Fast Load Balancing via Bounded Best Response

Baruch Awerbuch * Yossi Azar † Rohit Khandekar ‡

Abstract

It is known that the dynamics of best response in an environment of non-cooperative users may converge to a good solution when users play sequentially, but may cycle far away from the global optimum solution when users play concurrently. We introduce the notion of bounded best response where users react with best response subject to rules that are forced locally by the system. We investigate the problem of load balancing tasks on machines in a bipartite graph model and show that the dynamics of concurrent bounded best response converges to a near-optimum solution quickly, i.e., with poly-logarithmic number of rounds. This is in contrast to the concurrent best response dynamics which cycles far away from the optimum and to any sequential dynamics which requires at least a linear number of rounds to get to a reasonable solution.

1 Introduction

In most communication networks, it is infeasible to maintain one centralized authority to route traffic efficiently. As a result, users may decide individually how to route their traffic. Each user behaves strategically. Specifically, it wishes to minimize its transmission cost while being aware of the network congestion caused by other users. Selfish behavior is often analyzed by quality of the (Nash) equilibrium it induces.

However, even if Nash equilibrium is unique and constitutes the optimum solution, it is not clear whether selfishly acting users would converge to it or any other "good" solution. Actually, they may not converge at all. Thus, recently there has been a fair amount of work trying to understand the *convergence of the dynamics* of selfish users, rather than only analyzing the performance of a system in Nash equilibrium [8, 6, 2, 3, 9, 11]. One popular way to model the convergence issue is to assume that users are playing best response. In particular, there are several cases where it was proved that sequential best response converges to a good solution [8, 6, 2]. We note that such dynamics requires at least *linear* time (possibly polynomial or even more). However, in many cases and especially in large systems, sequential response is not an option. More appropriate model would be to assume *concurrent* response by all users. Unfortunately, the dynamics of concurrent best response results, in many cases, in cycles on states that are far away from good solutions.

To overcome this difficulty, we introduce the notion of *bounded* best response. In the bounded best response dynamics, the users play best response but are subjected to some rules that can be easily enforced locally by the system. An example of such a rule is the *multiplicative speed limit*: a user is allowed to increase (or decrease) its assignment to a certain option only by a certain percentage of the existing assignment. Such a speed limit is similar to the flow control rules on the Internet, where flow of a certain user is not allowed to drastically increase on a specific router. Obviously, such controls/restrictions are necessary since without them concurrent selfish decision making will lead to a collapse, as all flows may be continuously and simultaneously switching to new links, thus entering a vicious cycle. In contrast the bounded best response may actually converge to optimum.

As a case study, we focus on load balancing, or equivalently, assignment in bipartite graph model. We show that concurrent bounded best response reaches (and remains at) a near optimal solution very quickly, i.e., in poly-logarithmic number of rounds. The assignment problem in bipartite graph model is described as follows. The system consists of m machines (also called links) and n users where user

*baruch@cs.jhu.edu. Dept. of computer Science, Johns Hopkins University, Baltimore, MD, 21218.

†azar@tau.ac.il. Microsoft Research, Redmond and Tel-Aviv University, Tel-Aviv, 69978, Israel. Research supported in part by the Israel Science Foundation and by the German-Israeli Foundation.

‡rkhandekar@gmail.com. IBM T.J. Watson Research Center.

i has amount of tasks (also called traffic) w_i, and a subset S_i of allowable machines (induced by the graph). Each task has to be assigned to its allowable machines. We consider the fractional (or splittable) case where a task can be split fractionally among the allowable machines.

In our convergence dynamics, each user i is aware only of the loads on the machines in the set S_i. Each user behaves selfishly and wishes to minimize its cost by assigning its traffic to the least loaded machines. The global objective, however, is to minimize the load of the most loaded machines.

1.1 Our results We consider the concurrent case where all users react concurrently in rounds. We first observe the following

- If all users play best response (or even some approximation to that) concurrently, then the system can stay far away from the optimum solution.

Hence we have to impose some rules to ensure that the system would converge. We use two rules, called *Bounded step rule*, namely the fraction of the assignment of a task to a machine cannot change dramatically in a single round, and *Inertia rule*, namely that a user cannot move if it cannot improve its cost by a constant factor. In the *Bounded Best Response* dynamics, the users play the best response (or approximate best response) subject to the above rules. Note that all rules are stateless [1] and can be easily enforced by each machine *locally*.

- We show a fast convergence to a near optimal solution in the concurrent bounded response dynamics. It takes only poly-logarithmic number of rounds.

It is worth pointing out that while our mechanism yields fast convergence to a near-optimal solution, there is no evidence that it converges to (even approximate) Nash equilibrium, i.e., it appears that *optimum is easier to achieve than Nash equilibrium*. It appears that (approximate) Nash equilibrium may be hard to reach within a reasonable time, and thus it may not be the right concept in analyzing dynamics of a truly local system of selfish users. Note that in our case, the analysis of the equilibrium is trivial: the price of anarchy [10] is 1. However, since it does not appear that equilibrium is reached in poly-logarithmic time, this fact is not useful.

What makes our result non-trivial is proving that we reach near-optimality within poly-logarithmic time. The Bounded step rule makes it difficult to prove that some appropriately defined potential function (e.g., sum of the squares of the loads of machines) reduces fast enough while the system is "out of equilibrium". The reason is that a user cannot just move all its load from a highly loaded machine to a lightly loaded machine in a single round; rather, the user must build its traffic slowly to observe the rule. It may so happen that while this traffic is being built on a certain machine, this machine becomes overloaded as a result of actions of other users. Thus, one must start "chasing" another machine that is un-congested. In principle, the number of such changes can be linear or more, which is excessive for poly-logarithmic convergence time. Hence we are required to use a more refined analysis. Our analysis does not use a potential function. We instead show that

- either at a given time, the number of high loaded machines as a function of the load is a fast decreasing function, which in turn implies near-optimality (Lemma 4.3),

- or over the next poly-logarithmic rounds, the volume of tasks above some load threshold decreases significantly (Lemma 4.4) (it may not decrease significantly in a single round).

The latter is proved by showing that even when the loads of the machines fluctuate over time, the volume of consistently high tasks gets reduced.

1.2 Related work

Concurrent games. Past work on local greedy routing and analysis of routing dynamics, pretty much like in the current paper, does exist, but only provides partial results that work for special cases. Some of the closely related work includes recent ground-breaking results by Even-Dar et al. [4] and Fisher et al. [7, 5] who state results comparable to ours in the case that commodities operate in a *complete* network (clique). It appears that [5] also handles a general network topology with a common source and sink, which is essentially a single-commodity flow problem (this corresponds to symmetric users). We note that our load balancing prob-

lem corresponds to non-symmetric users as each user has a different set of strategies.

Similar to our paper, Awerbuch and Khandekar [1] considered minimization of max-load in general graphs. However, their dynamics is not best response in that agents are induced to work with a *different metric*, that is externally imposed upon them, which requires an additional enforcement mechanism. In addition, [1] makes an assumption that the global load in the network is known. In contrast, in the current paper, only local information is used, and no external cost metric is being introduced.

Sequential games. *Sequential* dynamics of best response where no concurrency effects exist, has been analyzed in prior work, by Fisher and Vöcking [6] by Chien and Sinclair [2]. The ε-moves similar to our notion of inertia rule have been used in [2].

It is worth noting that our proof techniques are completely different from those in [1, 2, 7, 5].

Paper structure: The paper is organized as follows. Section 2 defines the model and the problem and explains inadequacy of unrestricted best response. The bounded best response dynamics is defined in Section 3. The convergence of the best response dynamics is proved in Section 4.

2 The model and the problem

2.1 The model The bipartite model (also called restricted assignment model) is defined as follows: We are given a bipartite graph on n users (tasks) and m machines. A user i ($i = 1, \ldots, n$) has a task of weight w_i, that can be assigned to any machine j if there is an edge (i, j) in the graph. This is equivalent to having a subset S_i of the machines for any task i where the task can be assigned to. We consider the splittable case, i.e., each task can be split among some or all machines it may be assigned to. The load of a machine is the sum of the weights of the parts assigned to it. Given an instance of the problem, we define the *global optimum* (denoted by OPT) to be the fractional assignment of tasks to machines that minimizes the maximum load over all machines. Clearly OPT can be computed by a simple centralized flow algorithm. We abuse the notation and use OPT also to denote the maximum load in OPT.

We assume that each user is interested in minimizing its own cost with no regard to the global optimum. The user i is aware only of the loads of the machines in S_i. Given an assignment (also called system) A, let $p_{ij}^A \geq \eta$ be the fraction of task i that is assigned to machine $j \in S_i$ where η is a small constant to be defined later. For technical reasons, we always maintain a small fraction of every task on each machine it is connected to. We set η sufficiently small so that the effects due to this are negligible.

Clearly for any task i, the fractions add up to one: $\sum_{j=1}^{m} p_{ij}^A = 1$ for all i. For each machine j, let L_j^A be the total load on the machine: $L_j^A = \sum_{i=1}^{n} w_i \cdot p_{ij}^A$. We denote the maximum load by $L_{max}^A = \max_{1 \leq j \leq m} L_j^A$.

Our model works for various user-cost functions. Instead of defining a specific cost function, we define a characteristic of the cost functions: moving a small enough piece of a task from a high loaded machine to a low loaded machine reduces (or does not increase) the cost of the task.

One example of such a cost function is as follows: the cost of user i is the maximum over L_j^A for all $j \in S_i$ such that $p_{ij}^A > \eta$. Another natural cost function is as follows. Let f be a monotone increasing non-negative function. The cost of user i is $\sum_{j \in S_i} p_{ij}^A \cdot f(L_j^A)$.

2.2 Inadequacy of unrestricted best response Consider a process that runs in rounds. In round t, we are given an assignment A of tasks to machines where each user i can observe the load on all the machines in S_i. To simplify the notation, throughout the paper we omit the superscript A and may add instead the superscript t for the values in round t. Consider a task i. If there is some machine $j \in S_i$ such that $L_j > \min_{k \in S_i} L_k$ and still $p_{ij}^t > 0$ then user i can improve its cost by moving some fraction of its task from high loaded to low loaded machines in S_i (e.g., from j to the minimum loaded machine in S_i). We first show that the concurrent best response for all users may result in a very poor performance compared to the optimum.

THEOREM 2.1. *If all users perform concurrent best response then the system may cycle where the maximum load in some states is $\Theta(m)$ worse than the optimum.*

Proof. Assume that we have $m - 1$ tasks on m machines. For each $1 \leq i \leq m - 1$, let $S_i = \{i, m\}$. Assume that we start with an assignment such that $p_{ii} = 1$ and $p_{im} = 0$ for all i. Here the maximum load

is 1. This assignment is actually pretty close to the optimal assignment (the optimum load is $1 - 1/m$). However, since each user i sees an empty machine m, its best response would be to move half of its task to machine m in order to balance the load (formally, $p_{ii} = p_{im} = 1/2$). Since all users act concurrently, this would result in an assignment with a load of $(m-1)/2$ on machine m. At this time the best response for user i would be to move back to machine i. This results in a cycle (of two states). Clearly, half of the time, the system is in a state with maximum load $\Theta(m)$ times the optimum load. ■

The above example continues to hold even if we enforce that a task sends at least η fraction on each machine.

3 Rules of the game: bounded best response

Theorem 2.1 shows that concurrent (unrestricted) best response results in a bad performance. If we want to converge to the optimal assignment, we need to add or modify some rules. In this section, we will present a mechanism with such rules that induces convergence to an approximate optimal assignment in poly-logarithmic number of rounds. This mechanism will consist of two rules, *Inertia rule* (ε-moves) and *Bounded step rule* that restrict behavior of the users, and that can be locally enforced.

Inertia rule: This rule allows users to move from a high load machine to a low load machine only if the loads on the two machines differ significantly, e.g., by a multiplicative $(1+\varepsilon)^3$ factor, where ε is a small constant.

- A fraction of task i may move from machine j to r (where $j, r \in S_i$) only if $L_j \geq (1+\varepsilon)^3 L_r$.

The inertia rule has been introduced by Chien and Sinclair [2] under the name "ε-moves". We note that the bad example in Theorem 2.1 holds even if we enforce the Inertia rule; thus more rules are needed for achieving convergence.

The key to convergence is adding a new Bounded step rule that restricts the speed of movement of the users.

Bounded step rule: This is essentially the "multiplicative" speed limit. According to this rule, a task i can change the fraction of its assignment on a machine $j \in S_i$ at most by an ε fraction in a single round. More formally,

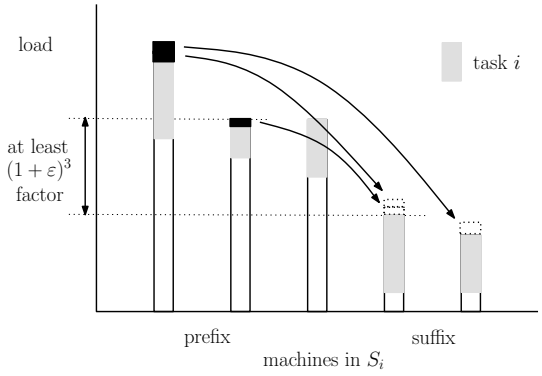

Figure 1: `Bounded best response of task` i.

- $\eta \leq p_{ij} \leq 1$ for all i and $j \in S_i$ where $\eta = \frac{\varepsilon}{m^2}$.
- $p_{ij}^t/(1+\varepsilon) \leq p_{ij}^{t+1} \leq (1+\varepsilon) p_{ij}^t$.

Since the multiplicative speed limit becomes ineffective if a fraction p_{ij}^t is zero (or very close to zero), we ensure that each task i sends at least $\eta = \frac{\varepsilon}{m^2}$ fraction to each machine $j \in S_i$. This affects the load on a machine by at most $\eta \sum_{i=1}^n w_i = \frac{\varepsilon}{m^2} \sum_{i=1}^n w_i$ which is negligible as compared to $\text{OPT} \geq \frac{1}{m} \sum_{i=1}^n w_i$.

It is easy to see that the above rules are *enforceable* by the system (the machines) and are not based on the willingness of the users to follow the rules honestly. Our key assumption is that the users are *maximally greedy* subject to the two rules above; such dynamics is called *bounded best response*.

- Each user i sorts the machines $j \in S_i$ in the decreasing order of the total load L_j and moves maximal possible fractions from high loaded machines in a prefix to low loaded machines in a suffix without violating the Inertia rule and the Bounded step rule.

The solution of the *optimization problem* to be solved by the user in order to minimize its cost is immaterial to us, the user is assumed to be "sophisticated" enough to solve this (relatively simple) problem. An illustration of bounded best response of task i is given in Figure 1.

4 Fast convergence to near optimum

We imagine that the tasks are divided into sufficiently small *pieces*. We further assume that these pieces

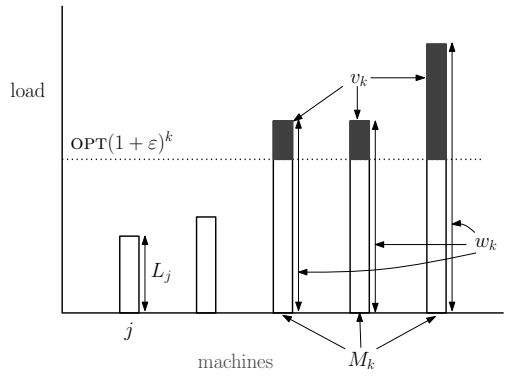

Figure 2: `Illustration of` L_j, M_k, w_k, `and` v_k.

are "named", i.e., each piece **p** of a task can be distinguished from the other pieces. We use the following definitions for any $k \geq 0$. Refer to Figure 2.

- M_k denotes the set of machines whose load is more than $\text{OPT}(1+\varepsilon)^k$.
- $m_k = |M_k|$. Clearly m_k is monotone non-increasing sequence.
- W_k denotes the set of pieces of tasks assigned to machines in M_k.
- w_k denotes the total volume (or weight) of pieces in W_k.
- v_k denotes the total weight of (pieces of) tasks above height $\text{OPT}(1+\varepsilon)^k$, i.e., $v_k = w_k - m_k\text{OPT}(1+\varepsilon)^k$.

We also use M_k^t, m_k^t, W_k^t, w_k^t, and v_k^t to denote the values of the above quantities in the beginning of round t.

We start with the following observation which is immediate from the Bounded step rule by summing over all tasks.

LEMMA 4.1. *For all machines j and time t we have $L_j^t/(1+\varepsilon) \leq L_j^{t+1} \leq (1+\varepsilon)L_j^t$.*

We imagine that the pieces of the tasks on a machine are arranged from bottom to top in some order (the order may change from round to round). Each piece of volume x would have a height from some h to $h+x$. When it moves to another machine, it would have a new height from h' to $h'+x$. We use the following order of the pieces on the machines. The top pieces are the pieces that may move in principle. Out of each piece p_{ij} we would put $p_{ij}\varepsilon/(1+\varepsilon)$ fraction at the top. Hence the total volume of the "top" pieces is $L_j\varepsilon/(1+\varepsilon)$ and hence their minimum height is $L_j/(1+\varepsilon)$ on the machine j.

All pieces that move to some machine r actually move to the top of that machine. Hence if a piece moves to a machine r, the new height of the piece is at most $L_r(1+\varepsilon)$. This follows since the total volume of the new pieces is at most εL_r.

LEMMA 4.2. *The dynamics is acyclic.*

Proof. Since the pieces corresponding to some task i move from a machine j to a machine r only if $L_j/L_r \geq (1+\varepsilon)^3$, we conclude that the heights of those pieces decrease from at least $L_j/(1+\varepsilon)$ to at most $L_r(1+\varepsilon)$, i.e., by at least $(1+\varepsilon)$ factor. This means that the sum of the heights over all pieces (or equivalently, the sum of squares of the loads of the machines) decreases over time. Therefore the dynamics is acyclic. ■

REMARK 1. *Since the pieces are moving from high load machines to low load machines, we also note that each v_k is a monotone non-increasing function of time. Moreover, each move results in strictly reducing at least one v_k for some k. In addition L_{max} is non-increasing over the process. Hence, once L_{max} is close to the optimum it would remain there.*

We now show that the maximum load L_{max} (which decreases over time) reaches close to the optimum in poly-logarithmic time.

LEMMA 4.3. *If for all $k \geq 0$, we have $m_k \geq m_{k+5}(1+\varepsilon)^k$ then $L_{\max} \leq (1+\delta)\text{OPT}$ for $\varepsilon = O(\frac{\delta^2}{\log m})$.*

Proof. We use the inequality iteratively (for k divisible by 5) and get

$$\begin{aligned} m \geq m_0 &\geq (1+\varepsilon)^{(0+5+10+\ldots+k)} m_{k+5} \\ &= (1+\varepsilon)^{k(k+5)/10} m_{k+5}. \end{aligned}$$

Now if $m_{k+5} \geq 1$, it implies that $k^2 = O(\frac{\log m}{\varepsilon})$. By the relation between ε and δ, we get that $(1+\varepsilon)^{k+5} \leq (1+\delta)$. ■

We divide the rounds into phases. Each phase consists of $\tau = O(\frac{1}{\varepsilon}\log\frac{m}{\varepsilon})$ consecutive rounds. If at any point, $v_k \le \varepsilon(1+\varepsilon)^k \text{OPT}$ holds for some k, then for all $k' \ge k+1$ we have $v_{k'} = 0$. This follows since for $v_{k+1} > 0$ to hold, it must be true that more than $((1+\varepsilon)^{k+1} - (1+\varepsilon)^k)\text{OPT} = \varepsilon(1+\varepsilon)^k\text{OPT}$ weight is above height $(1+\varepsilon)^k\text{OPT}$. Since $v_{k'}$ values are non-increasing, they always remain 0 for $k' \ge k+1$.

If in the beginning of a phase for all $k \ge 0$ we have

$$m_k \ge m_{k+5}(1+\varepsilon)^k,$$

then Lemma 4.3 implies that the system is near optimum. Therefore, we assume that there exists $k \ge 0$ such that $m_{k+5} > m_k/(1+\varepsilon)^k$. We show that in the next phase, either v_{k+1} or v_{k+4} decreases by factor of at least $\varepsilon/3$.

LEMMA 4.4. *If $m_{k+5} > m_k/(1+\varepsilon)^k$ holds in the beginning of a phase, then either v_{k+1} or v_{k+4} decreases by a factor of at least $\varepsilon/3$ in this phase.*

Before proving Lemma 4.4, (which is our main lemma) we prove its consequence.

THEOREM 4.1. *The bounded best response dynamics converges to $(1+\delta)$ approximation in $O(\frac{1}{\delta^6}\log^6\frac{m}{\delta})$ rounds.*

Proof. Note that the largest r for which $v_r > 0$ may hold in any round of the algorithm satisfies $r = O(\frac{1}{\varepsilon}\log m)$. This follows from the fact that $v_r > 0$ implies that $m_r > 0$ and the load on a machine in the set M_r is $(1+\varepsilon)^r\text{OPT} \le m \cdot \text{OPT}$. Lemma 4.4 together with Lemma 4.3 implies that, as long as the current assignment is not near optimum, at least one v_r decreases by a factor of $\varepsilon/3$, i.e., becomes $(1-\varepsilon/3)$ factor smaller. Since v_r is at most $m\cdot\text{OPT}$ in the beginning and at least $\varepsilon(1+\varepsilon)^r\text{OPT}$ before v_{r+1} becomes 0, the total number of phases v_r can decrease before v_{r+1} becomes 0 is $O(\frac{1}{\varepsilon}\log\frac{m}{\varepsilon(1+\varepsilon)^r})$. Summing this over $r = 0,\ldots,O(\frac{1}{\varepsilon}\log m)$, we get that the total number of phases before reaching near optimality is at most $O(\frac{1}{\varepsilon^2}\log^2\frac{m}{\varepsilon})$. Since each phase has $\tau = O(\frac{1}{\varepsilon}\log\frac{m}{\varepsilon})$ rounds, the total number of rounds in the algorithm is $O(\frac{1}{\varepsilon^3}\log^3\frac{m}{\varepsilon})$. The algorithm in the end achieves $(1+\delta)$ approximation to the optimum load where $\varepsilon = O(\frac{\delta^2}{\log m})$. Thus the total number of rounds in terms of the approximation factor δ is $O(\frac{1}{\delta^6}\log^6\frac{m}{\delta})$. ∎

4.1 Proof of Lemma 4.4 Fix a phase and let M_r^0, m_r^0, and v_r^0 denote the values of M_r, m_r, and v_r respectively in the beginning of this phase. Now assume on the contrary to Lemma 4.4 that neither v_{k+1} decreases by $\varepsilon v_{k+1}^0/3$ nor v_{k+4} decreases by $\varepsilon v_{k+4}^0/3$ in this phase. Let $k' = k+4$.

Rich machines. We call a machine "rich" if it is not in M_k in some round in the phase and it enters M_{k+1} in some later round during the phase (later it may again leave M_{k+1}). Note that a machine may become rich multiple times. For a machine to become rich once, to fill the gap between the levels $(1+\varepsilon)^{k+1}\text{OPT}$ and $(1+\varepsilon)^k\text{OPT}$, it must gain at least

$$(1+\varepsilon)^{k+1}\text{OPT} - (1+\varepsilon)^k\text{OPT} = \varepsilon(1+\varepsilon)^k\text{OPT}$$

volume. Since this volume must descend from the volume in v_{k+1}, the quantity v_{k+1} decreases by at least this amount. Hence, the number of rich machines (counting with the multiplicities) is at most

$$\frac{\varepsilon v_{k+1}^0}{3}\cdot\frac{1}{\varepsilon(1+\varepsilon)^k\text{OPT}} = \frac{v_{k+1}^0}{3(1+\varepsilon)^k\text{OPT}}.$$

Let M_{k+1}^* be the union of the sets M_{k+1}^t for all rounds t in the phase. This corresponds to the set of machines that had, in some round during the phase, a load of at least $(1+\varepsilon)^{k+1}\text{OPT}$. Let $m_{k+1}^* = |M_{k+1}^*|$. Hence

$$m_{k+1}^* \le m_k^0 + \frac{v_{k+1}^0}{3(1+\varepsilon)^k\text{OPT}}, \tag{4.1}$$

since for a machine to be in M_{k+1}^*, it should either already be in M_k^0 or be a rich machine.

Poor machines. We call a machine "poor" if it is in $M_{k'+1}$ in some round during the phase and then it leaves $M_{k'}$ in some later round in the phase. A machine can be counted poor more than once. For a machine to be poor once, $v_{k'}$ must be decreased by at least

$$(1+\varepsilon)^{k'+1}\text{OPT} - (1+\varepsilon)^{k'}\text{OPT} = \varepsilon(1+\varepsilon)^{k'}\text{OPT}.$$

This is since the machine must lose this volume to decrease its load from at least $(1+\varepsilon)^{k'+1}\text{OPT}$ to at most $(1+\varepsilon)^{k'}\text{OPT}$. Moreover this volume must leave $v_{k'}$ as well. Hence, the number of poor machines in the phase is at most

$$\frac{\varepsilon v_{k'}^0}{3}\cdot\frac{1}{\varepsilon(1+\varepsilon)^{k'}\text{OPT}} = \frac{v_{k'}^0}{3(1+\varepsilon)^{k'}\text{OPT}}.$$

Consistently high volume. Denote by $Z_{k'}$ the intersection of the sets $W^t_{k'}$ over all rounds t in the phase. This is the set of pieces that were always assigned to machines in $M^t_{k'}$, i.e., the machines with current load at least $(1+\varepsilon)^{k'}\text{OPT}$, at all rounds t in the phase. Let $z_{k'}$ be the total volume of pieces in $Z_{k'}$. The following lemma shows a lower bound on this volume.

LEMMA 4.5. *We have* $z_{k'} \geq \frac{2}{3}v^0_{k'} + m^0_k(1+\varepsilon)^4\text{OPT}$.

Proof. Let $z^t_{k'}$ denote the the volume of the pieces in the intersection of $W^{t'}_{k'}$ where t' ranges on rounds from the beginning of the phase till round t. During a phase $z^t_{k'}$ can only decrease. Initially it is at least

$$z_{k'} \geq v^0_{k'} + m^0_{k'}(1+\varepsilon)^{k'}\text{OPT}.$$

Now consider a decrease in $z^t_{k'}$ due to some piece $\mathbf{p}$ leaving the set $W^t_{k'}$ for the first time. We consider the following two cases:

1. The piece $\mathbf{p}$ was part of $v^t_{k'}$, i.e., was above height $(1+\varepsilon)^{k'}\text{OPT}$ on some machine j. In this case, $\mathbf{p}$ contributes a decrease in $v^t_{k'}$ as well.

2. The piece $\mathbf{p}$ was not part of $v^t_{k'}$, i.e., was below height $(1+\varepsilon)^{k'}\text{OPT}$ on some machine j. In this case, the loss of $\mathbf{p}$ from $z^t_{k'}$ is due to the machine j leaving the set $M^t_{k'}$. We now consider two cases:

   (a) $j \in M^{t'}_{k'}$ for all rounds t' from the beginning of the phase till round t. Such a machine causes $z^t_{k'}$ to decrease by an amount equal to $(1+\varepsilon)^{k'}\text{OPT}$ over and above the decrease in $v^t_{k'}$.

   (b) $j \notin M^{t'}_{k'}$ for some t' from the beginning of the phase till round t. Let t' be the latest such round. Note that $\mathbf{p}$ moved to machine j after round t'. When $\mathbf{p}$ moved to j, it was placed at the top, i.e., above height $(1+\varepsilon)^{k'}\text{OPT}$. Later $\mathbf{p}$ was moved below height $(1+\varepsilon)^{k'}\text{OPT}$ when it was swapped with some piece $\mathbf{p}'$ not in $z^t_{k'}$ and when $\mathbf{p}'$ left $v^{t''}_{k'}$ for some $t' < t'' \leq t$. Thus the loss of $\mathbf{p}$ from $z^t_{k'}$ can be charged to the decrease in $v_{k'}$.

Since the total decrease in $v_{k'}$ is at most $\frac{1}{3}\varepsilon v^0_{k'}$, the total decrease in $z_{k'}$ due to cases 1 and 2b is bounded by $\frac{1}{3}\varepsilon v^0_{k'}$. We now upper bound the decrease in $z_{k'}$ due to case 2a as follows.

For a machine in $M^0_{k'}$ to leave $M^t_{k'}$ for the first time, either it must belong to $M^0_{k'} \setminus M^0_{k'+1}$ or be a poor machine. The total number of such machines is at most

$$(m^0_{k'} - m^0_{k'+1}) + \frac{v^0_{k'}}{3(1+\varepsilon)^{k'}\text{OPT}}.$$

Each of these machines may contribute to a loss of $(1+\varepsilon)^{k'}\text{OPT}$ in $z_{k'}$. Hence, we have

$$\begin{aligned} z_{k'} &\geq v^0_{k'} + m^0_{k'}(1+\varepsilon)^{k'}\text{OPT} \\ &\quad -(m^0_{k'} - m^0_{k'+1})(1+\varepsilon)^{k'}\text{OPT} - \frac{v^0_{k'}}{3} \\ &= \frac{2}{3}v^0_{k'} + m^0_{k'+1}(1+\varepsilon)^{k'}\text{OPT} \\ &\geq \frac{2}{3}v^0_{k'} + m^0_k(1+\varepsilon)^4\text{OPT} \end{aligned}$$

where the last inequality follows from the assumption that $m^0_{k'+1} = m^0_{k+5} > m^0_k/(1+\varepsilon)^k$. ∎

Consistently high volume movable to low loaded machines. There are m^*_{k+1} machines with load at least $(1+\varepsilon)^{k+1}\text{OPT}$ in some round in the phase, where m^*_{k+1} is upper bounded as in (4.1). In the optimum solution these machines can hold at most a volume of $m^*_{k+1}\text{OPT}$. Hence a volume of at least $z_{k'} - m^*_{k+1}\text{OPT}$ is in $Z_{k'}$ and which OPT assigned to machines that were never in M_{k+1} during the phase.

We lower bound this volume as follows.

$$\begin{aligned} & z_{k'} - m^*_{k+1}\text{OPT} \\ &\geq \frac{2}{3}v^0_{k'} + m^0_k(1+\varepsilon)^4\text{OPT} - m^0_k\text{OPT} - \frac{v^0_{k+1}}{3(1+\varepsilon)^k} \\ &\geq m^0_k((1+\varepsilon)^4 - 1)\text{OPT} + \frac{2}{3}v^0_{k'} - \frac{v^0_{k+1}}{3(1+\varepsilon)^k} \\ &> 4\varepsilon \cdot m^0_k\text{OPT} + \frac{2}{3}v^0_{k'} - \frac{v^0_{k+1}}{3(1+\varepsilon)^k}. \end{aligned}$$

Next we use the fact that

$$\begin{aligned} & v^0_{k+1} - v^0_{k'} \\ &\leq m^0_{k+1}((1+\varepsilon)^{k'} - (1+\varepsilon)^{k+1})\text{OPT} \\ &= m^0_{k+1}(1+\varepsilon)^k((1+\varepsilon)^{k'-k} - (1+\varepsilon))\text{OPT} \\ &= m^0_{k+1}(1+\varepsilon)^k((1+\varepsilon)^4 - (1+\varepsilon))\text{OPT} \\ &\leq m^0_{k+1}(1+\varepsilon)^k \cdot 6\varepsilon \cdot \text{OPT} \\ &\leq m^0_k(1+\varepsilon)^k \cdot 6\varepsilon \cdot \text{OPT}. \end{aligned}$$

The first inequality follows since each of m^0_{k+1} machines can hold at most $((1+\varepsilon)^{k'} - (1+\varepsilon)^{k+1})\text{OPT}$ volume between the levels $(1+\varepsilon)^{k'}\text{OPT}$ and $(1+\varepsilon)^{k+1}\text{OPT}$. The inequality on the fifth line is due to the fact that ε is small enough and hence $(1+\varepsilon)^4 \leq 1+7\varepsilon$. The inequality on the last line follows from the fact that $m^0_{k+1} \leq m^0_k$. Thus we conclude

$$\begin{aligned}\frac{v^0_{k+1}}{3(1+\varepsilon)^k} &\leq \frac{v^0_{k'}}{3(1+\varepsilon)^k} + 2\varepsilon \cdot m^0_k \text{OPT} \\ &\leq \frac{1}{3}v^0_{k'} + 2\varepsilon \cdot m^0_k \text{OPT}.\end{aligned}$$

Hence

$$\begin{aligned}& z_{k'} - m^*_{k+1}\text{OPT} \\ &> 4\varepsilon \cdot m^0_k \text{OPT} + \frac{2}{3}v^0_{k'} - \frac{1}{3}v^0_{k'} - 2\varepsilon \cdot m^0_k \text{OPT} \\ &> 2\varepsilon \cdot \text{OPT}.\end{aligned}$$

Thus we just proved that the total volume of pieces in $Z_{k'}$ that OPT assigns to machines never in M_{k+1} during the phase is more than $2\varepsilon \cdot \text{OPT}$. This contradicts the lemma below.

LEMMA 4.6. *Consider a set of pieces of tasks that are in $Z_{k'}$ such that in the optimum solution, they are assigned to some subset of machines M which are never in M_{k+1} during the phase. Then the volume of such pieces is at most $2\varepsilon \cdot \text{OPT}$.*

Proof. Fix a task i such that OPT assigns some pieces of task i to machines in M which are never in M_{k+1} during the phase. Fix a machine $j^* \in S_i \cap M$. In the beginning of the phase, the fraction p_{ij^*} is at least $\eta = \varepsilon/m^2$. In any single round, by the Bounded step rule, p_{ij^*} may increase by a factor at most $(1+\varepsilon)$ by pushing more pieces of task i onto j^* or may decrease by a factor at most $(1+\varepsilon)$ by pulling some pieces of task i from j^*. We call a round *saturated* if p_{ij^*} increases in this round by a factor of $(1+\varepsilon)$. We call a round *unsaturated* otherwise.

We now argue that the number of unsaturated rounds in a phase is at least $\tau/3$ where $\tau = \Theta(\frac{1}{\varepsilon}\log\frac{m}{\varepsilon})$ is the total number of rounds in a phase. Assume on the contrary that the number of saturated rounds is more than $2\tau/3$. Thus p_{ij^*} would increase by at least a factor of $(1+\varepsilon)^{2\tau/3}$ in the saturated rounds and decrease by a factor of at most $(1+\varepsilon)^{\tau/3}$ in the unsaturated rounds, implying that the overall increase in p_{ij^*} is by a factor of

$$(1+\varepsilon)^{2\tau/3-\tau/3} = (1+\varepsilon)^{\tau/3} > m^2/\varepsilon.$$

However this is impossible, since we always have $\eta = \varepsilon/m^2 \leq p_{ij^*} \leq 1$.

We now observe that in any unsaturated round, p_{ij} must decrease by a factor of $(1+\varepsilon)$ for *all* machines $j \in M_{k'}$ such that $p_{ij} \geq \eta(1+\varepsilon) = \varepsilon(1+\varepsilon)/m^2$. (Furthermore, these pieces must be pushed to a machine that is not in M_{k+1}.) This holds since task i is maximally greedy and in an unsaturated round, there is an opportunity to move pieces to machine $j^* \notin M_{k+1}$. The condition $p_{ij} \geq \eta(1+\varepsilon)$ is needed since we require $p_{ij} \geq \eta$ in all rounds. Note also that $L_j/L_{j^*} \geq (1+\varepsilon)^3$ for any $j \in M_{k'}$.

For the purpose of analysis, we may assume that while deciding which pieces move from the machines in $M_{k'}$ to the machines not in M_{k+1}, we give a preference to the pieces in $Z_{k'}$. Let $z_{k'}(i)$ be the current volume of pieces in $Z_{k'}$ that belong to task i. As argued above, if $p_{ij} \geq \varepsilon(1+\varepsilon)/m^2$ for some $j \in M_{k'}$, than p_{ij} must drop by a factor of $(1+\varepsilon)$ in an unsaturated round. Thus after any unsaturated round, $z_{k'}(i)$ becomes at most

$$\frac{1}{1+\varepsilon}\left(z_{k'}(i) - \frac{\varepsilon(1+\varepsilon)}{m^2} \cdot m \cdot w_i\right) + \frac{\varepsilon(1+\varepsilon)}{m^2} \cdot m \cdot w_i.$$

This holds since at most $\varepsilon(1+\varepsilon)/m^2 \cdot m \cdot w_i$ volume in $z_{k'}(i)$ lies on the machines $j \in M_{k'}$ with $p_{ij} < \varepsilon(1+\varepsilon)/m^2$ and the remaining volume in $z_{k'}(i)$ must decrease by $(1+\varepsilon)$ factor. After simplifying, the above expression equals

$$\frac{z_{k'}(i)}{1+\varepsilon} + \frac{\varepsilon^2 \cdot w_i}{m}.$$

Since the number of unsaturated rounds is at least $\tau/3$ where $\tau = \Theta(\frac{1}{\varepsilon}\log\frac{m}{\varepsilon})$, at the end of the phase $z_{k'}(i)$ becomes at most $\frac{2\varepsilon \cdot w_i}{m}$. Thus we conclude that at the end of the phase, we have

$$z_{k'} = \sum_{i=1}^{n} z_{k'}(i) \leq \frac{2\varepsilon}{m}\sum_{i=1}^{n} w_i \leq \frac{2\varepsilon}{m} \cdot m \cdot \text{OPT} = 2\varepsilon \cdot \text{OPT}.$$

This completes the proof of Lemma 4.6. ∎

Due to the contradiction, we conclude that at least v_{k+1} or v_{k+4} must decrease by a factor of at least $\varepsilon/3$. Thus the proof of Lemma 4.4 is complete.

References

[1] B. Awerbuch and R. Khandekar. Greedy distributed optimization of multi-commodity flows. In *Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing*, pages 274–283, 2007.

[2] S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 169–178, 2007.

[3] G. Christodolou, V. S. Mirrokni, and A. Sidiropolous. Convergence and approximation in potential games. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*, pages 349–360, 2006.

[4] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 772–781, 2005.

[5] S. Fischer, H. Räcke, and B. Vöcking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 653–662, 2006.

[6] S. Fischer and B. Vöcking. On the evolution of selfish routing. In *12th Annual European Symposium on Algorithms*, Lecture Notes in Computer Science 3221, pages 323–334, 2004.

[7] S. Fischer and B. Vöcking. Adaptive routing with stale information. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing*, pages 276–283, 2005.

[8] M. Goemans, V. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 142–151, 2005.

[9] V. Mirrokni and A. Vetta. Convergence issues in competitive games. In *7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, Lecture Notes in Computer Science 3122, pages 183–194, 2004.

[10] T. Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, Department of Computer Science, 2002. See also `http://www.cs.cornell.edu/timr/`.

[11] A. Skopalik and B. Vöcking. Inapproximability of convergence in congestion games. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2007.