Coordination mechanisms*

Elias Koutsoupias
University of Athens and University of California, Los Angeles
elias@di.uoa.gr

Akash Nanavati
University of California, Los Angeles
akash@cs.ucla.edu

October 27, 2003

Abstract

We introduce the notion of coordination mechanisms to improve the performance in systems with independent selfish agents. The quality of a coordination mechanism is measured by its price of anarchy—the worst-case performance of a Nash equilibrium over the (centrally controlled) social optimum. We give upper and lower bounds for the price of anarchy for selfish resource allocation and analyze the case of selfish routing on a simple network.

1 Introduction

The price of anarchy, whose study initiated in [5], captures the deterioration of performance of a system when resources are allocated by selfish agents. The study of the price of anarchy seems to capture the lack of coordination between independent selfish agents as opposed to the lack of information (competitive ratio) or the lack of computational resources (approximation ratio). Unlike however the competitive and approximation ratios, the price of anarchy failed to suggest a framework in which coordination algorithms for selfish agents should be designed and evaluated.

In this work we attempt to remedy the situation. We study coordination policies (mechanisms) that attempt to minimize price of anarchy. These mechanisms are the parallel of online or approximation algorithms.

Consider for example the problem studied in [5]. There is a simple network of m parallel links and a set of n selfish users. Each user i has some traffic load w_i and wants to use exactly one of the parallel links to route it through. When the users act selfishly at a Nash equilibrium the use of the links may be suboptimal. The worst-case ratio of the maximum latency at a Nash equilibrium over the optimal allocation is termed price of anarchy or coordination ratio. It is known that the price of anarchy for this system is $\Theta(\log m / \log\log m)$ [5, 1, 4]. The question is "*How can we improve the price of anarchy?*"; what mechanisms one can use to improve the overall system performance even in the face of selfish behavior? We approach this question from the following point of view (although there may be other interesting ways to approach it): We want to design "scheduling" policies in each link to improve the price of anarchy (thus essentially forcing the agents to "cooperate" willingly). The designer of the system selects a mechanism —in this case scheduling policy— that minimizes

*Research supported in part by the IST program and by NSF.

the price of anarchy. The question is what kind of mechanisms are available to the designer. We want to consider mechanisms that do not require additional resources and do not alter the nature of the system. It is natural to consider local scheduling policies in which the schedule on each link depends *only on the loads of the link*. Otherwise, an obvious solution would be to force an optimal allocation to each link. It is also natural to allow each link to give priorities to the loads and introduce delays. A set of scheduling policies will be called a *coordination mechanism*.

Another problem whose price of anarchy has been studied intensively is the selfish routing problem [16]. How can we improve the performance in this case? It is natural to allow the edges to introduce delays that depend on the traffic through it. This will "slow down" the network but it will give incentives to players to coordinate.

1.1 Results

To study this and similar questions, we introduce a unifying framework: the notion of coordination models and coordination mechanisms. Using this framework we study coordination mechanisms for the two mentioned problems: the selfish resource allocation problem and the selfish routing problem. For the resource allocation problem we give a coordination mechanism that reduces the price of anarchy from $\Theta(\log m/ \log\log m)$ to a small constant, $4/3 - 1/3m$; we also give some weak lower bound. For the selfish routing we show coordination mechanisms for specific examples such as the Braess' paradox network, and argue that coordination mechanisms can improve the price of anarchy only if they use non-continuous delay functions; this is in sharp contrast to assumptions of continuity in the work of Roughgarden and Tardos [16]. We give coordination mechanisms for the network of two parallel links with arbitrary linear delay functions which improve the price of anarchy from 4/3 to less than 1.2.

1.2 Related work

Mechanisms to improve coordination of selfish agents is not a new idea and we only mention here work that directly relates to our approach. A central topic in game theory [10] is the notion of mechanism design in which the players are paid (or penalized) to "coordinate". The differences between mechanism design and the coordination mechanism model are numerous. The most straightforward comparison can be exhibited in the selfish routing problem: both aim at improving coordination, but mechanism design can be seen as a way to introduce *tolls*, while coordination mechanism is a way to introduce *traffic lights*. Also, the algorithmic and communication issues involved in mechanism design seem to be completely different than the ones involved in coordination mechanisms [9, 8, 12].

The idea of designing games to improve coordination appears also in the work of Korilis, Lazar, and Orda [3] but there the goal is to design games with a unique Nash equilibrium; there is no attempt to compare it with the potential optimum.

A problem that relates to coordination mechanisms for selfish routing, and studied in [14], asks to find a subnetwork of a given network that has optimal price of anarchy for a given total flow. This can be also cast as a special case of coordination mechanisms that allow either a given specific delay function or infinity (and fixed total flow).

2 Model

2.1 Coordination models

A Coordination Model is a tuple $(N, M, (\Sigma_i)_{i\in N}, (C^j)_{j\in M})$ where $N = \{1, \ldots, n\}$ is the set of players, M is a set of facilities, Σ_i is a collection of strategies for player i: a strategy $A_i \in \Sigma_i$ is a set of facilities, and finally C^j is a collection of cost functions associated with facility j: a cost function $c^j \in C^j$ is a function that takes as input a set of loads, one for each player that uses the facility, and outputs a cost to each participating player; it also outputs one additional cost, the facility's cost. More precisely, c^j is a cost function from R^N to $R^{N\cup\{0\}}$ with the property $c^j_i(w_1, \ldots, w_{i-1}, 0, w_{i+1}, \ldots, w_n) = 0$ which expresses exactly the property that players incur no cost when they don't use the facility. The facility's cost is given by $c^j_0(w_1, \ldots, w_n)$.

For the problems that we discuss in this paper the facility cost is always the maximum cost of the players that use the facility: $c^j_0(w_1, \ldots, w_n) = \max_{i=1,\ldots,n} c^j_i(w_1, \ldots, w_n)$.

Example 1 *The coordination model for* **selfish resource allocation** *is as follows: $N = \{1, \ldots, n\}$ is the set of players, $M = \{1, \ldots, m\}$ the set of facilities is a set of machines or links, all Σ_i's consists of all singleton subsets of M, $\Sigma_i = \{\{1\}, \ldots, \{m\}\}$, i.e., each player uses exactly one facility, and the cost functions are the possible finish times for scheduling the loads on a facility. More precisely, a function c^j is a cost function for facility j if for every set of loads $(w_1, \ldots, w_n)$ and every subset S of N, the maximum finish time of the players in S must be at least equal to the total length of the loads in S: $\max_{i\in S} c^j_i(w_1, \ldots, w_n) \geq \sum_{i\in S} w_i$. Notice that a facility is allowed to order the loads arbitrarily and introduce delays, but it cannot speed up the execution. As an example, a facility could schedule two loads w_1 and w_2 so that the first load finishes at time $w_1 + w_2/2$ and the second load at time $2w_1 + w_2$. Finally, the social cost for a facility is the last finish time of all players that use the facility, i.e., $c^j_0(w_1, \ldots, w_n) = \max_{i=1,\ldots,n} c^j_i(w_1, \ldots, w_n)$.*

The notion of coordination model is an appropriate generalization of the congestion model[13, 7]. In the congestion model the cost functions (a) have no input loads, i.e., $c^j : \{0, 1\}^N \mapsto R^{N\cup\{0\}}$, (b) are the same for all players, i.e., c^j_i is independent of i, and finally (c) they are symmetric, i.e., their value depends only the number of participating players: $c^j_i(w_1, \ldots, w_n) = d^j(\sum_{i=1}^m w_i)$ for some d^j. In the congestion model there is no notion of facility cost, but we can naturally define it to be equal to the cost of each player who uses the facility (as was done in [17]): $c^j_0(w_1, \ldots, w_n) = \max_{i=1,\ldots,n} c^j_i(w_1, \ldots, w_n)$. We remark however that the congestion model corresponds to a particular game —there is only one cost function for each facility— while in our model there is a collection of games —a set of cost functions for each facility.

Example 2 *We can extend a congestion model to a coordination model in a natural way, by allowing each facility to introduce delays. In particular, the set of cost functions of facility j consists of all symmetric functions $c^j_i(w_1, \ldots, w_n) = d'^j(\sum_{i=1}^m w_i) \geq d^j(\sum_{i=1}^m w_i)$. The facility cost is again the cost of each player who uses the facility. We will call this model the* **coordination model for congestion** *and we distinguish two cases: the symmetric case, in which we require the functions c^j_i to be the same for each player i, and the asymmetric case ([6] studies the extension of symmetric congestion games to the asymmetric case).*

Of special interest is the symmetric case of infinitely many players. Particularly, we will consider the **coordination model for routing** *which is based on the well-studied routing model of Wardrop [18, 16]. In this routing model, the set of facilities is the set of edges of a finite network, the set of strategies for a player i is the set of paths from a source s_i to a destination t_i, and the cost (delay) function c_e of an edge e depends on the amount of users (traffic) that use the edge. We define the* *coordination model for routing by allowing the edges to introduce delays that depend on their traffic: the set of cost functions on an edge e is the set of functions c'_e such that $c'_e(f) \geq c_e(f)$, for every f.*

2.2 Coordination mechanisms

The notion of coordination model sets the stage for an adversarial analysis of the deterioration in performance due to lack of coordination. The situation is best understood when we compare it with competitive analysis. The following table shows the correspondence.

Coordination model	$\leftrightarrow$	Online problem
Coordination mechanism	$\leftrightarrow$	Online algorithm
Price of anarchy	$\leftrightarrow$	Competitive ratio

It should be apparent from this correspondence that one cannot expect to obtain meaningful results for *every possible coordination model* in the same way that we don't expect to be able to find a unifying analysis of *every possible online problem*. Each particular coordination model that arises in "practice" or in "theory" should be analyzed alone. We now proceed to define the notion of coordination mechanism and its price of anarchy.

A *coordination mechanism* for a coordination model $(N, M, (\Sigma_i)_{i\in N}, (C^j)_{j\in M})$ is simply a set of cost functions, one for each facility. The simplicity of this definition may be misleading unless we take into account that the set of cost functions may be rich enough. A coordination mechanism is essentially a *distributed algorithm*; we select once and for all the cost functions for each facility, before the input is known. For example, for the coordination model for selfish resource allocation, a coordination mechanism is essentially a set of *local scheduling policies*, one for each machine; the scheduling on each machine depends only on the loads that use the machine. Fix a coordination mechanism $c = (c^1, \dots, c^m)$, a set of player loads $w = (w_1, \dots, w_n)$, and a set of strategies $A = (A_1, \dots, A_n) \in \Sigma_1 \times \cdots \times \Sigma_n$. Let $w^{j,A}$ be the set of loads that use facility j: $w_i^{j,A}$ is w_i if $j \in A$ and 0 otherwise. We define its *social cost* $sc(w; c; A)$ as the maximum (or sometimes the sum) of the social cost of all facilities, i.e.,

$$sc(w; c; A) = \max_{j\in M} c_0^j(w^{j,A}) \tag{1}$$

We also define the *social optimum* $opt(w)$ for a given set of player loads w as the minimum social cost of all coordination mechanisms and all strategies in $\Sigma_1 \times \cdots \times \Sigma_n$

$$opt(w) = \inf_{c,A} sc(w; c; A) \tag{2}$$

It is important to notice that the definition of $opt(w)$ refers to the absolute optimum which is independent of the coordination mechanism. For example, for the coordination model of the selfish resource allocation, a coordination mechanism is allowed to slow down the facilities, but *the optimum $opt(w)$ is computed using the original speeds.*

To a coordination mechanism c and set of player loads w corresponds a game; *the cost of a player* is the sum of the cost of all facilities used by the player. Let $Ne(w; c)$ be the set of (mixed) Nash equilibria of this game. We define the *price of anarchy* (or coordination ratio) of a coordination mechanism c as the maximum over all set of loads w and all Nash equilibria E of the social cost over the social optimum.

$$\mathrm{PA}(c) = \sup_{w} \sup_{E\in Ne(w;c)} \frac{sc(w; c; E)}{opt(w)} \tag{3}$$

We can also define the price of anarchy of a coordination model. It is the minimum price of anarchy over all coordination mechanisms.

This naturally defines the following "game" between us, the designers of coordination mechanisms who strive for small price of anarchy, and a fictitious adversary who tries to embarrass us. *We, the designers, select a coordination mechanism c (essentially a distributed algorithm) among all the possible cost functions of the facilities. Then the adversary selects a set of loads w that maximizes the price of anarchy of c. We then compute the worst-case social cost among Nash equilibria and divide by the social optimum.*

Thus the situation is very similar to the framework of competitive analysis in online algorithms or the analysis of approximation algorithms. Online algorithms address the lack of information by striving to reduce the coordination ratio; approximation algorithms address the lack of sufficient computational resources by striving to reduce the approximation ratio. In a similar way, coordination mechanisms address the lack of coordination due to selfish behavior by striving to reduce the price of anarchy.

The analogy also helps to clarify one more issue: *Why do we need to minimize the price of anarchy and not simply the cost of the worst-case Nash equilibrium?* In the same way that it is not in general possible to have an online algorithm that minimizes the cost for *every input*, it is not in general possible to have a mechanism that minimizes the cost of the worst-case Nash equilibrium for *every possible game of the coordination model*.

Finally, the analogy helps to introduce the notion of *randomized coordination mechanism* which is simply a probability distribution over deterministic mechanisms. Randomization usually helps online algorithms and approximation algorithms because it seems to "confuse" the adversary. One should expect similar gains for randomized coordination mechanisms.

3 Selfish resource allocation

We now focus on the coordination model for selfish resource allocation (Example 1). There are n players with loads and m identical facilities (machines or links). The objective of each player is to minimize the finish time. The mechanism designer has to select and announce a scheduling policy on each facility once and for all (without the knowledge of the loads). The scheduling policy on each facility must depend only on its own loads (and not on loads allocated to the other machines).

Let's first consider the case of $m = 2$ facilities. In retrospect, the coordination mechanisms considered in [5] schedule the loads on each link in a random order intermixed or not (the latter is called the batch model). The price of anarchy of these mechanisms is $3/2$ for the first and even higher for the second. To illustrate the issues, we discuss first a simple coordination mechanism:

> The loads are ordered by size. If two or more loads have the same size, their order is the lexicographic order of the associated players. Then the first facility schedules its loads in order of *increasing* size while the second facility schedules its loads in order of *decreasing* size.

The mechanism aims to brake the symmetry of loads. With this mechanism, it is easy to see that the agent with the minimum load goes always to the first link. Similarly, the agent with the maximum load goes to the second link.

The following is not hard to show:

Proposition 1 *The above increasing-decreasing coordination mechanism has price of anarchy $4/3$. In particular, for $n = 3$ players, it has price of anarchy 1.*

To show for example that the price of anarchy of the mechanism is no better than $4/3$, consider 4 players with loads $1, 1, 2, 2$. Then there is a Nash equilibrium in which the first two loads go to the first link while the other two loads go to the second link (this happens to be a pure Nash equilibrium). Its price of anarchy is $4/3$.

It is natural to ask whether there is a better coordination mechanism for 2 facilities. Also, since this mechanism is not easily extendible to more than 2 facilities, is there a good mechanism for $m > 2$ facilities? Surprisingly the answer is positive to both questions. To motivate the better coordination mechanism consider the case of $n = m$ players each with load 1. Symmetric coordination mechanisms in which all facilities have the same scheduling policy have very large price of anarchy: The reason is that there is a Nash equilibrium in which each player selects randomly (uniformly) among the facilities; this is similar to the classical bins-and-balls random experiment, and the price of anarchy is the expected maximum: $\Theta(\log m/\log\log m)$.

It is clear that the large price of anarchy results when players "collide". Intuitively this can be largely avoided in pure equilibria. To make this more precise consider the case where all loads have distinct sizes and furthermore all partial sums are also distinct. Consider now the coordination mechanism for m machines where every machine schedules the jobs in decreasing order; furthermore to break the "symmetry" assume that machine i has a multiplicative delay $i\epsilon$ for each job and for some small $\epsilon > 0$. Then in the only Nash equilibrium the largest job goes to the first machine, the next job goes to second machine and so on; the next job in decreasing size goes to the machine with the minimum load. There is a small complication if the multiplicative delays $i\epsilon$ create some tie, but we can select small enough ϵ so that this never happens.

It should be clear that this is a mechanism with small price of anarchy. But what happens if the jobs are not distinct or the multiplicative delays $i\epsilon$ creates ties? We can avoid both these problems with the following coordination mechanism that is based on two properties:

- Each facility schedules the loads in decreasing order (using the lexicographic order to break any potential ties).
- The cost of each player is different on each facility. More concretely, the cost $c_i^j(w_1, \ldots, w_n)$ is a number whose representation in the m-ary system ends at $j \pmod m$. Let f_i^j be the sum of the loads greater or equal to w_i that were allocated on facility j. Then we define $c_i^j(w_1, \ldots, w_n)$ as the greatest number in $\left[0, f_i^j(1+\delta)\right)$ whose representation in the m-ary system ends at $j \pmod m$; δ is a fixed (small positive) parameter.

For example for $m = 10$ machines and $\delta = 0.01$, if a job of size $w_i = 1$ is first (greatest) on machine 7 it will not finish at time 1 but at time 1.007.

Theorem 1 *The above coordination mechanism for n players and m facilities has price of anarchy $4/3 - 1/3m + \delta$.*

Proof. There is only one Nash equilibrium: The largest load is "scheduled" first on every facility independently of the remaining loads, but there is a unique facility for which the players' cost is minimum. Similarly for the second largest load there is a unique facility with minimum cost independently of the smaller loads. In turn this is true for each load. Notice however that this is exactly greedy scheduling with the loads ordered in decreasing size. It has been analyzed in Graham's seminal work [2] where it was established that its approximation ratio is $4/3 - 1/3m$. Given that the total delay introduced by the δ terms increases the social cost by at most a factor of δ, we conclude that the price of anarchy is at most $4/3 - 1/3m + \delta$.

To see that this bound is tight we reproduce Graham's lower bound: Three players have load m and for each $k = m+1, \ldots, 2m-1$, two players have load k. The social optimal is $3m$ but the coordination mechanism has social cost $4m-1$ (plus some δ term). $\square$

Notice some additional nice properties of this coordination mechanism: there is a unique Nash equilibrium (thus players are easy to "agree") and it has low computational complexity. In contrast, computing Nash equilibria is potentially a hard problem —its complexity is in general open.

The above theorem shows that good coordination mechanisms reduce the price of anarchy from $\Theta(\log m / \log \log m)$ to a small constant. Is there a coordination mechanism with better price of anarchy than $4/3 - 1/3m$? We conjecture that the answer is negative. In particular, for $m = 2$ facilities, there are strong indications that the ratio $4/3 - 1/3m = 7/6$ is optimal. We can only show a weaker result:

Theorem 2 *If the adversary is allowed to order same-size loads on a facility then the price of anarchy for 2 or more facilities is at least* $7/6$.

Proof. We want to show that there is no coordination mechanism that has small price of anarchy for the set of loads: the set of players $(w_1, \ldots, w_5) = (3, 3, 2, 2, 2)$ and the three sets that result when we replace exactly one 2 with 0. The intuition is that at the optimal allocation the players with load 3 must go to the same facility in the first case and to different facilities in the second one, which cannot be achieved by a coordination mechanism due to its distributed nature.

We sketch only the idea: If the first facility schedules $(3, 0, 2, 2, 0)$ with load 3 finishing last and the second facility schedules $(3, 3, 2, 0, 0)$ with load 2 finishing last, then there is Nash equilibrium with for $(3, 3, 2, 2, 0)$ with price of anarchy $6/5$. So we can assume that this is not the case (and the same holds for the symmetric cases). The rest of the proof is based on case analysis and shows that $(3, 3, 2, 2, 2)$ has a Nash equilibrium where the 3's end up at different facilities. This gives the desired lower bound. It is possible that the assumption about the adversary is not necessary, but we were not able to get rid of it. $\square$

4 Selfish routing

We now consider coordination mechanisms for the selfish routing problem studied by Roughgarden and Tardos [16] (Example 2). What should an appropriate mechanism be for this case? The most natural class of mechanisms are those which are allowed to introduce delays on each link. These extra delays hopefully forces the players to coordinate better. The advantage of these mechanisms is that they are relatively easy to be deployed. Take for example the simple network of two parallel links of Figure 1, with delay functions $L_u(x) = x$ and $L_d(x) = 1$. When the traffic load between nodes 1 and 2 is 1, there is a unique Nash equilibrium in which the whole traffic uses the u edge. However, the optimal allocation is to partition equally the traffic on both edges. The price of anarchy is $4/3$ [16]. A coordination mechanism however that increases the delay function L_u to

$$L'_u(x) = \begin{cases} x & \text{if } x \leq 1/2 \\ 1 & \text{otherwise} \end{cases}$$

achieves an optimal allocation. In fact, it is easily seen that the allocation is optimal independently of the total traffic. Therefore, this coordination mechanism has price of anarchy 1.

A similar example is given by the network in Figure 2 (the Braess' Paradox network). Again the price of anarchy is $4/3$ because when the total traffic between 1 and 4 is 1 the whole traffic follows the path $(1, 2, 3, 4)$. But if we increase the delay on link $(2, 3)$ from 0 to 1 (or higher), the price of

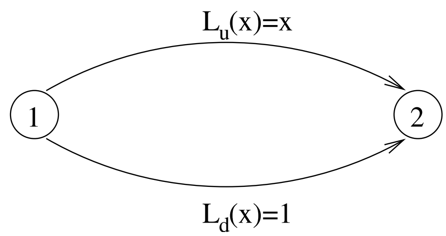

Figure 1: Two parallel links network

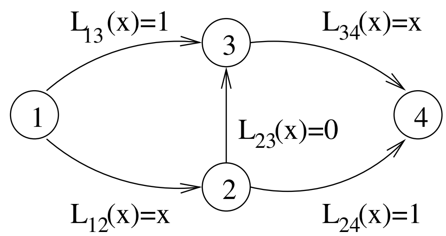

Figure 2: Braess' Paradox network

anarchy drops to 1. Is this a good coordination mechanism? Although it is optimal for total traffic 1, it is not optimal for other amounts of traffic. In particular, for total traffic $f < 1$, the optimal allocation routes traffic $\min(f, 1-f)$ through path $(1, 2, 3, 4)$ (the rest, if any, is split between the other two paths). It is clear that no coordination mechanism is optimal for all amounts of traffic. However, the coordination mechanism that increases the delay L_{23} to

$$L'_{23}(x) = \begin{cases} 0 & \text{if } x \leq (17 - \sqrt{97})/24 \\ 1 & \text{otherwise} \end{cases}$$

has price of anarchy of anarchy $96/(79 + \sqrt{97}) \approx 1.08$ (details omitted). There is an interesting relation to the work of Roughgarden [14]. The question there was: given a network and total traffic, find the set of edges that should be removed to have the minimum price of anarchy. Removing an edge is equivalent of setting its delay to infinity. Thus, Roughgarden's problem can be cast as finding the optimal among those coordination mechanisms that allow either a given specific delay function or infinity. As it is illustrated by the above example, finding coordination mechanisms when intermediate delay functions are allowed is a more complicated problem.

The above networks have bounded price of anarchy. Our last example shows that a coordination mechanism can improve dramatically performance of networks with arbitrarily high price of anarchy. Consider the network of Figure 1 of two parallel links but with delay functions $L_u(x) = x^p$ and $L_d(x) = 1$. It is known (see for example [16]) to have price of anarchy $1/(1 - p(p+1)^{-1/p}$ which tends to infinity as p tends to infinity. But it is easy to create a coordination mechanism that achieves price of anarchy 1: Increase the load from $L_1(x)$ to

$$L'_u(x) = \begin{cases} x^p & \text{if } x \leq (p+1)^{-1/p} \\ \infty & \text{otherwise} \end{cases}$$

The delay function $L'_u(x)$ has the effect that flow up to $(p+1)^{-1/p}$ goes to link 1 and the excess flow (if any) goes to the other link. Since this is the optimal flow, the price of anarchy is 1.

4.1 Arbitrary delays

A very important observation is that all the above mechanisms involve non-continuous delay functions. This is unavoidable since mechanisms with continuous delay functions cannot improve the price of anarchy at all. In general the following theorem holds:

Theorem 3 *If all delay functions of a network are continuous and non-decreasing then coordination mechanisms with continuous delay functions cannot improve the price of anarchy.*

Proof. We sketch the basic idea. If all delay functions of a network are continuous and non-decreasing, the expected delay experienced by the flow (denoted by $\ell(f)$) is clearly a non-decreasing function of the total flow f. Now consider a flow f which maximizes the price of anarchy. If we increase the delay on some edges, by continuity, there is a flow $f' < f$ such that $f'\ell'(f') = f\ell(f)$. The social cost of f' with new delay functions is equal to the social cost of f with the original delay functions. However, the optimal cost for f' cannot be greater than the optimal cost for f because the delay functions are non-decreasing. Therefore, the price of anarchy for f' is at least equal to the price of anarchy of f. $\Box$

An important consequence of the theorem is that the study of coordination mechanisms should depart from the standard assumption in [16] that the delay functions are continuous. It should be stressed however that non-continuous mechanisms do not seem harder to be implemented; they add however an extra layer of complexity. For example, it is not even clear that optimal coordination mechanisms are computable.

4.2 Linear delays for two links

We now turn our attention to networks of two parallel edges with linear delay functions $L_u(x) = a_1x + b_1$ and $L_d(x) = a_2x + b_2$. It is known that without a coordination mechanism their price of anarchy is at most 4/3 and the bound is tight for some networks (for example, the network of Figure 1). We will show that non-trivial yet simple coordination mechanisms can improve this ratio.

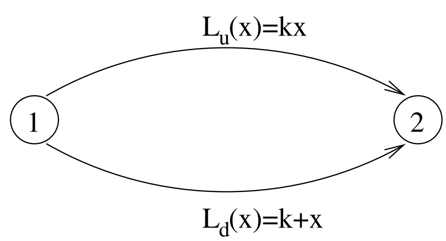

Figure 3: Two links with linear delay functions

Without loss of generality assume that $b_2 \geq b_1$. Then, we can estimate the optimum

$$\mathrm{opt}(f) = \begin{cases} a_1 f^2 + b_1 f & \text{if } f \leq \frac{b_2 - b_1}{2a_1} \\ \frac{a_1 a_2}{a_1 + a_2} f^2 + \frac{a_1 b_2 + a_2 b_1}{a_1 + a_2} f - \frac{(b_2 - b_1)^2}{4(a_1 + a_2)} & \text{otherwise} \end{cases}$$

and the social cost of a Nash equilibrium

$$sc(f) = \begin{cases} a_1 f^2 + b_1 f & \text{if } f \leq \frac{b_2 - b_1}{a_1} \\ \frac{a_1 a_2}{a_1 + a_2} f^2 + \frac{a_1 b_2 + a_2 b_1}{a_1 + a_2} f & \text{otherwise} \end{cases}$$

For $f = (b_2 - b_1)/a_1$ the price of anarchy sc/opt is given by $\mathrm{PA} = (1 - \frac{a_1}{4(a_1+a_2)} \frac{b_2-b_1}{b_2})^{-1}$. This is equal to $4/3$ when $a_2 = 0$ and $b_1 = 0$ (the network of Figure 1), and essentially (via scaling) it attains this maximum value only for these parameters. We can see that by scaling and translating (subtracting the same constant from all delay functions)it is possible to reduce the number of parameters. Henceforth and without loss of generality, we will assume that $b_1 = 0$ and $b_2 = a_1 = k$ and $a_2 = 1$ (Figure 3). The parameters were adjusted so that at total flow $f = 1$ and at the Nash equilibrium the flow starts using the down link d. It is easy to compute the price of anarchy PA (it achieves its value for flow $f = 1$). The optimal strategy places all flow up to $1/2$ in link u and the excess flow is split appropriately into the two links. In contrast, the Nash equilibrium strategy is identical except that it places all flow up to 1 in link u before the excess is split appropriately into the two links. A mechanism should try to force the Nash solution to put some flow in link d sooner. Equivalently, it should raise the delay function of link u for flow less than 1. By Theorem 3 the new delay functions should not be continuous (and in particular this should be true for $f = 1$).

We will therefore consider the mechanisms that raise the delay $L_u(x) = kx$ to

$$L'_u(x) = \begin{cases} kx & \text{if } x \leq f_0 \\ kx + v & \text{otherwise} \end{cases}$$

for some $f_0 \leq 1$ and $v \geq 0$. Thus flow up to f_0 is placed into link u, then flow up to $kf_0 + v - k$ goes into link d, and the excess flow is split into the two links (link d receives k more flow than link u). Let $f_1 = (k+1)f_0 + v - k$. The social cost then is

$$sc(f) = \begin{cases} kf^2 & \text{if } f \leq f_0 \\ kf_0^2 + (f - f_0)^2 + k(f - f_0) & \text{if } f_0 \leq f \leq f_1 \\ (\frac{k}{k+1}f + \frac{1}{k+1}f_1 - f_0 + k)f & \text{otherwise} \end{cases}$$

The price of anarchy is the maximum of $\mathrm{PA}_1 = sc(f_0)/\mathrm{opt}(f_0)$ and $\mathrm{PA}_2 = sc(f_1)/\mathrm{opt}(f_1)$.

For example for $k = 1$, the mechanism cannot improve the price of anarchy ($8/7$). But for higher values of k (as it turns out for $k > 1.6$) the coordination mechanism improves the price of anarchy. In particular for $k = 2$, without any mechanism the price of anarchy is $6/5$, but the optimal mechanism achieves a slightly better price of anarchy $\mathrm{PA} = 1.19..$ using $f_0 = 0.9826..$ and $v = 0.4422...$ Computational results suggest that this is approximately the worst case for any k. We obtain therefore the following theorem.

Theorem 4 *For a network of two parallel links with linear delay functions, there is a coordination mechanism with price of anarchy approximately* 1.19*.*

This special case network played a special role in the study of anarchy for the selfish routing problem. Essentially, it was shown in [15] that the problem of larger networks can be reduced to this case. If the same holds for coordination mechanisms, then the above theorem will be central in resolving the general case too.

5 Open problems

There are numerous interesting open problems suggested by the framework of coordination mechanisms. First, there are the obvious problems to improve the upper or lower bound for the resource allocation problem (Theorems 1 and 2). Then, in this work we simply initiated the study of coordination mechanisms for the selfish routing problem. What is the optimal price of anarchy for arbitrary networks and linear delay functions? What about general delay functions?

For the coordination model of congestion with finitely many players of Example 2, it is easy to observe that even for 2 players and 2 facilities, there are coordination models of congestion with arbitrarily high coordination ratio. But what is the price of anarchy for the asymmetric coordination mechanisms?

Also numerous computational complexity issues are open. How hard it is to compute optimal coordination mechanisms for the problems studied here? It is not even clear that the problem is decidable (although it seems very plausible).

Finally, in mechanism design there is the notion of truthfulness (strategyproof). Similar issues arise for coordination mechanisms. For example, the coordination mechanism for the resource allocation problem that achieves price of anarchy $4/3 - 1/3m$ has the property that it favors (schedules first) large loads. This is undesirable since it gives incentive to players to lie and pretend to have larger load. Consider now the mechanism that is exactly the same but schedules the loads in increasing order. Using the same ideas as in the proof of Theorem 1, we can show that this coordination mechanism has price of anarchy $2 - 1/m$. Although this is greater than $4/3 - 1/4m$, the mechanism is very robust in that the players have no incentive to lie (if we, of course, assume that they can't shrink their loads). Are there other robust coordination mechanisms with better price of anarchy?

References

[1] A. Czumaj and B. Vöcking. Tight Bounds for Worst-case Equilibria. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 413–420, January 2002.

[2] R. L. Graham. Bounds for certain multiprocessing anomalies, *Bell System Technical Journal*, 45, pages 1563-1581, 1966.

[3] Y. Korilis, A. Lazar and A. Orda, Architecting Noncooperative Networks. *IEEE Journal on Selected Areas in Communications,* Vol. 13, No. 7, pp. 1241–1251, September 1995.

[4] E. Koutsoupias, M. Mavronicolas and P. Spirakis. Approximate Equilibria and Ball Fusion. In *Proceedings of the 9th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2002

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

[6] I. Milchtaich. Congestion Games with Player-Specific Payoff Functions. *Games and Economic Behavior* 13, pages 111-124, 1996.

[7] D. Monderer and L. S. Shapley. Potential Games. *Games and and Economic Behavior* 14, pages 124-143, 1996.

[8] N. Nisan. Algorithms for selfish agents: Mechanism design for distributed computation. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 1–15, 1999.

[9] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[10] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* The MIT Press, 1994.

[11] C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, pages 749-753, 2001.

[12] A. Ronen. Algorithms for rational agents. In *Conference on Current Trends in Theory and Practice of Informatics*, pages 56–70, 2000.

[13] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65-67, 1973.

[14] T. Roughgarden. Designing networks for selfish users is hard. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pages 472–481, 2001.

[15] T. Roughgarden. The price of anarchy is independent of the network topology. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, pages 428-437, 2002.

[16] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236-259, 2002.

[17] T. Roughgarden and E. Tardos. Bounding the Inefficiency of Equilibria in Nonatomic Congestion Games. Cornell Technical Report TR2002-1866.

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