Polynomial-Time Upper Bound to the Taxman Score

Atli Fannar Franklín, *University of Iceland*
Robert K. Moniot, *Fordham University*

March 14, 2022

Abstract

We present a method of calculating an upper bound to the best possible player score in the game of Taxman. This upper bound is equal or close to the optimal score for all cases for which the optimal score is known. It can be calculated in polynomial time, making it practical to be computed out to large game sizes. We also discuss a bound that is apparently weaker but more tractable to analyze.

1 Introduction

The Taxman game was invented by Diane Resek of San Francisco State University, while she was working at the Lawrence Hall of Science in Berkeley, California from about 1969 to 1972 [9]. The purpose of the game was to give youngsters a fun way to practice their math skills. The game soon became popular with teachers of computer science as a programming exercise, since it is fairly easy but not trivial to implement, and provides a gentle introduction to important algorithm design principles [1]. It is also called "Number Shark."

Taxman is a one-person game, usually played with the aid of a computer. The game starts with a pot consisting of all of the positive numbers up to some chosen N. The player chooses a number, which is removed from the pot and added to the player's score. The taxman then takes as "tax" all of the numbers that divide that number, which are removed from the pot and added to the taxman's score. The player is not allowed to take a number that does not give the taxman any tax. This process is repeated until there are no numbers left in the pot that the player may take. The remaining numbers are taken by the taxman, an estate tax as it were.

The game has been studied to find optimal sequences of picks [2, 5]. The optimal scores as a function of pot size form a sequence that is listed on the Online Encyclopedia of Integer Sequences (OEIS) [8], sequence A019312. Since finding optimal play appears likely to be NP-hard, efforts have been made to find heuristic strategies that can always win [4, 10].

In the following, we let $s(N)$ denote the player's score for a pot containing N numbers.

2 Equivalence of Taxman game to a graph matching problem.

We now show that playing the Taxman game is equivalent to solving a graph matching problem. First some definitions.

We denote the initial pot of integers by $P = \{1, \ldots, N\}$.

Definition 1. *Suppose q and p are integers in P. We say that p covers q, denoted $q \to p$, if q is a proper divisor of p and there exists no $x \in P$ such that q is a proper divisor of x and x is a proper divisor of p.*

With this definition, we note that:

- No $x \in P$ satisfies $x \to x$.
- If $x, y \in P$, and $x \to y$, then $y \to x$ does not hold.

Definition 2. *For any integer $p \in P$, define the rank $\rho(p)$ as the number of prime factors of p, counted with multiplicity.*

By this definition $\rho(1) = 0$. Also, it follows that if $q, p \in P$ satisfy $q \to p$, then $\rho(p) = \rho(q) + 1$.

Definition 3. *Given a graph G, a matching M on G is a subset of the set of edges of G such that all of the vertices connected to edges in M are distinct. A matching is also called an independent edge set. If the edges have weights, the weight of the matching is the sum of the weights of the edges that are included in the matching.*

Theorem 1. *For a pot P of size N, construct a graph G in which the vertices are the elements of P, and there is an edge between vertex q and vertex p iff $q \to p$ or $p \to q$. Define the weight of an edge as the value of the larger vertex connected to it. Then every valid taxman game corresponds to a matching on G whose weight equals the player's score for the game.*

Proof. Construct the matching M as follows: if in round i of the game, the player takes p_i, let q_i be the largest value taken by the taxman. By the rules of the game, q_i exists, since the taxman must take something on every round, and q_i divides p_i. Now, suppose that $q_i \to p_i$ does not hold. Then there must be some $x \in P$ such that $q_i < x < p_i$ with $q_i | x$ and $x | p_i$. In order for x not to have been taken on this round, it must have been taken in some earlier round, by either the player or the taxman. But in either case, q_i would have been taken as well, so it would no longer be in the pot on round i, contradicting the assumption that it is taken by the taxman on that round. Therefore we have $q_i \to p_i$, so (q_i, p_i) is an edge of G. Place this edge in M.

For M to be a matching, it is necessary that all of the vertices connected to edges of M be distinct. This is clearly the case. For if q_i and p_i are taken by taxman and player respectively on round i, neither one could have been taken in an earlier round, nor are they available to be taken again in a later round. Hence neither q_i nor p_i can be equal to any

q_j or p_j for $i \neq j$. And $q_i \neq p_i$ since q_i is a proper divisor of p_i. Therefore M is a matching on G.

Since the weight of edge (q_i, p_i) is p_i, the weight of the matching is

$$W(M) = \sum_i p_i. \tag{1}$$

This is by definition the player's score.

□

However, not every matching corresponds to a valid move sequence. This is because the taxman may be able to take a number that is in the matching via an edge that is not in the matching.

A few comments that are not directly relevant to this result, but worth noting. First, G as constructed in Theorem 1 is an ordered $(k+1)$-partite graph, where $k = \lfloor \log_2 N \rfloor$. Simply observe that since every edge connects a pair of vertices whose ranks differ by 1, we can divide the vertices into sets of vertices having the same rank. Then each edge connects a vertex from a set of rank r to one from a set of rank $r-1$ or $r+1$. The minimum rank is 0 and the maximum rank k is achieved by the largest value of $2^k \leq N$. We also comment that for analysis of optimal play, the edges can be taken as directed from q to p where $q \to p$. Then G is a DAG. This approach has been used by Brian Chess [2] and Dan Hoey [5] to develop effective methods for finding optimal play sequences.

Corollary 1. *Given a pot size N, the maximum weight of a matching M on the corresponding graph G is an upper bound to the optimal player's score:*

$$s(N) \leq \max_{M \text{ on } G} W(M) \tag{2}$$

Proof. The player's score for a game is equal to the weight of the matching corresponding to the move sequence. Therefore it is not possible for a move sequence to achieve a score that exceeds the maximum weight of a matching. □

3 A simpler upper bound

Moniot [7] observed that a simple upper bound to the player's score can be found by noting that the taxman must take at least one number from the pot on every round, *i.e.,* at least half of the numbers in the pot in the course of the game. At best, the player can take the largest numbers while giving the taxman the smallest numbers. For even N, a bound on the player's score is thus

$$s(N) \leq \sum_{i=\frac{N}{2}+1}^{N} i = \frac{1}{8}N(3N+2). \tag{3}$$

As a fraction of the pot this bound is

$$\frac{\frac{1}{8}N(3N+2)}{\frac{1}{2}N(N+1)} = \frac{1}{4}\frac{3N+2}{N+1}, \tag{4}$$

which asymptotically approaches $3/4$ for large N.

This very simple calculation ignores the fact that the taxman must get all but one of the primes greater than $N/2$ at the end of the game. The player's best first move is to take the largest prime in the pot, giving the taxman 1 [1]. Thereafter, none of the remaining primes have any divisors available for the taxman to take, and so are not eligible to be taken by the player.

We can refine the analysis to take this fact into consideration. Denote the number of primes less than or equal to x by $\pi(x)$. The primes less than or equal to $N/2$ can go to the taxman during play as divisors of numbers the player takes. Then excluding all but one of the primes greater than $N/2$, the number of numbers that can be taken during play, by player or taxman, is

$$N' = N - (\pi(N) - \pi(N/2)) + 1, \quad N > 1. \tag{5}$$

(For $N = 1$ the equation is invalid since the assumption that there is at least one prime greater than or equal to $N/2$ fails.) Now, since the player can at best take all the numbers in the larger half of this set, the bound on the score is

$$s(N) \leq \sum_{i=\lceil N'/2 \rceil+1}^{N} i - \sum_{i=\pi(N/2)+1}^{\pi(N)-1} p_i, \tag{6}$$

where p_i is the ith prime. Let

$$\sigma(x) = \sum_{i=1}^{\pi(x)} p_i. \tag{7}$$

Then (6) can be written

$$s(N) \leq \frac{N(N+1)}{2} - \frac{\left\lceil \frac{N'}{2} \right\rceil \left(\left\lceil \frac{N'}{2} \right\rceil + 1 \right)}{2} - (\sigma(N) - \sigma(N/2)) + p_{\pi(N)}. \tag{8}$$

Clearly the player cannot do better than this (and for most N cannot equal it), so this formula provides an upper bound on the score.

There are well-known approximations

$$p_x \sim x \log x, \tag{9}$$

$$\pi(x) \approx \frac{x}{\log x}, \tag{10}$$

and

$$\sigma(x) \approx \frac{x^2}{2 \log x}. \tag{11}$$

Using these,

$$N' \approx N - \frac{N}{\log N} + \frac{N/2}{\log N/2} + 1 \tag{12}$$

and (8) can be rewritten as follows. (For simplicity, we neglect $p_{\pi(N)}$, which is small and not worth including given the error in the approximations.)

$$s(N) \lesssim \frac{N(N+1)}{2} - \frac{\left\lceil \frac{N'}{2} \right\rceil \left(\left\lceil \frac{N'}{2} \right\rceil + 1 \right)}{2} - \frac{N^2}{2} \left(\frac{1}{\log N} - \frac{1}{4 \log \frac{N}{2}} \right). \tag{13}$$

This shows that the sum of leftover primes the taxman takes at the end of the game grows only slightly more slowly (by the factor $1/\log N$) than the player's maximum take, and so remains significant out to quite large N. Of course, by that time, the other numbers that become unpickable during play will also be decreasing the true maximum player's take, so possibly the total "estate tax," the numbers taken by the taxman at the end of the game, always remains significant.

4 Discussion

We note that a maximum-weight matching can be computed in polynomial time. For instance, using Edmond's algorithm [3], the running time is $O(V^2E)$ where V is the number of vertices and E is the number of edges in the graph. More efficient algorithms exist, such as Blossom V by Kolmogorov [6]. Therefore the upper bound is practical to compute for values of N that are significantly larger than the current largest N for which the optimal score is known.

We compared the upper bounds given by (2) to the optimal scores found by Brian Chess for $N = 1$ to $N = 701$. For some values of $N \leq 90$ the optimal score is equal to this upper bound, but for $91 \leq N \leq 701$ this upper bound always exceeds the optimal score. The two largest fractional differences we found are for $N = 51$ and $N = 54$, where the optimal score is approximately 97.1% and 98.7%, respectively, of the upper bound. For all other values of N up to 701, the optimal score is more than 99% of the bound. We do not have a proof that the bound remains this tight for all larger values of N. But in any case, this bound can serve as a benchmark for testing proposed heuristic methods for winning at Taxman, for values of N beyond those for which the optimal scores are known.

The simpler bound given by (8) is equal to this upper bound for 27 values of N, the largest being $N = 36$. For larger N, the bound (2) is observed to be always tighter. For $37 \leq N \leq 1000$, the bound (8) is about 5.8% higher than the bound (2) on average. Although we don't have a big-$\mathcal{O}$ growth rate for the bound in (2), the observed trend is that the divergence between the two bounds increases as N increases.

References

[1] Carmony, Lowell A., and Holliday, Robert L. (1993). "An example from Artificial Intelligence for CS1." *SIGCSE Bulletin* **25:1,** 1–5.

[2] Chess, Brian (2021). https://github.com/bvchess/taxman

[3] Edmonds, Jack (1965). "Paths, trees, and flowers." *Canadian Journal of Mathematics* **17,** 449–467. doi:10.4153/CJM-1965-045-4.

[4] Hensley, Douglas (1988). "A Winning Strategy at Taxman." *Fibonacci Quarterly* **26:3,** 262.

[5] Hoey, Dan. Notes on A019312. Posted on OEIS at A019312.

[6] Kolmogorov, Vladimir (2009). "Blossom V: A new implementation of a minimum cost perfect matching algorithm." *Mathematical Programming Computation* **1,** 43–67. doi:10.1007/s12532-009-0002-8.

[7] Moniot, Robert K. (2007). "The Taxman Game." *Math Horizons* **14,** February, 18-–20.

[8] *On-Line Encyclopedia of Integer Sequences*™, published electronically at http://oeis.org. Accessed December, 2021.

[9] Resek, Diane (2008), private communication to one of us (RKM).

[10] Trono, John A. (1994). "Taxman revisited." *SIGCSE Bulletin* **26:4,** 56–58.