The Dynamic Double Coupon Collector Problem and its Applications to Randomized Alternation Algorithms

Michael Goodrich Michael Mitzenmacher Wojciech Szpankowski
Krzysztof Turowski

Abstract

We study a dynamic, paired version of the coupon collector problem and show how it can be used in the analysis of randomized alternation algorithms where a randomized query process alternates with a randomized obfuscation process. In the dynamic double coupon collector problem, there are $2n$ colored coupons organized into n mated pairs, such that a pair is collected only when both of its members are both collected with a common color. With each coupon draw there is a background ***recoloring*** step that changes the color of a randomly chosen mated pair to a new color, possibly undoing partial progress of the collector. As we show, this problem is ideally suited for the analysis of randomized alternation algorithms in information security and network topology discovery. Writing T_n for the number of draws needed to complete every pair, we first give elementary drift bounds $2nH_n \le \mathbb{E}[T_n] \le 12nH_n$, and then, through a poissonization / extreme-value / depoissonization argument, we pin down the exact leading term,

$$\mathbb{E}[T_n] \;=\; \frac{4}{5-\sqrt{17}}\, n\ln n\,(1+o(1)) \;\approx\; 4.5616\, n\ln n,$$

with Gumbel-type fluctuations on the scale of n. We also provide experimental data supporting this result.

1 Introduction

The classical ***coupon collector problem*** asks for the expected number of independent uniform draws (with replacement) from a set of n distinct coupons needed to obtain at least one of each. The answer is

$$nH_n = n \ln n + \gamma n + o(n),$$

where H_n is the n-th Harmonic number and $\gamma \approx 0.5772156649$ is the Euler–Mascheroni constant [16]. Moreover, in a suitably normalized collection, time converges to a Gumbel law [30], a fact going back to Erdős and Rényi [13] (see also Feller [15]). This classic version of the coupon collector problem is a fixture of probability theory and of the analysis of randomized algorithms [27, 28], where "covering all slots" or "resolving all items" is repeatedly governed by collector-type dynamics. In addition, analytic treatments unifying it with birthday and caching problems appear in Flajolet, Gardy and Thimonier [17].

Additional variations. Many generalizations of the coupon collector problem are known. For example, in the ***double Dixie cup*** version introduced by Newman and Shepp [31] the process progresses until it has obtained m complete sets of n items, with the expectation growing as $n \ln n + (m-1)n \ln \ln n$, showing that each subsequent complete set is expected to be obtained much faster than the first; its asymptotics, including limiting distributions and the case of unequal coupon probabilities, were sharpened by Doumas and Papanicolaou [9]. Neal [30] analyzes a general weighted version, and Myers and Wilf [29] study two simultaneous collectors. In all of these variants, however, the per-coupon progress is ***monotonic*** and ***static***—once a coupon (or a copy of it) is acquired, it is never lost and the underlying set of coupons never changes throughout the collection process.

Alternation algorithms. This limitation makes the existing variants of the coupon collector problem poor choices for analyzing what we refer to as ***alternation algorithms***, which are algorithms that alternate between actions performed by one process and responses performed by another. Examples of alternation algorithms include dynamic data structures [32] and online algorithms [3], but the prior work closest to the application domain of this paper is work on algorithms for evolving data [4, 6, 7, 18], where each major step in an algorithmic process is followed by a randomized obfuscation step that may undo progress made so far. For example, in the sorting problem with evolving data, each time a sorting process compares two elements a randomized obfuscation process chooses a pair of consecutive items in the current total order uniformly at random and swaps them. In spite of the adversarial nature of the two processes, it is possible to order n such elements in $O(n \log n)$ time so that their dislocation error from the sorted order is $O(n)$, which is optimal [6]. Also, in FOCS 2024, Giakkoupos, Kiwi, and Los [18] show that a randomized alternation algorithm is optimal and robust in this model for sorting. One of the goals of this paper is to introduce and analyze other examples of randomized alternation algorithms.

The dynamic double coupon collector problem. The variant of the coupon collector problem studied here departs from the classical setup in two ways. First, coupons come in mated pairs, and a pair counts as collected only once both of its members are held under a common label—a "paired" or "double" requirement reminiscent of two-probe hashing schemes such as ***cuckoo hashing*** [12, 14, 19, 33], in which each item is associated with a pair of hash values. Second, after each draw a background relabeling (***recoloring***) step refreshes a uniformly random coupon pair with a fresh, previously unused color; this can ***orphan*** a half-collected pair, as the old color for this orphan is not used for any future coupon. In this way, the recoloring step can undo partial progress. In particular, these features make the process dynamic and non-monotone—a singleton coupon from a pair can be lost and need to be collected again, perhaps multiple times—and are what distinguish the model from the static coupon collector problems discussed above.

We abstract the dynamic double coupon collector problem into a clean combinatorial model as follows. We have $2n$ coupons, each matched with another coupon, its ***mate***; the backs of a coupon and its mate share a unique color. Each day the collector receives one of the $2n$ coupons independently and uniformly at random. Then the dispenser picks a coupon i uniformly at random and repaints the backs of all its copies of i and its mate with a new previously-unused color. The goal is to collect all $2n$ coupons so that each coupon and its mate carry the same color on the back. We ask for the expected number, T_n, of trips to the window.

Our results and methods. In Section 2 of this paper, we give a number of randomized alternation algorithms, including algorithms with applications to information security and network topology discovery, which can be analyzed using the dynamic double coupon collector problem. We then turn to our analysis for this problem, which is the main theoretical result of this paper. We introduce some necessary notation and compute the transition kernel in Section 3. In Section 4, we give elementary bounds,

$$2nH_n \;\le\; \mathbb{E}[T_n] \;\le\; 12nH_n,$$

the lower bound by a waiting-time argument and the upper bound by an additive drift theorem applied to an explicit potential. In Section 5, we then determine the exact leading term. We poissonize the evolution of a single pair, obtaining a two-state transient continuous-time chain whose survival probability decays at rate $\alpha = \frac{5-\sqrt{17}}{4}$; an extreme-value analysis of the maximum of n such (asymptotically independent) completion times yields a Gumbel limit, and a spectral-perturbation argument shows the weak dependence between pairs is asymptotically negligible. A monotone depoissonization sandwich transfers the result back to discrete time, giving

$$\mathbb{E}[T_n] \;=\; \frac{n}{\alpha}\big(\ln n + \ln C_\alpha + \gamma\big) + o(n) \;=\; \frac{4}{5-\sqrt{17}}\, n \ln n\, (1 + o(1)) \;\approx\; 4.5616\, n \ln n.$$

Finally, we provide some experimental data that supports this result.

2 Randomized alternation algorithms

Before we give our main analysis results, let us first explore some algorithmic applications. The list of examples we provide is by no means exhaustive, however, and we anticipate that there are other applications for this problem.

Cuckoo hunting. Define the ***cuckoo hunting*** problem as that of an adversary who wants to discover the contents of an obfuscated cuckoo hash table. In particular, consider a set of key-value pairs, $x = (k, v)$, and split each one into two secret shares, s_1 and s_2, using a random nonce, r, as $s_1 = r$ and $s_2 = r \oplus x$. Use 4 random hash functions, h_1, h_2, h_3, and h_4, and a (large) random fingerprint function, $f(k)$, and, for each $x = (k, v)$, store

$$(1, f(k), s_1)$$

in position $h_1(k)$ or $h_2(f(k)) \oplus h_1(k)$, cuckoo style. Also, for each $x = (k, v)$, store $(2, f(k), s_2)$ in position $h_3(k)$ or $h_4(f(k)) \oplus h_3(k)$, cuckoo style (where these four places are distinct). To do a lookup, we just need to lookup the two shares for key k, and then xor them. We can find these shares in $O(1)$ worst-case time using our knowledge of $f(k)$ and h_1, h_2, h_3, and h_4. To bounce an s_1 element in cell i for cuckoo hash-table reasons, we just bounce it to cell $i \oplus h_2(f(k))$, using the value of $f(k)$ stored in the cell. Likewise, to bounce an s_2 item in cell i, we just bounce it to $i \oplus h_4(f(k))$. The success analysis for this 2-out-of-4 placement is similar to the analysis for 2-out-of-3 cuckoo tables and cuckoo filters; see, e.g., [12, 14, 19].

Define a random inspector, who independently chooses a cell i to examine uniformly at random in each of his inspections. If the cell is empty, the inspector tries again; hence, let us assume that the inspector always retrieves a non-empty cell (since we can compute his expected success simply by dividing by the load factor). Note that if the inspector has probed two cells, (j_1, F, s') and (j_2, F, s''), for the same key, k, he can compute $x = (k, v) = s' \oplus s''$ and verify whether $F = f(k)$ to confirm these cells are for the same key-value pair, k. If we know the cell that the inspector probed, we can simply change that element's random nonce along with its mate and we will achieve secrecy for our scheme equal to the security of the fingerprint function, $f(k)$, because the inspector is guaranteed to never have both shares for any key using the same nonce. If we cannot make any updates to this obfuscated cuckoo table, however, the inspector will discover all the key-value pairs in time that can be characterized by the classical coupon collector problem. Suppose instead that after each random probe by the inspector we are allowed to pick a random cell, j, and xor the two shares for the (k, v) pair stored in cell j with a new random nonce, R. This way, the random inspector may be able to eventually discover a (k, v) pair only if he discovers both shares for (k, v) before we have had a chance to re-randomize them. Thus, the time needed by the inspector to discover all our data is characterized by the the dynamic double coupon collector problem.

Adversarial network topology discovery. In graph reconstruction algorithms, an agent learns the topology of a network by issuing simple queries to the graph representing the network; see, e.g., [1, 2, 20, 24, 26]. Still, a network manager may wish to restrict users from easily and quickly discovering the network's topology by simple queries. Suppose, for instance, each edge, e, in a graph, G, has a unique random label, $L(e)$, and we have an list, $I(G)$, of the vertex-edge incidences in G, i.e., there is an entry, $(v, L(e))$, in $I(G)$ for each pair, (v, e), such that the edge e is incident to the vertex, v, in G. If a network query agent is able to randomly querying entries in $I(G)$, then the agent will discover all edges in G and learn the complete topology of G in a number of queries that can be analyzed by the classical coupon collector problem. However, suppose that each time the querying agent performs a query, the network's manager chooses an edge in G uniformly at random and changes its label, updating the two entries in $I(G)$ for this edge. In this case, the expected time for the query agent to reliably map the entire network topology of G is equal to the expected time needed for this instance of the dynamic double coupon collector problem. Thus, the increased data protection provided by this simple obfuscation technique is equal to the constant factor in the leading term of the expected time, T_n, for solving the dynamic double coupon collector problem.

RFID yoking proofs. In physical security and supply chain tracking, auditors often need cryptographic proof that two objects were in the same place at the same time (e.g., a high-value item and its security manual) and if the objects have RFID tags, methods establishing these relationships are known as RFID yoking proofs [8, 23]. Suppose each such ***yoked*** pair of RFID tags are assigned the same unique random nonce. Further, suppose an RFID reader uniformly polls tags at random and performs a challenge to request its nonce. To generate a yoking proof, the reader must successfully poll an RFID tag and its paired tag, which is a process that could be characterized using the classical coupon collector problem. However, suppose after each time the RFID reader queries an RFID tag there is an RFID updating process that randomly picks a yoked pair and changes the nonce for each member of this pair. In this case, the expected time it will take for the reader to successfully authenticate all required yoked pairs of tags is equal to the expected time for this instance of the dynamic double coupon collector problem.

Having introduced some motivating randomized alternation algorithms, let us next turn to our analysis for the dynamic double coupon collector problem.

3 Transition probabilities

First, we silently discard duplicated coupons after the drawing phase: this does not change the number of drawings and simplifies the state. Second, if mated coupons a_1, a_2 are repainted while we hold a_1 but not a_2, our held copy of a_1 keeps its old (now orphaned) color, so we must collect both anew; equivalently, we may ***remove*** a_1 from the held set. Thus the repaint step acts as follows: if we hold a single coupon of the repainted pair, discard it; if we already hold a mated pair, keep it.

Formally, let

- S_m be the number of unfinished pairs holding exactly one usable coupon ("singletons") after m steps;
- P_m be the number of completed pairs after m steps;
- $R_m = n - P_m$ be the number of unfinished pairs;
- $T_n(s,p)$ be the hitting time of $(0,n)$ for the chain started from $(S_0, P_0) = (s,p)$, and $T_n := T_n(0,0)$.

The state space is $\mathcal{S}_n = \{(s,p) : 0 \le p \le n,\ 0 \le s \le n-p\}$ with absorbing state $(0,n)$; the sequence $(P_m)_{m\ge 0}$ is non-decreasing.

There are three drawing events,

- $D_{0,0}$ — draw a coupon already held (a singleton or one from a pair): no change;
- $D_{-1,+1}$ — draw the mate of a held singleton: $S \to S-1$, $P \to P+1$;
- $D_{+1,0}$ — draw a coupon from a yet-unseen pair: $S \to S+1$,

and two repaint events,

- $R_{0,0}$ — repaint a pair of which we hold both or none: no change;
- $R_{-1,0}$ — repaint a pair of which we hold exactly one (discarding it): $S \to S-1$.

Hence there are up to six draw-and-repaint events per step. The draw probabilities, evaluated before the draw, are

$$\Pr[D_{0,0} \mid (s,p)] = \frac{s+2p}{2n}, \qquad \Pr[D_{-1,+1} \mid (s,p)] = \frac{s}{2n}, \qquad \Pr[D_{+1,0} \mid (s,p)] = 1 - \frac{s+p}{n}.$$

If s' denotes the number of singletons ***immediately before*** the repaint substep, then

$$\Pr[R_{0,0} \mid (s,p)] = 1 - \frac{s'}{n}, \qquad \Pr[R_{-1,0} \mid (s,p)] = \frac{s'}{n}.$$

Because the repaint occurs after the draw, s' is the ***post-draw*** number of singletons. Combining the two substeps gives the transition probabilities

$$\begin{aligned}
\Pr[(s,p) \to (s,p) \mid (s,p)] &= \frac{s+2p}{2n}\Big(1 - \frac{s}{n}\Big) + \Big(1 - \frac{s+p}{n}\Big)\frac{s+1}{n}, \\
\Pr[(s,p) \to (s-1,p) \mid (s,p)] &= \frac{s+2p}{2n}\,\frac{s}{n}, \\
\Pr[(s,p) \to (s-1,p+1) \mid (s,p)] &= \frac{s}{2n}\Big(1 - \frac{s-1}{n}\Big), \\
\Pr[(s,p) \to (s+1,p) \mid (s,p)] &= \Big(1 - \frac{s+p}{n}\Big)\Big(1 - \frac{s+1}{n}\Big), \\
\Pr[(s,p) \to (s-2,p+1) \mid (s,p)] &= \frac{s}{2n}\,\frac{s-1}{n}.
\end{aligned}$$

(In the first line the repaint following a $D_{+1,0}$ draw sees $s' = s + 1$ singletons, so it removes one with probability $\frac{s+1}{n}$, returning the chain to (s, p); this matches the $(s+1, p)$ transition, which uses $1 - \frac{s+1}{n}$. One checks directly that the five probabilities sum to 1.) Transitions whose target lies outside $\mathcal{S}_n$ have probability zero.

4 Simple bounds

The following lower bound is straightforward.

Theorem 4.1 (Simple lower bound)**.** *For every $n \geq 1$, $\mathbb{E}[T_n] \geq 2nH_n$, where H_n is the n-th harmonic number.*

Proof. If $R_m = k$ then at most k unfinished pairs hold a singleton, so the probability that the next draw completes a new pair (event $D_{-1,+1}$, with probability $s/(2n)$ and $s \leq k$) is at most $k/(2n)$. Hence, while $R_m = k$, the expected waiting time until P_m next increases is at least $2n/k$. Summing over the levels $k = n, n-1, \ldots, 1$ that R_m visits,

$$\mathbb{E}[T_n] \;\geq\; \sum_{k=1}^{n} \frac{2n}{k} \;=\; 2nH_n. \qquad \square$$

For the upper bound, define for all $p < n$ the potential

$$V(s,p) \;=\; 6n\, H_{n-p} \;-\; \frac{ns}{n-p}, \qquad V(0,n) = 0,$$

and write $V_m := V(S_m, P_m)$.

Lemma 4.2. *For all $n \geq 2$, $0 \leq p \leq n-1$, $0 \leq s \leq n-p$,*

$$\mathbb{E}[V_m - V_{m+1} \mid (S_m, P_m) = (s,p)] \;\geq\; \tfrac{1}{2}.$$

Proof. For $p = n-1$ (so $u := n - p = 1$):

- if $s = 0$, the only potential-changing transition is $(0, n-1) \to (1, n-1)$, which has probability $(n-1)/n^2$ and decreases V by n; thus $\mathbb{E}[V_m - V_{m+1} \mid (0, n-1)] = \frac{n-1}{n} \geq \frac{1}{2}$;
- if $s = 1$, a completion $(1, n-1) \to (0, n)$ happens with probability $1/(2n)$ and decreases V by $5n$, while repainting the singleton after drawing an already-held coupon, $(1, n-1) \to (0, n-1)$, has probability $(2n-1)/(2n^2)$ and increases V by n; hence $\mathbb{E}[V_m - V_{m+1} \mid (1, n-1)] = \frac{3n+1}{2n} \geq \frac{1}{2}$.

For $0 \leq p \leq n-2$ write $u = n - p \geq 2$. The potential changes are

$$\begin{aligned}
(s,p) \to (s-1,p):&\quad V_m - V_{m+1} = -\frac{n}{u},\\
(s,p) \to (s+1,p):&\quad V_m - V_{m+1} = \frac{n}{u},\\
(s,p) \to (s-1,p+1):&\quad V_m - V_{m+1} = 6n(H_u - H_{u-1}) - \frac{ns}{u} + \frac{n(s-1)}{u-1} = -\frac{n(s-6)}{u} + \frac{n(s-1)}{u-1},\\
(s,p) \to (s-2,p+1):&\quad V_m - V_{m+1} = 6n(H_u - H_{u-1}) - \frac{ns}{u} + \frac{n(s-2)}{u-1} = -\frac{n(s-6)}{u} + \frac{n(s-2)}{u-1}.
\end{aligned}$$

Multiplying each change by its transition probability and simplifying gives

$$\mathbb{E}[V_m - V_{m+1} \mid (s,p)] = \frac{2u(u-1)(n-1) + s\big(n(u-2)+3u-2\big) + s^2(n-1)}{2u(u-1)n}.$$

For $n \geq u \geq 2$ we have $n(u-2)+3u-2 > 0$, so every term in the numerator is nonnegative and

$$\mathbb{E}[V_m - V_{m+1} \mid (s,p)] \;\geq\; \frac{2u(u-1)(n-1)}{2u(u-1)n} = \frac{n-1}{n} \;\geq\; \tfrac{1}{2}. \qquad \square$$

We couple this with the following additive drift theorem.

Theorem 4.3 ([25, Theorem 7]). *Let $\mu \leq 0$, let $(X_m)_{m\in\mathbb{N}}$ be real random variables, and let $T = \inf\{m : X_m \leq 0\}$. Suppose* (a) *$X_m \geq \mu$ for all $m \leq T$, and* (b) *there is $\delta > 0$ with $X_m - \mathbb{E}[X_{m+1} \mid X_0,\ldots,X_m] \geq \delta$ for all $m < T$. Then $\mathbb{E}[T \mid X_0] \leq \dfrac{X_0 - \mathbb{E}[X_T \mid X_0]}{\delta} \leq \dfrac{X_0-\mu}{\delta}$.*

Corollary 4.4 (Simple upper bound). *For every $n \geq 2$, $\mathbb{E}[T_n] \leq 12nH_n$.*

Proof. On every transient state $V(s,p) = 6nH_{n-p} - \frac{ns}{n-p} \geq n(6H_{n-p}-1) \geq 5n > 0$, and $V(0,n) = 0$, so $T = \inf\{m : V_m \leq 0\}$ coincides with the absorption time T_n. With $\mu = 0$ and $\delta = \frac{1}{2}$ the drift condition of Theorem 4.3 holds by Lemma 4.2, hence $\mathbb{E}[T_n] \leq 2V_0$. Since $V_0 = V(0,0) = 6nH_n$, we get $\mathbb{E}[T_n] \leq 12nH_n$. $\square$

5 Exact rate of growth

By Theorem 4.1 and Corollary 4.4 we have $2nH_n \leq \mathbb{E}[T_n] \leq 12nH_n$; asymptotically $2n\ln n \lesssim \mathbb{E}[T_n] \lesssim 12n\ln n$. To obtain the exact leading term we proceed more carefully.

5.1 Poissonization for the one-pair process

We first analyze the poissonized evolution of a fixed pair in scaled continuous time $\tau = t/n$, in which draw and repaint events involving that pair each occur at total rate 1. A fixed pair has two transient states 0 and 1 (no held coupon, and exactly one held coupon, respectively) and an absorbing state 2 (completed pair).

Decomposing the one-pair generator into draw and repaint contributions, the draw generator is

$$D = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}, \qquad R = \begin{pmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

the repaint having an effect only in state 1, where it removes the present coupon and returns the pair to state 0. Hence the full one-pair generator is

$$Q = D + R = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -\frac{3}{2} & \frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}.$$

Since state 2 is absorbing, the transient parts D^*, R^*, Q^* are the submatrices on states $0, 1$.

Theorem 5.1. *Let T^* be the scaled completion time of a single pair started from state 0. For every $\tau \geq 0$,*

$$\Pr[T^* > \tau] = C_\alpha e^{-\alpha\tau} + C_\beta e^{-\beta\tau}, \qquad \alpha,\beta = \frac{5 \mp \sqrt{17}}{4}, \quad C_\alpha, C_\beta = \frac{\sqrt{17} \pm 5}{2\sqrt{17}}.$$

In particular T^* is dominated by the slower mode: $\Pr[T^* > \tau] = \Theta(e^{-\alpha\tau})$.

Proof. For a transient continuous-time chain with generator Q^* and initial distribution X_0, the survival probability is $\Pr[T^* > \tau] = X_0^\top \exp(Q^*\tau)\,\mathbf{1}$ (e.g. [5, Prop. 4.1]). Here $X_0 = (1,0)^\top$; the eigenvalues of Q^* solve $\lambda^2 + \frac{5}{2}\lambda + \frac{1}{2} = 0$, i.e. $\lambda = -\alpha, -\beta$ with $\alpha, \beta = \frac{5 \mp \sqrt{17}}{4}$, and a direct evaluation of the spectral coefficients gives the displayed formula. Since $0 < \alpha < \beta$, the first mode dominates. $\square$

5.2 Independent-pair completion time

Let $T_1^*, T_2^*, \ldots$ be independent copies of T^* and set $\widetilde{T}_n^{\mathrm{ind}} = \max_{1\le i\le n} T_i^*$. Extreme-value theory gives the poissonized independent n-pair asymptotics.

Lemma 5.2. *Let $X_1, \ldots, X_n$ be i.i.d. nonnegative with $\Pr[X > x] = Ce^{-ax}(1 + o(1))$ for constants $a, C > 0$ as $x \to \infty$. Let $X_{(n)} = \max_i X_i$ and*

$$Y_{(n)} = \frac{X_{(n)} - b_n}{c_n}, \qquad b_n = \frac{1}{a}\ln(nC), \quad c_n = \frac{1}{a}.$$

Then $Y_{(n)}$ converges in expectation to a standard Gumbel variable G: $\lim_{n\to\infty} \mathbb{E}[Y_{(n)}] = \mathbb{E}[G] = \gamma$, the Euler–Mascheroni constant.

Proof. For fixed x, $b_n + c_n x \to \infty$, so $\lim_n n\Pr[X > b_n + c_n x] = \lim_n nCe^{-a(b_n + c_n x)}(1 + o(1)) = e^{-x}$, and by the standard extreme-value criterion [11, Prop. 3.3.2], $Y_{(n)} \to G$ in distribution. For convergence of expectations we show asymptotic uniform integrability [36, Thm. 2.20], [10, Thm. 5.5.2]:

$$\lim_{K\to\infty} \limsup_{n\to\infty} \Big(K\Pr[|Y_{(n)}| > K] + \int_K^\infty \Pr[|Y_{(n)}| > x]\, dx \Big) = 0,$$

treating the two tails separately. Choose x_0 and $C_+ > C > C_- > 0$ with $C_- e^{-ax} \le \Pr[X > x] \le C_+ e^{-ax}$ for $x \ge x_0$. For the right tail, since $b_n \to \infty$, for large n and all $x \ge K$,

$$\Pr[Y_{(n)} > x] = 1 - F^n(b_n + c_n x) \le n\big(1 - F(b_n + c_n x)\big) \le nC_+ e^{-a(b_n + c_n x)} \le \tfrac{C_+}{C} e^{-x},$$

using $1 - u^n \le n(1-u)$, whence $\int_K^\infty \Pr[Y_{(n)} > x]\, dx \le \frac{C_+}{C} e^{-K}$, which vanishes as $K \to \infty$. For the left tail, first when $b_n - c_n x \ge x_0$,

$$\Pr[Y_{(n)} < -x] = F^n(b_n - c_n x) \le \exp\!\big(-n(1 - F(b_n - c_n x))\big) \le \exp\!\Big(-\tfrac{C_-}{C} e^x\Big),$$

using $u \le e^{-(1-u)}$; since e^{-de^x} is integrable on $[0,\infty)$ for each $d > 0$, this contribution vanishes as $K \to \infty$. The remaining range $b_n - c_n x < x_0$ contributes, after substituting $y = b_n - c_n x$,

$$\int_{L_n}^\infty F^n(b_n - c_n x)\, dx = a \int_0^{x_0} F^n(y)\, dy \le a\, x_0\, F(x_0)^n,$$

which tends to 0 exponentially since $F(x_0) < 1$. Thus $(Y_{(n)})_n$ is uniformly integrable and $\mathbb{E}[Y_{(n)}] \to \mathbb{E}[G] = \gamma$. $\square$

Corollary 5.3. *Under the hypotheses of Lemma 5.2, $\mathbb{E}[X_{(n)}] = \frac{1}{a}(\ln n + \ln C + \gamma) + o(1)$.*

Applying Corollary 5.3 to $T_1^*, T_2^*, \ldots$ (with $a = \alpha$, $C = C_\alpha$) and returning to unscaled time $t = n\tau$ gives the independent-pair prediction

$$n\,\mathbb{E}[\widetilde{T}_n^{\mathrm{ind}}] = \frac{n}{\alpha}\big(\ln n + \ln C_\alpha + \gamma\big) + o(n).$$

As we show next, the same asymptotics holds for the actual n-pair process, despite the weak dependence between pairs.

5.3 Including dependence

In both the independent and the actual poissonized process (scaled time), draw-repaint updates occur at total rate n: for each ordered pair $(i,j) \in \{1,\dots,n\}^2$ run an independent Poisson clock of rate $1/n$, and when it rings, pair i is drawn from and pair j is repainted, in that order. Let

$$Z_{i,n}(\tau) = \mathbf{1}\{\text{pair } i \text{ unfinished at scaled time } \tau\}, \qquad U_n(\tau) = \sum_{i=1}^{n} Z_{i,n}(\tau),$$

and let $Z_i^*(\tau)$ be the corresponding independent-model indicators.

Lemma 5.4. *For every fixed $r \geq 1$, uniformly for $\tau = O(\ln n)$ with $\tau \to \infty$,*

$$\Pr[Z_{1,n}(\tau) = \cdots = Z_{r,n}(\tau) = 1] = \Pr[T^* > \tau]^r\,(1 + o(1)).$$

Proof. Consider the "killed" process on the first r pairs, where completion of any selected pair kills the process; its transient state space is $\{0,1\}^r$, and $Q^* = \left(\begin{smallmatrix} -1 & 1 \\ 1 & -3/2 \end{smallmatrix}\right)$ is the one-pair transient generator. If the r pairs evolved independently, the killed generator would be the Kronecker sum $(Q^*)^{\oplus r} = \sum_{a=1}^{r} I^{\otimes(a-1)} \otimes Q^* \otimes I^{\otimes(r-a)}$. Let $A_{r,n}$ be the actual killed generator, with D_a, R_a the draw/repaint operators on coordinate a and H_{ab} the one-event operator in which coordinate a is drawn and coordinate b repainted in the same global update, so $(Q^*)^{\oplus r} = \sum_a (D_a + R_a)$ and

$$A_{r,n} = \left(1 - \tfrac{r}{n}\right) \sum_{a=1}^{r} (D_a + R_a) + \frac{1}{n} \sum_{a,b=1}^{r} H_{ab} = (Q^*)^{\oplus r} + \frac{1}{n} B_r, \quad B_r = \sum_{a,b=1}^{r} (H_{ab} - D_a - R_b).$$

Intuitively, $A_{r,n}$ is a sum of two parts: the first term accounts for updates (with rate one) where an event on one of r pairs interacts with another event outside these pairs (which happens with probability $1 - \frac{r}{n}$), and the second term captures updates (with rate $\frac{1}{n}$) with both events happening on chosen r pairs.

The local matrices are

$$H_{aa} - D_a - R_a = \begin{pmatrix} 1 & -1 \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, \qquad H_{ab} - D_a - R_b = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \ (a \neq b),$$

so $\|B_r\| \leq r \cdot 2 + r(r-1) \cdot 4 = 4r^2 - 2r = O_r(1)$. The matrix Q^* has eigenvalues $-\alpha, -\beta$ with $\beta > \alpha$, so $(Q^*)^{\oplus r}$ has a simple principal eigenvalue $-r\alpha$ with all others of real part at most $-r\alpha - (\beta - \alpha)$ [21, Thm. 4.2.12]. As $A_{r,n}$ is a finite-dimensional $O(1/n)$ perturbation, standard spectral perturbation theory [34, Thm. 2.3] gives a simple principal eigenvalue $-\lambda_{r,n}$ and coefficient $a_{r,n}$ with $\lambda_{r,n} = r\alpha + O_r(1/n)$, $a_{r,n} = C_\alpha^r + O_r(1/n)$, and

$$\Pr[Z_{1,n} = \cdots = Z_{r,n} = 1] = a_{r,n} e^{-\lambda_{r,n}\tau} + O_r\!\left(e^{-(r\alpha + (\beta-\alpha)/2)\tau}\right).$$

For $\tau = O(\ln n)$, $e^{-\lambda_{r,n}\tau} = e^{-r\alpha\tau} e^{O_r(\tau/n)} = e^{-r\alpha\tau}(1 + o(1))$ and $a_{r,n} = C_\alpha^r(1 + o(1))$, so the right-hand side equals $C_\alpha^r e^{-r\alpha\tau}(1 + o(1)) = \Pr[T^* > \tau]^r (1 + o(1))$. $\square$

Let $\widehat{T}_n := \inf\{\tau \geq 0 : U_n(\tau) = 0\}$ be the completion time of the actual poissonized process in scaled time.

Theorem 5.5. *Let $Y_n = \dfrac{\widehat{T}_n - b_n}{c_n}$ with $b_n = \frac{1}{\alpha}\ln(nC_\alpha)$, $c_n = \frac{1}{\alpha}$. Then Y_n converges in expectation to a standard Gumbel variable G, and*

$$\mathbb{E}[\widehat{T}_n] = \frac{1}{\alpha}\left(\ln n + \ln C_\alpha + \gamma\right) + o(1).$$

Proof. Fix $x \in \mathbb{R}$. By Theorem 5.1, $\lim_n n\Pr[T^* > b_n + c_n x] = e^{-x}$. For each fixed r, Lemma 5.4 gives the r-th factorial moment

$$\lim_n \mathbb{E}\big[(U_n(b_n + c_n x))_r\big] = \lim_n (n)_r \Pr[Z_{1,n} = \cdots = Z_{r,n} = 1] = \lim_n (n)_r \Pr[T^* > b_n + c_n x]^r = e^{-rx},$$

where $(n)_r = n(n-1)\cdots(n-r+1)$. These are the factorial moments of a Poisson(e^{-x}) law, which is moment-determinate, so $U_n(b_n + c_n x) \xrightarrow{d} \mathrm{Poi}(e^{-x})$ and

$$\Pr[\widehat{T}_n \le b_n + c_n x] = \Pr[U_n(b_n + c_n x) = 0] \to e^{-e^{-x}}.$$

Convergence of expectations follows from uniform integrability via lower and upper tail bounds as in the independent case (Lemma 5.2), whence $\mathbb{E}[\widehat{T}_n] = b_n + c_n\, \mathbb{E}[Y_n] + o(1) = \frac{1}{\alpha}(\ln n + \ln C_\alpha + \gamma) + o(1)$. In particular $n\,\mathbb{E}[\widehat{T}_n] = \frac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma) + o(n)$, matching the independent-pair prediction. $\square$

5.4 Depoissonization

We now transfer the scaled poissonized result to the original discrete-time chain. Let $F_n(m) = \Pr[T_n \le m]$, which is nondecreasing in m since $(0, n)$ is absorbing, and define its Poisson transform (see [22, 35])

$$\widehat{F}_n(t) = e^{-t} \sum_{m \ge 0} F_n(m) \frac{t^m}{m!}.$$

If $N_t \sim \mathrm{Poi}(t)$ is independent of the update sequence then $\widehat{F}_n(t) = \Pr[T_n \le N_t]$, the completion distribution of the discrete chain run under a rate-one Poisson clock; equivalently $\widehat{F}_n(t) = \Pr[n\widehat{T}_n \le t]$.

Lemma 5.6. *Let $F : \mathbb{N} \to [0,1]$ be nondecreasing with Poisson transform $\widetilde{F}(t) = e^{-t}\sum_{k\ge0} F(k)t^k/k!$. For $m \in \mathbb{N}$ and $h > 0$, let $N_- \sim \mathrm{Poi}(m-h)$ and $N_+ \sim \mathrm{Poi}(m+h)$. Then*

$$\widetilde{F}(m-h) - \Pr[N_- > m] \;\le\; F(m) \;\le\; \widetilde{F}(m+h) + \Pr[N_+ < m].$$

Proof. The upper bound follows from $\widetilde{F}(m+h) = \mathbb{E}[F(N_+)] \ge \mathbb{E}[F(N_+)\mathbf{1}_{N_+ \ge m}] \ge F(m)\Pr[N_+ \ge m] \ge F(m) - \Pr[N_+ < m]$, and the lower bound from $\widetilde{F}(m-h) = \mathbb{E}[F(N_-)] \le \mathbb{E}[F(N_-)\mathbf{1}_{N_- \le m}] + \Pr[N_- > m] \le F(m) + \Pr[N_- > m]$. $\square$

Pick $m = n(b_n + c_n x) + O(1)$ and any sequence h_n with $h_n = o(n)$ and $h_n = \omega(\sqrt{n}\ln n)$. The first condition keeps $(m \pm h_n)/n$ in the same $O(1)$ critical window as m/n; the second, with Chernoff bounds for Poisson variables, gives $\Pr[N_+ < m] \to 0$ and $\Pr[N_- > m] \to 0$ for $N_\pm \sim \mathrm{Poi}(m \pm h_n)$. Lemma 5.6 then yields

$$\widehat{F}_n(m - h_n) - o(1) \le F_n(m) \le \widehat{F}_n(m + h_n) + o(1),$$

and since $\widehat{F}_n(m \pm h_n) \to \exp(-e^{-x})$ by Theorem 5.5,

$$\lim_{n\to\infty} \Pr\left[\frac{T_n/n - b_n}{c_n} \le x\right] = \exp(-e^{-x}).$$

For the expectation no separate argument is needed. Let $\widetilde{T}_n^{\mathrm{emb}}$ be the completion time of the embedded discrete chain run under a rate-one Poisson clock. Conditional on $T_n = m$, $\widetilde{T}_n^{\mathrm{emb}}$ is a sum of m independent mean-one exponentials, so $\mathbb{E}[\widetilde{T}_n^{\mathrm{emb}} \mid T_n] = T_n$ and hence $\mathbb{E}[\widetilde{T}_n^{\mathrm{emb}}] = \mathbb{E}[T_n]$. But $\widehat{T}_n = \widetilde{T}_n^{\mathrm{emb}}/n$, just with different scaling. Combining,

$$\boxed{\mathbb{E}[T_n] = n\;\mathbb{E}[\widehat{T}_n] = \frac{n}{\alpha}\big(\ln n + \ln C_\alpha + \gamma\big) + o(n),}$$

with leading constant $\dfrac{1}{\alpha} = \dfrac{4}{5-\sqrt{17}} = \dfrac{5+\sqrt{17}}{2} \approx 4.5616$. The sandwich uses only monotonicity of F_n and concentration of Poisson variables; all dependence between pairs was handled in the poissonized analysis.

Remark 5.7. *The leading constant* 4.5616 *lies strictly between the elementary bounds* 2 *and* 12 *of Section 4. Direct simulation of the discrete process is consistent with the full prediction* $\frac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma)$*, the ratio of empirical mean to prediction approaching* 1 *as* n *grows.*

6 Experiments

To corroborate the leading-order result we simulated the discrete draw-and-repaint chain directly: at each step a uniformly random coupon is drawn and a uniformly random pair is recolored, tracking each pair's status in $\{0, 1, 2\}$, and T_n is recorded as the first step at which all n pairs are completed. Table 1 reports the empirical mean $\widehat{\mathbb{E}}[T_n]$ (with standard error) over independent trials, against the prediction $\frac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma)$ of Section 5. The ratio of the two is within 0.5% of 1 for every $n \geq 80$, comfortably inside the residual $o(n)$ term.

n	trials	$\widehat{\mathbb{E}}[T_n]$ (s.e.)	$\dfrac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma)$	ratio
100	10 000	2430.5 (5.8)	2410.1	1.008
200	10 000	5460.7 (11.4)	5452.5	1.001
400	10 000	12178.9 (23.5)	12169.7	1.001
800	10 000	26838.1 (46.0)	26868.9	0.999
1600	10 000	58687.6 (93.5)	58796.8	0.998
3200	10 000	127686.4 (185.7)	127711.5	1.000
6400	10 000	276000.8 (375.9)	275658.7	1.001
12800	10 000	592466.1 (743.5)	591788.7	1.001
25600	10 000	1266075.9 (1512.7)	1264520.2	1.001
51200	10 000	2687479.0 (3007.3)	2690926.0	0.999

Table 1: Empirical mean completion time of the dynamic double coupon collector process ($k = 2$) versus the leading-order prediction. Each row reports the empirical mean $\widehat{\mathbb{E}}[T_n]$ (with standard error in parentheses) over 10 000 independent trials, the theoretical value $\frac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma)$, and their ratio, which stays within about 1% of unity across the whole range of n.

Two finer diagnostics from Table 1 confirm our theoretical statements. First, the normalized mean $\widehat{\mathbb{E}}$ converges to the Gumbel mean $\gamma \approx 0.5772$, in agreement with Theorem 5.5. Second, we also verify on a smaller set of data that the empirical distribution of $Y_n = (T_n/n - b_n)/c_n$ tracks the Gumbel law $e^{-e^{-x}}$ closely already at $n = 320$ (Figure 1(a)), while the empirical means lie essentially on the predicted curve across the whole range of n (even for relatively small n) tested, between the elementary bounds of Section 4 (Figure 1(b)).

7 Conclusion

In this paper, we have introduced and analyzed the dynamic double coupon collector problem and shown how it can be applied to randomized alteration algorithms that involve elements that are grouped into pairs, which correspond to pairs of colored coupons in the coupon collector formulation. This approach

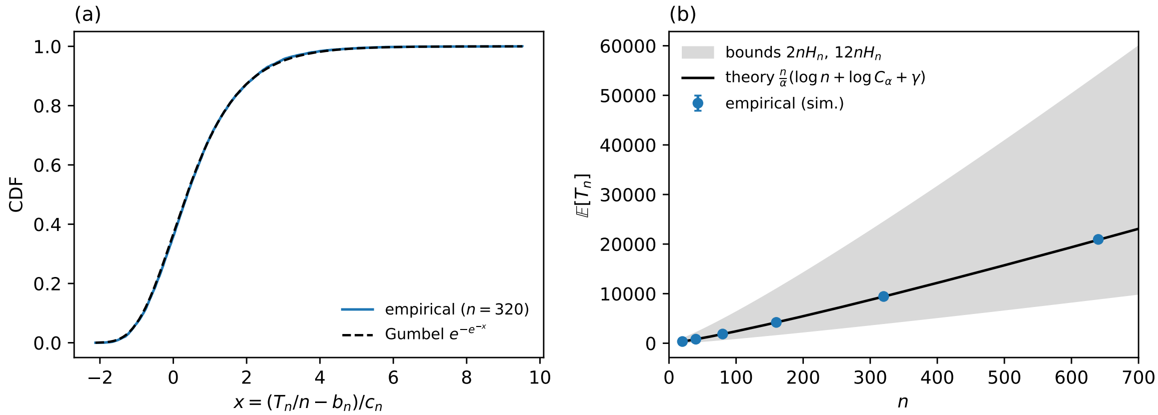

Figure 1: (a) Empirical CDF of the normalized completion time $Y_n = (T_n/n - b_n)/c_n$ at $n = 320$ (10^4 trials) against the standard Gumbel CDF $e^{-e^{-x}}$. (b) Empirical mean $\widehat{\mathbb{E}}[T_n]$ (markers, error bars within the symbols) against the theoretical prediction $\frac{n}{\alpha}(\ln n + \ln C_\alpha + \gamma)$ (solid), with the simple bounds $2nH_n$ and $12nH_n$ of Section 4 shaded.

in principle could be generalized for a problem with kn colored coupons organized into n groups of k elements each, where we draw one coupon at random and recolor one group. Assuming k is constant, one would find generators Q_k and Q_k^* for a poissonized process for a single group, find $-\alpha_k$ as the eigenvalue of Q_k^* with a smallest real part, prove completion time for k independent groups, and finally, after taking into account weak dependencies, claim that the original process matches asymptotic $\frac{n}{\alpha_k}(\ln n + \ln C_{\alpha_k} + \gamma)$. Since there are no known general formulas for Q_k and α_k, however, we leave finding the closed-form asymptotics as a function of k as an open problem.

Acknowledgments

During the preparation of this work, the author(s) used OpenAI ChatGPT for verification of some of the proofs. After using this tool, the author(s) amended proofs as needed and take(s) full responsibility for their contents. The author(s) also used Generative AI to assist in creating software for experiments.

References

[1] Ramtin Afshar, Michael T. Goodrich, Pedro Matias, and Martha C. Osegueda. Parallel network mapping algorithms. In *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 410–413, 2021.

[2] Ramtin Afshar, Michael T. Goodrich, and Evrim Ozel. Efficient exact learning algorithms for road networks and other graphs with bounded clustering degrees. In *20th International Symposium on Experimental Algorithms (SEA 2022)*, pages 9–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.

[3] Susanne Albers. Online algorithms: A survey. *Mathematical Programming*, 97(1):3–26, 2003.

[4] Aris Anagnostopoulos, Ravi Kumar, Mohammad Mahdian, and Eli Upfal. Sorting and selection on dynamic data. *Theoretical Computer Science*, 412(24):2564–2576, 2011.

[5] Søren Asmussen. *Applied Probability and Queues*. Springer, 2 edition, 2003.

[6] Juan Jose Besa, William E. Devanny, David Eppstein, Michael T. Goodrich, and Timothy Johnson. Optimally Sorting Evolving Data. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81:1–81:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[7] Juan Jose Besa, William E. Devanny, David Eppstein, Michael T. Goodrich, and Timothy Johnson. Quadratic time algorithms appear to be optimal for sorting evolving data. In *2018 Proceedings of the Meeting on Algorithm Engineering and Experiments (ALENEX)*, pages 87–96, 2018.

[8] Leonid Bolotnyy and Gabriel Robins. Generalized "Yoking-Proofs"' for a group of RFID tags. In *2006 Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pages 1–4, 2006.

[9] Aristides V. Doumas and Vassilis G. Papanicolaou. The coupon collector's problem revisited: generalizing the double dixie cup problem of newman and shepp. *ESAIM: Probability and Statistics*, 20:367–399, 2016.

[10] Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 5 edition, 2019.

[11] Paul Embrechts, Claudia Klüppelberg, and Thomas Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer, 1997.

[12] David Eppstein, Michael T. Goodrich, Michael Mitzenmacher, and Manuel R. Torres. 2-3 cuckoo filters for faster triangle listing and set intersection. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 247–260, 2017.

[13] Paul Erdős and Alfréd Rényi. On a classical problem of probability theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:215–220, 1961.

[14] Bin Fan, David G. Andersen, Michael Kaminsky, and Michael D. Mitzenmacher. Cuckoo filter: Practically better than bloom. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies (CoNEXT '14)*, pages 75–88, New York, NY, USA, 2014. ACM.

[15] William Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1*. Wiley, 3 edition, 1968.

[16] Marco Ferrante and Monica Saltalamacchia. The coupon collector's problem. *Materials Matematics*, 2014(2):1–35, 2014.

[17] Philippe Flajolet, Danièle Gardy, and Loÿs Thimonier. Birthday paradox, coupon collectors, caching algorithms and self-organizing search. *Discrete Applied Mathematics*, 39(3):207–229, 1992.

[18] George Giakkoupis, Marcos Kiwi, and Dimitrios Los. Naively sorting evolving data is optimal and robust. In *IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2217–2242. IEEE, 2024.

[19] Michael T. Goodrich. Answering spatial multiple-set intersection queries using 2-3 cuckoo hash-filters. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–4, 2017.

[20] Michael T. Goodrich, Songyu Liu, and Ioannis Panageas. Exact learning of weighted graphs using composite queries. In *International Workshop on Combinatorial Algorithms*, pages 103–117. Springer, 2025.

[21] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[22] Philippe Jacquet and Wojciech Szpankowski. Analytical depoissonization and its applications. *Theoretical Computer Science*, 201(1-2):1–62, 1998.

[23] Ari Juels. “Yoking-proofs” for RFID tags. In *IEEE Annual Conference on Pervasive Computing and Communications Workshop*, pages 138–143, 2004.

[24] Sampath Kannan, Claire Mathieu, and Hang Zhou. Graph reconstruction and verification. *ACM Transactions on Algorithms (TALG)*, 14(4):1–30, 2018.

[25] Timo Kötzing and Martin S. Krejca. First-hitting times under drift. *Theoretical Computer Science*, 796:51–69, 2019.

[26] Claire Mathieu and Hang Zhou. A simple algorithm for graph reconstruction. *Random Structures & Algorithms*, 63(2):512–532, 2023.

[27] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, 2 edition, 2017.

[28] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[29] Amy N. Myers and Herbert S. Wilf. Some new aspects of the coupon collector's problem. *SIAM Review*, 48(3):549–565, 2006.

[30] Peter Neal. The generalised coupon collector problem. *Journal of Applied Probability*, 45(3):621–629, 2008.

[31] Donald J. Newman and Lawrence Shepp. The double dixie cup problem. *The American Mathematical Monthly*, 67(1):58–61, 1960.

[32] Mark H. Overmars. *The Design of Dynamic Data Structures*. Springer, 1983.

[33] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *Journal of Algorithms*, 51(2):122–144, 2004.

[34] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[35] Wojciech Szpankowski. *Average Case Analysis of Algorithms on Sequences*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 2001.

[36] Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.