A decomposition algorithm for optimal static load balancing in tree hierarchy network configurations. *IEEE Transactions on Parallel and Distributed Systems*, 5(5):540–548, 1994.

[138] J. Li and H. Kameda. Optimal load balancing in star network configurations with two-way traffic. *Journal of Parallel and Distributed Computing*, 23(3):364–375, Dec 1994.

[139] J. Li and H. Kameda. Optimal load balancing problems for multiclass jobs in distributed/parallel computer systems. *IEEE Transactions on Computers*, 47(3):322–332, Mar 1998.

[140] L. Libman and A. Orda. The designer's perspective to atomic noncooperative network. *IEEE/ACM Transactions on Networking*, 7(6):875–884, Dec 1999.

[141] L. Libman and A. Orda. Atomic resource sharing in noncooperative network. *Telecommunication Systems*, 17(4):385–409, Aug 2001.

[142] L. Libman and A. Orda. Optimal sliding-window strategies in networks with long round-trip delays. *Computer Networks*, 46:219–235, 2004.

[143] H. Lin, T. Roughgarden, and É. Tardos. A stronger bound on Braess's paradox. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 333–334, 2004.

[144] H. Lin, T. Roughgarden, E. Tardos, and A. Walkover. Braess's paradox, fibonacci numbers, and exponential inapproximability. In *Proceeding of the 32nd International Colloquium on Automata, Languages and Programming*, Jul 2005.

[145] S. Liu, T. Başer, and R. Srikant. Controlling the internet: A survey and some new results. In *Proceeding of the 42th IEEE Conference on Decision and Control*, pages 3048–3057, Dec 2003.

[146] Y. Liu and M. A. Simaan. Noninferior Nash strategies for multi-team systems. *Journal of Optimization Theory and Applications*, 120(1):29–51, Mar 2004.

[147] Y. Liu and M. A. Simaan. Competitive team strategies for routing control in parallel-link communications networks. In *Proceeding of the American Control Conference*, Jun 2005.

[148] Y. Liu and M. A. Simaan. Non-inferior Nash strategies for routing control in parallel-link communication networks. *International Journal of Communication Systems*, 18:347–361, Mar 2005.

[149] Y. Liu, H. Zhang, W. Gong, and D. Towsley. On the interaction between overlay routing and traffic engineering. In *Proceeding of IEEE INFOCOM 2005*, 2005.

[150] S. Low and D. Lapsley. Optimization flow control –I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, Dec 1999.

[151] S. H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Transactions on Networking*, 11(4):525–536, Aug 2003.

[152] P. Marbach. Priority service and max-min fairness. *IEEE Transactions on Networking*, 11(5):733–746, 2003.

[153] J. W. Mark. Fairness issue in computer communication networks. In *Proceeding of the IEEE Internationl Conference on Communications*, pages 1368–1372, 1985.

[154] L. Massoulié and J. Roberts. Bandwidth sharing: Objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10(3):320–328, Jun 2002.

[155] Y. Masuda and S. Whang. Capacity management in decentralized networks. *Management Science*, 48:1628–1634, Dec 2002.

[156] M. Matsumoto and T. Nishimura. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, Jan 1998.

[157] M. Mavronicolas and P. Spirakis. The price of selfish routing. In *Proceeding of the 33rd Annual ACM Symposium on the Theory of Computing*, pages 510–519, 2001.

[158] A. Mayer, Y. Ofek, and M. Yung. Approximating max-min fair rates via distributed local scheduling with partial information. In *Proceeding of the IEEE INFOCOM 1996*, pages 926–936, Mar 1996.

[159] R. Mazumdar, L. G. Mason, and C. Douligeris. Fairness in network optimal flow control: Optimality of product forms. *IEEE Transactions on Communications*, 39(5):237–252, May 1991.

[160] D. Mitra and R. J. Gibbens. State-dependent routing on symmetrical loss networks with trunk reservation. *IEEE Transactions on Communications*, 41(2):400–411, Feb 1993.

[161] M. Mitzenmacher. How useful is old information. *IEEE Transactions on Parallel and Distributed Systems*, 11(1):1033–1047, Jan 2000.

[162] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, Oct 2000.

[163] J. D. Murchland. Braess's paradox of traffic flow. *Transportion Research*, 4:391–394, 1970.

[164] J. F. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[165] J. F. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[166] A. Orda, R. Rom, and N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1:614–627, 1993.

[167] C. H. Papadimitriou. Algorithms, games and the internet. In *33rd Annual ACM Symposium on Theory of Computing*, pages 749–753, Jul 2001.

[168] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, Jun 1993.

[169] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, Apr 1994.

[170] V. Pareto. *Manuel d'Économie Politique*. Girard et Brière, 1909.

[171] K. Park, M. Sitharam, and S. Chen. Quality of service provision in noncooperative networks with diverse user requirements. *Decision Support Systems*, 28(1-2):101–122, Mar 2000.

[172] M. Patriksson. *The Traffic Assignment Problem: Models and Methods*. Utrecht, The Netherlands: VSP, 1994.

[173] M. L. Puterman. *Markov Decision Processes*. Wiley Interscience, 1994.

[174] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker. On selfish routing in Internet-like environments. In *Proceeding of the ACM SIGCOMM 2003*, pages 151–162, 2003.

[175] S. H. Rhee and T. Konstantopoulos. Optimal flow control and capacity allocation in multi-service networks. In *Proceeding of the 37th IEEE Conference on Decision and Control*, pages 1662–1667, Dec 1998.

[176] J. Roberts and L. Massoulié. Bandwidth sharing and admission control for elastic traffic. In *Proceeding of the 11th ITC Specialist Seminar on Teletraffic Issues related to Multimedia and Nomadic Communications*, Oct 1998.

[177] J. B. Rosen. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica*, 33:153–163, 1965.

[178] T. Roughgarden. Designing networks for selfish users is hard. In *Proceeding of the Symposium on Foundations of Computer Sciences*, pages 472–481, Oct 2001.

[179] T. Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, USA, May 2002.

[180] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, Sep 2003.

[181] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.

[182] T. Roughgarden and É. Tardos. How bad is selfish routing. *Journal of the ACM*, 49(2):236–259, Mar 2002.

[183] T. Roughgarden and É. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2):389–403, May 2004.

[184] A. K. S. Chung and K. W. Ross. Computing approximate blocking probabilities for large loss networks with state-dependent routing. *IEEE Transactions on Networking*, 1(1):105–115, 1993.

[185] I. Sahin and M. A. Simaan. A game theoretic flow and routing control policy for two-node parallel link communication networks with multiple users. In *Proceeding of the 15th IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, pages 2478–2482, 2004.

[186] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: A case for informed internet routing and transport. *IEEE Micro*, 19(1):50–59, Jan 1999.

[187] J. M. Selga. New flow control power is decentralizable and fair. In *Proceeding of IEEE INFOCOM 1984*, pages 87–94, Apr 1984.

[188] S. J. Shenker. Making greed work in networks: A game theoretic analysis of switch service disciplines. *IEEE/ACM Transaction on Networking*, 3(6):819–831, Dec 1995.

[189] R. Srinivasan and A. Somani. On achieving fairness and efficiency in high-speed shared medium access. *IEEE/ACM Transactions on Networking*, 8(5):111–124, Feb 2003.

[190] H. V. Stackelberg. *Marktform und Gleichgewicht*. Julius Springer, Vienna, Austria, 1934.

[191] N. F. Stewart. Equilibrium vs system-optimal flow: some examples. *Transportation Research Part A*, 14:81–84, 1980.

[192] A. Taguchi. Braess' paradox in a two-terminal transportation network. *Journal of the Operations Research Society of Japan*, 25(4):376–388, Dec 1982.

[193] A. Tang, J. Wang, and H. Low. Is fair allocation always inefficient. In *Proceeding of the IEEE INFOCOM 2004*, Mar 2004.

[194] A. N. Tantawi and D. Towsley. A general model for optimal static load balancing in star network configurations. In E. Gelenbe, editor, *PERFORMANCE'84*, pages 277–291. Elsevier Science Publishers, 1984.

[195] A. N. Tantawi and D. Towsley. Optimal static load balancing in distributed computer systems. *Journal of the ACM*, 32(2):445–465, Apr 1985.

[196] C. Touati. *Les Principes d'Equité Appliqués aux Réseaux de Télécommunications*. PhD thesis, INRIA, France, Sep 2003.

[197] C. Touati, E. Altman, and J. Galtier. On fairness in bandwidth allocation. Technical report, INRIA Research Report RR 4269, Sep 2001.

[198] C. Touati, E. Altman, and J. Galtier. Utility based fair bandwidth allocation. In *Proceeding of the IASTED International Conference on Networks, Parallel and Distributed Processing and Applications*, pages 126–131, Oct 2002.

[199] C. Touati, A. Inoie, and H. Kameda. Some properties of Pareto sets in load balancing. In *Proceeding of the 11th International Symposium on Dynamic Games and Applications*, pages 1033–1040, Dec 2004.

[200] C. Touati, H. Kameda, and A. Inoie. Fairness in non-convex systems. Technical report, CS-TR-05-4, Sep 2005.

[201] W.-H. Wang, M. Palaniswami, and S. H. Low. Optimal flow control and routing in multi-path networks. *Performance Evaluation*, 52:119–132, 2003.

[202] J. G. Wardrop. Some theoretical aspects of road traffic research. In *Institute of Civil Engineers, Pt. II*, volume 1, pages 325–378, 1952.

[203] B. W. Wie. A difference game approach to the dynamic mixed behavior traffic network equilibrium problem. *European Journal of Operational Research*, 83:117–136, 1995.

[204] B. W. Wie and R. L. Tobin. On the relationship between dynamic Nash and insrtantaneous user equilibria. *Networks*, 32:141–163, 1998.

[205] F. Wong and J. R. B. deMarca. Fairness in window flow controlled computer networks. *IEEE Transactions on Communications*, 37(5):475–480, May 1989.

[206] H. Yaïche, R. R. Mazumdar, and C. Rosenberg. A game theoretic framework for bandwidth allocation and pricing in broadband network. *IEEE/ACM Transactions on Networking*, 8(5):667–677, Oct 2000.

[207] Z. Zeng and B. Veeravalli. Design and analysis of a non-preemptive decentralized load balancing algorithm for multi-class jobs in distributed networks. *Computer Communications*, 27:679–693, 2004.

[208] Y. Zhang, H. Kameda, and K. Shimizu. Parametric analysis of optimal load balancing in distributed computer systems. *Jornal of Information Processing*, 14(4):433–441, 1992.

[209] Z. Zhang and C. Douligeris. Convergence of synchronous and asynchronous greedy algorithms in a multiclass telecommunications environment. *IEEE Transactions on Communications*, 40(8):1277–1281, Aug 1992.

[210] S. Zhou. A trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, 14(9):1327–1341, Feb 1988.

Appendix A

Existence and Uniqueness of Solutions in Flow Control Problems of M/M/m queues

First, we show in this appendix, that there exist solutions to problems (I) and (II) in Chapter 3. Then, we prove that they are unique. Note that for M/M/1 queueing system models, closed form solutions to noncooperative and overall optimization problems have been obtained (see e.g., [43]). It seems, however, that there is no solution to M/M/m models for $m > 1$, has been published.

Lemma A.1. 1) *For $0 \le \Lambda \le m\mu$, the overall power, $P(\Lambda)$ is strictly concave in Λ.*

2) *For $0 \le \lambda_l \le m\mu - \sum_{j \ne l} \lambda_j$, the power of player l, $p_l(\boldsymbol{\lambda})$, is strictly concave in λ_l for $l = 1, 2, \ldots, r$.*

Proof. Since 1) is equivalent to 2) in the case where $r = 1$, we prove 2). The second derivative of p_l with respect to λ_l is as follows:

$$\begin{aligned}\frac{\partial^2 p_l(\boldsymbol{\lambda})}{\partial \lambda_l^2} &= 2\frac{\partial}{\partial \lambda_l}\left\{\frac{1}{T(\Lambda)}\right\} + \lambda_l \frac{\partial^2}{\partial \lambda_l^2}\left\{\frac{1}{T(\Lambda)}\right\} \\ &= -\frac{2}{T(\Lambda)^2}\frac{\partial T(\Lambda)}{\partial \lambda_l} + \lambda_l \frac{\partial^2}{\partial \lambda_l^2}\left\{\frac{1}{T(\Lambda)}\right\}.\end{aligned} \tag{A.1}$$

Now, we wish to show (A.1) is negative. From [63], [69], and [135], the response time $T(\Lambda)$ is strictly increasing in ρ, where

$$\rho = \sum_{l=1}^{r} \frac{\lambda_l}{m\mu} = \frac{\Lambda}{m\mu},$$

and hence $T(\Lambda)$ is also strictly increasing in λ_l. Therefore, the first term of (A.1) is negative.

We rewrite the second term of (A.1) as follows:

$$\lambda_l \frac{\partial^2}{\partial \lambda_l^2} \left\{ \frac{1}{T(\Lambda)} \right\} = \lambda_l \frac{\partial^2}{\partial \rho^2} \left\{ \frac{1}{T(\Lambda)} \right\} \left(\frac{\partial \rho}{\partial \lambda_l} \right)^2 .$$

From [70], we have

$$\frac{d^2}{d\rho^2} \left\{ \frac{1}{T(\Lambda)} \right\} < 0.$$

Note that

$$\frac{\partial \rho}{\partial \lambda_l} = \frac{1}{m\mu}. \tag{A.2}$$

Therefore, the second term of (A.1) is nonpositive. Finally, the left-hand side of (A.1) is negative.

□

We define the function g by

$$g(\Lambda) = \frac{T(\Lambda)}{T'(\Lambda)} - \frac{\Lambda}{r}, \tag{A.3}$$

where $T'(\Lambda) = dT(\Lambda)/d\Lambda$.

Lemma A.2. *Any solution to the noncooperative optimization problem (I) is symmetric, that is,*

$$\widehat{\lambda}_l = \widehat{\lambda}_{l'} = \frac{\widehat{\Lambda}}{r}, \ \ l, l' = 1, 2, \ldots, r, \tag{A.4}$$

and satisfies $g(\widehat{\Lambda}) = 0$.

Proof. We denote a solution to (I) by $\widehat{\boldsymbol{\lambda}}$. It satisfies $\widehat{\lambda}_l > 0$ for all $l = 1, 2, \ldots, r$ and $\sum_l \widehat{\lambda}_l < m\mu$. Moreover, it is a solution to $\partial p_l(\boldsymbol{\lambda})/\partial \lambda_l = 0,\ l = 1, 2, \ldots, r$, which is equivalent to

$$\lambda_l \frac{\partial T(\Lambda)}{\partial \lambda_l} - T(\Lambda) = 0,\ l = 1, 2, \ldots, r. \tag{A.5}$$

Since $T(\Lambda)$ depends only on Λ, then

$$\frac{\partial T(\Lambda)}{\partial \lambda_l} = \frac{\partial T(\Lambda)}{\partial \lambda_{l'}} = \frac{dT(\Lambda)}{d\Lambda},\ l, l' = 1, 2, \ldots, r.$$

Then, any solution to problem (I) satisfies $\lambda_l = T(\Lambda)/T'(\Lambda)$ for all $l = 1, 2, \ldots, r$ with $T'(\Lambda) = dT(\Lambda)/d\Lambda$. $\widehat{\boldsymbol{\lambda}}$ is therefore symmetric, and given by $g(\widehat{\Lambda}) = 0$ and $\widehat{\lambda}_l = \widehat{\Lambda}/r,\ l = 1, 2, \ldots, r$. □

Therefore, we obtain the following proposition:

Proposition A.1. 1) *The solution to the noncooperative optimization problem (I) is unique, and is a solution of the following equation:*

$$g(\Lambda) = \frac{T(\Lambda)}{T'(\Lambda)} - \frac{\Lambda}{r} = 0. \tag{A.6}$$

2) *The solution to the overall optimization problem (II) is unique, and is a solution of the following equation:*

$$\frac{T(\Lambda)}{T'(\Lambda)} - \Lambda = 0. \tag{A.7}$$

Proof. 1) From Lemma A.2, a solution to problem (I) is given by $g(\Lambda) = 0$, which is (A.6). If g is a strictly decreasing function of Λ for $0 \le \Lambda \le m\mu$, and satisfies $g(0) > 0$ and $g(m\mu) < 0$, there exists a unique solution to (A.6). The derivative of g with respect to Λ is

$$\begin{aligned}\frac{dg(\Lambda)}{d\Lambda} &= \frac{T'(\Lambda)^2 - T(\Lambda)T''(\Lambda)}{T'(\Lambda)^2} - \frac{1}{r} \\ &= -\frac{T(\Lambda)T''(\Lambda) - 2T'(\Lambda)^2}{T'(\Lambda)^2} - 1 - \frac{1}{r},\end{aligned}$$

where

$$T''(\Lambda) = \frac{d^2T(\Lambda)}{d\Lambda^2}.$$

In [70], it is shown that $d^2T^{-1}/d\Lambda^2 < 0$, that is to say that

$$T(\Lambda)T''(\Lambda) - 2T'(\Lambda)^2 > 0.$$

Then, we have $dg(\Lambda)/d\Lambda < 0$ for $0 \leq \Lambda \leq m\mu$. Therefore, g is a strictly decreasing function of Λ.

Next, we show $g(0) > 0$ and $g(m\mu) < 0$. From [63], [69], and [135], the function T is increasing and strictly convex in Λ for $0 \leq \Lambda \leq m\mu$. Then $T'(\Lambda)$ is positive for $\Lambda = 0$. From (3.1), $T(\Lambda)$ is also positive for $\Lambda = 0$. Therefore, from (A.3) we have $g(0) > 0$.

The derivatives of T and B_m are

$$\frac{dT(\Lambda)}{d\Lambda} = \frac{B'_m(\Lambda)(m\mu - \Lambda) + B_m(\Lambda)}{(m\mu - \Lambda)^2}$$

and

$$\begin{aligned} B'_n(\Lambda) &= \frac{dB_m(\Lambda)}{d\Lambda} \\ &= \frac{B_m(\Lambda)[m\mu\Lambda(1 - B_m(\Lambda)) + m(m\mu - \Lambda)^2]}{m\mu\Lambda(m\mu - \Lambda)}, \end{aligned}$$

respectively. Then if $\Lambda \neq 0$, $g(\Lambda)$ is rewritten as follows:

$$g(\Lambda) = \frac{m\Lambda(m\mu - \Lambda)(\mu B_m(\Lambda) + m\mu - \Lambda)}{B_m(\Lambda)[m\mu\Lambda + (1 - B_m(\Lambda)) + m(m\mu - \Lambda)^2]} - \frac{\Lambda}{r}.$$

Since $B_m(m\mu) = 1$, we obtain $g(m\mu) < 0$. Therefore, (A.6) has a unique solution.

2) We denote a solution to (II) by $\widetilde{\Lambda}$. Note that P is positive for $0 < \Lambda < m\mu$, and zero for $\Lambda \leq 0$ and $\Lambda \geq m\mu$. Also from Lemma A.1, P is strictly concave for $0 \leq \Lambda \leq m\mu$. Then problem (II) has a unique solution, and $\widetilde{\Lambda}$ is a solution to $dP(\Lambda)/d\Lambda = 0$ for $0 < \widetilde{\Lambda} < m\mu$, which is equivalent to (A.7). □

Each of eqs. (A.6) and (A.7) has a single variable, respectively. Therefore, we can obtain the solutions simply by using an iterative algorithm. We provide such an algorithm.

Algorithm

Step 1. Set $x_0 = 0$, $y_0 = m\mu$, $\varepsilon_0 = y_0 - x_0$, $\varepsilon > 0$, and $k = 0$.

Step 2. (I) If

$$\frac{T(\Lambda)}{T'(\Lambda)} - \frac{\Lambda}{r} \leq 0,$$

then set $y_k = \varepsilon_k$ otherwise set $x_k = \varepsilon_k$.

(II) If

$$\frac{T(\Lambda)}{T'(\Lambda)} - \Lambda \leq 0,$$

then set $y_k = \varepsilon_k$ otherwise set $x_k = \varepsilon_k$.

Step 3. Compute $\varepsilon_k = y_k - x_k$.

Step 4. If $|\varepsilon_k| > \varepsilon$ then go to step 2.

Step 5. (I) Set $\widehat{\Lambda} = \varepsilon_k$.

(II) Set $\widetilde{\Lambda} = \varepsilon_k$.

List of Publications and Presentations

1. Atsushi Inoie, Hisao Kameda and Corinne Touati, "A Paradox in Optimal Flow Control of M/M/m queues," Proc. 43rd IEEE Conference on Decision and Control (CDC), Dec 2004, Bahamas.

2. Atsushi Inoie, Hisao Kameda and Corinne Touati, "A Paradox in Optimal Flow Control of M/M/n queues," Computers and Operations Research, vol. 33, no. 2, pp 356-368, 2006.

3. Atsushi Inoie, Hisao Kameda and Corinne Touati, "Pareto Set, Fairness, and Nash Equilibrium: A Case Study on Load Balancing," Proc. 11th International Symposium on Dynamic Games and Applications (ISDG), Dec 2004, Tucson, USA.

4. Corinne Touati, Atsushi Inoie and Hisao Kameda , "Some Properties of Pareto Sets in Load balancing," Proc. 11th International Symposium on Dynamic Games and Applications (ISDG), Dec 2004, Tucson, USA.

5. Atsushi Inoie, Hisao Kameda and Corinne Touati, "Braess Paradox in Dynamic Routing for the Cohen-Kelly Network", Proc. 4th IASTED International Conference on Communications, Internet and Information Technology (CIIT), Oct-Nov 2005, Cambridge, USA.

6. Corinne Touati, Hisao Kameda and Atsushi Inoie, "Fairness in Non-convex Systems," CS Technical Report CS-TR-05-4, September, 2005.