Probability in the Engineering and Informational Sciences, **30**, 2016, 455–469.
doi:10.1017/S0269964816000097

PERFORMANCE OF NON-COOPERATIVE ROUTING OVER PARALLEL NON-OBSERVABLE QUEUES

Olivier Brun
LAAS-CNRS
Université de Toulouse
7 Avenue du Colonel Roche
31077 Toulouse, France
E-mail: brun@laas.fr

Autonomic computing is emerging as a significant new approach to the design of computer services. Its goal is the development of services that are able to manage themselves with minimal direct human intervention, and, in particular, are able to sense their environment and to tune themselves to meet end-user needs. However, the impact on performance of the interaction between multiple uncoordinated self-optimizing services is not yet well understood. We present some recent results on a non-cooperative load-balancing game which help to better understand the result of this interaction. In this game, users generate jobs of different services, and the jobs have to be processed on one of the servers of a computing platform. Each service has its own dispatcher which probabilistically routes jobs to servers so as to minimize the mean processing cost of its own jobs. We first investigate the impact of heterogeneity in the amount of incoming traffic routed by dispatchers and present a result stating that, for a fixed amount of total incoming traffic, the worst-case overall performance occurs when each dispatcher routes the same amount of traffic. Using this result we then study the so-called Price of Anarchy (PoA), an oft-used worst-case measure of the inefficiency of non-cooperative decentralized architectures. We give explicit bounds on the PoA for cost functions representing the mean delay of jobs when the service discipline is PS or SRPT. These bounds indicate that significant performance degradations can result from the selfish behavior of self-optimizing services. In practice, though, the worst-case scenario may occur rarely, if at all. Some recent results suggest that for the game under consideration the PoA is an overly pessimistic measure that does not reflect the performance obtained in most instances of the problem.

1. INTRODUCTION

Even small degradations in the performance and availability of modern computer services can have a considerable business impact. These services usually require continuous operation over time, always maintaining the response time below an acceptable threshold, in order to avoid damage to brand reputation, lost revenue and reduced productivity. Yet, modern computer services have reached a level of complexity where the human effort required to get the systems up and running is becoming prohibitively expensive. Autonomic computing has been proposed by IBM [20,30] as an approach for reducing the cost of operating complex

 0269-9648/16 $25.00

computer systems, while at the same time improving their performance and availability. Inspired by the autonomic nervous system of the human body, this approach aims at enabling computer systems to manage themselves with minimal direct human intervention.

In particular, autonomic systems are self-optimizing systems that are able to sense their environment and to tune themselves to meet end-user needs, for example, by dispatching incoming jobs to the best available resources in order to maintain and improve the requested quality of service (QoS) in response to dynamically changing workloads. An interesting example is the approach for smart workload allocation to cloud servers presented in [52,53]. Inspired by the Cognitive Packet Network [13,23] adaptive routing protocol for packet networks, this paper investigates adaptive allocation algorithms that make measurement-based fast online decisions to address QoS. Measurement data are collected by a controller which uses a Random Neural Network (RNN) based [24] decision scheme, or a greedy scheme called "sensible routing" [22] that probabilistically allocates successive tasks to hosts based on a real-time estimate of the one that can give the best QoS. It is shown that when the hosts have significantly different performance characteristics, the autonomic approach comes out clearly better with respect to static allocation schemes (e.g., Round Robin scheme) because of its use of on-line measurement data and of on-line adaptation.

Even if there is no doubt on the potential for Autonomic Computing to improve the performance and availability of individual computer services, we lack the hindsight necessary to understand the outcome of the interaction of many self-optimizing services. If multiple self-optimizing services share the resources of a computing platform, each one seeking to minimize the mean processing time of its own jobs without any coordination with the others, does not it lead to an anarchic situation in which everyone will lose? In other words, is there a significant price to pay in terms of performance for the lack of coordination of autonomic services? The present paper addresses these issues by reviewing some recent results from Game Theory which help to better understand the overall performance resulting from the interaction of uncoordinated self-optimizing services.

Game Theory is a systematic framework to model, analyze, and solve decentralized design problems involving multiple autonomous agents that interact strategically in a rational and intelligent way [4,27,42]. In particular, non-cooperative Game Theory is used to study and understand decentralized algorithms in which the autonomous agents behave "selfishly", that is each agent makes decisions so as to optimize its own performance, without coordination with the other agents. In the past decade, Game Theory has found applications in as diverse areas as load-balancing in server farms [2,5,9,10,15,18,28], power control and spectrum allocation in wireless networks [14,25,26,35,38,39,43,44,51], congestion control in the Internet [1,21,37,49,55] or decentralized routing in communication networks [4,16,29,31,36,45].

In the present article, we consider a non-cooperative routing game which was originally introduced in the seminal paper of Orda, Rom and Shimkin [45]. In this game, users generate jobs of different services, and the jobs have to be processed on one of the servers of a computing platform. Each service has its own dispatcher which probabilistically routes jobs to servers so as to minimize the mean processing cost of its own jobs. In the following, since there is no central authority for dispatching jobs to servers, this routing scheme will be referred to as a decentralized or non-cooperative load-balancing scheme (We shall use the terms load-balancing and routing interchangeably). In this load-balancing scheme, the optimal routing strategy of a dispatcher depends on the strategy of the others and the dispatchers are therefore the players of a non-cooperative routing game. We can distinguish two different settings depending on the number of dispatchers. If the number of dispatchers is finite, then it is said that the game is "atomic" and a well-known equilibrium strategy is given by the so-called Nash Equilibrium, that is, a routing strategy from which unilateral

deviation does not help any dispatcher in improving the performance perceived by the jobs it routes. When the number of dispatcher grows to infinity (every arriving job is handled by a dispatcher and it takes its own routing decision) the game is referred to as a "non-atomic" game and the corresponding equilibria is given by the notion of Wardrop Equilibrium. In this case, the equilibrium point is characterized by the fact that the performance in every (used) server is the same. In the present article we are mostly interested in the "atomic" setting, and we refer to [2,15,28,54] for some related works in the "non-atomic" setting.

The main issue we address in this paper is that of the performance of non-cooperative load-balancing schemes. We first show that there always exists a unique Nash Equilibrium, that is, a routing strategy from which no dispatcher has any incentive to deviate. We then present a result on the worst-case traffic conditions for non-cooperative routing, which states that the worst Nash Equilibrium occurs when the amount of traffic that every dispatcher routes is exactly the same. One immediate consequence is that the routing game under consideration belongs to a particular class of games known as Potential Games [41], which implies in particular that equilibrium performance can be computed as the solution of a standard convex optimization problem. We then compare the performance of the globally optimal routing strategy with that given by the Nash Equilibrium, or in other words, the performance when there is only one dispatcher which routes all the traffic so as to optimize the performance of *all* jobs, and the performance when there are several dispatchers, each one seeking to optimize the performance of its *own* jobs. In order to do so, we first look at the *Price of Anarchy* (PoA) which was introduced by Koutsoupias and Papadimitriou [34]. The PoA is a worst-case measure of the inefficiency of a non-cooperative scheme. It is defined as the ratio between the performance obtained by the worst Nash Equilibrium and the global optimal solution. We present explicit bounds on the PoA for cost functions representing the mean delay of jobs when the service discipline is PS or SRPT. These bounds indicate that as the number of dispatchers increases, the loss of efficiency may grow unboundedly, implying that the "selfish" behavior of uncoordinated self-optimizing services can lead to significant performance degradations. In practice, though, the worst-case scenario may occur rarely, if at all. We review some recent results suggesting that for the game under consideration the PoA is an overly pessimistic measure that does not reflect the performance obtained in most instances of the problem.

The rest of the paper is organized as follows. In Section 2, we present the non-cooperative load-balancing game under consideration. We then study the worst-case traffic conditions for non-cooperative routing in Section 3. In Section 4, we explain how bounds on the PoA are derived for cost functions representing the mean delay of jobs when the service discipline is PS or SRPT. Section 5 is devoted to the analysis of a new measure, called the *inefficiency*, for the comparison of the non-cooperative and centralized load-balancing schemes. Finally, some conclusions are drawn in Section 6.

2. NON-COOPERATIVE LOAD-BALANCING GAME

We consider K computer services sharing the computing resources of S servers. Users generate jobs of each service, which have to be processed on one of the servers. We assume that each service has its own job dispatcher which selects the best available server on which to process each job so as to optimize the performance of its own individual jobs. In the following, we let $\mathcal{C} = \{1, \ldots, K\}$ be the set of dispatchers and $\mathcal{S} = \{1, \ldots, S\}$ be the set of servers (see Figure 1).

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

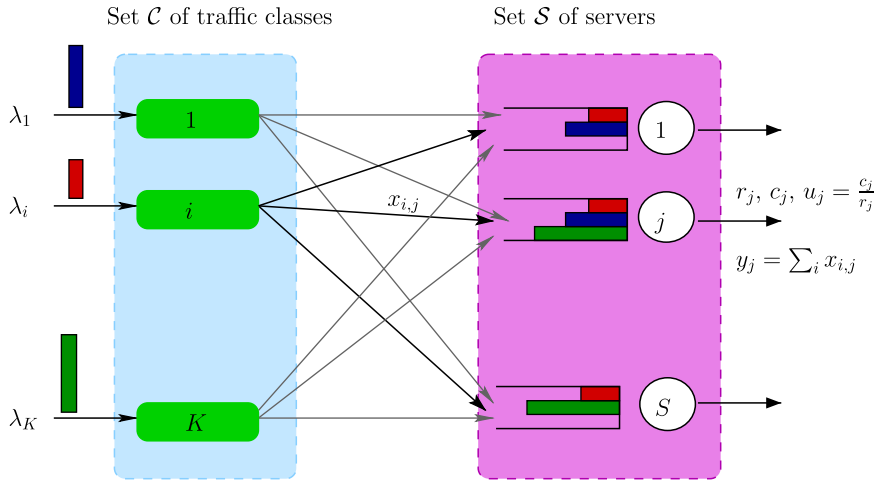

FIGURE **1.** Non-cooperative load balancing: each dispatcher controls a portion of the total traffic intensity and probabilistically routes its jobs so as to minimize their own mean processing cost.

Jobs arrive to the system according to independent Poisson processes and those received by dispatcher i are said to be jobs of class i. They have class-independent generally distributed service times. In the following, we let λ_i be the traffic intensity of class i, and $\bar{\lambda} = \sum_{i\in\mathcal{C}} \lambda_i$ be the total traffic intensity.

Server $j \in \mathcal{S}$ has capacity r_j and a holding cost c_j per job is incurred for each job sent to this server. It will be assumed throughout the paper that the total capacity of the system $\overline{r} = \sum_{n\in\mathcal{S}} r_n$ is such that $\bar{\lambda} < \overline{r}$, which is the necessary and sufficient condition to guarantee the stability of the system.

Dispatcher i uses a Bernoulli routing policy, that is, it probabilistically routes arrivals to servers so as to optimize the performance of its own jobs (Routing policies where a dispatcher has the memory of its previous routing decisions have also been considered in [6–8]). This model follows from the assumption that routing decisions are made without observing the queue lengths and that the dispatcher reacts to periodic performance measurements attained from each server with the goal of minimizing the processing cost of its own jobs. Let $\mathbf{x}_i = (x_{i,j})_{j\in\mathcal{S}}$ denote the routing strategy of dispatcher i, with $x_{i,j}$ being the amount of traffic it sends towards server j. Let

$$\mathcal{X}_i = \left\{ \mathbf{x}_i \in \mathbb{R}^S \ : \ 0 \le x_{i,j} \le r_j \ , \forall j \in \mathcal{S}; \sum_{j\in\mathcal{S}} x_{i,j} = \lambda_i \right\}$$

denote the set of feasible routing strategies for dispatcher i. Given a vector $\mathbf{x} \in \mathcal{X} = \bigotimes_{i\in\mathcal{C}} \mathcal{X}_i$, we let $y_j = \sum_{i\in\mathcal{C}} x_{i,j}$ be the total flow on server j and $\rho_j = y_j/r_j$ be the utilization rate of that server. We assume that the mean response time of server j depends only on its utilization rate and has the form $\phi(\rho_j)/r_j$, where ϕ is a strictly increasing function of the load ρ_j (see below for some examples). Note that from Little's law $\frac{x_{i,j}}{r_j}\phi(\rho_j)$ is the mean number of class-i jobs on server j, so that $c_j \frac{x_{i,j}}{r_j} \phi(\rho_j)$ represents the mean cost to be paid by dispatcher i for sending jobs to server j at rate $x_{i,j}$.

Dispatcher i seeks to minimize its total cost $T_i(\mathbf{x})$ for processing jobs, which is assumed to be the sum of the costs incurred on all the servers. This optimization problem can be

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

formulated as follows:

$$\begin{aligned} \text{minimize} \quad & T_i(\mathbf{x}) = \sum_{j\in\mathcal{S}} c_j \, \frac{x_{i,j}}{r_j} \, \phi(\rho_j), \\ \text{subject to} \quad & \mathbf{x}_i \in \mathcal{X}_i. \end{aligned} \tag{OPT-i}$$

In particular, when the holding cost is the same in every server, every dispatcher independently seeks to minimize the mean response time of its own jobs. Since there is no coordination between the dispatchers, which are in competition for the capacities of the servers, we shall refer to this routing scheme as a non-cooperative routing scheme. Note that the cost incurred by class i on server j depends not only on its amount of flow $x_{i,j}$ on that server, but also on the total amount of flow y_j through that server, which determines the server performance. Hence, the optimal routing strategy of dispatcher i depends on the routing strategies of other dispatchers, which means that the dispatchers are involved in a non-cooperative routing game. A Nash equilibrium of this routing game is a point $\mathbf{x} \in \mathcal{X}$ from which no class finds it beneficial to deviate unilaterally, that is $\mathbf{x} \in \mathcal{X}$ is a Nash equilibrium point (NEP) if and only if

$$\mathbf{x}_i = \arg\min_{\mathbf{z}\in\mathcal{X}_i} \; T_i(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{z}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_K), \quad \forall i \in \mathcal{C}.$$

The existence of a unique NEP can be established under some assumptions on the function ϕ (see Theorem 2.1 in [45]). More precisely, we shall assume that $\phi : [0, 1) \to [1, \infty)$ is a continuously differentiable, strictly increasing and convex function whose second derivative ϕ'' exists, and such that $\phi(0) = 1$ and $\lim_{\rho\to 1^-} \phi(\rho) = +\infty$. Typical examples are the $M/G/1/PS$ function $\phi(\rho) = \frac{1}{1-\rho}$, or the $M/G/1/FCFS$ function $\phi(\rho) = 1 + \rho \, \frac{1+c_b^2}{2\,(1-\rho)}$ when all job classes have the same squared coefficient of variation c_b^2 of job sizes. Another interesting example is the delay function of the $M/Pareto/1/SRPT$ queue in heavy-traffic, which is given by $\frac{1}{(1-\rho)^m}$, where m depends on the shape parameter of the Pareto distribution [15].

The main issue addressed in the present paper is that of the performance guarantees that can be obtained for non-cooperative routing schemes. To this end, we compare the performance at the Nash equilibrium with that of a globally optimal routing strategy. An optimal strategy is that of a centralized routing scheme, with a single dispatcher controlling all the traffic (*i.e.*, $K = 1$ and $\lambda_1 = \bar{\lambda}$) and routing jobs so as to optimize the performance of all jobs. An optimal routing strategy is given by $x^*_{i,j} = \frac{\lambda_i}{\bar{\lambda}} y^*_j$, where the y^*_j are the optimal solution of the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{j\in\mathcal{S}} c_j \, \rho_j \, \phi(\rho_j), \\ \text{subject to} \quad & \\ & \sum_j y_j = \bar{\lambda}, \\ & y_j \geq 0, \quad \forall j \in \mathcal{S}. \end{aligned} \tag{OPT}$$

The quantity $D(\mathbf{x}) = \sum_{j\in\mathcal{S}} c_j \, \rho_j \, \phi(\rho_j)$ represents the mean processing cost of all jobs in the routing strategy $\mathbf{x} \in \mathcal{X}$, and is known as the social cost of this routing strategy. In the following, the optimal value $D(\mathbf{x}^*)$ of the social cost will be denoted by $D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c})$ in order to make explicit its dependence on the total traffic intensity $\bar{\lambda}$ and on the vectors $\mathbf{r} = (r_j)_{j\in\mathcal{S}}$ and $\mathbf{c} = (c_j)_{j\in\mathcal{S}}$ of server capacities and costs. When there are $K > 1$ dispatchers, the social

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

cost $D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}) = D(\mathbf{x})$ at the NEP $\mathbf{x}$ corresponds to the sum of individual player costs, that is, $D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}) = \sum_{i \in \mathcal{C}} T_i(\mathbf{x})$, and it depends in addition on the traffic vector $\boldsymbol{\lambda}$, that is on the precise amount of traffic controlled by each dispatcher. In order to better understand the performance degradation resulting from the selfish behavior of the dispatchers, we shall compare the social costs of the decentralized and centralized routing schemes. We first start by studying the worst-case traffic conditions for the non-cooperative routing.

Remark 1: Apart from the performance of non-cooperative routing schemes, another important issue is related to the convergence to the Nash equilibrium. Do uncoordinated routing agents converge to a Nash equilibrium, and, if so, how long do they need? This issue is not addressed in this paper, but interested readers may refer to, for example, [3,12,17,40,45].

3. WORST-CASE TRAFFIC CONDITIONS

As noted in Section 2, the performance of the non-cooperative routing scheme depends on the precise amount of traffic controlled by each of the K dispatchers. In general, the evaluation of the social cost at the Nash equilibrium for an arbitrary traffic vector $\boldsymbol{\lambda}$ is difficult. It turns out that this evaluation becomes simpler under the worst-case traffic conditions. In this section, we present a result describing the traffic conditions under which the worst-case performance is obtained and discuss some implications of this result.

We start by comparing the optimality conditions for the decentralized and centralized settings. Let $\boldsymbol{\lambda}$ be a traffic vector and consider the associated NEP $\mathbf{x}$. Denote by $\mathbf{y}$ and $\boldsymbol{\rho}$ the vectors of offered traffics and utilization rates at that NEP, respectively. In the non-cooperative routing scheme, each and every dispatcher i minimizes its private cost $T_i(\mathbf{x})$. According to the Karush–Kuhn–Tucker (KKT) optimality conditions, this implies that at the NEP $\mathbf{x}$ each player i sends a positive amount of traffic only to those servers having a minimal *marginal private cost* for that player. Formally, it means that there exist multipliers $\mu_1, \mu_2, \ldots, \mu_K$, such that

$$\frac{\partial T_i}{\partial x_{i,j}}(\mathbf{x}) = \frac{c_j}{r_j}\left[\phi(\rho_j) + \frac{x_{i,j}}{r_j}\,\phi'(\rho_j)\right] \geq \mu_i, \quad \forall j \in \mathcal{S}, \forall i \in \mathcal{C}, \tag{1}$$

with equality if $x_{i,j} > 0$. This is in contrast to the optimality conditions for the centralized routing scheme, which states that at an optimal routing solution $\mathbf{x}^*$ only those servers having a minimal *marginal social cost* receive a positive amount of traffic. This implies that there exists a multiplier μ^* such that at point $\mathbf{x}^*$

$$\frac{\partial D}{\partial y_j}(\mathbf{x}^*) = \frac{c_j}{r_j}\left[\phi(\rho_j^*) + \frac{y_j^*}{r_j}\,\phi'(\rho_j^*)\right] \geq \mu^*, \quad \forall j \in \mathcal{S}, \tag{2}$$

with equality if $y_j^* > 0$. The latter condition is not necessarily satisfied at the NEP. In particular, it is proven in [11] that the lower the cost per unit capacity of a server, the greater is its marginal social cost at the NEP, that is

$$\frac{c_n}{r_n} \leq \frac{c_m}{r_m} \implies \frac{\partial D}{\partial y_n}(\mathbf{x}) \geq \frac{\partial D}{\partial y_m}(\mathbf{x}), \tag{3}$$

with strict inequality if $y_n > y_m$. Hence, it is possible that at the NEP two servers with different marginal social costs receive a positive amount of flow, which contradicts the optimality conditions of the social cost. Property (3) of the NEP has been used in [11] to

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

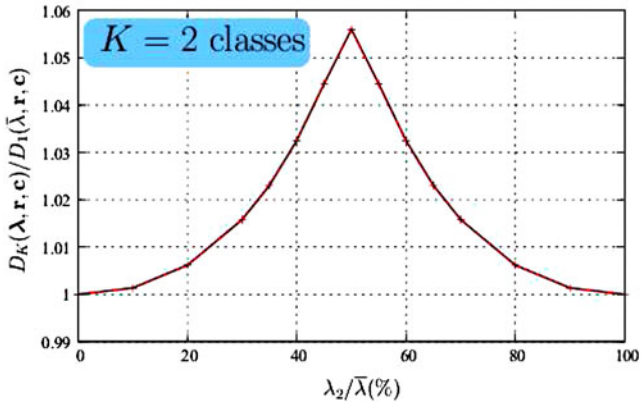

FIGURE **2.** Evolution of the social cost at the NEP in the case $K = 2$ as a function of the amount of traffic controlled by class 2.

prove that the worst-case performance is obtained when all dispatchers control the same amount of traffic.

THEOREM 1 ([11]): *Let*

$$\Lambda(\bar{\lambda}) = \left\{ \boldsymbol{\lambda} \in \mathbb{R}_+^K \ : \ \sum_{i \in \mathcal{C}} \lambda_i = \bar{\lambda} \right\},$$

be the set of all feasible traffic vectors, and let $\boldsymbol{\lambda}^{=} = \frac{\bar{\lambda}}{K}\, \boldsymbol{e}$ *be the symmetric vector, where* $\boldsymbol{e}$ *is the all-ones vector. The social cost* $D_K(\boldsymbol{\lambda}, \boldsymbol{r}, \boldsymbol{c})$ *achieves its maximum when* $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{=}$*, that is,*

$$\sup_{\boldsymbol{\lambda} \in \Lambda(\bar{\lambda})} D_K(\boldsymbol{\lambda}, \boldsymbol{r}, \boldsymbol{c}) = D_K(\boldsymbol{\lambda}^{=}, \boldsymbol{r}, \boldsymbol{c}).$$

This result is illustrated in Figure 2 in the case of $K = 2$ classes where we plot the ratio $D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c})/D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c})$ as a function of the amount of traffic λ_2 controlled by dispatcher 2. When λ_2 is 0 or $\bar{\lambda}$, the ratio of social costs is 1, implying that the non-cooperative routing scheme is optimal. The worst performance of this routing scheme is achieved for the symmetric game, that is when $\lambda_2 = \lambda_1 = \frac{\bar{\lambda}}{2}$.

The key idea of the proof of Theorem 1 is to show that, starting from an arbitrary traffic vector $\boldsymbol{\lambda}$, the symmetric traffic vector $\boldsymbol{\lambda}^{=}$ can be reached in a finite number of steps R by a sequence $\{\boldsymbol{\lambda}^n\}_{n \geq 0}$ such that $\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}$ and $D_K(\boldsymbol{\lambda}^{n+1}) \geq D_K(\boldsymbol{\lambda}^n)$. Such a sequence is obtained by considering a certain transformation $\boldsymbol{\lambda} \to \hat{\boldsymbol{\lambda}}$ of the traffic vector, which amounts to transferring traffic from the most loaded dispatchers to the least loaded ones, thus reducing the heterogeneity of the traffic vector. It is shown in [11] that the load on the most attractive servers (those with the smaller cost per unit capacity) cannot decrease under this transformation. However, since, according to (3), those servers are precisely those with the greatest marginal social cost, the convexity of the social cost $D_K()$ implies that it cannot be reduced under the transformation, so that $D_K(\boldsymbol{\lambda}, \mathbf{r}, \mathbf{c}) = D_K(\boldsymbol{\lambda}^0, \mathbf{r}, \mathbf{c}) \leq D_K(\boldsymbol{\lambda}^1, \mathbf{r}, \mathbf{c}) \leq \cdots \leq D_K(\boldsymbol{\lambda}^R, \mathbf{r}, \mathbf{c}) = D_K(\boldsymbol{\lambda}^{=}, \mathbf{r}, \mathbf{c})$.

An important consequence of Theorem 1 is that, for the worst-case analysis of non-cooperative routing, we can restrict ourselves to the symmetric game. It is well known that in this case the non-cooperative routing game is a potential game [41] (see e.g. Theorem

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

4.1 in [16]). In other words, although each and every dispatcher independently optimizes its own cost function, they collectively solve a standard convex optimization problem. This is formally stated in Proposition 1.

PROPOSITION 1 ([11]): *If the vector* $\boldsymbol{\rho}$ *is a global optimum of the following convex optimization problem*

$$\begin{aligned} \underset{\boldsymbol{\rho}}{\text{minimize}} \quad & \sum_{j\in\mathcal{S}} c_j \rho_j \phi(\rho_j) + (K-1) \int_0^{\rho_j} c_j\, \phi(z) dz \\ \text{s.t.} \quad & \\ & \textstyle\sum_{j\in\mathcal{S}} r_j \rho_j = \bar{\lambda}, \\ & 0 \le \rho_j < 1, \quad \forall j \in \mathcal{S}, \end{aligned}$$

then the routing strategy $\boldsymbol{x}$ *such that* $x_{i,j} = r_j \frac{\rho_j}{K}$, $\forall i \in \mathcal{C}$, $\forall j \in \mathcal{S}$, *is the NEP of the symmetric game.*

Note that when $K = 1$, the above problem reduces to the global optimization problem solved by the centralized scheme. When $K \to \infty$, the equivalent problem states the common function that is jointly optimized by an infinite number of players and is characteristic of the Wardrop equilibrium. As we shall see in the following, the fact that the worst-case analysis of the non-cooperative routing scheme reduces to the analysis of a convex optimization problem considerably simplifies the comparison with the centralized routing scheme.

4. WORST-CASE PERFORMANCE ANALYSIS

In this section, we compare the performance of the global optimum with that given by the Nash equilibrium, or in other words, the performance when there is only one dispatcher which routes all the traffic, and the performance when there are several dispatchers each one seeking to optimize its own performance. In order to do so we look at the PoA which was introduced by Koutsoupias and Papadimitriou [34]. The PoA is an oft-used measure of the inefficiency of a decentralized scheme, which for our model is defined as

$$\text{PoA}(K) = \sup_{\boldsymbol{\lambda},\mathbf{r},\mathbf{c}} \frac{D_K(\boldsymbol{\lambda},\mathbf{r},\mathbf{c})}{D_1(\bar{\lambda},\mathbf{r},\mathbf{c})}.$$

Note that the PoA lies in the interval $[1,\infty)$. We have seen in Theorem 1 that the worst-case performance is obtained when all dispatchers control the same amount of traffic. Therefore,

$$\text{PoA}(K) = \sup_{\mathbf{r},\mathbf{c}} \frac{D_K(\boldsymbol{\lambda}^{=},\mathbf{r},\mathbf{c})}{D_1(\bar{\lambda},\mathbf{r},\mathbf{c})}.$$

Several recent works have shown that non-cooperative load-balancing can be very inefficient in the presence of non-linear delay functions, see, for example, [9,10,15,28]. The PoA has been analyzed both in the so-called non-atomic scenario where every arriving job can select the server in which it will be served, and in the atomic scenario considered in this paper where each player controls a non-negligible amount of flow. For the non-atomic scenario, Haviv and Roughgarden have shown in [28] that the PoA corresponds to the number of servers, implying that, in a server farm with S servers, the mean response time of jobs can be as high as S times the optimal one! For the atomic scenario, we present below bounds on the PoA obtained in the case of $M/G/1/PS$ queues [9], and in the case of

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

$M/Pareto/1/SRPT$ queues [11], which prove that the PoA can grows unboundedly with the number of dispatchers K. The fact that the Nash equilibrium can be very inefficient has paved the way to a lot of research on mechanism design that aims at architecting Nash equilibria so that they are efficient with respect to the centralized setting [32,33,46].

4.1. Bounds on the PoA for $M/G/1/PS$ queues

Let us first assume that the servers on which jobs are processed are modeled as $M/G/1/PS$ queues, in which case we have $\phi(\rho) = 1/(1-\rho)$. Note that since we have assumed Poisson arrivals of jobs at the dispatchers and Bernoulli routing, the assumption of Poisson arrivals at the servers is consistent. The following lower and upper bounds on the PoA have been established in [9].

THEOREM 2 ([9]): *For a system with two or more servers,*

$$\frac{K}{2\sqrt{K}-1} \leq PoA(K) \leq \sqrt{K}.$$

This result states that the PoA is of the order of $\sqrt{K}$ independently of the number of servers. In other words, in the worst-case scenario, the performance degradation with K self-optimizing services can be of order $\sqrt{K}$.

The proof of the upper bound is based on the observation that for $\phi(\rho) = 1/(1-\rho)$ the potential function of Proposition 1 takes the simple form,

$$\sum_{j\in\mathcal{S}} \frac{1}{K}\left[\frac{\rho_j}{1-\rho_j} + (K-1)\log\left(\frac{1}{1-\rho_j}\right)\right],$$

from which an explicit solution of the symmetric game can be obtained. This explicit solution is used in [9] to establish that:

- The distributed scheme with K dispatchers uses only a subset of the servers used by the centralized scheme. In other words, if $\mathcal{S}^*(K)$ denotes the set of servers used at the NEP of the symmetric game with K dispatchers, we have $\mathcal{S}^*(K) \subset \mathcal{S}^*(1)$.
- If $y_j(K)$ represents the offered traffic in equilibrium in the K player symmetric game, then

$$\frac{y_j(K)}{r_j - y_j(K)} \leq \sqrt{K}\frac{y_j(1)}{r_j - y_j(1)}, \quad \forall j \in \mathcal{S}^*(1).$$

The upper bound on the PoA is then obtained as follows:

$$D_K(\boldsymbol{\lambda}^{=}, \mathbf{r}, \mathbf{c}) = \sum_{j\in\mathcal{S}^*(K)} c_j \frac{y_j(K)}{r_j - y_j(K)} \leq \sum_{j\in\mathcal{S}^*(1)} c_j \frac{y_j(K)}{r_j - y_j(K)} \leq \sqrt{K} D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c}).$$

The lower bound is established in [9] by exhibiting an example of a symmetric game for which $D_K(\boldsymbol{\lambda}^{=}, \mathbf{r}, \mathbf{c}) \geq K/(2\sqrt{K}-1) D_1(\bar{\lambda}, \mathbf{r}, \mathbf{c})$. The procedure for obtaining this example is as follows: using the KKT conditions, define the parameters $\bar{\lambda}$, $\mathbf{r}$ and $\mathbf{c}$ such that for $K > 1$ only the least costly server is used, whereas for $K = 1$ more than one server is used. For such a symmetric game, by splitting the traffic over several servers, the centralized setting does much better than the decentralized one.

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

4.2. Bounds on the PoA for $M/Pareto/1/SRPT$ queues

Let us now assume that job sizes follow a class-independent power law probability distribution, and that the servers use the shortest-remaining-processing-time (SRPT) scheduling discipline [48]. This scheduling discipline, which preemptively runs the job with shortest remaining processing requirement, is known to be optimal with respect to mean response time [47,50]. Although no simple closed form formula is known for the mean response time of $M/Pareto/1/SRPT$ queues, it is shown in [15] that in heavy-traffic conditions the mean delay has the form $\phi(\rho) = 1/(1-\rho)^m$, where m depends on the shape parameter of the Pareto distribution.

One of the main difficulty in the analysis of the PoA in this case is that, in contrast to the case of $M/G/1/PS$ queues, a closed-form solution of the optimization problem stated in Proposition 1 cannot be obtained. This turns out to be a major obstacle for the derivation of an upper bound on the PoA. Nevertheless, a lower bound can be obtained by following the same procedure as that used for $M/G/1/PS$ queues.

Proposition 2 ([11]): *For* $\phi(\rho) = \frac{1}{(1-\rho)^m}$,

$$PoA(K) \geq \frac{K}{(1+m)K^{1/(m+1)} - m}. \tag{4}$$

Note that this lower bound on the PoA is independent of $\mathbf{r}$ and $\mathbf{c}$. It proves that, for $M/Pareto/1/SRPT$ queues, the PoA grows at least as fast as $O(K^{\frac{m}{m+1}})$ as K grows. Hence, as the number of dispatchers $K \to \infty$, the performance degradation with respect to the centralized setting tends to infinity. Although we do not have an upper bound for PoA as in the $M/G/1/PS$ case, it is conjectured in [11] that the lower bounds constructed using the above procedure give the right order of the PoA, just as was proved for the case of $M/G/1/PS$ delay functions.

5. IS THE POA THE RIGHT MEASURE OF INEFFICIENCY?

In Section 4, the lower bounds on the PoA have been established by computing the system parameters in such a way that in the decentralized setting only the least costly server is used, whereas more than one server is used in the centralized setting. Similarly, in the non-atomic scenario considered in [28], the worst-case architecture has one server whose capacity is much larger (tending to infinity) compared to that of the other servers. It is doubtful that such asymmetries will occur in data-centers where processors are more than likely to have similar characteristics. This suggests that the worst-case analysis of the inefficiency of selfish routing is overly pessimistic and that high PoAs are obtained in pathological instances that hardly occur in practice.

In [19], the authors adopt this point of view. Assuming that the holding cost is the same in every server (that is, $\mathbf{c} = \mathbf{e}$), which is equivalent to assuming that each dispatcher seeks to minimize the mean processing time of its own jobs, they propose a new measure, called the *inefficiency*, to compare the non-cooperative routing scheme with K dispatchers and S servers and the centralized one. This new measure is defined as follows

$$I_K^S(\mathbf{r}) = \sup_{\boldsymbol{\lambda} \in \Lambda(\bar{\lambda}), \bar{\lambda} < \overline{r}} \frac{D_K(\boldsymbol{\lambda}, \mathbf{r})}{D_1(\bar{\lambda}, \mathbf{r})}. \tag{5}$$

The rationale for this definition is that in practice the system administrator controls neither the total incoming traffic nor how it is split between the dispatchers, whereas the

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

number of servers and their capacities are fixed. Therefore, it makes sense to consider a *fixed data-center architecture* under the worst traffic conditions for the *inefficiency* of selfish routing (provided the system is stable). As is true of the PoA, *inefficiency* can take values between 1 and ∞. A higher value of *inefficiency* indicates a worse performance of selfish routing compared to centralized routing. As opposed to the PoA, the *inefficiency* depends on the parameters (the server speeds and the number of servers in our case) of the architecture. By calculating the worst possible *inefficiency*, one retrieves the PoA, that is, $\mathrm{PoA}(K,S) = \sup_{\mathbf{r}} I_K^S(\mathbf{r})$.

It follows from Theorem 1 that

$$I_K^S(\mathbf{r}) = \sup_{\bar{\lambda}<\bar{r}} \frac{D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})}{D_1(\bar{\lambda},\mathbf{r})}, \tag{6}$$

so that, as for the PoA, the *inefficiency* depends only on the total traffic intensity and not on individual traffic flows to each of the dispatcher.

The setting considered in [19] is similar to the one introduced in Section 2, but the authors restrict themselves to the case of two classes of servers, which are modeled as $M/G/1/PS$ queues: there are S_1 "fast" servers of capacity r_1, and $S_2 = S - S_1$ "slow" servers, each one of capacity $r_2 < r_1$. Let $\beta = \frac{r_1}{r_2} \geq 1$ be the ratio of server capacities, $\alpha = \frac{S_1}{S_2} > 0$ be the ratio of the numbers of servers of each type, and define

$$\bar{\lambda}^{OPT} = S_1 r_1 \left(1 - \frac{1}{\sqrt{\beta}}\right), \tag{7}$$

and

$$\bar{\lambda}^{NE} = S_1 r_1 \left(1 - \frac{2}{\sqrt{(K-1)^2 + 4K\beta} - (K-1)}\right) > \bar{\lambda}^{OPT}. \tag{8}$$

The following lemma gives the conditions on $\bar{\lambda}$ under which the centralized setting and the decentralized one use only the fast class of servers, or both classes, and describes the evolution of the ratio $D_K(\frac{\bar{\lambda}}{K}\mathbf{e},\mathbf{r})/D_1(\bar{\lambda},\mathbf{r})$.

LEMMA 1 ([19]):

1. *If $\bar{\lambda} \leq \bar{\lambda}^{OPT}$, both settings use only the "fast" servers, and the ratio of social costs is equal to 1,*
2. *if $\bar{\lambda}^{OPT} \leq \bar{\lambda} \leq \bar{\lambda}^{NE}$, the decentralized setting uses only the "fast" servers, while the centralized one uses all servers, and the ratio of social costs is strictly increasing,*
3. *if $\bar{\lambda}^{NE} < \bar{\lambda} < \bar{r}$, both settings use all servers, and the ratio of social costs is strictly decreasing.*

This behavior of the ratio of social costs is illustrated for $K = 2$ and $K = 5$ in Figure 3 in the case of a server farm with $S_1 = 100$ fast servers of capacity $r_1 = 100$, and $S_2 = 300$ slow servers of capacity $r_2 = 10$.

Since the ratio $D_K(\bar{\lambda},\mathbf{r})/D_1(\bar{\lambda},\mathbf{r})$ is a continuous function of $\bar{\lambda}$ over the interval $[0,\bar{r})$, a direct consequence of Lemma 1 is the following Theorem.

THEOREM 3 ([19]): *The inefficiency is worst when the total arriving traffic intensity equals $\bar{\lambda}^{NE}$, namely,*

$$I_K^S(\boldsymbol{r}) = \frac{D_K(\frac{\bar{\lambda}^{NE}}{K}\boldsymbol{e},\boldsymbol{r})}{D_1(\bar{\lambda}^{NE},\boldsymbol{r})}, \tag{9}$$

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

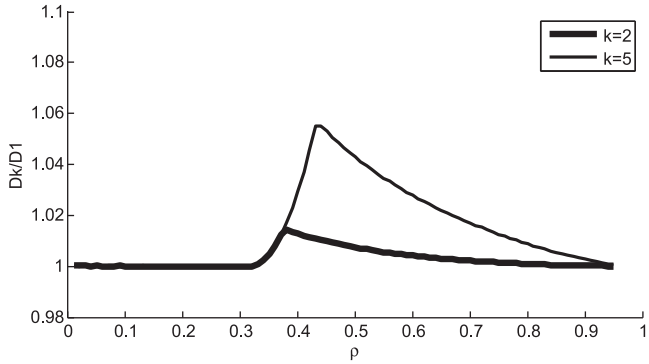

FIGURE **3.** Evolution of the ratio of social costs for $K = 2$ and $K = 5$ as the utilization rate ranges from 0 to 100%.

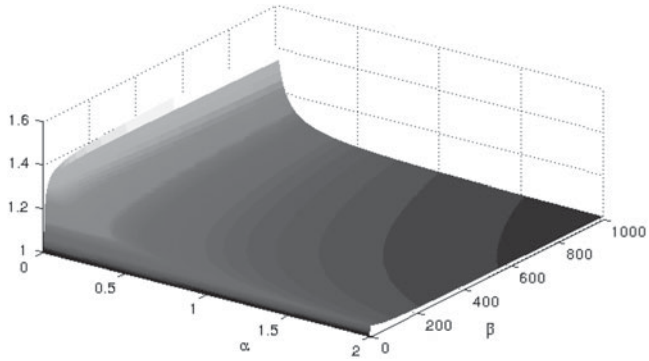

FIGURE **4.** Evolution of the *inefficiency* as a function of α and β for $K = 5$ dispatchers and $S = 1000$ servers.

Theorem 3 fully characterizes the worst case traffic conditions for a server farm with two classes of servers. It states that the worst inefficiency of the decentralized setting is achieved when (a) each dispatcher controls the same amount of traffic and (b) the total traffic intensity is such that the decentralized setting only starts using the slow servers.

The analysis of the symmetric game obtained for $\bar{\lambda} = \bar{\lambda}^{NE}$ can be done with Proposition 1. This allows to obtain an explicit expression for the *inefficiency* of selfish routing for data-centers with two classes of PS servers [19]:

$$I_K^S(\mathbf{r}) = \frac{1}{2} \frac{\sqrt{(K-1)^2 + 4K\beta} - (K+1)}{\frac{(\frac{1}{\alpha}+\sqrt{\beta})^2}{\frac{1}{\alpha}+\frac{2\beta}{\sqrt{(K-1)^2+4K\beta}-(K-1)}}} - (\frac{1}{\alpha}+1)}, \tag{10}$$

where we recall that $\beta = \frac{r_1}{r_2} \geq 1$ and $\alpha = \frac{S_1}{S_2} > 0$. Note that the *inefficiency* $I_K^S(\mathbf{r})$ does not depend on the total number of servers S, but only on the ratio of server capacities and on the ratio of the numbers of servers of each type. In Figure 4, we plot the *inefficiency* $I_K^S(\mathbf{r})$ of the non-cooperative routing scheme with $K = 5$ dispatchers and $S = 1000$ servers as the parameters α and β change from $\frac{1}{S-1}$ to 2 and from 1 to 1,000, respectively. It can be observed that even for unbalanced scenarios (α small and β large), the *inefficiency* is always fairly close to 1, indicating that, even in the worst case traffic conditions, the gap between the NEP and the optimal routing solution is not significant.

https://doi.org/10.1017/S0269964816000097 Published online by Cambridge University Press

6. CONCLUSION

Self-optimizing services are able to use on-line measurements to dispatch incoming jobs to the best available computing resources in order to maintain and improve the QoS in response to dynamically changing workloads. The interaction between uncoordinated dispatchers can however result in significant performance degradations. We have seen that the worst-case traffic conditions occur when all dispatchers control the same amount of traffic. We have also presented explicit bounds on the PoA for cost functions representing the mean delay of jobs when the service discipline is PS or SRPT. These bounds indicate that as the number of dispatchers increases, the loss of efficiency may grow unboundedly, implying that the "selfish" behavior of uncoordinated self-optimizing services can lead to significant performance degradations. It is nevertheless important to keep in mind that these bounds are obtained for worst-case scenarios, which do not necessarily occur in practice. In data-centers where processors are more than likely to have similar characteristics, no significant performance degradation is observed with respect to a globally optimal routing strategy. However, more work is needed to confirm or contradict this observation.

References

1. Akella, A., Seshan, S., Karp, R., Shenker, S., & Papadimitriou, C. (2002). Selfish behavior and stability of the internet: a game-theoretic analysis of TCP. In *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pittsburgh, Pennsylvania, USA, 19–23 August 2002.
2. Altman, E., Ayesta, U., & Prabhu, B.J. (2011). Load balancing in processor sharing systems. *Telecommunication Systems* 47(1–2): 35–48.
3. Altman, E., Basar, T., Jiménez, T., & Shimkin, N. (2001). Routing into two parallel links: Game-theoretic distributed algorithms. *Journal of Parallel and Distributed Computing* 61(9): 1367–1381.
4. Altman, E., Boulogne, T., Azouzi, R.E., Jimenez, T., & Wynter, L. (2006). A survey on networking games in telecommunications. *Computers and Operations Research* 33(2): 286–311.
5. Anselmi, J. & Gaujal, B. (2010). Optimal routing in parallel, non-observable queues and the price of anarchy revisited. In *22nd International Teletraffic Congress (ITC)*, Amsterdam.
6. Anselmi, J. & Gaujal, B. (2010). The price of anarchy in parallel queues revisited. In *ACM Sigmetrics*, New-York, USA, pp. 353–354.
7. Anselmi, J. & Gaujal, B. (2011). The price of forgetting in parallel and non-observable queues. *Performance Evaluation* 68(12): 1291–1311.
8. Anselmi, J., Gaujal, B., & Nesti, T. (2015). Control of parallel non-observable queues: asymptotic equivalence and optimality of periodic policies. *Stochastic Systems* 5(1): 120–145.
9. Ayesta, U., Brun, O., & Prabhu, B.J. (2011). Price of anarchy in non-cooperative load balancing games. *Performance Evaluation* 68: 1312–1332.
10. Bell, C.H. & Stidham, S. (1983). Individual versus social optimization in the allocation of customers to alternative servers. *Management Science* 29: 831–839.
11. Brun, O. & Prabhu, B. (2014). Worst-case analysis of non-cooperative load balancing. *Annals of Operations Research* 1–25. http://dx.doi.org/10.1007/s10479-014-1747-7.
12. Brun, O., Prabhu, B., & Seregina, T. (2013). On the convergence of the best-response algorithm in routing games. In *Proceedings of the Seventh International Conference on Performance Evaluation Methodologies and Tools (ValueTools'13)* pp. 136–144.
13. Brun, O., Wang, L., & Gelenbe, E. (2016). Data driven self-managing routing in intercontinental overlay networks. *Accepted for publication in the IEEE Journal on Selected Areas in Communications*, p. 9.
14. Charilas, D.E. & Panagopoulos, A.D. (2010). A survey on game theory applications in wireless networks. *Computer Networks* 54(18): 3421–3430.
15. Chen, H.-L., Marden, J.R., & Wierman, A. (2008). The effect of local scheduling in load balancing designs. *SIGMETRICS Performance Evaluation Review* 36(2):110–112.
16. Cominetti, R., Correa, J.R., & Stier-Moses, N.E. (2009). The impact of oligopolistic competition in networks. *Operations Research* 57(6): 1421–1437.

17. Coucheney, P., Gaujal, B., & Mertikopoulos, P. (2014). Penalty-regulated dynamics and robust learning procedures in games. Technical Report, Inria Grenoble - Rhône-Alpes and LIG (Laboratoire d'Informatique de Grenoble), April 2014. https://hal.inria.fr/hal-01073497
18. Czumaj, A., Krysta, P., & Vocking, B. (2010). Selfish traffic allocation for server farms. *SIAM Journal on Computing* 39(5): 1957–1987.
19. Doncel, J., Ayesta, U., Brun, O., & Prabhu, B. (2014). Is the price of anarchy the right measure for load-balancing games? *ACM Transactions on Internet Technology (TOIT)* 14(2–3): 18.1–18.20.
20. Ganek, A.G. & Corbi, T.A. (2003). The dawning of the autonomic computing era. *IBM Systems Journal* 42(1): 5–18.
21. Garg, R., Kamra, A., & Khurana, V. (2002). A game-theoretic approach towards congestion control in communication networks. *ACM SIGCOMM Computer Communication Review* 32(3): 47–61.
22. Gelenbe, E. (2003). Sensible decisions based on QoS. *Computational Management Science* 1(1): 1–14.
23. Gelenbe, E., Lent, R., & Nunez, A. (2004). Self-aware networks and QoS. *Proceedings of the IEEE* 92(9): 1478–1489.
24. Gelenbe, E. & Timotheou, S. (2008). Random neural networks with synchronized interactions. *Neural Computation* 20(9): 2308–2324.
25. Gunturi, S., Paganini, F., T. Instruments, and I. Bangalore. (2003). Game theoretic approach to power control in cellular CDMA. In *Vehicular Technology Conference. VTC 2003-Fall*.
26. Han, Z., Ji, Z., & Liu, K. (2007). Non-cooperative resource competition game by virtual referee in multi-cell OFDMA networks. *IEEE Journal on Selected Areas in Communications* 25(6): 1079–1090.
27. Hassin, R. & Haviv, M. (2003). *To queue or not to queue- equilibrium behavior in queueing systems*. Springer US.
28. Haviv, M. & Roughgarden, T. (2007). The price of anarchy in an exponential multi-server. *Operations Research Letters* 35: 421–426.
29. Keidar, I., Melamed, R., & Orda, A. (2009). Equicast: Scalable multicast with selfish users. *Computer Networks* 53(13): 2373–2386.
30. Kephart J.O. & Chess, D.M. (2003). The vision of autonomic computing. *Computer* 36(1): 41–50.
31. Korilis, Y., Lazar, A., & Orda, A. (1997). Capacity allocation under noncooperative routing. *IEEE Transactions on Automatic Control* 42(3): 309–325.
32. Korilis, Y., Lazar, A., & Orda, A. (2006). Architecting noncooperative networks. *IEEE Journal on Selected Areas in Communications* 13(7): 1241–1251.
33. Korilis, Y.A., Lazar, A.A., & Orda, A. (1997). Achieving network optima using stackelberg routing strategies. *IEEE/ACM Transactions on Networking* 5(1): 161–173.
34. Koutsoupias, E. & Papadimitriou, C.H. (1999). Worst-case equilibria. In *STACS 1999*.
35. Leshem, A. & Zehavi, E. (2008). Cooperative game theory and the Gaussian interference channel. *IEEE Journal on Selected Areas in Communications* 26: 1078–1088.
36. Libman, L. & Orda, A. (1999). The designer's perspective to atomic noncooperative networks. *IEEE/ACM Transactions on Networking (TON)* 7(6): 875–884.
37. López, L., del Rey Almansa, G., Paquelet, S., & Fernández, A. (2005). A mathematical model for the TCP tragedy of the commons. *Theoretical Computer Science* 343(1–2): 4–26.
38. MacKenzie, A.B. & Wicker, S.B. (2001). Game theory in communications: motivation, explanation and application to power control. In *IEEE Global Telecommunications Conference*, San Antonio, Texas, USA, 25–29 November 2001, pp. 821–826.
39. Menon, R., MacKenzie, A., Hicks, J., Buehrer, R., & Reed, J. (2009). A game-theoretic framework for interference avoidance. *IEEE Transactions on Communications* 57(4): 1087–1098.
40. Mertzios, G. (2009). Fast convergence of routing games with splittable flows. In *Proceedings of the 2nd International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS)*, Orlando, FL, USA, July, pp. 28–33.
41. Monderer, D. & Shapley, L.S. (1996). Potential games. *Games and Econ. Behavior* 14: 124–143.
42. Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. (eds.) (2007). *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press.
43. Niyato, D. & Hossain, E. (2008). Competitive pricing for spectrum sharing in cognitive radio networks: dynamic game, inefficiency of NASH equilibrium, and collusion. *IEEE Journal on Selected Areas of Communications* 26(1): 192–202.
44. Niyato, D. & Hossain, E. (2008). Competitive spectrum sharing in cognitive radio networks: a dynamic game approach. *IEEE Transactions on Wireless Communications* 7(7): 88–94.
45. Orda, A., Rom, R., & Shimkin, N. (1993). Competitive routing in multi-user communication networks. *IEEE/ACM Transactions on Networking* 1: 510–521.

46. Roughgarden, T. (2005). *Selfish routing and the price of anarchy*. Cambridge, MA, USA: MIT Press.
47. Schrage, L.E. (1968). A proof of the optimality of the shortest remaining processing time discipline. *Operations Research* 16: 678–690.
48. Schrage, L.E. & Miller, L.W. (1966). The queue M/G/1 with the shortest remaining processing time discipline. *Operations Research* 14: 670–684.
49. Shenker, S.J. (1995). Making greed work in networks: a game-theoretic analysis of switch service disciplines. *IEEE/ACM Transactions on Networking (TON)* 3(6): 819–831.
50. Smith, D. (1976). A new proof of the optimality of the shortest remaining processing time discipline. *Operations Research* 26: 197–199.
51. Suris, J., DaSilva, L., Han, Z., & MacKenzie, A. (2007). Cooperative game theory for distributed spectrum sharing. In *IEEE International Conference on Communications*, Glasgow, Scotland, 24–28 June 2007, pp. 5282–5287.
52. Wang, L. & Gelenbe, E. (2015). Adaptive dispatching of tasks in the cloud. *IEEE Transactions on Cloud Computing* PP(99): 1.
53. Wang, L. & Gelenbe, E. (2015). Experiments with smart workload allocation to cloud servers. In *IEEE Fourth Symposium on Network Cloud Computing and Applications*, Munich, Germany, pp. 31–35, doi:10.1109/NCCA.2015.15.
54. Wu, T. & Starobinski, D. (2008). A comparative analysis of server selection in content replication networks. *IEEE/ACM Trans. Netw.* 16(6): 1461–1474.
55. Yaïche, H., Mazumdar, R.R., & Rosenberg, C. (2000). A game theoretic framework for bandwidth allocation and pricing in broadband networks. *IEEE/ACM Transactions on Networking (TON)* 8(5): 667–678.