2017 IEEE International Parallel and Distributed Processing Symposium

Tight Load Balancing via Randomized Local Search

Petra Berenbrink
Simon Fraser University
Burnaby, Canada
petra@sfu.ca

Peter Kling
Simon Fraser University
Burnaby, Canada
pkling@sfu.ca

Christopher Liaw
University of British Columbia
Vancouver, Canada
cvliaw@cs.ubc.ca

Abbas Mehrabian
University of British Columbia
& Simon Fraser University
Greater Vancouver, Canada
abbasmehrabian@gmail.com

***Abstract*—We consider the following balls-into-bins process with n bins and m balls: Each ball is equipped with a mutually independent exponential clock of rate 1. Whenever a ball's clock rings, the ball samples a random bin and moves there if the number of balls in the sampled bin is smaller than in its current bin.**

This simple process models a typical load balancing problem where users (balls) seek a selfish improvement of their assignment to resources (bins). From a game theoretic perspective, this is a randomized approach to the well-known *KP-model* [1], while it is known as Randomized Local Search (RLS) in load balancing literature [2], [3]. Up to now, the best bound on the expected time to reach perfect balance was $O\big((\ln n)^2 + \ln(n) \cdot n^2/m\big)$ due to [3]. We improve this to an asymptotically tight $O\big(\ln(n) + n^2/m\big)$. Our analysis is based on the crucial observation that performing *destructive moves* (reversals of RLS moves) cannot decrease the balancing time. This allows us to simplify problem instances and to ignore "inconvenient moves" in the analysis.

***Keywords*-Balls-into-bins; load balancing; randomized local search; coupling**

I. Introduction

We consider a system of n identical resources and m identical users. Each user must be assigned to exactly one resource. A user on resource i experiences a load equal to the number of users on resource i. Users can migrate between resources, and the goal is to find a fast and simple migration strategy that reaches a *perfectly balanced* state in which the load experienced by the users differs by at most 1.

Such load balancing (or reallocation) problems are well-studied and have a multitude of applications, ranging from scheduling in peer-to-peer systems [4] and channel allocation in wireless networks [5] to numerical applications such as computation of dynamics [6]. In the past decade, scalability and robustness concerns caused a shift from relatively complex centralized protocols to simple, often randomized distributed protocols. Indeed, consider applications such as multicore computers, routing in and between data centres, or load distribution in peer-to-peer networks. Due to the sheer size of these systems, maintainability issues, and increasing user demand, we have to push for distributed protocols that are easy to implement and that do not rely on global knowledge or user coordination.

A. Protocol and Results in a Nutshell

We analyze the following natural and simple load balancing process: Each of the m users is activated by an independent exponential clock of rate 1. Upon activation, a user chooses one of the n resources uniformly at random and compares his currently experienced load with the load it would experience at the new resource. He migrates to the new resource if and only if doing so does not result in a worse load. From a game theoretic perspective, this is a simple randomized approach to the well-known KP-model with unit weights and capacities [1]. In load balancing, this strategy is known as Randomized Local Search (RLS) [2], [3][1].

Our main result (Theorem 1) is that RLS reaches perfect balance in expected time $O\big(\ln(n) + n^2/m\big)$, and with high probability in time $O\big(\ln(n) + \ln(n) \cdot n^2/m\big)$. These bounds are asymptotically tight and improve the previously best bounds [3] by a logarithmic factor. At the heart of our improvement lies the simple but tremendously useful Destructive Majorization Lemma (DML, see Section IV). The DML formalizes the intuition that performing *destructive moves* (the reverse of a move permitted by RLS) cannot result in a speedup. This allows us to reverse unwanted protocol moves during the analysis. Moreover, in any analysis phase we can simplify the given configuration to a worst-case instance for that phase, reducing the number of cases to consider.

B. Outline of the Paper

We continue with a survey of related literature in Section II. Formal problem and protocol definitions are given in Section III. Section IV states our main result and introduces the above mentioned destructive moves argument. The analysis of RLS can be found in Section V. The paper closes with a short conclusion in Section VI

Petra Berenbrink and Peter Kling are now affiliated with University of Hamburg (Germany). Christopher Liaw is supported by an NSERC graduate scholarship. Abbas Mehrabian was supported by a PIMS Postdoctoral Fellowship and an NSERC Postdoctoral Fellowship.

[1]Here, "local" refers to the closeness of two consecutive solutions in the solution space, since they differ by the placement of at most one ball.

1530-2075/17 $31.00 © 2017 IEEE
DOI 10.1109/IPDPS.2017.52

II. Related Work

The following literature survey on load balancing adapts the balls-into-bins terminology and the notation n for the number of bins (resources) and m for the number of balls (users)[2]. Balls-into-bins games come in a vast number of variations and have a long tradition to model load balancing and similar problems [7]. Many of the most recent results consider the effect of the "power of 2 choices" [8] and processes that are in some form "self-stabilizing" [9], [10], [11]. We refer to [10], [11] for a recent and comprehensive overview of these variants. Here, we focus on three classes of more closely related balls-into-bins processes:

1) *Local Search:* With respect to our work, the most relevant type of protocols are *local search protocols*, where balls are sequentially activated and relocated with the goal to achieve a perfectly balanced situation. The term "local" is with respect to the solution space, since one step of the protocol changes the current solution by the placement of at most one ball. Protocols in this class are typically quite simple in that movement decisions depend only on the involved bins. Most closely related to our work are [2], [3]. They study exactly the same process (RLS). [2] claims an $\mathrm{O}(n^2)$ upper bound on the expected time to reach perfect balance. This is improved by [3] to $\mathrm{O}\big((\ln(n))^2 + \ln(n) \cdot n^2/m\big)$, which is $\mathrm{O}\big((\ln(n))^2\big)$ for the important case $m \gg n$. In the present work, we provide a tight upper bound of expected time $\Theta\big(\ln(n) + n^2/m\big)$ (i.e., $\Theta(\ln n)$ for $m \gg n$).

[12] study another local search protocol. Here, initially each ball picks two alternative bins and is placed arbitrarily in one of them. Now, in each step of the protocol a pair of bins (b_1, b_2) is chosen uniformly at random. If there is a ball in b_1 with alternative bin b_2, then this ball is placed in the least loaded bin among b_1 and b_2. One of the major results in [12] is that if the balls are initially placed via the power of 2 choices, then perfect balance is reached in $n^{\mathrm{O}(1)}$ steps (the hidden constant is ≥ 4). In the same situation, RLS needs only $\mathrm{O}(n^2)$ activations (see Phase 2 in Section V). Moreover, RLS can be started from an arbitrary load situation.

2) *Selfish Load Balancing:* Another strain of related work has a game theoretical background and is published under the theme of *selfish load balancing*[3]. A key difference to local search protocols is that balls act simultaneously. This introduces a problem: moves that, in isolation, improve the load of a ball might become bad if many balls perform this move. So, while one can compare the balancing times to local search protocols[4], such a direct comparison should be taken with a grain of salt. In particular, the results below suggest that the time to perfect balance in selfish load balancing has an inherent dependency on m, while there is no such dependency for local search protocols.

[14] consider selfish load balancing protocols with global knowledge (e.g., the average load). This allows them to reach perfect balance in expected $\mathrm{O}(\ln\ln m + \ln n)$ steps. [15] consider a protocol without global knowledge. Here, balls move to a randomly sampled bin with a probability depending on the load difference. They bound the expected balancing time by $\mathrm{O}\big(\ln\ln m + n^4\big)$. In a follow-up work, [16] suggested another protocol with expected balancing time $\mathrm{O}(\ln m + n \cdot \ln n)$. In comparison with [14], these results indicate that avoiding global knowledge in selfish load balancing might increase the dependency on n substantially.

3) *Threshold Load Balancing:* A third series of articles evolves around the idea of *threshold load balancing*: each ball has a threshold and moves with a certain probability to a random bin whenever its experienced load is above that threshold[5]. As in selfish load balancing, balls act simultaneously (resulting in a similar, seemingly inherent dependency on m). An interesting observation is that the RLS protocol can be seen as a (sequential) threshold protocol with an adaptive, local threshold (the sampled bin's load).

[17] introduced the idea of threshold load balancing and gave a protocol that balances up to a constant multiplicative factor in time $\mathrm{O}(\ln m)$ and up to an additive constant in time $\mathrm{O}\big(n^2 \cdot \ln m\big)$. [18], [19] extended these protocols to general graphs. Recent improvements by [20] show that, on general graphs, one can balance up to a constant multiplicative factor in time $\mathrm{O}(\tau_{\text{mix}} \cdot \ln m)$, τ_{mix} being the graph's mixing time.

III. Model and Notation

We describe our load balancing problem in terms of balls and bins. There are n bins (resources/processors) and m balls (users/tasks). We use the shorthands $[n] := \{1, 2, \ldots, n\}$ and $[m] := \{1, 2, \ldots, m\}$ for the set of bins and balls, respectively. A *configuration* $\boldsymbol{\ell} = (\ell_i)_{i \in [n]} \in \mathbb{N}_0^n$ is an n-dimensional vector with $\sum_{i \in [n]} \ell_i = m$. Its i-th component $\ell_i \in \mathbb{N}_0$ denotes the number of balls in bin i (its *load*).

We seek a simple distributed load balancing protocol (to be executed by each ball) such that all bins end up with almost the same load. To define this formally, let $\varnothing := m/n$ denote the *average load* of the system. The *discrepancy* of configuration $\boldsymbol{\ell}$ is $\operatorname{disc}(\boldsymbol{\ell}) := \max_{i \in [n]} |\ell_i - \varnothing|$. We say a configuration $\boldsymbol{\ell}$ is *x-balanced* if $\operatorname{disc}(\boldsymbol{\ell}) \leq x$ and *perfectly balanced* if $\operatorname{disc}(\boldsymbol{\ell}) < 1$.

Protocol Description: Let us formally describe the *Randomized Local Search* (RLS) protocol. Each ball is equipped with an exponential clock of rate 1, and the clocks are mutually independent. A ball is *activated* whenever its clock rings. Consider a configuration $\boldsymbol{\ell}$ and assume ball $j \in [m]$ in bin $i \in [n]$ is activated. Then, it chooses a *destination bin* $i' \in [n]$ uniformly at random and moves

[2]Note that n and m may be swapped in some other papers.

[3]See [13] for a comprehensive survey.

[4]In one selfish load balancing time step all m balls are activated. Similarly, in one time unit of RLS m balls are activated in expectation.

[5]Note that [14] also falls into this category.

from i to i' if and only if $\ell_i \geq \ell_{i'} + 1$.[6]

Randomized Local Search (RLS)

```
{code executed by ball j in bin i when j is activated}
sample random bin i'
if ℓ_i ≥ ℓ_i' + 1:
    move to bin i'
```

This describes a continuous time stochastic process. Starting from an initial configuration $\boldsymbol{\ell}$ we write $\boldsymbol{\ell}(t)$ for the configuration at time t. Note that $\boldsymbol{\ell}$ depends on the random ball activations as well as the random destination bin choices of each ball. Notice that RLS has the desirable properties that the discrepancy never increases, the minimum load never decreases, and the maximum load never increases.

Additional Notation: We use $\mathbb{N} := \{1, 2, \dots\}$ for the (positive) natural numbers and $\mathbb{N}_0 := \{0, 1, 2 \dots\}$ to include zero in the natural numbers. For any $k \in \mathbb{N}$ we define $H_k := \sum_{i=1}^{k} 1/i = \ln(k) + \mathrm{O}(1)$ as the k-th harmonic number. Given two random variables A and B, we write $B \preceq A$ if A stochastically dominates B (i.e., if $\Pr(A \geq x) \geq \Pr(B \geq x)$ for all $x \in \mathbb{R}$), and we write $A \overset{d}{=} B$ if A and B are equal in distribution. We say an event E holds with high probability (w.h.p.) if $\Pr(E) \geq 1 - n^{-\Omega(1)}$. $\mathrm{Bin}(n, p)$ denotes a binomial random variable with parameters n and p, and $\mathrm{Exp}(\lambda)$ denotes an exponential random variable with parameter λ.

IV. Results and Proof Outline

Our main result is the following theorem.

Theorem 1. *Consider a system of n identical bins and m identical balls in an arbitrary initial configuration. Let T be the time when RLS reaches a perfectly balanced configuration. We have $\mathbb{E}[T] = \mathrm{O}\big(\ln(n) + n^2/m\big)$ and w.h.p. $T = \mathrm{O}\big(\ln(n) + \ln(n) \cdot n^2/m\big)$.*

As observed in [3], our bounds are asymptotically tight. Indeed, assume that initially all balls are in the same bin. To reach a perfectly balanced configuration, we need to activate at least $m - \varnothing$ balls. The expected time to do so is at least $\sum_{k=\varnothing+1}^{m} 1/k = H_m - H_\varnothing = \Omega(\ln n)$, yielding the first term in the lower bound. For the second term, suppose $\varnothing$ is an integer, and consider a configuration in which exactly one bin has load $\varnothing + 1$, one other bin has load $\varnothing - 1$, and every other bin has load $\varnothing$. The time to balance perfectly is exactly the time until one of the balls in the overloaded bin is activated and samples the underloaded bin. The latter is an exponential random variable with parameter $(\varnothing + 1) \cdot 1/n$. Thus, the expected time to reach perfect balance is $n/(\varnothing + 1) = \Omega\big(n^2/m\big)$. Requiring a high probability result gives an additional $\ln n$ factor.

[6]The protocol studied by [2], [3] is slightly different: they allow movement from i to i' if $\ell_i > \ell_{i'} + 1$. However, since the bins and the balls are identical, the two protocols have precisely the same balancing time.

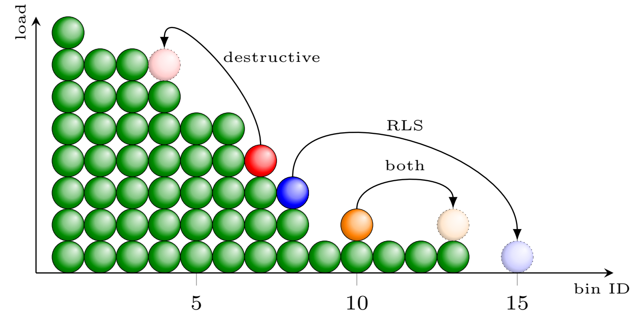

Figure 1. Illustration of RLS moves versus destructive moves.

Destructive Moves: Before we continue, we state an auxiliary lemma which is used throughout our analysis. In a nutshell, it states that reversing a ball movement of RLS cannot improve the time to reach perfect balance. This intuitively simple observation turns out to be extremely useful. Basically, we will be able to reduce arbitrary initial configurations to "well shaped" configurations (decreasing the number of cases to consider) and to ignore certain (at the moment unwanted) moves of the protocol.

To formalize this idea, consider a configuration $\boldsymbol{\ell}$. We call a movement of a ball from bin i to bin j *destructive* if $\ell_i \leq \ell_j + 1$. Note that a movement is destructive if and only if it is the reversal of a valid protocol move. Also note that, if $\ell_i = \ell_j + 1$, then a move from i to j is both a valid protocol move and a destructive move (see Figure 1). Such a move is called a *neutral move*.

Lemma 2 (Destructive Majorization Lemma)**.** *For any $t \geq 0$, consider the load vector $\boldsymbol{\ell}(t)$ resulting from protocol RLS at time t. Let $\tilde{\boldsymbol{\ell}}(t)$ denote the load vector resulting from RLS at time t under the presence of an adversary[7] who performs an arbitrary number of destructive moves after each ball movement. Then $\mathrm{disc}(\boldsymbol{\ell}(t)) \preceq \mathrm{disc}(\tilde{\boldsymbol{\ell}}(t))$.*

Proof: For $k \in \mathbb{N}_0$, consider the random process $P^{(k)}$ that executes our protocol starting in the initial configuration $\boldsymbol{\ell}(0)$ under the presence of the first k adversarial (destructive) moves (ignoring any further adversarial moves). Let $\boldsymbol{\ell}^{(\boldsymbol{k})}(t)$ denote the configuration at time t under process $P^{(k)}$. Note that $\boldsymbol{\ell}^{(\mathbf{0})}(t) \overset{d}{=} \boldsymbol{\ell}(t)$ and $\boldsymbol{\ell}^{(\boldsymbol{\infty})}(t) \overset{d}{=} \tilde{\boldsymbol{\ell}}(t)$ for all $t \geq 0$. We show that for all $k \in \mathbb{N}_0$ and $t \geq 0$ we have $\mathrm{disc}\big(\boldsymbol{\ell}^{(\boldsymbol{k})}(t)\big) \preceq \mathrm{disc}\big(\boldsymbol{\ell}^{(\boldsymbol{k+1})}(t)\big)$. The lemma follows from this via the transitivity of stochastic domination.

Call a configuration $\boldsymbol{\ell}'$ *close* to a configuration $\boldsymbol{\ell}$ if $\boldsymbol{\ell}'$ is constructed from $\boldsymbol{\ell}$ by at most one destructive move. Two immediate observations are:

(i) Either $\boldsymbol{\ell}' = \boldsymbol{\ell}$ or there are two bins $i_{\mathrm{L}} \neq i_{\mathrm{R}}$ with $\ell_{i_{\mathrm{R}}} \leq \ell_{i_{\mathrm{L}}} + 1$ such that $\ell'_{i_{\mathrm{L}}} = \ell_{i_{\mathrm{L}}} + 1$, $\ell'_{i_{\mathrm{R}}} = \ell_{i_{\mathrm{R}}} - 1$, and

[7]The adversary may have full knowledge of the protocol and its random choices (even future).

$\ell'_i = \ell_i$ for all $i \notin \{ i_L, i_R \}$ (i.e., $\boldsymbol{\ell'}$ is constructed from $\boldsymbol{\ell}$ by a destructive move from i_R to i_L[8].

(ii) We have $\mathrm{disc}(\boldsymbol{\ell}) \leq \mathrm{disc}(\boldsymbol{\ell'})$.

Fix a $k \in \mathbb{N}_0$ and consider the processes $P^{(k)}$ and $P^{(k+1)}$. Initially, we have $\boldsymbol{\ell^{(k)}}(0) = \boldsymbol{\ell^{(k+1)}}(0)$ and, thus, $\boldsymbol{\ell^{(k+1)}}(0)$ is close to $\boldsymbol{\ell^{(k)}}(0)$. To show the majorization $\mathrm{disc}\big(\boldsymbol{\ell^{(k)}}(t)\big) \preceq \mathrm{disc}\big(\boldsymbol{\ell^{(k+1)}}(t)\big)$, it is sufficient (by the above observations) to define a coupling between $P^{(k)}$ and $P^{(k+1)}$ that maintains $\boldsymbol{\ell^{(k+1)}}(t)$ being close to $\boldsymbol{\ell^{(k)}}(t)$ for all $t \geq 0$. So fix $t \in \mathbb{N}$ and assume $\boldsymbol{\ell^{(k+1)}}(t-1)$ is close to $\boldsymbol{\ell^{(k)}}(t-1)$. To simplify notation, let $\boldsymbol{\ell} := \boldsymbol{\ell^{(k)}}(t-1)$ and $\boldsymbol{\ell'} := \boldsymbol{\ell^{(k+1)}}(t-1)$. Without loss of generality, let both $\boldsymbol{\ell}$ and $\boldsymbol{\ell'}$ be sorted non-increasingly, such that $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_n$ and $\ell'_1 \geq \ell'_2 \geq \cdots \geq \ell'_n$[9]. If $\boldsymbol{\ell'} = \boldsymbol{\ell}$ we use the identity coupling. Otherwise, $\boldsymbol{\ell'}$ is constructed from $\boldsymbol{\ell}$ by a destructive move from a bin i_R to a bin i_L with $i_L < i_R$. Without loss of generality, let m be the ball in which $\boldsymbol{\ell}$ and $\boldsymbol{\ell'}$ differ and assume all other balls $j \in [m-1]$ are in the same bin in configuration $\boldsymbol{\ell}$ and $\boldsymbol{\ell'}$. We couple the random choices of $P^{(k+1)}$ to the random choices of $P^{(k)}$ as follows: Assume $P^{(k)}$ activates ball $j \in [m]$ who is in source bin i_S in $\boldsymbol{\ell}$ and chooses destination bin $i_D \in [n]$ (the i_D-th fullest bin in $\boldsymbol{\ell}$). Then $P^{(k+1)}$ activates j and chooses destination bin i_D.

See Figure 2 for an illustration of the coupling and the following case discrimination. It remains to show that the resulting configuration $\boldsymbol{\ell^{(k+1)}}(t)$ of $P^{(k+1)}$ is close to the resulting configuration $\boldsymbol{\ell^{(k)}}(t)$ of $P^{(k)}$. This is immediate if $\boldsymbol{\ell} = \boldsymbol{\ell'}$ (since then we use the identity coupling). Otherwise, $\boldsymbol{\ell'}$ is constructed from $\boldsymbol{\ell}$ by a destructive move from a bin i_R to a bin i_L with $i_L < i_R$. We distinguish the following cases depending on the source and destination bins of process $P^{(k)}$:

(1) $\underline{i_S, i_D \notin \{ i_L, i_R \}}$
The processes behave identical and $\boldsymbol{\ell^{(k+1)}}(t)$ still results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_L.

(2) $\underline{i_S = i_R}$
The activated ball might be m. If that is the case, $P^{(k)}$ activates a ball in bin i_R, while $P^{(k+1)}$ activates a ball in bin i_L. We distinguish three subcases depending on the destination bin: If $i_D \leq i_L$, both moves fail and nothing changes. If $i_L < i_D \leq i_R$, only the move in $P^{(k+1)}$ succeeds and either the configurations become identical (if $i_D = i_R$) or $\boldsymbol{\ell^{(k+1)}}(t)$ results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_D. If $i_D > i_R$, both moves succeed and the configurations become identical.
If the activated ball is not m, both processes activate a ball in i_R. We distinguish three subcases depending on the destination bin: If $i_D \in \{ i \in [n] \mid \ell_i > \ell_{i_R} - 1 \}$, both moves fail and nothing changes. If $i_D \in \{ i \in [n] \mid \ell_i = \ell_{i_R} - 1 \}$, only the move in $P^{(k)}$ succeeds (a neutral move) and $\boldsymbol{\ell^{(k+1)}}(t)$ results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_D to i_L. If $i_D \in \{ i \in [n] \mid \ell_i < \ell_{i_R} - 1 \}$, both moves succeed and $\boldsymbol{\ell^{(k+1)}}(t)$ still results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_L.

(3) $\underline{i_S = i_L}$
The activated ball cannot be m, so both processes activate a ball in bin i_L. We distinguish three subcases depending on the destination bin: If $i_D \in \{ i \in [n] \mid \ell'_i > \ell'_{i_L} - 1 \}$, both moves fail and nothing changes. If $i_D \in \{ i \in [n] \mid \ell'_i = \ell'_{i_L} - 1 \}$, only the move in $P^{(k+1)}$ succeeds (a neutral move) and $\boldsymbol{\ell^{(k+1)}}(t)$ results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_D. If $i_D \in \{ i \in [n] \mid \ell'_i < \ell'_{i_L} - 1 \}$, both moves succeed and $\boldsymbol{\ell^{(k+1)}}(t)$ still results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_L.

(4) $\underline{i_S \notin \{ i_L, i_R \}, i_D = i_L}$
The activated ball cannot be m, so both processes activate a ball in the same bin i_S. We distinguish three subcases depending on the source bin: If $i_S \in \{ i \in [n] \mid \ell'_i < \ell'_{i_L} \}$, both moves fail and nothing changes. If $i_S \in \{ i \in [n] \mid \ell'_i = \ell'_{i_L} \}$, only the move in $P^{(k)}$ succeeds (a neutral move) and $\boldsymbol{\ell^{(k+1)}}(t)$ results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_S. If $i_S \in \{ i \in [n] \mid \ell'_i > \ell'_{i_L} \}$, both moves succeed and $\boldsymbol{\ell^{(k+1)}}(t)$ still results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_L.

(5) $\underline{i_S \notin \{ i_L, i_R \}, i_D = i_R}$
The activated ball cannot be m, so both processes activate a ball in the same bin i_S. We distinguish three subcases depending on the source bin: If $i_S \in \{ i \in [n] \mid \ell_i < \ell_{i_R} \}$, both moves fail and nothing changes. If $i_S \in \{ i \in [n] \mid \ell_i = \ell_{i_R} \}$, only the move in $P^{(k+1)}$ succeeds (a neutral move) and $\boldsymbol{\ell^{(k+1)}}(t)$ results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_S to i_L. If $i_S \in \{ i \in [n] \mid \ell_i > \ell_{i_R} \}$, both moves succeed and $\boldsymbol{\ell^{(k+1)}}(t)$ still results from $\boldsymbol{\ell^{(k)}}(t)$ by a destructive move from i_R to i_L.

In all cases, $\boldsymbol{\ell^{(k+1)}}(t)$ is close to $\boldsymbol{\ell^{(k)}}(t)$. This completes the proof. ∎

[8] If you think of the bins ordered non-increasingly, the destructive move goes from **R**ight (i_R) to **L**eft (i_L).

[9] RLS is ignorant of the bin order, so we can assume it sorts configurations non-increasingly before each step. Also, sorting both $\boldsymbol{\ell}$ and $\boldsymbol{\ell'}$ maintains $\boldsymbol{\ell'}$ being close to $\boldsymbol{\ell}$: Assume $\boldsymbol{\ell}$ is sorted and construct $\boldsymbol{\ell'}$ from $\boldsymbol{\ell}$ by a destructive move from i'_R to i'_L. This implies $i'_L \leq i'_R$. Let $\vec{\boldsymbol{\ell'}}$ be the sorted version of $\boldsymbol{\ell'}$, $i_R := \max\{ i \in [n] \mid \ell_i = \ell_{i'_R} \}$, and $i_L := \min\{ i \in [n] \mid \ell_i = \ell_{i'_L} \}$. Then $\vec{\boldsymbol{\ell'}}$ is constructed from $\boldsymbol{\ell}$ by a destructive move from i_R to i_L and $i_L \leq i_R$.

V. Analysis of RLS

In this section we analyze the RLS protocol. For simplicity we would like to assume that $m \geq 2n$ and that n divides m. The following two lemmas justify these assumptions.

Lemma 3. *Suppose $m \leq n$ and let T be the time until perfect balance. Then $\mathbb{E}[T] \leq \mathrm{O}(n)$ and $T \leq \mathrm{O}(n \ln(n))$ w.h.p.*

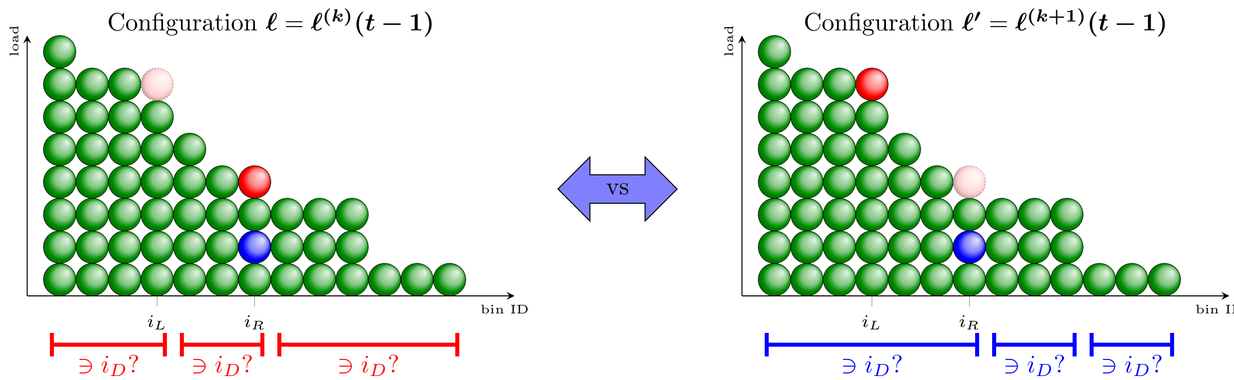

Figure 2. Illustration of the coupling used for Lemma 2. The red ball is m. Thus, ℓ' results from ℓ by a destructive move from $i_R = 7$ to $i_L = 4$. One of the balls (e.g., the blue or red ball) is activated in both processes and tries to move from its source to a random destination. If the red ball is activated, the source is $i_S = 7$ on the left and 4 on the right. If the blue ball is activated, the source is $i_S = 7$ both on the left and right. The destination i_D is always the same on both sides. If the red ball is activated, the red intervals indicate the three subcases of the first part of Case (2) from the proof. If the blue ball is activated, the blue intervals indicate the three subcases of the second part of the same case (we drew the intervals below the right figure for space reasons). Similar pictures can be drawn for the remaining cases.

Proof: By Lemma 2, we may assume that all balls start in the first bin, and that we may wait for each of the m balls to move to m distinct empty bins (and ignore any other move). Note that this is possible since $m \leq n$. If there are $2 \leq r \leq m$ balls left in the first bin, then there are at least $r-1$ empty bins, hence the time it takes for one of the r balls to be activated and choose an empty bin is an exponential with rate $r \times (r-1)/n$. Therefore, the total expected time for the m balls to choose m distinct empty bins is at most

$$\sum_{r=2}^{m} n/r(r-1) < \sum_{r=1}^{\infty} 2n/r^2 = \mathrm{O}(n). \qquad (1)$$

This shows $\mathbb{E}[T] = \mathrm{O}(n)$. Lemma 16 then implies $T \leq \mathrm{O}(n \ln(n))$ w.h.p. ∎

Note that the above lemma implies Theorem 1 when $m \leq n$. The next lemma implies that, in order to prove Theorem 1, it is sufficient to consider the case when n divides m.

Lemma 4. *Suppose $m \geq n$ and write $m = kn + r$ for some $k \in \mathbb{N}$ and $r \in \{0, \ldots, n-1\}$. Let T be the time until perfect balance. Furthermore, suppose that for any configuration with n bins and kn balls, RLS balances in expected time at most $f(n, kn)$ and in time at most $g(n, kn)$ w.h.p. Then $\mathbb{E}[T] \leq \mathrm{O}(\ln(n)) + f(n, kn)$ and $T \leq \mathrm{O}(\ln(n)) + g(n, kn)$ w.h.p.*

Proof: By Lemma 2, we may assume that all balls start in the first bin. We first wait for r balls in the first bin to move to r distinct empty bins (and ignore any other move). Once each of these balls finds a bin, we no longer allow it to move. After these r balls are moved to distinct bins, we run the RLS protocol assuming it had only kn balls (these assumptions can only slow down the protocol by Lemma 2). To complete the proof, we need only show that the running time of the initial phase is $\mathrm{O}(\ln n)$ in expectation and with high probability. Denote this running time by T'.

Note that $T' = \sum_{i=1}^{r} T_i$, where T_i is the time for the ith ball to activate and choose an empty bin. Observe that T_i is an exponential random variable with parameter $(kn + r - i + 1)(n-i)/n > n - i$. Thus,

$$\mathbb{E}[T'] < \sum_{i=1}^{r} \frac{1}{n-i} \leq \mathrm{O}(\ln(n)), \text{ and} \qquad (2)$$

$$\mathrm{Var}[T'] < \sum_{i=1}^{r} \frac{1}{(n-i)^2} \leq \mathrm{O}(1). \qquad (3)$$

By concentration of sums of exponential random variables (see Lemma 14), we have $T' = \mathrm{O}(\ln n)$ w.h.p. ∎

In light of the previous two lemmas, in the following we assume $m \geq n$ and that n divides m. Our analysis of RLS proceeds via the following three phases:

Phase 1: In Section V-A we show that, from any initial configuration, w.h.p. it takes time $\mathrm{O}(\ln n)$ to become $\mathrm{O}(\ln n)$-balanced.

Phase 2: In Section V-B we show that, from any $\mathrm{O}(\ln n)$-balanced configuration, it takes expected time $\mathrm{O}(n/\varnothing)$ to become 1-balanced.

Phase 3: In Section V-C we show that, from any 1-balanced configuration, it takes expected time $\mathrm{O}(n/\varnothing)$ to become perfectly balanced.

Standard arguments imply that Phase 1 takes expected time $\mathrm{O}(\ln n)$, and that Phases 2 and 3 take time $\mathrm{O}(\ln n \cdot n/\varnothing)$ w.h.p. (see Lemma 16 and Lemma 17 in the appendix). Since $\varnothing = m/n$, these imply the total time to reach perfect balance is in expectation $\mathrm{O}\big(\ln(n) + n^2/m\big)$ and w.h.p. $\mathrm{O}\big(\ln(n) + \ln(n) \cdot n^2/m\big)$.

A. Phase 1: Reaching an $\mathrm{O}(\ln n)$-balanced Configuration

We first consider how long it takes to go from an arbitrary initial configuration to a configuration ℓ with $\mathrm{disc}(\boldsymbol{\ell}) =$

$\mathrm{O}(\ln n)$. We distinguish between two cases, depending on whether $\varnothing$ is large or small.

Phase 1 for small $\varnothing$: The easier case, for $\varnothing \leq 16 \cdot \ln n$, is covered by the following lemma.

Lemma 5. *Assume $\varnothing \leq 16 \cdot \ln n$ and consider an arbitrary initial configuration $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$. Let $T := \inf\{\, t \mid \operatorname{disc}(\boldsymbol{\ell}(t)) \leq 96 \cdot \ln n \,\}$. Then, w.h.p., $T = \mathrm{O}(\ln n)$.*

Proof: By the assumption we have $\varnothing - 96 \cdot \ln n \leq 0$. Thus, $\ell_i(t) \geq 0 \geq \varnothing - 96 \cdot \ln n$ for all bins $i \in [n]$ and times $t \geq 0$. Consequently, T is the first time t such that $\ell_i(t) \leq \varnothing + 96 \cdot \ln n$ for all i. By Lemma 2, we can assume that, initially, all balls are in the same bin (by performing up to $m-1$ destructive movements). So assume (w.l.o.g.) that all balls start in bin 1. Let us first bound the time T' until $m - \varnothing$ balls move from bin 1 to one of the other $n-1$ bins. Applying Lemma 2 once more, we ignore movements of balls to bin 1, movements between any of the remaining $n-1$ bins, and assume that all movements from bin 1 to any of the remaining $n-1$ bins are successful. If T_i denotes the time in which the load of bin 1 decreases from i to $i-1$, we have $T' = \sum_{i=\varnothing+1}^{m} T_i$. The different T_i are independent exponential random variables with parameter $i \cdot (n-1)/n$. This yields

$$\mathbb{E}[T'] = \sum_{i=\varnothing+1}^{m} \left(i \cdot \frac{n-1}{n} \right)^{-1} \leq 2 \cdot \ln n, \text{ and} \tag{4}$$

$$\operatorname{Var}[T'] = \sum_{i=\varnothing+1}^{m} \left(i \cdot \frac{n-1}{n} \right)^{-2} = \mathrm{O}(1/\varnothing). \tag{5}$$

By concentration of sums of independent exponential random variables (Lemma 14 in Appendix A), we have, w.h.p., $T' = \mathrm{O}(\ln n)$.

To complete the proof, it suffices to show that w.h.p. $T \leq T'$. Whenever one of these $m - \varnothing$ balls is activated, we may assume it chooses one other bin uniformly at random and moves there (without checking the load). While this might violate our original protocol (balls could move to a bin with a higher load), such a violation would be due to a destructive movement. By Lemma 2, this merely slows the process down. Hence, at time T', the number of balls in every other bin is $\operatorname{Bin}(m - \varnothing, 1/(n-1))$, which has mean $(m - \varnothing)/(n-1) = \varnothing$. Using a Chernoff bound (Lemma 13) with the union bound, the maximum of n such binomials is not more than $96 \cdot \ln n$ w.h.p. ∎

Phase 1 for large $\varnothing$: We now turn to the more interesting case where $\varnothing > 16 \cdot \ln n$ and consider two subphases. First, Lemma 6 shows that we reach a $\varnothing/2$-balanced configuration in time $\mathrm{O}(\ln n)$. The proof is basically identical to the proof of Lemma 5, the only difference being that we use a different Chernoff bound at the end. Afterward, Lemma 7 shows that it takes an additional $\mathrm{O}(\ln n)$ time to become $\mathrm{O}(\ln n)$-balanced.

Lemma 6. *Assume $\varnothing > 16 \cdot \ln n$ and consider an arbitrary initial configuration $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$. Let $T := \inf\{\, t \mid \operatorname{disc}(\boldsymbol{\ell}(t)) \leq \varnothing/2 \,\}$. Then, w.h.p., $T = \mathrm{O}(\ln n)$.*

Proof Sketch: The proof is basically identical to the proof of Lemma 5, the only difference being that we use a Chernoff bound for large expected values (Inequality (7)) at the end. As in the previous proof, by Lemma 2 (destructive movements) we can assume all balls to be in bin 1 at time 0, and define T' as the time until $m - \varnothing$ balls move to one of the other $n-1$ bins. The same calculations yield (w.h.p.) $T' \leq \mathrm{O}(\ln n)$. Once more, Lemma 2 allows us to majorize the ball distribution in each of these $n-1$ remaining bins at time T' by the binomial distribution $\operatorname{Bin}(m - \varnothing, 1/(n-1))$ with mean $\varnothing$. Applying a Chernoff bound (this time the variant for large expected values, i.e., Equation (7)) and a union bound yields that, w.h.p., the load of all bins is within $[\varnothing - 2\sqrt{\varnothing \ln n}, \varnothing + 2\sqrt{\varnothing \ln n}]$. Since $\varnothing > 16 \ln n$, this means w.h.p. $\operatorname{disc}(\boldsymbol{\ell}(T')) \leq 2\sqrt{\varnothing \ln n} \leq \varnothing/2$. ∎

Lemma 7. *Assume $m > n$ and consider an initial configuration $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$ with $\operatorname{disc}(\boldsymbol{\ell}) \leq \varnothing/2$. Let $T := \inf\{\, t \mid \operatorname{disc}(\boldsymbol{\ell}(t)) \leq 8 \ln n \,\}$. Then, w.h.p., $T \leq \mathrm{O}(\ln n)$.*

For proving Lemma 7, we will apply the following lemma iteratively.

Lemma 8. *Consider an initial configuration $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$ with $\operatorname{disc}(\boldsymbol{\ell}) \leq x$ for some $x \geq 4 \cdot \ln n$. Let $T_x := \inf\{\, t \mid \operatorname{disc}(\boldsymbol{\ell}(t)) \leq 2\sqrt{x \cdot \ln n} \,\}$. Then, with probability $\geq 1 - n^{-1}$ we have $T_x \leq \ln\left(\frac{\varnothing + x}{\varnothing - x}\right)$. Moreover, T_x is dominated by $Y \cdot \ln\left(\frac{\varnothing + x}{\varnothing - x}\right)$, where Y is a geometric random variable with parameter $1 - n^{-1}$.*

Proof: First note that the second statement (domination by $Y \cdot \ln\left(\frac{\varnothing + x}{\varnothing - x}\right)$) follows from the first one via Lemma 17. We now prove the first statement. Assume for simplicity that $\varnothing \pm x$ are integers. Let $p := 2x/(x + \varnothing)$ and $t := \ln(\varnothing + x) - \ln(\varnothing - x)$. Note that the probability of activation of each ball during the interval $[0, t]$ is $1 - \exp(-t) = p$. Using Lemma 2 we make the following simplifying assumptions (see also Figure 3):

1) At time 0, we move some balls from the $n/2$ lightest bins (the *light bins*) to the $n/2$ heaviest bins (the *heavy bins*) in such a way that all light bins have exactly $\varnothing - x$ balls and all heavy bins have exactly $\varnothing + x$ balls. All these moves are destructive, thus we can assume (by Lemma 2) that we start in the resulting configuration. Bins labeled as light/heavy in the beginning keep this label (regardless of how their loads changes) during the time interval $[0, t]$.
2) During the time interval $[0, t]$, we ignore activations of balls in light bins (as we could reverse them via Lemma 2).
3) During the time interval $[0, t]$, we ignore movements between any two heavy bins (as we could reverse them

via Lemma 2).

4) During the time interval $[0,t]$, if a ball in a heavy bin i is activated and tries to move to a light bin i', it does so unconditionally (i.e., even if $\ell_i < \ell_{i'}+1$; in that case it is a destructive move which we may allow via Lemma 2).

First, consider a heavy bin. During the time interval $[0,t]$, each of its balls is activated with probability p, and moves to a light bin with probability $1/2$. So this bin loses $\mathrm{Bin}(\varnothing + x, p/2)$ balls. This binomial has expected value $x \geq 4\ln n$. Thus, by Chernoff, with probability $\geq 1-n^{-2}$ its value is in $[x-2\sqrt{x\ln n}, x+2\sqrt{x\ln n}]$. This bin had $\varnothing + x$ balls initially, so with probability $\geq 1-n^{-2}$, it will have between $\varnothing - 2\sqrt{x\cdot\ln n}$ and $\varnothing + 2\sqrt{x\cdot \ln n}$ balls at time t.

Next, consider a light bin. There are $(\varnothing + x)(n/2)$ balls it can potentially receive during the time interval $[0,t]$. It receives each one with probability p/n, so the number of balls it receives is $\mathrm{Bin}((\varnothing+x)(n/2), p/n)$. This binomial has expected value $x \geq 4\ln n$. Thus, by Chernoff, with probability $\geq 1-n^{-2}$ its value is in $[x-2\sqrt{x\ln n}, x+2\sqrt{x\ln n}]$. This bin had $\varnothing - x$ balls initially, so with probability $\geq 1-n^{-2}$, it will have between $\varnothing - 2\sqrt{x\cdot\ln n}$ and $\varnothing + 2\sqrt{x\cdot\ln n}$ balls at time t. Applying the union bound over all bins completes the lemma's proof. ■

We now present the proof of Lemma 7.

Proof of Lemma 7: Note that, whenever $x \leq \varnothing/2$, we have $\ln(\varnothing + x) - \ln(\varnothing - x) = \ln\left(1+\frac{2x}{\varnothing - x}\right) \leq \frac{2x}{\varnothing - x} \leq 4x/\varnothing$. Define $t_0 := 0$, $x_0 := \varnothing/2$, and $x_k := \sqrt{4x_{k-1}\cdot \ln n}$ for $k > 0$. By induction, $x_k \leq 4\ln(n)\cdot x_0^{1/2^k}$. Let $r = \log_2\log_2\varnothing$. Since $x_0 \leq \varnothing$, we have $x_r \leq 4\ln(n)\cdot \varnothing^{1/\log_2\varnothing} = 8\cdot\ln n$. Let $Y_1, Y_2, \dots$ be independent geometric random variables with parameter $1-n^{-1}$. Applying Lemma 8 iteratively, the time to reach an x_r-balanced configuration is stochastically dominated by

$$Z_r := \sum_{i=0}^{r-1} Y_i \cdot 4x_i/\varnothing \leq \sum_{i=0}^{r-1} c_i \cdot Y_i, \tag{6}$$

where $c_i := 16\cdot\ln(n)\cdot x_0^{1/2^i}/\varnothing$. Straightforward calculations yield $\max c_i = \mathrm{O}(\ln n)$, $\sum c_i = \mathrm{O}(\ln n)$, and $\sum c_i^2 = \mathrm{O}(\ln^2 n)$. By concentration of sums of geometric random variables (see Lemma 15), we find that $Z_r = \mathrm{O}(\ln n)$ w.h.p., completing the proof. ■

B. Phase 2: Reaching a 1-balanced Configuration

Lemma 9. *Consider an initial configuration* $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$ *with* $\mathrm{disc}(\boldsymbol{\ell}) = \mathrm{O}(\ln n)$. *Let* $T := \inf\{\, t \mid \mathrm{disc}(\boldsymbol{\ell}(t)) \leq 1 \,\}$. *Then* $\mathbb{E}[T] = \mathrm{O}(n/\varnothing)$.

Before we prove Lemma 9, let us introduce the notion of *overloaded bins/balls*: A bin i is *overloaded* if $\ell_i > \varnothing$. The quantity $\sum_i \max\{\,0, \ell_i(t) - \varnothing\,\}$ is called *the number of overloaded balls*. If we enumerate the balls in each bin arbitrarily using natural numbers, this is simply the number of balls whose number is greater than $\varnothing$. For instance, in Figure 3 (left), the number of overloaded balls is 6. Note that this is also the number of "holes" (i.e., $\sum_i \max\{\,0, \varnothing - \ell_i(t)\,\}$).

We split this phase into two subphases. First, we show that it takes $\mathrm{O}\left((\ln n)^2/\varnothing\right)$ time to reduce the number of overloaded balls to n (Lemma 10). Afterward (Lemma 11), we prove that if both the discrepancy is logarithmic and the number of overloaded balls is small, a 1-balanced configuration is reached in time $\mathrm{O}(n/\varnothing)$. Together, these immediately imply Lemma 9.

Lemma 10. *Suppose* $\mathrm{disc}(\boldsymbol{\ell}(0)) = \mathrm{O}(\ln n)$ *and let* $T := \inf\{\, t > 0 \mid \sum_i \max\{\,0, \ell_i(t) - \varnothing\,\} \leq n \,\}$. *Then* $\mathbb{E}[T] = \mathrm{O}\left((\ln n)^2/\varnothing\right)$.

Proof: Fix a time $t > 0$ and let A denote the number of overloaded balls. Let h and k be the number of bins with load $> \varnothing$ and $< \varnothing$, respectively. Observe that $\varnothing - \mathrm{O}(\ln n) \leq \ell_{\min} \leq \ell_{\max} \leq \varnothing + \mathrm{O}(\ln n)$ implies $\min\{\,h,k\,\} = \Omega(A/\ln n)$ and, thus, $h\cdot k = \Omega\left(A^2/(\ln n)^2\right)$.

We wait for some ball in some overloaded bin to choose an underloaded bin and move there. Using Lemma 2, we ignore any other move. There are h overloaded bins and at least $h\cdot\varnothing$ balls in them. The probability that such a ball, when activated, chooses an underloaded bin is k/n. Hence, the expected time for such a move to happen is at most $\frac{1}{h\cdot\varnothing}\cdot\frac{n}{k} = \mathrm{O}\left(\frac{n\cdot(\ln n)^2}{A^2\cdot\varnothing}\right)$. When such a move happens, the value of A decreases by 1. Hence, total expected time to reduce A to n is bounded by

$$\sum_{A=n}^{\infty} \mathrm{O}\left(\frac{n\cdot(\ln n)^2}{A^2\cdot\varnothing}\right) = \mathrm{O}\left(\frac{n\cdot(\ln n)^2}{\varnothing}\sum_{A=n}^{\infty} A^{-2}\right)$$
$$= \mathrm{O}\left(\frac{n\cdot(\ln n)^2}{\varnothing}\int_{n-1}^{\infty} x^{-2}\,\mathrm{d}x\right) = \mathrm{O}\left((\ln n)^2/\varnothing\right).$$

■

Lemma 11. *Assume that* $\mathrm{disc}(\boldsymbol{\ell}(\mathbf{0})) = \mathrm{O}(\ln n)$ *and that the number of overloaded balls is at most* n. *Let* $T := \inf\{\, t \mid \mathrm{disc}(\boldsymbol{\ell}(t)) \leq 1 \,\}$. *Then* $\mathbb{E}[T] = \mathrm{O}(n/\varnothing)$.

Proof: Let A denote the number of overloaded balls. Suppose that h bins have load $> \varnothing$, r bins have load $= \varnothing$, and k bins have load $< \varnothing$. Note that $h + r + k = n$. We use the quantity $3A - k - h$ as a potential function and prove the following claim.

Claim. if $A > \min\{\,h,k\,\}$, then the expected time to decrease $3A - k - h$ by at least 1 is $\leq 3/\varnothing$.

To see that this implies the lemma, note that we always have $A \geq \max\{\,h,k\,\}$; moreover, if $A = \min\{\,h,k\,\}$ then $\varnothing - 1 \leq \ell_{\min} \leq \ell_{\max} \leq \varnothing + 1$, which means the discrepancy is 1. Since $3A - k - h$ is always between 0 and $3n$ and never increases over time, the claim implies that it takes expected time $\mathrm{O}(n/\varnothing)$ to achieve discrepancy 1.

We prove the claim by considering three cases.

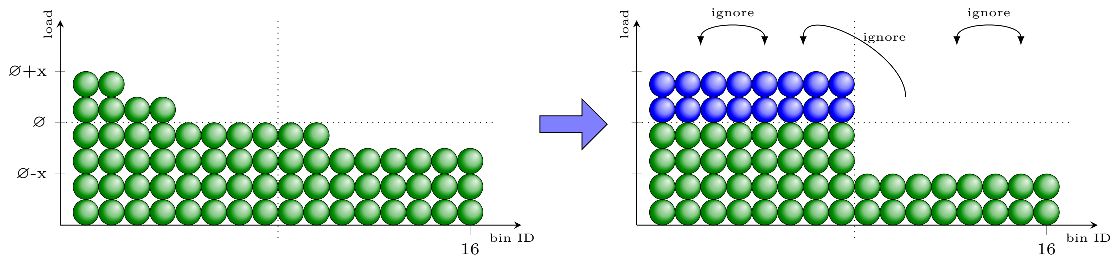

Figure 3. Using Lemma 2, we reorder the balls such that the distance to the average is exactly $x = 2$ in the first and last eight bins. We also allow only moves from heavy to light bins; all other moves are ignored.

Case 1: $r \geq n/3$ and $A > h$. We wait for some ball in a bin with load $> \varnothing + 1$ to choose a bin with load $\varnothing$ and move there (ignoring any other move via Lemma 2). Since $A > h$, there is at least one bin with load $> \varnothing + 1$ and, hence, $\geq \varnothing$ such balls. The probability that such a ball, when activated, chooses a bin with load $\varnothing$ is r/n. Hence the expected time for such a move to happen is at most $\frac{1}{\varnothing} \cdot \frac{n}{r} \leq 3/\varnothing$. When such a move happens, A and k do not change but h increases by 1, so the potential decreases by 1, as required.
Case 2: $r \geq n/3$ and $A > k$. We wait for some ball in a bin with load $\varnothing$ to choose a bin with load $< \varnothing - 1$ and move there (ignoring any other move via Lemma 2). Since $A > k$, there is at least one bin with load $< \varnothing - 1$ and, hence, the expected time for such a move to happen is at most $\frac{1}{r \cdot \varnothing} \cdot \frac{n}{1} \leq 3/\varnothing$. When such a move happens, A and h do not change but k increases by 1, so the potential decreases by 1, as required.
Case 3: $r < n/3$. Note that $h + r + k = n$ and, since we are not yet balanced, $h, k \geq 1$. So $h + k > 2n/3$ gives $h \cdot k \geq h + k - 1 > 2n/3 - 1 > n/3$. We wait for some ball in an overloaded bin to choose an underloaded bin and move there (ignoring any other move via Lemma 2). There are h overloaded bins and, hence, $> h \cdot \varnothing$ such balls. The probability that such a ball, when activated, choose an underloaded bin is k/n. Hence the expected time for such a move to happen is at most $\frac{1}{h \cdot \varnothing} \cdot \frac{n}{k} \leq 3/\varnothing$. When such a move happens, the value of A decreases by 1, while the values of k, h can decrease by at most 1. So, the potential decreases by at least 1, as required.

■

C. Phase 3: Reaching Perfect Balance

Lemma 12. *Consider an initial configuration $\boldsymbol{\ell} = \boldsymbol{\ell}(0)$ with $\operatorname{disc}(\boldsymbol{\ell}) \leq 1$. Let $T := \inf\{\, t \mid \operatorname{disc}(\boldsymbol{\ell}(t)) < 1 \,\}$. Then $\mathbb{E}[T] = \mathrm{O}(n/\varnothing)$.*

Proof: Note that $\varnothing - 1 \leq \ell_{\min} \leq \ell_{\max} \leq \varnothing + 1$. By Lemma 2, we may ignore any movement of balls from bins with load exactly m/n. Suppose there are A bins of load $> \varnothing$, and so there are also A bins of load $< \varnothing$. If the configuration is not already balanced, then $A \geq 1$ and there are $> \varnothing \cdot A$ balls that, when activated, find a bin with load $< \varnothing$ with probability A/n. The expected time for the first such move to happen is at most $\frac{1}{A \cdot \varnothing} \cdot \frac{1}{A/n} = \frac{n}{\varnothing \cdot A^2}$. Since A decreases one by one until balancing out (and at that point we would have $A \geq 1$), the expected total time to balance out is at most $\sum_{A=1}^{n} \frac{n}{\varnothing \cdot A^2} \leq \mathrm{O}(n/\varnothing)$, as required. ■

VI. Conclusion

We analyzed the randomized local search protocol that was first introduced by [2] and showed that the protocol achieves perfect balance in expected time $\mathrm{O}\big(\ln(n) + n^2/m\big)$. Moreover, there is a matching lower bound so our analysis is tight. We now present a few possible future directions.

The first direction is to extend the analysis to the setting where the bins may have different speeds, and the load of a bin is defined as its number of balls divided by its speed. One can consider a similar protocol to RLS: a ball chooses a random bin on activation, and moves there if and only if doing so improves its load. A second direction is to study the protocol when the balls may have different weights. In particular, can we obtain similar balancing times in the weighted case as in the non-weighted case? The third direction is to analyze the protocol in network topologies other than the complete graph.

Appendix

This section gathers some simple results. The corresponding proofs are left for the full version.

A. General Auxiliary Results

Lemma 13 (Chernoff Bound, see [21, Theorem 4.4])**.** *Consider the binomial distribution $\mathrm{Bin}(n, p)$ with parameters $n \in \mathbb{N}$ and $p \in [0, 1]$. Then, for any $\varepsilon \in [0, 3/2]$ and $R \geq 6np$ we have*

$$\Pr(|\mathrm{Bin}(n,p) - np| > \varepsilon \cdot np) < 2e^{-\frac{\varepsilon^2 \cdot np}{3}} \quad \textit{and} \tag{7}$$

$$\Pr(\mathrm{Bin}(n,p) \geq R) \leq 2^{-R}. \tag{8}$$

Lemma 14 (Concentration: Sum of Independent Exponentials)**.** *Let X be a sum of independent exponential random variables, each having parameter $\geq \lambda$. Then $\forall \delta$*

$$\Pr(X \geq \mathbb{E}[X] + \delta) \leq e^{\lambda^2 \operatorname{Var}[X]/4 - \lambda\delta/2}. \tag{9}$$

Lemma 15 (Concentration: Sum of Independent geometric random variables)**.** *Let $Y_1, \ldots, Y_k$ be independent geometric random variables with parameter $p \in [0,1)$. Define $L := -\ln(1-p)$, and let $c_1, \ldots, c_k, M, S, V$ be positive constants satisfying $M := \max_i c_i$, $S \geq \sum_i c_i$, and $V \geq \sum_i c_i^2$. Then for any t we have*

$$\Pr\left(\sum_i c_i Y_i \geq t\right) \leq \exp\left(\frac{V}{4M^2} + \frac{S + SL - tL}{2M}\right). \quad (10)$$

Lemma 16. *Let $d_1 \leq d_2$ and suppose that, for any initial d_2-balanced configuration, the expected time to reach a d_1-balanced configuration is t. Then, for any d_2-balanced configuration, the time to reach a d_1-balanced configuration is at most $2t \log_2 n$ with high probability.*

Lemma 17. *Let $d_1 \leq d_2$ and suppose that, for any initial d_2-balanced configuration, the time to reach a d_1-balanced configuration is at most t with probability at least p. Let Y be a geometric random variable with parameter p. Then, for any d_2-balanced configuration, the time to reach a d_1-balanced configuration is stochastically dominated by tY, and so has expected value at most t/p.*

B. Auxiliary Results for section V

Next we give the missing calculations from Lemma 7.

Lemma 18. *Let $x_0 := \varnothing/2$ and $r = \log_2 \log_2 \varnothing$. Let $c_i := 16 \cdot \ln(n) \cdot x_0^{1/2^i}/\varnothing$ for $i = 0, 1, \ldots, r-1$. Then, we have $\max c_i = \mathrm{O}(\ln n)$, $\sum c_i = \mathrm{O}(\ln n)$, and $\sum c_i^2 = \mathrm{O}(\ln^2 n)$.*

Proof: We have $\max c_i = c_0 = 8 \cdot \ln n$. Also, note that for any $y \geq 1$ we have $\sum_{i=0}^{r-1} y^{1/2^i} \leq 2y + 4r$. Indeed, let k be the smallest integer such that $y^{1/2^k} < 4$. Then,

$$\sum_{i=0}^{r-1} y^{1/2^i} = \sum_{i=0}^{k} y^{1/2^i} + \sum_{i=k+1}^{r-1} y^{1/2^i} \leq \sum_{i=0}^{k} y/2^i + 4r < 2y + 4r.$$

Using this, we bound

$$\begin{aligned}\sum_{i=0}^{r-1} c_i &= \frac{16 \cdot \ln(n)}{\varnothing} \cdot \sum_{i=0}^{r-1} x_0^{1/2^i} \leq \frac{16 \cdot \ln(n)}{\varnothing} \cdot (2x_0 + 4r) \\ &= \frac{16 \cdot \ln(n)}{\varnothing} \cdot (\varnothing + 4 \log\log \varnothing) \leq 16 \cdot \ln(n) \cdot 2\end{aligned}$$

Finally, using a similar analysis we get

$$\begin{aligned}\sum_{i=0}^{r-1} c_i^2 &= \left(\frac{16 \cdot \ln(n)}{\varnothing}\right)^2 \cdot \sum_{i=0}^{r-1} x_0^{2/2^i} \\ &\leq \left(\frac{16 \cdot \ln(n)}{\varnothing}\right)^2 \cdot (2x_0^2 + 4r) \\ &= \left(\frac{16 \cdot \ln(n)}{\varnothing}\right)^2 \cdot \left(\frac{\varnothing^2}{2} + 4 \cdot \log\log \varnothing\right) \\ &\leq \left(\frac{16 \cdot \ln(n)}{\varnothing}\right)^2 \cdot \varnothing^2 = 256 \cdot (\ln n)^2,\end{aligned}$$

completing the proof. ∎

References

[1] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS)*. Springer-Verlag, 1999, pp. 404–413.

[2] P. W. Goldberg, "Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game," in *Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2004, pp. 131–140.

[3] A. Ganesh, S. Lilienthal, D. Manjunath, A. Proutiere, and F. Simatos, "Load balancing via random local search in closed and open systems," *Queueing Systems*, vol. 71, no. 3, pp. 321–345, 2012.

[4] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, "Load balancing in dynamic structured peer-to-peer systems," *Performance Evaluation*, vol. 63, no. 3, pp. 217–240, 2006.

[5] M. Petrova, N. Olano, and P. Mahonen, "Balls and bins distributed load balancing algorithm for channel allocation," in *Proceedings of the 7th International Conference on Wireless On-demand Network Systems and Services (WONS)*, 2010, pp. 25–30.

[6] J. E. Boillat, F. Brugé, and P. G. Kropf, "A dynamic load-balancing algorithm for molecular dynamics simulation on multi-processor systems," *Journal of Computational Physics*, vol. 96, no. 1, pp. 1–14, 1991.

[7] N. L. Johnson and S. Kotz, *Urn Models and Their Application: An Approach to Modern Discrete Probability Theory*, ser. Wiley Series in Probability and Mathematical Statistics. New York: Wiley, 1977.

[8] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, October 2001.

[9] A. Czumaj, "Recovery time of dynamic allocation processes," in *Proceedings of the 10th Symposium on Parallel Algorithms and Architectures (SPAA)*. ACM, 1998, pp. 202–211.

[10] P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, L. Nagel, and C. Wastell, "Self-stabilizing balls & bins in batches: The power of leaky bins [extended abstract]," in *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, ser. PODC '16. New York, NY, USA: ACM, 2016, pp. 83–92, available at http://arxiv.org/abs/1603.02188.

[11] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and G. Posta, "Self-stabilizing repeated balls-into-bins," in *Proceedings of the 27th Symposium on Parallelism in Algorithms and Architectures (SPAA)*, 2015, pp. 332–339.

[12] A. Czumaj, C. Riley, and C. Scheideler, "Perfectly balanced allocation," in *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 240–251.

[13] B. Vöcking, "Selfish load balancing," in *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds. Cambridge University Press, 2007.

[14] E. Even-Dar and Y. Mansour, "Fast convergence of selfish rerouting," in *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*. ACM, 2005, pp. 772–781.

[15] P. Berenbrink, T. Friedetzky, L. A. Goldberg, P. W. Goldberg, Z. Hu, and R. Martin, "Distributed selfish load balancing," *SIAM Journal on Computing*, vol. 37, no. 4, pp. 1163–1181, 2007.

[16] P. Berenbrink, T. Friedetzky, I. Hajirasouliha, and Z. Hu, "Convergence to equilibria in distributed, selfish reallocation processes with weighted tasks," *Algorithmica*, vol. 62, no. 3, pp. 767–786, 2012.

[17] H. Ackermann, S. Fischer, M. Hoefer, and M. Schöngens, "Distributed algorithms for qos load balancing," *Distributed Computing*, vol. 23, no. 5, pp. 321–330, 2011.

[18] M. Hoefer and T. Sauerwald, "Brief announcement: Threshold load balancing in networks," in *Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2013, pp. 54–56.

[19] ——, "Threshold load balancing in networks," *CoRR*, vol. abs/1306.1402, 2013.

[20] P. Berenbrink, T. Friedetzky, F. Mallmann-Trenn, S. Meshkinfamfard, and C. Wastell, "Threshold load balancing with weighted tasks," in *Proceedings of the 29th IEEE Parallel & Distributed Processing Symposium (IPDPS)*, 2015, pp. 550–558.

[21] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.