

A GENERALIZED COUPON COLLECTOR PROBLEM

WEIYU XU *** AND

A. KEVIN TANG, * ** *Cornell University*

Abstract

This paper presents an analysis of a generalized version of the coupon collector problem, in which the collector receives d coupons each run and chooses the least-collected coupon so far. In the asymptotic case when the number of coupons n goes to infinity, we show that, on average, $(n \log n)/d + (n/d)(m - 1) \log \log n + O(mn)$ runs are needed to collect m sets of coupons. An exact algorithm is also developed for any finite case to compute the exact mean number of runs. Numerical examples are provided to verify our theoretical predictions.

Keywords: Coupon collector problem; expected run; state-space representation; wireless communication; opportunistic scheduling

2010 Mathematics Subject Classification: Primary 60G70

Secondary 60C05

1. Introduction

The classic coupon collector problem asks the following question: Given that a collector randomly receives a coupon each run, how many runs are necessary to collect a complete set of n different coupons? The answer is nH_n , where $H_n = \sum_{k=1}^n 1/k$ is the harmonic number [2]. One can further ask how many runs are necessary to collect m complete sets of coupons. This question has been addressed by Newman and Shepp [9].

The coupon collector problem and its variants are of traditional and recurrent interest [3], [4], [5], [7], [8]. Besides their rich theoretical structures and implications, they have various applications, including dynamic resource allocation, hashing, and online load balancing [1], to name just a few. In particular, these problems also serve as basic models to analyze the delay for opportunistic scheduling in broadcast wireless fading channels [10]. For example, to maximize system throughput, we should serve to the user whose channel condition is the best at every time slot. In order to evaluate performance, we could determine the expected number of time slots needed for all users to be served at least once. Assuming that all channels are independent and identically distributed, this is equivalent to the classic coupon collector problem.

In this paper we investigate a natural generalization of the coupon collector problem. Instead of receiving one coupon, the collector receives d ($1 \leq d \leq n$) distinct coupons randomly each run and chooses the least-collected coupon so far. Formally, we denote the number of runs (a ‘run’ is often referred to as a ‘time slot’ or simply as a unit of ‘time’ in this paper) necessary to collect m sets of coupons as $D_{m,n}^d$. We are interested in characterizing the mean value of the random variable $D_{m,n}^d$, especially in the asymptotic region when n is large. Clearly, when

Received 25 November 2010; revision received 11 June 2011.

* Postal address: School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853, USA.

** Email address: wx42@cornell.edu

*** Email address: atang@ece.cornell.edu

$d = 1$, we go back to the classic cases; when $d = n$, there is no randomness and $D_{m,n}^n = mn$. In the scheduling transmission context discussed above, d can be viewed as a parameter that controls the trade-off between efficiency (high throughput) and fairness among all users, with $d = 1$ purely focusing on efficiency while $d = n$ corresponds to perfect fairness. Essentially, we are interested in investigating ‘the power of multiple choices’ in speeding up the wireless scheduling; it is worth mentioning that ‘the power of two choices’ has been investigated in the randomized load balancing setting, where ‘two choices’ can exponentially reduce the maximum load; see, for example, [6].

The remainder of this paper is organised as follows. We first briefly review existing related results in Section 2. Although they are all special cases of the general problem, the techniques used to derive them cannot be applied directly to the general case. Instead, we develop a new technique to characterize $E(D_{m,n}^d)$, and provide upper and lower bounds for $E(D_{m,n}^d)$ in Section 3 and Section 4. An asymptotic analysis shows that the upper bound and lower bound match in the asymptotic regime of $n \rightarrow \infty$ in Section 5. Furthermore, for any finite n , an algorithm is motivated and proposed in Section 6 to calculate $E(D_{m,n}^d)$ exactly. We use numerical examples to validate our theoretical predictions in Section 7. Concluding remarks and future work are given in Section 8.

2. Existing results

Existing results for special cases are listed below. If $d = 1$ then the problem is solved for all $m \geq 1$. If $d > 1$ then only the $m = 1$ case is known.

- $d = 1$ and $m = 1$ (see [2]). It is clear that the number of runs needed to obtain the $(i + 1)$ th coupon after obtaining the i th coupon follows a geometric distribution with parameter $(n - i)/n$. Therefore,

$$E(D_{1,n}^1) = nH_n = n \sum_{k=1}^n \frac{1}{k}.$$

For large n ,

$$E(D_{1,n}^1) = n \log n + nO(1).$$

We see that the randomness cost is expressed approximately by a factor $\log n$.

- $d = 1$ and $m \geq 1$ (see [9]). We have

$$E(D_{m,n}^1) = n \int_0^\infty (1 - (1 - S_m(t)e^{-t})^n) dt,$$

where

$$S_m(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!}.$$

For fixed m and large n ,

$$E(D_{m,n}^1) = n \log n + n(m - 1) \log \log n + nO(1).$$

It is interesting to note that although collecting the first set of coupons needs about $n \log n$ runs, all later sets need only $n \log \log n$ runs per set.

- $d \geq 1$ and $m = 1$ (see [10]). This has applications in the scheduling of data packet transmission over wireless channels. Here

$$E(D_{1,n}^d) = \sum_{i=0}^{n-1} \frac{1}{1 - \binom{i}{d} / \binom{n}{d}},$$

where $\binom{i}{d} = 0$ if $i < d$.

For fixed m and large n ,

$$E(D_{1,n}^d) \sim \frac{1}{d} n \log n.$$

This shows that, for the $m = 1$ case, choosing d coupons randomly each time decreases the expected number of runs, with the greatest reduction occurring between $d = 1$ and $d = 2$.

- $d \geq 1$ and $m \geq 1$. In the context of scheduling, a transmitter who wants to send m packets to each of the n users can only transmit one packet to one user chosen from the d users who have the best wireless communication channels. Owing to the time varying nature of the wireless channels, it is natural to assume that, for each time index, the d users who have the best communication channels are uniformly distributed among the n users. So $E(D_{m,n}^d)$ gives an estimate on the total delay in delivering these m packets, and, in this paper, we will offer a characterization of $E(D_{m,n}^d)$.

3. Lower bound on $E(D_{m,n}^d)$

We will first lower bound $E(D_{m,n}^d)$ by considering a different coupon collecting process. In this new process, each time, we uniformly select d distinct coupons out of n coupons, instead of keeping only one coupon out of these d selected coupons, we would keep all the d coupons. The expected time of collecting m sets of coupons in this way will be no larger than the process in which we keep only one coupon each. However, it is not so straightforward to directly obtain an estimate for this new process. This motivates us to consider another process in which each time we collect d uniformly, independently chosen (allowing repetition) coupons we keep all of them. This process stops when m sets of coupons are fully collected.

Lemma 1. *Let t_1 be the expected time to collect m sets of coupons for the process in which each time d uniformly chosen distinct coupons are kept. Let t_2 be the expected time to collect m sets of coupons for the process in which each time d uniformly chosen (allowing repetition) coupons are kept. Then*

$$t_1 \geq \frac{\binom{n}{d}}{n^d} t_2. \tag{1}$$

Proof. We simulate the process of choosing d distinct coupons through an expurgated process of choosing d independent coupons (allowing repetition). If the d coupons we independently choose (allowing repetition) are not distinct, we will discard this group of d coupons; if the d coupons are all distinct, we will keep them. The kept coupons from the expurgated process follow the same distribution as the chosen d distinct coupons. However, the expected time to collect a group of d distinct coupons is clearly $n^d / \binom{n}{d}$. So in the worst case, $t_2 \leq n^d t_1 / \binom{n}{d}$.

In summary, in order to give a lower bound on $E(D_{m,n}^d)$, we first need a lower bound on t_2 for the process of keeping d uniformly, randomly chosen coupons (allowing repetition). To do this, we follow the approach of generating functions in [9].

Let p_i be the probability of failing to obtain m sets of coupons when we have kept i coupons. Let P_{x_1, \dots, x_n} be a power series, and let $\{P_{x_1, \dots, x_n}\}$ be the power series when all terms having exponents greater than or equal to m have been removed. Thus,

$$t_2 = \sum_{j=0}^{\infty} p_{dj}$$

and

$$p_{dj} = \frac{\{(x_1 + \dots + x_n)^{dj}\}}{n^{dj}},$$

with x_1, \dots, x_n all equal to 1.

In addition, we know that

$$E(D_{m,n}^1) = \sum_{q=0}^{d-1} \sum_{j=0}^{\infty} p_{dj+q} = n \int_0^{\infty} (1 - (1 - S_m(t)e^{-t})^n) dt,$$

where $S_m(t) = \sum_{k=0}^{m-1} t^k/k!$ [9].

We also note that p_i is nonincreasing as i grows, so

$$t_2 = \sum_{j=0}^{\infty} p_{dj} \geq \frac{1}{d} \left(\sum_{j=0}^{\infty} p_j \right) \geq \frac{n}{d} \int_0^{\infty} (1 - (1 - S_m(t)e^{-t})^n) dt.$$

So by (1) we know that

$$E(D_{m,n}^d) \geq \frac{\binom{n}{d}}{n^d} \frac{E(D_{m,n}^1)}{d}.$$

4. Upper bound on $E(D_{m,n}^d)$

In this section we upper bound the expected time necessary to collect m complete sets of coupons. To achieve this, we upper bound the expected time necessary to collect m complete sets of coupons in a suboptimal process. In this new process, each time, we uniformly and independently choose d coupons (allowing repetition). Among this group of d coupons, if the i th ($1 \leq i \leq d$) coupon is the first coupon to so far have fewer than m copies then we will keep this coupon and discard the remaining $d - i$ coupons.

First, we observe that d distinct coupons are favorable in terms of minimizing the collection time compared to d coupons with possible repetition.

Theorem 1. *The minimized expected time of collecting m sets of coupons when the collector is given d uniformly chosen, distinct coupons each run, but is allowed to keep only one coupon, is no larger than the minimized expected time of collecting m sets of coupons when the collector is given d uniformly chosen coupons (allowing repetition) each run, but is allowed to keep only one coupon.*

Proof. We consider two coupled coupon collecting processes: process P , which receives d independent, uniformly chosen coupons (allowing repetition) each run, and process Q , which receives d distinct coupons each run. For each run, process Q sequentially receives d distinct coupons in this way: if the i th ($1 \leq i \leq d$) received coupon of process P is a repetition of one of the first $i - 1$ received coupons of process Q , the i th received coupon of process Q will be

uniformly chosen among the unreceived $n - (i - 1)$ coupons of process Q in this run; otherwise, the i th ($1 \leq i \leq d$) received coupon of process Q will be the same as the i th ($1 \leq i \leq d$) received coupon of process P . Since in each run all types of coupon received in process P are also received in process Q , the coupon collector using process Q can keep the same coupon as in process P and, thus, has no larger collecting time than using process P .

Second, we show that it is an optimal strategy for the coupon collector to keep the least-collected coupon out of the d incoming coupons (whether allowing repetition or not).

Theorem 2. *The expected time to collect m sets of coupons is minimized when the collector keeps the least-collected coupon each run, if the collector is allowed to keep only one out of the d offered coupons.*

Proof. Suppose that we have c_1 coupons of type j ($1 \leq j \leq n$) and c_2 coupons of type l ($1 \leq l \leq n$), where $j \neq l$ and $c_1 < c_2 < m$. Now consider two coupon collectors P and Q . From an incoming set of d coupons which contain both type j and type l , P keeps a coupon of type l , and we use state A , represented by the tuple $\{c'_1, c'_2\} = (c_1, c_2 + 1)$, to record the numbers of type- j and type- l coupons kept by P . Collector Q instead keeps a coupon of type j , and we correspondingly use state B , represented by $\{c''_1, c''_2\} = (c_1 + 1, c_2)$.

Now we need only to argue that, to collect m sets of coupons, on average, collector Q , who starts from state B , will take no longer than collector P , who starts from state A . In each run, P and Q receive the same set of d coupons, and a collector can keep a coupon only if he/she has fewer than m copies of that type. We consider the ‘following-and-updating’ process described below.

We let P make his/her optimal keeping decision each run to minimize the expected collecting time. Then we let collector Q keep the same type of coupon as collector P . For a run where collectors P and Q decide to keep a coupon of type j or type l , we update their states A and B . There are two possible cases.

In the first case, $c''_1 = c'_2$ and it is the coupon of type j that both collectors decide to keep in that run. Then P and Q are in states $\{c'_1 + 1, c'_2\}$ and $\{c''_1 + 1, c''_2\}$, respectively. We update state A to $\{c'_1 + 1, c'_2\}$, but instead symmetrically update the state of Q to $B = \{c''_2, c''_1 + 1\}$, which has the same optimized expected collecting time as state $\{c''_1 + 1, c''_2\}$. If $c''_1 + 1 = c'_2$, A and B are two equivalent states (note that $c'_1 + c'_2 = c''_1 + c''_2$); thus, we achieve our objective.

In the second case, after keeping a new coupon of type j or type l , we simply update states A and B to record the new numbers of type- j and type- l coupons.

Before the states of the two collectors become equivalent, the collectors repeat the ‘following-and-updating’ process described above, starting from new states A and B each iteration. With a little abuse of notation, we still use $\{c'_1, c'_2\}$ and $\{c''_1, c''_2\}$ to represent states A and B before each iteration, even though the values of $c'_1, c'_2, c''_1,$ and c''_2 may have changed over iterations. It is not hard to see that the following conditions always hold before each iteration:

$$\begin{aligned} c'_1 + c'_2 &= c''_1 + c''_2, & c'_1 &\leq c'_2, & c''_1 &\leq c''_2, \\ c'_2 - c'_1 &> c''_2 - c''_1, & c'_1, c'_2, c''_1, c''_2 &\leq m. \end{aligned}$$

Because of these conditions, in each run, when P keeps a certain type of coupon, Q can also follow P 's decision and keep the same type of coupon.

In each iteration, we always increase $c'_1 + c'_2$ and $c''_1 + c''_2$, and because $c'_1, c'_2, c''_1, c''_2 \leq m$, we will eventually obtain a pair of equivalent states in some iteration. Since we can always arrive at two equivalent states for the two coupon collectors, starting from B to collect m sets of coupons will not take any longer than starting from A .

Now we prepare a final lemma before presenting the upper bound on $E(D_{m,n}^d)$.

Lemma 2. *The function*

$$f(i) = \frac{n}{di} - \frac{1}{1 - (1 - i/n)^d}$$

is nonincreasing in $i \in [1, n]$, and $1/d - 1 \leq f(i) \leq 0$ for $1 \leq i \leq n$.

Proof. We need to show that, for $i \in [1, n]$, the following derivative is nonpositive:

$$f'(i) = \frac{d(1 - i/n)^{d-1}}{n(1 - (1 - i/n)^d)^2} - \frac{n}{di^2}.$$

Let $g(x) = (1 - x)^d + dx(1 - x)^{(d-1)/2}$, so

$$g'(x) = -\frac{1}{2}d(1 - x)^{(d-3)/2}(x + dx + 2(1 - x)^{(d+1)/2} - 2).$$

Also, let $h(x) = x + dx + 2(1 - x)^{(d+1)/2} - 2$, so

$$h'(x) = 1 + d - (1 + d)(1 - x)^{(d-1)/2} \geq 0,$$

and $h(x) \geq h(0) = 0$ for $x > 0$. Because

$$g'(x) = -\frac{1}{2}d(1 - x)^{(d-3)/2}h(x) \leq 0 \quad \text{for } 0 < x \leq 1,$$

we have $g(x) \leq g(0) = 1$ for $0 < x \leq 1$. This translates into

$$1 - (1 - x)^d \geq dx(1 - x)^{(d-1)/2} \quad \text{for } 0 < x \leq 1,$$

so

$$\frac{d(1 - x)^{d-1}}{n(1 - (1 - x)^d)^2} \leq \frac{1}{ndx^2} \quad \text{for } 0 < x \leq 1.$$

Plugging in $x = i/n$, we have

$$\frac{d(1 - i/n)^{d-1}}{n(1 - (1 - i/n)^d)^2} \leq \frac{n}{di^2} \quad \text{for } 1 \leq i \leq n.$$

So $f'(i) \leq 0$ for $1 \leq i \leq n$. Calculating $f(n)$, we have

$$\frac{n}{di} - \frac{1}{1 - (1 - i/n)^d} \geq \frac{1}{d} - 1 \quad \text{for } 1 \leq i \leq n.$$

At this point, we are ready to present the following upper bound for $E(D_{m,n}^d)$.

Theorem 3. *Suppose that the coupon collector is given d uniformly randomly chosen d distinct coupons and that he/she is allowed to keep only one out of these d distinct coupons. Then the expected time $E(D_{m,n}^d) \leq E(D_{m,n}^1)/d + mn(1 - 1/d)$.*

Proof. Inspired by Theorems 1 and 2, we consider the process in which each run d independently chosen coupons (allowing repetition) are presented and among them the coupon collector keeps only the first ‘useful’ coupon which he/she has fewer than m copies of.

We focus on upper bounding the expected finishing time, conditioning on a specific sequence of mn kept ‘useful’ coupons, specified by their type and the order in which they are kept. Suppose that immediately after the r th kept coupon in this ‘keeper’ sequence, there are s types of coupon for which the coupon collector has m copies.

Conditioning on the specified ‘keeper’ coupon sequence, unless an incoming coupon belongs to the s types of coupon for which the collector already has m copies, this incoming coupon must be the $(r + 1)$ th ‘keeper’ coupon. So, given that the ‘keeper’ sequence is known, with probability $(1 - (n - s)/n)^d$, none of the d uniformly chosen (allowing repetition) coupons is the known $(r + 1)$ th ‘keeper’. So, conditioning on the specified ‘keeper’ sequence, the expected time to collect the $(r + 1)$ th ‘keeper’ coupon is

$$E' = \frac{1}{1 - (1 - (n - s)/n)^d}.$$

However, if the coupon collector is offered only one instead of d coupons each run, the expected time to collect the $(r + 1)$ th coupon is $E'' = n/(n - s)$.

By Lemma 2, $E' - E''/d \leq 1 - 1/d$ for any $1 \leq s \leq n$. Since a whole ‘keeper’ sequence S has mn coupons, the total expected time E'_S to collect them, when d coupons (allowing repetition) are offered each run, and the total expected time E''_S , when one coupon is offered each run, satisfy $E'_S - E''_S/d \leq mn(1 - 1/d)$.

We note that the probability that the coupon collector follows a specific sequence of ‘keeper’ coupons is the same no matter whether $d = 1$ or not. This is because, among a batch of d offered coupons, the collector still checks them one by one and keeps only the first coupon that is ‘useful’. So this implies that $E(D_{m,n}^d) \leq E(D_{m,n}^1)/d + mn(1 - 1/d)$ for any d .

We remark that the upper bound in Theorem 3 can also be obtained by considering the classic coupon collector process and dividing the incoming coupons into batches of d coupons: after each coupon is accepted, the collector throws away the next $d - 1$ coupons to ensure that the next kept coupon belongs to a fresh batch of d coupons. The proof of Theorem 3 reduces to an exact result for the $m = 1$ case, and can lead to tighter upper bounds for general m if a finer analysis is applied (for example, using the exact formula instead of the upper bound in Lemma 2, which is possible at least for collecting the last coupon).

5. An asymptotic analysis ($n \rightarrow \infty$)

In this section we provide an asymptotic analysis for the upper and lower bounds of $E(D_{m,n}^d)$, and we see how this analysis behaves asymptotically for fixed d and m as n goes to ∞ . We will begin with an asymptotic analysis through an exact expression for $E(D_{1,n}^d)$.

Theorem 4. *When n is large enough and $d > 1$,*

$$E(D_{1,n}^d) = n \left(\frac{\log n}{d} + O(1) \right).$$

Proof. We have

$$\begin{aligned} E(D_{1,n}^d) &= \sum_{i=0}^{n-1} \frac{1}{1 - \binom{i}{d} / \binom{n}{d}} \\ &= \sum_{i=0}^{n-1} \frac{1}{1 - i(i-1) \cdots (i-d+1) / n(n-1) \cdots (n-d+1)} \end{aligned}$$

$$\begin{aligned} &\geq d + \sum_{i=d}^{n-1} \frac{1}{1 - ((i - d + 1)/(n - d + 1))^d} \\ &= d + \sum_{i=1}^{n-d} \frac{1}{1 - (1 - i/(n - d + 1))^d}. \end{aligned}$$

Since $(1 - x)^d \geq 1 - dx$ for $0 \leq x \leq 1$,

$$E(D_{1,n}^d) \geq \sum_{i=1}^n \frac{n - d + 1}{di} = \frac{n - d + 1}{nd} \sum_{i=0}^{n-1} \frac{n}{n - i} = \left(1 - \frac{d - 1}{n}\right) \frac{E(D_{1,n}^1)}{d}.$$

Thus,

$$\begin{aligned} E(D_{1,n}^d) &= \sum_{i=0}^{n-1} \frac{1}{1 - i(i - 1) \cdots (i - d + 1)/n(n - 1) \cdots (n - d + 1)} \\ &\leq \sum_{i=0}^{n-1} \frac{1}{1 - (i/n)^d} \\ &= \sum_{i=1}^n \frac{1}{1 - (1 - i/n)^d} \\ &\leq \sum_{i=1}^n \left(\frac{n}{di} - \frac{1}{d} + 1\right) \\ &= \frac{1}{d} E(D_{1,n}^1) + n \left(1 - \frac{1}{d}\right). \end{aligned}$$

Theorem 5. *When m is fixed, for any $d > 1$,*

$$\lim_{n \rightarrow \infty} \frac{E(D_{m,n}^d) - n \log(n)/d}{(n(m - 1) \log \log n)/d} = 1.$$

Proof. From the lower bound and upper bound for $E(D_{m,n}^d)$ in Section 3 and Section 4, we know that

$$\lim_{n \rightarrow \infty} \frac{E(D_{m,n}^d)}{E(D_{m,n}^1)} = \frac{1}{d}.$$

Then the asymptotic expression emerges immediately by recalling the asymptotic expression for $E(D_{m,n}^1)$.

6. An algorithmic approach (for any finite n)

In this section we give an algorithm which calculates exactly $E(D_{m,n}^d)$ for specified m, n , and d based on a state-space representation of the Markov process of collecting the coupons. For each $n_0, n_1, n_2, \dots, n_m \geq 0$ satisfying $n_0 + n_1 + \dots + n_m = n$, define $S_m = (n_0, n_1, \dots, n_m)$ to be the state, where n_i ($0 \leq i \leq m$) is the number of coupons that the coupon collector has

collected i times. Hence, $E(D_{mn}^d)$ is the expected number of runs necessary for the coupon collector to go from state $(n, 0, \dots, 0)$ to state $(0, \dots, 0, n)$.

We now provide an algorithm to calculate $E(D_{m,n}^d)$. Define $N_m^d(S_m)$ to be, starting from state S_m , the number of runs after which m -completed sets of coupons have been collected, i.e. the number of runs from state S_m to $(0, \dots, 0, n)$. Clearly,

$$N_m^d(n, 0, \dots, 0) = D_{m,n}^d$$

and

$$N_m^d(0, \dots, 0, n) = 0.$$

Suppose that we are in state $S_m = (n_0, n_1, \dots, n_m)$. After one run, the transition probability from S_m to the following two states are as follows:

$$\begin{aligned} (n_0, n_1, \dots, n_m) & \text{ with probability } \binom{n_m}{d} / \binom{n}{d}, \\ (n_0, \dots, n_i - 1, n_{i+1} + 1, \dots, n_m) & \text{ with probability } p_i, \quad 0 \leq i < m, \end{aligned}$$

where

$$p_i = \left(\binom{\sum_{t=i}^m n_t}{d} - \binom{\sum_{t=i+1}^m n_t}{d} \right) / \binom{n}{d}.$$

Therefore, we have

$$\begin{aligned} E(N_m^d(n_0, \dots, n_m)) &= 1 + \left(\binom{n_m}{d} / \binom{n}{d} \right) E(N_m^d(n_0, \dots, n_m)) \\ &+ \sum_{i=0}^{m-1} p_i E(N_m^d(n_0, \dots, n_i - 1, n_{i+1} + 1, \dots, n_m)). \end{aligned}$$

So

$$\begin{aligned} E(N_m^d(n_0, \dots, n_m)) &= \left(\binom{n}{d} / \left(\binom{n}{d} - \binom{n_m}{d} \right) \right) \\ &\times \sum_{i=0}^{m-1} \left(\binom{\sum_{t=i}^m n_t}{d} - \binom{\sum_{t=i+1}^m n_t}{d} \right) / \left(\binom{n}{d} - \binom{n_m}{d} \right) \\ &\times E(N_m^d(n_0, \dots, n_i - 1, n_{i+1} + 1, \dots, n_m)). \end{aligned} \tag{2}$$

Define the map

$$\Phi: \{(n_0, \dots, n_m) : n_0, n_1, \dots, n_m \geq 0, n_0 + n_1 + \dots + n_m = n\} \rightarrow \mathbb{N},$$

where

$$\Phi(n_0, n_1, n_2, \dots, n_m) = \sum_{i=0}^m (1+n)^{m-i} n_i.$$

Obviously, Φ is an injection and

$$\Phi(n, 0, \dots, 0) = n(1+n)^m, \quad \Phi(0, \dots, 0, n) = n.$$

Since

$$\begin{aligned} & \Phi(n_0, \dots, n_m) - \Phi(n_0, \dots, n_i - 1, n_{i+1} + 1, \dots, n_m) \\ &= ((1 + n)^{m-i} n_i + (1 + n)^{m-i-1} n_{i+1}) \\ &\quad - ((1 + n)^{m-i} (n_i - 1) + (1 + n)^{m-i-1} (n_{i+1} + 1)) \\ &= (1 + n)^{m-i} - (1 + n)^{m-i-1} \\ &> 0, \end{aligned}$$

by (2), the expected number of runs from state S depends only on the expected number of runs from the states S^* with $\Phi(S^*) < \Phi(S)$. Therefore, we can order all the states (n_0, \dots, n_m) according to the value of $\Phi(n_0, \dots, n_m)$, and compute $E(N_m^d(n_0, \dots, n_m))$ one by one, from the starting state $(0, \dots, 0, n)$ to the last state $(n, 0, \dots, 0)$. The algorithm is described below.

Algorithm 1. (Calculating $E(D_{m,n}^d)$.)

```

for  $n_0 = 0$  to  $n$ 
  for  $n_1 = 0$  to  $n - n_0$ 
     $\vdots$ 
    for  $n_{m-1} = 0$  to  $n - \sum_{i=0}^{m-2} n_i$ 
      do  $n_m = n - \sum_{i=0}^{m-1} n_i$ 
        if  $n_m = n$ 
          then  $E(N_m^d(n_0, \dots, n_m)) = 0$ 
          else use (2) to compute  $E(N_m^d(n_0, \dots, n_m))$ 
  
```

Since the number of nonnegative integer solutions to the equation $n_0 + \dots + n_m = n$ is $\binom{n+m}{n}$, the number of states is $\binom{n+m}{n}$, and the complexity of Algorithm 1 is $O(\binom{n+m}{n})$.

To conclude this section, we now use a simple example ($n = 6$ and $m = 2$) to illustrate Algorithm 1. When $m = 2$, each state has three parameters: n_0, n_1 , and n_2 . Since $n_0 + n_1 + n_2 = n$, we could draw the state transition diagram as in Figure 1. Algorithm 1 computes $E(N_2^d(n_0, n_1, n_2))$ for each state (n_0, n_1, n_2) by the order shown in Figure 2. Note that

- the expected number of runs from any state depends only on the number of runs from its descents in Figure 1,

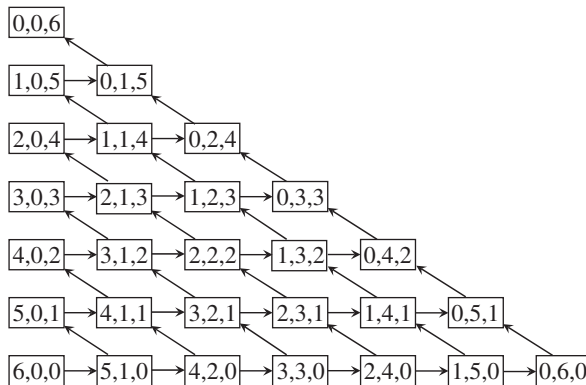


FIGURE 1: State transition diagram for $n = 6$ and $m = 2$. Self-loops are omitted. The nodes are labeled with the values of n_0, n_1 , and n_2 .

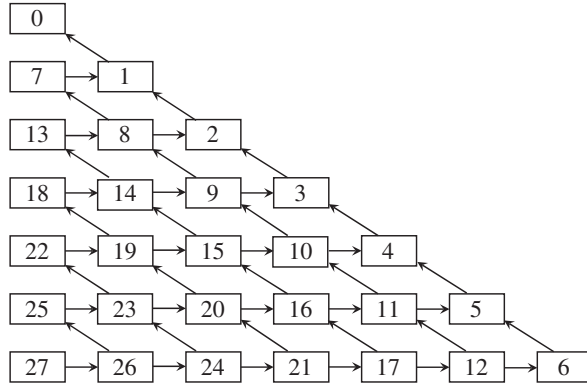


FIGURE 2: State transition diagram for $n = 6$ and $m = 2$. The nodes are labeled with the computation order. The highest node, which represents $(0, 0, 6)$, is labeled 0 because $N_2^d(0, 0, 6)$ is known to be 0.

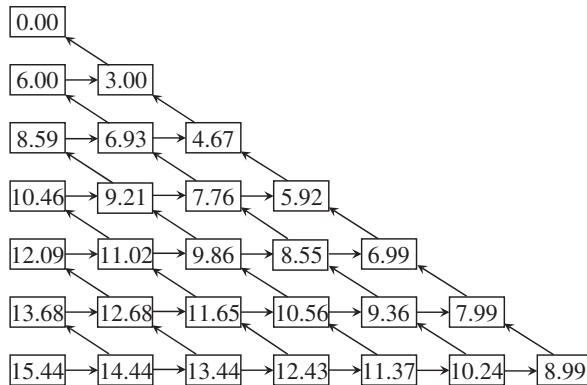


FIGURE 3: State transition diagram for $n = 6$ and $m = 2$. The nodes are labeled with the value of $E(N_2^d(n_0, n_1, n_2))$.

- the computation of $E(N_2^d(\dots))$ for any state is carried out after the computations for its descents by Figure 2.

The values of $E(N_2^d(n_0, n_1, n_2))$ for each state (n_0, n_1, n_2) is shown in Figure 3.

7. Numerical examples

We now engage in numerical exercises to support the results of the last two sections, i.e. the correctness of Algorithm 1 and the derived upper and lower bounds of $E(D_{m,n}^d)$.

7.1. Algorithm

First, we give numerical results for the expected collection time when $n = 100$ and $m = 1, 2, 3$, respectively, in Tables 1–3. The results show that Algorithm 1 gives an expected delay consistent with the simulation results.

TABLE 1: Numerical results for the expected collection time when $m = 1$ and $n = 100$.

	d				
	1	2	3	4	5
Algorithm	518.74	292.93	220.06	184.79	164.27
Simulation	518.69	292.40	219.33	184.59	164.18

TABLE 2: Numerical results for the expected collection time when $m = 2$ and $n = 100$.

	d				
	1	2	3	4	5
Algorithm	728.81	418.69	327.02	286.75	264.84
Simulation	728.20	419.13	327.29	286.81	264.68

TABLE 3: Numerical results for the expected collection time when $m = 3$ and $n = 100$.

	d				
	1	2	3	4	5
Algorithm	910.87	531.34	428.75	386.97	364.86
Simulation	910.09	531.33	428.72	386.65	364.90

7.2. Asymptotic results

Two cases are considered: $d = 3$ and $m = 1$, and $d = 3$ and $m = 2$. For each case, in Figures 4 and 5 we plot the lower bound from Theorem 2, the upper bound from Theorem 1, and the result computed from Algorithm 1 for n from 100 to 500. The results show that the upper and lower bounds bound the expected collecting time very well. In fact, when m and d are fixed, the upper and lower bounds both scale as $(1/d) E(D_{m,n}^1)$ as $n \rightarrow \infty$.

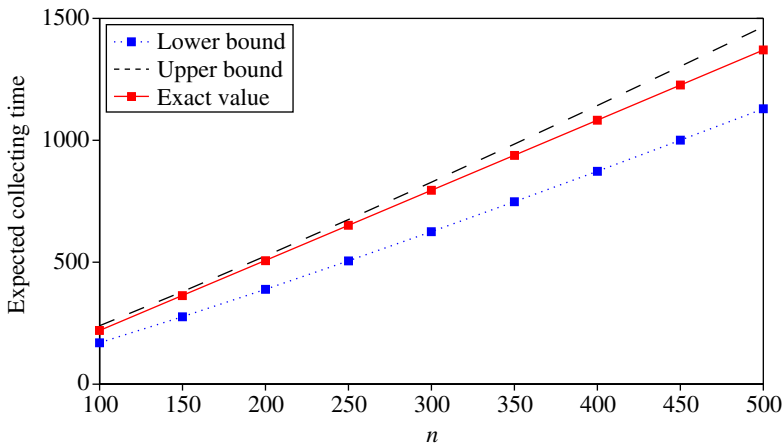


FIGURE 4: Asymptotics for $m = 1$ and $d = 3$.

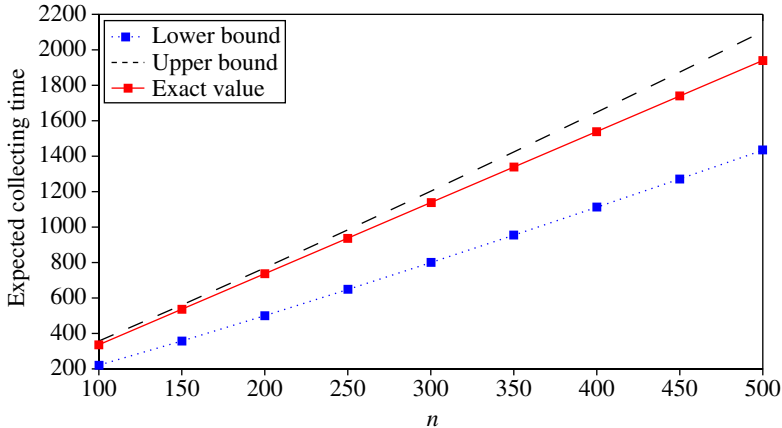


FIGURE 5: Asymptotics for $m = 1$ and $d = 3$.

8. Conclusion and future work

In this paper we have considered a generalized coupon collector problem in which the coupon collector needs to collect $m \geq 1$ sets of coupons and has the freedom to keep one coupon out of the $d \geq 1$ coupons offered each time. We obtained asymptotically matching upper and lower bounds for the expected collection time. We also provided an algorithm to calculate the expected collection time exactly based on a state representation for the coupon collecting process. We should note that, asymptotically, even if the coupon collector is allowed to keep only one coupon out of the d coupons, the needed time will still be shortened by a factor of d , compared to if the coupon collector is allowed to keep all the d coupons offered each time.

There is much scope for future work on this problem. First, one could attempt to obtain a closed-form expression for $E(D_{m,n}^d)$. Second, one could attempt to improve Algorithm 1. Algorithm 1 has a run time of $\binom{n+m}{n}$. To take advantage of this run time requires constant time indexing. The direct approach is to index the states in an n -dimensional matrix of size $(n+1)^m + 1$. However, since there are a total of $\binom{n+m}{n}$ states, a large fraction of the matrix space is not required. Hence, it would be helpful to find an algorithm which carries out triangular indexing in constant time. This would reduce the memory requirements and increase the range of parameters over which the problem is computationally feasible. One could further observe that although there are $\binom{n+m}{n}$ states, only $\binom{n+m-1}{m-1}$ are actually needed at any time. So, with constant time triangular indexing, one could reduce the memory requirements further, although the gain from the second reduction is minimal.

Acknowledgements

The authors would like to thank the anonymous reviewers for insightful comments and suggestions. The authors would also like to thank the input of Xiaojie Gao and Wuhan Desmond Cai.

References

[1] AZAR, Z., BRODER, A. Z., KARLIN, A. R. AND UPFAL, E. (1999). Balanced allocations. *SIAM J. Comput.* **29**, 180–200.
 [2] FELLER, W. (1950). *An Introduction to Probability Theory and Its Applications*. John Wiley, New York.

- [3] FOATA, D. AND ZEILBERGER, D. (2003). The collector's brotherhood problem using the Newman-Shepp symbolic method. *Algebra Universalis* **49**, 387–395.
- [4] HOLST, L. (2001). Extreme value distributions for random coupon collector and birthday problems. *Extremes* **4**, 129–145.
- [5] KAN, N. D. (2005). Martingale approach to the coupon collection problem. *J. Math. Sci.* **127**, 1737–1744.
- [6] MITZENMACHER, M. D. (1996). The power of two choices in randomized load balancing. Doctoral Thesis, University of California, Berkeley.
- [7] MYERS, A. N. AND WILF, H. S. (2003). Some new aspects of the coupon-collector's problem. *SIAM J. Discrete Math.* **17**, 1–17.
- [8] NEAL, P. (2008). The generalised coupon collector problem. *J. Appl. Prob.* **45**, 621–629.
- [9] NEWMAN, D. J. AND SHEPP, L. (1960). The double dixie cup problem. *Amer. Math. Monthly* **67**, 58–61.
- [10] SHARIF, M. AND HASSIBI, B. (2006). Delay considerations for opportunistic scheduling in broadcast fading channels. *IEEE Trans. Wireless Commun.* **6**, 3353–3363.