# Myopic distributed protocols for singleton and independent-resource congestion games*

D. Kalles[3], A.C. Kaporis[1], and P.G. Spirakis[1,2]

[1] Dept of Computer Eng and Informatics, University of Patras, Greece
[2] Research Academic Computer Technology Institute, N. Kazantzaki Str, University of Patras Campus, Greece
[3] Hellenic Open University & Open University of Cyprus (dkalles@acm.org, kaporis@ceid.upatras.gr, spirakis@cti.gr)

**Abstract.** Let $n$ atomic players be routing their unsplitable flow on $m$ resources. When each player has the option to drop her current resource and select a better one, and this option is exercised sequentially and unilaterally, then a Nash Equilibrium (NE) will be eventually reached. Acting sequentially, however, is unrealistic in large systems. However, allowing concurrency, with an arbitrary number of players updating their resources at each time point, leads to an oscillation away from NE, due to big groups of players moving simultaneously and due to non-smooth resource cost functions. In this work, we validate experimentally simple concurrent protocols that are distributed and myopic yet are scalable, realistic (since players migrating out of the same source resource may only go to the same target resource at one step) and require only information local at each resource and, still, quickly reach a NE for a range of arbitrary cost functions.

**Keywords**: Algorithms, concurrent atomic congestion games, Nash equilibria

## 1 Introductory motivation

Alice enters a large University library at the evening determined to copy some pages from a friend's notes. Miraculously, she finds a quite peaceful environment where no student opts to shift from her copier at hand. All students know that no copier will decrease their waiting time. This operating point is a *Nash equilibrium* (NE) over copiers and it is quite straightforward to think of other library's facilities also being operated at a NE.

Suppose, however, that Alice observes groups of students rushing to copiers when she enters the library in the early morning. She observes student $S$, currently pending on copier $C$, contemplating to move to copier $C'$ which seems more appealing.

At this critical decision-making point, there are two issues for $S$. The first issue is that, if a group of students shift to $C'$ alongside $S$, then $S$'s waiting time is likely to increase. The second issue is that, if printer $C'$'s speed decreases abruptly even due to the slightest increase in demand, then it is even more likely that $S$'s waiting time will increase, and it may do so beyond any anticipation on the part of $S$.

These obstacles naturally give rise to oscillations. No oscillations occur if all students shift to copiers sequentially, one at a time. But, Alice is old enough to know that only Wonderland's disciplined students are determined to shift sequentially. Of course, even in Wonderland, certain side effects do persist: acting sequentially may last long until a NE is reached.

Back in the real world, however, imposing global synchronization is unrealistic. On the other hand, it seems realistic that students pending on copier $C$ will briefly discuss their options before deciding how some of them might wisely move to $C'$. Their "on-the-fly" discussions are independent and not affected by decisions taken within any other group of students currently pending on any other copier. Moreover, it is also unrealistic that their local speculations will improve by any global (thus, expensive) information supplied (such information might consist of all copier's congestion and average waiting time).

The central question is thus framed as:

* The 2nd and 3rd author were partially supported by the IST Program of the European Union under contract number IST-015964 (AEOLUS).

*Question:* Is it possible to model such concurrent migrations as a simple distributed protocol within available resources, based on local speculations and greedy decisions taken on the fly? Is it possible to show that such a distributed protocol, despite its simplicity, is powerful enough so as to quickly reach a NE?

Apparently, as soon as all massive and concurrent migrations to copiers have taken place, it may turn out that many students feel tempted to subsequently shift to newly appealing copiers. This may lead to an endless copier-oriented migration process, oscillating eternally away from a NE, and presenting formidble obstacles in our attempt at analyzing concurrent selfish play. Note that our example is modeled as a *singleton* congestion game, where each player selects one resource over the available ones. It can also be generalized (and become more severe) if not all players' tasks are of the same value, i.e. if weights are introduced.

The general problem identified in the above example regards *all* situations where selfish actors must compete for a set of resources and can make decisions based only on information about where they are and where they might want to go (i.e. they do not have access to what other distant actors do). Not surprisingly, there are numerous other fields of computer science that deal with similar situations, most notably in load balancing, scheculing and (most generally) distributed computing.

In this paper we focus on the development of decision making protocols to be used by actors who want to decide to which resource they should migrate. We want these protocols to promote selfish behaviour and only use information that is available to actors in their current resource (and the one they want to move to). We experimentally show that the protocols we develop lead towards a Nash Equilibrium for a wide range of resource cost functions, allowing both weighted and unweighted versions. Moreover, they do so in a number of steps that scales similarly with a baseline protocol that makes quite strong assumptions on the type of actors and the resources.

The rest of this paper is structured as follows: we first review the work on sequential and concurrent congestion games, as well as work on realistic assumptions for experimental settings. We then move on to describe in detail the (widely used) congestion game model we employ and, based on that, we describe two selfish protocols for making migration decisions. In the next section we validate the protocols and in the final section we discuss the implications of our findings.

## 2 Related work

### 2.1 *Sequential* and *Concurrent* congestion games

Congestion games (CG) provide a natural model for non-cooperative resource allocation and have been the subject of intensive research in algorithmic game theory. A *congestion game* is a non-cooperative game where selfish players compete over a set of resources. The players' strategies are subsets of resources. The cost of each player for selecting a particular resource is given by a non-negative and non-decreasing latency function of the load (or congestion) of the resource. The individual cost of a player is equal to the total cost for the resources in her strategy. A natural solution concept is that of a pure Nash equilibrium (NE), a state where no player can decrease her individual cost by unilaterally changing her strategy.

In a classical paper, Rosenthal [31] showed that pure Nash equilibria on atomic congestion games correspond to local minima of a natural potential function. Given the non-cooperative nature of congestion games, a natural question is whether the players, while trying to improve their cost and converge to a pure NE, actually manage to do so in a reasonable number of steps. The potential function of Rosenthal [31] decreases every time a single player changes her strategy and improves her individual cost. Hence every sequence of improving moves will eventually converge to a pure Nash equilibrium. However, this may require an exponential number of steps, since computing a pure Nash equilibrium of a congestion game is *PLS-complete* [14].

On a *singleton* CG each player can select only one amongst $m$ resources. The importance of such games was demonstrated by [23], which initiated the study of important issues on the quality of NE, such as the *Coordination Ratio*, and attracted a lot of subsequent research.

There are strong reasons why sequential protocols are subject to critique. The Elementary Step System hypothesis is assumed in the analysis of [9, 12, 18, 19, 25, 26, 29], under which at most one user performs an improving move in each round, and convergence to a NE is guaranteed. However, sequential moves require $\Omega(n)$ rounds in the worst-case until $n$ users reach a NE. Moreover, central control is imposed on moves. This is not an appealing model of modern networking, where simple decentralized distributed protocols can better reflect the reality of the net's liberal nature. Furthermore, classical proofs of sequential convergence are based on assumptions of unbounded rationality and global knowledge. In real-world networks it is unrealistic to assume that any player may be capable of monitoring the entire network per round. Even if a user could grasp the whole picture, it would be computationally demanding to decide her best move.

Uniform sampling is the cheapest way of searching the available resources. However, it typically results in slow convergence time, since it does not amplify highly appealing resources. On the contrary, proportional sampling highly boosts the speed of the process, since it reroutes large groups of users towards the most appealing resources at hand.

However, unlike sequential play, concurrent play may eternally oscillate away from NE (a similar problem is also well-known in the Network and Telecommunications Community [22, 24, 30]). There are two key obstacles here: first, players have limited global info on making decisions and, second, the cost of resources may increase unboundedly on new demand.

The main negative effect of the first obstacle is that an exceedingly big group of players can cause bottleneck phenomena on their destination resources. This can be avoided by allowing each user to sample *uniformly* and independently, with appropriately small probability, for a new resource. If the resources have (nearly) identical cost functions, this migration probability usually depends only on the departure-destination pair of resources, eliminating any requirement for global information [4]. However, if cost functions are arbitrary, we need more information to better tune this migration probability. For example, the overall average cost [13, 15] may be employed or, we may even revert to resource sampling *in proportion* with a global parameter [13, 15].

Concerning the second obstacle, suppose that user $i$ finds appealing to migrate to resource $e$ and that $e$ is associated with a smooth cost function. Then, moving to $e$ will likely not substantially affect $i$'s *a priori* estimated migration profit, even if many other users opt for $e$. However, that anticipated profit may deteriorate abysmally if $e$'s cost function is not smooth, and it might even take just one extra mate migrating to $e$ to cause this problem. A common way out is to consider cost functions that satisfy an $\alpha$-bounded jump condition [8] (or relative slope [15]) where, intuitively, on adding $t$ new players the new cost on $e$ is at most $\alpha^t$ times $e$'s previous cost.

When speaking about atomic concurrent congestion games, a typical approach [13] considers $n$ players concurrently probing for a better link amongst $m$ parallel links per round (singleton CG). Therein, link $j$ has linear latency $s_j x_j$, where $x_j$ is the number of players and $s_j$ is the constant speed of the link $j$. This is the KP model [23]. However, this migration protocol, though concurrent, is not completely decentralized, since it uses global information in order to allow only appropriate groups of users to migrate. More precisely, only users with latency exceeding the overall average link latency $\overline{L}_t$ at round $t$ are allowed to sample (on parallel) for a new link $j$ with an appropriate probability.

We stress here that, for the case of multiple different links, this sampling for a link $j$ is *proportional* to $d_t(j) = n_t(j) - s_j \overline{L}_t$, where $n_t(j)$ is the number of users on link $j$. Once more, this type of proportional sampling exposes global information to amplify favorable links, in contrast to the myopic scenario of sampling a random user, which in turn amplifies links proportionally to their load. All in all, these criteria highly boost the convergence time, requiring expectedly $O(\log \log n + \log m)$ rounds. Note, though, that the work in [13] trailed [20], which first validated a series of similar concurrent protocols.

The analysis of a concurrent protocol on identical links and players was presented in [4]. Therein, the important aspect of the analysis is that no global information was given to the migrants. In that protocol, during round $t$, each user $b$ on resource $i_b$ with load $X_{i_b}(t)$ selects a random resource $j_b$ and if $X_{i_b}(t) > X_{j_b}(t)$ then $b$ migrates to $j_b$ with probability $1 - X_{j_b}(t)/X_{i_b}(t)$. Note, again, that users act in parallel.

Despite that users perform only uniform sampling, this protocol quickly reaches an $\varepsilon$-NE in $O(\log \log n)$, or an exact NE in $O(\log \log n + m^4)$ rounds, in expectation. The reason that proportional sampling turns out to not be so crucial here, is the fact that all links are identical, so there is no need to reroute many users to any particular speedy link. Thus, an important question is to what extent such myopic distributed protocols can cope with links that have large differences amongst their latency functions.

The work in [28, 16] removes the assumption of perfect information, in the sense that decisions are taken on the basis of a bulletin board which does not depict the most up-to-date state. If the information depicted on this board is too old and not regularly updated then oscillations occur. The analysis tunes the rate of updating the bulletin toward eventual system convergence (see also [7, 11, 1]). Though an important simplification of the classical assumptions that up to now were used for proving convergence, the assumption of a bulletin board implicitly makes use of global information for important characteristics of the system.

## 2.2 Insights from distributed computing and traffic distributions

In our work, there are quite a few points where our research draws from advances in other fields of computing, beyond that of algorithmic game theory.

A key such point is the distribution of weights of players in the weighted version of a CG. Therein, the problem of estimating the typical workload distribution over servers of the Web has attracted a lot of research. Knowledge of this distribution helps evaluate the performance of proxies, servers, virtual networks and other Web related applications.

The work in [3] has influenced a lot of subsequent research. It presents experimental evidence that up to a critical file size (the *cutoff* value) the distribution behaves as *Lognormal*, while for larger sizes as *Power Law*. Also, embarking from [3], and many other important papers in that direction, [27] presented a rigorous justification of this particular interplay amongst Power Law and Lognormal and their natural emergence as a file-size law.

The second such key point concerns the nature of protocols that decide who migrates between resources and how, as well as the extent to which such migrations effectively and efficiently achieve some notion of optimality. The field that has been most influential in that respect is that of load balancing. Drawing from [5] we note that key results from that field recommend that migration protocols are realistic when they assume that (now, we switch to the game nomenclature) a number of players moving from one resource at a given time point actually move to the *same* target, and are not distributed amongst more than one target [10]. This differentiation is described as the contrast between *diffusion* and *dimension exchange* methods, where the latter impose that a resource will only communicate (sample) with *one* potential target resource, to determine where to allow some of its migrants to move to (if at all). It is important to note that this assumption improves the robustness of the migration protocol since, when considering which players to move out of a resource, we do not need to collect expensive information (as is the case, for example, in [13]) from *all* available resources but we just focus on sampling one potential target. To appreciate the robustness potential consider what would happen in a network where we might need to sample many resources, yet find that many of the links seem to be broken, as is quite likely of course.

The justification for our protocols can be further seen in [17], where load balancing between processors is examined and the recommendations therein suggest that it is reasonable to expect more than one migrant per time slot from the same resource, though all migrants from that source resource move to the same target resource. Indeed, therein it is argued that the standard way of moving one migrant per time is an unwarranted pessimism and that it is more realistic to assume that more than one player may move at a time out of source resource and towards the (same) destination one. Therein, it is also argued why a resource cannot be expected to communicate in parallel with other resources, leading to the observation that sequential communication means that all migrants from a source will all go to the same target. Note that the above points have been also stressed in [10].

Morever, also according to [17], we note that our protocols indeed realistically assume that only local information is made available to the migrating candidates; note that, in stark contrast to this recommendation, [13] assume that players have access to accurate global statistics (like average load) to compute their next move.

A further justification for our protocols is the design pattern discussion in [2], where analogues are drawn to several biological processes that have influenced the design of distributed computing protocols and algorithms, and where a central recurring theme is the identification of processes that rely on strictly local information yet manage to achieve some notion of effective global behavior.

## 3 An efficient selfish distributed protocol

The discussion to this point squarely manifests the importance of distributed protocols that allow an arbitrary number of users to reroute per round, on the basis of selfish migration criteria.

It is important that migration rules are simple and myopic, while strong enough for the players to quickly reach (learn) a stable state. Herein, terms "simple" and "myopic" mean that any selfish decision is taken by inexpensive computations based on local information only. In other words, the decision does not rely on global or expensive information about the overall current state of resources.

The basic idea of our protocol is that, per round, each player independently and concurrently selfishly moves on the basis of her corresponding costs, as measured for the current and destination resources. All players update their selection of resources without any knowledge of *global information* nor any *tuning probabilities*: destination resources are uniform at random. Thus, our protocols grasp the real-life's nature of "on-the-fly" human decision making. In essence, a player decides to migrate if the anticipated cost, after her move, to a target resource, is favourably compared to her cost at the current resource.

There are finite sets of $n$ players $N = \{1, \ldots, n\}$ and $m$ resources $E = \{e_1, \ldots, e_m\}$, respectively. The strategy space of player $i$ is $S_i = \{X \subseteq E : |X| = 1\}$; player $i$ selects as her strategy $s_i(t)$ a *single* edge at round $t$. The game consists of a sequence of rounds $t = 0, \ldots, t^*$. Initially player $i$ selects a random recourse $s_i(0) \in S_i$. Next, per round $t = 1, \ldots, t^*$, each player $i$ updates *concurrently and independently* his current strategy $s_i(t)$ to $s_i(t+1)$ according to an appropriate protocol.

The number $f_e(t)$ of players on resource $e$ is $f_e(t) = |\{j : e \in s_j(t)\}|$. On an unweighted CG, resource $e$ has a cost $\ell_e(f_e(t))$, which is a function of the number of players on $e$, $f_e(t)$. On a weighted CG, each player $j$ has weight $w_j$ and the weight $w_e(t)$ of players on resource $e$ is $w_e(t) = \sum_{\{j:e\in s_j(t)\}} w_j$, which is the corresponding sum-weight of the players on $e$. On an weighted CG, that cost is $\ell_e(w_e(t))$, a function of the sum of weights of the players on $e$. The cost $c_i(t)$ of player $i$ is the cost of the resource where this player resides, $l_e$. A given state is a NE, if it is not beneficial for any player to change unilaterally her strategy at hand.

First, we present our protocol for the unweighted case.

---

**B2B:** During round $t$, do in parallel on each resource $e \in E$:

1. **$\forall$ player $i$ on $e$, sample a random resource $e'_i$.**
   /* Each player samples myopically a new destination resource. This requires no global information. */
2. **$\forall$ player $i$ on $e$, let $OUT_{e'_i} = \max\{0, \ldots, f_e(t)\}$ sufficient to hold $\ell_{e'_i}(f_{e'_i}(t) + OUT_{e'_i}) < \ell_e(f_e(t) - OUT_{e'_i} + 1)$.**
   /* Each player on a given resource estimates the maximum number of her binmates that can follow her to her new sample destination, in a way that the destination will remain appealing *after* migration. Again, no global information is required. */
3. **Select a random LEADER player $i$,** amongst those that have sampled appealing resources ($OUT_{e'_i} > 0$) and allow her with all her estimated binmates to migrate to $e'_i$. Note that this does not require expensive communication amongst the players on a given resource.

---

The cautious reader might observe that, based on the above decription, each player incurs the cost of selecting a target resource and estimating the maximum number of accompanying mates. We have presented it in that way to emphasize the distributed nature of the protocol. One could simply decide to embed all estimate calculations into the last step. However, this is just an implementation issue and does not affect the generality of the approach.

We now present the weighted case.

---

**W-B2B:** During round $t$, do in parallel on each resource $e \in E$:

1. $\forall$ player $i$ on $e$, sample a random resource $e'_i$.
2. $\forall$ player $i$ on $e$, select an arbitrary subset $S_i$ of $i$'s mates, with their *maximum* corresponding weight sum $WOUT_{e'_i}$, sufficient to hold: $\ell_{e'_i}(w_{e'_i}(t) + WOUT_{e'_i}) < \ell_e(w_e(t) - WOUT_{e'_i} + w_i)$.
3. Select a random LEADER player $i$, amongst those that have sampled appealing resources ($WOUT_{e'_i} > 0$) and allow her with all her estimated binmates to migrate to $e'_i$.

---

The above descriptions can be easily extended to consider more than one leader per resource and to more sophisticated techniques for selecting leaders. We will defer such discussion to Section 5.1; suffice to say that, hereafter, we shall be using the notations **B2B** and **B2B**(1) interchangeably.

## 4 Experimental Validation of singleton CGs

At the initialization of an unweighted CG instance $I$, each one of $n$ players randomly selects one of $m$ resources. For each resource $e \in I$, a random cost is assigned. Following that, we experiment with PURE and MIX classes of cost functions as detailed below.

The $m$ resources of each random CG instance $I$ are associated with one of the following **PURE** classes of cost functions: **LIN**: $\ell_e(x) = a_e x + b_e$, **LOG**: $\ell_e(x) = b_e \log_{a_e} x$, **EXP**: $\ell_e(x) = b_e a_e^x$, **MM1**: $\ell_e(x) = \frac{b_e}{a_e - x}$, where $x$ is load and $a_e, b_e$ coefficients characterizing resource $e$.

Within a given PURE cost class, coefficients are independently drawn per cost function, with each coefficient drawn uniformly from $[0 + \epsilon, A], \epsilon = 1.05$, with parameter $A = 10$, in order to minimize concentration of coefficients's around any point $y^* \in [0 + \epsilon, A]$, and to avoid similarity amongst the $m$ resource's functions. This is motivated by the *independent coefficients* model in [32, Section 2.2].

To ensure that our experiments are not optimistically biased, for each instance $I$ with costs drawn from **MM1** PURE class, we take care that the overall capacity $\sum_e a_e$ is slightly above $n$ players (or the overall player's weight, for the weighted case). This guarantees that costs increases are not $\alpha$-bounded (so, they will peak abruptly) on almost all resources within $I$ and for almost all rounds except the final one.

Additionally, to avoid similar random costs within a game instance, we introduce **MIX** classes (see details below) containing mixtures of the above PURE classes, thus inducing the highest load-dependent cost variation.

In Figure 1 (Top) we plot our protocols' speed scaling with $n$ players, when the cost functions are drawn from one of the above PURE classes of cost functions. The *density* $r$ is the ratio of $n$ players to $m$ resources. We fix resources at $m_0 = 2^6$ and set players $n = rm_0$, with $r$ increasing *exponentially* as $2^y$ with $y = 0, 1, \ldots, 10$. Let $T_{NE}(r)$ be the number of rounds until a NE for a given protocol (averaged over 10 random instances).

The lower LB-plot shows the running time of the protocol in [4, Fig. 2], on an input of $m_0$ identical resources and $n = rm_0$ identical players. This LB-plot will serve for comparison.

Each plot labeled "$\mathcal{C}$" shows the running time $T_{NE}(r)$ of our protocol **B2B**(1) on an input of $m_0$ resources, each drawn from a fixed PURE class $\mathcal{C} \in$ {LIN, LOG, EXP, MM1} and on $n = rm_0$ identical players. Since each $\mathcal{C}$ plot (with $m_0$ fixed) is almost parallel to the LB-plot, **B2B**(1) scales similarly to the protocol in [4, Fig. 2] with respect to $n$.

| $r$ | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 326 | 512 | 1024 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| EXP | 16 | 19 | 31 | 38 | 87 | 139 | 257 | 509 | 880 | 1701 | 3607 |
| MIX | 8 | 8 | 10 | 22 | 48 | 226 | 254 | 1198 | 1346 | 2427 | 4439 |
| MIX(EXP) | 16 | 21 | 24 | 42 | 67 | 126 | 248 | 467 | 880 | 1784 | 3570 |

**Table 1.** This table shows the logarithm of Rosenthal's initial potential value $\ln(\Phi(0))$ at round $t = 0$, before the protocol starts. Resources are fixed at $m_0 = 64$, rows correspond to cost classes and columns correspond to density $r$.

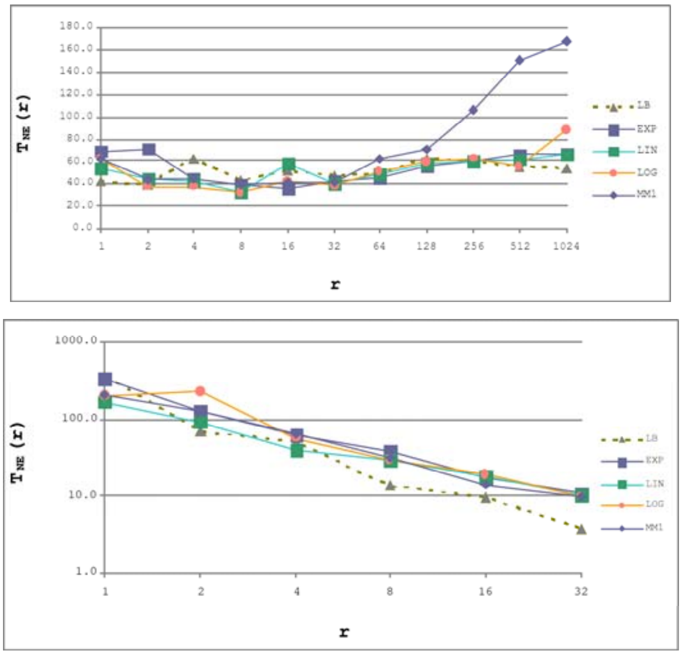


**Fig. 1. Unweighted CGs and** PURE **cost classes. Top: B2B(1)**'s speed scaling with $n$ players. Resources are fixed at $m_0 = 2^6$. Players are $n = n(r) = rm_0$ with $r$ increasing *exponentially* as $r = 2^t, t = 0, 1, \dots, 10$. **Bottom: B2B(1)**'s speed scaling with $m$ resources.

Note that, for class **EXP**, in Table 1 (Row 1) there appear the values of $\ln(\Phi_r(0))$ with respect to $r$, to compare them with the speed $\lceil n\alpha\varepsilon^{-1}\log(nC)\rceil$ of the sequential protocol in [8, Th. 1.3]. Observe here that for a sufficiently high $r$, $T_{NE}(r) < \ln(\Phi_r(0))$, showing **B2B**(1)'s speed compared to [8, Th. 1.3].

**MM1** instances seem to be the most difficult for protocol **B2B(1)**, when density really increases. For class **MM1**, such $\ln(\Phi(0))$ values do not appear in Table 1 since many resources have infinite cost in almost all rounds (but, this is exactly how those experiments were designed to avoid optimistic bias).

In Figure 1 (Bottom) we illustrate how **B2B**(1) scales with $m$ resources. Players are fixed at $n_0 = 2^8$ and $m = n_0/r$ decrease exponentially as $m = n_0/2^y$ when $y = 0, 1, \dots, 5$. We get the corresponding $\mathcal{C}$ plots, which are almost parallel to the LB-plot of the protocol in [4, Fig. 2].

A main concern of ours was to use a quite wide window of rate of cost growth with respect to the cost classes within instance $I$, ranging from **DLOG**[4], with a very smooth rate, to **EXP** and **MM1** classes, the most peaky ones. Towards this, we tried to illustrate more accurately the ability of our protocols to handle resources with fierce behavior on the slightest change of load, by considering classes **MIX**, **MIX(C)**.

A random CG instance $I$ belongs to class **MIX**, if each resource $e \in I$ has a cost function formed randomly according to a random PURE class. On such instances, see how **B2B**(1) scales with $n$ in Figure 2 (Top). A CG instance $I$ belongs to class **MIX(C)**[5] if exactly 1 resource belongs to **DLOG**, while the remaining $m-1$ resources belong to a fixed PURE class C. One can observer how **B2B**(1) scales with $n$ in Figure 2 (Top). In doing so, we essentially tried to mislead our protocols, by hiding 1 precious resource amongst $m-1$ costly ones. The corresponding scaling, as regards $m$, of **B2B**(1) appears in Figure 2 (Bottom).

[4] **DLOG**: $\ell_e(x) = b_e \log_{a_e} \ln x$, the functions therein are exponentially slower than in **LOG**.

[5] In class **MIX(MM1*)** 1 precious resource has MM1 cost function of high capacity and $m-1$ resources belong to **MM1**.

Note that **MIX(EXP)** instances seem to be the most difficult for protocol **B2B(1)**. For classes **MIX & MIX(EXP)**, in Table 1 (Rows 2 and 3) there appear $\ln(\Phi(0))$ values with respect to $r$ to compare them with $\lceil n\alpha\varepsilon^{-1}\log(nC)\rceil$ of the sequential protocol in [8, Th. 1.3], observing that **B2B**(1) does remarkably fast.

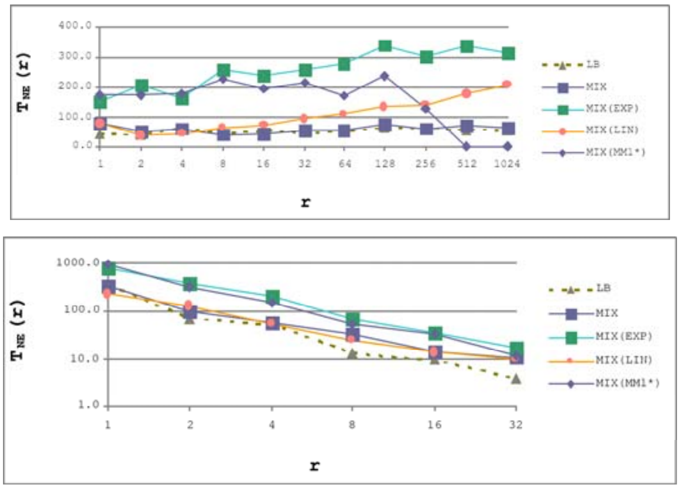


**Fig. 2. Unweighted CGs &** MIX **cost classes**. **Top: B2B(1)**'s speed scaling with $n$ players. **Bottom: B2B(1)**'s speed scaling with $m$ resources.

As far as weighted singleton CGs are concerned, we assign a random weight $X$ to each player, according to the distribution in Section 2.2. So, for $x < 133000$ (cutoff value) a random weight $X$ has Lognormal density $f(x) = (\sqrt{2\pi}\sigma x)^{-1}e^{-(\ln x-\mu)^2/(2\sigma^2)}$ with parameters $\mu = 9.357, \sigma = 1.318$. If $x \geq 133000$ a random weight $X$ obeys Pareto $g(x) = ak^a x^{-a-1}$ with parameters $k = 133000$ and $a = 1.1$.

Figure 3 (Top) shows, for the case of PURE classes, the corresponding speed scaling of protocol **W-B2B**(1) with respect to $n$, as compared to the LB plot of protocol [4, Fig. 2]. Figure 3 (Bottom) shows the corresponding plot with respect to $m$. Figure 4 shows the corresponding scalings for classes **MIX**, **MIX(C)**.

## 5 On the validity and the implications of the results

Besides being competitive at-large, our protocols avoid oscillations for *$\alpha$-bounded jump* cost functions, with experimentally tested $\alpha \leq 10$. Such cost functions include polynomials of bounded degree and exponentials that scale up to $10^x$ (see section 4 for details). Remarkably, our protocols remain fast for MM1 cost functions (widely met in real word applications) that do not satisfy any $\alpha$-bounded condition.

Our protocols's speed is compared to $O(\log\log n + m^4)$ achieved in [4] and to $O(\log\log n + \log m)$ in [13]. Both protocols [4, 13] scale (as $\log\log n$) with $n$ players (thus, they remain robust when adding players to the system), while [13] outperforms ($\log m$ vs $m^4$) [4] on $m$ resources (therefore [13] remains more robust than [4] when extending the system's resources). Using global information one may scale as $\log m$, by performing *proportional sampling* amongst $m$ resources, guiding the migration of players towards appealing resources [13]. When just *uniform sampling* amongst $m$ resources is employed, however, scaling deteriorates to $m^4$ [4].

Briefly reviewing the properties of the protocols we have developed, we note that:

– Our protocols are as simple and myopic as in [4], requiring no tuning of migration probability and exhibiting a similar speed scaling with $n$ players as in [4, 13]. They, also, scale with $m$ resources as in [4], but they are not as fast as $O(\log m)$ in [13] (therein, proportional sampling amplifies fast resources).

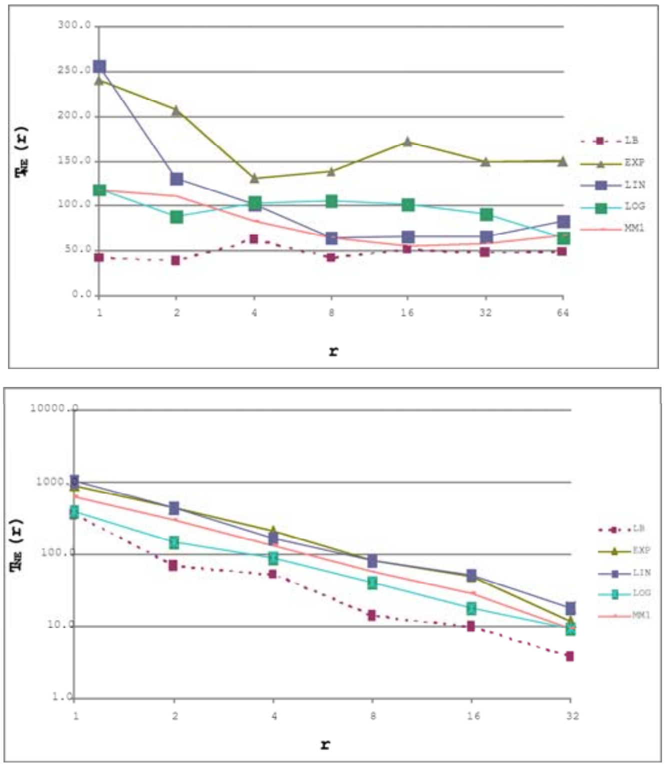


**Fig. 3. Weighted CGs &** Pure **cost classes**, as scaling for various densities, for fixed resources (Top) and fixed players (Bottom).

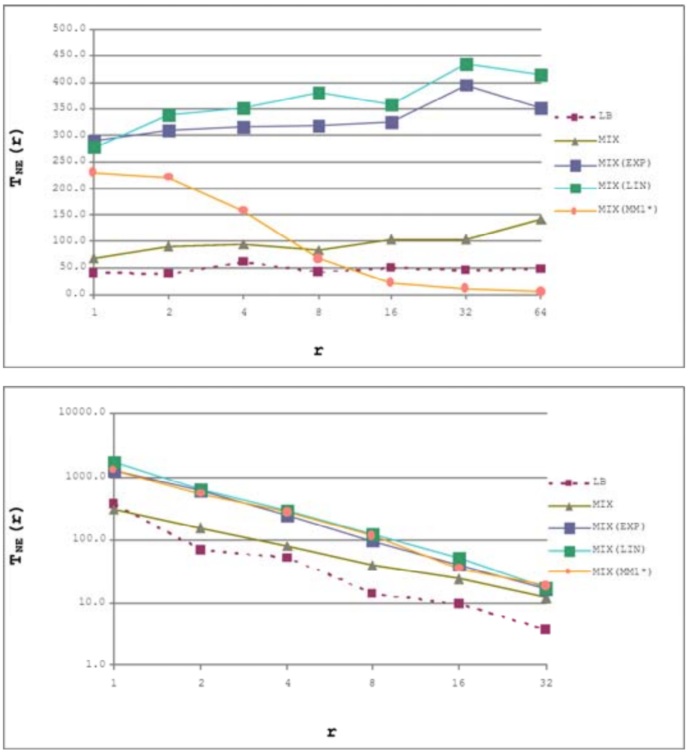


**Fig. 4. Weighted CGs &** MIX **cost classes**, as scaling for various densities, for fixed resources (Top) and fixed players (Bottom).

- Our protocols employ a realistic amount of parallelism. Specifically, they assume that during a migration step, players moving out of one resource may only go to the same target resource [17, 10].
- On any symmetric CG with $\alpha$-bounded latencies the *sequential* protocol in [8, Th. 1.3] reaches an[6] $\varepsilon$-NE in $\lceil n\alpha\varepsilon^{-1}\log(nC)\rceil$ rounds[7], which is $\geq poly(n)$ on the number of players. Our concurrent protocols reach a NE in $O(\log \Phi(0))$, with $\Phi(0)$ being Rosenthal's potential value at round $t = 0$, where it is well known that $\Phi(0) \leq nC$ (see the open problem in [8, Sec 7: Case 4]).
- Our protocols apply to a wide class of costs. The protocol in [4] balances load (number of players) over identical resources, with each cost being equal to load. The protocol in [13] is limited to linear cost functions with no constant term.
- Our protocols do apply to independent-resource CGs (but, see, Section 5.2 for details).
- Our protocols can also handle weighted players, unlike [4]. This property is also shared by [13], howeverm therein this is done with the use of global information and *just* for linear costs with no constant terms. Note that, with weighted players, an arbitrary weight assignment requires $\Omega(\sqrt{n})$ rounds till a NE [13]. We experimentally improve this lower bound by considering a realistic weight distribution described in Section 2.2.

### 5.1 Sensitivity experiments

We now summarize two further types of experiments, to estimate the sensitivity of our protocols to some build-in parameters of the experimentation (figures were omitted due to space limitations).

First, we experimented with the running time of **B2B** on instances within a particular cost class with coefficients $a_e, b_e \in [1.05, A]$ for $A = 10, 100, 1000, 10000$. These experiments were carried out with $m_0 = 64$ resources and density $r_0 = 32$. We observed that the corresponding average running time of **B2B** was almost the same or even better for values $A > 10$, which suggests that all our previous results are pessimistic and even better speed-ups should be expected.

For the second type of experiments, we have developed a variant, named **B2B**(2), where we select 2 *random* players $i_1, i_2$ amongst the players on resource $e$ such that $OUT_{e'_i} > 0$. We then select the most *influential* $i^*$ between those two such that $OUT_{e'_{i^*}} = \max\{OUT_{e'_{i_1}}, OUT_{e'_{i_2}}\}$ and then allow exactly $OUT_{e'_{i^*}}$ players to migrate to $e'$.

Furthemore, we have also developed **Hint**, another variant, where each leader player $j$ migrating out of $e$ additionally transmits her $OUT_{e'_j}$-value as a token to exactly 1 other random player amongst all available players. Essentially, this is as if she drops a hint and the first passer-by picks it up. Then, the LEADER player takes the best choice amongst her own-sampled one and the most recent OUT-token she may have received from another migrant.

So, motivated by the hardness of class **MIX(EXP)** for protocol **B2B**, as illustrated in Figure 2 (Top), we compared **B2B(1)**, **B2B(2)** and **Hint** on this class. The findings for **B2B(2)** and **Hint** were just slightly better throughout, suggesting that this problem class is indeed a tough nut to crack.

### 5.2 Independent Resource congestion games

Let us now generalize the introductory example. Suppose our student also has to consider her best choice amongst *many* groups of University resources (for example, the fasted public bus from town to campus, the most efficient pc in the lab, the least crowded studying room). Suppose that the University has $k$ such groups of resources. Then, this process amounts to a concurrent congestion game, where the strategy of player $i$ is a $k_i$-tuple of resources, each drawn from a group of similar resources.

Such CGs are *Independent Resource* ones.

Now, extending our intruductory discussion to nonatomic concurrent congestion games, we note that powerful concurrent protocols have been analyzed in a continuous setting with respect to the Wardrop model

[6] At an $\varepsilon$-NE bicriteria state, no player unilaterally changing her strategy can decrease the cost at hand by more than an $\varepsilon$-portion.
[7] $C$ upper bounds any player's cost. It is well known that if $\Phi(0)$ is Rosenthal's potential value at round $t = 0$ then $\Phi(0) \leq nC$.

(nonatomic flows) on general $k$ commodities nets. The fact that each agent controls an infinitesimal amount of flow facilitates the analysis, since any concurrent migration of a lower order population of players causes almost no oscillation effect. However, again, a great difficulty arises when a significant subset of the population concurrently migrates.

A series of important papers [6, 15] provide strong intuition on this subject, with [15] studying specific policies designated to yield fast convergence and [6] concluding that as long as all players concurrently employ arbitrary *no-regret* policies, they will eventually achieve convergence.

We now note (using the nomenclature introduced in Section 3) that the strategy space of player $i$ is $S_i = \{X \subseteq E : |X| = k_i, 1 \leq k_i \leq m\}$; player $i$ selects as her strategy $s_i(t)$, a $k_i$-tuple of edges at round $t$. Essentially, the set $E$ of all resources is partitioned into $k$ parts (or colors) $E_1, \ldots, E_k$, each part $E_j$ containing all resources of the same kind (color). The cost of resource $e$ is defined as for singleton CGs above.The cost of player $i$ is the sum of the corresponding costs of the resources in her strategy.

There is a convenient transparency amongst independent-resource and singleton congestion games [21]. More precisely, each player $i$ competing over $k_i$ kinds of resources can be interpreted as $k_i$ *clones*, each acting independently and selfishly in his corresponding group of resources. Now, let $n_j$ the number of clones in the subset $E_j$, containing all resources of a given kind. Then, it is convenient to view the *overall game* $G$ as $k$ independent congestion sub-games $G_1, \ldots, G_k$. In particular, sub-game $G_j$ is a singleton CG on $n_j$ players over $|E_j|$ resources, $j = 1, \ldots, k$. This method will raise up to at most $k$ times the corresponding singleton protocol's running time (actually, the experimental results for the independent-resource CGs are even better than $k$ times the corresponding ones for singleton CGs as shown in Section 4, but we omit them due to space limitations.)

## 6 Conclusions

We have presented a protocol for leading concurrent congestion games to Nash Equilibrium in a number of steps that is competitive to a baseline protocol [4]. We stress that the competitive quality of our protocol is underlined by the unrealistic assumption of the baseline protocol (as pointed out by the load balancing literature [17] [10]), that players currently at one resource may arbitrarily migrate to more than one target resource. In contrast to that, our protocol follows the ”realistic assumption” recommendation of [10].

## References


1. B. Awerbuch, Y. Azar, A. Fiat, and T. Leighton. Making commitments in the face of uncertainty: How to pick a winner almost every time. In *Proceedings of the 28th ACM Symposium on the Theory of Computing, 1996, pp. 519–530.*, 1996.
2. Ozalp Babaoglu, Geoffrey Canright, Andreas Deutsch, Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella, Niloy Ganguly, Mark Jelasity, Roberto Montemani, and Tore Urnes. Design Patterns from Biology for Distributed Computing. *ACM Transactions on Autonomous and Adaptive Systems*, 1(1):26–66, 2006.
3. Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *SIGMETRICS*, pages 151–160, 1998.
4. Petra Berenbrink, Tom Friedetzky, Leslie Ann Goldberg, Paul Goldberg, Zengjian Hu, and Russell Martin. Distributed selfish load balancing. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 354–363, New York, NY, USA, 2006. ACM Press.
5. Petra Berenbrink, Tom Friedetzky, and Zengjian Hu. A new analytical method for parallel, diffusion-type load balancing. In *Proc. of the 20th International Parallel and Distributed Processing Symposium (IPDPS).*, 2006.
6. Avrim Blum, Eyal Even-Dar, and Katrina Ligett. Routing without regret: on convergence to nash equilibria of regret-minimizing algorithms in routing games. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 45–52, New York, NY, USA, 2006. ACM Press.
7. J. Cao and C. Nyberg. An approximate analysis of load balancing using stale state information for servers in parallel. In *Proc. of the 2nd IASTED Int. Conf. on Communications, Internet, and Information Technology*, Nov 2003.
8. S. Chien and A. Sinclair. Convergece to Approximate Nash Equilibria in Congestion Games. In *Proc. of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA '07)*, page to appear, 2007.
9. George Christodoulou, Vahab S. Mirrokni, and Anastasios Sidiropoulos. Convergence and approximation in potential games. In *STACS*, pages 349–360, 2006.

10. George Cybenko. Dynamic Load Balancing for Distributed Memory Multiprocessors. *Journal of Parallel Distributed Computing*, 7(2):279–301, 1989.
11. M. Dahlin. Interpreting Stale Load Information. *IEEE Transactions on Parallel and Distributed Systems*, 11(10), Oct 2001.
12. Eyal Even-Dar, Alex Kesselman, and Yishay Mansour. Convergence Time to Nash Equilibria. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming (ICALP '03)*, pages pages 502–513, 2003.
13. Eyal Even-Dar and Yishay Mansour. Fast convergence of selfish rerouting. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, pages 772–781, 2005.
14. A. Fabrikant, C. Papadimitriou, and K. Talwar. The Complexity of Pure Nash Equilibria. In *Proc. of the 36th ACM Symp. on Theory of Computing (STOC '04)*, pages 604–612, 2004.
15. Simon Fischer, Harald Räcke, and Berthold Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *STOC*, pages 653–662, 2006.
16. Simon Fischer and Berthold Vöcking. Adaptive routing with stale information. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 276–283, New York, NY, USA, 2005. ACM Press.
17. Bhaskar Ghosh and S. Muthukrishnan. Dynamic load balancing in parallel and distributed networks by random matchings. In *Proc. of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA).*, pages 220–225, 1994.
18. Michel X. Goemans, Vahab S. Mirrokni, and Adrian Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–154, 2005.
19. P. W. Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game. In *In Proc. of the twenty-third annual ACM symposium on Principles of distributed computing (PODC '04)*, pages 131–140, New York, NY, USA, 2004. ACM Press.
20. T. Grenager, R. Powers, and Y. Shoham. Dispersion games: general definitions and some specific learning results. In *AAAI*, 2002.
21. Samuel Ieong, Robert McGrew, Eugene Nudelman, Yoav Shoham, and Qixiang Sun. Fast and compact: A simple class of congestion games. In *AAAI*, pages 489–494, 2005.
22. A. Khanna and J. Zinky. The revised arpanet routing metric. In *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, pages 45–56, New York, NY, USA, 1989. ACM Press.
23. E. Koutsoupias and C. Papadimitriou. Worst-case Equilibria. In *Proc. of the 16thAnnual Symposium on Theoretical Aspects of Computer Science (STACS '99)*, pages 404–413. Springer-Verlag, 1999.
24. J. F. Kurose and K.W. Ross. *Computer Networking, a top down approch featuring the Internet*. Addison-Wesley Longman (3rd ed.), 2004.
25. Lavy Libman and Ariel Orda. Atomic resource sharing in noncooperative networks. *Telecommunication Systems*, 17(4):385–409, 2001.
26. Vahab S. Mirrokni and Adrian Vetta. Convergence issues in competitive games. In *APPROX-RANDOM*, pages 183–194, 2004.
27. M. Mitzenmacher. Dynamic models for file sizes and double pareto distributions. *Internet Mathematics*, 1(3):305–334, 2004.
28. Michael Mitzenmacher. How useful is old information? *IEEE Trans. Parallel Distrib. Syst.*, 11(1):6–20, 2000.
29. Ariel Orda, Raphael Rom, and Nahum Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.
30. J. Rexford. *Handbook of optimization in Telecommunications, chapter Route optimization in IP networks*. Kluwer Academic Publishers, 2005.
31. R.W. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
32. Greg Valiant and Tim Roughgarden. Braess's paradox in large random graphs. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 296–305, New York, NY, USA, 2006. ACM Press.