Worst-Case Nash Equilibria in Restricted Routing*

Pinyan Lu[1] and Changyuan Yu[1]

Institute for Theoretical Computer Science, Tsinghua University
Beijing, 100084, P. R. China
{lpy, yucy05}@mails.tsinghua.edu.cn

Abstract. We study a restricted related model of the network routing problem. There are m parallel links with possibly different speeds, between a source and a sink. And there are n users, and each user i has a traffic of weight w_i to assign to one of the links from a subset of all the links, named his/her allowable set. We analyze the *Price of Anarchy* (denoted by PoA) of the system, which is the ratio of the maximum delay in the worst-case Nash equilibrium and in an optimal solution. In order to better understand this model, we introduce a parameter λ for the system, and define an instance to be λ-*good* if for every user, there exist a link with speed at least $\frac{s_{max}}{\lambda}$ in his/her allowable set. In this paper, we prove that for λ-good instances, the Price of Anarchy is $\Theta\big(\min\{\frac{\log \lambda m}{\log\log \lambda m}, m\}\big)$. We also show an important application of our result in coordination mechanism design for task scheduling game. We propose a new coordination mechanism, *Group-Makespan*, for unrelated selfish task scheduling game. Our new mechanism ensures the existence of pure Nash equilibrium and its PoA is $O\big(\frac{\log^2 m}{\log\log m}\big)$. This result improves the best known result of $O(\log^2 m)$ by Azar, Jain and Mirrokni in [2].

1 Introduction

Network routing is one of the most important problems in the network management. In most networks, especially in a large-scale network like internet, it is unlikely that there is a centralized controller who can coordinate the behavior of all the users in the network. In such situations, every user in the network decides how to rout his/her traffic, aware of the congestion caused by other users. Users only care about the delay they suffer, and their selfish behavior often leads the whole network to a suboptimal state. Recently, researchers start to investigate the performance degradation due to the lack of the coordination for the users.

In the model first studied by Kautsoupias and Papadimitriou [9], there are m identical parallel links from the same origin to the same destination. There

* Supported by the National Natural Science Foundation of China Grant 60553001 and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

are n users, and each with a traffic of weight w_i. We assume that the traffic of each user can not be split and as a result each user chooses exactly one link. After all the users choose their links, the delay of a link is equal to the total weight of the traffics on it, and the delay a user suffers is equal to the delay of the link he chooses. The performance of the system we consider here is the maximum delay of all the links. We are mainly interested in stable states, where no user can decrease his delay by unilaterally changing his choice. In game theory, such a state is also called a *Nash equilibrium*. In order to measure the performance degradation, they compared the performance of Nash equilibrium with the optimal solution when there is centralized coordination. In particular, we analyze the *Price of Anarchy* (PoA for short) of the system, which is defined to be the performance ratio between the worst-case Nash equilibrium and an optimal solution. In [9], Kautsoupias and Papadimitriou showed that the PoA of that system is at most $2 - 1/m$.

Since then, a lot of research works have be done along this line. There are mainly two generalized models of this problem which are well studied. One model is routing with related links, where different links may have different speeds and the delay of a link is equal to the total weight on this link over its speed. In this uniform related model, Czumaj and Vöcking proved that the PoA is $\Theta\left(\frac{\log m}{\log\log m}\right)$ [4]. The other model is routing with restricted links, where each user i is only allowed to choose links from a subset S_i of all the links. However the links are still identically in the sense that the speed of each link is the same. In this restricted model, Awerbuch et al. proved that the PoA is also $\Theta\left(\frac{\log m}{\log\log m}\right)$ [1].

In light of these results, one may conjecture that the common extension of these two models, where the links are both related and restricted, also has a PoA of $\Theta\left(\frac{\log m}{\log\log m}\right)$. In fact, this model was studied by Gairing et al. in [7], and they showed that the PoA of this problem can be as large as $m-1$. However, in their bad instance demonstrating the lower bound of $m-1$, some users can only use extremely slow links (with speed less than $\frac{s_{\max}}{(m-1)!}$, where $s_{\max}$ is the largest speed). This is a little artificial and unlikely to appear in the real world. So in order to better understand this model, we introduce a property called λ-*goodness* for the system. An instance is called λ-good if and only if every user can at least use a link with speed no less than $\frac{s_{\max}}{\lambda}$. Now in our notation, the result in [7] says that the PoA can be as large as $m-1$ when the system is only $(m-1)!$-good. So what is the exact relation between the PoA and the λ-goodness of a system? In this paper, we answer this question completely by giving a tight bound for the PoA of a λ-good system in term of λ.

Theorem 1. *For λ-good instances, the price of anarchy is* $\Theta\big(\min\{\frac{\log \lambda m}{\log\log \lambda m}, m\}\big)$.

In the proof of Czumaj and Vöcking for related links, they essentially used the property of uniform related, which means that each link has a fixed speed and all the users can choose it. And in the proof of Awerbuch et al. for restricted links, they essentially used the property of identical, which means that all the links have the same speed. In our extended model, namely restricted related links, none of the two properties hold and as a result none of their technique can be adopted to analyze the PoA of the new model directly. In this paper, we use a new proof approach. We calculate the delay of links interval by interval, obtain some recursive relations between them based on the property of Nash equilibrium, and finally we are able to derive a bound of the maximum delay in the system.

Our result also has an important application in task scheduling game with coordination mechanism. Task scheduling can be viewed as another model for routing problem by treating the links as machines, the traffics as tasks, the delay of a user as the completion time of his/her task, and the delay of the system as the makespan of the system. Then we have scheduling with identical machines, related machines, and restricted machines corresponding to the above three models of routing problems. Further more, we also have a more general model, called scheduling with unrelated machines, in which each machine may have different speeds for different tasks. An instance of scheduling unrelated machines is denoted by a matrix $t = (t_{ij})$, where t_{ij} denotes the processing time that machine j needs for task i. In this language, when each machines uses the *Makespan* policy, i.e. to process its tasks in such a parallel way that all of them are completed at the same time, the task scheduling game is essentially the same as the routing problem. However, as observed by Christodoulou, Koutsoupias and Nanavati in [3], the scheduling policies of the machines may affect the choices of the users, and hence the PoA of the system. So they considered the problem of designing a set of local scheduling policies such that the PoA of the system is small. Such a set of scheduling policies are called coordination mechanism, and the PoA of the system with a coordination mechanism is also called the PoA of this mechanism.

Using our main result, we propose a new coordination mechanism, named *Group-Makespan* mechanism, for scheduling unrelated machines. This Group-Makespan mechanism ensures the existence of a pure Nash equilibrium and its PoA is $O\left(\frac{\log^2 m}{\log\log m}\right)$, improving the best known result $O(\log^2 m)$ by Azar, Jain and Mirrokni in [2].

2 Preliminaries and Notations

In this section, we define our problem formally. There are m independent links from certain origin to destination, and n independent users. We use $[m]$ and $[n]$ to denote the link set $\{1, \cdots, m\}$ and user set $\{1, \cdots, n\}$ respectively. Each link $j \in [m]$ has a speed s_j and w.l.o.g, we assume $s_1 \geq s_2 \geq \cdots \geq s_m$. Each user $i \in [n]$ has a traffic of weight w_i, which can only be assigned to a link from a set $S_i \subseteq [m]$. We use $< w, s, \mathcal{S} >$ to denote an instance of the problem, where $w = (w_1, \cdots, w_n)$, $s = (s_1, \cdots, s_m)$ and $\mathcal{S} = \{S_1, \cdots, S_n\}$ denote the weights, speeds and allowable link sets. We introduce the property of λ-goodness for a instance $< w, s, \mathcal{S} >$.

Definition 1. (λ-Goodness) *An instance $< w, s, \mathcal{S} >$ is λ-good if and only if the following condition holds: for any user $i \in [n]$, there exists a machine $j \in S_i$ such that the speed s_j is at least s_1/λ.*

We consider pure strategies for users, and each user's strategy is to decide which link to assign his/her traffic. We use $a = (a_1, \cdots, a_n) \in S_1 \times \cdots \times S_n$ to denote a combination of all users' strategies, where user i selects a link $a_i \in S_i$. We also use a_{-i} to denote the strategies of all the other users except user i. In a state a, the delay of link j, denoted by l_j^a, is the total weights on it over its speed, and the delay of the system, denoted by l^a, is the maximum delay over all the links. That is $l_j^a = \frac{1}{s_j} \sum_{i:a_i=j} w_i, \quad l^a = \max_j l_j^a$.

We consider the optimum when there is centralized coordination, that is, the minimal delay of the system over all the possible states. We use *opt* to denote the optimum as well as an optimal solution.

We assume the users are all non-cooperative and each one wishes to minimize his/her own cost, without any regard to the performance of the system. The cost of user i in a state a is the delay of link a_i and we use c_i^a to denote it. We have $c_i^a = l_{a_i}^a$.

Now we define the Nash equilibria of the system formally.

Definition 2. (Nash Equilibrium) *A state a is called a Nash Equilibrium (NE for short)of the system if and only if no user can decrease his/her cost by unilaterally changing a link. That is, for any user $i \in [n]$, any strategy $a_i' \in S_i$ and $a' = (a_{-i}, a_i')$, we have $c_i^a \leq c_i^{a'}$.*

For any instance of the problem, pure Nash Equilibrium always exists. The proof of this fact is using a quite common method with an elegant potential function, which is pointed out in several places(see [5] for example).

Theorem 2. (Existence of Nash Equilibrium) *For $\lambda \geq 1$ and any λ-good instance $< w, s, \mathcal{S} >$, there exists a Nash Equilibrium state a of it.*

To compare the performance of Nash Equilibrium with the optimum, we give the definition of *Price of Anarchy.*

Definition 3. (Price of Anarchy) *For instance of restricted routing problem, the Price of Anarchy (PoA for short) is defined as the performance ratio between the worst-case Nash equilibrium and the optimal solution. That is*

$$\text{PoA} = \max_{\substack{a \in S_1 \times \cdots \times S_n \\ a \text{ is a } NE}} \frac{l^a}{opt} \quad .$$

And for any family of instances, its Price of Anarchy is defined to be the largest PoA among all its possible instances.

3 PoA of λ-Good Restricted Routing

In this section, we prove our main result Theorem 1. If $\lambda > (m-1)!$, Gairing et al. gave a tight bound $\Theta(m)$ [7]. So in this section, we always assume $\lambda \leq (m-1)!$

and prove that the PoA of the family of λ-good instances is $\Theta\left(\frac{\log \lambda m}{\log\log \lambda m}\right)$. We only give the proof for the upper bound and omit the tight example here.

Theorem 3. **(Upper Bound)** *Given any λ-good instance $< w, s, \mathcal{S} >$ and a state $a \in S_1 \times \cdots \times S_n$ which is a Nash equilibrium, delay of the system l^a is at most $opt \cdot O\left(\frac{\log \lambda m}{\log\log \lambda m}\right)$.*

For notational simplicity, we scale the speeds and weights such that $s_1 = 1$ and *opt* $= 1$. We also define several notations used in the proof. For any $k \in R^+$ and $j \in [m]$, let $W_j^k = \max\{l_j^a - k, 0\} \cdot s_j$ and $W^k = \sum_{j\in[m]} W_j^k$. Especially, we use $W_j = W_j^0$ to denote the total weight assigned to link j, and $W = W^0$ to denote the total weight of all the users. Fix an optimal solution *opt*, let O_j be the set of users assigned to link j in *opt*. We also define O_j^k to be the set of users who choose link j in *opt* and have costs at least k, that is, $O_j^k = \{i \in O_j, c_i^a \geq k\}$.

Our proof of the upper bound theorem comes from the following lemmas. In Lemma 1, we give a initial condition of W^k and this is the only point we use the condition that the instance is λ-good. Then Lemma 2 and Lemma 3 give recursive relations between W^ks, which basically says that W^k should increase significantly when k become small. So we can bound the total weight W from below in terms of makespan l^a and λ. On the other hand, the total weigh is bounded from above by m. Putting things together, we can bound l^a.

Lemma 1. *For any λ-good instance and any Nash equilibrium a, we have $W^{l^a-2} \geq \frac{1}{\lambda}$.*

Proof. Consider a link whose delay achieves l^a, say link j^*. Let i be a user on link j^*, and let link $j \in S_i$ has the maximum speed in S_i. Now if $j = j^*$, we have $l_j^a = l^a$. If $j \neq j^*$, since a is a Nash equilibrium, i cannot decrease his/her cost by changing from link j^* to link j. We have $l^a = c_{j^*}^a \leq l_j^a + \frac{w_i}{s_j}$.

As in the optimal solution, task i can only be assigned to a link from S_i, whose speed is at most s_j, we have $w_i/s_j \leq$ *opt* $= 1$. Therefore, we have $l_j^a \geq l^a - 1$. So no matter whether $j = j^*$ or not, we have $l_j^a \geq l^a - 1$, hence

$$W^{l^a-2} \geq W_j^{l^a-2} \geq 1 \cdot s_j \geq \frac{1}{\lambda},$$

where the last inequality is because the instance is λ-good. □

Lemma 2. *For any Nash equilibrium a and $0 \leq k \leq l^a - 2$, we have $W^k \geq \frac{l^a}{l^a-(k+2)} W^{k+2}$.*

Proof. Firstly, we want to prove that $W_j^k \geq \sum_{i\in O_j^{k+2}} w_i$. If O_j^{k+2} is empty, then we are done. Otherwise, for any task $i \in O_j^{k+2}$, $c_i^a \geq k+2$, by the definition of Nash equilibrium, we have

$$k + 2 \leq c_i^a \leq l_j^a + w_i/s_j \leq l_j^a + 1.$$

The last inequality is because that the task i is assigned to link j in *opt*. Therefore, $l_j^a \geq k+1$ and $W_j^k \geq 1 \cdot s_j \geq \sum_{i \in O_j} w_i \geq \sum_{i \in O_j^{k+2}} w_i$. Noticing that $\bigcup_j O_j^k = \{i : c_i^a \geq k\}$, we can bound W^k as follows:

$$W^k = \sum_{j \in [m]} W_j^k \geq \sum_{j \in [m]} \sum_{i \in O_j^{k+2}} w_i = \sum_{i: c_i^a \geq k+2} w_i = \sum_{j: l_j^a \geq k+2} W_j \tag{1}$$

By the definition of W_j and W_j^{k+2}, for any j, $l_j^a > k+2$, we have:

$$W_j = \frac{l_j^a}{l_j^a - (k+2)} W_j^{k+2} \geq \frac{l^a}{l^a - (k+2)} W_j^{k+2} \tag{2}$$

The last inequality is because the function $f(x) = \frac{x}{x-(k+2)}$ is monotone decreasing when $x > k+2$ and for all j, we have $l_j^a \leq l^a$.

So from (1) and (2), we have:

$$W^k \geq \frac{l^a}{l^a - (k+2)} \sum_{j: l_j^a > k+2} W_j^{k+2} = \frac{l^a}{l^a - (k+2)} W^{k+2}$$

From lemma 1 and lemma 2, we have recursive relation about W^k and an initial condition. These ensures us to prove an upper bound on l^a, which is $O(\log \lambda m)$. There is a little gap between our expected bound. The reason is that in the above estimation in (2), we bounded all the l_j^a from above by l^a. This is a little weak since there cannot be too many links with large l_j^a. The following lemma uses a more careful estimation, and explores a recursive relation between W^k,W^{k+2}, and W^{k+4}, which helps us to obtain a better bound on l^a.

Lemma 3. *For any λ-good instance and any Nash equilibrium a, we have $W^k \geq \frac{k+6}{4}(W^{k+2} - 2W^{k+4})$.*

Proof. First, we omit some links in the summation of the last term in (1), then

$$W^k \geq \sum_{j: l_j^a > k+2} W_j \geq \sum_{j: k+6 \geq l_j^a > k+2} W_j.$$

Now, the estimation occurred in (2) can be more tight: for any link j such that $k+6 \geq l_j^a > k+2$, we have

$$W_j = \frac{l_j^A}{l_j^a - (k+2)} W_j^{k+2} \geq \frac{k+6}{k+6-(k+2)} W_j^{k+2} = \frac{k+6}{4} W_j^{k+2}$$

So, we can bound W^k as

$$W^k \geq \frac{k+6}{4} \sum_{j: k+6 \geq l_j^a > k+2} W_j^{k+2} = \frac{k+6}{4} \left(W^{k+2} - \sum_{j: l_j^a > k+6} W_j^{k+2} \right) \tag{3}$$

For $\forall j$, $l_j^a > k+6$, we have

$$W_j^{k+2} = (l_j^a - (k+2)) \cdot s_j \qquad \text{and} \qquad W_j^{k+6} = (l_j^a - (k+6)) \cdot s_j,$$

hence $2W_j^{k+4} = W_j^{k+2} + W_j^{k+6}$. Using this equality, we can bound the negative term in (3) as follows:

$$\sum_{j:l_j^a > k+6} W_j^{k+2} \le \sum_{j:l_j^a > k+6} 2W_j^{k+4} \le 2W^{k+4}.$$

Substituting this into (3), and we finish the proof. □

Putting things together, we have the proof of Theorem 3.
Proof of Theorem 3: Let $k_0 = \lfloor \frac{l^a}{6} \rfloor$. For any $k \ge l^a - 2k_0 \ge \frac{2l^a}{3}$, we have:

$$\begin{aligned} W^k &\ge \frac{k+6}{4}\left(W^{k+2} - 2W^{k+4}\right) \\ &\ge \frac{k+6}{4}\left(W^{k+2} - 2 \cdot \frac{l^a - (k+4)}{l^a} W^{k+2}\right) \\ &= \frac{2(k+4) - l^a}{4l^a} \cdot (k+6) W^{k+2} \\ &\ge \frac{2(\frac{2l^a}{3} + 4) - l^a}{4l^a} \cdot (\frac{2l^a}{3} + 6) W^{k+2} \\ &\ge \frac{l^a}{18} W^{k+2} \end{aligned}$$

The first inequality is by lemma 3 and the second inequality is by lemma 2.
so using this recursive relation and lemma 1, we have:

$$W^{l^a - 2k_0} \ge W^{l^a - 2} \cdot \left(\frac{l^a}{18}\right)^{\frac{l^a}{6}} \ge \frac{1}{\lambda} \cdot \left(\frac{l^a}{18}\right)^{\frac{l^a}{6}}.$$

Since $\forall j, s_j \le s_1 = 1$, and $opt = 1$, we have $W \le opt \cdot \sum_j s_j \le m$. By $W \ge W^{l^a - 2k_0}$, we have $\left(\frac{l^a}{18}\right)^{\frac{l^a}{6}} \le \lambda m$. Since the solution to the equation $x^x = y$ is $x = \Theta\left(\frac{\log y}{\log \log y}\right)$, we can obtain that l^a is at most $O\left(\frac{\log \lambda m}{\log \log \lambda m}\right)$. □

4 An Application in Coordination Mechanism

In this section, we see an application in coordination mechanism design for selfish task scheduling game. We give the high level ideas of our new mechanism. This new mechanism is inspired by the mechanism *Split & Shortest* in [2]. Given an instance t for scheduling with unrelated machines, we can define $t_i = \min_{j \in [m]} t_{ij}$ as the weight of task i, and define the speed s_{ij} of a machine

j with respect to a task i as $s_{ij} = t_i/t_{ij}$, namely the minimum running time of task i on all the machines over the running time of task i on machine j. In our Group-Makespan mechanism, every machine simulates $\log m$ sub machines and submachine k of machine j only run those tasks i for which machine j has speed $s_{ij} \in [2^{-k}, 2^{-k+1})$. We artificially delay a task so that the k-th sub machines of different machines all have fixed speed 2^{-k}. Each machine simulates its sub machines by round-robin, and for each submachine we use the Makespan scheduling policy. In the submachine level, each submachine has a fixed speed, and a task can only be assigned to some of the sub machines. So it becomes a problem of scheduling with restricted related machines. Further more, all the instance obtained in this way have a very good structure, namely it is 1-good. Therefore in the submachine level, the PoA is bounded by $\Theta\left(\frac{\log m}{\log\log m}\right)$. Since each machine has to simulate $\log m$ machines all the time, this may loss a factor of at most $\log m$.

We give the theorem as following, and omit the formal definition of the Group-Makespan mechanism and the proof of this theorem due to the space limitation. Readers may see [2] for idea of the submachine and related analysis.

Theorem 4. *The Group-Makespan mechanism for scheduling m unrelated machines ensures the existence of pure Nash equilibria, and the PoA of the task scheduling game with this mechanism is $O\left(\frac{\log^2 m}{\log\log m}\right)$.*

References

1. B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. *Tradeoffs in worst-case equilibria.* Theoretical Computer Science, 361:200 – 209, 2006.
2. Y. Azar, K. Jain, and V. Mirrokni. *(Almost) optimal coordination mechanisms for unrelated machine scheduling.* In Proceedings of SODA 2008, pages 323 – 332.
3. G. Christodoulou, E. Koutsoupias, and A. Nanavati. *Coordination mechanisms.* In Proceedings of ICALP 2004, pages 345 – 357.
4. A. Czumaj and B. Vöcking. *Tight bounds for worst-case equilibria.* In Proceedings of SODA 2002, pages 413 – 420, 2002.
5. D. Fotakis, S. Kontogiannis, E. Koutsoupias, M. Mavronicolas and P. Spirakis. *The Structure and Complexity of Nash Equilibria for a Selfish Routing Game.* In Proceedings of ICALP 2002, pp. 123 – 134.
6. M. Gairing, T. Lucking, M. Mavronicolas, and B. Monien. *Computing nash equilibria for scheduling on restricted parallel links.* In Proceedings of STOC 2004, pages 613 – 622, 2004.
7. M. Gairing, T. Lucking, M. Mavronicolas, and B. Monien. *The Price of Anarchy for Restricted Parallel Links.* Parallel Processing Letters 16(1): 117 – 132 (2006).
8. N. Immorlica, L. Li, V.Mirrokni, and A. Schulz. *Coordination mechanisms for selfish scheduling.* In Proceedings of WINE 2005, pages 55 – 69.
9. E. Koutsoupias and C. H. Papadimitriou. *Worst-case equilibria.* In Proceedings of STACS 1999, pages 404 – 413.
10. M. Mavronicolas and P. Spirakis. *The Price of Selfish Routing.* Algorithmica 48(1):91 – 126, 2007.