# Coordination mechanisms*

George Christodoulou[1], Elias Koutsoupias[1], and Akash Nanavati[2]

[1] Department of Informatics, University of Athens,
Panepistimiopolis Ilisia, Athens 15784, Greece.
{gchristo,elias}@di.uoa.gr

[2] Computer Science Department, University of California Los Angeles,
Los Angeles, CA 90095, USA.
akash@cs.ucla.edu

**Abstract.** We introduce the notion of coordination mechanisms to improve the performance in systems with independent selfish and non-colluding 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 task allocation and congestion games.

## 1 Introduction

The price of anarchy [11, 18] measures the deterioration in performance of systems on which resources are allocated by selfish agents. It captures 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). However unlike 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 propose a framework to study some of these problems and define the notion of coordination mechanisms (the parallel of online or approximation algorithms) which attempt to redesign the system to reduce price of anarchy. To introduce the issues, we consider first two different situations from which the notion of coordination mechanisms emerges in a natural way.

Consider first the selfish task allocation problem studied in [11]. There is a simple network of $m$ parallel links or $m$ identical machines and a set of $n$ selfish users. Each user $i$ has some load $w_i$ and wants to schedule it on one of the machines. When the users act selfishly at a Nash equilibrium the resulting allocation may be suboptimal. The price of anarchy, that is, the worst-case ratio of the maximum latency at a Nash equilibrium over the optimal allocation can be as high as $\Theta(\log m/\log\log m)$ [11, 5, 10]. The question is "*How can we improve*

* Research supported in part by the IST (FLAGS, IST-2001-33116) program and by NSF.

*the price of anarchy?*"; and what mechanisms one can use to improve the overall system performance even in the face of selfish behavior? We will assume that the system designer can select the scheduling policies of each machine; we then ask whether some scheduling policies can reduce the price of anarchy and by how much. An important aspect of the problem is that the designer must design the system once and for all, or equivalently that *the scheduling policies should be defined before the set of loads* is known. Another important and natural condition is the decentralized nature of the problem: *the scheduling on a machine should depend only on the loads assigned to it* and should be independent of the loads assigned to other machines (otherwise an optimal allocation can be easily enforced by a centralized authority and all game-theoretic issues vanish). This framework is very similar to competitive analysis, especially if we consider the worst-case price of anarchy: *We, the designers, select the scheduling policies for each machine. Then an adversary selects a set of loads. We then compute the makespan of the worst Nash equilibrium and divide by the makespan of the optimal allocation.* It is important to clarify that we divide with the absolute (original) optimum which is independent of our choice of scheduling policies.

As a second example, consider the selfish routing problem whose price of anarchy was studied by Roughgarden and Tardos [23]. In a network in which the latency experienced by the traffic on an edge depends on the traffic traversing the edge, selfish users route traffic on minimum-latency paths. The price of anarchy can be as high as $4/3$ for linear latency functions and unbounded for arbitrary latency functions. How can we improve the price of anarchy in this situation? For the famous Braess' paradox case, a simple solution is to remove some edges. The removal of edges however does not improve the price of anarchy in general; even for the Braess' paradox network, the removal of an edge can make the situation much worse for other amounts of traffic. We propose to study mechanisms that slow down the traffic on some edges to improve the performance. More precisely, we, the designers select for each edge $e$ a new latency function $\hat{c}^e$ which is equal or greater than the original latency function $c^e$; then the adversary selects a flow and we evaluate the price of anarchy. Notice that, as in the case of the selfish task allocation, we should divide the Nash equilibrium latency (computed using the new latency functions $\hat{c}^e$) by the optimal latency (of the original latency functions $c^e$).

### 1.1 Our Contributions

To study the above and similar problems, we introduce a unifying framework: the notion of *coordination models* which is an appropriate generalization of congestion games and the notion of *coordination mechanisms* which generalizes the scheduling policies and the increase in the cost and latency functions of the above examples.

Using this framework, we study the selfish task allocation problem (Section 3). We give a coordination mechanism (i.e., scheduling policies) with price of anarchy $4/3-1/(3m)$, improving significantly over the original $\Theta(\log m/\log\log m)$. We conjecture that this bound is tight, but we were able to show only that every coordination mechanism has price of anarchy strictly greater than 1 (this still allows the infimum price of anarchy to be 1).

We also study coordination mechanisms for congestion games (Section 4). We show an interesting relation between the potential and the social cost of a set of strategies; based on these we give a coordination mechanism with price of anarchy $n$ for the single-commodity congestion games. We also show that the bound $n$ is tight. We conjecture that the same bound holds for the general congestion games; but we were able to show only that the coordination mechanism that we employed for the single-commodity games fails in the general case (details in the full version).

Finally, for the case of selfish routing, non-continuous coordination mechanisms may perform arbitrarily better than continuous ones; this asks for removing the assumptions of continuity in the work of Roughgarden and Tardos [23]. We have positive results only for very special cases of small networks (details in the full version).

### 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 [17] 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* (see for example [2, 3]), 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 [16, 15, 19, 1].

The idea of designing games to improve coordination appears also in the work of Korilis, Lazar, and Orda [9] but there the goal is to design games with a unique Nash equilibrium; there is no attempt to compare it with the potential optimum.

In an attempt to reduce total delay at Nash equilibrium in the selfish routing problem, [2, 3] analyzes the problem of assigning *taxes* on network edges. Also, [14] analyzes how much total money one has to spend in order to influence the outcome of the game, when the interested party gives payments to agents on certain outcomes.

A problem that relates to coordination mechanisms for selfish routing, and studied in [21], 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 The Model

Congestion games [20, 13, 6], introduced by Rosenthal, is an important class of games that capture many aspects of selfish behavior in networks. A congestion game is defined by a tuple $(N, M, (\Sigma_i)_{i \in N}, (c^j)_{j \in M})$ where $N$ is the set of players, $M$ is the set of facilities, $\Sigma_i$ is a collection of strategies for player $i$, and $c^j$ is the cost (delay) function of facility $j$. The characterizing property of congestion games is that the cost of players for using facility $j$ is the same for all players and depends only on the number of players using the facility: when $k$ players use facility $j$, the cost of each player for using the facility is $c^j(k)$. The total cost of each player is the sum of the individual cost of each facility used by the player.

There are three important classes of congestion games: the single-commodity, the multi-commodity, and the general congestion games. In the most restricted class, the single-commodity congestion game, there are $n$ selfish players that want to establish a path from a fixed node $s$ to a fixed destination $t$. The facilities are the edges of the network and the strategies for each player are the paths from $s$ to $t$. In the more general class of multi-commodity games, each player may have its own source and destination. Finally, in the most general class there is no network. It is well-known that every congestion game has at least one pure Nash equilibrium.

To define the price of anarchy of a congestion game, we need first to agree on the social cost (i.e., the system cost) of a set of strategies. Two natural choices are the maximum or the average cost per player —the first one was used in the selfish task allocation problem of [11] and corresponds to the makespan, and the second one was used in the selfish routing problem in [23]. The price of anarchy is then defined as the worst-case ratio, among all Nash equilibria, over the optimal social cost, among all possible set of strategies.

One can generalize congestion games in two directions: First, to allow the players to have loads or weights and second, to allow asymmetric cost functions where players experience different cost for using a facility [12]. These generalizations are realized by cost functions $c_i^j$, one for each player —the cost of player $i$ for using facility $j$ is now $c_i^j(w^j)$ where $w^j$ is the sum of weights of the players using facility $j$.

How can we improve the price of anarchy of congestion games? There are two simple ways: First, by introducing delays, and second, by distinguishing between players and assigning priorities to them. Given a generalized congestion game $(N, M, (\Sigma_i)_{i \in N}, (c_i^j)_{j \in M, i \in N})$, we shall define the set of all possible games that result when we add delays and priorities; we will call these games coordination mechanisms. The introduction of delays is straightforward: the set of allowed games have cost functions $\hat{c}_i^j$ where $\hat{c}_i^j(w) \geq c_i^j(w)$. We will call these *symmetric coordination mechanisms*. The way to introduce priorities is less obvious but we can approach the problem as follows: Let facility $j$ assign priorities to players so that it services first player $t_1$, then player $t_2$ and so on. The cost (delay) of the first player $t_1$ cannot be less than $c_{t_1}^j(w_{t_1})$, the cost of using the facility itself. Similarly, the cost of the $k$-th player $t_k$ cannot be less than $c_{t_k}^j(w_{t_1} + \cdots + w_{t_k})$.

The natural problem is to select a coordination mechanism with small price of anarchy among all those coordination mechanisms with delays and priorities. To define this problem precisely and generalize the above discussion, we introduce the notion of coordination model in the next subsection.

### 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, $\Sigma_i$ 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. More precisely, $c^j$ is a cost function from $R^N$ to $R^N$. A natural property is that $c^j_i(w_1, \ldots, w_{i-1}, 0, w_{i+1}, \ldots, w_n) = 0$ which expresses exactly the property that players incur no cost when they don't use the facility.

In most coordination models, the strategies and cost functions are defined implicitly; for example, by introducing delays and priorities to a given congestion game. 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 1.* The coordination model for **selfish task allocation** that corresponds to the problem studied in [11] 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 $\Sigma_i$'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 $2w_1 + w_2$.

### 2.2 Coordination mechanisms

The notion of coordination model defined in the previous subsection 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 very rich. A coordination mechanism is essentially a *decentralized 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 task 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 $c = (c^1, \ldots, c^m)$, a set of player loads $w = (w_1, \ldots, w_n)$, and a set of strategies $A = (A_1, \ldots, A_n) \in \Sigma_1 \times \cdots \times \Sigma_n$. Let $(\mathrm{cost}_1, \ldots, \mathrm{cost}_n)$ denote the cost incurred by the players. We define the *social cost* $\mathrm{sc}(w; c; A)$ as the maximum (or sometimes the sum) cost among the players, i.e., $\mathrm{sc}(w; c; A) = \max_{i\in N} \mathrm{cost}_i$.

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 $\Sigma_1 \times \cdots \times \Sigma_n$, i.e., $opt(w) = \inf_{c,A} \mathrm{sc}(w; c; A)$.

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 task 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.

$$\mathrm{PA}(c) = \sup_{w} \sup_{E\in \mathrm{Ne}(w;c)} [\mathrm{sc}(w; c; E)/opt(w)]$$

We define the price of anarchy of a coordination model as the minimum price of anarchy over all its coordination mechanisms.

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 competitive 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*.

## 3 Selfish task allocation

We now turn our attention to the coordination model for selfish task allocation. 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 mechanism considered in [11] schedules the loads on each link in a random order resulting in the price of anarchy of 3/2. Consider now the following mechanism:

Increasing-Decreasing: "*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.*"

This mechanism aims to break the symmetry of loads. 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.

**Proposition 1.** *The above increasing-decreasing coordination mechanism has price of anarchy 1 for* $n \leq 3$ *and* $4/3$ *for* $n \geq 4$.

Is there a better coordination mechanism for 2 or more facilities? 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 $\epsilon$ 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 $i\epsilon$ create ties? We can avoid both 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).
- For each player, the cost on the facilities are different. To achieve this, the cost $c_i^j(w_1, \ldots, w_n)$ is a number whose representation in the $(m+1)$-ary system ends at $j$. To achieve this, the facility may have to introduce a small delay (at most a multiplicative factor of $\delta$, for some fixed small $\delta$). For example for $m = 9$ 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 $4/3 - 1/(3m)$.*

*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 the greedy scheduling with the loads ordered in decreasing size. It has been analyzed in Graham's seminal work [8] where it was established that its approximation ratio is $4/3 - 1/(3m)$. Given that the total delay introduced by the $\delta$ terms increases the social cost by at most a factor of $\delta$, we conclude that the price of anarchy is at most $4/3-1/(3m)+\delta$. The infimum as $\delta$ tends to 0 is $4/3-1/(3m)$.

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 $\delta$ term). □

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.

Finally we observe that the above mechanism reduces the question about the price of anarchy to the question of the approximation ratio of the greedy algorithm. This naturally extends to the case of machines with speeds. In this case, the price anarchy is $2 - 2/(m+1)$ and it follows from results in [7].

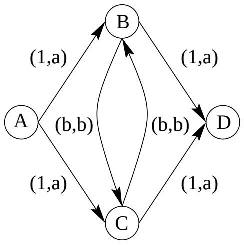
**Theorem 2.** *The above coordination mechanism for $n$ players and $m$ facilities with different speeds has price of anarchy $2 - 2/(m+1)$.*

The mechanism is appropriate for congestion games on any *network with linear cost functions* (the above discussion concerns the special case of $m$ parallel edges). In this case, if we apply the same mechanism to every edge of the network, the price of anarchy is the approximation ratio of the greedy algorithm for selecting $n$ paths. We point out that the price of anarchy is not known for these congestion games, yet we can still analyze the price of anarchy of the associated coordination mechanisms (in analogy, the analysis of Graham's algorithm is easier than determining the exact price of anarchy for $m$ machines). For lack of space, we leave the analysis of these extensions for the full version of the paper.

## 4 Congestion Games

In the previous section, we discussed coordination mechanisms for linear delay functions. In this section we will discuss coordination mechanisms for arbitrary delay functions. We will also consider pure equilibria —these games have at least one pure equilibrium.

Consider the single-commodity congestion game with $n = 2$ players defined by the network of the figure, where the labels on the edges represent facility/edge costs: $(c^e(1), \ldots, c^e(n))$. For $a \gg b \gg 1$, there is a Nash equilibrium where player 1 selects path $ABCD$ and player 2 selects path $ACBD$; its social cost is $2 + b$. *opt* is $(ABD, ACD)$ with cost 2. Hence the price of anarchy is $(2 + b)/2$ which can be arbitrarily high. Therefore




**Proposition 2.** *Without a coordination mechanism, the price of anarchy of congestion games (even of single-commodity ones) is unbounded.*

We consider symmetric coordination mechanisms that can increase the cost $c^j(k)$ of each facility. Can coordination mechanisms reduce the price of anarchy for congestion games? We believe that the answer is positive for general congestion games with *monotone* facility costs, i.e., when $c^j(k) \leq c^j(k+1)$ for all $j$ and $k$[3]. But we were able to establish it only for single-commodity games.

### 4.1 Single-commodity congestion games

Let $n$ denote the number of players. Our lower bound is (proof in the full version):

**Theorem 3.** *There are congestion games (even single-commodity ones) for which no coordination mechanism has price of anarchy less than $n$.*

We will now show that this lower bound is tight.

**Theorem 4.** *For every single-commodity congestion game there is a coordination mechanism with price of anarchy at most $n$.*

[3] For the unnatural case of non-monotone facility costs, it can easily shown that no coordination mechanism has bounded price of anarchy.

The proof uses the notion of potential [20, 13] of a set of strategies/paths. To define it, let $A = (A_1, \ldots, A_n)$ be strategies for the $n$ players and let $n^e = n^e(A)$ denote the number of occurrences of edge $e$ in the paths $A_1, \ldots, A_n$. The potential $P(A)$ is defined as $\sum_e \sum_{k=1}^{n^e} c^e(k)$ and plays a central role: The set of strategies $A$ is a Nash equilibrium if and only if $P(A)$ is a local minimum (i.e., when we change the strategy of only one player, the potential can only increase). It is also useful to bound the social cost as suggested by the following lemma (proof in the full version).

**Lemma 1.** *For every strategy $A$: $sc(A) \leq P(A) \leq n \cdot sc(A)$.*

The idea of a coordination mechanism for Theorem 4 is simple: Let $A^* = (A_1^*, \ldots, A_n^*)$ be a set of strategies that minimize the social cost (and achieve the social optimal). Let $n^e(A^*)$ be the number of occurrences of edge $e$ in the paths $A_1^*, \ldots, A_n^*$. The coordination mechanism keeps the same cost $c^e(k)$ for $k \leq n^e(A^*)$, but changes the cost $c^e(k) = a$ for $k > n^e(A^*)$ to some sufficiently large constant $a \gg 1$:

$$\hat{c}^e(k) = \begin{cases} c^e(k) & k \leq n^e(A^*) \\ a^2 & \text{for every } k \text{ when } n^e(A^*) = 0 \\ a & \text{otherwise} \end{cases}$$

The last two cases assign very high cost to edges that are used beyond the capacity determined by the optimal solution $A^*$. The middle case assigns even higher cost to edges not used at all by $A^*$ to guarantee that they are not used by any Nash equilibrium also.

The idea of the mechanism is that the high cost $a$ will discourage players to use each edge $e$ more than $n^e(A^*)$ times and therefore will end up at a set of strategies $A$ with the same occurrences of edges as in $A^*$. This in turn would imply that $A$ and $A^*$ have the same potential and the theorem will follow from Lemma 1. However natural this idea for coordination mechanism may be, it is not guaranteed to work —there may exist Nash equilibria that use some edges more than $A^*$ (with cost $a$) but each individual player cannot switch to a path consisting entirely of low cost edges. We have an example for general congestion games where this happens, but the following lemma shows that this cannot happen for single-commodity games (details in the full version):

**Lemma 2.** *Let $G$ be a directed acyclic (multi)graph (dag) whose edges can be partitioned into $n$ edge-disjoint paths from $s$ to $t$. Let $A_1, \ldots, A_n$ be any paths from $s$ to $t$. Then there is some $i$ and a path $A_i'$ from $s$ to $t$ which is edge-disjoint from the paths $A_1, \ldots, A_{i-1}, A_{i+1}, \ldots, A_n$.*

*Proof (of Theorem 4).* Consider an optimal set of strategies $A^* = (A_1^*, \ldots, A_n^*)$. The multigraph $G$ formed by these $n$ paths from $s$ to $t$ should be acyclic. Consider also a Nash equilibrium $A = (A_1, \ldots, A_n)$ for the above-defined coordination mechanism $\hat{c}$. The paths use only edges of $G$, otherwise some player would benefit by switching to a (any) s-t path of $G$. Using Lemma 2 we can also guarantee that the paths use edges of $G$ with multiplicity equal or smaller than the multiplicity of $G$. In conclusion, the potential $P(A)$ is no greater than the potential $P(A^*)$ and the theorem follows from Lemma 1. □

Another interesting fact that follows easily from similar considerations is that the above coordination mechanism $\hat{c}$ has price of anarchy at most $V-1$ for single-commodity networks of $V$ nodes.

It is open whether the above coordination mechanism works well for multi-commodity games. But, as mentioned above, it does not work for general games (details in the full version). We conjecture however that there are (other) coordination mechanisms with price of anarchy $n$ for every congestion game with positive monotone costs.

## 5 Open problems

There are many variants of congestion games for which we don't know their price of anarchy, let alone the price of anarchy of the corresponding coordination models and mechanisms. The problems are parameterized by whether we consider *pure or mixed* Nash equilibria, by whether the flow is *splittable or unsplittable*, and by whether the social cost is the *maximum or the average* cost of the players. Then there is the class of delay functions: linear ($c(x) = a \cdot x$), affine ($c(x) = a \cdot x + b$), or general. Finally, we can distinguish between the *weighted and unweighted* cases (where the loads are all equal or not) and between *symmetric or asymmetric* coordination mechanisms (in the latter case the mechanism can prioritize the players).

The immediate problems that are left open by our results include the gap between the upper and the lower bound for the task allocation problem. Also in Section 4.1, we considered only congestion games with no weights (and no adversary). What is the price of anarchy when the players have weights $w_i$ or simply when an adversary can select which players will participate (this corresponds to 0-1 weights)? A more distributed mechanism is required in this case.

Finally, in mechanism design there is the notion of truthfulness (strategyproof). Similar issues arise for coordination mechanisms. For example, the coordination mechanism for the task 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 $2-2/(m+1)$. Although this is greater than $4/3-1/(3m)$, the mechanism is very robust (truthful) 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? Also, for the case of different speeds, the mechanism that orders the job in increasing size has non-constant price of anarchy (at least logarithmic [4]). Are there truthful mechanisms with constant price of anarchy for this case?

## References


1. A. Archer and E. Tardos. Frugal Path Mechanisms. In *ACM-SIAM SODA*, 2002.
2. R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing. In *ACM EC*, pages 98–107, 2003.
3. R. Cole, Y. Dodis, and T. Roughgarden. Pricing network edges for heterogeneous selfish users. In *ACM STOC*, pages 521–530, 2003.
4. Y. Cho and S. Sahni. Bounds for list schedules on uniform processors. *SIAM J. Comput.* 9(1):91–103, February 1980.
5. A. Czumaj and B. Vöcking. Tight Bounds for Worst-case Equilibria. In *ACM-SIAM SODA*, pp. 413–420, 2002.
6. A. Fabrikant, C. Papadimitriou, and K. Tulwar. On the complexity of pure equilibria. `www.cs.berkeley.edu/~christos/papers/pure.ps`
7. T. Gonzalez, O. Ibarra, and S. Sahni. Bounds for LPT schedules on uniform processors. *SIAM J. Comput.* 6(1):155–166, March 1977.
8. R. L. Graham. Bounds for certain multiprocessing anomalies, *Bell System Technical Journal*, 45: 1563-1581, 1966.
9. 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.
10. 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
11. E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *STACS*, pages 404-413, 1999.
12. I. Milchtaich. Congestion Games with Player-Specific Payoff Functions. *Games and Economic Behavior* 13, pages 111-124, 1996.
13. D. Monderer and L. S. Shapley. Potential Games. *Games and and Economic Behavior* 14, pages 124-143, 1996.
14. D. Monderer and M. Tennenholtz. k-Implementation. In *ACM EC*, pp 19–28, 2003.
15. N. Nisan. Algorithms for selfish agents: Mechanism design for distributed computation. In *STACS*, pp 1–15, 1999.
16. N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.
17. M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* The MIT Press, 1994.
18. C. H. Papadimitriou. Algorithms, games, and the Internet. In *ACM STOC*, pp 749-753, 2001.
19. A. Ronen. Algorithms for rational agents. In *Conference on Current Trends in Theory and Practice of Informatics*, pages 56–70, 2000.
20. R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65-67, 1973.
21. T. Roughgarden. Designing networks for selfish users is hard. In *IEEE FOCS*, pp 472–481, 2001.
22. T. Roughgarden. The price of anarchy is independent of the network topology. In *ACM STOC*, pp 428-437, 2002.
23. T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236-259, 2002.
24. 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.