

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Theoretical Computer Science

www.elsevier.com/locate/tcs

Parallel Load Balancing on constrained client-server topologies [☆]

Andrea Clementi ^{a,*}, Emanuele Natale ^b, Isabella Ziccardi ^{c,*}

^a *Università di Roma Tor Vergata, Rome, Italy*

^b *Université Côte d'Azur, CNRS, INRIA, Sophia Antipolis, France*

^c *Università dell'Aquila, L'Aquila, Italy*

ARTICLE INFO

Article history:

Received 28 September 2020

Received in revised form 7 July 2021

Accepted 18 September 2021

Available online xxxx

Communicated by C. Kaklamani

Keywords:

Parallel balanced allocations

Balls-into-bins processes

Randomized algorithms

ABSTRACT

We study parallel *Load Balancing* protocols for the client-server distributed model defined as follows. There is a set \mathcal{C} of n clients and a set \mathcal{S} of n servers where each client has (at most) a constant number $d \geq 1$ of requests that must be assigned to some server. The client set and the server one are connected to each other via a fixed bipartite graph: the requests of client v can only be sent to the servers in its neighborhood $N(v)$. The goal is to assign every client request so as to minimize the maximum load of the servers.

In this setting, efficient parallel protocols are available only for dense topologies. In particular, a simple protocol, named RAES, has been recently introduced by Becchetti et al. [1] for regular dense bipartite graphs. They show that this symmetric, non-adaptive protocol achieves constant maximum load with parallel completion time $\mathcal{O}(\log n)$ and overall work $\mathcal{O}(n)$, w.h.p.

Motivated by proximity constraints arising in some client-server systems, we analyze RAES over almost-regular bipartite graphs where nodes may have neighborhoods of small size. In detail, we prove that, w.h.p., the RAES protocol keeps the same performances as above (in terms of maximum load, completion time, and work complexity, respectively) on any almost-regular bipartite graph with degree $\Omega(\log^2 n)$.

Our analysis significantly departs from that in [1] since it requires to cope with non-trivial stochastic-dependence issues on the random choices of the algorithmic process which are due to the worst-case, sparse topology of the underlying graph.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

1.1. The framework and our goal

We study parallel *Load Balancing* allocation in client-server distributed systems. We have a client-server bipartite graph $G(V = (\mathcal{C}, \mathcal{S}), E)$ where: \mathcal{C} is the set of *clients*, each one having a number of requests which is bounded by some constant

[☆] This work has been presented at the 32nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'20).

* Corresponding authors.

E-mail addresses: clementi@mat.uniroma2.it (A. Clementi), natale@unice.fr (E. Natale), isabella.ziccardi@graduate.univaq.it (I. Ziccardi).

¹ AC work on this project was partially supported by the ALBLOTECH project n. E89C20000620005 under the University of Rome Tor Vergata Programme "Beyond Borders".

<https://doi.org/10.1016/j.tcs.2021.09.026>

0304-3975/© 2021 Elsevier B.V. All rights reserved.

$d \geq 1$; \mathcal{S} is the set of servers; the edge set E represents the client-server assignments which are considered admissible because of proximity constraints (a client can send a request only to the servers in its neighborhood).

The algorithmic goal of the entities is to assign the requests in parallel so as to minimize the maximum server load.²

To analyze the performance of the proposed protocol for the above distributed task, we adopt the standard synchronous distributed model introduced for parallel *balls-into-bins* processes by Adler et al. in [2]: here, clients and servers are autonomous computing entities that can exchange information (only) over the edges of G . Adler et al. introduce the class of symmetric, non-adaptive protocols and show several tight bounds on the trade-offs between the maximum load and the complexity (i.e. *completion time* and *work complexity*³) of the proposed solutions. Informally, a protocol is said to be *symmetric* if the entities are anonymous, so all the clients (servers) act in the same way and, moreover, all possible request destinations are chosen independently and uniformly at random. The protocol is said to be *non-adaptive* if each client restricts itself to a fixed number of (possibly random) candidate servers in its neighborhood before communication starts. Symmetric, non-adaptive protocols have the practical merits to be easy to implement and more flexible [2]. Such solutions have interesting applications, such as Load Balancing in communication networks, request scheduling and hashing [3–6].

We notice that efficient symmetric, non-adaptive protocols are known (only) for dense regular bipartite graphs and almost-tight lower bounds are known for this important class of parallel protocols [2,1,7] (see also Subsection 1.3 for a short description of such results).

The main goal of this paper does not consist of improving previous solutions with respect to specific complexity measures. Rather, still aiming at efficient solutions that achieve bounded maximum load,⁴ we focus on symmetric, non-adaptive Load Balancing protocols that work over restricted, non dense graph topologies. This natural extension of previous work is inspired by possible network applications where: i) based on previous experiences, a client (a server) may decide to send (accept) the requests only to (from) a fixed subset of *trusted* servers (clients) and/or ii) clients and servers are placed over a metric space so that only non-random client-servers interactions turn out to be feasible because of proximity constraints. Such possible scenarios motivated previous important studies on sequential Load Balancing algorithms [8–10]. To the best of our knowledge, efficient solutions for non-dense graphs are in fact available only for the classic sequential model. Here, each client request is scheduled once at time so that, for instance, the well-known *best-of-k-choices* strategy [11] can be applied: the loads of the servers are updated at each assignment and the new considered request is assigned to a server that has the current minimal load out of k servers chosen independently and uniformly at random [8–10].

The RAES Algorithm. As for the parallel distributed model we adopt in this paper, in [1] Becchetti et al. propose a symmetric, non-adaptive algorithm, named RAES (for *Request a link, then Accept if Enough Space*), which is based on the well-known *Threshold* criterion [2]. This approach has been adopted in the context of simulation of PRAM algorithms over the Distributed Memory Machine (DMM) under the name of *c-collision process* [12]. Informally, RAES works in rounds, each consisting of two phases. Initially, each client has $d = \Theta(1)$ balls.⁵ In the first phase of each round, if client u has $d' \geq 1$ *alive* balls (i.e. to be still accepted by some server), u selects d' servers independently and uniformly at random (with replacement) from $N(u)$. It then submits each of the d' balls to each selected client. In the second phase of the round, each server accepts all requests received in the first phase of the current round, unless doing so would cause it to exceed the limit of cd accepted balls, where the parameter c is a suitable large constant; if this is the case, the server is said to be *saturated* and rejects all requests it received in the first phase of the current round. The algorithm completes when every client has no further balls to be submitted.

We observe that servers only give back Boolean answers to the clients requests and, moreover, if the algorithm terminates, the maximum load of the servers will be at most cd . Becchetti et al. prove⁶ that, over any Δ -regular bipartite graph with $\Delta = \Omega(n)$, RAES terminates within $O(\log n)$ rounds⁷ and the total work is $\Theta(n)$, *with high probability*⁷ (for short, *w.h.p.*).

1.2. Our contribution

We analyze the RAES algorithm on every bipartite graph $G(V = (\mathcal{C}, \mathcal{S}), E)$ of degree $\Delta = \Omega(\log^2 n)$ (recall that $|\mathcal{C}| = |\mathcal{S}| = n$) that satisfies the following almost-regularity property: the ratio between the maximal degree of the servers and the minimal degree of the client is bounded by some absolute constant.

Under this setting, we prove it is possible to set the algorithm parameter c as a sufficiently large constant, such that, for every constant request number d , the RAES process terminates within $O(\log n)$ rounds and requires $\Theta(n)$ work, *w.h.p.*

The formal definition of almost-regular graphs and the formal statement of the above result are given in Theorem 1, while we here remark that the considered notion of almost regularity allows a certain variance of the degrees of entities of the same type. Just as a (“non-extremal”) example, we may consider a bipartite graph where: most of the clients have

² The load of a server is the overall number of requests which have been assigned to it.

³ The work complexity is the overall number of exchanged messages performed by the protocol.

⁴ According to our parameter setting, the maximum load is clearly at least d and we aim at keeping an $O(d)$ bound for it.

⁵ The terms *ball* and *request* will be used interchangeably.

⁶ Even though, not related to our context, we remark that Becchetti et al. shows that RAES can be used to efficiently construct a bounded-degree expander subgraph of G [1].

⁷ As usual, we say that an event E holds with high probability if a constant $\gamma > 0$ exists such that $\mathbf{P}(E) \geq 1 - n^{-\gamma}$.

(minimal) degree $\Theta(\log^2 n)$, while few of them have degree $\Theta(\sqrt{n})$; most of the servers have (maximal) degree $\Theta(\log^2 n)$, while few of them have degree $o(\log n)$.

Algorithm analysis: an overview. In the case of dense graphs, the key-fact exploited by Becchetti et al.'s analysis of the RAES algorithm [1] is the following. Since each client has $\Theta(n)$ servers in its neighborhood, it is possible to fix a sufficiently large constant c , such that, at every round, the fraction of non-saturated⁸ servers in the neighborhood $N(v)$ of every client v is always at least $1/2$. Thanks to a basic counting argument, this fact holds *deterministically* and *independently* of the previous load configurations yielded by the process. So, every alive client request has probability at least $1/2$ to be accepted at each round: this allows to get a logarithmic completion time of RAES on dense graphs.

In the case of non-dense graphs (i.e. for node degree $o(n)$), the key property above does not hold deterministically: the fraction of non-saturated servers in a fixed neighborhood is a random variable that can even take value 1 and, very importantly, it depends on the graph topology and on the random choices performed by the nodes during the previous rounds. This scenario makes the analysis considerably harder than that of the dense case.

To cope with the issues above, we consider a variant of RAES, called SAER (*Stop Accepting if Exceeding Requests*) that works like RAES with the exception that, whenever a server v , in the second phase of a given round, gets an overall load larger than cd , then v rejects all requests arrived in the first phase of the current round and it becomes *burnt*. Once a server gets burnt, it will never accept any further request for all successive rounds (see Algorithm 2 in Section 3).

We consider this variant of RAES since the notion of burnt server is more restrictive and somewhat more “static” w.r.t. that of saturated servers used in RAES: this allows us to analyze the number of burnt servers and that of alive requests in a rigorous way. Moreover, being a natural alternative, its analysis may have a per se interest.

We then show a simple coupling between RAES and SAER that implies that both the completion time and the work of the former cannot be stochastically larger than those of the latter. Thanks to this result, we can thus focus on the analysis of the SAER algorithm. Similarly to RAES, if this new version terminates, then each server will have load at most cd and, hence, the main technical issue is to provide a bound (if any) on the number of rounds required by SAER to let every client ball assigned to some server. For an arbitrary client v , we look at its server neighborhood $N(v)$ and we establish a clean recursive formula that describes the expected decreasing rate of the overall number $r_t(N(v))$ of requests that the neighborhood of v receives at round t . This expectation is derived for round t by conditioning on the sequence of the maximum fractions of burnt servers in any client's neighborhood produced by the algorithmic process at rounds $1, 2, \dots, t-1$. It turns out that, for a sufficiently large c , the conditional expected decreasing rate of $r_t(N(v))$ is exponential. Then, using a coupling argument, we derive a concentration bound for this rate that holds as long as the conditional expectation of $r_t(N(v))$ keeps of magnitude $\Omega(\log n)$. To complete our argument, we consider a further (and final) stage of the process⁹ that starts when $r_t(N(v)) = O(\log n)$: here, we do not look anymore at the decreasing rate of $r_t(N(v))$, rather we show that, w.h.p., the fraction of burnt servers in $N(v)$ can increase, along a time window of length $\Theta(\log n)$, by an overall additive factor of magnitude at most $O(1/c)$. Thanks to this fact, we can then show that the $O(\log n)$ requests that survived the first stage have high chances to be assigned during this last stage if the latter lasts $\Theta(\log n)$ additional rounds.

1.3. Previous work

Load Balancing algorithms have been the subject of a long and extremely active line of research with important applications in several topics of Computer Science such as hashing, PRAM simulation, and scheduling. A well-established and effective way to model such problems is by using the classic *balls-into-bins* processes. In such processes, there are typically m balls that must be assigned to n bins. In what follows, we use this framework to describe those previous results which are more related to the setting of this work.

Sequential algorithms on the complete bipartite graph. It is well-known that if n balls are thrown independently and uniformly at random into n bins, the maximum load of a bin is bounded by $\Theta(\log n / \log \log n)$, w.h.p. (see for instance [13]). Azar et al. [11] proved the following breakthrough result. Assume the balls are assigned sequentially, one at a time and, for each ball, $k \geq 1$ bins are chosen independently and uniformly at random, and the ball is assigned to the least full bin (with ties broken arbitrarily). This greedy strategy is also known as “best of k choices”. Then, they prove that the final maximum load is $\Theta(\log \log n / \log k + 1)$, w.h.p. A similar result was also derived in a different version of the model by Karp et al. in [14]. Berenbrink et al. extended the analysis of the greedy algorithm for the heavily-loaded case $m \gg n$ [15]. Then, several versions of this sequential algorithm have been studied by considering, for instance, non-uniform choices in the assignment process [16–18]. Moreover, several works addressed weighted balls [19–21], while the case of heterogeneous bins was studied in [18]. Recently, balls-into-bins processes have also been analyzed over game theoretic frameworks [22,23].

Sequential algorithms on restricted bipartite graphs. Sequential algorithms for restricted balls-bins (i.e. client-server) topologies have been considered in [24,9,10]: here, each ball $u = 1, \dots, m$ comes with its admissible cluster of bins and decides its strategy according to the current loads in its cluster determined by the choices of the previous balls $u' < u$. In this setting, Kenthapadi and Panigrahy [10] analyze the well-known sequential greedy algorithm [11]: each client u , in turn,

⁸ Recall that a server is saturated at round t if its load is larger than cd .

⁹ Notice that this stage is only in our analysis and not on the protocol, the latter being symmetric and non-adaptive.

chooses a pair of servers uniformly at random from $N(u)$ and assigns the ball to the server having the current minimum load. They prove that, if the size of every $N(u)$ is at least $n^{\Theta(1/\log \log n)}$, then the greedy algorithm achieves maximum load $\Theta(\log \log n)$, w.h.p. In [9], Godfrey analyzed the sequential greedy algorithm on the input model where a *random* cluster of servers $N(u)$ is assigned to each client u before the algorithm starts. In more detail, each client u places its ball in a uniform-random server among those in $N(u)$ with the current fewest number of balls. He proves that, if the random subsets $N(u)$ are chosen according to any fixed *almost-uniform* distribution over the server set \mathcal{S} and the subsets $N(u)$ have size $\Omega(\log n)$, then the greedy algorithm achieves optimal maximum load, w.h.p. The overall work is $\Theta(n\Delta_{\max}(C))$, where $\Delta_{\max}(C) = \max\{|N(u)| : u \in C\}$. Further bounds are determined when the overall number m of balls is smaller than the size of the server set $|\mathcal{S}| = n$. Berenbrink et al. [24] consider the sequential framework adopted in [9] and improve the analysis of the greedy algorithm along different directions. In detail, they consider weaker notions of almost-uniform distributions for the random server clusters assigned to the clients and, moreover, they also consider an input framework formed by deterministic, worst-case server clusters of size $\beta \log n$. In the case where the overall number of balls is $n < \alpha m$, with any $\alpha < 1/12$ and $\beta > 18$, they show that a suitable version of the sequential greedy algorithm achieves maximum load 1, w.h.p. Notice that the greedy algorithm adopted in [10,9] does require every server to give information to their clients about its current load: in some applications, this feature of the algorithm might yield critical issues in terms of privacy and security of the involved entities [25,26]. On the other hand, we notice that, the simple threshold approach adopted by both RAES and SAER can be implemented in a fully decentralized fashion so that the clients cannot get a good approximation about the current load of the servers (see also the remarks after Algorithm 2 in Section 3).

Parallel algorithms on the complete bipartite graph. Inspired by applications arising from parallel distributed systems, a rich and active research has been focused on computational entities which are able to communicate with each other (with some constraints that depend on the specific version of the model). Then, protocols operate in synchronous rounds, in each of which balls and bins exchange messages once. In [2], Adler et al. consider some non-adaptive symmetric protocols and analyze their performances in terms of maximum load, number of rounds, and message complexity. For instance, they introduce a parallelization of the greedy algorithm [11] and show that for any constant number r rounds and for any constant number of random choices k , it achieves maximum load $\mathcal{O}((\log n / \log \log n)^{1/r})$, w.h.p. They also give a more complex greedy algorithm that works in $\log \log n / \log k + 2k + \mathcal{O}(1)$ rounds and achieves $\log \log n / \log k + 2k + \mathcal{O}(1)$ maximum load, w.h.p. Interestingly enough, they prove that the above performance trade-offs are essentially optimal for the restricted class of non-adaptive, symmetric algorithms. This class also includes the *Threshold* algorithms where, informally speaking, at every round, every bin, that receives more than a fixed threshold T of balls, re-throws the excess balls in the next round (such rejected balls can be chosen in an arbitrary “fair” way). Parallel Threshold algorithms have been introduced by Lenzen et al. in [7] for the heavily-loaded case, i.e. when $m \gg n$. Finally, we mention some adaptive and/or non-symmetric protocols on the complete graph that have been presented in recent work (e.g. [27,28,7,29]) that achieve significantly better performances than symmetric and/or non-adaptive ones [2]. Such strategies are rather complex and so their setting is far from the aim of this paper (as discussed in the previous subsection, this being the analysis of basic, non-adaptive symmetric protocols over restricted client-server topologies).

Parallel algorithms on restricted bipartite graphs. The only rigorous analysis of parallel protocols for restricted client-server topologies we are aware of is that in [1] by Becchetti et al. for the RAES protocol which has been discussed in the previous part of this introduction.

2. Preliminaries

In the *Load Balancing* problem we have a system formed by a client-server bipartite graph $G(V = (\mathcal{C}, \mathcal{S}), E)$ where:

- the subset $\mathcal{C} = \{v_1, \dots, v_n\}$ represents the set of *clients*;
- the subset $\mathcal{S} = \{u_1, \dots, u_n\}$ represents the set of *servers*;
- the edge set E determines, for each client v , the subset $N(v)$ of servers the client v can make a *request* to (i.e. it can send a *ball*¹⁰). For any node (client or server) $w \in V = (\mathcal{C}, \mathcal{S})$, we denote its degree in G as $\Delta(w)$, i.e. $\Delta(w) = |N(w)|$ and we define

$$\Delta_{\min}(\mathcal{C}) = \min\{\Delta(v) : v \in \mathcal{C}\} \text{ and } \Delta_{\max}(\mathcal{S}) = \max\{\Delta(u) : u \in \mathcal{S}\}.$$

At the beginning, each client has (at most d) balls where $d \geq 1$ is an arbitrary constant (w.r.t. n) that, in the sequel, we call *request number*, and the goal is to design a parallel distributed protocol that assigns each ball of every client $v \in \mathcal{C}$ to one server in $N(v)$.

According to previous work [2,29], we study the Load Balancing problem over the fully-decentralized computational model \mathcal{M} where bi-directional node communications take place only along the edges in E , in synchronous rounds. In more detail, each round consists of two phases: in the first phase, each client v sends its unsettled (i.e. alive) balls to a selection of servers in $N(v)$. Then, in the second phase, servers answer to each of the received balls by either accepting or rejecting.

¹⁰ Recall that the terms *ball* and *request* will be used interchangeably.

We remark that our considered algorithms work under the following further model constrains: clients may only send the ball IDs¹¹; servers may only answer each ball request with one bit (accept/reject); moreover, there is no global labeling of the nodes of G , each node v just keeps a local labeling of its links.

We analyze the cost of the proposed algorithmic solution with respect to two fundamental complexity measures:

- the *completion time* which is defined as the number of rounds required by the protocol to successfully assign all the client balls to the servers;
- the (overall) *work* which is defined as the overall number of exchanged messages among the nodes of the network during the protocol's execution.

3. Two simple protocols for Load Balancing and their performances

We analyze two simple protocols for Load Balancing, the first one is called RAES (*Request a link, then Accept if Enough Space*) and it is the one introduced in [1], while the second one is a variant of RAES and it is named SAER (*Stop Accepting if Exceeding Request*). They are both based on a simple, non-adaptive threshold criterion the servers use to accept or reject the incoming balls. The protocol is organized in rounds and, in turn, each round consists of two phases. For the sake of readability, we consider the case where every client has exactly d balls, where the request number d is an arbitrary fixed constant: the analysis of the general case ($\leq d$) is in fact similar.

The RAES protocol was informally described in the introduction, a more formal description is given in Algorithm 1.

Algorithm 1 Protocol RAES(c, d).

```

1: PHASE 1: ▷  $d_v^{\text{out}}$ : current number of the balls  $v \in C$  which have been accepted by some server
2: for  $v \in C$  do
3:    $v$  picks  $d - d_v^{\text{out}}$  neighbors in  $G$  independently and uniformly at random (with replacement)
4:    $v$  submits a ball request to each of them
5: end for
6: PHASE 2: ▷  $d_u^{\text{in}}$ : current number of the balls accepted by  $u \in S$ 
7: for  $u \in S$  do
8:   if  $u$  received  $\leq cd - d_u^{\text{in}}$  ball requests in the previous phase then
9:      $u$  accepts all of the balls received in Phase 1 and updates  $d_u^{\text{in}}$ 
10:   else
11:      $u$  rejects all the balls received in Phase 1 of the current round
12:   end if
13: end for
14: for  $v \in C$  do
15:    $v$  updates its value  $d_v^{\text{out}}$ 
16:   if  $d_v^{\text{out}} = 0$  then
17:      $v$  gets into the final state DONE and terminates.
18:   end if
19: end for

```

To analyze the performance of RAES we consider its variant, the SAER protocol, that adopts a more restrictive way to assign balls to the servers: informally, once a server has received a certain overall number (i.e. a threshold) of balls, it will reject the new balls for all the next steps of the process. The formal description of SAER is given in Algorithm 2.

Remarks. Some simple facts easily follow from the protocol description above. (i) The protocols complete at round $T \geq 1$ and only if every client has successfully placed all its d balls within round T . If this happens, then the maximum load of the servers is clearly bounded by cd . The main technical question is thus to provide bounds on concentration on the completion time of the protocols and on their performed work. This issue will be the subject of the next section.

As for the decentralized implementation, we observe that the knowledge of the parameter c (which, in turn, depends on the degree of the underlying almost-regular bipartite graph - see Theorem 1 in the next subsection) is required only by the servers while clients need no knowledge of global parameters. Interestingly enough, this fact implies that, for reasons of security and/or privacy, the servers may suitably choose c so that the clients cannot get any good approximation of their current load.

3.1. The main theorem

According to the definition of client-server bipartite graphs and those of the two protocols RAES and SAER(c, d) we gave above, we can state our main technical contribution as follows.

¹¹ It suffices that each client keeps a local labeling of its ball set.

Algorithm 2 Protocol SAER(c, d).

```

1: PHASE 1:                                     ▷  $d_v^{\text{out}}$ : current number of the balls of  $v \in C$  which have been accepted by some server
2: for  $v \in C$  do
3:    $v$  picks  $d - d_v^{\text{out}}$  neighbors in  $G$ , independently and uniformly at random (with replacement)
4:    $v$  submits a ball request to each of them
5: end for
6: PHASE 2:                                     ▷  $d_u^{\text{in}}$ : current number of the balls accepted by  $u \in S$ 
7: for  $u \in S$  do
8:   if  $u$  is BURNT then
9:      $u$  rejects all the balls received in Phase 1 of the current round
10:  else
11:    if  $u$  received  $> cd$  balls since the start of the process then
12:       $u$  rejects all the balls received in Phase 1 of the current round and becomes BURNT
13:    else
14:       $u$  accepts all of the balls received in Phase 1 and updates  $d_u^{\text{in}}$ 
15:    end if
16:  end if
17: end for
18: for  $v \in C$  do
19:    $v$  updates its value  $d_v^{\text{out}}$ 
20:   if  $d_v^{\text{out}} = 0$  then
21:      $v$  gets into the final state DONE and terminates.
22:   end if
23: end for

```

Theorem 1 (Performances of SAER and RAES processes). For any constants $\eta, \rho \in \mathbb{R}^+$ with $\rho \geq 1$, and for any sufficiently large $n \in \mathbb{N}$, let $G((C, S), E)$ be any bipartite graph such that: $|C| = |S| = n$, $\Delta_{\min}(C) \geq \eta \log^2 n$, and $\Delta_{\max}(S)/\Delta_{\min}(C) \leq \rho$. For any constant $d \in \mathbb{N}$, consider the Load Balancing problem on G with request number d . Then, for any sufficiently large constant $c > 0$,¹² both the protocols RAES(c, d) and SAER(c, d) have completion time $\mathcal{O}(\log n)$ and their work is $\Theta(n)$, w.h.p.

A simple counting argument implies that $\Delta_{\min}(C) \leq \Delta_{\max}(S)$ for every bipartite graph while Theorem 1 requires the “almost-regularity” hypothesis $\Delta_{\max}(S)/\Delta_{\min}(C) = \Theta(1)$. We remark that this condition allows a relative-large variance of the node degree. For instance, the theorem holds for a topology where: the minimum client degree and the maximum server degree are $\Theta(\log^2 n)$, some clients have degree $\Theta(\sqrt{n})$, while some servers have (minimal) degree $\Theta(1)$.

4. Proof of Theorem 1 for SAER

To prove the claim of Theorem 1 for the SAER process, in Subsection 4.1, we provide an upper bound on the fraction of burnt servers that holds for each round of a time window of logarithmic length. In Subsection 4.2, we first show how to easily get the logarithmic bound on the completion time and, then, we derive a linear bound on the work performed by this protocol.

4.1. On the fraction of burnt servers

In this section, we prove the claim of Theorem 1 for SAER on any bipartite graph $G(V = (C, S), E)$ that satisfies the conditions: $\Delta_{\min}(C) \geq \eta \log^2 n$ and $\Delta_{\max}(S)/\Delta_{\min}(C) \leq \rho$. Since the protocol SAER makes a crucial use of burnt servers, in what follows, we define this notion and some important random variables of the algorithmic process which are related to it. For every round $t \geq 1$ and every server $u \in S$, let $r_t(u)$ be the random variable indicating the number of balls that server u receives at round t .

Definition 2 (Burnt servers). For every integer $t \geq 1$, we say that a server $u \in S$ is BURNT at round t of the process SAER if

$$\sum_{i=1}^t r_i(u) > cd.$$

Moreover, for every client $v \in C$, define $B_t(v)$ as the fraction of BURNT servers in the neighborhood of v at time t , i.e.,

$$B_t(v) = \frac{\{u \in S : u \in N(v) \text{ and } \sum_{i=1}^t r_i(u) > cd\}}{\Delta(v)}.$$

We also define B_t as the maximum fraction of burnt nodes in any client’s neighborhood at round t , i.e., $B_t = \max_{v \in C} B_t(v)$.

¹² Our analysis will show that the value of c depends (only) on the constants η and ρ .

The proof of Theorem 1 relies on the following upper bound on the fraction of burnt servers that holds for a time window of logarithmic length.

Lemma 3 (On the fraction of burnt servers). *Under the hypothesis of Theorem 1, for any constant $c \geq \max(32\rho, 288/(\eta d))$, it holds*

$$\Pr \left(\bigcap_{1 \leq t \leq 3 \log n} \left\{ B_t \leq \frac{1}{2} \right\} \right) \geq 1 - \frac{1}{n^2}. \quad (1)$$

The next subsection is devoted to the proof of Lemma 3.

4.1.1. Proof of Lemma 3

We start by defining the random variables that describe the SAER process.

Definition 4. For every round $t \geq 1$ and for every client $v \in \mathcal{C}$, let $r_t(N(v))$ be the overall number of balls that all the servers in the neighborhood $N(v) \subseteq \mathcal{S}$ receive at round t ; moreover, let r_t be the maximum number of balls that every server neighborhood receives at round t . Formally,

$$r_t(N(v)) = \sum_{u \in N(v)} r_t(u) \quad \text{and} \quad r_t = \max_{v \in \mathcal{C}} r_t(N(v)).$$

We observe that since a burnt server at round t has received more than cd balls since the start of the process, for every $v \in \mathcal{C}$ and every $t \geq 1$,

$$B_t(v) \leq \frac{1}{cd\Delta(v)} \sum_{i=1}^t r_i(N(v)). \quad (2)$$

We also name the expression in the r.h.s. of the inequality above since it will be often used in our analysis.

Definition 5. For every round $t \geq 1$ and every client $v \in \mathcal{C}$, let

$$K_t(v) = \frac{1}{cd\Delta(v)} \sum_{i=1}^t r_i(N(v)) \quad \text{and} \quad K_t = \max_{v \in \mathcal{C}} K_t(v).$$

Notice that the above definitions and (2) easily imply that, for every $v \in \mathcal{C}$ and every $t \geq 1$,

$$B_t \leq K_t \quad \text{and} \quad K_t(v) = K_{t-1}(v) + \frac{1}{cd\Delta(v)} r_t(N(v)). \quad (3)$$

We next write the random variable $r_t(N(v))$ in terms of more “elementary” random variables.

Definition 6. For every round $t \geq 1$, client $v \in \mathcal{C}$ and ball $i \in [d]$, let $a_t^{(i)}(v)$ be the binary random variable indicating whether the v 's i -th ball is still *alive* at round t in the SAER process, i.e., it has still not been accepted by some server at the beginning of round t . Let $\tilde{a}_t^{(i)}(v)$ be the equivalent random variable in the RAES process.

Definition 7. For every round $t \geq 1$, client $v \in \mathcal{C}$, server $u \in N(v) \subseteq \mathcal{S}$ and ball $i \in [d]$, let $z_t^{(i)}(v, u)$ be the binary random variable indicating whether the (random) contacted server for the v 's i -th ball at round t is u in the SAER process. Let $\tilde{z}_t^{(i)}(v, u)$ be the equivalent random variable in the RAES process.

Notice that, according to the above definitions, if a ball i is not alive at round t , the corresponding value of the random variables $z_t^{(i)}(v, u)$ is not relevant for the evolution of the process and, thus, they will not be considered.

Then, for every fixed client $v \in \mathcal{C}$ and $t \geq 1$, we can write

$$r_t(N(v)) = \sum_{u \in N(v)} r_t(u) = \sum_{u \in N(v)} \sum_{w \in N(u)} \sum_{i=1}^d a_t^{(i)}(w) \cdot z_t^{(i)}(w, u). \quad (4)$$

The above random variables have the following useful properties.

Lemma 8.

1. For every $t \geq 1$, $i \in [d]$, $v \in \mathcal{C}$ and $u \in \mathcal{S}$, the random variables $z_t^{(i)}(v, u)$ and $a_t^{(i)}(v)$ are mutually independent.
2. For every $v \in \mathcal{C}$ and every choice of positive reals $b_j \leq 1$ for $j = 1, \dots, t - 1$, it holds

$$\Pr \left(a_t^{(i)}(v) = 1 \mid B_1 \leq b_1, \dots, B_{t-1} \leq b_{t-1} \right) \leq \prod_{j=1}^{t-1} b_j.$$

3. For every $t \geq 1$, the random variables $\left\{ z_t^{(i)}(v, u) \right\}_{v \in \mathcal{C}, u \in N(v), i \in [d]}$ are negatively associated.¹³

Proof. Claim 1 follows from the observation that SAER is non-adaptive and symmetric and, hence, at each round, each client $v \in \mathcal{C}$ chooses the (random) destination of its i -th request regardless of the value of $a_t^{(i)}(v)$ while the latter determines whether the request is really sent or not.

As for Claim 2, notice that $a_t^{(i)}(v) = 1$ iff v 's i -th request has been rejected at each previous round, and this happens iff the destination of the i -th request is a burnt server.

Finally, Claim 3 follows from the fact that, for each $v \in \mathcal{C}$, if $z_t^{(i)}(v, u) = 1$ for $u \in N(v)$ then, for every $u' \in \mathcal{C}$ with $u' \neq u$, it holds that $z_t^{(i)}(v, u') = 0$. Moreover, for each fixed $u \in \mathcal{S}$ the random variables $\{z_t^{(i)}(v, u), v \in \mathcal{C}, i \in [d]\}$ are independent. \square

Step-by-step analysis via induction. We first consider the first round of the process and give the following bound on the maximum number of balls a client neighborhood can receive.

Lemma 9 (First round). *Under the hypothesis of Theorem 1, for every $v \in \mathcal{C}$, with probability at least $1 - 1/n^3$, it holds*

$$r_1(N(v)) \leq 2d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})}$$

and

$$K_1 \leq \frac{2}{c} \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})}. \tag{5}$$

Proof. The random variable $r_1(N(v))$ can be written as in (4). Then, since for each $w \in \mathcal{C}$ and $u \in N(v)$, $z_1^{(i)}(w, u)$ is a Bernoulli random variable of parameter $1/\Delta(w)$, we get

$$\mathbf{E}[r_1(N(v))] = d \sum_{u \in N(v)} \sum_{w \in N(u)} \frac{1}{\Delta(w)} \leq d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})}.$$

From Claim 3 of Lemma 8, we can apply the Chernoff bound for negatively associated random variables with $\varepsilon = 1$ (Theorem 23) thus obtaining

$$\Pr \left(r_1(N(v)) \geq 2d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})} \right) \leq e^{-\frac{1}{3}d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})}} \leq e^{-\frac{1}{3}d\Delta_{\max}(\mathcal{S})} \leq \frac{1}{n^3}. \tag{6}$$

Observe that the last inequality holds for every sufficiently large n since, in every bipartite graph, we have $\Delta_{\max}(\mathcal{S}) \geq \Delta_{\min}(\mathcal{C}) \geq \eta \log^2 n$. Finally, from Definition 5, (6) and from a union bound, we get (5). \square

The next result is a key step of the proof of Lemma 3. We look at every fixed round $t \geq 2$ of the random process and derive, for every client $v \in \mathcal{C}$, an upper bound in concentration on the random variable $r_t(N(v))$, assuming some fixed bound on the variable K_{t-1} . This bound shows that, conditional on the bound sequence above, the number of alive balls in $N(v)$ decreases, at each round t , by a factor that explicit depends on the fraction of burnt servers at round $t - 1$. For every $v \in \mathcal{C}$ we give an upper bound on $\mathbf{E}[r_t(N(v))]$ conditional to some fixed upper bound on K_{t-1} .

Lemma 10 (Round $t \geq 2$ by induction). *Under the hypothesis of Theorem 1, for every client $v \in \mathcal{C}$ and every real k_1, \dots, k_{t-1}*

$$\mathbf{E}[r_t(N(v)) \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}] \leq d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})} \prod_{i=1}^{t-1} k_i. \tag{7}$$

¹³ The definition of negative association is given in Definition 22 in Appendix A. This property allows to apply concentration bounds (see Theorem 23 in Appendix A).

Moreover, for every $\mu \geq d\Delta(v) \cdot (\Delta_{\max}(\mathcal{S})/\Delta_{\min}(\mathcal{C})) \cdot \prod_{i=1}^{t-1} k_i$,

$$\Pr(r_t(N(v)) \geq 2\mu \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}) \leq e^{-\frac{\mu}{3}}.$$

Proof. By expressing $r_t(N(v))$ as the sum in (4), we can apply the first two claims in Lemma 8 and get

$$\mathbb{E}[r_t(N(v)) \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}] \leq d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})} \prod_{i=1}^{t-1} k_i.$$

In order to get the claimed bound in concentration, we need to apply the Chernoff bound to the sum of random variables of the form $a_t^{(i)}(w) \cdot z_t^{(i)}(w, u)$. To this aim, we know that for every $u \in \mathcal{S}$ and every $w \in N(u)$, $z_t^{(i)}(w, u)$ is a Bernoulli random variable of parameter $1/\Delta(w)$. However, the distributions of $a_t^{(i)}(w) \cdot z_t^{(i)}(w, u)$ are rather difficult to analyze since there are several correlations among the random variables in $\{a_t^{(i)}(w) : w \in \mathcal{C}, i \in [d]\}$. To cope with this issue, we exploit Claim 2 of Lemma 8 and construct nd ad-hoc independent Bernoulli random variables, $(X_t^{(i)}(w))_{w \in \mathcal{C}, i \in [d]}$ for which:

$$\Pr\left(X_t^{(i)}(w) = 1 \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) = \prod_{i=1}^{t-1} k_i \tag{8}$$

and such that each $X_t^{(i)}(w)$ stochastically dominates $a_t^{(i)}(w)$. Formally, thanks to (8) and Claim 2 of Lemma 8, we can define a coupling¹⁴ between $a_t^{(i)}(w)$ and $X_t^{(i)}(w)$ such that

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{a_t^{(i)}(w) \leq X_t^{(i)}(w)\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) = 1. \tag{9}$$

To define the coupling, we consider nd uniform and independent random variables in $[0, 1]$, $U_w^{(i)}$ with $i \in [d]$ and $w \in \mathcal{C}$. Given $t \geq 1$, $w \in \mathcal{C}$ and $i \in [d]$, we define the following set of random variables:

$$A_{t,i,w} = \{a_t^{(j)}(v) : j < i \text{ for } v = w \text{ and } j \in [d] \text{ for } v < w\}$$

which is nothing but the previous random variables of $a_t^{(i)}(w)$ according to the following sorting ($w = v_h$ for some h):

$$\begin{aligned} & a_t^{(1)}(v_1), a_t^{(2)}(v_1), \dots, a_t^{(d)}(v_1) \\ & a_t^{(1)}(v_2), a_t^{(2)}(v_2), \dots, a_t^{(d)}(v_2) \\ & \dots \\ & a_t^{(1)}(v_n), a_t^{(2)}(v_n), \dots, a_t^{(d)}(v_n). \end{aligned}$$

In the next definition, we will improperly use the term $A_{t,i,w}$ to denote the event in which the random variables $a_t^{(j)}(v)$ of subset $A_{t,i,w}$ take any fixed values in $\{0, 1\}$. For every $i \in [d]$ and $w \in \mathcal{C}$, given $\{K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\}$ we define the following two events

$$H_{t,i,w} = \{U_w^{(i)} \leq \prod_{j=0}^{t-1} k_j\}$$

$$K_{t,i,w} = \{U_w^{(i)} \leq \Pr(a_t^{(i)}(w) = 1 \mid A_{t,i,w}, K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1})\}.$$

Now we can define the coupling. For $h_{i,w}, k_{i,w} \in \{0, 1\}$

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{(X_t^{(i)}(w), a_t^{(i)}(w)) = (h_{i,w}, k_{i,w})\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) =$$

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{(\mathbb{1}_{H_{t,i,w}}, \mathbb{1}_{K_{t,i,w}}) = (h_{i,w}, k_{i,w})\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right).$$

Now we show that the coupling is well defined, i.e. the marginal laws are the same of $X_t^{(i)}(w)$ and $a_t^{(i)}(w)$. It's trivial that

¹⁴ See for instance Chapter 05 of [30].

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{X_t^{(i)}(w) = h_{i,w}\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) =$$

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{\not\llcorner_{H_{t,i,w}} = h_{i,w}\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right).$$

We have also that

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{\not\llcorner_{K_{t,i,w}} = k_{i,w}\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right)$$

$$= \prod_{i,w:k_{i,w}=1} \Pr(K_{t,i,w} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}) \cdot$$

$$\prod_{i,w:k_{i,w}=0} \Pr(K_{t,i,w}^C \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}) \tag{10}$$

$$= \prod_{i,w:k_{i,w}=1} \Pr(a_t^{(i)}(w) = 1 \mid A_{t,i,w}, K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}) \cdot$$

$$\prod_{i,w:k_{i,w}=0} \Pr(a_t^{(i)}(w) = 0 \mid A_{t,i,w}, K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1})$$

$$= \prod_{i,w} \Pr(a_t^{(i)}(w) = k_{i,w} \mid A_{t,i,w}, K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1})$$

$$= \Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{a_t^{(i)}(w) = k_{i,w}\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right). \tag{11}$$

(10) follows by the independence of the random variables $U_w^{(i)}$ with $i \in [d]$ and $w \in \mathcal{C}$. (11) follows by the chain rule with the same sorting adopted in the definition of $A_{t,i,w}$. It's easy to see that the coupling satisfies (9). Indeed

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{a_t^{(i)}(w) \leq X_t^{(i)}(w)\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) =$$

$$\Pr\left(\bigcap_{i \in [d], w \in \mathcal{C}} \{\not\llcorner_{K_{t,i,w}} \leq \not\llcorner_{H_{t,i,w}}\} \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) = 1.$$

Indeed, for every $i \in [d]$ and $w \in \mathcal{C}$,

$$\{K_{t,i,w}\} \subseteq \{H_{t,i,w}\},$$

since for every $w \in \mathcal{C}$

$$\Pr\left(a_t^{(i)}(w) = 1 \mid A_{t,i,w}, K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right) \leq \prod_{j=0}^{t-1} k_j,$$

and we can derive the last inequality from the fact that

$$\Pr\left(a_t^{(i)}(w) = 1 \mid A_{t,i,w}, B_1(w) = b_1(w), \dots, B_{t-1}(w) = b_{t-1}(w)\right) = \prod_{j=0}^{t-1} b_j(w).$$

By using the coupling, from (9), we get

$$\Pr(r_t(N(v)) \geq 2\mu \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1})$$

$$\leq \Pr\left(\sum_{i=1}^d \sum_{u \in N(v)} \sum_{w \in N(u)} X_t^{(i)}(w) \cdot z_t^{(i)}(w, u) \geq 2\mu \mid K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\right)$$

$$\leq e^{-\frac{\mu}{3}}, \tag{12}$$

where $\mu > 0$ is every positive real that satisfies $\mu \geq d\Delta(v)(\Delta_{\max}(\mathcal{S})/\Delta_{\min}(\mathcal{C})) \cdot \prod_{i=1}^{t-1} k_i$. In detail, (12) follows from (9) and, moreover, to get (12) we apply the Chernoff bound with $\varepsilon = 1$ for negatively associated random variables (see Theorem 23 in Appendix A). Indeed, Claim 3 of Lemma 8 and (8) imply that the random variables

$$\left(X_t^{(i)}(w) \cdot z_t^{(i)}(w, u) \right)_{i \in [d], u \in N(v), w \in N(u)},$$

conditioning on the event $\{K_1 \leq k_1, \dots, K_{t-1} \leq k_{t-1}\}$, are distributed as Bernoulli ones of parameter $\prod_{i=1}^{t-1} k_i / \Delta(w)$ and they are negatively associated (see Definition 22 in Appendix A). \square

Wrapping up: process analysis in two time stages. Lemmas 9 and 10 provide the decreasing rate of the number of alive balls in every fixed $N(v)$, conditioning on the events “ $K_j \leq k_j$ ”.

We now need to derive the specific sequence of k_j that effectively works for our process and that leads to Lemma 3. Moreover, we notice that (7) in Lemma 10 (only) allows a sufficiently strong concentration as long as the bound μ we can use on the expectation of $r_t(N(v))$ keeps of order $\Omega(\log n)$, while we clearly need to get an effective concentration bound until this value reaches 0.

To address the issues above, we split our analysis in two time stages. Roughly speaking, the first stage proceeds as long as the expectation of $r_t(N(v))$ is $\Omega(\log n)$ and we show it is characterized by an exponential decreasing of $r_t(N(v))$ (see Lemma 11 and Lemma 12). In the second stage, our technical goal is instead to show that the fraction of burnt nodes in $N(v)$ keeps bounded by some constant < 1 , while neglecting the decreasing rate of the balls received by $N(v)$ (since we cannot anymore get strong concentration bounds on this random variable). Essentially, our analysis shows that: i) the process starts this second stage when the expectation of $r_t(N(v))$ is $\Theta(\log n)$; ii) during a subsequent window of $\mathcal{O}(\log n)$ rounds, the fraction of burnt nodes in $N(v)$ keeps bounded by some constant < 1 and, hence, all the alive requests will be successfully assigned, w.h.p.

As for the first stage, we consider the sequence $\{\gamma_t\}_{t \in \mathbb{N}}$ defined by the following recurrence

$$\begin{cases} \gamma_t = \gamma_{t-1} + \frac{2}{c} \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i & \text{if } t \geq 2; \\ \gamma_1 = \frac{2}{c} \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)}. \end{cases} \quad (13)$$

In Appendix B, we will prove the following properties.

Lemma 11. For every $c \geq 1$, let $\{\gamma_t\}_{t \geq 0}$ be the sequence defined by the recurrence (13). Then, for every $c \geq 32\rho$, the following facts hold:

- $\{\gamma_t\}_{t \in \mathbb{N}}$ is increasing;
- for all $t \geq 1$, $\gamma_t \leq \frac{1}{4}$.

The next lemma provides some useful concentration bounds on the random variables K_t and $r_t(N(v))$ for the first stage.

Lemma 12 (Stage I: fast decreasing of the active balls). Under the hypothesis of Theorem 1, an integer $T = \mathcal{O}(\log(d\Delta_{\max}(S)/\log n))$ exists such that, for every $0 \leq t < T$,

$$\Pr \left(\bigcap_{v \in \mathcal{C}} \left\{ r_t(N(v)) \leq 2d\Delta(v) \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1} \right\} \right) \geq 1 - \frac{1}{n^3} \quad (14)$$

and

$$\Pr(K_t \leq \gamma_t \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1}) \geq 1 - \frac{1}{n^3}. \quad (15)$$

Proof. We consider γ_t as in (13) and apply Lemma 10 for every $v \in \mathcal{C}$, with

$$\mu = d\Delta(v) \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i.$$

We get, for every $v \in \mathcal{C}$,

$$\begin{aligned} & \Pr \left(r_t(N(v)) \geq 2d\Delta(v) \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1} \right) \\ & \leq e^{-\frac{1}{3}d\Delta(v) \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i} \leq e^{-\frac{1}{3}d\Delta_{\max}(S) \prod_{i=1}^{t-1} \gamma_i}, \end{aligned} \quad (16)$$

where the last inequality in (16) holds because, for every $v \in \mathcal{C}$, $\Delta(v)/\Delta_{\min}(C) \geq 1$. From (3), we know that $K_t(v) \leq K_{t-1} + \frac{1}{cd\Delta(v)}r_t(N(v))$, so, using a union bound over all clients v , we get

$$\begin{aligned}
 & \Pr(K_t \leq \gamma_t \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1}) \\
 & \geq \Pr\left(\bigcap_{v \in \mathcal{C}} \left\{ K_t(v) \leq \gamma_{t-1} + \frac{2}{c} \frac{\Delta_{\max}(C)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i \right\} \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1}\right) \\
 & \geq \Pr\left(\bigcap_{v \in \mathcal{C}} \left\{ r_t(N(v)) \leq 2d\Delta(v) \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i \right\} \mid K_1 \leq \gamma_1, \dots, K_{t-1} \leq \gamma_{t-1}\right) \\
 & \geq 1 - ne^{-\frac{1}{3}d\Delta_{\max}(S) \prod_{i=1}^{t-1} \gamma_i},
 \end{aligned} \tag{17}$$

where in the first inequality we also used the definition of γ_t given in (13). We now establish when (17) turns to be a “high probability”. Since $c \geq 32\rho$, Lemma 11 and the fact that $\Delta_{\max}(S) \geq \eta \log^2 n$ ensure that for a sufficiently large n we can take $T \geq 1$ as the smallest integer for which

$$d\Delta_{\max}(S) \prod_{i=1}^{T-1} \gamma_i \leq 12 \log n \tag{18}$$

and, hence,

$$d\Delta_{\max}(S) \prod_{i=1}^{t-1} \gamma_i > 12 \log n \quad \text{for every } t < T. \tag{19}$$

Indeed, from of Lemma 11, since $c \geq 32\rho$, then $\prod_{i=1}^{T-1} \gamma_i \leq (\gamma_{T-1})^{T-1} \leq (1/4)^{T-1}$ and so, from (18), we can say that T verifies

$$T \leq \frac{1}{2} \log \frac{d\Delta_{\max}(S)}{12 \log n} + 1.$$

Finally, by using (19) in (17), we get (14) and (15) for every $t < T$. \square

Lemma 13 (Stage II: the fraction of burnt servers keeps small). *Under the hypothesis of Theorem 1, there exists $T \geq 1$ (it can be the same stated in the previous lemma) such that, for every t in the range $[T, \dots, 3 \log n]$,*

$$\Pr\left(\bigcap_{v \in \mathcal{C}} \left\{ r_t(N(v)) \leq \frac{24\Delta(v) \log n}{\Delta_{\min}(C)} \right\} \mid E_{T,t}\right) \geq 1 - \frac{1}{n^3}$$

and

$$\Pr(K_t \leq \delta_t \mid E_{T,t}) \geq 1 - \frac{1}{n^3},$$

where

$$E_{T,t} = \{K_1 \leq \gamma_1, \dots, K_{T-1} \leq \gamma_{T-1}, K_T \leq \delta_T, \dots, K_{t-1} \leq \delta_{t-1}\}$$

and γ_t is defined in (13) and δ_t is defined by the recurrence

$$\begin{cases} \delta_t = \delta_{t-1} + \frac{24 \log n}{cd\Delta_{\min}(C)} & \text{if } t \geq T; \\ \delta_{T-1} = \gamma_{T-1}. \end{cases} \tag{20}$$

Proof. At first, we analyze some properties of the sequence δ_t in (20). We notice that the sequence is increasing and that, for every $t \in [T, \dots, 3 \log n]$ and $c \geq 288/(\eta d)$,

$$\delta_t = \delta_{t-1} + \frac{24 \log n}{cd\Delta_{\min}(C)} = \gamma_{T-1} + \frac{24 \log n(t-T)}{cd\Delta_{\min}(C)} \leq \frac{1}{4} + \frac{72 \log^2 n}{cd \cdot \eta \log^2 n} \leq \frac{1}{2}.$$

As in the proof of the previous lemma, we take T as the first integer such that

$$d\Delta_{\max}(S) \prod_{i=1}^{T-1} \gamma_i \leq 12 \log n. \tag{21}$$

So, for every t such that $T \leq t \leq 3 \log n$, (21) and Lemma 10 imply that

$$\begin{aligned} \mathbf{E} [r_t(N(v)) \mid E_{T,t}] &\leq d\Delta(v) \frac{\Delta_{\max}(\mathcal{S})}{\Delta_{\min}(\mathcal{C})} \prod_{i=1}^{T-1} \gamma_i \prod_{j=T}^t \delta_j \\ &\leq d\Delta_{\max}(\mathcal{S}) \prod_{i=1}^{T-1} \gamma_i \cdot \frac{\Delta(v)}{\Delta_{\min}(\mathcal{C})} \leq 12 \log n \cdot \frac{\Delta(v)}{\Delta_{\min}(\mathcal{C})}. \end{aligned}$$

Taking $\mu = 12 \log n \cdot \frac{\Delta(v)}{\Delta_{\min}(\mathcal{C})}$, for Lemma 10 we get

$$\Pr \left(r_t(N(v)) \geq \frac{24\Delta(v) \log n}{\Delta_{\min}(\mathcal{C})} \mid E_{T,t} \right) \geq 1 - e^{-4 \frac{\Delta(v) \log n}{\Delta_{\min}(\mathcal{C})}} \geq 1 - \frac{1}{n^4}.$$

By a union bound over all the clients $v \in \mathcal{C}$, from (3) and from the definition of δ_t in (20),

$$\begin{aligned} \Pr (K_t \leq \delta_t \mid E_{T,t}) &\geq \Pr \left(\bigcap_{v \in \mathcal{C}} \left\{ K_t(v) \leq \delta_{t-1} + \frac{24 \log n}{cd \cdot \Delta_{\min}(\mathcal{C})} \right\} \mid E_{T,t} \right) \\ &\geq \Pr \left(\bigcap_{v \in \mathcal{C}} \{ r_t(N(v)) \leq \frac{24\Delta(v)}{\Delta_{\min}(\mathcal{C})} \log n \} \mid E_{T,t} \right) \geq 1 - \frac{1}{n^3}. \quad \square \end{aligned}$$

Remark. The analysis given in the above proof is the key point in which the density hypothesis $\Delta_{\min}(\mathcal{C}) = \Omega(\log^2 n)$ is required. More in detail, we use this hypothesis to keep δ_t constant. In fact, we would need the weaker condition $\log n / \Delta_{\min}(\mathcal{C}) = o(1)$, however, we consider here a stronger hypothesis to make the argument simpler.

Lemma 12 and 13 imply Lemma 3. Indeed, for the chain rule, taking $T' = \lfloor 3 \log n \rfloor$ and $c \geq \max(32\rho, 288/(\eta d))$ we get

$$\Pr \left(\bigcap_{t=1}^{T'-1} \{K_t \leq \gamma_t\} \bigcap \bigcap_{t=T'}^{T'} \{K_t \leq \delta_t\} \right) = \left(1 - \frac{1}{n^3} \right)^{T'} \geq 1 - T' \frac{1}{n^3} \geq 1 - \frac{1}{n^2}, \quad (22)$$

where in the first inequality of (22) we used the chain rule, Lemma 12 and 13, while the second last inequality of (22) follows from the binomial inequality, i.e., for every $x \geq -1$ and for every $m \in \mathbb{N}$, $(1+x)^m \geq 1+mx$. Concluding, we have shown that $K_t \leq \gamma_t$ for every $t \leq T$, and $K_t \leq \delta_t$ for all t such that $T \leq t \leq 3 \log n$, with probability at least $1 - 1/n^2$. Since $c \geq \max\{32\rho, 288/(\eta d)\}$, we have that $\delta_t \leq 1/2$ and $\gamma_t \leq 1/2$ and, whereas $B_t \leq K_t$, we have that, with probability at least $1 - 1/n^2$, $B_t \leq 1/2$ for every t such that $t \leq 3 \log n$.

4.2. On the performance of SAER

To complete the proof of Theorem 1, we show the following consequence of Lemma 3.

Corollary 14. *Under the assumptions of Theorem 1, the protocol SAER(c, d) has completion time $\mathcal{O}(\log n)$ and its work is $\Theta(n)$, w.h.p.*

Proof. Consider every fixed ball of a client $v \in \mathcal{C}$. By choosing¹⁵ the parameter c as indicated by Lemma 3, (1) implies that the probability the ball is not accepted for all rounds $t \leq 3 \log n$, conditioning on the bound given in Lemma 3, is $(1/2)^{3 \log n} = (1/n)^3$. Then, by applying a union bound for all balls and all clients, and considering the probability of the conditioning event, we get that SAER(c, d) completes in $3 \log n$ rounds, with probability at least $1 - O(1/n^2)$.

To analyze the overall work performed by SAER we proceed using an approach similar to that in the analysis of the Becchetti et al.'s algorithm RAES. For every $v \in \mathcal{C}$ and every ball $i \in [d]$, recall the random variable $a_t^{(i)}(v)$ introduced in Definition 6. Then, the random variable counting the total number of requests performed by the clients (plus the relative answers by the servers) to assign the nd balls can be easily bounded by

$$W = 2 \cdot \sum_{t=1}^{\infty} \sum_{i=1}^d \sum_{v \in \mathcal{C}} a_t^{(i)}(v).$$

To prove that $W = \mathcal{O}(dn)$ w.h.p., we show that, for every fixed $t \leq 3 \log n$ and every $k \geq nd / \log n$, it holds

$$\Pr \left(\sum_{i=1}^d \sum_{v \in \mathcal{C}} a_t^{(i)}(v) > \frac{4}{5} k \mid \sum_{i=1}^d \sum_{v \in \mathcal{C}} a_{t-1}^{(i)}(v) = k \right) \leq e^{-\frac{k}{25cd}}. \quad (23)$$

¹⁵ Since $d \geq 1$, the suitable value for c can be fixed by the servers by looking only at η . We also remark our analysis does not optimize several aspects such as the bound on c and its relation with η .

To this aim, we use the *method of bounded differences* (see Theorem 24 in Appendix A). We notice that the random variable $\sum_{i=1}^d \sum_{v \in \mathcal{C}} a_t^{(i)}(v)$, conditioning on a number k of alive balls at the end of round $t-1$, can be written as a function of k independent random variables that satisfies the Lipschitz condition with coefficient $2cd$ (see definition in Theorem 24). Indeed, we define the random variables $w^{(t-1)}$ as the set of alive balls at the end of round t and the random variables $\{Y_i\}_{i \in w^{(t-1)}}$, taking values in $\mathcal{S} = [n]$, indicating the server-destination in \mathcal{S} the alive ball tries to connect to at round t . The random variables Y_i with $i \in w^{(t-1)}$ are mutually independent, and we can write, given the number k of alive balls at round $t-1$,

$$\sum_{i=1}^d \sum_{v \in \mathcal{C}} a_t^{(i)}(v) = f(Y_{i_1}, \dots, Y_{i_k}).$$

The function f satisfies the Lipschitz condition with coefficient $2cd$ because, if we change one of the values Y_i , we are changing the destination of a ball from some $u_1 \in \mathcal{S}$ to some $u_2 \in \mathcal{S}$. If u_2 has received fewer than cd requests since the start of the process, the change of the destination of the i -th ball from u_1 to u_2 would not have any impact. On the other hand, in the worst case, at most cd balls that try to settle in u_2 switch from settled to not settled. A symmetric argument holds for u_1 and so if

$$\mathbf{Y} = (v_{i_1}, \dots, v_{i_j}, \dots, v_{i_k}) \text{ and } \mathbf{Y}' = (v_{i_1}, \dots, v'_{i_j}, \dots, v_{i_k})$$

then

$$|f(\mathbf{Y}) - f(\mathbf{Y}')| \leq 2cd.$$

Lemma 3 implies that at every round $t \leq 3 \log n$ the fraction of burnt nodes in any node's neighborhood remains bounded by $1/2$ with probability at least $1 - 1/n^2$. Therefore, for every $t \leq 3 \log n$ holds

$$\mathbb{E} \left[\sum_{i=1}^d \sum_{v \in \mathcal{C}} a_t^{(i)}(v) \mid \sum_{i=1}^d \sum_{v \in \mathcal{C}} a_{t-1}^{(i)}(v) = k \right] \leq \frac{k}{2} + \frac{1}{n^2}$$

and we can apply Theorem 24 with $\mu = 3k/5$ (since $k \geq nd/\log n$) and $M = k/5$, obtaining (23).

From (23) and the chain rule, it follows that for $T = \Omega(\log \log n)$ rounds the number of alive balls decreases at each round by a factor $4/5$, w.h.p. Hence, at the end of the T -th round, the number of alive balls is smaller than $O(nd/\log n)$, w.h.p. From Theorem 1, we know that the remaining $nd/\log n$ alive balls are assigned within $\mathcal{O}(\log n)$ round: this implies an additional work of $\mathcal{O}(nd)$. Observe that the work until round T is $nd \sum_{t=1}^T (4/5)^t = \mathcal{O}(nd)$. Hence, for any integer $d \geq 0$, we get the claimed linear bound for the work complexity of SAER(c, d). \square

5. Proof of Theorem 1 for RAES

Our analysis of the RAES process proving Theorem 1 proceeds as follows. We first introduce a suitable coupling between the SAER process and the RAES one. We then show that this coupling implies an upper bound on the fraction of saturated nodes in the RAES process which is similar to that on the number of burnt servers for the SAER process we proved in Lemma 3. Then, we apply (in a way similar to what we did for SAER) the obtained bound to derive the performance bounds on the RAES process.

To analyze the RAES process we first need to introduce the following random variables.

Definition 15. Let us consider the RAES process. For every $u \in \mathcal{S}$ and every $t \geq 1$, let $\tilde{r}_t(u)$ be the number of balls the server u receives at round t , and let $\text{acc}_t(u)$ be the total number of balls the server u accepts within round t .

In the RAES process, like the SAER process, the server never accepts more than cd balls, but in a different and more natural way: a server $u \in \mathcal{S}$ rejects the received balls at round t if it gets *saturated* at round t .

Definition 16. For every integer $t \geq 1$, we say that a server $u \in \mathcal{S}$ is *SATURATED* at round t of the process RAES if

$$\text{acc}_t(u) + \tilde{r}_t(u) > cd.$$

Moreover, for every client $v \in \mathcal{C}$, let $S_t(v)$ be the fraction of *saturated* servers in the neighborhood of v at round t , i.e.,

$$S_t(v) = \frac{\{u \in \mathcal{S} : u \in N(v) \text{ and } \text{acc}_t(u) + \tilde{r}_t(u) > cd\}}{\Delta(v)}.$$

We also define S_t as the maximum fraction of saturated nodes in any client's neighborhood at round t , i.e. $S_t = \max_{v \in \mathcal{C}} S_t(v)$.

To establish an upper bound on the fraction of saturated servers in the RAES process, we introduce a suitable coupling between the SAER process and the RAES one.

The coupling between SAER and RAES. Let us consider the RAES and SAER processes on the same client-server graph $G((\mathcal{C}, \mathcal{S}), E)$ and with the same initial client's requests. Then, informally speaking, our next aim is to show that the maximum fraction of burnt servers B_t , according to the RAES process. To make this argument rigorous, we use the notion of stochastic domination, denoted as $S_t \preceq B_t$, that we recall in Definition 25 in Appendix A. This will allow us to prove a result equivalent to that in Lemma 3 for the RAES process. To show that $S_t \preceq B_t$, we construct a coupling $(B_{t,C}, S_{t,C})$ between the two random variables such that $\Pr(B_{t,C} \leq S_{t,C}) = 1$. The simple coupling forces, at any round t , any non-settled ball i of a client $v \in \mathcal{C}$ to choose the same server destination in the two processes. Essentially, according to this coupling, we easily get that the number of alive balls in the RAES process turns out to be never larger than that in the SAER process.

Definition 17. For every $t \geq 1$, $v \in \mathcal{C}$ and $i \in [d]$, let $Y_t^{(i)}(v)$ be the random variables taking values in \mathcal{S} indicating the contacted server for the v 's i -th ball at round t in the SAER process. Let $\tilde{Y}_t^{(i)}(v)$ the equivalent random variable in the RAES process.

We observe that the random variables $\{Y_t^{(i)}(v), v \in \mathcal{C}, i \in [d], t \geq 1\}$ are independent, since each server's ball chooses the server destination independently from the other balls. We observe also that, from Definition 7, we have that $Y_t^{(i)}(v) = u$ if and only if $z_t^{(i)}(v, u) = 1$.

We remark that both processes at each round t are completely described by the random variables $\{Y_t^{(i)}(v), i \in [d], v \in \mathcal{C}, t \geq 1\}$. Indeed, if we know the destination of each server's ball at each round, we can derive which is the set of saturated/burnt nodes, which balls have been rejected and which requests are still active at each round t . So, we can define the coupling between the two processes thanks to such random variables.

To define the coupling, we introduce the following random variables.

Definition 18. For every $t \geq 1$, client $v \in \mathcal{C}$ and ball $i \in [d]$, let $W_t^{(i)}(v)$ be the random variable indicating the destination of the i -th's ball of v . Since it is chosen uniformly at random in $N(v)$, it holds that

$$\begin{cases} \Pr(W_t^{(i)}(v) = u) = \frac{1}{\Delta(v)} & \text{if } u \in N(v); \\ \Pr(W_t^{(i)}(v) = u) = 0 & \text{if } u \notin N(v). \end{cases}$$

Definition 19 (Coupling between SAER and RAES). For every $t \geq 1$, every client $v \in \mathcal{C}$, and every ball $i \in [d]$, we define the coupling $(X_{t,C}^{(i)}(v), \tilde{X}_{t,C}^{(i)}(v))$ between $X_t^{(i)}(v)$ and $\tilde{X}_t^{(i)}(v)$ such that

$$\Pr(Y_{t,C}^{(i)}(v), \tilde{Y}_{t,C}^{(i)}(v) = (W_t^{(i)}(v), W_t^{(i)}(v))) = 1.$$

So, in the coupling, each ball $i \in [d]$ of a client $v \in \mathcal{C}$ at each round $t \geq 1$ chooses the same server destination in both processes RAES and SAER.

In what follows we will make use of the subscript C to indicate random variables that are defined over the joint probability space the coupling is defined over.

On the fraction of saturated servers. We now use the above coupling to show the analogousness of Lemma 3 for RAES.

Lemma 20 (On the fraction of saturated servers). Under the hypothesis of Theorem 1, for any $c \geq \max(32\rho, 288/(\eta d))$, with probability at least $1 - 1/n^2$, it holds that, for every $t \geq 3 \log n$, the fraction of saturated nodes in RAES(c, d) satisfies

$$S_t \leq \frac{1}{2}.$$

Proof. To prove the lemma, we use the coupling in Definition 19, to show that, for every round $t \geq 1$, the random variable S_t is stochastically dominated by B_t , i.e.,

$$S_t \preceq B_t. \tag{24}$$

Then, thanks to Lemma 3, we can derive that, for every $t \geq 1$, it holds $\Pr(S_t \leq \frac{1}{2}) \geq 1 - 1/n^2$.

We next prove that the considered coupling satisfies (24): in more detail, thanks to Lemma 26 in Appendix A, it suffices to prove that, according to the coupling, for every $t \geq 1$, it holds

$$\Pr(S_{t,C} \leq B_{t,C}) = 1. \quad (25)$$

We first prove, by induction on t , that, at each round $t \geq 1$, if a fixed server $u \in \mathcal{S}$ is saturated in RAES, then it is also burnt in SAER.

As for the first round, we notice that every ball of the clients is alive, and, moreover, a node is saturated iff it is burnt since, at the first round, for each server u , it holds $\text{acc}_1(u) = 0$ (see Definition 16). Then, by the definition of the coupling, we can state that the two processes are identical, and get the claim for $t = 1$.

As for $t > 1$, we can assume that, for every round $j < t$, if a server is saturated in RAES at round j , it is also burnt in SAER at round j and we want to prove the same holds at round t . Notice that, by definition of the coupling, each ball chooses the same destination in both processes. Thus, by induction hypothesis, we have that each ball which is alive in the RAES process at round $j \leq t$ is also alive in the SAER process at round $j \leq t$, i.e. for every round $j \leq t$, for every client $v \in \mathcal{C}$, and for every ball $i \in [d]$,

$$\tilde{a}_{j,C}^{(i)}(v) \leq a_{j,C}^{(i)}(v).$$

The above equation implies that, for every round $j \leq t$ and for every server $u \in \mathcal{S}$, $\tilde{r}_{j,C}(u) \leq r_{j,C}(u)$. Indeed we have

$$\tilde{r}_{j,C}(u) = \sum_{v \in N(u)} \sum_{i=1}^d \tilde{a}_{j,C}^{(i)}(v) \cdot \tilde{z}_{j,C}^{(i)}(v, u) \leq \sum_{v \in N(u)} \sum_{i=1}^d a_{j,C}^{(i)}(v) \cdot z_{j,C}^{(i)}(v, u) = r_{j,C}(u), \quad (26)$$

and, moreover, from the coupling, at every round $j \leq t$, for every $u \in \mathcal{C}$ and $v \in N(v)$, $\tilde{z}_{j,C}^{(i)}(u, v) = z_{j,C}^{(i)}(u, v)$.

Now, we get that if a server $u \in \mathcal{S}$ is saturated in RAES at round t , then it is also burnt in SAER at round t . Indeed, if u is saturated at round t , it holds

$$\text{acc}_{t,C}(u) + \tilde{r}_{t,C}(u) > cd, \quad (27)$$

and, since $\text{acc}_{t,C}(u) \leq \sum_{i=1}^{t-1} \tilde{r}_{i,C}(u)$, from (26) and (27),

$$\sum_{i=1}^t r_{i,C}(u) \geq \sum_{i=1}^t \tilde{r}_{i,C}(u) \geq \text{acc}_{t,C}(u) + \tilde{r}_{t,C}(u) > cd.$$

It thus follows that the server u is also burnt in the SAER process.

This fact implies that for all $v \in \mathcal{C}$ and each $t \geq 1$, the fraction of saturated server in the neighborhood of v in RAES is dominated by the fraction of burnt server in RAES, i.e.

$$\Pr\left(\bigcap_{v \in \mathcal{C}} \{S_{t,C}(v) \leq B_{t,C}(v)\}\right) = 1$$

and so (25) holds. \square

Performance analysis of RAES. Theorem 1 is a consequence of Lemma 20: its formal proof proceeds in the same way to that for the SAER process in Subsection 4.2 and, thus, it is omitted.

Corollary 21. *Under the assumptions of Theorem 1, the protocol RAES(c, d) has completion time $\mathcal{O}(\log n)$ and its work is $\Theta(n)$, w.h.p.*

6. Conclusions and future work

We studied two simple parallel Load Balancing protocol and we give a probabilistic analysis of their performances. The main novelty of this paper lies in considering client-server bipartite graphs that are much more sparse than those considered in previous work. This new setting can model important network scenarios where proximity and/or trust issues force very restricted sets of admissible client-server assignments. From a technical point of view, such sparse topologies yield new probabilistic issues that make our analysis more challenging than that for the dense case and rather different from the previous ones.

Several interesting open questions are left open by our paper. In particular, we are intrigued by the analysis of the protocols (or simple variants of them) over graphs with $o(\log^2 n)$ degree and/or in the presence of a dynamic framework where, for instance, the client requests arrive on line and some random topology change may happen during the protocol execution. We discussed the former question in the remark after Lemma 13. As for the latter, we believe that the simple structure of the two considered protocols can well manage such a dynamic scenario and achieves a metastable regime with good performances.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Mathematical tools

Definition 22 (Negative association, [31]). The random variables X_i , $i \in [n]$ are *negatively associated* if for all disjoint subsets $I, J \subseteq [n]$ and all nondecreasing functions f and g ,

$$\mathbf{E}[f(X_i, i \in I)g(X_j, j \in J)] \leq \mathbf{E}[f(X_i, i \in I)]\mathbf{E}[g(X_j, j \in J)].$$

Theorem 23 (Chernoff for negatively associated random variables, [31]). Let X_1, \dots, X_n a family of random variables in $\{0, 1\}$ negatively associated and $X = X_1 + \dots + X_n$. Let $p_i = \mathbf{E}[X_i]$ and define $\mu = \mathbf{E}[X] = p_1 + \dots + p_n$. Then, for any reals $\varepsilon \in (0, 1)$

$$\Pr(X \geq (1 + \varepsilon)\mu) \leq e^{-\frac{\varepsilon^2}{3}\mu}.$$

Theorem 24 (Method of bounded differences, [31]). Let $\mathbf{Y} = (Y_1, \dots, Y_m)$ be independent random variables, with Y_j taking values in the set A_j . Suppose the real-valued function f defined on $\prod_j A_j$ satisfies the Lipschitz condition with coefficients β_j , i.e.

$$|f(\mathbf{y}) - f(\mathbf{y}')| \leq \beta_j$$

whenever vectors \mathbf{y}, \mathbf{y}' differs only in the j -th coordinate. Let μ an upper bound to the expected value of r.v. $f(\mathbf{Y})$. Then, for any $M > 0$, it holds that

$$\Pr(f(\mathbf{Y}) - \mu \geq M) \leq e^{-\frac{2M^2}{\sum_{j=1}^m \beta_j}}.$$

Definition 25 (Stochastic domination). A random variable Y stochastically dominates a random variable X ($X \preceq Y$) if the inequality

$$\Pr(X > x) \leq \Pr(Y > x)$$

holds for all $x \in \mathbb{R}$.

Lemma 26 ([32], Theorem 3.1). A random variable Y stochastically dominates a random variable X if and only if there exists a coupling (X_C, Y_C) of X and Y such that $\Pr(X_C \leq Y_C) = 1$.

Appendix B. Proof of Lemma 11

Since, from (13), for every $t > 1$

$$\gamma_t = \gamma_{t-1} + \frac{2}{c} \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \prod_{i=1}^{t-1} \gamma_i, \quad (\text{B.1})$$

the sequence is increasing. Now we want to prove, by induction, that each term of the sequence verifies $\gamma_t \leq \frac{1}{4} - \frac{1}{4^{t+1}}$ for every $t \geq 1$. From that, the lemma clearly follows. We notice that the hypothesis holds for γ_1 , since $\Delta_{\max}(C)/\Delta_{\min}(C) \leq \rho$ and $c \geq 32\rho$

$$\gamma_1 = \frac{2}{c} \frac{\Delta_{\max}(S)}{\Delta_{\min}(C)} \leq \frac{2\rho}{c} \leq \frac{1}{16} \leq \frac{1}{4} - \frac{1}{4^2}.$$

Now, assuming that

$$\gamma_{t-1} \leq \frac{1}{4} - \frac{1}{4^t} \quad (\text{B.2})$$

we will show that $\gamma_t \leq \frac{1}{4} - \frac{1}{4^{t+1}}$. From (B.1) and from (B.2) and since $\Delta_{\max}(S)/\Delta_{\min}(C) \leq \rho$ and $c \geq 32\rho$ we get that

$$\gamma_t - \gamma_{t-1} \leq \frac{2\rho}{c} \prod_{i=1}^{t-1} \gamma_i \leq \frac{1}{16} \cdot \frac{1}{4^{t-1}} \leq \frac{1}{4^{t+1}}. \quad (\text{B.3})$$

Then, we have that from (B.2) and (B.3)

$$\gamma_t \leq \gamma_{t-1} + \frac{1}{4^{t+1}} \leq \frac{1}{4} - \frac{1}{4^t} + \frac{1}{4^{t+1}} \leq \frac{1}{4} - \frac{1}{4^{t+1}}.$$

References

- [1] L. Becchetti, A.E.F. Clementi, E. Natale, F. Pasquale, L. Trevisan, Finding a bounded-degree expander inside a dense one, in: *Proceedings of SODA 2020: the 31st ACM-SIAM Symposium on Discrete Algorithms*, SODA 2020, Salt Lake City, UT, USA, January 5–8, 2020, SIAM, 2020, pp. 1320–1336.
- [2] M. Adler, S. Chakrabarti, M. Mitzenmacher, L.E. Rasmussen, Parallel randomized load balancing, *Random Struct. Algorithms* 13 (2) (1998) 159–188.
- [3] J. Aspnes, Y. Azar, A. Fiat, S.A. Plotkin, O. Waarts, On-line routing of virtual circuits with applications to load balancing and machine scheduling, *J. ACM* 44 (3) (1997) 486–504, <https://doi.org/10.1145/258128.258201>.
- [4] B. Awerbuch, M.T. Hajiaghayi, R.D. Kleinberg, T. Leighton, Online client-server load balancing without global information, in: *Proceedings of SODA 2005: the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2005, pp. 197–206, <http://dl.acm.org/citation.cfm?id=1070432.1070461>.
- [5] B. Bosek, D. Leniowski, P. Sankowski, A. Zych, Online bipartite matching in offline time, in: *Proceedings of FOCS 2014: the 55th IEEE Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, 2014, pp. 384–393.
- [6] H. Räcke, Minimizing congestion in general networks, in: *Proceedings of FOCS 2002: the 43rd Symposium on Foundations of Computer Science*, IEEE Computer Society, 2002, pp. 43–52.
- [7] C. Lenzen, M. Parter, E. Yogev, Parallel balanced allocations: the heavily loaded case, in: *Proceedings of SPAA 2019: the 31st ACM on Symposium on Parallelism in Algorithms and Architectures*, ACM, 2019, pp. 313–322.
- [8] P. Berenbrink, A. Brinkmann, T. Friedetzky, L. Nagel, Balls into non-uniform bins, *J. Parallel Distrib. Comput.* 74 (2) (2014) 2065–2076, <https://doi.org/10.1016/j.jpdc.2013.10.008>.
- [9] B. Godfrey, Balls and bins with structure: balanced allocations on hypergraphs, in: *Proceedings SODA 2008: the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, 2008, pp. 511–517, <http://dl.acm.org/citation.cfm?id=1347082.1347138>.
- [10] K. Kenthapadi, R. Panigrahy, Balanced allocation on graphs, in: *Proceedings of SODA 2006: the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM Press, 2006, pp. 434–443, <http://dl.acm.org/citation.cfm?id=1109557.1109606>.
- [11] Y. Azar, A.Z. Broder, A.R. Karlin, E. Upfal, Balanced allocations (extended abstract), in: *Proceedings of STOC 1994: the 26th Annual ACM Symposium on Theory of Computing*, ACM, 1994, pp. 593–602.
- [12] M. Dietzfelbinger, F.M. auf der Heide, Simple, efficient shared memory simulations, in: *Proceedings of SPAA 1993: the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, ACM, 1993, pp. 110–119.
- [13] M. Mitzenmacher, E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd edition, Cambridge University Press, 2017.
- [14] R.M. Karp, M. Luby, F.M. auf der Heide, Efficient PRAM simulation on a distributed memory machine, *Algorithmica* 16 (4/5) (1996) 517–542, <https://doi.org/10.1007/BF01940878>.
- [15] P. Berenbrink, A. Czumaj, A. Steger, B. Vöcking, Balanced allocations: the heavily loaded case, *SIAM J. Comput.* 35 (6) (2006) 1350–1385, <https://doi.org/10.1137/S009753970444435X>.
- [16] J.W. Byers, J. Considine, M. Mitzenmacher, Geometric generalizations of the power of two choices, in: *Proceedings of SPAA 2004: the 16th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, ACM, 2004, pp. 54–63.
- [17] B. Vöcking, How asymmetry helps load balancing, *J. ACM* 50 (4) (2003) 568–589, <https://doi.org/10.1145/792538.792546>.
- [18] U. Wieder, Balanced allocations with heterogeneous bins, in: *Proceedings of SPAA 2007: the 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, ACM, 2007, pp. 188–193.
- [19] P. Berenbrink, T. Friedetzky, Z. Hu, R.A. Martin, On weighted balls-into-bins games, *Theor. Comput. Sci.* 409 (3) (2008) 511–520, <https://doi.org/10.1016/j.tcs.2008.09.023>.
- [20] P. Berenbrink, F.M. auf der Heide, K. Schröder, Allocating weighted jobs in parallel, *Theory Comput. Syst.* 32 (3) (1999) 281–300, <https://doi.org/10.1007/s002240000119>.
- [21] E. Koutsoupias, M. Mavronicolas, P.G. Spirakis, Approximate equilibria and ball fusion, *Theory Comput. Syst.* 36 (6) (2003) 683–693, <https://doi.org/10.1007/s00224-003-1131-5>.
- [22] P. Berenbrink, T. Friedetzky, L.A. Goldberg, P.W. Goldberg, Z. Hu, R.A. Martin, Distributed selfish load balancing, *SIAM J. Comput.* 37 (4) (2007) 1163–1181, <https://doi.org/10.1137/060660345>.
- [23] R. Kleinberg, G. Piliouras, É. Tardos, Load balancing without regret in the bulletin board model, *Distrib. Comput.* 24 (1) (2011) 21–29, <https://doi.org/10.1007/s00446-011-0129-5>.
- [24] P. Berenbrink, A. Brinkmann, T. Friedetzky, L. Nagel, Balls into bins with related random choices, *J. Parallel Distrib. Comput.* 72 (2) (2012) 246–253, <https://doi.org/10.1016/j.jpdc.2011.10.006>.
- [25] S.D. Gantz, D.R. Philpott, *FISMA and the Risk Management Framework: The New Practice of Federal Cyber Security*, 1st edition, Syngress Publishing, 2012.
- [26] C. Zhang, J. Sun, X. Zhu, Y. Fang, Privacy and security for online social networks: challenges and opportunities, *IEEE Netw.* 24 (4) (2010) 13–18, <https://doi.org/10.1109/MNET.2010.5510913>.
- [27] P. Berenbrink, T. Friedetzky, C. Lammersen, T. Sauwervald, Parallel randomized load balancing, unpublished manuscript.
- [28] P. Berenbrink, K. Khodamoradi, T. Sauerwald, A. Stauffer, Balls-into-bins with nearly optimal load distribution, in: *Proceedings of SPAA 2013: 25th ACM Symposium on Parallelism in Algorithms and Architectures*, ACM, 2013, pp. 326–335.
- [29] C. Lenzen, R. Wattenhofer, Tight bounds for parallel randomized load balancing: extended abstract, in: *Proceedings of STOC 2011: the 43rd ACM Symposium on Theory of Computing*, ACM, 2011, pp. 11–20.
- [30] D.A. Levin, Y. Peres, E.L. Wilmer, *Markov Chains and Mixing Times*, American Mathematical Society, 2006.
- [31] D.P. Dubhashi, A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms*, Cambridge University Press, 2009, <http://www.cambridge.org/gb/knowledge/isbn/item2327542/>.
- [32] H. Thorisson, *Coupling, Stationarity, and Regeneration*, Springer, New York, 2000.