Recovery Time of Dynamic Allocation Processes *

Artur Czumaj
Heinz Nixdorf Institute and
Department of Mathematics & Computer Science
University of Paderborn
D-33095 Paderborn, Germany
(artur@uni-paderborn.de)

Abstract

Many distributed protocols arising in applications in on-line load balancing and dynamic resource allocation can be modeled by dynamic allocation processes related to the "balls into bin" problems. Traditionally the main focus of the research on dynamic allocation processes is on verifying whether a given process is stable, and if so, on analyzing its behavior in the limit (i.e., after sufficiently many steps). Once we know that the process is stable and we know its behavior in the limit, it is natural to analyze its *recovery time*, which is the time needed by the process to recover from any arbitrarily bad situation and to arrive very closely to a stable (i.e., a typical) state. This investigation is important to provide assurance that even if at some stage the process has reached a highly undesirable state, we can predict with high confidence its behavior after the estimated recovery time.

In this paper we present a *general framework to study the recovery time of discrete-time dynamic allocation processes*. We model allocation processes by suitably chosen ergodic Markov chains. For a given Markov chain we apply *path coupling* arguments to bound its convergence rates to the stationary distribution, which directly yields the estimation of the recovery time of the corresponding allocation process. Our coupling approach provides in a relatively simple way an accurate prediction of the recovery time. In particular, we show that our method can be applied to significantly improve estimations of the recovery time for various allocation processes related to allocations of balls into bins, and for the edge orientation problem studied before by Ajtai et al.

*Research partially supported by EU ESPRIT Long Term Research Project 20244 (ALCOM-IT) and by DFG-Sonderforschungsbereich 376 "Massive Parallelität: Algorithmen, Entwurfsmethoden, Anwendungen."

SPAA 98 Puerto Vallarta Mexico

1 Introduction

Many distributed scheduling protocols arising in applications in on-line load balancing and dynamic resource allocation can be modeled by stochastic processes describing the behavior of the system. In this paper we consider various such stochastic processes that are related to randomized allocating of balls into bins (cf. [1, 5, 6, 11, 15, 16, 21, 22, 24]).

Consider the classical problems of estimating the maximum load in processes sequentially allocating m balls into n bins, $m \geq n$. If each ball is placed into a bin chosen *i.u.r.* (*independently and uniformly at random*), then it is well-known that the maximum load in any bin is $\Theta(\frac{\ln n}{\ln\lceil 1+\frac{n}{m}\ln n\rceil} + \frac{m}{n})$ *w.h.p.*[1] Azar et al. [5] proposed a modification of this scheme, that we call ABKU[d], in which each ball is placed sequentially into the least full of d randomly chosen bins. They showed that for $d \geq 2$, the maximum load drops down to $\frac{\ln\ln n}{\ln d}(1+o(1)) + \Theta(\frac{m}{n})$ w.h.p.

In many applications, however, one rather is interested in *dynamic* versions of allocation processes. We start with m balls that are arbitrarily allocated into n bins, and then repeatedly either remove some ball or allocate a new ball. Azar et al. [5] analyzed a dynamic version of ABKU[d], that we call $I_{\mathcal{A}}$-ABKU[d], in which we repeatedly remove a random bin and then allocate a new bin according to the original static rule of ABKU[d]. Using complicated arguments, they showed that for $n = m$, $d \geq 2$, and independently of the input configuration of the balls, after $\mathcal{O}(n^3)$ steps the maximum load of the bins will be $\frac{\ln\ln n}{\ln d} + \mathcal{O}(1)$ w.h.p.

Recently Mitzenmacher [18, 22] presented a general framework to study the behavior of related processes, in both, static and dynamic scenarios. His approach was to first analyze infinite and continuous systems using differential equations. Then he related the behavior of these systems to finite discrete-time systems using Kurtz's work on density dependent jump Markov processes (see [18, 19, 20, 21, 22] for more details).

Using this framework, Mitzenmacher [18, 21, 22] derived bounds for the maximum load of ABKU[d] and (in the limit of) $I_{\mathcal{A}}$-ABKU[d], as well as for many other related processes. For example, he also analyzed (see Section 4.4.3 in [22]) the dynamic version of the process of Azar et al. [5], that we call $I_{\mathcal{B}}$-ABKU[d], in which we repeatedly remove a ball from a randomly chosen *nonempty bin* and then allocate a new

[1] Throughout the paper term w.h.p., which is the abbreviation of *with high probability*, is used to denote that a given event holds with probability $1 - \mathcal{O}(n^{-1})$, where n is a parameter measuring the input size.

ball using the static scheduling rule ABKU[d]. The approach of Mitzenmacher can be also used to derive bounds for the expected fairness in the edge orientation problem studied by Ajtai et al. [2].

Even if the technique of Mitzenmacher is very general and powerful, and provides (in a relatively simple way) tight bounds for the maximum load in the limit of many dynamic allocation processes, it tells nothing about the convergence rates to the predicted maximum load. For example, using Mitzenmacher's method one can show that both, $I_{\mathcal{A}}$-ABKU[d] and $I_{\mathcal{B}}$-ABKU[d], after sufficiently many steps will achieve maximum load $\frac{\ln\ln n}{\ln d}(1+o(1))$ w.h.p. But this technique does not tell anything about the number of steps required to obtain the desired maximum load. How fast can we predict the behavior of the system after the crash, when it reaches an undesirable state? How long does it take until the system recovers?

In this paper we consider dynamic ergodic stochastic processes on balls and bins in which, independently of the state at time τ_0 and for sufficiently large t, the process reaches a typical (predicted) maximum load (or other critical measure of the system) at time $\tau_0 + t$ w.h.p. Our aim is to estimate how small the number t can be taken, so that the desired maximum load (or other critical measure of the system) is achieved w.h.p. Thus, even if the system reaches an undesirable state (crashes), we want to bound the number of steps required by the system to recover and convert to a typical state.

The main aim of this paper is to provide a *general technique* to study the *recovery times*, i.e., the convergence rates to the predicted maximum load or other critical measure of the system. We demonstrate how the *path coupling* technique, a recent refinement of the classical coupling method due to Bubbley and Dyer [7], can be very efficiently used to estimate the recovery times. Already in [11] we applied a similar method to show that independently of the initial distribution of n balls in n bins, after $\mathcal{O}(n \ln n)$ steps of $I_{\mathcal{A}}$-ABKU[d] the maximum load in any bin will be $\frac{\ln\ln n}{\ln d} + \mathcal{O}(1)$ w.h.p. In this paper we simplify and significantly extend our analysis from [11]. We demonstrate our method on three classes of dynamic allocation processes.

We first analyze process $I_{\mathcal{A}}$-ABKU[d] of Azar et al. [5] and the dynamic version of the adaptive protocols investigated by Czumaj and Stemann [11], which are extensions of $I_{\mathcal{A}}$-ABKU[d]. We provide tight estimations of the recovery times of these processes. Our analysis *extends* and *simplifies* the analysis presented in [11] (in that paper a more complicated analysis of $I_{\mathcal{A}}$-ABKU[d] that uses classical coupling arguments (cf. [3]) has been presented). Further we study the dynamic process $I_{\mathcal{B}}$-ABKU[d] of Azar et al. [5] and its extension to adaptive processes. Although one may expect that the analysis of these processes should be very similar to that of $I_{\mathcal{A}}$-ABKU[d], we find these two scenarios of removing a ball very different. In particular, the processes in which in each step a ball removed is chosen at random (as in $I_{\mathcal{A}}$-ABKU[d]) seem to be much simpler to analyze than the processes in which we are removing balls by picking them from nonempty bins chosen at random (as in $I_{\mathcal{B}}$-ABKU[d]). Then, we apply our technique to estimate the recovery time of the edge orientation problem of Ajtai et al. [2], and obtain an almost optimal bound.

We also will briefly discuss possible extensions of our technique to open processes, in which the number of balls may vary during the run of the process, and to dynamic allocation processes in which relocation of resources (balls) is allowed (in a limited way) in each step.

All the previously known bounds for the recovery time of the aforementioned dynamic processes (except the analysis of $I_{\mathcal{A}}$-ABKU[d] in [11]) were missing the optimal bounds by superlinear terms. Our approach seems to be arguably simpler than those presented in the previous papers and also provide much tighter rates. For example, the recovery time for the edge orientation problem given in [2] was at least $\mathcal{O}(n^5)$; we improve it to $\mathcal{O}(n^2 \ln^2 n)$ and notice that it is $\Omega(n^2)$.

Our approach is to analyze dynamic allocation processes by considering them as suitable ergodic Markov chains. We define the state space in a way that allows us to follow the behavior of the Markov chain. Given that, our aim is to analyze the convergence rate of the Markov chain to its stationary distribution. Since this convergence is independent of the initial state, it directly yields the estimation for the recovery time of the underlying process.

The main technical tool we use in the analysis of Markov chains is the method of *coupling* and its recent refinement *path coupling*. The coupling arguments were widely used in applied probability (see, e g., [17]) and have already proved to be useful in theoretical computer science. Bubbley and Dyer [7] presented very recently a new, simple approach to the coupling method, called *path coupling*, that allowed them to extend applicability of coupling to many new problems (cf. [7, 8, 9, 10]). In this paper we show how to apply this technique to estimate the recovery times of dynamic allocation processes. As far as we know, we are the first who use coupling in this context.

We emphasize here that our technique cannot be used to estimate the maximum load (or other similar parameters) in the stationary distribution. Therefore, we believe that our technique is especially powerfully when applied together with the method of Mitzenmacher. His framework would be used to estimate the maximum load in any bin and our approach would be used to bound the recovery time and the number of steps required to achieve the desired maximum load with high probability.

1.1 Applications

The main motivation of our study was to analyze dynamic on-line randomized allocation schemes that arise from applications in on-line load balancing and dynamic resource allocation problems. We present here briefly two possible scenarios underlying our protocols.

Dynamic Resource Allocation Consider a non-centralized parallel dynamic system in which n jobs are assigned to identical and non-distinguishable n servers (or files stored in disks, etc.). In each time step one job is finishing and is removed from the system, and then one new job is to be assigned on-line. We consider two possible scenarios of removing the jobs — either one server, chosen i.u.r., has finished one job, or one job, chosen i.u.r., terminates [2]. In order to assign a new job in the system one possible strategy is to sample the load of $d \geq 2$ servers (chosen i.u.r.) in the system and then submitting the job to the least loaded server. The

[2] We believe that in applications to Dynamic Resource Allocation the first scenario seems to be more appropriated, but there are some related applications (e.g. in hashing) where the latter scenario could be applied.

result of Azar et al. [5] (see also [22]) implies that after sufficiently many steps the maximum load of the servers in each of these scenarios is $\frac{\ln \ln n}{\ln d} + \mathcal{O}(1)$ w.h.p. Our results provide more inside into these problems. We can show that, independently of the initial assignment of the jobs to the servers, we can recover from arbitrarily bad distribution of the jobs and obtain the maximum load of the servers $\frac{\ln \ln n}{\ln d} + \mathcal{O}(1)$ w.h.p. already after $\mathcal{O}(n^2 \ln n)$ steps in the first scenario, and after $\mathcal{O}(n \ln n)$ steps in the second scenario. The first bound is optimal up to a logarithmic factor and the second one is tight.

Fair Allocations Consider a distributed computer network with a centralized controller and client workstations issuing jobs. Each time a new job arrives some subset of servers is available (these might be either idle servers or servers that can execute given job). A scheduling protocol is seek for determining on which available server the job is to be executed. The main issue of our concern is to develop protocols that behaves *fair* for all the servers to encourage their continued work. Fagin and William [13] provided a natural definition of *fairness* (they consider this problem under the name the "carpool problem"). Ajtai et al. [2] showed that many problems in fairness of scheduling can be reduced (at the price of doubling the expected fairness) to the *edge orientation problem*. Therefore, by reduction of Ajtai et al., our analysis of the recovery time of the expected fairness in the edge orientation problem can be applied to the analysis of the fair allocation problem when the subset of servers available for each job has independent and uniform distribution (we refer to [2] and [13] for more details). In such a model, even if the system reaches an undesirable (or atypical) state, it recovers after $\mathcal{O}(n^2 \ln^2 n)$ next steps and arrives into a typical state w.h.p.

2 Random allocation processes

In this section we present the main ideas behind our approach. We first describe formally processes we analyze and then provide main ingredients of our analysis.

The first two general classes of allocation processes are defined as follows:

Scenario A Let m balls be arbitrarily placed in n bins. Repeat the following steps:

- Remove a ball chosen i.u.r. among the balls in the system.
- Place a new ball using given *scheduling rule*.

Scenario B Let m balls be arbitrarily placed in n bins. Repeat the following steps:

- Remove one ball from a non-empty bin chosen i.u.r.
- Place a new ball using given *scheduling rule*.

These two classes of dynamic allocation processes depend on the scheduling rules allocating new balls. The main motivation of our investigations was to study dynamic versions of random allocation processes analyzed by Azar et al. [5] and by Czumaj and Stemann [11].

Azar et al. [5] analyzed allocation processes governed by the following static scheduling rule, which we denote by ABKU[d]. Let $d \in \mathbb{N}$, $d \geq 1$.

ABKU[d]:

- Pick d bins i.u.r. (with replacements) and place the ball into the least full of the chosen bins.

Czumaj and Stemann [11] considered the following extension of ABKU[d], which we call ADAP($\mathbf{x}$). Let $\mathbf{x} = (x_0, x_1, \dots)$ be any nondecreasing sequence of positive integers.

ADAP($\mathbf{x}$):

Let $M = 1$.
repeat
  Choose a bin b_M i.u.r. from $[n]$.
  Let b be the bin of the minimum load among bins $\{b_1, \dots, b_M\}$ and let l be the load of l.
  if $x_l \leq M$, *then* place the ball into bin b_s.
  else $M = M + 1$.
until the ball is placed.

Using the definitions above we can define the dynamic versions of static processes of Azar et al., and of Czumaj and Stemann. If we apply the scheduling rule ABKU[d] (or ADAP($\mathbf{x}$)) to scenario A, then the obtained protocol will be called $\mathrm{I}_{\mathcal{A}}$-ABKU[d] (or $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$), respectively); if we apply the scheduling rule ABKU[d] (or ADAP($\mathbf{x}$)) to scenario B, then the obtained protocol will be called $\mathrm{I}_{\mathcal{B}}$-ABKU[d] (or $\mathrm{I}_{\mathcal{B}}$-ADAP($\mathbf{x}$), respectively).

Edge Orientation Problem
The following problem was investigated by Ajtai et al. [2].

Let G be a directed edge-less multigraph on a set of n vertices. New edges (defined initially as undirected pairs of distinct vertices) arrive one by one, and each edge is added to the multigraph by orienting upon its arrival. The goal is to design a strategy of orienting the edges so that in every vertex at every moment of time the difference between the indegree and outdegree is as small as possible. This is measured by the *unfairness* at any time, defined to be the maximum over vertices, of the (absolute value of the) difference between indegree and outdegree. We are studying the model in which the edges are arriving *with uniform distribution*, i.e., in each step an independently and uniformly chosen undirected edge is arriving. For this model we analyze the so-called *greedy* protocol that can be defined formally as follows:

Let G be a directed edge-less multigraph on a set of n vertices. Repeat the following step:

- Pick two distinct vertices i.u.r. and add to G an edge between these two vertices that is oriented from the vertex with the smaller difference between the outdegree and indegree to the one with the larger difference.

Azar et al. [2] (see also Section 4.4.6. in [22]) proved that the greedy protocol behaves very well in this model, and independently of the current situation, after sufficiently many steps it achieves the expected unfairness of $\Theta(\log \log n)$.

2.1 General approach

In this paper we shall study the *recovery times* of the aforementioned processes. By recovery time we mean the number of steps required (w.h.p.) by the process to move from an arbitrary state to a typical state. We describe each of the processes above by a suitably defined ergodic Markov chain (see Section 3.3). For such a Markov chain the recovery time is exactly the *mixing time* of the chain. Our main contribution is the use of *path coupling*, which is a recent refinement

of the classical *coupling* arguments (see [3] and Section 3), to provide in a simple way accurate bounds for the mixing times of the aforementioned random allocation processes.

In the path coupling technique the main aim is to consider two copies of the Markov chain at hand and "couple" them so that the copies will be the same after as few steps as possible. Then the Path Coupling Lemma (see Section 3) can be used to provide a bound for the mixing time of the Markov chain.

3 Technical preliminaries

Let $\mathbb{N} = \{0, 1, \dots\}$ and let $[k]$ denote the set $\{1, \dots, k\}$. For any random variable X its probability distribution is denoted by $\mathcal{L}(X)$. For any real n-vector $\mathbf{v} = (v_1, \dots, v_n)$ we use standard notation $\|\mathbf{v}\|_1 = \sum_{i=1}^{n} |v_i|$.

Let $\mathfrak{M} = (\mathcal{M}_t)_{t\in\mathbb{N}}$ be a discrete-time Markov chain with a finite state space $\mathcal{X}$. It is well known (see, e.g., [23]) that if $\mathfrak{M}$ is ergodic (i.e., irreducible and aperiodic) then it has a unique *stationary distribution* π, i.e., for all $x \in \mathcal{X}$: $\pi(x) = \lim_{t\to\infty} \mathbf{Pr}[\mathcal{M}_t = x | \mathcal{M}_0 = y]$ independently of $y \in \mathcal{X}$. One important issue we shall study in this paper is the rate of convergence of ergodic Markov chains, i.e., how many steps one has to run $\mathfrak{M}$ so that the distance between π and $\mathbf{Pr}[\mathcal{M}_t = \cdot \,| \mathcal{M}_0 = y]$ will be very small (see, e.g., [12, 14, 23]).

A standard measure of the separation between two probability distributions is the *variation distance*. For any two random variables X and Y defined jointly on the same space, the *variation distance* between $\mathcal{L}(X)$ and $\mathcal{L}(Y)$ is defined as

$$\|\mathcal{L}(X) - \mathcal{L}(Y)\| = \sup_A |\mathbf{Pr}[X \in A] - \mathbf{Pr}[Y \in A]| \ .$$

The standard measure of the convergence of an ergodic Markov chain $\mathfrak{M}$ to its stationary distribution is the *mixing time* of $\mathfrak{M}$, denoted by $\tau_{\mathfrak{M}}(\varepsilon)$, which is defined as

$$\tau_{\mathfrak{M}}(\varepsilon) = \min\{T \in \mathbb{N} : \forall t \geq T \max_{x\in\mathcal{X}} \|\mathcal{L}(\mathcal{M}_t | \mathcal{M}_0 = x) - \pi\| \leq \varepsilon\} \ .$$

The main technical tool we shall use in the paper is the *coupling* technique (cf. [3, 12, 17]).

Definition 3.1 Let $\mathfrak{M}$ be a discrete-time Markov chain with a finite state space $\mathcal{X}$. A *coupling* $(X_t, Y_t)_{t\in\mathbb{N}}$ for $\mathfrak{M}$ is a discrete-time Markov chain on $\mathcal{X} \times \mathcal{X}$ such that each of $(X_t)_{t\in\mathbb{N}}$, $(Y_t)_{t\in\mathbb{N}}$, considered independently, is a faithful copy of $\mathfrak{M}$, i.e.,

$$\begin{aligned} \mathcal{L}(X_t) &= \mathcal{L}(\mathcal{M}_t | \mathcal{L}(\mathcal{M}_0) = \mathcal{L}(X_0)) \text{ and} \\ \mathcal{L}(Y_t) &= \mathcal{L}(\mathcal{M}^*_t | \mathcal{L}(\mathcal{M}^*_0) = \mathcal{L}(Y_0)) \ . \end{aligned}$$

Path Coupling We shall use a recent refinement of the coupling method due to Bubley and Dyer [7] (cf. also [8, 9]). The main difficulty in applying coupling lies in analyzing processes $(X_t, Y_t)_{t\in\mathbb{N}}$ on the whole product space $\mathcal{X} \times \mathcal{X}$. The *path coupling* method allows to consider a coupling only for a subset of $\mathcal{X} \times \mathcal{X}$.

Lemma 3.1 (Path Coupling Lemma [7]) Let Δ be an integer valued metric defined on $\mathcal{X} \times \mathcal{X}$ which takes values in $\{0, \dots, D\}$. Let Γ be any subset of $\{(X, Y) \in \mathcal{X} \times \mathcal{X}\}$. Suppose that for every $(X, Y) \in \mathcal{X} \times \mathcal{X}$ there exists a sequence $X = Z_0, Z_1, \dots, Z_{r-1}, Z_r = Y$, where $(Z_i, Z_{i+1}) \in \Gamma$ for all $0 \leq i < r$, and $\sum_{i=0}^{r-1} \Delta(Z_i, Z_{i+1}) = \Delta(X, Y)$. Suppose also that there exists a coupling $(X_t, Y_t)_{t\in\mathbb{N}}$ for $\mathfrak{M}$ such that for some positive real β we have $\mathbf{E}[\Delta(X_{t+1}, Y_{t+1})] \leq \beta \cdot \Delta(X_t, Y_t)$ for all $(X_t, Y_t) \in \Gamma$.

(1) If $\beta < 1$, then $\tau_{\mathfrak{M}}(\varepsilon) \leq \left\lceil \frac{\ln(D\cdot\varepsilon^{-1})}{\ln \beta^{-1}} \right\rceil$.

(2) If $\beta \leq 1$ and $\mathbf{Pr}[\Delta(X_{t+1}, Y_{t+1}) \neq 1] \geq \alpha$ for all $(X_t, Y_t) \in \Gamma$ and for some $\alpha > 0$, then $\tau_{\mathfrak{M}}(\varepsilon) \leq \left\lceil \frac{e\cdot D^2}{\alpha} \right\rceil \cdot \lceil \ln \varepsilon^{-1} \rceil$.

We emphasize here that the main power of the use of path coupling is that we need define the coupling *only* for pairs $(X_t, Y_t) \in \Gamma$.

3.1 Normalized load vectors

Fix n. For an integer vector $\mathbf{v} = (v_1, \dots, v_n)$, let $\pi_{\mathbf{v}}$ be a permutation of n elements such that $v_{\pi_{\mathbf{v}}(1)} \geq v_{\pi_{\mathbf{v}}(2)} \geq \cdots \geq v_{\pi_{\mathbf{v}}(n)}$. If $v_1 \geq v_2 \geq \cdots \geq v_n$ (i.e. $\pi_{\mathbf{v}}$ may be chosen to be the identity permutation) then we say $\mathbf{v}$ is *normalized*. The operation of applying $\pi_{\mathbf{v}}$ to $\mathbf{v}$ so that the resulted vector $\mathbf{v}^*$ is normalized is called *normalization of* $\mathbf{v}$. We shall also call $\mathbf{v}^*$ the *normalized* vector of $\mathbf{v}$.

We denote by Ω_m the set of all *non-negative* normalized n-vectors $\mathbf{v}$ satisfying $\|\mathbf{v}\|_1 = m$ and we let $\Omega = \bigcup_{m\in\mathbb{N}} \Omega_m$.

Let $\mathbf{e}_i$ be the n-vector $\mathbf{e}_i = (e_{i,1}, e_{i,2}, \dots, e_{i,n})$ such that $e_{i,j} = \delta_{i,j}$, and let $\mathbf{0} = (0, \dots, 0) \in \Omega_0$. For every $i \in [n]$, we write $\mathbf{v} \oplus \mathbf{e}_i$ to denote the normalized vector of $\mathbf{v} + \mathbf{e}_i$. Similarly, $\mathbf{v} \ominus \mathbf{e}_i$ denotes the normalized vector of $\mathbf{v} - \mathbf{e}_i$.

We shall frequently use the following simple fact.

Fact 3.2 Let $\mathbf{v} \in \Omega$, $i \in [n]$, $j = \min\{t : v_t = v_i\}$, and $s = \max\{t : v_t = v_i\}$. Then

$$\mathbf{v} \oplus \mathbf{e}_i = \mathbf{v} + \mathbf{e}_j \quad \text{and} \quad \mathbf{v} \ominus \mathbf{e}_i = \mathbf{v} - \mathbf{e}_s \ .$$

3.2 Probability distributions and right-oriented random functions

In this paper we shall use three special probability distributions.

Definition 3.2 For any $m \in \mathbb{N}$ and any nonnegative integer n-vector $\mathbf{v} \in \Omega_m$, we let $\mathcal{A}(\mathbf{v})$ be the probability distribution on $[n]$ such that $\mathbf{Pr}[\mathcal{A}(\mathbf{v}) = i] = \frac{v_i}{m}$ for every $i \in [n]$.

Definition 3.3 For any n-vector $\mathbf{v} \in \Omega$ with $s = \max\{i : v_i > 0\}$, we let $\mathcal{B}(\mathbf{v})$ be the probability distribution on $[n]$ such that $\mathbf{Pr}[\mathcal{B}(\mathbf{v}) = i] = \frac{1}{s}$ for every $i \in [s]$.

Throughout the paper we let $\mathcal{D}$ denote a *random function* from Ω to $[n]$. Informally, one may think on $\mathcal{D}$ as on the set of probability distributions with domain $[n]$ defined for every $\mathbf{v} \in \Omega$. Formally, we define $\mathcal{D}$ as follows. Let $\mathcal{RS}$ be an arbitrary set and let $\mathbb{RS}$ be a random function that returns $\mathrm{rs} \in \mathcal{RS}$ according to some distribution. Let $\overline{\mathcal{D}} : \Omega \times \mathcal{RS} \to [n]$ be a deterministic function. Then we define $\mathcal{D}(v) = \overline{\mathcal{D}}(v, \mathbb{RS})$. Observe that randomness of $\mathcal{D}$ depends only on $\mathbb{RS}$. From now on, when discussing about a random function $\mathcal{D}$ we shall implicitly mean the quadruple $(\mathcal{RS}, \mathbb{RS}, \overline{\mathcal{D}}, \mathcal{D})$ as above.

Definition 3.4 We say a random function $\mathcal{D} = (\mathcal{RS}, \mathbb{RS}, \overline{\mathcal{D}}, \mathcal{D})$ from Ω to $[n]$ is *right-oriented* if there exists a permutation $\Phi_{\mathcal{D}} : \mathcal{RS} \to \mathcal{RS}$ such that for every $\mathrm{rs} \in \mathcal{RS}$, $m \in \mathbb{N}$, $i \in [n]$, $\mathbf{v}, \mathbf{u} \in \Omega_m$:

if $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) = i < \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ then $v_i > u_i$,
if $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) > i = \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ then $v_i < u_i$.

The following lemma is key for our applications of right-oriented functions.

Lemma 3.3 Let $\mathcal{D} = (\mathcal{RS}, \mathbb{RS}, \overline{\mathcal{D}}, \mathcal{D})$ be a right-oriented random function from Ω to $[n]$ and let $\Phi_{\mathcal{D}}$ be the permutation from Definition 3.4. Let $\mathbf{v}, \mathbf{u} \in \Omega_m$. Let rs be the value returned by $\mathbb{RS}$. If we set $\mathbf{v}^\diamond = \mathbf{v} \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{v}^*, \mathrm{rs})}$ and $\mathbf{u}^\diamond = \mathbf{u} \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{u}^*, \Phi_{\mathcal{D}}(\mathrm{rs}))}$, then $\|\mathbf{v} - \mathbf{u}\|_1 \geq \|\mathbf{v}^\diamond - \mathbf{u}^\diamond\|_1$.

Proof : Fix $\mathrm{rs} \in \mathcal{RS}$, the value returned by $\mathbb{RS}$. Let $\lambda = \min\{j : v_t = v_{\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs})}\}$ and $\delta = \min\{j : u_t = u_{\overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))}\}$. Then $\mathbf{v}^\diamond = \mathbf{v} + \mathbf{e}_\lambda$, $\mathbf{u}^\diamond = \mathbf{u} + \mathbf{e}_\delta$, and $\mathbf{w}^\diamond = \mathbf{w} + \mathbf{e}_\lambda - \mathbf{e}_\delta$. Therefore $\Delta(\mathbf{w}^\diamond) > \Delta(\mathbf{w})$ would imply that $w_\lambda \geq 0 \geq w_\delta$. We show that this cannot be the case unless $\lambda = \delta$ (in which case $\mathbf{w}^\diamond = \mathbf{w}$).

We consider five cases.

- If $\lambda = \delta$ then clearly $\mathbf{w}^\diamond = \mathbf{w}$, and hence $\Delta(\mathbf{w}^\diamond) = \Delta(\mathbf{w})$.
- If $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) = \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ and $\lambda < \delta$ then $v_\lambda = v_\delta$ and $u_\lambda > u_\delta$, and hence $w_\lambda < w_\delta$.
- If $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) = \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ and $\lambda > \delta$ then $v_\lambda < v_\delta$ and $u_\lambda = u_\delta$, and hence $w_\lambda < w_\delta$.
- If $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) < \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ then $v_{\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs})} < u_{\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs})}$ because $\mathcal{D}$ is right-oriented. Thus $v_\lambda = v_{\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs})} < u_{\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs})} \leq u_\lambda$, and hence $w_\lambda < 0$.
- If $\overline{\mathcal{D}}(\mathbf{v}, \mathrm{rs}) > \overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))$ then $v_{\overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))} > u_{\overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))}$ because $\mathcal{D}$ is right-oriented. Thus $v_\delta \geq v_{\overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))} > u_{\overline{\mathcal{D}}(\mathbf{u}, \Phi_{\mathcal{D}}(\mathrm{rs}))} = u_\delta$, and hence $0 < w_\delta$.

Thus in any case $\lambda = \delta$, or $w_\lambda < w_\delta$, or $w_\lambda < 0$, or $0 < w_\delta$. □

3.3 Underlying Markov chains

Consider a dynamic allocation process in which a "phase" consists of removing a ball according to some (usually randomized) protocol and then allocating a new ball according to a given (usually randomized) scheduling rule. Suppose that initially m balls are allocated (arbitrarily) in n bins. We model the state of the system by normalized n-vectors $\mathbf{v} \in \Omega_m$. We say we are at state $\mathbf{v}$ if there exists a permutation π such that v_i represents the current load of bin $\pi(i)$. Vector $\mathbf{v}$ will be called the *load vector*. Note that a load vector contains all relevant information about the state of the algorithm: when we are analyzing the values v_i, the ordering of bins is insignificant.

A dynamic random allocation process will we modeled by the underlying Markov chain $\mathfrak{M}$ having state space Ω_m of load vectors. The transitions probabilities of $\mathfrak{M}$ are dependent on the scheduling rule used by the allocation process.

Consider for example scenario A for $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$) and let $\mathbf{v}$ be the corresponding load vector. Observe that since $\mathbf{v}$ is normalized, we have $v_1 \geq v_2 \geq \cdots \geq v_n$.

We first remove from the system a ball chosen i.u.r. This corresponds to decreasing some v_i by one. We can do this as follows: Choose a number $i \in [n]$ independently at random according to the probability distribution $\mathcal{A}(\mathbf{v})$, i.e., such that $\mathbf{Pr}[i = j] = \frac{v_j}{m}$ for every $j \in [n]$. Then decrease v_i by one and normalize the resulting vector.

Now we place a new ball to the system according to the ADAP($\mathbf{x}$) rule. This corresponds to increasing some value v_j by one in the following way. Let $\mathcal{RS}$ be the set of all sequences $(b_1, b_2, \dots)$ with $b_i \in [n]$ and let $\mathbb{RS}$ be a random function that returns an element of $\mathcal{RS}$ i.u.r. For a given sequence $\mathbf{b} = (b_1, b_2, \dots) \in \mathcal{RS}$, let $p(\mathbf{b})_i = \max\{b_j : 1 \leq j \leq i\}$. Let $\mathcal{D}$ be a random function from Ω to $[n]$ such that

$$\mathcal{D}(\mathbf{v}) = \overline{\mathcal{D}}(\mathbf{v}, \mathbf{b}) = p(\mathbf{b})_j \text{ with } j = \min\{t \in \mathbb{N} : x_{v_{p(\mathbf{b})_t}} \leq t\}. \quad (1)$$

Now, choose a number $j = \mathcal{D}(\mathbf{v})$, increase v_j by one, and normalize the resulting vector.

It is easy to see that this procedure mimics the behavior of $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$). In a similar way we can define protocol $\mathrm{I}_{\mathcal{B}}$-ADAP($\mathbf{x}$); the only difference is the use of the probability distribution $\mathcal{B}(\cdot)$ rather than $\mathcal{A}(\cdot)$.

In this paper we focus our attention on dynamic processes following scenarios A or B, in which each new ball added to the system is allocated using a right-oriented random function. To show that our analysis can be also applied to $\mathrm{I}_{\mathcal{A}}$-ABKU[d], $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$), $\mathrm{I}_{\mathcal{B}}$-ABKU[d], and $\mathrm{I}_{\mathcal{B}}$-ADAP($\mathbf{x}$), we prove now the following lemma.

Lemma 3.4 If $\mathcal{D}$ is defined as in formula (1), then $\mathcal{D}$ is right-oriented.

Proof : We show that $\mathcal{D}$ is right-oriented with $\Phi_{\mathcal{D}}$ being the identity permutation. Let $i \in [n]$, $m \in \mathbb{N}$, $\mathbf{v}, \mathbf{u} \in \Omega_m$. Suppose that $\overline{\mathcal{D}}(\mathbf{v}, \mathbf{b}) = i < \overline{\mathcal{D}}(\mathbf{u}, \mathbf{b})$ for some $\mathbf{b} = (b_1, b_2, \dots) \in \mathcal{RS}$. Then let $\overline{\mathcal{D}}(\mathbf{v}, \mathbf{b}) = p(\mathbf{b})_t = b_j < b_s = \overline{\mathcal{D}}(\mathbf{u}, \mathbf{b})$, where $t = \min\{l \in \mathbb{N} : x_{v_{p(\mathbf{b})_l}} \leq l\}$. Since $\overline{\mathcal{D}}(\mathbf{u}, \mathbf{b}) \neq b_j$ and $j < s$ (otherwise it would be $p(\mathbf{b})_t = b_s$), we get that $x_{v_{b_j}} \leq t < x_{u_{b_j}}$. Since the sequence $x_0, x_1, \dots$ is nondecreasing, this implies that $v_{b_j} < u_{b_j}$, what completes the proof that $\mathcal{D}$ is right-oriented. □

Let us notice here that since ABKU[d] is a special case of ADAP($\mathbf{x}$), Lemma 3.4 implies that the function $\mathcal{D}$ used in the definition of each of $\mathrm{I}_{\mathcal{A}}$-ABKU[d], $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$), $\mathrm{I}_{\mathcal{B}}$-ABKU[d], and $\mathrm{I}_{\mathcal{B}}$-ADAP($\mathbf{x}$) is right-oriented. Therefore, every claim about the processes in scenario A (or B, respectively) that holds for any right-oriented probability distribution for inserting new balls, can be directly applied to $\mathrm{I}_{\mathcal{A}}$-ABKU[d] and $\mathrm{I}_{\mathcal{A}}$-ADAP($\mathbf{x}$) ($\mathrm{I}_{\mathcal{B}}$-ABKU[d] and $\mathrm{I}_{\mathcal{B}}$-ADAP($\mathbf{x}$), respectively).

4 Recovery times of processes in scenario A

In this section we present a tight estimation of the recovery time of allocation processes in scenario A that uses right-oriented functions to allocate new balls. Our analysis in this section uses the path coupling technique and it significantly simplifies and extends the arguments used in [11] to bound the mixing time of $\mathrm{I}_{\mathcal{A}}$-ABKU[d].

Throughout this section we assume that $\mathcal{D} = (\mathcal{RS}, \mathbb{RS}, \overline{\mathcal{D}}, \mathcal{D})$ is right-oriented. Consider a Markov chain $\mathfrak{M}$ with state space Ω_m and transitions $\mathbf{v} \mapsto \mathbf{v}^\diamond$ defined by the following randomized procedure, which we call $\mathrm{I}_{\mathcal{A}}$:

- Choose $i \in [n]$ at random according to $\mathcal{A}(\mathbf{v})$.
- Set $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_i$.
- Let rs be the value returned by $\mathbb{RS}$.
- Set $\mathbf{v}^\diamond = \mathbf{v}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{v}^*, \mathrm{rs})}$.

We apply the Path Coupling Lemma to bound the mixing time (and thus the recovery time) of $\mathfrak{M}$. Let $\mathbf{v}, \mathbf{u} \in \Omega_m$. We define the distance between $\mathbf{v}, \mathbf{u}$, denoted by $\Delta(\mathbf{v}, \mathbf{u})$, as $\Delta(\mathbf{v}, \mathbf{u}) = \frac{1}{2}\|\mathbf{v} - \mathbf{u}\|_1 = \sum_{i=1}^{n} \max\{v_i - u_i, 0\}$. Notice that $\Delta(\mathbf{v}, \mathbf{u}) \leq m - \lceil m/n \rceil$ for every $\mathbf{v}, \mathbf{u} \in \Omega_m$. We define also set $\Gamma = \{(\mathbf{v}, \mathbf{u}) \in \Omega_m \times \Omega_m : \Delta(\mathbf{v}, \mathbf{u}) = 1\}$.

Now we define a coupling $(\mathbf{v}, \mathbf{u}) \mapsto (\mathbf{v}^\diamond, \mathbf{u}^\diamond)$ for all $\mathbf{v}$ and $\mathbf{u}$ with $\Delta(\mathbf{v}, \mathbf{u}) = 1$. Let $\mathbf{v}, \mathbf{u} \in \Omega_m$ such that $\Delta(\mathbf{v}, \mathbf{u}) = 1$. Then $\mathbf{v} = \mathbf{u} + \mathbf{e}_\lambda - \mathbf{e}_\delta$ for some $\lambda \neq \delta$. We assume without loss of generality that $\lambda < \delta$.

Coupling for $\mathbf{v}, \mathbf{u} \in \Omega_m$

Let us first pick a number $i \in [n]$ according to the distribution $\mathcal{A}(\mathbf{v})$. Now we set $j \in [n]$ as follows:

- if $i \neq \lambda$ then $j = i$;
- if $i = \lambda$ then set $j = \delta$ with probability $\frac{1}{v_\lambda}$, and set $j = i$ with the remaining probability $1 - \frac{1}{v_\lambda}$.

We set now $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_i$ and $\mathbf{u}^* = \mathbf{u} \ominus \mathbf{e}_j$.
Then we place a new ball using the coupling from Lemma 3.3: Let $\Phi_{\mathcal{D}}$ be the permutation from Definition 3.4 and let rs be the value returned by $\mathbb{RS}$. Set $\mathbf{v}^\diamond = \mathbf{v}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{v}^*, \mathrm{rs})}$ and $\mathbf{u}^\diamond = \mathbf{u}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{u}^*, \Phi_{\mathcal{D}}(\mathrm{rs}))}$.

Analysis of the coupling

It is easy to see that the transitions in the coupling described above are performed according to $\mathfrak{M}$, and hence $(\mathbf{v}, \mathbf{u}) \mapsto (\mathbf{v}^\diamond, \mathbf{u}^\diamond)$ is properly defined. The following key lemma shows how the distance between two vectors can change after applying the coupling.

Lemma 4.1 For every $\mathbf{v}, \mathbf{u} \in \Omega_m$ with $\Delta(\mathbf{v}, \mathbf{u}) = 1$ we have $\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond) \leq \Delta(\mathbf{v}, \mathbf{u})$. Moreover, if $i \neq j$ in the coupling, then $\mathbf{v}^\diamond = \mathbf{u}^\diamond$.

Proof: Let $\alpha = \max\{s \geq i : v_s = v_i\}$ and $\beta = \max\{s \geq j : u_s = u_j\}$. Then, by Fact 3.2, $\mathbf{v}^* = \mathbf{v} - \mathbf{e}_\alpha$ and $\mathbf{u}^* = \mathbf{u} - \mathbf{e}_\delta$. We consider two cases: $i = j$ and $i \neq j$.

If $i = j$ and $\alpha = \beta$, then it is easy to see that $\Delta(\mathbf{v}^*, \mathbf{u}^*) = 1$. Otherwise, if $i = j$ and $\alpha \neq \beta$, then either $\alpha = \lambda$ and $\beta = \lambda - 1$, or $\alpha = \delta - 1$ and $\beta = \delta$. In the first case we have $\mathbf{v}^* = \mathbf{v} - \mathbf{e}_\lambda = \mathbf{u} - \mathbf{e}_\delta = \mathbf{u}^* + \mathbf{e}_{\lambda-1} - \mathbf{e}_\delta$, and in the second case we obtain $\mathbf{v}^* = \mathbf{v} - \mathbf{e}_{\delta-1} = \mathbf{u} + \mathbf{e}_\lambda - \mathbf{e}_\delta - \mathbf{e}_{\delta-1} = \mathbf{u}^* + \mathbf{e}_\lambda - \mathbf{e}_{\delta-1}$. Therefore $\Delta(\mathbf{v}^*, \mathbf{u}^*) \leq 1$.

Now let us consider the case $i \neq j$. Then we must have $i = \lambda$ and $j = \delta$. Therefore we obtain $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_\lambda = \mathbf{v} - \mathbf{e}_\lambda = \mathbf{u} - \mathbf{e}_\delta = \mathbf{u} \ominus \mathbf{e}_\delta = \mathbf{u}^*$.

Thus we have shown so far that $\Delta(\mathbf{v}^*, \mathbf{u}^*) \leq 1$ and that if $i \neq j$ in the coupling, then $\mathbf{v}^* = \mathbf{u}^*$. To complete the proof apply the bound above and observe that $\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond) \leq \Delta(\mathbf{v}^*, \mathbf{u}^*)$ by Lemma 3.4. □

Lemma 4.1 directly implies the following.

Corollary 4.2 For every $\mathbf{v}, \mathbf{u} \in \Omega_m$ with $\Delta(\mathbf{v}, \mathbf{u}) = 1$ we have $\mathbf{E}[\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond)] \leq 1 - \frac{1}{m}$.

Now we can apply the Path Coupling Lemma with $D = m$ and $\beta = 1 - \frac{1}{m}$ to obtain the following theorem.

Theorem 1 For Markov chains $\mathfrak{M}$ defined by protocol $\mathrm{I}_{\mathcal{A}}$ we have $\tau_{\mathfrak{M}}(\varepsilon) = \lceil m \cdot \ln(m \cdot \varepsilon^{-1}) \rceil$.

One can also show (by considering two vectors $\mathbf{v}(0) = m \cdot \mathbf{e}_1$ and $\mathbf{u}(0)$ with $u(0)_1 - u(0)_n \leq 1$) that under some natural assumptions concerning the random right-oriented function $\mathcal{D}$ (that are satisfied by ABKU[d] and ADAP(x)), the bound above is tight up to lower-order terms.

5 Simple estimations of recovery times of processes in scenario B

In this section we present a simple scheme that enables a rough estimation of the recovery times of allocation processes in scenario B that use right-oriented functions to allocate new balls. Our analysis in this section uses the path coupling technique.

Throughout this section we assume that $\mathcal{D} = (\mathcal{RS}, \mathbb{RS}, \overline{\mathcal{D}}, \mathcal{D})$ is right-oriented. We consider a Markov chain $\mathfrak{M}$ with state space Ω_m and transitions $\mathbf{v} \mapsto \mathbf{v}^\diamond$ defined by the following randomized procedure, which we call $\mathrm{I}_{\mathcal{B}}$:

- Choose $i \in [n]$ at random according to $\mathcal{B}(\mathbf{v})$.
- Set $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_i$.
- Let rs be the value returned by $\mathbb{RS}$.
- Set $\mathbf{v}^\diamond = \mathbf{v}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{v}^*, \mathrm{rs})}$.

Below we present a simple proof of an $\mathcal{O}(nm^2 \ln \varepsilon^{-1})$ bound for the mixing time of $\mathfrak{M}$.

We apply the Path Coupling Lemma to bound the mixing time of $\mathfrak{M}$. For any $\mathbf{v}, \mathbf{u} \in \Omega_m$, let us define the distance between $\mathbf{v}$ and $\mathbf{u}$, denoted by $\Delta(\mathbf{v}, \mathbf{u})$, as $\Delta(\mathbf{v}, \mathbf{u}) = \frac{1}{2}\|\mathbf{v} - \mathbf{u}\|_1 = \sum_{i=1}^{n} \max\{v_i - u_i, 0\}$. Notice that $\Delta(\mathbf{v}, \mathbf{u}) \leq m - \lceil m/n \rceil$ for every $\mathbf{v}, \mathbf{u} \in \Omega_m$. Let us also define the set $\Gamma = \{(\mathbf{v}, \mathbf{u}) \in \Omega_m \times \Omega_m : \Delta(\mathbf{v}, \mathbf{u}) = 1\}$.

Let $\mathbf{v}$ and $\mathbf{u}$ be normalized vectors in Ω_m that are at distance 1 apart, i.e., there are $1 \leq \lambda, \delta \leq n$ such that $\mathbf{v} = \mathbf{u} + \mathbf{e}_\lambda - \mathbf{e}_\delta$. We assume, without loss of generality, that $\lambda < \delta$. Our aim is to define a coupling $(\mathbf{v}, \mathbf{u}) \mapsto (\mathbf{v}^\diamond, \mathbf{u}^\diamond)$ for all $\mathbf{v}$ and $\mathbf{u}$ with $\Delta(\mathbf{v}, \mathbf{u}) = 1$ such that $\mathbf{E}[\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond)] \leq 1$ and $\mathbf{Pr}[\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond) \neq 1] \geq \frac{1}{n}$.

Let $s_1 = \max\{k : v_k > 0\}$ and $s_2 = \max\{k : u_k > 0\}$. We first describe the coupling defining $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_i$ and $\mathbf{u}^* = \mathbf{u} \ominus \mathbf{e}_{i^*}$, such that i is uniformly distributed in $[s_1]$ and i^* is uniformly distributed in $[s_2]$. We consider separately two cases depending on whether $s_1 = s_2$ or not.

(i) $s_1 = s_2$

We first pick $i \in [s_1]$ i.u.r. and then set $i^* = \begin{cases} \delta & \text{if } i = \lambda \\ \lambda & \text{if } i = \delta \\ i & \text{otherwise} \end{cases}$.

It is easy to see that this is a well defined coupling. The following simple claim will be used to estimate the mixing time of $\mathfrak{M}$.

Claim 5.1 If $s_1 = s_2$ then

$$\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = \begin{cases} 0 & \text{if } i = \lambda \\ 2 & \text{if } i = \delta \\ 1 & \text{otherwise} \end{cases}.$$

Proof: We consider three cases depending on whether $i = \lambda$, $i = \delta$, or $i \neq \lambda, \delta$.

- Let $i = \lambda$ and $i^* = \delta$. Notice that $v_\lambda > v_{\lambda+1}$ and either $\delta = n$ or $u_\delta > u_{\delta+1}$. Therefore $\mathbf{v} \ominus \mathbf{e}_\lambda = \mathbf{v} - \mathbf{e}_\lambda$ and $\mathbf{u} \ominus \mathbf{e}_\delta = \mathbf{u} - \mathbf{e}_\delta$. Hence we obtain $\mathbf{v} \ominus \mathbf{e}_\lambda = (\mathbf{u} + \mathbf{e}_\lambda - \mathbf{e}_\delta) - \mathbf{e}_\lambda = \mathbf{u} - \mathbf{e}_\delta = \mathbf{u} \ominus \mathbf{e}_\delta$.
- Let $i = \delta$ and $i^* = \lambda$. Let $\alpha = \min\{k : v_k = v_\delta\}$ and $\beta = \{k : u_k = u_\lambda\}$. Then

$$\begin{aligned} \mathbf{v} \ominus \mathbf{e}_\delta &= \mathbf{v} - \mathbf{e}_\alpha \\ &= (\mathbf{u} + \mathbf{e}_\lambda - \mathbf{e}_\delta) - \mathbf{e}_\alpha \\ &= (\mathbf{u} - \mathbf{e}_\beta) + \mathbf{e}_\lambda + \mathbf{e}_\beta - \mathbf{e}_\delta - \mathbf{e}_\alpha \\ &= (\mathbf{u} \ominus \mathbf{e}_\lambda) + (\mathbf{e}_\lambda + \mathbf{e}_\beta - \mathbf{e}_\delta - \mathbf{e}_\alpha) . \end{aligned}$$

- Otherwise, let $i = i^* \neq \lambda, \delta$. Let $\alpha = \min\{k : v_k = v_i\}$ and $\beta = \{k : u_k = u_i\}$. If $\alpha = \beta$, then clearly $\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = \Delta(\mathbf{v}, \mathbf{u})$. On the other hand, if $\alpha \neq \beta$, then either $\alpha = \lambda$ and $\beta = \lambda - 1$, or $\alpha = \delta - 1$ and $\beta = \delta$. In the first case we have $\mathbf{v} \ominus \mathbf{e}_i = \mathbf{e} - \mathbf{e}_\lambda = \mathbf{u} - \mathbf{e}_\delta = \mathbf{e} - \mathbf{e}_{\lambda-1} + \mathbf{e}_{\lambda-1} - \mathbf{e}_\delta = (\mathbf{u} \ominus \mathbf{e}_i) + \mathbf{e}_{\lambda-1} - \mathbf{e}_\delta$, and in the second case we obtain $\mathbf{v} \ominus \mathbf{e}_i = (\mathbf{u} \ominus \mathbf{e}_i) + \mathbf{e}_\lambda - \mathbf{e}_{\delta-1}$.

□

This clearly implies that $\mathbf{E}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*})] = 1$ and $\mathbf{Pr}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) \neq 1] \geq \frac{2}{n}$.

(ii) $s_1 \neq s_2$

In this case there is $s_1 = \delta - 1$ and $s_2 = \delta$. We first pick $i^* \in [s_2]$ i.u.r. If $i^* \neq \lambda, \delta$, then we set $i = i^*$. If $i^* = \delta$ then we set $i = \lambda$. Otherwise $i^* = \lambda$ and we pick $i \in [s_1]$ i.u.r.

Similarly as in the first case, it is easy to see that this is a properly defined coupling. We also have the following claim.

Claim 5.2 If $s_1 \neq s_2$ then

$$\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = \begin{cases} 0 & \text{if } i = \lambda \text{ and } i^* = \delta \\ 2 & \text{if } i^* = \lambda \neq i \text{ and } v_i = u_\lambda \\ 1 & \text{otherwise .} \end{cases}$$

Proof : As for $s_1 = s_2$, we consider here three cases.

- If $i = \lambda$ and $i^* = \delta$, then $\mathbf{v} \ominus \mathbf{e}_i = \mathbf{v} - \mathbf{e}_\lambda = \mathbf{u} - \mathbf{e}_\delta = \mathbf{u} \ominus \mathbf{e}_{i^*}$. Hence $\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = 0$.
- If $i^* = \lambda$ and $i \neq \lambda$, then let us define $\alpha = \min\{k : v_k = v_i\}$ and $\beta = \{k : u_k = u_i\}$. In this case we have $\mathbf{v} \ominus \mathbf{e}_i = \mathbf{v} - \mathbf{e}_\alpha = (\mathbf{u} - \mathbf{e}_\beta) + (\mathbf{e}_\lambda + \mathbf{e}_\beta - \mathbf{e}_\delta - \mathbf{e}_\alpha) = (\mathbf{u} \ominus \mathbf{e}_{i^*}) + (\mathbf{e}_\lambda + \mathbf{e}_\beta - \mathbf{e}_\delta - \mathbf{e}_\alpha)$. Hence $\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = 2$ unless $v_i \neq u_\lambda$.
- The case $i = i^*$ is the same as the third case considered for $s_1 = s_2$.

□

Now, the claim above implies immediately that

$$\begin{aligned} \mathbf{Pr}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = 0] &= \frac{1}{\delta}, \\ \mathbf{Pr}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) = 2] &\leq \frac{1}{\delta} \cdot (1 - \tfrac{1}{\delta - 1}), \end{aligned}$$

and hence $\mathbf{E}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*})] < 1$ and $\mathbf{Pr}[\Delta(\mathbf{v} \ominus \mathbf{e}_i, \mathbf{u} \ominus \mathbf{e}_{i^*}) \neq 1] \geq \frac{1}{\delta} \geq \frac{1}{n}$.

Thus we have defined the first step of coupling such that independently of the values s_1 and s_2, $\mathbf{E}[\Delta(\mathbf{v}^*, \mathbf{u}^*)] \leq 1$ and $\mathbf{Pr}[\Delta(\mathbf{v}^*, \mathbf{u}^*) \neq 1] \geq \frac{1}{n}$, where $\mathbf{v}^* = \mathbf{v} \ominus \mathbf{e}_i$ and $\mathbf{u}^* = \mathbf{u} \ominus \mathbf{e}_{i^*}$.

Now we define coupling $(\mathbf{v}^*, \mathbf{u}^*) \mapsto (\mathbf{v}^\diamond, \mathbf{u}^\diamond)$ according to the coupling described in Lemma 3.3. Let $\Phi_{\mathcal{D}}$ be the permutation from Definition 3.4. Pick rs, the value returned by $\mathbb{RS}$. We define $\mathbf{v}^\diamond = \mathbf{v}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{v}^*, \mathrm{rs})}$ and $\mathbf{u}^\diamond = \mathbf{u}^* \oplus \mathbf{e}_{\overline{\mathcal{D}}(\mathbf{u}^*, \Phi_{\mathcal{D}}(\mathrm{rs}))}$. Now, by Lemma 3.3 we know that $\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond) \leq \Delta(\mathbf{v}^*, \mathbf{u}^*)$. This immediately implies that $\mathbf{E}[\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond)] \leq 1$ and that $\mathbf{Pr}[\Delta(\mathbf{v}^\diamond, \mathbf{u}^\diamond) \neq 1] \geq \frac{1}{n}$. Thus we can apply the Path Coupling Lemma to conclude with the following claim.

Claim 5.3 For Markov chains $\mathfrak{M}$ defined by protocol $\mathrm{I}_{\mathcal{B}}$ we have $\tau_{\mathfrak{M}}(\varepsilon) = \mathcal{O}(n \cdot m^2 \cdot \ln \varepsilon^{-1})$.

We can provide a more complicated analysis, which is deferred to the full version of the paper, that improves this bound and shows that $\tau_{\mathfrak{M}}(\varepsilon) = \mathcal{O}(m^2 \cdot (\ln(n\varepsilon^{-1})))$. We notice also that one can easily show that $\tau_{\mathfrak{M}}(\frac{1}{2}) = \Omega(n \cdot m)$, and that for sufficiently large m also holds $\tau_{\mathfrak{M}}(\frac{1}{2}) = \Omega(m^2)$.

6 Recovery time in the edge orientation problem

In this section we analyze the recovery time in the edge orientation problem. One could apply the same approach as in the proof of Claim 5.3 to show the recovery time $\mathcal{O}(n^5 \cdot \log \varepsilon^{-1})$ for the edge orientation problem. The main reason for such a high bound is that the maximum distance between any two states in the edge orientation problem is $\Theta(n^2)$ rather than $\Theta(m)$, as in the processes analyzed in Section 5. In this section we show another proof that uses path coupling arguments and improves that bound significantly.

We follow first a similar notation as that used in the previous sections. We can model the process by integer n-vectors such that $\sum_{i=1}^n v_i = 0$. We say we are at state $\mathbf{v}$ if v_i represents the current difference between the outdegree and indegree of the ith vertex in the multigraph. One can show (see [2, 4]) that if we start with the empty graph, i.e., if we begin at the state $\mathbf{v} = \mathbf{0}$, then the state space is a subset of $\{-\lceil \frac{n-1}{2} \rceil, \ldots, \lceil \frac{n-1}{2} \rceil\}^n$. One could define a Markov chain based on such a representation of the states of the edge orientation problem, but we prefer to use another (but equivalent) representation. We represent each state $\mathbf{v}$ by an integer n-vector $\mathbf{x}$ such that $x_i = \#\{j : v_j = \lceil \frac{n+1}{2} \rceil - i\}$. That is, x_1 is the number of vertices with the difference between the outdegree and indegree equal to $\lceil \frac{n-1}{2} \rceil$, x_2 with the difference $\lceil \frac{n-1}{2} \rceil - 1$, and so on. Let Ψ be the set of all states $\mathbf{x}$ that are reachable in the edge orientation problem from the state $\check{\mathbf{x}}$ with $\check{x}_{\lceil \frac{n+1}{2} \rceil} = n$ and $\check{x}_i = 0$ for all $i \neq \lceil \frac{n+1}{2} \rceil$ (i.e., $\check{\mathbf{x}}$ corresponds to the vector $\mathbf{v} = \mathbf{0}$).

We model the behavior of the edge orientation problem by a Markov chain $\mathfrak{M}$ with state space Ψ, initial state $\check{\mathbf{x}}$, and transitions $\mathbf{x} \mapsto \mathbf{x}^*$ defined by the following randomized procedure:

- Pick ϕ and ψ, $\phi < \psi$, i.u.r. from $[n]$. (I.e., every pair ϕ, ψ has the same probability $\binom{n}{2}^{-1}$ of being chosen.)
- Set $i = \max\{l : \sum_{s=1}^l x_s \leq \phi\}$.
- Set $j = \max\{l : \sum_{s=1}^l x_s \leq \psi\}$.
- Pick a bit b i.u.r.
- If b then $\mathbf{x}^* = \mathbf{x} - \mathbf{e}_i + \mathbf{e}_{i+1} - \mathbf{e}_j + \mathbf{e}_{j-1}$;
- otherwise $\mathbf{x}^* = \mathbf{x}$.

Remark 1 We have introduced bit b (which is not used in the original edge orientation problem) to ensure that $\mathfrak{M}$ is ergodic. It should be clear that the process defined above describes the edge orientation problem of Azar et al. [2] that may be slow down roughly by a factor of two (more formally, with very high probability, after $\Omega(n)$ steps the slowdown factor will be $2 \pm o(1)$).

Let us notice also that one could choose i and j using the following (equivalent) procedure:

Pick at random i such that $\mathbf{Pr}[i = k] = \frac{x_k}{n}$ and then pick at random j such that $\mathbf{Pr}[j = k] = \begin{cases} \frac{x_k}{n-1} & \text{if } k \neq i \\ \frac{x_k - 1}{n-1} & \text{if } k = i. \end{cases}$

If $i > j$ then exchange i with j. ♣

We apply the Path Coupling Lemma to the Markov chain $\mathfrak{M}$ for the edge orientation problem. We first describe the distance metric $\Delta(\cdot, \cdot)$ on $\Psi \times \Psi$ used in our construction.

Definition 6.1 For every $\mathbf{x} \in \Psi$, let us define $\mathfrak{S}(\mathbf{x})$ as the set of all vectors $\mathbf{y}$ in Ψ for which there is a λ, $1 \leq \lambda \leq n - 2$, such that $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - 2\mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+2}$.

To ensure symmetry, let us also define $\overline{\mathfrak{S}}(\mathbf{x}) = \mathfrak{S}(\mathbf{x}) \cup (\mathfrak{S}(\mathbf{x}))^{-1}$. That is, $\overline{\mathfrak{S}}(\mathbf{x})$ is the set of all vectors $\mathbf{y}$ in Ψ for which there is a λ, $1 \le \lambda \le n-2$, such that either $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - 2\mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+2}$ or $\mathbf{x} = \mathbf{y} - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2}$.

Definition 6.2 For every $\mathbf{x} \in \Psi$ and $k \ge 1$, let us define $\mathfrak{Z}_k(\mathbf{x})$ as the set of all vectors $\mathbf{y}$ in Ψ for which there is a λ, $1 \le \lambda \le n-k-1$, such that $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+k} + \mathbf{e}_{\lambda+k+1}$ and $x_i = 0$ for every $\lambda+1 \le i \le \lambda+k$.

To ensure symmetry, let us also define $\overline{\mathfrak{Z}_k}(\mathbf{x}) = \mathfrak{Z}_k(\mathbf{x}) \cup (\mathfrak{Z}_k(\mathbf{x}))^{-1}$. That is, $\overline{\mathfrak{Z}_k}(\mathbf{x})$ is the set of all vectors $\mathbf{y}$ in Ψ for which there is a λ, $1 \le \lambda \le n-k-1$, such that either $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+k} + \mathbf{e}_{\lambda+k+1}$ and $x_i = 0$ for every $\lambda+1 \le i \le i+k$, or $\mathbf{x} = \mathbf{y} - \mathbf{e}_\lambda + \mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+k} - \mathbf{e}_{\lambda+k+1}$ and $y_i = 0$ for every $\lambda+1 \le i \le i+k$.

Definition 6.3 For every pair $\mathbf{x}, \mathbf{y} \in \Psi$ let us define the distance between $\mathbf{x}$ and $\mathbf{y}$ as follows:

- if $\mathbf{x} = \mathbf{y}$ then $\Delta(\mathbf{x}, \mathbf{y}) = 0$,
- if $\mathbf{y} \in \overline{\mathfrak{S}}(\mathbf{x})$ then $\Delta(\mathbf{x}, \mathbf{y}) = 1$, and
- otherwise

$$\Delta(\mathbf{x}, \mathbf{y}) = \min \begin{cases} k & \text{if } \mathbf{y} \in \overline{\mathfrak{Z}_k}(\mathbf{x}) \\ 1 + \min_{\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{x})} \{\Delta(\mathbf{z}, \mathbf{y})\} & \text{otherwise} \end{cases} .$$

Claim 6.1 $\Delta(\cdot, \cdot)$ is a metric over $\Psi \times \Psi$.

Proof : The definition of $\Delta(\cdot, \cdot)$ immediately implies that all the conditions for being a metric (nonnegativity, zero-property, symmetry, and triangle inequality) provided $\Delta(\mathbf{x}, \mathbf{y})$ is well defined (i.e., finite) for every pair $\mathbf{x}, \mathbf{y} \in \Psi$. One can further show that $\Delta(\mathbf{x}, \mathbf{y})$ is finite for every $\mathbf{x}, \mathbf{y} \in \Psi$. □

Now we shall define a path coupling $(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x}^*, \mathbf{y}^*)$ for $\mathfrak{M}$ with the metric $\Delta(\cdot, \cdot)$ defined on $\Psi \times \Psi$ and with $\Gamma = \{(\mathbf{x}, \mathbf{y}) \in \Psi \times \Psi :$ either $\mathbf{y} \in \overline{\mathfrak{S}}(\mathbf{x})$ or $\mathbf{y} \in \overline{\mathfrak{Z}_k}(\mathbf{x})$ for some $k \in \mathbb{N}\}$. Thus let us consider a pair $(\mathbf{x}, \mathbf{y}) \in \Gamma$ such that $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+k} + \mathbf{e}_{\lambda+k+1}$ for some $1 \le \lambda \le n-k-1$. Notice that since $(\mathbf{x}, \mathbf{y}) \in \Gamma$, we must have either $k = 1$ or $x_{\lambda+1} = \cdots = x_{\lambda+k} = 0$.

Coupling

Pick ϕ and ψ, $\phi < \psi$, i.u.r. from $[n]$. Set further

$$\begin{aligned} i &= \max\{l : \textstyle\sum_{s=1}^{l} x_s \le \phi\} , \\ j &= \max\{l : \textstyle\sum_{s=1}^{l} x_s \le \psi\} , \\ i^* &= \max\{l : \textstyle\sum_{s=1}^{l} y_s \le \phi\} , \\ j^* &= \max\{l : \textstyle\sum_{s=1}^{l} y_s \le \psi\} . \end{aligned}$$

(Observe that we set i and i^* such that $i = i^*$ unless either $i = \lambda$ and $i^* = \lambda+1$, or $i = \lambda+k+1$ and $i^* = \lambda+k$. Similarly, j and j^* are set such that if $j \ne j^*$ then either $j = \lambda$ and $j^* = \lambda+1$, or $j = \lambda+k+1$ and $j^* = \lambda+k$.)

Finally, pick bit b i.u.r. and set

$$b^* = \begin{cases} 1-b & \text{if } \lambda = \delta - 2, i = \lambda, j = \lambda+2, i^* = j^* = \lambda+1 \\ b & \text{otherwise} . \end{cases}$$

We define coupling $(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x}^*, \mathbf{y}^*)$ by setting

$$\begin{aligned} \mathbf{x}^* &= \begin{cases} \mathbf{x} - \mathbf{e}_i + \mathbf{e}_{i+1} - \mathbf{e}_j + \mathbf{e}_{j-1} & \text{if } b = 1 , \\ \mathbf{x} & \text{if } b = 0 , \end{cases} \\ \mathbf{y}^* &= \begin{cases} \mathbf{y} - \mathbf{e}_{i^*} + \mathbf{e}_{i^*+1} - \mathbf{e}_{j^*} + \mathbf{e}_{j^*-1} & \text{if } b^* = 1 , \\ \mathbf{y} & \text{if } b^* = 0 . \end{cases} \end{aligned}$$

Analysis of the coupling

One can easily verify that this is a properly defined coupling $(\mathbf{x}, \mathbf{y}) \mapsto (\mathbf{x}^*, \mathbf{y}^*)$. Now we state two key lemmas that describe the behavior of the distance $\Delta(\mathbf{x}^*, \mathbf{y}^*)$ after applying the coupling. The proofs of these lemmas are quite simple, though they are rather technical and use the case analysis.

Lemma 6.2 If $\Delta(\mathbf{x}, \mathbf{y}) = 1$, then $\mathbf{E}[\Delta(\mathbf{x}^*, \mathbf{y}^*)] \le 1 - \binom{n}{2}^{-1}$.

Proof : We consider the following seven cases:

(1) If $b = 0$ and not $(i = \lambda, i^* = \lambda+1 = j^*, j = \lambda+2)$, then $\mathbf{x}^* = \mathbf{x}$ and $\mathbf{y}^* = \mathbf{y}$. Hence $\Delta(\mathbf{x}^*, \mathbf{y}^*) = 1$.

(2) If $b = 1$, $i = i^*$, and $j = j^*$, then $\mathbf{x}^* = \mathbf{y}^* + \mathbf{e}_\lambda - 2\mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+2}$. Therefore $\Delta(\mathbf{x}^*, \mathbf{y}^*) = 1$.

(3) If $b = 1$, $i = i^*$, $j = \lambda$, and $j^* = \lambda+1$, then $\mathbf{x}^* = \mathbf{y}^* + \mathbf{e}_{\lambda-1} - \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+2}$. Now we consider three cases:

(3.1) If $x^*_\lambda > 0$, then we may set $\mathbf{z} = \mathbf{x}^* - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2}$, and since $\mathbf{y}^* = \mathbf{z} - \mathbf{e}_{\lambda-1} + 2\mathbf{e}_\lambda - \mathbf{e}_{\lambda+1}$, we get $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{x}) \cup \overline{\mathfrak{S}}(\mathbf{y})$, and thus $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(3.2) If $x^*_\lambda = 0$ and $x^*_{\lambda+1} > 0$, then we may set $\mathbf{z} = \mathbf{x}^* - \mathbf{e}_{\lambda-1} + 2\mathbf{e}_\lambda - \mathbf{e}_{\lambda+1}$, and since $\mathbf{y}^* = \mathbf{z} - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2}$, we obtain $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{x}) \cup \overline{\mathfrak{S}}(\mathbf{y})$, and hence $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(3.3) If $x^*_\lambda = x^*_{\lambda+1} = 0$, then clearly $\mathbf{x}^* \in \overline{\mathfrak{Z}_2}(\mathbf{y}^*)$, and thus $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(4) If $b = 1$, $i = \lambda+2$, $i^* = \lambda+1$, and $j = j^*$, then $\mathbf{x}^* = \mathbf{y}^* + \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2} + \mathbf{e}_{\lambda+3}$. Now, as in case (3), we consider three subcases:

(4.1) If $x^*_{\lambda+1} > 0$, then for $\mathbf{z} = \mathbf{x}^* - \mathbf{e}_{\lambda+1} + 2\mathbf{e}_{\lambda+2} - \mathbf{e}_{\lambda+3}$ we will have $\mathbf{y}^* = \mathbf{z} - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2}$, and hence $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{x}) \cup \overline{\mathfrak{S}}(\mathbf{y})$, and thus $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(4.2) If $x^*_{\lambda+1} = 0$ and $x^*_{\lambda+2} > 0$, then for $\mathbf{z} = \mathbf{x}^* - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2}$ we obtain $\mathbf{y}^* = \mathbf{z} - \mathbf{e}_{\lambda+1} + 2\mathbf{e}_{\lambda+2} - \mathbf{e}_{\lambda+3}$, and thus $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{x}) \cup \overline{\mathfrak{S}}(\mathbf{y})$, and hence $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(4.3) If $x^*_{\lambda+1} = x^*_{\lambda+2} = 0$, then clearly $\mathbf{x}^* \in \overline{\mathfrak{Z}_2}(\mathbf{y}^*)$, and thus $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$.

(5) If $b = 1$, $i = i^*$, $j = \lambda+2$, and $j = \lambda+1$, then $\mathbf{x}^* = \mathbf{y}^*$.

(6) If $b = 1$, $i = \lambda$, $i^* = \lambda+1$, and $j = j^*$, then $\mathbf{x}^* = \mathbf{y}^*$.

(7) If $i = \lambda, i^* = \lambda+1 = j^*, j = \lambda+2$, then we have two possibilities.

- If $b = 1$, then $\mathbf{y}^* = \mathbf{y}$ and $\mathbf{x}^* = \mathbf{x} - \mathbf{e}_\lambda + 2\mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2} = \mathbf{y}^*$;
- If $b = 0$, then $\mathbf{y}^* = \mathbf{y} + \mathbf{e}_\lambda - 2\mathbf{e}_{\lambda+1} + \mathbf{e}_{\lambda+2} = \mathbf{x} = \mathbf{x}^*$.

Therefore we obtain $\Delta(\mathbf{x}^*, \mathbf{y}^*) = 0$ in cases (5)–(7), $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 2$ in cases (3)–(4), and $\Delta(\mathbf{x}^*, \mathbf{y}^*) \le 1$ in the remaining cases. Now one can show that

$$\begin{aligned} \mathbf{Pr}[\Delta(\mathbf{x}^*, \mathbf{y}^*) = 0] &= \frac{1}{n} + \frac{1 + x_{\lambda+1}}{n(n-1)} \ge \frac{1}{n} + \frac{1}{n(n-1)} , \\ \mathbf{Pr}[\Delta(\mathbf{x}^*, \mathbf{y}^*) = 2] &= \frac{1}{n} - \frac{1 + x_{\lambda+1}}{n(n-1)} \le \frac{1}{n} - \frac{1}{n(n-1)} . \end{aligned}$$

Therefore

$$\mathbf{E}[\Delta(\mathbf{x}^*, \mathbf{y}^*)] = 1 - \frac{2 \cdot (1 + x_{\lambda+1})}{n(n-1)} \le 1 - \frac{2}{n(n-1)} .$$

□

Lemma 6.3 If $\Delta(\mathbf{x},\mathbf{y}) = k$ and $\mathbf{y} \in \overline{\mathfrak{Z}_k}(\mathbf{x})$, then $\mathbf{E}[\Delta(\mathbf{x}^*,\mathbf{y}^*)] \leq \Delta(\mathbf{x},\mathbf{y}) - \binom{n}{2}^{-1}$.

Proof : Since the proof of Lemma 6.3 is similar to the proof of Lemma 6.2 (and actually, even more technical), a sketch of the proof is only presented.

Let $\Delta(\mathbf{x},\mathbf{y}) = k$ and $\mathbf{y} \in \overline{\mathfrak{Z}_k}(\mathbf{x})$ for some $k \geq 2$. We assume, without loss of generality, that $\mathbf{x} = \mathbf{y} + \mathbf{e}_\lambda - \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+k} + \mathbf{e}_{\lambda+k+1}$ for some $1 \leq \lambda \leq n-k-1$, and that $x_{\lambda+1} = \cdots = x_{\lambda+k} = 0$.

Similarly as in the proof of Lemma 6.2, we consider seven main cases.

(1) If $b = 0$, then clearly $\Delta(\mathbf{x}^*,\mathbf{y}^*) = \Delta(\mathbf{x},\mathbf{y}) = k$.

(2) If $b = 1$, $i = i^*$, and $j = j^*$, then the case analysis shows that $\Delta(\mathbf{x}^*,\mathbf{y}^*) \leq \Delta(\mathbf{x},\mathbf{y}) = k$. Let us notice that this case is much more complicated that the similar case in the proof of Lemma 6.2. The main difference is that if $i = \lambda$ or $j = \lambda + k + 1$, then must deal with a nonpleasent situation that $\mathbf{y}^* \notin \overline{\mathfrak{Z}_k}(\mathbf{x}^*)$. But in that case we can always either find a $\mathbf{z} \in \Psi$ such that $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{y}^*)$ and $\mathbf{z} \in \overline{\mathfrak{Z}_{k-1}}(\mathbf{x}^*)$, or find a pair $\mathbf{z},\mathbf{u} \in \Psi$ such that $\mathbf{z} \in \overline{\mathfrak{S}}(\mathbf{y}^*)$, $\mathbf{u} \in \overline{\mathfrak{S}}(\mathbf{z})$, and $\mathbf{u} \in \overline{\mathfrak{Z}_{k-2}}(\mathbf{x}^*)$. Thus in either case we obtain $\Delta(\mathbf{x}^*,\mathbf{y}^*) \leq k$.

(3)–(4) If $b = 1$ and either $i = i^*, j = \lambda, j^* = \lambda + 1$, or $i = \lambda + k + 1, i^* = \lambda + k, j = j^*$, then $\Delta(\mathbf{x}^*,\mathbf{y}^*) \leq \Delta(\mathbf{x},\mathbf{y}) + 1 = k + 1$. Here, similarly as in case (2), we must pay special attention to the case $\mathbf{y}^* \notin \overline{\mathfrak{Z}_{k+1}}(\mathbf{x}^*)$.

(5)–(6) If $b = 1$ and either $i = i^*, j = \lambda + k + 1, j^* = \lambda + k$, or $i = \lambda, i^* = \lambda + 1, j = j^*$, then $\Delta(\mathbf{x}^*,\mathbf{y}^*) \leq \Delta(\mathbf{x},\mathbf{y}) - 1 = k - 1$. As above, special care is required for the case $\mathbf{y}^* \notin \overline{\mathfrak{Z}_{k+1}}(\mathbf{x}^*)$.

(7) If $b = 1, i = \lambda + k + 1, i^* = \lambda + k, j = \lambda + k + 1$, and $j^* = \lambda + k$, then $\mathbf{x}^* = \mathbf{y}^* + \mathbf{e}_{\lambda+1} - \mathbf{e}_{\lambda+2} - \mathbf{e}_{\lambda+k} + \mathbf{e}_{\lambda+k+1}$. Therefore we have $\mathbf{x}^*_{\lambda+2} = \cdots = \mathbf{x}^*_{\lambda+k} = 0$, and hence either $\mathbf{x}^* = \mathbf{y}^*$ for $k = 2$, or $\mathbf{y}^* \in \overline{\mathfrak{Z}_{k-2}}(\mathbf{x}^*)$, and thus $\Delta(\mathbf{x}^*,\mathbf{y}^*) \leq \Delta(\mathbf{x},\mathbf{y}) - 2 = k - 2$ for $k > 2$.

The estimation of the expected value of $\Delta(\mathbf{x}^*,\mathbf{y}^*)$ is entirely the same as in the proof of Lemma 6.2. □

Since these two lemmas imply that for every pair $(\mathbf{x},\mathbf{y}) \in \Gamma$ we have $\mathbf{E}[\Delta(\mathbf{x}^*,\mathbf{y}^*)] \leq \Delta(\mathbf{x},\mathbf{y}) - \binom{n}{2}^{-1}$, we may use the trivial bound $\Delta(\mathbf{x},\mathbf{y}) \leq n$ (notice that this bound holds only for $(\mathbf{x},\mathbf{y}) \in \Gamma$) to obtain $\mathbf{E}[\Delta(\mathbf{x}^*,\mathbf{y}^*)] \leq \Delta(\mathbf{x},\mathbf{y}) \cdot \left(1 - \frac{1}{n} \cdot \binom{n}{2}^{-1}\right)$. Therefore the Path Coupling Lemma yields the following corollary.

Corollary 6.4 For the Markov chain $\mathfrak{M}$ for the edge orientation problem we have $\tau_{\mathfrak{M}}(\varepsilon) = \mathcal{O}(n^3(\ln n + \ln \varepsilon^{-1}))$.

We can further improve this bound by using some subtle technical details of the Path Coupling Lemma.

Theorem 2 For the Markov chain $\mathfrak{M}$ for the edge orientation problem we have $\tau_{\mathfrak{M}}(\frac{1}{n}) = \mathcal{O}(n^2 \ln^2 n)$.

Proof : We only sketch the proof of the theorem. Our main observation is that if we could provide a better bound for the distance between any pair of the elements in Γ, then we could immediately improve the bound for the mixing time of $\mathfrak{M}$.

We proceed in two steps. First we show that independently of the initial distribution of $X_{t_0} \in \Psi$, after another $\tau = \mathcal{O}(n^2 \ln n)$ steps of $\mathfrak{M}$ the maximum difference between the outdegree and the indegree in $X_{t_0+\tau}$ will be $\mathcal{O}(\ln n)$ w.h.p. Moreover such a bound will hold in the next n^3 steps w.h.p.

We define now the coupling $(\mathbf{X}_t,\mathbf{Y}_t)_{t\in\mathbb{N}}$ of $\mathfrak{M}$ for any $\mathbf{X}_0,\mathbf{Y}_0 \in \Psi$ such that in the first $\tau = \mathcal{O}(n^2 \ln n)$ steps each of $(\mathbf{X}_t)_{t\leq\tau}$, $(\mathbf{Y}_t)_{t\leq\tau}$ runs independently. Thus we know that for every t, $\tau \leq t \leq n^3$, the maximum difference between the outdegree and the indegree in each of $\mathbf{X}_t$ and $\mathbf{Y}_t$ is $\mathcal{O}(\ln n)$ w.h.p. From now on we shall condition on that event.

This allows us to apply the Path Coupling Lemma for $(\mathbf{X}_t,\mathbf{Y}_t)_{\tau\leq t\leq n^3}$ such that there always exists a sequence $\mathbf{X}_t = \mathbf{Z}_0,\mathbf{Z}_1,\ldots,\mathbf{Z}_{r-1},\mathbf{Z}_r = \mathbf{Y}_t$, where $\sum_{i=0}^{r-1} \Delta(\mathbf{Z}_i,\mathbf{Z}_{i+1}) = \Delta(\mathbf{X}_t,\mathbf{Y}_t)$, $(\mathbf{Z}_i,\mathbf{Z}_{i+1}) \in \Gamma$, and moreover $\Delta(\mathbf{Z}_i,\mathbf{Z}_{i+1}) = \mathcal{O}(\ln n)$ for every $0 \leq i < r$. Given that, we may apply the Path Coupling Lemma with the bound $\mathbf{E}[\Delta(\mathbf{x}^*,\mathbf{y}^*)] \leq \Delta(\mathbf{x},\mathbf{y}) \cdot \left(1 - \frac{1}{\mathcal{O}(\ln n)} \cdot \binom{n}{2}^{-1}\right)$ obtained from Lemmas 6.2 and 6.3 to obtain the required mixing time. □

Since it is easy to see that $\tau_{\mathfrak{M}}(\frac{1}{2}) = \Omega(n^2)$, the bound in Theorem 2 is almost tight.

7 Conclusions

We have presented a general framework that can be used to study recovery times of closed discrete-time dynamic processes. Although we have assumed that in dynamic processes in each step a random ball is removed, or the load of a random non-empty bin is decreased, our techniques can be also applied to processes in which we remove a ball according to other probability distributions.

In the paper we have focused exclusively on closed systems. Our approach can be also used to study *open systems*, i.e. systems in which the number of balls may change in time. However, in such systems our techniques seem to be more problem-dependent. There are two different classes of processes to which different variants of our method can be applied. In the first class one requires the number of balls to be bounded all the time. The approach used in the paper can be refined to be applicable to such systems. In the second class we do not assume any upper bound for the number of balls existing in the system. (For example, start with 0 balls and repeatedly, with probability $\frac{1}{2}$ remove a random existing ball and with probability $\frac{1}{2}$ allocate a new ball into a random bin.) In such systems we can use the coupling approach presented in the paper to estimate the time until the distributions of the process for any given pair of input states will be almost the same. (E.g., in the example above, we could estimate the time until the process that starts with 0 balls has almost the same distribution of balls as the process that starts with m balls arbitrarily distributed.)

Finally, we defer to the full version analysis of dynamic processes that allow *relocations of the balls.*

References

[1] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen. Parallel randomized load balancing. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 238–247, Las Vegas, Nevada, May 29–June 1, 1995. ACM Press, New York, NY.

[2] M. Ajtai, J. Aspnes, M. Naor, Y. Rabani, L. J. Schulman, and O. Waarts. Fairness in scheduling. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 477–485, San Francisco, CA, January 22–24, 1995. SIAM, Philadelphia, PA.

[3] D. Aldous. Random walks of finite groups and rapidly mixing Markov chains. In J. Azéma and M. Yor, editors, *Séminaire de Probabilités XVII, 1981/82*, volume 986 of *Lecture Notes in Mathematics*, pages 243–297. Springer-Verlag, Berlin, 1983.

[4] R. Anderson, L. Lovász, P. Shor, J. Spencer, E. Tardos, and S. Winograd. Disks, balls, and walls: Analysis of a combinatorial game. *American Mathematical Monthly*, 96(6):481–493, 1989.

[5] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 593–602, Montreal, Quebec, Canada, May 23–25, 1994. ACM Press, New York, NY.

[6] P. Berenbrink, F. Meyer auf der Heide, and K. Schröder. Allocating weighted jobs in parallel. In *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 302–310, Newport, RI, June 22–25, 1997. ACM Press, New York, NY.

[7] R. Bubley and M. Dyer. Path coupling: A technique for proving rapid mixing in Markov chains. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 223–231, Miami Beach, FL, October 19–22, 1997. IEEE Computer Society Press, Los Alamitos, CA.

[8] R. Bubley and M. Dyer. Faster random generation of linear extensions. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 350–354, San Francisco, CA, January 25–27, 1998. SIAM, Philadelphia, PA.

[9] R. Bubley, M. Dyer, and C. Greenhill. Beating the 2Δ bound for approximately counting colourings: A computer-assisted proof of rapid mixing. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 355–363, San Francisco, CA, January 25–27, 1998. SIAM, Philadelphia, PA.

[10] A. Czumaj, P. Kanarek, M. Kutyłowski, and K. Loryś. Delayed path coupling and its applications to analysing distributed stochastic processes for generating random permutations. Manuscript, April 1998.

[11] A. Czumaj and V. Stemann. Randomized allocation processes. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 194–203, Miami Beach, FL, October 19–22, 1997. IEEE Computer Society Press, Los Alamitos, CA.

[12] P. Diaconis. *Group Representations in Probability and Statistics*, volume 11 of *Lecture Notes - Monograph Series*. Institute of Mathematical Statistics, Hayward, CA, 1988.

[13] R. Fagin and J. H. Williams. A fair carpool scheduling algorithm. *IBM Journal of Research and Development*, 27(2):133–139, March 1983.

[14] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: An approach to approximate counting and integration. In D. S. Hochbaum, editor, *Approximation Algorithms for $\mathcal{NP}$-Hard Problems*, chapter 12, pages 482–520. PWS Publishing Company, Boston, MA, 1996.

[15] N. L. Johnson and S. Kotz. *Urn Models and Their Application: An Approach to Modern Discrete Probability Theory*. Wiley series in probability and mathematical statistics: Applied probability and statistics. John Wiley & Sons, New York, NY, 1977.

[16] V. F. Kolchin, B. A. Sevast'yanov, and V. P. Chistyakov. *Random Allocations*. V. H. Winston & Sons, Washington, D.C., 1978.

[17] T. Lindvall. *Lectures on the Coupling Method*. Wiley series in Probability and Mathematical Statistics. John Wiley & Sons, New York, NY, 1992.

[18] M. Mitzenmacher. Load balancing and density dependent jump Markov processes. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, pages 213–222, Burlington, Vermont, October 14–16, 1996. IEEE Computer Society Press, Los Alamitos, CA.

[19] M. Mitzenmacher. How useful is old information. In *Proceedings of the 16th ACM Symposium on Principles of Distributed Computing*, pages 83–91, Santa Barbara, August 21–24, 1997. ACM Press, New York, NY.

[20] M. Mitzenmacher. On the analysis of randomized load balancing schemes. In *Proceedings of the 9th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 292–301, Newport, RI, June 22–25, 1997. ACM Press, New York, NY.

[21] M. Mitzenmacher. Studying balanced allocations with differential equations. Technical Report SRC Technical Note 1997-024, System Research Center of Digital Equipment Corporation, Palo Alto, CA, October 1997.

[22] M. D. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, Computer Science Department, University of California at Berkeley, September 1996.

[23] A. Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Progress in Theoretical Computer Science. Birkhäuser, Boston, MA, 1993.

[24] V. Stemann. Parallel balanced allocations. In *Proceedings of the 8th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 261–269, Padua, Italy, June 24–26, 1996. ACM Press, New York, NY.