Coordination Mechanisms*

George Christodoulou† Elias Koutsoupias‡ Akash Nanavati§

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.

Keywords: Game theory, Price of Anarchy, Mechanism, Congestion Games, Selfish task allocation

1 Introduction

The price of anarchy [25, 33] 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 the price of anarchy. To introduce the issues, we consider first some situations from which the notion of coordination mechanisms emerges in a natural way.

Consider first the selfish task allocation problem studied in [25]. 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 $\Theta(\log m / \log \log m)$ [25, 14, 26]. The question is "*How can we improve 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

*Research supported in part by the IST (FLAGS IST-33116, AEOLUS IST-15964) program and by NSF. A preliminary version of this appeared in the 31st International Colloquium on Automata, Languages and Programming, 2004.

†Max-Planck-Institut für Informatik, Saarbrücken, Germany. Email: gchristo@mpi-inf.mpg.de

‡Department of Informatics, University of Athens. Email: elias@di.uoa.gr

§Computer Science Department, University of California Los Angeles. Email: akash@cs.ucla.edu

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. We also want to clarify that in this work we assume that the number of players is fixed. There are interesting questions when we assume that the number of players is not fixed in advance, but we do not treat them in this work.

Our second example is based on congestion games [35, 28]. To simplify the discussion let's consider the special class of single-commodity (that is, symmetric) congestion games in this section. In a 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 edges of the network have latency (cost) functions c^e so that when exactly k players use the edge e, each one of the players incurs cost $c^e(k)$. The cost of a player is the sum of the costs of the edges that he uses. We define the social cost to be the maximum cost among the players. The selfish behavior of the players may result in suboptimal Nash equilibria and the congestion game may have a high price of anarchy[1]. How can we reduce it? 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. Notice that, as in the case of the selfish task allocation, we should divide the Nash equilibrium social cost (computed using the new latency functions $\hat{c}^e$) by the optimal latency (of the original latency functions c^e).

There are two major difference between the models of the above examples: The first difference is that in the first model of selfish task allocation, there is an *adversary* who selects the weights; in the second example of congestion games, the players have no weights and there is no place for an adversary (as we already mentioned, we assume that the number of player is also known to the coordination mechanism). The second difference is that in the first model the coordination mechanism can assign priorities and therefore the cost function of the facilities (machines) is player-specific; in the second example, all players that use a facility pay the same cost (we call these coordination mechanisms symmetric).

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.

We also study symmetric coordination mechanisms for congestion games (Section 4). We first show that any symmetric coordination mechanism has price of anarchy at least n. Then, 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 series-parallel network congestion games.

[1]See [4, 10, 9, 1] for results on price of anarchy of congestion games for linear and polynomial latency functions. In this work, we consider general latencies.

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 [32] is the notion of mechanism design[2] in which the players are paid (or penalized) to "coordinate". Another attempt is the introduction of taxes (tolls) on the edges of the network in selfish routing games. In both of these approaches, the objectives of the players are modified by introducing side payments to their utilities.

In the former class of problems, part of the input is unknown to the system designer and is controlled by the selfish players. The objective of the mechanism designer is to convince the players to reveal their true values (truthfulness) by paying them, while simultaneously optimize (as much possible) the system objective. In the latter class of problems (see for example [13, 12, 16, 20, 21]), the objective of the designer is to determine optimal taxes on the edges that result in an optimal routing, assuming that the players' objective is affected by the taxes imposed. Our approach generalizes the notion of taxes: The tax on each edge is constant, independent of the traffic of the edge; in coordination mechanisms, the delay is a general function of the traffic on the edge. Thus, coordination mechanisms make sense even when the global traffic rates are not known to the mechanism. This raises the interesting question–which we do not pursue in this work—for designing mechanisms that work for any rates or any number of players; clearly constant taxes are very limited, if we do not know the total amount of traffic.

In our setting, we modify the objective of the players by adding delays and setting priorities on each edge. We do not make use of side payments. We also do not assume knowledge of the exact instance; we assume knowledge of the network topology but we do not know the weight of the players that participate. Also, the algorithmic and communication issues involved in mechanism design seem to be completely different than the ones involved in coordination mechanisms [31, 30, 34, 3].

The idea of designing games to improve coordination appears also in the work of Korilis, Lazar, and Orda [24] but there the goal is to design games with a unique Nash equilibrium; there is no attempt to compare it with the potential optimum.

Also, [29] 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 [36], 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).

Following the conference version of this work [11] a few publications studied coordination mechanisms. In particular, [19, 22] study coordination mechanisms for the selfish task allocation, while [2, 8] consider truthful mechanisms for the same problem. Recently, Azar et al. [5] and Caragiannis [6] consider coordination mechanisms for selfish scheduling on unrelated machines. Also, Kollias [22] gave lower bounds for coordination mechanisms for the selfish task allocation problem when no delays are allowed.

2 The Model

Congestion games [35, 28, 15], 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, Σ_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

[2] See [31] for an introduction to the algorithmic aspects of this notion.

j, the cost of each player for using the facility is $c^j(k)$. The 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 or symmetric, 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 [25] and corresponds to the makespan, and the second one was used in the selfish routing problem in [37]. 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 [27, 17]. 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 [4, 10, 9, 1]? 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, 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 n loads, one for each player, and outputs a cost to each player. More precisely, c^j is a cost function from R^N to R^N. When a player does not use the facility, his load is 0 and we must have $c_i^j(w_1, \ldots, w_{i-1}, 0, w_{i+1}, \ldots, w_n) = 0$, which expresses exactly the property that players incur no cost when they do not use the facility.

A symmetric coordination mechanism is one in which the cost functions are the same for all players who use it and depend only on the sum of their weights: $c_i^j(w) = c^j(\sum_{i=1}^m w_i)$.

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. The coordination model for **selfish task allocation** that corresponds to the problem studied in [25] 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: $\max_{i \in S} c_i^j(w_1, \ldots, w_n) \geq \sum_{i \in S} w_i$. 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 $2w_1 + w_2$.

Example. We can extend a congestion model to a symmetric coordination model in a natural way, by allowing each facility to introduce delays. Consider a given weighted congestion game G in which the cost $d^j(k)$ of facility j depends on the sum k of weights of the players using the facility. A symmetric coordination model contains all games with cost functions greater or equal to the cost functions of G for every edge j: We will call this model the **symmetric coordination model for congestion games**. Here we will study only the unweighted case where the players have always weight $w_i = 1$.

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 do not 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 facility; the scheduling on each facility depends only on the loads that use the facility. Observe that a facility knows the entire vector $(w_1, \ldots, w_n)$, where w_i is either the load of player i when he uses the facility, or 0 when he does not use it. Specifically, a facility knows the total number of players n and the loads and IDs of those players who use it. A facility does not know the loads and strategies of those players who do not use it.

For the coordination model for congestion games the situation is simpler: a facility knows all parameters of the original congestion game G and it has to select a cost function greater or equal than the original one (for every number of players).

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 $(\text{cost}_1, \ldots, \text{cost}_n)$ denote the cost incurred by the players. We define the *social cost* $\text{sc}(w; c; A)$ as the maximum cost among the players, i.e., $\text{sc}(w; c; A) = \max_{i \in N} \text{cost}_i$.

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.,

$$opt(w) = \inf_{c,A} \ \text{sc}(w; c; A) \tag{1}$$

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.

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

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 her finishing 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 [25] 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 at least $4/3$ for $n \geq 4$.*

Proof. First, consider 3 jobs with weights $w_1 \geq w_2 \geq w_3$. Clearly, in the optimum allocation the first job is executed alone in a separate machine, while the other two jobs are executed together. Notice that in the Nash equilibrium, the first job will be executed first in the "decreasing" machine, so this is a dominant strategy for this job. The third job will analogously choose the "increasing" machine, while the second job will choose the "increasing" machine, because it will cost her time $w_3 + w_2$ which is less comparing to the time $w_1 + w_2$ that she would need in the "decreasing" machine. Hence the only Nash equilibrium of this game coincides with the optimum.

For the lower bound for $n \geq 4$, consider $w_1 = 2+\epsilon, w_2 = 2, w_3 = 1-\epsilon, w_4 = 1$, for arbitrarily small ϵ. In the only Nash equilibrium of this instance, the "decreasing" machine will get the first job, while the rest go to the "increasing" machine, giving a makespan of $4 - \epsilon$. In the optimum allocation the first and the third jobs go together giving a makespan of 3.

□

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 j introduces a delay $j\epsilon$ 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 delays $j\epsilon$ 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 delays $j\epsilon$ create ties? We can avoid both problems with a coordination mechanism 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. That is, when we fix the weights and strategies of the remaining players, there is a unique optimal machine for the player.

To achieve the second property, the cost $c_i^j(w_1, \ldots, w_n)$ is a number whose representation in the $(m+1)$-ary system ends in j. The facility may have to introduce a small delay (at most a multiplicative factor of δ, for some fixed small δ). In particular, suppose that a job is to finish at time t in machine j. Then the machine will release it at some time t' in the interval $[t, (1+\delta)t]$, where the representation of t' in the $(m+1)$-ary system ends in j. Since there are infinitely many choices of t', the mechanism must select one, for example, the choice with the shortest representation in the $(m+1)$-ary system. As an example, consider $m = 9$ machines and $\delta = 0.01$. If a job is to finish at time $t = 1.362$ in machine $j = 8$, then it will be released at some time in the interval $[1.362, (1+\delta) \cdot 1.362] = [1.362, 1.37562]$ and the release time must end in 8 in the (conveniently selected) decimal system. There are many choices for t'; the one with the shortest representation is 1.368.

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 *(aka LPT scheduling)*. It has been analyzed in Graham's seminal work [18] 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 δ, we conclude that the price of anarchy is at most $4/3 - 1/(3m) + \delta$. The infimum as δ 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 δ term). □

Notice some additional nice properties of this coordination mechanism: there is a unique Nash equilibrium (thus it is easy for players to "agree") and it has low computational complexity. In contrast, computing Nash equilibria is potentially a hard problem [7].

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 (see [22] for a partial answer, which shows that delays are indispensable).

Finally we observe that the above mechanism reduces the question about the price of anarchy to the question of the approximation ratio of the LPT algorithm. This naturally extends to the case of machines with speeds. In this case, the best known lower and upper bounds for the LPT algorithm are 1.54 and $1 + \sqrt{3}/3$, due to Kovács [23]. Therefore we get the following:

Theorem 2. *The above coordination mechanism for n players and m facilities with different speeds has price of anarchy between* 1.54 *and* $1 + \sqrt{3}/3$.

4 Congestion Games

In the previous section, we discussed coordination mechanisms for linear delay functions. In this section we will discuss symmetric coordination mechanisms for arbitrary delay functions. A symmetric coordination mechanism for an (unweighted) congestion game G is simply a congestion game with cost functions greater or equal to the cost functions of G. We will also consider pure Nash equilibria—congestion games and therefore their coordination mechanisms have at least one pure Nash equilibrium.

Unlike the model of the previous section, now there are no weights to be selected by an adversary. The analysis here is not analogous with competitive analysis.

Consider the single-commodity congestion game with $n = 2$ players defined by the network of the Figure 1, 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 $2 + b$. The optimal solution 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.*

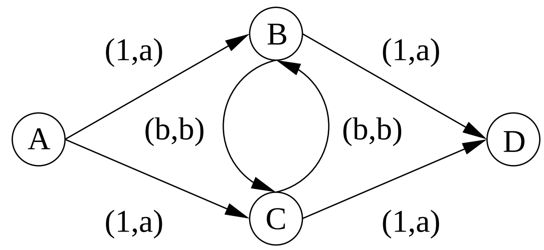

Figure 1: Unbounded price of anarchy

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

Definition 1. A congestion game is a *series-parallel congestion game*, if it is a single-commodity network congestion game with common source a vertex s and common destination a vertex t, and its underlying network is an s-t series parallel graph.

4.1 Series-parallel congestion games

We first establish the lower bound.

Theorem 3. *There are congestion games (even series-parallel ones) for which no symmetric coordination mechanism has price of anarchy less than the number of players, n.*

Proof. Consider a network with nodes $v_0, v_1, \ldots, v_n$ and n pairs of parallel edges (v_i, v_{i+1}), the upper edge with costs $(0, \ldots, 0, a)$ and the lower edge with costs $(a, \ldots, a)$ for some large constant a; Figure 2 depicts the network for 3 players. An optimal set of strategies is for player i to select the upper edges with the single exception that between v_{i-1} and v_i where it selects the lower edge; its social cost is a. On the other hand, a Nash equilibrium occurs when all players select the upper edges only. Its social cost is $n \cdot a$ and its price of anarchy n. No symmetric

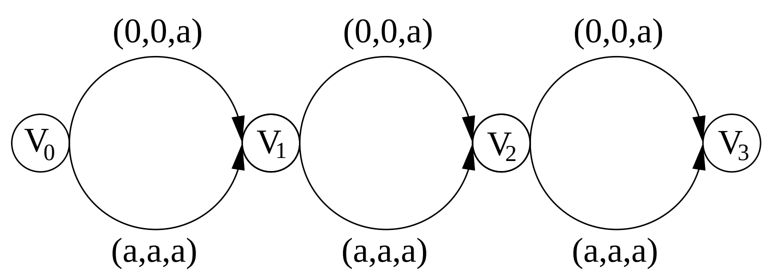

Figure 2: Price of anarchy n.

coordination mechanism can improve this: Clearly, at stage i from node v_{i-1} to node v_i, at least one player will incur cost at least a. Since the stages are independent, there is a Nash equilibrium in which the same player incurs cost at least a in every stage. This Nash equilibrium has social cost at least $n \cdot a$ and its price of anarchy is at least n. $\square$

We will now show that this lower bound is tight.

Theorem 4. *For every series-parallel 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 be easily shown that no coordination mechanism has bounded price of anarchy.

The proof uses the notion of potential [35, 28] of a set of strategies/paths due to Rosenthal. 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 in [35] 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

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

Proof. The lemma follows from the definitions of $P(A) = \sum_e \sum_{k=1}^{n^e} c^e(k)$ and $sc(A) = \max_i c_i$. The left inequality is derived as:

$$sc(A) = \max_i c_i = \max_i \sum_{e \in A_i} c^e(n^e) \le \sum_e c^e(n^e) \le \sum_e \sum_{k=1}^{n^e} c^e(k) = P(A),$$

and the right one as:

$$P(A) = \sum_e \sum_{k=1}^{n^e} c^e(k) \le \sum_e n^e c^e(n^e) = \sum_i c_i \le n \max_i c_i = nsc(A)$$

□

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 $k \le n^e(A^*)$, but changes the cost $c^e(k)$ for $k > n^e(A^*)$ to some sufficiently large constant $a \gg 1$:

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

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: the cost of one such edge exceeds the sum of cost of all—less than m—edges used by the optimal solution.

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 single-commodity network congestion games where this happens (Subsection 4.2), but the following graph-theoretic lemma shows that this cannot happen for games of series-parallel graphs[4]:

Lemma 2. *In a series-parallel multi-graph consider n $s-t$ edge-disjoint paths $A_1^*, \ldots, A_n^*$ and let $A_1, \ldots, A_k$ be any other $s-t$ paths with $k < n$. Then there is an $s-t$ path with edges that appear in $A_1^*, \ldots, A_n^*$ but not in $A_1, \ldots, A_k$.*

[4]In the conference version of the paper, we erroneously claimed that the mechanism has approximation n for single-commodity games.

Proof. We will prove this by induction on the structure of the graph. The basis of the induction (with a graph of only one edge) is trivial (we must have $n = 1$ and $k = 0$).

For the induction step, if the last composition of the graph is a series composition, we get the desired path by simply applying the series composition to the paths of the two parts.

If the last composition is a parallel composition, let n_1, k_1 and n_2, k_2 be the number of paths in the two parts (with $n = n_1 + n_2$ and $k = k_1 + k_2$). Since $n > k$, we must have $n_1 > k_1$ or $n_2 > k_2$. By induction, there is a path in the component i where the inequality $n_i > k_i$ is true. □

Proof of Theorem 4. Consider a series-parallel (directed) graph and an optimal set of strategies $A^* = (A_1^*, \ldots, A_n^*)$. Consider also a Nash equilibrium $A = (A_1, \ldots, A_n)$ for the above-defined coordination mechanism $\hat{c}$.

We will show that for every edge e, its multiplicity $n^e(A)$ in the paths of A does not exceed its multiplicity $n^e(A^*)$ in the paths of A^*, i.e. $n^e(A) \leq n^e(A^*)$.

First of all, the paths in A use only edges that appear in A^*, otherwise some player would benefit by switching to any path of A^* (this is guaranteed by the cost $a \cdot m$ assigned by the mechanism).

Fix now a player i and consider the paths A_{-i} of the remaining players. There is a path whose edges multiplicity $n^e(A_{-i})$ is at most $n^e(A^*) - 1$, as it follows directly from Lemma 2 for the paths A^* and A_{-i}. This path constitutes a strategy for player i which is less expensive than any path that uses some edge e with multiplicity more than $n^e(A^*)$. Since A is a Nash equilibrium, it follows that player i uses only edges with multiplicity at most $n^e(A^*)$. Of course, the same is true for all players.

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

Theorem 5. *The above coordination mechanism $\hat{c}$ has price of anarchy at most d for series-parallel congestion games with maximum s-t distance d.*

Proof. As in the proof of Theorem 4, let A, A^* be a Nash equilibrium and an optimal set of strategies; it was shown there that for every edge e, $n^e(A) \leq n^e(A^*)$. Let us assume that in A the player with maximum cost is i. We then have:

$$\begin{aligned}
sc(A) &= c_i(A) \\
&= \sum_{e \in A_i} \hat{c}^e(n^e(A)) \\
&\leq |A_i| \cdot \max_e \hat{c}^e(n^e(A)) \\
&\leq |A_i| \cdot \max_e \hat{c}^e(n^e(A^*)) && \text{because } n^e(A) \leq n^e(A^*) \text{ for every } e \\
&= |A_i| \cdot \max_e c^e(n^e(A^*)) && \text{by the definition of } \hat{c} \\
&\leq |A_i| \cdot sc(A^*) \\
&\leq d \cdot sc(A^*).
\end{aligned}$$

□

4.2 Counterexample of single-commodity network congestion game

The algorithm of Theorem 4, does not hold for single-commodity network congestion games. In this section we give a counterexample where the algorithm fails to provide a bounded Price of Anarchy.

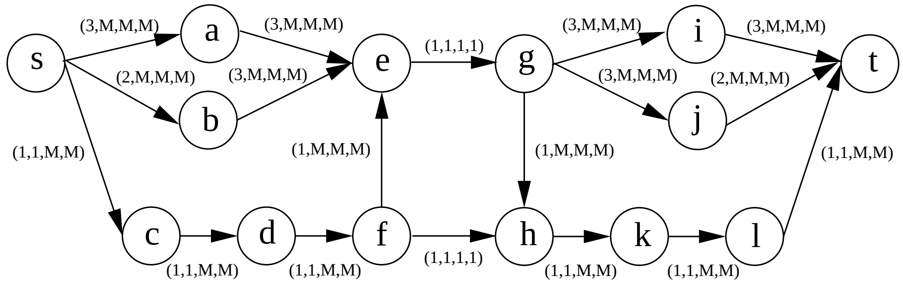

Figure 3: Unbounded price of anarchy.

Assume that we have the single-commodity instance of Figure 3 where 4 players want to route their traffic from s to t.

An optimum profile has cost 11 and selects the paths as follows:

$$
\begin{aligned}
A_1^* &= (s,a,e,g,h,k,l,t)\\
A_2^* &= (s,b,e,g,j,t)\\
A_3^* &= (s,c,d,f,e,g,i,t)\\
A_4^* &= (s,c,d,f,h,k,l,t).
\end{aligned}
$$

If we apply the symmetric coordination mechanism of Theorem 4 to this instance, the cost function of edge (e,g) will increase to $(1,1,1,M')$, while the cost function of (f,h) will increase to $(1,M',M',M')$, where $M' >> M$. Furthermore, it will change all occurences of M to M'.

Now consider the following paths:

$$
\begin{aligned}
A_1 &= (s,a,e,g,h,k,l,t)\\
A_2 &= (s,b,e,g,h,k,l,t)\\
A_3 &= (s,c,d,f,e,g,i,t)\\
A_4 &= (s,c,d,f,e,g,j,t)
\end{aligned}
$$

The above paths form a Nash equilibrium with social cost $9+2M'$, and therefore the price of anarchy of the aforementioned mechanism can be arbitrarily high.

5 Open problems

The immediate problem left open by our results is to close 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.

Acknowledgement: We would like to thank an anonymous reviewer for pointing out that some earlier version of this paper contained an incorrect generalization of Theorem 4. We would also like to thank him/her for providing the counterexample of Section 4.2.

References

[1] Sebastian Aland, Dominic Dumrauf, Martin Gairing, Burkhard Monien, and Florian Schoppmann. Exact price of anarchy for polynomial congestion games. In *23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 218–229, 2006.

[2] Eric Angel, Evripidis Bampis, and Fanny Pascual. Truthful algorithms for scheduling selfish tasks on parallel machines. In *Internet and Network Economics, First International Workshop (WINE)*, pages 698–707, 2005.

[3] Aaron Archer and Éva Tardos. Frugal path mechanisms. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 991–999, 2002.

[4] Baruch Awerbuch, Yossi Azar, and Amir Epstein. Large the price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 57–66, 2005.

[5] Y. Azar, K. Jain, and V. S. Mirrokni. Almost optimal coordination mechanisms for unrelated machine scheduling. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 323–332, 2008.

[6] Ioannis Caragiannis. Efficient coordination mechanisms for unrelated machine scheduling. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, to appear, 2009.

[7] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 261–272, 2006.

[8] George Christodoulou, Laurent Gourvès, and Fanny Pascual. Scheduling selfish tasks: about the performance of truthful algorithms. In *13th Annual International Computing and Combinatorics Conference (COCOON)*, pages 187–197, 2007.

[9] George Christodoulou and Elias Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Algorithms - ESA 2005, 13th Annual European Symposium*, pages 59–70, 2005.

[10] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–73, 2005.

[11] George Christodoulou, Elias Koutsoupias, and Akash Nanavati. Coordination mechanisms. In *Automata, Languages and Programming: 31st International Colloquium (ICALP)*, pages 345–357, 2004.

[12] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 521–530, 2003.

[13] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. How much can taxes help selfish routing? *J. Comput. Syst. Sci.*, 72(3):444–467, 2006.

[14] Artur Czumaj and Berthold Vöcking. Tight bounds for worst-case equilibria. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 413–420, 2002.

[15] A. Fabrikant, C. Papadimitriou, and K. Talwar. On the complexity of pure equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[16] Lisa Fleischer, Kamal Jain, and Mohammad Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *45th Symposium on Foundations of Computer Science (FOCS)*, pages 277–285, 2004.

[17] Martin Gairing, Burkhard Monien, and Karsten Tiemann. Routing (un-) splittable flow in games with player-specific linear latency functions. In *Automata, Languages and Programming, 33rd International Colloquium (ICALP (1))*, pages 501–512, 2006.

[18] R. L. Graham. Bounds for certain multiprocessors anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.

[19] Nicole Immorlica, Li Li, Vahab S. Mirrokni, and Andreas Schulz. Coordination mechanisms for selfish scheduling. In *Internet and Network Economics, First International Workshop (WINE)*, pages 55–69, 2005.

[20] George Karakostas and Stavros G. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. In *45th Symposium on Foundations of Computer Science (FOCS)*, pages 268–276, 2004.

[21] George Karakostas and Stavros G. Kolliopoulos. The efficiency of optimal taxes. In *Combinatorial and Algorithmic Aspects of Networking, First Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN)*, pages 3–12, 2004.

[22] Konstantinos Kollias. Non-preemptive coordination mechanisms for identical machine scheduling games. In *Structural Information and Communication Complexity, 15th International Colloquium (SIROCCO)*, pages 197–208, 2008.

[23] Annamária Kovács New Approximation Bounds for LPT Scheduling. *Algorithmica (to appear)*, 2008.

[24] Yannis A. Korilis, Aurel A. Lazar, and Ariel Orda. Architecting noncooperative networks. *IEEE Journal on Selected Areas in Communications*, 13(7):1241–1251, 1995.

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

[26] Elias Koutsoupias, Marios Mavronicolas, and Paul G. Spirakis. Approximate equilibria and ball fusion. *Theory Comput. Syst.*, 36(6):683–693, 2003.

[27] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, 1996.

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

[29] Dov Monderer and Moshe Tennenholtz. k-implementation. In *Proceedings 4th ACM Conference on Electronic Commerce (EC)*, pages 19–28, 2003.

[30] Noam Nisan. Algorithms for selfish agents. In *16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 1–15, 1999.

[31] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[32] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[33] Christos H. Papadimitriou. Algorithms, games, and the internet. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.

[34] Amir Ronen. Algorithms for rational agents. In *27th Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, pages 56–70, 2000.

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

[36] T. Roughgarden. Designing networks for selfish users is hard. In *42nd Symposium on Foundations of Computer Science (FOCS)*, pages 472–481, 2001.

[37] T. Roughgarden and E. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.