Load Balancing in Processor Sharing Systems

Eitan Altman
INRIA Sophia Antipolis
2004, route des Lucioles
06902 Sophia Antipolis,
France
altman@sophia.inria.fr

Urtzi Ayesta
LAAS-CNRS
Université de Toulouse
7, Avenue Colonel Roche
F-31077 Toulouse, France
urtzi@laas.fr

Balakrishna Prabhu
LAAS-CNRS
Université de Toulouse
7, Avenue Colonel Roche
F-31077 Toulouse, France
bjprabhu@laas.fr

ABSTRACT

In this paper, we investigate optimal load balancing strategies for a multi-class multi-server processor-sharing system with a Poisson input stream, heterogeneous service rates, and a server-dependent holding cost per unit time. Specifically, we study (i) the centralized setting in which a dispatcher routes incoming jobs based on their service time requirements so as to minimize the weighted mean sojourn time in the system; and (ii) the decentralized, distributed non-cooperative setting in which each job, aware of its service time, selects a server with the objective of minimizing its weighted mean sojourn time in the system. For the decentralized setting we show the existence of a potential function, which allows us to transform the non-cooperative game into a standard convex optimization problem.

For the two aforementioned settings, we characterize the set of optimal routing policies and obtain a closed form expression for the load on each server under any such policy. Furthermore, we show the existence of an optimal policy that routes a job independently of its service time requirement. We also show that the set of servers used in the decentralized setting is a subset of set of servers used in the centralized setting. Finally, we compare the performance perceived by jobs in the two settings by studying the so-called Price of Anarchy (PoA), that is, the ratio between the decentralized and the optimal centralized solutions. When the holding cost per unit time is the same for all servers, it is known that the PoA is upper bounded by the number of servers in the system. Interestingly, we show that the PoA for our system can be unbounded. In particular this indicates that in our system, the performance of selfish routing can be extremely inefficient.

Keywords

Load balancing, M/G/1 processor-sharing queues, server farms, potential game, Price of Anarchy

1. INTRODUCTION

Communication services such as web server-farms, database systems and grid computing clusters, routinely employ multi-server systems to provide a range of services to their customers. An important issue in such systems is to determine the server to which an incoming request should be routed to in order to optimize a given performance criterion. From the service provider's perspective, this choice of the strategy (centralized or decentralized) and the service discipline (Processor Sharing (PS), First-Come-First-Served (FCFS), etc.) determines the amount of resources it needs to deploy in order to guarantee a certain Quality-of-Service (QoS) to its customers. Thus, an investigation of load balancing or routing strategies in multi-server systems can give guidelines to the service provider on dimensioning its system.

In this paper we study the optimal load balancing in a multi-server processor-sharing system with heterogeneous service capacities. This configuration is also known as processor-sharing server-farms, and is a popular architecture in computing centers, used for example in the Cisco Local Director, IBM Network Dispatcher and Microsoft Sharepoint (see [5] for a recent survey). This configuration can also be used to model a web server farm, where requests for files (or HTTP pages) arrive to a dispatcher are dispatched immediately to one of the servers in the farm for processing. With each server, we associate a service capacity (i.e., some servers could be faster than the others) and a holding cost per unit time. We assume that requests arrive as a Poisson process, and that the service requirement of each request is sampled from a finite set. For such a multi-server system, we investigate load balancing in two different settings: *(i)* the centralized setting in which a dispatcher assigns the server to an incoming request with the objective of minimizing the weighted mean sojourn time of jobs in the system, and *(ii)* the distributed non-cooperative setting in which an incoming request selects a server in order to minimize its own weighted mean sojourn time in the system. In both cases we assume that the only information available to the decision maker (the dispatcher or the request itself) is the service time requirement of the request. This might be the case, for example, in situations where not all the servers are in the same location and it may be costly to gather information on the current queue lengths at the various servers.

The main contributions of the present work are as follows. For both settings, we characterize the set of optimal routing policies, and give closed-form expressions for the load on each server under any optimal policy. It is worthwhile

to note that for the distributed non-cooperative setting this is done by showing the existence of a potential function, which allows us to transform the non-cooperative game into a standard convex optimization problem. We then give an optimal policy in which an incoming request is routed to a server with a probability that is independent of the service requirement of the request. This property of the PS discipline could be useful in systems in which the service requirement of requests is not known *a priori* and it illustrates an important difference between the optimal load balancing policy in a PS server-farm and FCFS server-farm, since in the case of a FCFS server-farm it has been shown that the optimal load balancing does use information on the service requirement of each request [9, 7]. Further, we show that higher the ratio of the holding cost per unit time to the service capacity of a server the lighter is the load on it, thus defining an index to order the servers. For certain input parameters (i.e., an arrival process, service time distribution, available service capacities, holding cost per unit time), it is thus possible that some of the servers will not be processing any requests. We show that the set of servers processing requests in the decentralized setting is a subset of that in the centralized setting. Thus, there is a trade-off in the performance gains and cost of servers to be considered when choosing between the two settings. We also note that, given the input parameters, this analysis gives the set of servers that a service provider should choose in order to minimize the mean sojourn time in its system. Finally, we compare the performance perceived by jobs in the two settings by studying the so-called Price of Anarchy (PoA), that is, the ratio between the selfish decentralized and the optimal centralized solutions. When the holding cost per unit of time is the same in every server it is has been shown that the PoA is upper bounded by the number of servers in the system, see for example [21, 10]. Interestingly, we show that for our system the PoA is unbounded, that is, it can be arbitrarily close to infinity. This indicates that unequal holding costs may have a profound impact on the system's performance. In particular, the performance of selfish routing can be unboundedly worse than the performance obtained by a centralized routing.

1.1 Related work

Load balancing in multi-server systems has been previously investigated not only in the context of communications services but also in the broader context of queueing systems. Global and Individual optimality in load balancing are considered in the monograph [12], which does not consider decisions based on knowledge of the amount of load. Systems with general service time distribution and FCFS scheduling discipline were studied in [6, 2, 3, 7], while [16, 10] studied systems with exponential service time distributions and arbitrary scheduling discipline. In [8] the authors analysed a multi-servers PS system where requests join the server that has the smallest number of requests. Our work is closely related to [21] and [10]. The main differences are that *(i)* we consider a multi-class job arrival process, allowing the dispatcher to use information on the size of the requests and *(ii)* the addition of a heterogeneous holding cost per time unit in each server. As we will see, both *(i)* and *(ii)* generalizations allow us to draw important conclusions, that to the best of our knowledge were not known before.

By considering a multi-class system, we wish to analyze how the information on the service requirements of users impacts the structure of the optimal load balancing. Our results show that the structure of the optimal routing in a system with the PS scheduling discipline is radically different with respect to the FCFS case. For a multi-server FCFS system with homogeneous service capacities it was conjectured in [9], and proved in [7], that the optimal load balancing scheme consists in assigning to each server all jobs whose processing times fall within non-overlapping, continuous intervals of processing times. The intuitive explanation to this result comes from the fact that this strategy reduces the variability of service times for each queue. Since the mean delay in a FCFS queue is directly proportional to the variability of the service time distribution (Pollaczek-Khinchin formula), an interval-based policy can minimize the overall mean delay in the system. Interestingly, if the service capacities are heterogeneous an interval-based strategy need not be optimal [7]. In contrast, we show that in the case of a multi-server PS system the optimal load balancing strategy does not take advantage of the service time information, that is, the probability that a job joins a given server is independent of the job's service requirement.

1.2 Organization of the paper

The rest of this paper is organized as follows. In Section 2, we describe the system model, state the assumptions, and give the mathematical formulation for the problem under consideration. In Section 3, we treat the centralized setting, which is followed by the treatment of the decentralized setting in Section 4. In Section 5, we compare the performance of the two settings using various measures, such as the server utilization and the Price of Anarchy.

2. MODEL FORMULATION

Consider a server farm consisting of a set of C servers. Let $\mathcal{S} = \{1, 2, ..., C\}$ denote the index set of the set of servers. Server j has a service rate r_j, for all $j \in \mathcal{S}$. At every server, jobs are served according to the processor sharing (PS) discipline. Customers arrive to the system according to a Poisson process with rate λ. Depending on the application in mind, a customer may correspond to a job with a certain amount of service requirement, or of a file that has to be transmitted and has a certain size. In the latter case we shall identify the service requirement of the file as being its size.

Let $\{\sigma_k : 1 \leq k \leq K\}$ denote the set of possible service requirements (i.e. the job sizes) and assume that K is finite. Let $\mathcal{K} = \{1, 2, ..., K\}$ denote the index set of the set of possible service requirement. Customers have independent and identically distributed service requirements which are sampled from $\{\sigma_k : k \in \mathcal{K}\}$ such that the probability that a customer has service requirement σ_i^{-1} is given by β_i, for all $i \in \mathcal{K}$.

As mentioned in the Introduction, we are interested in comparing the performance between the globally optimal solution and the distributed non-cooperative problem. We assume that decisions are open-loop: they are taken without knowledge of the queue sizes. However, we assume that the service requirement of an arriving user is known, both to the dispatcher in the centralized case and to the user itself in the distributed non-cooperative setting. The decision on

which queue an arrival joins is assumed to depend only on that information. Since the processes generated by splitting a Poisson process are still Poisson, each server can be seen as an $M/G/1-PS$ queue. We recall that the mean delay in a PS queue depends on the service time distribution only through its mean (the so-called insensitivity property of PS [13]), therefore the mean number of jobs in an $M/G/1-PS$ queue is the same as in an $M/M/1$ queue.

All arrivals with a given size are called a class. We thus have K classes of jobs where jobs of class i have mean size σ_i^{-1}. We associate with class i an arrival rate $\lambda_i = \lambda\beta_i$, and a traffic intensity $\eta_i = \lambda_i\sigma_i^{-1}$. Let

$$\overline{\eta} = \sum_{i\in\mathcal{K}} \eta_i$$

denote the total input traffic intensity.

REMARK 1. *Note that the value of K is arbitrary. Therefore our formulation allows us to approximate a continuous distribution arbitrarily closely, and thus we can investigate the optimal size-based routing strategy.*

Notation. We shall use a lower case bold-faced character to denote a vector. The elements of a vector will be denoted by the corresponding lower case characters. For example, $\boldsymbol{a}$ denotes the $1\times m$ vector $(a_1, a_2, ..., a_m)$ where m is the size of $\boldsymbol{a}$. The vectors $\mathbf{0}_m$ and $\mathbf{1}_m$ will denote the $1\times m$ vectors with all elements as 0 and 1, respectively. We shall use the symbol $\preceq$ to denote elementwise inequality for vectors.

Strategies. A strategy for a class i of customers is defined to be the probability vector $(p_{i1}, ..., p_{iC})$, where p_{ij} is the probability that a class i customer goes to queue j. Note that for any strategy $\sum_{j=1}^{C} p_{ij} = 1$. We define a **multi-strategy** $\mathbf{p} = p_(ij)$, $1\le i\le K$, $1\le j\le C$ as the matrix of strategies of all classes.

For a multi-strategy $\mathbf{p}$, let $\rho_j^i(\mathbf{p})$ denote the load on server j due to class i. The total load on server j is given by

$$\rho_j(\mathbf{p}) = \sum_{i\in\mathcal{K}} \rho_j^i(\mathbf{p}) = \sum_{i\in\mathcal{K}} \frac{\eta_i p_{ij}}{r_j}. \tag{1}$$

From queueing theory we know that server j is stable if $\rho_j(\mathbf{p}) < 1$. We shall say that $\mathbf{p}$ is a stable multi-strategy if all servers are stable. The next proposition states the necessary and sufficient condition for the existence of a stable multi-strategy.

PROPOSITION 1. *There exists a stable multi-strategy if and only if*

$$\sum_{j\in\mathcal{S}} r_j > \overline{\eta}. \tag{2}$$

PROOF. For a multi-strategy $\mathbf{p}$, from (1) we get

$$r_j\rho_j(\mathbf{p}) = \sum_{i\in\mathcal{K}} \eta_i p_{ij}, \quad \text{for all } j\in\mathcal{S}.$$

Summing over all j and interchanging the two summations on the right-hand side we get

$$\sum_{j\in\mathcal{S}} r_j\rho_j(\mathbf{p}) = \sum_{i\in\mathcal{K}} \eta_i \sum_{j\in\mathcal{S}} p_{ij} = \overline{\eta}. \tag{3}$$

If $\sum_{j\in\mathcal{S}} r_j < \overline{\eta}$, then the load on some server must be larger than 1 for (3) to hold. Thus, (2) is necessary for the existence of a stable multi-strategy.

Now, assume (2) and consider the multi-strategy defined by

$$p_{ij} = \frac{r_j}{\sum_{k\in\mathcal{S}} r_k}, \text{ for all } i\in\mathcal{K}, \text{ and for all } j\in\mathcal{S}.$$

Due to the splitting property of Poisson processes, the arrival process to each of the queues will also be Poisson under this multi-strategy. Then, each server can be modeled as an $M/G/1$ queue with

$$\rho_j(\mathbf{p}) = \sum_{i\in\mathcal{K}} \frac{\eta_i p_{ij}}{r_j} = \frac{\sum_{i\in\mathcal{K}} \eta_i}{\sum_{k\in\mathcal{S}} r_k} < 1. \tag{4}$$

and as a consequence every server j is stable. Thus, (2) is sufficient for the existence of a stable multi-strategy. □

ASSUMPTION 1. *The traffic intensities and the service rates are such that (2) is always satisfied.*

Note that if $\mathbf{p}$ is a stable multi-strategy, then necessarily $\sum_{j=1}^{C} \rho_j(\mathbf{p}) < C$.

Since all the queues in our system are $M/G/1-PS$ queues, the mean number of jobs at any queue has the insensitivity property: it depends on the service distribution only through its expectation. For all $j\in\mathcal{S}$, the mean number of jobs is given by

$$E[N_j(\mathbf{p})] = \frac{\rho_j(\mathbf{p})}{1-\rho_j(\mathbf{p})}, \tag{5}$$

for $\rho_j(\mathbf{p}) < 1$, and is infinity otherwise.

The total arrival rate to server j is $\sum_{i=1}^{K} \lambda_i p_{ij}$. Thus, by Little's law the mean sojourn time at queue j is given by

$$E[T_j(\mathbf{p})] = \frac{E[N_j(\mathbf{p})]}{\sum_{i=1}^{K} \lambda_i p_{ij}}. \tag{6}$$

Even though sometimes we will not make the dependency explicit, $E[N_j]$, ρ_j and $E[T_j]$, for all $j\in\mathcal{S}$, shall be understood to depend on the multi-strategy relevant to the context.

Our objective is to determine the multi-strategy $\mathbf{p}$ that minimizes the weighted mean number of jobs in the system, that is,

$$\operatorname*{argmin}_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j], \tag{7}$$

where c_j are some constants that depend on the index of the of the queue and that can represent, for example, a cost

on the holding time. We recall that in all previous works, the case $c_j = c$, for all $j \in \mathcal{S}$, was studied. By Little's law, minimizing the weighted mean number of jobs is equivalent to minimizing the weighted mean sojourn time in the system.

Finally we note that throughout the paper we will assume the servers are labeled such that

$$\frac{c_1}{r_1} \leq \frac{c_2}{r_2} \leq \ldots \leq \frac{c_C}{r_C}. \tag{8}$$

REMARK 2. *Since the objective function defined in (7) depends only on the mean service time at each of the servers, we could also interpret that the arrival stream is composed of K classes, where jobs of different classes have different service time distributions. The mean service time of class i jobs is σ_i^{-1}, for $i \in \mathcal{K}$. All the results in the present paper would hold under this interpretation as well. Nevertheless, for conciseness, in the present paper we stick to the interpretation expressed in Remark 1.*

3. THE GLOBAL OPTIMIZATION PROBLEM

In this section we consider the global optimization problem, in which a dispatcher decides where each job will get service so as to minimize the weighted mean number of jobs in the system. The global optimization problem can be formulated in terms of the following Mathematical Program (MP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{j\in\mathcal{S}} c_j E[N_j(\mathbf{p})] && (9)\\
\text{subject to} \quad & \sum_{j\in\mathcal{S}} p_{ij} = 1, \ \text{ for all } i \in \mathcal{K}; && (10)\\
& \mathbf{p} \succeq \mathbf{0}; && (11)\\
& \sum_{i\in\mathcal{K}} \eta_i p_{ij} < r_j, \ \text{ for all } j \in \mathcal{S}. && (12)
\end{aligned}$$

We note that if condition (2) is satisfied, then there exists a multi-strategy which satisfies these constraints and *vice versa*.

Since the objective function is convex and the constraints are linear, MP is a standard convex programme, and its solution can be found in polynomial time in the number of unknowns and in the number of constraints. We note that there may exist multiple multi-strategies that minimize (9) subject to (10)-(12).

3.1 Size-unaware multi-strategies

The following result will play a key role in the rest of the paper. It shows that there exists a size-unaware multi-strategy that is optimal.

PROPOSITION 2. *Let $\mathbf{p}$ be a multi-strategy satisfying the constraints (10)-(12). The multi-strategy $\hat{\mathbf{p}}$ defined by*

$$\hat{p}_{ij} = \frac{\sum_{l\in\mathcal{K}} \eta_l p_{lj}}{\overline{\eta}} = \frac{\rho_j(\mathbf{p}) r_j}{\overline{\eta}}, \tag{13}$$

for all $i \in \mathcal{K}$ and for all $j \in \mathcal{S}$, also satisfies the constraints (10)-(12). Moreover, the load on a server under $\hat{\mathbf{p}}$ is equal to the load on it under $\mathbf{p}$.

PROOF. The equality

$$\sum_{j\in\mathcal{S}} \hat{p}_{ij} \quad = \quad \sum_{j\in\mathcal{S}} \frac{\sum_{l\in\mathcal{K}} \eta_l p_{lj}}{\overline{\eta}} = 1,$$

for all $i \in \mathcal{K}$, shows that $\hat{\mathbf{p}}$ satisfies (10).

Since η_i is non-negative for all $i \in \mathcal{K}$, and $\mathbf{p}$ satisfies (11), $\hat{\mathbf{p}}$ also satisfies (11).

The equality

$$\sum_{i\in\mathcal{K}} \eta_i \hat{p}_{ij} = \sum_{i\in\mathcal{K}} \eta_i \frac{\sum_{l\in\mathcal{K}} \eta_l p_{lj}}{\overline{\eta}} = \sum_{l\in\mathcal{K}} \eta_l p_{lj}$$

helps us to verify that $\hat{\mathbf{p}}$ indeed satisfies (12).

Finally, since

$$\rho_j(\hat{\mathbf{p}}) = \frac{\sum_{i\in\mathcal{K}} \eta_i \hat{p}_{ij}}{r_j} = \frac{\sum_{l\in\mathcal{K}} \eta_l p_{lj}}{r_j} = \rho_j(\mathbf{p}),$$

for all $j \in \mathcal{S}$, the load on a server is the same under both $\mathbf{p}$ and $\hat{\mathbf{p}}$. □

From Proposition 2, we can infer that, for every feasible multi-strategy, there exists a feasible size-unaware multi-strategy such that both these strategies induce the same load on the servers. Since the objective function in the MP depends on the multi-strategy only through the induced load (cf. (5)), we can conclude that one may restrict oneself without loss of optimality to finding policies that take routing decisions independently of the (known) amount of service requirement of a job. The result of Proposition 2 further illustrates that the optimal load balancing in PS server farms is rather different than in FCFS server farms, where the size of jobs is used by the optimal routing policy.

Moreover, the value of the mathematical programming (9)-(12) can be obtained by optimizing directly over the loads. The routing probabilities can be determined later from (13), once the load on each server is determined.

Let

$$f_j(x) = \begin{cases} c_j x/(1-x), & \text{for} \quad 0 \leq x < 1; \\ \infty, & \text{otherwise.} \end{cases}$$

From (5) and Proposition 2, we can conclude that an optimal load balancing policy is obtained by applying (13) to the solution of the following Reduced Mathematical Program (RMP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{j\in\mathcal{S}} f_j(\rho_j) && (14)\\
\text{subject to} \quad & \mathbf{0} \preceq \boldsymbol{\rho} \prec \mathbf{1}; && (15)\\
& \sum_{j\in\mathcal{S}} r_j \rho_j = \overline{\eta}. && (16)
\end{aligned}$$

Constraint (16) guarantees that all incoming jobs are served.

3.2 Characterizing the solution

Depending on the values of the service rates and the holding costs per unit time, the optimal multi-strategy may not use all servers, but due to constraint (16) we are certain that at

least one server will be used. Let $\mathcal{S}_G \subseteq \mathcal{S}$ denote the subset of servers that the optimal multi-strategy uses.

In the following theorem we characterize the solution of (14)-(16). In particular we note that the solution to (14)-(16) is unique.

THEOREM 1. *The subset of servers that are used in the optimal load balancing is $\mathcal{S}_G = \{1, \ldots, j^*\}$, where*

$$j^* = \sup\left\{ j \leq C : \sum_{k=1}^{j} \sqrt{c_j r_j} > \left(\sum_{k=1}^{j} r_k - \overline{\eta} \right) \sqrt{\frac{c_j}{r_j}} \right\} \quad (17)$$

Under the optimal multi-strategy, the load on server $j \in \mathcal{S}_G$ is

$$\rho_j^* = 1 - \sqrt{\frac{c_j}{r_j}} \frac{\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}}{\sum_{k\in\mathcal{S}_G} \sqrt{c_k r_k}}. \quad (18)$$

PROOF. The Lagrangian associated with the RMP can be defined as

$$L(\boldsymbol{\rho}, \boldsymbol{\nu}, \boldsymbol{\zeta}, \gamma^G) = \sum_{j\in\mathcal{S}} f_j(\rho_j) + \sum_{j\in\mathcal{S}} \nu_j(0 - \rho_j) + \sum_{j\in\mathcal{S}} \zeta_j(\rho_j - 1) + \gamma_G \left(\sum_{j\in\mathcal{S}} r_j \rho_j - \overline{\eta} \right), \quad (19)$$

where $\boldsymbol{\nu} \succeq \mathbf{0}$, $\boldsymbol{\zeta} \succeq \mathbf{0}$ and $\gamma^G \in \mathbb{R}$.

Note that the RMP is convex. From Proposition 1 (see (4)) there exists a feasible solution. As a consequence by Slater's condition [4, Section 5.2.3] strong duality is satisfied. Then, $\boldsymbol{\rho}^*$ and $(\gamma^{G*}, \boldsymbol{\nu}^*, \boldsymbol{\zeta}^*)$ are primal and dual optimal with zero duality gap if they satisfy the Karush-Kuhn-Tucker (KKT) conditions

$$\mathbf{0} \preceq \boldsymbol{\rho}^* \prec \mathbf{1};$$
$$\sum_{j\in\mathcal{S}} r_j \rho_j^* = \overline{\eta};$$
$$\gamma^{G*} \in \mathbb{R}; \quad \boldsymbol{\nu}^* \succeq \mathbf{0}; \quad \boldsymbol{\zeta}^* \succeq 0;$$

$$\nu_j^* \rho_j^* = 0, \quad \zeta_j^*(\rho_j^* - 1) = 0, \text{ for all } j \in \mathcal{S}; \quad (20)$$

$$c_j \frac{1}{(1-\rho_j^*)^2} - \gamma^{G*} r_j - \nu_j^* + \zeta_j^* = 0, \text{ for all } j \in \mathcal{S}. \quad (21)$$

Condition (20) are the so-called complementary slackness, which hold due to strong duality.

Since the objective function tends to infinity when ρ_j tends to 1 at any server j, it follows that necessarily $\rho^* \prec \mathbf{1}$. Therefore, from (20) it follows that $\boldsymbol{\zeta}^* = \mathbf{0}$. Since $\boldsymbol{\nu} \succeq \mathbf{0}$, from (21) we get

$$\gamma^{G*} \leq \frac{c_j}{r_j} \frac{1}{(1-\rho_j^*)^2}, \text{ for all } j \in \mathcal{S}, \quad (22)$$

and on eliminating the variables ν_j from (20), we get

$$\left(c_j \frac{1}{(1-\rho_j^*)^2} - \gamma^{G*} r_j \right) \rho_j^* = 0, \text{ for all } j \in \mathcal{S}. \quad (23)$$

For a given server j, if γ^{G*} is greater than c_j/r_j, then (22) can only be satisfied if ρ_j^* is greater than 0 as well, which together with (23) implies that

$$\rho_j = 1 - \sqrt{\frac{c_j}{r_j}} \sqrt{\frac{1}{\gamma^{G*}}}. \quad (24)$$

Assume now that $\gamma^{G*} \leq c_j/r_j$. If ρ_j is greater than 0 then

$$\gamma^{G*} \leq c_j/r_j < \frac{c_j}{(1-\rho_j^*)^2 r_j},$$

which violates the complementary slackness condition (23). Thus, if $\gamma^{G*} \leq c_j/r_j$, then ρ_j^* is equal to 0. In conclusion, we have

$$\rho_j^* = \begin{cases} 1 - \sqrt{\frac{c_j}{r_j}} \sqrt{\frac{1}{\gamma^{G*}}}, & \text{if } \gamma^{G*} > c_j/r_j; \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

From the above equation, we see that ρ_j^* are non-decreasing in γ^{G*}. Therefore, there is a unique value of γ^{G*} such that constraint (16) is satisfied. Since c_j/r_j is non-decreasing in j, it now follows that $\mathcal{S}_G = \{1, \ldots, j^*\}$, where j^* can be computed using (22) and is such that

$$\frac{c_{j^*}}{r_{j^*}} < \gamma^{G*} < \frac{c_{j^*+1}}{r_{j^*+1}}. \quad (26)$$

From (24) and (16), we obtain

$$\sqrt{\frac{1}{\gamma^{G*}}} = \frac{\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}}{\sum_{k\in\mathcal{S}_G} \sqrt{c_k r_k}}, \quad (27)$$

which together with (26) gives

$$j^* = \sup\left\{ j \leq C : \frac{c_j}{r_j} < \left(\frac{\sum_{k=1}^{j} \sqrt{c_k r_k}}{\sum_{k=1}^{j} r_k - \overline{\eta}} \right)^2 \right\},$$

which is an equivalent condition to the one stated in (17)

On combining (26) and (25), we get

$$\rho_j^* = 1 - \sqrt{\frac{c_j}{r_j}} \frac{\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}}{\sum_{k\in\mathcal{S}_G} \sqrt{c_k r_k}},$$

which is the result stated in (18). □

COROLLARY 1. *The size-unaware multi-strategy, $\hat{\mathbf{p}}^*$, is given by*

$$\hat{p}_{ij}^* = \frac{\rho_j^* r_j}{\overline{\eta}}, \text{ for all } i \in \mathcal{K} \text{ and for all } j \in \mathcal{S}. \quad (28)$$

REMARK 3. *The solution structure of Theorem 1 is known as water-filling. We will say more about this in Section 4.4.*

From Theorem 1 we see that $\rho_j^* > \rho_i^*$, for any $j < i$. Since the mean number of jobs in a server increases with its load, we conclude that, under any optimal multi-strategy, $E[N_j] > E[N_i]$ for any $j < i$. Interestingly, in the next proposition we show that, even though $\rho_j^* > \rho_i^*$, the weighted mean sojourn time in server j will be smaller than the weighted mean sojourn time in server i.

PROPOSITION 3. *For the multi-strategy (28), and for any two servers j and i in $\mathcal{S}_G$,*

$$c_j E[T_j] < c_i E[T_i], \text{ for } j < i.$$

PROOF. From Little's law (see equation (6)) and the multi-strategy (2) we have

$$c_j E[T_j] = \frac{c_j E[N_j]}{\sum_{i\in\mathcal{K}} \lambda_i \hat{p}^*_{ij}} = \frac{c_j E[N_j]}{\sum_{i\in\mathcal{K}} \lambda_i \frac{\rho_j r_j}{\overline{\eta}}}.$$

Substituting (18) we get

$$c_j E[T_j] = \sqrt{\frac{c_j}{r_j}} \frac{\overline{\eta} \sum_{k\in\mathcal{S}_G} \sqrt{c_k r_k}}{\sum_{k\in\mathcal{K}} \lambda_i \left(\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}\right)}.$$

The proof now concludes by noting that for any $j < i$, $c_j/r_j < c_i/r_i$. □

3.3 Alternative characterization of the optimal solution

In this subsection we write in vector form the KKT conditions that characterize the optimal solution to the global optimization problem. This representation will play a crucial role in determining the optimal routing strategy in the distributed non-cooperative setting. For simplicity in the exposition, we assume that all servers are used.

Let us first introduce the Hadamard product for matrices. For two arbitrary matrices $\mathbf{X} = (x)_{ij}$ and $\mathbf{Y} = (y)_{ij}$ of the same dimension, we denote by $\mathbf{X} \bullet \mathbf{Y}$ the matrix whose (i,j) element is $a_{ij}b_{ij}$. Thus, the Hadamard product just refers to the element-wise product of matrices. The standard product of two matrices is denoted by $\mathbf{X} \cdot \mathbf{B}$. Finally for an arbitrary matrix $\mathbf{X}$ we denote by $\mathbf{X}^T$ its transpose matrix.

Let $\mathbf{t}(\mathbf{p})$ be the gradient of the objective function, i.e., $\mathbf{t}(\mathbf{p})$ is a matrix of dimension $K \times C$ whose (i,j) element is given by

$$t_{ij} = \frac{\partial \sum_{k\in\mathcal{S}} f_k(\mathbf{p})}{\partial p_{ij}}. \tag{29}$$

Then, similar to the derivation of (22)-(23), $\mathbf{p}$ is optimal for the original problem (9)-(12) if and only if there exist Lagrange multipliers $\gamma_1, ..., \gamma_C$ and a matrix $\mathbf{\Gamma}$ of dimensions $K \times C$ whose (i,j) element is given by

$$\mathbf{\Gamma}_{ij} = \gamma_j,$$

such that

$$(\mathbf{t} + \mathbf{\Gamma}) \bullet \mathbf{p} = \mathbf{0}, \tag{30}$$
$$\mathbf{t} + \mathbf{\Gamma} \succeq \mathbf{0}, \tag{31}$$
$$\mathbf{1}_C \cdot \mathbf{p}^T = \mathbf{1}_K, \quad \mathbf{p} \succeq \mathbf{0}. \tag{32}$$

Note that equations (30) and (31) are the analogue of equations (23) and (22), respectively.

This equivalent characterization through complementarity inequalities of a globally optimal solution will be essential for the next section.

4. THE INDIVIDUAL OPTIMALITY

We study now the distributed non-cooperative setting, where an arriving customer, say of class i, aware of its required amount of service $(\sigma_i)^{-1}$, wishes to minimize its own weighted expected sojourn time. The weighting is done according to the queue to which the file is sent as can be viewed as a pricing that may vary from one queue to another. If a class-i user chooses to be served by server j then its weighted conditional expected sojourn time there is

$$\tau_{ij}(\mathbf{p}) = c_j E[T_j(\mathbf{p})|i] = \frac{c_j}{r_j \sigma_i} \times \frac{1}{1 - \rho_j(\mathbf{p})}. \tag{33}$$

DEFINITION 1. *We say that customers of class i use queue j if $\rho^i_j > 0$; i.e., queue j receives a strictly positive load from class i.*

DEFINITION 2. *We say that a strategy $\mathbf{p}$ is* **an equilibrium** *for the individual optimization problem if for each $i = 1, ..., K$, each $j = 1, ..., C$ and each queue k used by class i,*

$$E[c_k T_k(\mathbf{p})|i] = \min_{j=1,...,K} E[c_j T_j(\mathbf{p})|i]. \tag{34}$$

Without loss of generality, we can replace the equilibrium condition in (34) with the condition

$$E[d_i c_k T_k(\mathbf{p})|i] = \min_{j=1,...,K} d_i E[c_j T_j(\mathbf{p})|i]. \tag{35}$$

where d_i are arbitrary strictly positive constants.

Equation (34) characterizes the equilibrium, since only when (34) is satisfied users will not have an incentive to deviate from their strategy.

4.1 A potential game approach to obtain the equilibrium

Denote by $\mathbf{T}(\mathbf{p})$ a $K \times C$ matrix whose (i,j) element is $\tau_{ij}(\mathbf{p})$. Let $\mathbf{a}$ be the matrix of dimensions $K \times C$ whose (i,j) element is given by $a_{ij} = a_j$.

We can characterize the equilibrium by the following relations: $\mathbf{p}$ is an equilibrium if and only if there is some $\mathbf{a}$ such that the following holds.

$$\big(\mathbf{T}(\mathbf{p}) + \mathbf{a}\big) \bullet \mathbf{p} = \mathbf{0}, \tag{36}$$
$$\mathbf{T}(\mathbf{p}) + \mathbf{a} \succeq \mathbf{0}, \tag{37}$$
$$\mathbf{1}_C \cdot \mathbf{p}^T = \mathbf{1}_K, \quad \mathbf{p} \succeq \mathbf{0}. \tag{38}$$

We observe (36)-(38) and note that they are the same as the system (30)-(32), provided that we identify the minimum cost vector $\mathbf{a}$ with the Lagrange multiplier vector $\mathbf{\Gamma}$, and we identify $\mathbf{T}$ as a gradient vector of some potential function G.

Since system (30)-(32) were equivalent to a global minimization, we conclude that (36)-(38) are equivalent to the equilibrium $\mathbf{p}$ being the global minimum of the function G subject to the constraints (38). Note that the minimum is unique in terms of ρ_j if G is a strictly convex function of ρ_j.

Games that can be transformed into an equivalent optimization problem with a common function optimized jointly by all users are known as potential games. They have been introduced in [1] in the context of road traffic, see also [17, 15, 18, 20]. In particular, the existence of a potential function is a sufficient condition for various greedy dynamics of the game to converge to equilibrium.

PROPOSITION 4. *The distributed non-cooperative game can be transformed into a standard convex optimization problem of minimizing*

$$\sum_{k=1}^{C} c_k \log \overline{T}(\rho_k(\mathbf{p})) \tag{39}$$

subject to the constraints (10)-(12) where $\overline{T}(z) := 1/(1-z)$ *for* $0 \le z < 1$ *and* ∞ *for* $z \ge 1$.

PROOF. Define

$$G(\mathbf{p}) := \sum_{k=1}^{C} \int_{z=0}^{\rho_k(\mathbf{p})} c_k \overline{T}(z) dz. \tag{40}$$

Then

$$G(\mathbf{p}) = \sum_{k=1}^{C} \int_{z=0}^{\rho_k(\mathbf{p})} c_k \overline{T}(z) dz = \sum_{k=1}^{C} c_k \log \overline{T}(\rho_k(\mathbf{p}))$$

Thus,

$$\begin{aligned} \frac{\partial G(\mathbf{p})}{\partial p_{ij}} &= c_j \overline{T}_j(\mathbf{p}) \times \frac{\mathrm{d}\rho_j}{\mathrm{d}p_{ij}} \\ &= \frac{c_j}{1-\rho_j(\mathbf{p})} \times \frac{\lambda_i}{\sigma_i r_j} = \lambda_i \Big(c_j E[T_j(\mathbf{p})|i] \Big) \end{aligned}$$

We conclude that G is indeed a potential as its gradient coincides with the original costs as given in (35), where $d_i = \lambda_i$.

The optimal solution $\mathbf{p}$ to (39) is given by the only vector that satisfies the KKT conditions, which in turn are precisely given by (36)-(38), where $\mathbf{a}$ denotes the Lagrange multiplier vector. □

This implies that indeed the game can be transformed into a standard convex optimization problem of minimizing G subject to the constraints (10)-(12), whose solutions are equilibria in the original game.

As we did in Section 3.1, we can further simplify the above optimization problem. Indeed, the value is directly obtained through minimizing $G(\mathbf{p}) := \sum_{k=1}^{C} \int_{z=0}^{\rho_k} c_k \overline{T}(z) dz$ subject to (15)-(16). The solution to the game problem is obtained from the loads that achieve the minimization by using (13).

4.2 Fairness

Let us interpret the meaning of the potential function G. Define $\Delta_k := 1 - \rho_k$ to be the **excess capacity** at server k. We note that the argument that achieves the minimization of $G(\mathbf{p})$ achieves the maximum of the product of $(\Delta_1)^{c_1} \times (\Delta_2)^{c_2} \times \cdots \times (\Delta_C)^{c_C}$. We conclude the following:

THEOREM 2. *The individual optimal load balancing solution coincides with the routing strategy that achieves the weighted proportional fair excess capacities between the C servers, where the weight for server k is given by the powers* c_k.

PROOF. The result is a direct consequence of (39) and the definition of Proportional Fair allocation. □

4.3 Characterizing the Individual Optimal solution

Since we have shown that the individual setting corresponds to a potential game, in equilibrium, the optimal routing strategy will minimize (40) subject to (15)-(16). We have the following result.

THEOREM 3. *The subset of servers that are used in the optimal routing strategy in the non-cooperative setting is of type* $\mathcal{S}_I = \{1, \ldots, j^*\}$, *where*

$$j^* = \sup \left\{ j \le C : \sum_{k=1}^{j} c_j > \left(\sum_{k=1}^{j} r_k - \overline{\eta} \right) \frac{c_j}{r_j} \right\} \tag{41}$$

For every $j \in \mathcal{S}_I$, *the load is*

$$\rho_j = 1 - \frac{c_j}{r_j} \frac{\sum_{k=1}^{j} r_k - \overline{\eta}}{\sum_{k=1}^{j} c_k}. \tag{42}$$

PROOF. The derivation follows the same steps of the proof of Theorem 1. From Proposition 1 (see equation (4)) there exists a feasible solution. As a consequence, by Slater's condition [4, Section 5.2.3] strong duality holds. Then from the Karush-Kuhn-Tucker (KKT) conditions if

$$0 \le \rho_j \le 1, \quad j = 1, \ldots, C,$$

$$\sum_{j \in \mathcal{S}_I} r_j \rho_j = \overline{\eta},$$

$$\gamma^I \in \mathbb{R}, \quad \nu_j \ge 0, \quad \zeta_j \ge 0, \quad j = 1, \ldots, C,$$

$$\nu_j \rho_j = 0, \quad \zeta_j (\rho_j - 1) = 0, \quad j = 1, \ldots, C, \tag{43}$$

$$\frac{c_j}{(1-\rho_j)} - \gamma^I r_j - \nu_j + \zeta_j = 0, \tag{44}$$

then ρ_j, $j = 1, \ldots, C$ and (γ^I, ν, ζ) are primal and dual optimal with zero duality gap.

Since the objective function tends to infinity if $\rho_j \to 1$ at some server, it follows that necessarily $\rho_j < 1$, $j = 1, \ldots, C$. Because of (43) this implies that $\zeta_j = 0$, for all j. Now note that ν_j are slack variables which can be eliminated. Since $\nu_j \ge 0$, from (44) we get

$$\gamma^I \le \frac{c_j}{r_j} \frac{1}{(1-\rho_j)}, \tag{45}$$

and from (43) we have

$$\left(\frac{c_j}{(1-\rho_j)} - \gamma^I r_j \right) \rho_j = 0. \tag{46}$$

Now, if $\gamma^I > c_j/r_j$, equation (45) can only be satisfied if $\rho_j > 0$, and from (46) this implies that

$$\rho_j = 1 - \frac{c_j}{r_j}\frac{1}{\gamma^I}. \quad (47)$$

Assume now that $\gamma^I \leq c_j/r_j$. If $\rho_j > 0$ then this implies that $\gamma^I \leq c_j/r_j < \frac{c_j}{(1-\rho_j)r_j}$, which violates the complementary slackness condition (46). Thus if $\gamma^I \leq c_j/r_j$ then $\rho_j = 0$. In conclusion we have that

$$\rho_j = \begin{cases} 1 - \frac{c_j}{r_j}\frac{1}{\gamma^I} & \gamma^I > c_j/r_j \\ 0 & \gamma^I < c_j/r_j. \end{cases}$$

It follows that $\rho_j > 0$ are non-decreasing in γ^I. Thus there is a unique value of γ^I such that constraint (16) is satisfied. It follows that $\mathcal{S}_I = \{1, \ldots, j^*\}$. From (45) we have that the index j^* is such that

$$\frac{c_{j^*}}{r_{j^*}} < \gamma^I < \frac{c_{j^*+1}}{r_{j^*+1}}. \quad (48)$$

Substituting (47) in (16) we get

$$\frac{1}{\gamma^I} = \frac{\sum_{k\in\mathcal{S}_I} r_k - \sum_{i=1}^K \eta_i}{\sum_{k\in\mathcal{S}_I} c_k}. \quad (49)$$

This proves equation (42).

From (48) we get that server j is used if and only if

$$\frac{c_j}{r_j} < \frac{\sum_{k=1}^j c_k}{\sum_{k=1}^j r_k - \sum_{i=1}^K \eta_i},$$

from where (41) follows. □

We note that a routing strategy that achieves the desired load (42) in every server (and as a consequence the same performance) can be obtained by (13).

REMARK 4. *From (42) it is easy to see that (34) is satisfied for each $i = 1, \ldots, K$ and each $j \in \mathcal{S}_I$. This can also be seen from equation (47), which implies that in every server $j \in \mathcal{S}_I$ that is used the mean cost per unit of service required at the server, $\frac{c_j/r_j}{1-\rho_j} = \gamma^I$, is independent of the server.*

From Remark 4 and Proposition 3 we observe the main difference between the global and individual optimal solutions. In the individual optimal solution is constrained to a solution such that the mean sojourn time is the same in each server. In the global optimal solution the weighted mean sojourn time varies across the servers, and in fact, it increases as the index of the server increases (see Proposition 3).

When $c_i = c$, $\forall i$, equation (41) becomes

$$r_{j+1} < (\sum_{k=1}^j r_k - \sum_{k=1}^K \eta_i)\frac{1}{j}. \quad (50)$$

Equation (50) has a clear interpretation. Server $j+1$ will not be used if the exceeding capacity per server when j servers are used is larger than r_{j+1}.

4.4 The structure of the selfish routing

We recall from (8) that servers are relabeled in increasing order with respect to the ratio c_j/r_j, $j = 1, \ldots, C$. Let there be M_1 servers with $c_i/r_i = c_1/r_1$. Let there be M_2 servers with $c_i/r_i = c_{M_1+1}/r_{M_1+1}$. Let there be M_k servers with $c_i/r_i = c_{M_{k-1}+1}/r_{M_{k-1}+1}$.

Then, from (34), the optimal policy has the following water-filling structure. For λ sufficiently small, only the first M_1 servers receive positive flow. This flow is assigned in a way that equalizes the expected delay among the first $M1$ servers. We increase λ till a point where

$$\frac{c_1}{r_1} \times \frac{1}{1-\rho_1(\mathbf{p})} = \frac{c_2}{r_2}.$$

From this point, we route flow to all $M_1 + M_2$ first servers in a way that equalizes the expected delays on these servers. No flow is sent to other servers.

This type of solution is often referred as to water-filling.

5. COMPARING THE GLOBAL AND INDIVIDUAL OPTIMUM SOLUTIONS

In this section we compare the optimal load balancing expressed in Theorems 1 and 3. Our first result shows $\mathcal{S}_I \subseteq \mathcal{S}_G$, that is, the number of servers that are used in the global optimum solution is greater or equal to the number of servers used in the distributed non-cooperative setting. This indicates that in the non-cooperative setting, users will tend to overload fast servers, and fail to recognize the benefits that using a slower server can have. A similar property was proven in [2] for a exponential multi-server system.

In this section, ρ_j^G and ρ_j^I will denote the load in server j in the global and individual optimal solution, respectively. In view of (24) and (47) we will consider that both $\rho_j^G := \rho_j^G(\gamma)$ and $\rho_j^I := \rho_j^I(\gamma)$ are a function of a common variable γ.

We start with the following Lemma.

LEMMA 1. *For $0 < \gamma \leq c_j/r_j$, $\rho_j^G(\gamma) = \rho_j^I(\gamma) = 0$. For $\gamma > c_j/r_j$, $\rho_j^G(\gamma) < \rho_j^I(\gamma)$.*

PROOF. The case $\gamma \leq c_j/r_j$ is obvious. For the second case, we have

$$\begin{aligned} \gamma &> c_j/r_j \\ \frac{\sqrt{\gamma}}{\sqrt{\gamma}}\sqrt{\gamma} &> \sqrt{\frac{c_j}{r_j}} = \frac{\frac{c_j}{r_j}}{\sqrt{\frac{c_j}{r_j}}} \\ \sqrt{\frac{c_j}{r_j}}\sqrt{\frac{1}{\gamma}} &> \frac{c_j}{r_j}\frac{1}{\gamma} \end{aligned}$$

and from equations (24) and (47) it follows that $\rho_j^G(\gamma) < \rho_j^I(\gamma)$. □

PROPOSITION 5. *For any arrival rate and service time distribution it holds $\mathcal{S}_I \subseteq \mathcal{S}_G$*

PROOF. From Theorems 1 and 3 (equations (27) and (49)) it is sufficient to prove that $\gamma^G > \gamma^I$. We prove the statement by contradiction. Assume that $\gamma^G \leq \gamma^I$. If $\gamma^I < c_j/r_j$, then $\rho_j^I(\gamma) = \rho_j^G(\gamma) = 0$. If $\gamma^I > c_j/r_j$ then $\rho_j^I(\gamma^I) > 0$ and from Lemma 1 we have

$$\begin{array}{rcl} \rho_j^I & = & \rho_j^I(\gamma^I) \\ & \overset{\gamma^I \geq \gamma^G}{\geq} & \rho_j^I(\gamma^G) \\ & \overset{Lemma\ 1}{>} & \rho_j^G(\gamma^G) = \rho_j^G. \end{array}$$

It follows then that

$$\sum_{j=1}^{C} r_j(\rho_j^I - \rho_j^G) > 0,$$

but this is a contradiction with (3), and as a consequence $\gamma^G > \gamma^I$. □

In the following theorem we show that the individual optimal overloads the servers with smallest c_j/r_j.

THEOREM 4. *There exists an index i^* such that*

$$\begin{cases} \rho_j^G < \rho_j^I & j < i^* \\ \rho_j^G > \rho_j^I & j \geq i^*. \end{cases}$$

PROOF. Due to constrain (3), there exists an index i^* such that $\rho_{i^*}^G > \rho_{i^*}^I$. Now it suffices to show that $\rho_j^G > \rho_j^I$, for all $j > i^*$. From (24) and (47) we have that

$$\begin{array}{rcl} \rho_{i^*}^G & > & \rho_{i^*}^I \\ \sqrt{\frac{c_{i^*}}{r_{i^*}}}\sqrt{\frac{1}{\gamma^G}} & < & \frac{c_{i^*}}{r_{i^*}}\frac{1}{\gamma^I} \\ \gamma^I & < & \sqrt{\frac{c_{i^*}}{r_{i^*}}}\sqrt{\gamma^G}. \end{array}$$

Since $j > i^*$, it follows that $c_j/r_j > c_{i^*}/r_{i^*}$. Thus

$$\begin{array}{rcl} \gamma^I & < & \sqrt{\frac{c_{i^*}}{r_{i^*}}}\sqrt{\gamma^G} \\ & < & \sqrt{\frac{c_j}{r_j}}\sqrt{\gamma^G} \\ & = & \frac{\frac{c_j}{r_j}}{\sqrt{\frac{c_j}{r_j}}}\sqrt{\gamma^G}, \end{array}$$

and rearranging we get

$$\sqrt{\frac{c_j}{r_j}}\sqrt{\frac{1}{\gamma^G}} < \frac{c_j}{r_j}\frac{1}{\gamma^I}.$$

From (24) and (47) it follows that $\rho_j^G > \rho_j^I$. □

5.1 Price of Anarchy

We now study the so-called Price of Anarchy.

Definition. The price of anarchy (PoA) is defined as the ratio between the performance (mean delay) obtained by the Wardrop equilibrium and the global optimal solution [14] (see also [19]).

By Little's law, calculating the ratio between the mean delays is equivalent to calculating the ratio of the mean number of users. Then from the objective function (7) and the solution of Theorems 1 and (3) we get (note that $\frac{x}{1-x} = \frac{1}{1-x} - 1$):

$$PoA = \frac{\frac{\sum_{k\in\mathcal{S}_I} c_k \sum_{k\in\mathcal{S}_I} r_k}{\sum_{k\in\mathcal{S}_I} r_k - \overline{\eta}} - \sum_{k\in\mathcal{S}_I} c_k}{\frac{\left(\sum_{k\in\mathcal{S}_G}\sqrt{c_k r_k}\right)^2}{\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}} - \sum_{k\in\mathcal{S}_G} c_k}. \qquad (51)$$

The Price of Anarchy has been studied as a measure of the inefficiency of selfish-routing (or non-cooperative decentralized) in networks. This measure has received lot of attention in recent years. For example, in an important general result, it has been shown that when the cost function in every arc is linear, then for any arbitrary multi-commodity network the PoA is upper bounded by 4/3 [19]. In [10] and [21] the authors study a multi-server system with the objective of minimizing (7) with equal costs, that is, $c_j = c$, $\forall j$, and show that $PoA \leq C$, with C denoting the number of servers. Note that the upper bound holds for any parameter configuration. In addition, in [10, Example 3.1] it is shown that the upper bound is tight, i.e., there exists a network configuration such that the PoA is arbitrarily close to C. This result indicates that the inefficiency of selfish routing is limited. In Theorem 5 we show that this changes dramatically when holding costs per unit of time associated to each server are considered in the objective function. In this case the PoA is unbounded, that is, for every $\theta < \infty$, there exist a set of values such that $PoA > \theta$. Our main result on the Price of Anarchy is the following.

THEOREM 5. *For every θ, there exist c_j and r_j, $j \in \mathcal{S}$, such that $PoA > \theta$.*

PROOF. In order to prove this result we construct an example in which PoA can be unbounded. Let $r_1 > \overline{\eta}$, and let $c_j = r_j = 1$ for $2 \leq j \leq C$. Let

$$\frac{(r_1 - \overline{\eta})}{r_1} < c_1 < r_1 - \overline{\eta}. \qquad (52)$$

For this particular choice of costs and server speeds, c_j/r_j is non-decreasing in j.

We first show that in the globally optimal multi-strategy all the servers are used, whereas in the solution of the individual optimization problem only the first server is used.

Global optimization: Note that $c_j/r_j = 1$, $\forall j \geq 2$. In view of (17), server j, $j \geq 2$, will be used if

$$\begin{array}{rcl} \sum_{k=1}^{j}\sqrt{c_j r_j} & = & \sqrt{c_1 r_1} + j - 1 \\ & > & r_1 - \overline{\eta} + j - 1 \\ & = & \left(\sum_{k=1}^{j} r_j - \overline{\eta}\right)\sqrt{\frac{c_j}{r_j}}, \end{array}$$

where the inequality follows from the assumption $c_1 > \frac{(r_1-\overline{\eta})^2}{r_1}$. Since this is true for every $j \leq C$, the load on every server is positive.

Individual optimization: For $j = 2$, the left-hand side of (41)

$$\begin{aligned} c_1 + c_2 &= c_1 + r_2 \\ &< r_1 - \overline{\eta} + r_2 \\ &= (r_1 + r_2 - \overline{\eta})\sqrt{\frac{c_2}{r_2}}, \end{aligned}$$

where the inequality follows from the assumption $c_1 < r_1 - \overline{\eta}$. Thus, in the non-cooperative setting all the jobs choose to go to the first server.

From (51), the Price of Anarchy

$$\begin{aligned} PoA &= \left(\frac{c_1 r_1}{r_1 - \overline{\eta}} - c_1\right) \\ &\quad \times \frac{1}{\frac{(\sum_{k\in\mathcal{S}_G}\sqrt{c_k r_k})^2}{\sum_{k\in\mathcal{S}_G} r_k - \overline{\eta}} - \sum_{k\in\mathcal{S}_G} c_k} \\ &= \frac{c_1\overline{\eta}}{r_1 - \overline{\eta}} \times \end{aligned}$$

$$\frac{r_1 - \overline{\eta} + (C-1)}{(\sqrt{c_1 r_1} + (C-1))^2 - (c_1 + (C-1))(r_1 - \overline{\eta} + (C-1))} \tag{53}$$

Since $\frac{(r_1-\overline{\eta})^2}{r_1} < c_1 < r_1 - \overline{\eta}$, let

$$\begin{aligned} c_1 &= \frac{1}{2}\left(\frac{(r_1-\overline{\eta})^2}{r_1} + r_1 - \overline{\eta}\right) \\ &= (r_1 - \overline{\eta})\frac{2r_1 - \overline{\eta}}{2r_1}. \end{aligned} \tag{54}$$

Now as $r_1 \downarrow \overline{\eta}$, the numerator of (53) tends to $\overline{\eta}^2(C-1)$, whereas the denominator tends to 0.

Therefore, by choosing r_1 close enough to $\overline{\eta}$, the Price of Anarchy for this system can be made to exceed any given real number. □

REMARK 5. *We note that examples where the PoA is unbounded have been previously found. For instance, it is easy to determine an instance of the popular Prisoner's dilemma where the PoA is unbounded. It also follows from the network studied in [11] that the PoA is unbounded.*

5.1.1 Discussion on Theorem 5

In order to provide an intuitive idea behind Theorem 5, first note that a key underlying idea is that in the global optimal all servers are used, whereas in the non-cooperative setting only one server is used. This property follows directly from the the upper and lower bounds of (52). Let us consider the lower bound in (52). From equations (26) and (27) and the water-filling structure of the solution, we see that if $\frac{r_1 c_1}{(r_1-\overline{\eta})^2} < \frac{c_2}{r_2} = 1$, only server 1 will be used. Server 2 (and similarly all other servers), will start being used exactly when $c_1 > (r_1 - \overline{\eta})^2/r_1$, which explains the lower bound on c_1 in (52). Similarly, from (48) and (49) we can see that the upper bound in (52) guarantees that only server 1 is used in the non-cooperative setting.

As we have seen, the Price of Anarchy is given by $PoA = \frac{\min_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j^I]}{\min_{\mathbf{p}} \sum_{j=1}^{C} c_j E[N_j^G]}$. Let us look to the numerator and denominator separately.

In the non-cooperative solution only server 1 is used. Thus $\sum_{j=1}^{C} c_j E[N_j^I] = c_1 E[N_1^I]$, and server 1 is a standard $M/G/1$ queue. Thus, as $r_1 \downarrow \overline{\eta}$, $E[N_1^I]$ tends to infinity, but this is compensated by the fact that $c_1 \to 0$, and overall $c_1 E[N_1^I] \to \overline{\eta}/2$. Another way to see this is from equation (33), where we see that $\tau_{i1} = \frac{c_1}{r_1 - \overline{\eta}}$. Thus, with c_1 given from (54), it turns out that as $r_1 \downarrow \overline{\eta}$, the performance (weighted with the cost) that users joining server 1 remains unchanged.

In the global optimal solution, always all servers are used. As $r_1 \downarrow \overline{\eta}$, the global optimal also tends to route everything towards server 1, but the key property is that since all servers are used, the global optimal can do this in such a way that $E[N_1^G]$ grows more slowly than the decrease of c_1, and as a consequence $c_1 E[N_1^G] \to 0$.

More specifically, this is what happens with the global optimal solution. First, for all $j \geq 2$, as $r_1 \downarrow \overline{\eta}$ (and c_1 given by (54)), $\rho_j \to 0$. Since c_j, $\forall j \geq 2$, remain constant this implies that $\sum_{j=2}^{C} c_j E[N_j^G] \to 0$. Concerning server 1, from (24), as $r_1 \downarrow \overline{\eta}$, $\rho_1 = 1 - o(\sqrt{r_1 - \overline{\eta}})$, which implies that $E[N_1^G] = O(1/\sqrt{r_1 - \overline{\eta}})$. Since $c_1 = o(r_1 - \overline{\eta})$ as $r_1 \downarrow \overline{\eta}$, it turns out that $c_1 E[N_1^G] \to 0$. Thus, for the global optimal solution $\sum_{j=1}^{C} c_j E[N_j^G] \to 0$ as $r_1 \downarrow \overline{\eta}$, which explains why the PoA can not be bounded.

This result states that the PoA is unbounded for the load balancing problem under consideration. It is in complete contrast to finite upper bounds obtained by [10, 21], for similar models but without holding costs per unit of time associated to each server. Thus, when holding costs are taken into account, a significantly different PoA is obtained.

5.2 The case when r_j and c_j/r_j are not equal

Theorem 5 can be extended to the case when not all r_j are equal and c_j/r_j are not necessarily equal.

Let $\overline{r} = \sum_{j\in\mathcal{S}} r_j$ be the aggregate available service rate of system. Let us assume that we are given a sequence of server rates r_j such that $r_1 > \overline{\eta}$. We wish to show that there exists a sequence $\{c_j, j \in \mathcal{S}\}$, such that c_j/r_j is strictly increasing and that the following two inequalities are satisfied

$$c_1 + c_2 < (r_1 + r_2 - \overline{\eta})c_2/r_2, \tag{55}$$

$$\sum_{j\in\mathcal{S}} \sqrt{c_j r_j} > (\sum_{j\in\mathcal{S}} r_j - \overline{\eta})\sqrt{c_C/r_C}, \tag{56}$$

which would imply that only the first server is used in the solution of the individual optimization problem whereas all the servers are used in the global solution.

From (55), we require

$$\frac{c_2}{r_2} > \frac{c_1}{r_1}\frac{r_1}{r_1 - \overline{\eta}}.$$

For $2 \leq j \leq C$, let

$$\frac{c_j}{r_j} = \frac{c_1}{r_1}\frac{r_1}{r_1 - \overline{\eta}}\alpha^{2j},$$

which results in an increasing sequence $\{c_j/r_j, j \in \mathcal{S}\}$ provided that $\alpha > 1$. We shall show that there exists an $\alpha > 1$ such that the two inequalities (55) and (56) are satisfied.

The left-hand side of (56)

$$\begin{aligned}\sum_{j\in\mathcal{S}}\sqrt{c_j r_j} &= \sum_{j\in\mathcal{S}}\sqrt{\frac{c_j}{r_j}}r_j \\ &= \sqrt{\frac{c_1}{r_1}}r_1 + \sum_{j\geq 2}\sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\alpha^2 r_j \\ &> \sqrt{\frac{c_1}{r_1}}r_1 + \sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\left(\sum_{j\geq 2} r_j\right).\end{aligned}$$

Thus, we need to find an α larger than 1 which satisfies the inequality

$$\begin{aligned}\sqrt{\frac{c_1}{r_1}}r_1 + \sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\left(\sum_{j\geq 2} r_j\right) &> (\overline{r}-\overline{\eta})\sqrt{c_C/r_C} \\ &= (\overline{r}-\overline{\eta})\sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\alpha^C.\end{aligned}$$

The left-hand side of the above inequality,

$$\begin{aligned}&\sqrt{\frac{c_1}{r_1}}r_1 + \sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\left(\sum_{j\geq 2} r_j\right) \\ &= \sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}\left(\sqrt{r_1(r_1-\overline{\eta})} + \sum_{j\geq 2} r_j\right) \\ &> \sqrt{\frac{c_1}{r_1}}\sqrt{\frac{r_1}{r_1-\overline{\eta}}}(\overline{r}-\overline{\eta})\end{aligned}$$

where the inequality follows from the fact that $\sqrt{r_1(r_1-\overline{\eta})} > \sqrt{(r-\overline{\eta})(r_1-\overline{\eta})} = r_1 - \overline{\eta}$. Thus, there exists an α larger than 1 for which $\mathcal{S}_I = \{1\}$ and $\mathcal{S}_I \subset \mathcal{S}_G$. As $r_1 \downarrow \overline{\eta}$, PoA will become unbounded in this case as well.

6. REFERENCES

[1] M. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics and Transportation*. Yale University, 1956.

[2] C.H. Bell and S. Stidham. Individual versus social optimization in the allocation of customers to alternative servers. *Management Science*, 29:831–839, 1983.

[3] S.C. Borst. Optimal probabilistic allocation of customer types to servers. In *Proceedings of ACM SIGMETRICS*, pages 116–125, September 1995.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[5] V. Cardellini, E. Casalicchio, M. Colajanni, and P.S. Yu. The state of the art in locally distributed Web-server systems. *ACM Computing Surveys*, 34(2):263–311, 2001.

[6] 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, 1979.

[7] H. Feng, V. Misra, and D. Rubenstein. Optimal state-free, size-aware dispatching for heterogeneous M/G/-type systems. *Performance Evaluation*, 62(1–4):36–39, 2005.

[8] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt. Analysis of join-the-shortest-queue routing for web server farms. In *Proceedings of Performance*, page 180, 2007.

[9] M. Harchol-Balter, M. Crovella, and C. Murta. On choosing a task assignment policy for a distributed server system. *IEEE Journal of Parallel and Distributed Computing*, 59(2):204–228, 1999.

[10] M. Haviv and T. Roughgarden. The price of anarchy in an exponential multi-server. *Operations Research Letters*, 35:421–426, 2007.

[11] H. Kameda, E. Altman, O. Pourtallier, J. Li, and Y. Hosokawa. Paradoxes in performance optimization of distributed systems. In *Proceedings of SSGRR 2000 Computer and ebusiness conference*, 2000.

[12] H. Kameda, J. Li, C. Kim, and Y. Zhang. *Optimal load balancing in distributed computer systems*. Springer-Verlag, 1997.

[13] F. Kelly. *Stochastic Networks and Reversibility*. Wiley, Chichester, 1979.

[14] E. Koutsoupias and C.H. Papadimitriou. Worst-case equilibria. In *Proceedings of STACS 1999*, 1999.

[15] D. Monderer and L.S. Shapley. Potential games. *Games and Econ. Behavior*, 14:124–143, 1996.

[16] L.M. Ni and K. Hwang. Optimal load balancing in a multiple processor with many job classes. *IEEE Trans. Software Eng.*, 11(5):491–496, 1985.

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

[18] R.W. Rosenthal. A class of games possessing pure strategy Nash equilibria. *Int. J. Game Theory*, 2:65–67, 1973.

[19] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.

[20] W.H. Sandholm. Potential games with continuous player sets. *Journal of Economic Theory*, 97:81–108, 2001.

[21] D. Starobinski and T. Wu. Performance of server selection algorithms for content replication networks. In *IFIP Networking*, 2005.