Estimating the Price of Anarchy Using Load Balancing Measure

Jae-Hoon Kim, *Member, KIMICS*

***Abstract*—** We consider the problem of optimizing the performance of a system with resources shared by non-cooperative users. The worst-cast ratio between the cost of a Nash equilibrium and the optimal cost, called *Price of Anarchy*, is investigated. It measures the performance degradation due to the users' selfish behavior. As the objective function of the optimization problem, we are concerned in a load balancing measure, which is different from that used in the previous works. Also we consider the *Stackelberg scheduling* which can assign a fraction of the users to resources while the remaining users are free to act in a selfish manner.

***Index Terms*—** Nash equilibrium, load balancing, Price of Anarchy.

I. INTRODUCTION

In large-scale communication networks, for example, the Internet, there is no central authority to control the allocation of shared resources. So the users of the network sharing the resources are free to act due to their private interest. Such systems with non-cooperative users have already been studied in the early 1950s in the context of road traffic systems [14]. In computer science, this kind of systems have become increasingly important. This can be modeled by a *non-cooperative game* and it is needed game-theoretic concepts such as *Nash equilibria*.

The performance of a system can be evaluated through several measures. In particular, we focus on the *load balancing measure*. It measures how evenly the load is spread over resources. It is well-known that Nash equilibria lead to inefficient system performance. The *Price of Anarchy* was introduced in [9] to qualify the performance degradation according to the users' selfish behavior. It is the worst-case ratio of the cost of a Nash equilibrium to the optimal cost.

In many systems, a fraction of the users are willing to follow a strategy suggested by the system manager, which leads to a mixture situation of *selfishly controlled* and *centrally controlled* users. For this setting, Korilis *et.al.* [8] introduced the notion of *Stackelberg scheduling*. In Stackelberg scheduling, a central authority coordinates a fixed fraction of users and assigns them to minimize the performance degradation due to the selfish behavior of the remaining users.

Manuscript received April 9, 2009 ; revised May 21, 2009.

Jae-Hoon Kim is with the Division of Computer Engineering , Pusan University of Foreign Studies, Busan, 608-738, Korea (Tel: +82-51-640-3421, Fax: +82-51-640-3038, Email: jhoon@pufs.ac.kr)

II. Related Work

Koutsoupias and Papadimitriou [9] introduced a simple network model consisting of a single source and a single destination which are connected by parallel links. Each of agents sends a particular amount of traffic along a link from the source to the destination. The traffics are *unsplittable*, also called *atomic*. Associated with each link is a capacity representing the rate at which the link processes load of traffics. So the latency functions are linear. Also as a global objective function, usually coined as *social cost*, they considered the maximum latency of agents. Inspired by their work, there have been many researches [1, 2, 5, 7, 10] to analyze the Price of Anarchy in atomic model, which mostly deal with the total latency of agents.

The model in which the traffics can be split into arbitrary pieces, called *non-atomic*, have already been studied in the 1950's [14]. Recently, Roughgarden and Tardos [13] re-investigated this model. In particular, Roughgarden [12] proved that it is NP-hard to compute an optimal Stackelberg schedule and investigated the Price of Anarchy of Stackelberg scheduling algorithms. Also there are recent papers [6, 11] to extend the results of [12] in several directions.

In this paper, we focus on the non-atomic model. Also as a global objective function, we consider a load balancing measure. The load balancing is one of the important optimization problems and has been extensively studied [3, 4]. We will adopt the L_p norm of loads used in load balancing literature. It is a significant difference from the previous results.

III. Model

We have a set M of m machines on which jobs are allocated. Given a positive *rate* r of job arrivals, an *assignment* of jobs to machines is an m-vector $x \in \mathbb{R}^m$ such that $\sum_{i=1}^{m} x_i = r$. Here each x_i represents the load of jobs assigned on machine i. Each machine i has a *latency function* $\ell_i(\cdot)$ that estimates a load-dependent time to execute a job. We assume that each latency function is nonnegative, continuous, and nondecreasing. In particular, we will deal with a linear latency function $\ell_i(x_i) = a_i x_i$. It means that a latency on a machine is proportional to its load. Also we note that all jobs assigned to the same machine experience the same latency. In other words, we can imagine that the machines execute the jobs in round-robin. In this paper, as the cost of an assignment x, we are concerned in a load balancing measure, particularly, $C(x) = \sum_{i=1}^{m} x_i^p$, which is the L_p norm of the assignment x.

First, let us consider an optimal assignment x^* in the norm L_p. Specifically, the optimal assignment x^* minimizes the cost $C(x)$ such that $\sum_{i=1}^{m} x_i = r$. Here we can obtain an explicit representation of x^* in the following lemma. It states the optimal assignment distributes the loads equally to machines.

Lemma 1 *If x^* is an assignment to minimize $C(x) = \sum_{i=1}^{m} x_i^p$, then $x_i^* = \frac{r}{m}$, $\forall i$.*

IV. Selfish Assignment

The jobs are generated and voluntarily assigned to machines by selfish agents, each of who seeks to minimize the latency of her job. Under this assumption, we consider the stable state or the equilibrium of assignments of jobs; in this state, no job can strictly decrease the latency it experiences by changing machines. Specifically, an assignment x to M is said to be at *Nash equilibrium* or a *Nash assignment* if for any $i, j \in M$ with $x_i, x_j > 0$, $\ell_i(x_i) = \ell_j(x_j)$. Denote a Nash assignment by $x^N = (x_1^N, \cdots, x_m^N)$. Then we can explicitly get a Nash assignment x^N.

Lemma 2 *If x^N is a Nash assignment, that is, $\ell_i(x_i) = \ell_j(x_j)$, $\forall\, i, j$,* then

$$x_i^N = \frac{1}{a_i} \cdot \frac{r}{\sum_{i=1}^{m} \frac{1}{a_i}}.$$

Proof: Let $\mathcal{L}$ be the equal latency of the Nash assignment x^N, that is, $\mathcal{L} = a_1 x_1^N = a_2 x_2^N = \cdots = a_m x_m^N$. Then, $x_i^N = \frac{\mathcal{L}}{a_i}, \forall i$. Since $\sum_{i=1}^{m} x_i^N = r$, $\mathcal{L} \cdot \sum_{i=1}^{m} \frac{1}{a_i} = r$. Thus,

$$\mathcal{L} = \frac{r}{\sum_{i=1}^{m} \frac{1}{a_i}}.$$

Consequently, the Nash assignment x^N is given from the explicit representation of $\mathcal{L}$.

V. Price of Anarchy

In this section, we will estimate the Price of Anarchy (PoA), which is the worst-case ratio between the cost of a Nash assignment and that of the optimal assignment. First, we present an upper bound on PoA which depends on the number of machines or the latency functions. Also a lower bound is obtained to show that the dependence on the number of machines is necessary.

Theorem 1 For m machines with latency functions $\ell_i(x_i) = a_i x_i$, $a_1 \leq \cdots \leq a_m$,

$$PoA = \frac{C(x^N)}{C(x^*)} \leq \min\{m^{p-1}, \left(\frac{a_m}{a_1}\right)^p\}.$$

Proof: In the previous sections, the optimal assignment x^* and the Nash assignment x^N are revealed. From Lemma 1, the cost of the optimal assignment, $C(x^*)$, is given by $\sum_{i=1}^{m} \left(\frac{r}{m}\right)^p = \frac{r^p}{m^{p-1}}$. Also the cost of the Nash assignment, from Lemma 2, is given as follows:

$$C(x^N) = \sum_{i=1}^{m} \left(\frac{1}{a_i}\right)^p \left(\frac{r}{\sum_{i=1}^{m} \frac{1}{a_i}}\right)^p = \frac{r^p}{\left(\sum_{i=1}^{m} \frac{1}{a_i}\right)^p} \sum_{i=1}^{m} \left(\frac{1}{a_i}\right)^p.$$

Thus the PoA is upper bounded as

$$\frac{C(x^N)}{C(x^*)} = \frac{m^{p-1}\sum_{i=1}^{m}\left(\frac{1}{a_i}\right)^p}{\left(\sum_{i=1}^{m}\frac{1}{a_i}\right)^p} \le m^{p-1}.$$

The last inequality is valid from the Cauchy-Schwartz inequality. Also from the equality, we can derive another upper bound as follows:

$$\frac{C(x^N)}{C(x^*)} \le \frac{m^{p-1}m\left(\frac{1}{a_1}\right)^p}{m^p\left(\frac{1}{a_m}\right)^p} = \left(\frac{a_m}{a_1}\right)^p,$$

because $a_1 \le \cdots \le a_m$.

Next, we will show that PoA is not independent of the number of machines.

Theorem 2 There is an instance of m machines such that $PoA \ge \frac{m^{p-1}}{2^p}$.

Proof: The m machines have latency functions $\ell_i(x_i) = a_i x_i$, where $a_1 = \frac{1}{m-1}$, $a_2 = \cdots = a_m = 1$. From the proof of Theorem 1

$$PoA = m^{p-1}\sum_{i=1}^{m}\left(\frac{1}{a_i}\right)^p \Big/ \left(\sum_{i=1}^{m}\frac{1}{a_i}\right)^p.$$

Thus, for the instance,

$$\begin{aligned} PoA &= \frac{m^{p-1}\{(m-1)^p + (m-1)\}}{2^p(m-1)^p} \\ &= \frac{m^{p-1}}{2^p} + \frac{1}{2^p}\left(\frac{m}{m-1}\right)^{p-1} \ge \frac{m^{p-1}}{2^p}. \end{aligned}$$

VI. Stackelberg Scheduling

In this section, we investigate the situation in which a fraction of jobs, denoted by α, is centrally controlled. After the centrally controlled jobs are assigned to machines, the remaining jobs are assigned in a non-cooperative and selfish manner to result in a stable state. We will assign the centrally controlled jobs to machines in order to improve on the PoA. This assignment is referred to as Stackelberg scheduling.

Assume that $a_1 \le \cdots \le a_m$ and let an integer $k \ge 0$ be such that $\frac{k}{m} \le \alpha < \frac{k+1}{m}$. We consider a Stackelberg scheduling strategy $\mathcal{S}$ with an assignment $x^{\mathcal{S}}$. In $\mathcal{S}$, the $\frac{k}{m}$−fraction of the jobs are assigned to the machines $m-k+1, \dots, m$ as in the optimal assignment, that is, $x_i^{\mathcal{S}} = \frac{r}{m}$, for $i = m-k+1, \dots, m$. The remaining jobs are selfishly assigned.

First, we show that the remaining jobs are selfishly assigned only to machines $1, \dots, m-k$ after the $\frac{k}{m}$−fraction of the jobs are assigned to the machines $m-k+1, \dots, m$.

Lemma 3 For any $k\ (1 \le k \le m)$, in the Stackelberg scheduling strategy $\mathcal{S}$, if the $\frac{k}{m}$−rate of jobs are assigned to each of machines i, $m-k+1 \le i \le m$, then the selfishly behaved remaining jobs are assigned only to machines i, $1 \le i \le m-k$.

Proof: Let $\nu = m-k+1$. Then we see that the latency $\ell_i(x_i)$ of machine i, $\nu \le i \le m$, is given by $a_i \cdot \frac{r}{m}$. Assume that there are only machines i, $1 \le i \le \nu-1$, for the remaining $\left(1-\frac{k}{m}\right)r$−rate of jobs. Then since they selfishly behave, there is an $\mathcal{L}$ such that $\mathcal{L} = a_1x_1 = \cdots = a_{\nu-1}x_{\nu-1}$. Thus,

$$\mathcal{L} = \left(1-\frac{k}{m}\right)r \Big/ \sum_{i=1}^{\nu-1}\frac{1}{a_i}$$

since $\sum_{i=1}^{\nu-1} x_i = \left(1-\frac{k}{m}\right)r$. So it is sufficient to show that $\mathcal{L} \le \frac{a_\nu}{m}$, because the lowest latency of machines i, $\nu \le i \le m$, is $\frac{a_\nu r}{m}$. It holds since

$$\mathcal{L} = \frac{\frac{\nu-1}{m}r}{\sum_{i=1}^{\nu-1}\frac{1}{a_i}} \le \frac{\frac{\nu-1}{m}r}{(\nu-1)\frac{1}{a_{\nu-1}}} = \frac{a_{\nu-1}r}{m} \le \frac{a_\nu r}{m}.$$

Here we can estimate the PoA of the assignment produced by the Stackelberg scheduling strategy $\mathcal{S}$.

Theorem 3 For the Stackelberg scheduling strategy $\mathcal{S}$ on m machines with latency functions $\ell_i(x_i) = a_i x_i$, $a_1 \le \cdots \le a_m$, and for the integer $k\ (0 \le k \le m)$ such that $\frac{k}{m} \le \alpha < \frac{k+1}{m}$,

$$PoA = \frac{C(x^{\mathcal{S}})}{C(x^*)}$$

$$\le \begin{cases} \min\left\{(m-k)^{p-1}, \frac{k}{m} + \left(1-\frac{k}{m}\right)\left(\frac{a_{m-k}}{a_1}\right)^p\right\} & \text{if } 0 \le k < m, \\ 1 & \text{if } k = m. \end{cases}$$

Proof: When $k = m$, it is obvious. So we assume that $0 \le k < m$. In $\mathcal{S}$, the $\frac{r}{m}$−rate of jobs are assigned to each of machines i, $m-k+1 \le i \le m$.

From Lemma 3, the remaining $\left(1-\frac{k}{m}\right)r$ –rate of jobs are selfishly assigned only to machines i, $0 \le i \le m-k$. So we can see that $x_i^S = \frac{1}{a_i}\frac{(1-\frac{k}{m})r}{\sum_{i=1}^{m-k}\frac{1}{a_i}}$, for $0 \le i \le m-k$. Thus,

$$
\begin{aligned}
PoA &= \frac{\sum_{i=m-k+1}^{m}\left(\frac{r}{m}\right)^p + \sum_{i=1}^{m-k}\left(\frac{1}{a_i}\right)^p\left(\frac{(1-\frac{k}{m})r}{\sum_{i=1}^{m-k}\frac{1}{a_i}}\right)^p}{\frac{r^p}{m^{p-1}}} \\
&\le \frac{k}{m} + m^{p-1}\left(1-\frac{k}{m}\right)^p = \frac{k+(m-k)^p}{m} \\
&= 1 + \frac{m-k}{m}\left((m-k)^{p-1}-1\right) \\
&\le (m-k)^{p-1}
\end{aligned}
$$

Also from the above equation, we obtain another upper bound as follows:

$$
\begin{aligned}
PoA &= \frac{k}{m} + \frac{m^{p-1}\left(1-\frac{k}{m}\right)^p}{\left(\sum_{i=1}^{m-k}\frac{1}{a_i}\right)^p}\sum_{i=1}^{m-k}\left(\frac{1}{a_i}\right)^p \\
&\le \frac{k}{m} + \frac{m^{p-1}\left(1-\frac{k}{m}\right)^p}{(m-k)^p\left(\frac{1}{a_{m-k}}\right)^p}(m-k)\left(\frac{1}{a_1}\right)^p \\
&= \frac{k}{m} + \left(\frac{m}{m-k}\right)^{p-1}\left(1-\frac{k}{m}\right)^p\left(\frac{a_{m-k}}{a_1}\right)^p \\
&= \frac{k}{m} + \left(1-\frac{k}{m}\right)\left(\frac{a_{m-k}}{a_1}\right)^p.
\end{aligned}
$$

REFERENCES

[1] S. Aland, D. Dumrauf, M. Gairing, B. Monien, and F. Schoppmann, "Exact price of anarchy for polynomial congestion games", *Pro. 23rd Int. Symp. on Theoretical Aspects of Computer Science*, pp. 218–229, 2006.

[2] B. Awerbuch, Y. Azar, and A. Epstein, "The price of routing unsplittable flow", *Proc. 37th ACM Symp. on Theory of Computing*, pp. 57–66, 2005.

[3] B. Awerbuch, Y. Azar, E. Grove, M. Kao, P. Krishnan, and J. Vitter, "Load Balancing in the L_p norm", *Proc. 36th Annual Symp. on Foundation of Computer Science*, pp. 383–391, 1995.

[4] Y. Azar, "On-line load balancing", *Online Algorithms- The State of the Art*, pp. 178–195, 1998.

[5] G. Christodoulou and E. Koutsoupias, "The price of anarchy of finite congestion games", *Proc. 37th ACM Symp. on Theory of Computing*, pp. 67–73, 2005.

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

[7] M. Garing, T. Lucking, M. Mavronicolas, B. Monien, and M. Rode, "Nash equilibria in discrete routing games with convex latency functions", *Proc. 31st Int. Col. on Automata, Languages and Programming*, pp. 645–657, 2004.

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

[9] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria", *Pro. 16th Int. Symp. on Theoretical Aspects of Computer Science*, pp. 404–413, 1999.

[10] T. Lucking, M. Mavronicolas, B. Monien, and M. Rode, "A new model for selfish routing", *Pro. 21th Int. Symp. on Theoretical Aspects of Computer Science*, pp. 547–558, 2004.

[11] T. Roughgarden, "The price of anarchy is independent of the network topology", *Journal of Computer and System Science*, Vol. 67(1), pp. 341–364, 2003.

[12] T. Roughgarden, "Stackelberg scheduling strategies", *SIAM Journal on Computing*, Vol. 33(2), pp. 332–350, 2004.

[13] T. Roughgarden and E. Tardos, "How bad is selfish routing ?", *Journal of the ACM*, Vol. 49(2), pp. 236–259, 2002.

[14] J. G. Wardrop, "Some theoretical aspects of road traffic research", *Proc.of the Institute of Civil Engineers, Pt. II*, Vol. 1, pp. 325–378, 2002.

Jae-Hoon Kim
Received his B.S. degree in Mathematics from Sogang University in 1994, his M.S. degree in Mathematics from KAIST in 1996, and his Ph.D. degree in Computer Science from KAIST in 2003. He is currently an associate professor at Division of Computer Engineering, Pusan University of Foreign Studies. His research interests include on-line algorithms, scheduling, and computational geometry.