A CONSTRUCTIVE APPROACH TO UNIT FRACTIONS WITH SHIFTED PRIME DENOMINATORS

GUO-NIU HAN

Abstract. A conjecture of Zhi-Wei Sun states that every positive rational number can be written as a finite sum of unit fractions whose denominators are of the form $p+1$ or $p-1$, where p runs over distinct primes. Both cases have recently been proved in full generality by Thomas Bloom [1]. Bloom's theorems are existence results; even for small integers such as $r = 2$ or $r = 3$, producing an explicit decomposition remains highly non-trivial, as the required primes can have hundreds of decimal digits. We give a computational construction of explicit, certified prime sets for two integer cases — $r = 2$ with denominators $p+1$ and $r = 3$ with denominators $p-1$ — and describe a beam-search method, guided by splitting tables, that also succeeds on several finite families of rational inputs. These experiments lead us to conjecture that every reciprocal $1/d$ admits such a decomposition with at most $\log_2 d + 1$ distinct primes.

1. Introduction

The representation of a positive rational as a sum of distinct unit fractions $1/n$ is a classical topic going back to Egyptian fraction expansions [2, 5]. A more recent variant restricts the denominators to lie in arithmetic sets of the form $\{p+s : p \text{ prime}\}$ for a fixed integer shift s. Two natural choices are $s = +1$ and $s = -1$, which lead to the following results, initially conjectured by Sun [3] and recently proved by Thomas Bloom [1].

Theorem 1 (Bloom [1])**.** *For every positive rational number r, there exists a finite set P of prime numbers such that*

$$r = \sum_{p \in P} \frac{1}{p+1}.$$

Theorem 2 (Bloom [1])**.** *For every positive rational number r, there exists a finite set P of prime numbers such that*

$$r = \sum_{p \in P} \frac{1}{p-1}.$$

Sun had verified these statements for many small rationals. Qing-Hu Hou has verified them for all rational numbers $r \in (0, 1)$ with denominator not exceeding 100 [3]. However, even for small integers such as $r = 2$ or $r = 3$, finding an explicit decomposition is highly non-trivial: the required primes can have hundreds of decimal digits, and the construction involves hundreds of elimination steps.

Since Bloom's theorems settle existence but do not yield explicit decompositions, the emphasis of the present paper is different: our contribution is to provide explicit, independently certified examples together with a practical computational method. Concretely, we give such explicit constructions for the two integer cases $r = 2$ (shift $s = +1$) and

Date: July 8, 2026.

2020 *Mathematics Subject Classification.* 11A67, 11D68, 11Y16.

Key words and phrases. Egyptian fractions, unit fractions, prime numbers, Sun's conjecture, beam search, greedy algorithm.

$r = 3$ (shift $s = -1$), by means of a beam-search algorithm that may also serve as a starting point for related problems.

Theorem 3 (Case $r = 2$, $s = +1$)**.** *There exists a set P_+ of* 1963 *distinct prime numbers, the largest of which has* 181 *decimal digits and equals*

$$236194408181126266158142002999377936900474925811289085690070849364988353181005517315740420015211654400649794482370710161800450310696463310723444416974240855851004334621478631467439 9,$$

such that

$$\sum_{p \in P_+} \frac{1}{p+1} \;=\; 2.$$

Theorem 4 (Case $r = 3$, $s = -1$)**.** *There exists a set P_- of* 698 *distinct prime numbers, the largest of which has* 68 *decimal digits and equals*

$$21405811313494409045983653494328316787629317190709732873740497324001,$$

such that

$$\sum_{p \in P_-} \frac{1}{p-1} \;=\; 3.$$

The same algorithm, run on a wide range of inputs, also establishes the following further verifications.

Theorem 5 (Further verified cases)**.** *For each shift $s \in \{+1, -1\}$, Theorems 1 and 2 hold for every rational r in the following families:*

1. *all reduced fractions a/b with $1 \le a < b \le 1000$;*
2. *all reciprocals $1/q$ for primes $q \le 100\,000$;*
3. *all reciprocals $1/2^k$ for integers $1 \le k \le 40$.*

In each case an explicit set P of distinct primes with $\sum_{p \in P} 1/(p+s) = r$ is produced and verified in exact rational arithmetic. Moreover, the produced sets are short: for every r in each family, the number of primes $|P|$ satisfies the bound collected in Table 1.

Family	shift s	bound on $\lvert P\rvert$
Fractions a/b, $1 \le a < b \le 1000$	± 1	$b + 8$
Prime reciprocals $1/q$, $q \le 10^5$	± 1	$\log_2 q + 1$
Power reciprocals $1/2^k$, $k \le 40$	± 1	$k + 1$

TABLE 1. Length bounds for the produced sets P

The bounds in Table 1 are established by direct verification over the finite families; they are properties of the particular sets P produced by `sunconj.py`, not of every possible decomposition.

We observe that the reciprocal families (2) and (3) obey the same logarithmic length bound $|P| \le \log_2 d + 1$, where d denotes the denominator. This suggests that the bound may hold for the reciprocal of an arbitrary integer, as recorded in the following conjecture.

Conjecture 6. *Let $d \ge 2$ be an integer and let $s \in \{+1, -1\}$. Then $1/d$ admits a prime decomposition*

$$\sum_{p \in P} \frac{1}{p+s} \;=\; \frac{1}{d} \qquad (P \subset \mathcal{P} \text{ a finite set of distinct primes})$$

whose length satisfies

$$|P| \;\le\; \log_2 d + 1.$$

The existence of *some* such decomposition is already guaranteed by Bloom's Theorems 1 and 2; the content of Conjecture 6 lies entirely in the *length* bound. Bloom's proof, however, settles existence non-constructively and gives no control over the number of terms. The logarithmic bound $|P| \le \log_2 d + 1$ is therefore beyond the reach of [1], and appears to be a genuinely constructive, algorithmic phenomenon. We have verified Conjecture 6 for both shifts and for every integer $2 \le d \le 10\,000$, as well as for all primes $d \le 10^5$ and all powers $d = 2^k$ with $k \le 40$ (Table 1); the check for arbitrary d is carried out by the self-contained program `sunconj_reciprocal.py`.

We deliberately do not conjecture a companion bound on the largest prime $\max P$: empirically its exponent $\log_d(\max P)$ is erratic and grows with d, so no fixed power of d appears to bound it for all reciprocals.

The proofs are constructive and use exactly the same algorithm, run with different inputs: the sets P_+ and P_- are produced by the program `sunconj.py` described in Section 3 and listed in full in the supplementary files `prime_set_P_plus2.txt` and `prime_set_P_minus3.txt`. All identities are verified in arbitrary-precision rational arithmetic (`gmpy2.mpq`). All programs and data files are available in a permanent public repository, allowing independent verification of the computations:

`https://github.com/GuoniuHan/sunconj`

The paper is organised as follows. Section 2 introduces notation. Section 3 describes the algorithm, with worked examples. Section 4 illustrates the first three iterations on the run $(r = 2, s = +1)$. Section 5 gives the proofs of Theorems 3 and 4.

2. Notation and setting

We denote by $\mathcal{P}$ the set of primes. Throughout the paper, s denotes a fixed shift in $\{+1, -1\}$. For a finite multiset M of primes we write

$$S(M) \;=\; \sum_{p \in M} \frac{1}{p+s}$$

(counting multiplicities). When the shift needs to be displayed explicitly we write $S^{(s)}(M)$. A *prime decomposition* of a positive rational r at shift s is a finite *set* $P \subset \mathcal{P}$ such that $S^{(s)}(P) = r$.

A central elementary identity for our algorithm is

$$\frac{1}{p+s} \;=\; \frac{1}{p+s+1} \;+\; \frac{1}{(p+s)(p+s+1)}, \tag{1}$$

which splits one term $1/(p+s)$ into two terms with strictly larger denominators. More generally, for a prime p and a positive integer m, one is led to the following sub-problem:

Find a finite set Q of primes with $\min Q > p$ *such that*

$$\frac{m}{p+s} \;=\; \sum_{q \in Q} \frac{1}{q+s}.$$

We call such a Q a *prime replacement* for the triple (p, m, s). The existence of prime replacements for every (p, m) would imply Theorems 1 and 2 constructively for all positive rationals. In our algorithm we do not solve this sub-problem in full generality. Instead, we solve it heuristically for the finitely many triples (p, m, s) that arise during a run, and we verify each replacement in exact arithmetic.

A crucial concept of our approach is the *excess count* of a multiset. For a finite multiset M of primes we define

$$\mathtt{ct}(M) \;=\; \sum_{p\in M,\,\mu_M(p)\geq 2} \big(\mu_M(p)-1\big),$$

where $\mu_M(p)$ is the multiplicity of p in M. Thus $\mathtt{ct}(M)=0$ if and only if M is a set. The whole point of Phase 2 of the algorithm is to drive `ct` down to 0.

3. The algorithm

The algorithm is implemented in the Python program `sunconj.py`, which uses `gmpy2` [4] for exact rational arithmetic and primality testing. The shift $s\in\{+1,-1\}$ is a parameter; on the cases treated here, the program runs in roughly 6 minutes on a standard desktop computer for $(r,s)=(2,+1)$, and in well under a minute for $(r,s)=(3,-1)$.

3.1. Outline.

The algorithm has two phases.

Phase 1.: *Initial multiset.* Find a multiset M_0 of primes with $S^{(s)}(M_0)=r$.

Phase 2.: *Deduplication.* Iteratively replace one repeated prime at a time, until the multiset becomes a set.

Both phases share a common subroutine, the *beam search* of Section 3.3, which produces decompositions of a rational r as a sum of distinct unit fractions $1/(p+s)$ with p prime and p above a prescribed lower bound.

3.2. Splitting tables.

The beam search is guided by a precomputed list of partitions of 1 into two or three reduced fractions; we call this list a *splitting table*. Concretely, we use:

- all pairs $(a,1-a)$ with a a reduced fraction j/k, $1\leq j<k$, denominator k below a small bound;
- all triples $(a,b,1-a-b)$, ordered $a<b<1-a-b$, with the same denominator constraint;
- the trivial table entry [1].

Three tables are built, of increasing size and thoroughness: Fast (denominators <10 for pairs, <7 for triples), Middle (<20 and <13), Deep (<30 and <17).

Given a target rational r and a table entry $[w_1,\dots,w_t]$ (with $w_1+\cdots+w_t=1$), the algorithm replaces the single problem "decompose r" by the t independent sub-problems "decompose w_ir" $(i=1,\dots,t)$, each handled by a separate call to the beam search, since

$$w_1r+\cdots+w_tr=r.$$

Since $w_i<1$, the sub-problems are "easier" because each w_ir is strictly smaller than r, so the greedy search reaches zero in fewer steps.

This trick is essential: a direct greedy expansion of a generic rational r tends to produce extremely long sequences of primes, or to fail to terminate within reasonable cutoffs, whereas after splitting via a table entry the individual sub-rationals are typically solvable in a few steps.

The Fast table contains 29 entries: the trivial entry [1], the 27 ordered pairs $(j/k,\,1-j/k)$ with $\gcd(j,k)=1$, $1\leq j<k<10$, and the single triple with denominator below 7:

$$\frac{1}{6}+\frac{1}{3}+\frac{1}{2}\;=\;1.$$

The first few pairs, in the order they appear in the table, are

$$\frac{1}{2}+\frac{1}{2},\quad \frac{1}{3}+\frac{2}{3},\quad \frac{1}{4}+\frac{3}{4},\quad \frac{1}{5}+\frac{4}{5},\quad \frac{2}{5}+\frac{3}{5},\quad \frac{1}{6}+\frac{5}{6},\quad \cdots$$

See Examples 9 and 10 for illustrations of how the splitting tables work.

3.3. **Beam search.** The beam search is the core decomposer. Given a rational $r > 0$ and a lower bound b, it seeks a list of distinct primes $b < p_1 < p_2 < \cdots < p_k$ with $\sum 1/(p_i + s) = r$. It proceeds as follows.

(1) *Candidate generation.* For each rational remainder encountered, list the n smallest primes $p > b$ with $1/(p + s) \le r$. The width n is 3, 4, or 6 in modes Fast, Middle, Deep respectively.
(2) *Beam expansion.* The *frontier* (the current set of partial decompositions being explored, each recorded as a pair [primes chosen so far, remaining sum]) is expanded for 1 round (mode Fast) or 3 rounds (Middle/Deep), each round multiplying the frontier size by n.
(3) *Greedy completion.* From every leaf of the beam, a pure greedy completion is attempted: at each step one subtracts the largest feasible $1/(p + s)$ until the remainder reaches 0. If the remainder has not reached 0 after 99 primes have been selected, the completion returns FALSE and the leaf is discarded. There is no prime-size cutoff: when the unsolved remainder is a very small rational, the beam candidates are necessarily huge primes, and the beam expansion (Step 2) returns the smallest one with no restriction on its size. This is how primes with hundreds of digits appear in the final set P.
(4) *Selection.* Among all successful completions, the one minimising the sum of primes is returned.

Example 7. Beam search on target $r = 1/3$, lower bound $b = 1$, mode Fast.

Step 1: candidate generation. List the $n = 3$ smallest primes $p > b$ with $1/(p+1) \le 1/3$, i.e. $p \ge 2$:

$$p = 2\ (\tfrac{1}{3}), \quad p = 3\ (\tfrac{1}{4}), \quad p = 5\ (\tfrac{1}{6}).$$

Step 2: beam expansion (1 round). Branch on each candidate and record the remainder:

$$\begin{aligned}
&\text{branch } p = 2: \ \text{take } \tfrac{1}{3}, \ \text{remainder } = \tfrac{1}{3} - \tfrac{1}{3} = 0 \quad \Rightarrow \quad \text{already complete},\\
&\text{branch } p = 3: \ \text{take } \tfrac{1}{4}, \ \text{remainder } = \tfrac{1}{3} - \tfrac{1}{4} = \tfrac{1}{12},\\
&\text{branch } p = 5: \ \text{take } \tfrac{1}{6}, \ \text{remainder } = \tfrac{1}{3} - \tfrac{1}{6} = \tfrac{1}{6}.
\end{aligned}$$

Step 3: greedy completion of each leaf.

- Leaf [2]: remainder 0, done. Result: $\{2\}$.
- Leaf [3]: remainder $\frac{1}{12}$. Smallest prime $p > 3$ with $\frac{1}{p+1} \le \frac{1}{12}$ is $p = 11$; taking it leaves remainder 0. Result: $\{3, 11\}$.
- Leaf [5]: remainder $\frac{1}{6}$. Take $p = 7$ ($\frac{1}{8}$), remainder $\frac{1}{6} - \frac{1}{8} = \frac{1}{24}$; then $p = 23$ ($\frac{1}{24}$), remainder 0. Result: $\{5, 7, 23\}$.

Step 4: selection. Among the three completions, the one minimising the sum of primes is chosen:

$$\{2\}\ (\text{sum } = 2), \quad \{3, 11\}\ (\text{sum } = 14), \quad \{5, 7, 23\}\ (\text{sum } = 35).$$

The beam search returns $\{2\}$, and indeed $\frac{1}{2+1} = \frac{1}{3}$. Note that all three sets are valid decompositions of $\frac{1}{3}$; the beam search finds all of them and picks the simplest.

Example 8 (Beam search on target $r = 1/2$, lower bound $b = 6$, mode Fast)**.** This is the sub-problem arising in iteration 3 of Phase 2 (elimination of $p^* = 5$, $m = 4$, target $(m - 1)/(p^* + 1) = 1/2$). The lower bound $b = p^* + 1 = 6$ ensures that all replacement primes are strictly greater than 5.

Step 1: candidate generation. The $n = 3$ smallest primes $p > 5$ with $1/(p+1) \le 1/2$ are:

$$p = 7\ (\tfrac{1}{8}), \quad p = 11\ (\tfrac{1}{12}), \quad p = 13\ (\tfrac{1}{14}).$$

Step 2: beam expansion (1 round).

$$\begin{aligned} &\text{branch } p = 7\text{:} \quad \text{take } \tfrac{1}{8}, \quad \text{remainder} = \tfrac{1}{2} - \tfrac{1}{8} = \tfrac{3}{8}, \\ &\text{branch } p = 11\text{:} \quad \text{take } \tfrac{1}{12}, \quad \text{remainder} = \tfrac{1}{2} - \tfrac{1}{12} = \tfrac{5}{12}, \\ &\text{branch } p = 13\text{:} \quad \text{take } \tfrac{1}{14}, \quad \text{remainder} = \tfrac{1}{2} - \tfrac{1}{14} = \tfrac{3}{7}. \end{aligned}$$

Step 3: greedy completion of each leaf.

- Leaf [7], remainder $\frac{3}{8}$: take $p = 11$ ($\frac{1}{12}$, rem. $\frac{7}{24}$), $p = 13$ ($\frac{1}{14}$, rem. $\frac{37}{168}$), …, eventually reaching $p = 2{,}169{,}227$. Result: $\{7, 11, 13, 17, 19, 23, 29, 31, 127, 1613, 2169227\}$, sum of primes $\approx 2.17 \times 10^6$.
- Leaf [11], remainder $\frac{5}{12}$: take $p = 13, 17, 19, 23, 29, 31, 37, 41, 43, 47$. Result: $\{11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47\}$, sum $= 311$.
- Leaf [13], remainder $\frac{3}{7}$: take $p = 17, 19, 23, 29, 31, 37, 41, 43, 47, 53$. Result: $\{13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53\}$, sum $= 353$.

Step 4: selection. The beam selects the completion with the smallest sum of primes, namely leaf [11] with sum 311. The beam search therefore returns $Q = \{11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47\}$, and indeed

$$\sum_{p \in Q} \frac{1}{p+1} = \frac{1}{12} + \frac{1}{14} + \frac{1}{18} + \frac{1}{20} + \frac{1}{24} + \frac{1}{30} + \frac{1}{32} + \frac{1}{38} + \frac{1}{42} + \frac{1}{44} + \frac{1}{48} = \frac{1}{2}.$$

Comparing the three leaves, leaf [7] produces a prime with more than six decimal digits — illustrating why a pure greedy search (always taking the largest possible term) is impractical. The beam search, by exploring multiple branches simultaneously, finds the much cleaner solution from leaf [11].

Example 9 (Splitting table + beam search, target $r = 1/2$, lower bound $b = 6$, mode Middle)**.** This is the sub-problem arising in iteration 3 of Phase 2 (elimination of $p^* = 5$, $m = 4$, replacement target $(m-1)/(p^*+1) = 3/6 = 1/2$, lower bound $b = p^* + 1 = 6$). The actual run uses mode Middle (since `prefer_large=True`).

The function `_decompose_via_split_table` iterates through the Middle table (82 entries) until one fully succeeds. In our example, the first successful entry is Entry 1, the pair [1/4, 3/4], which splits the target 1/2 into two independent sub-problems:

$$\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}, \qquad \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}.$$

Beam on 1/8*, lower bound* 6. The unique prime $p > 5$ with $1/(p+1) = 1/8$ is $p = 7$. The beam returns $\{7\}$ immediately.

Beam on 3/8*, lower bound* 6. The beam (with $n = 4$ candidates in Middle mode, and 3 expansion rounds) returns $\{7, 11, 19, 23, 29, 31, 97, 4703\}$. Indeed,

$$\frac{1}{8} + \frac{1}{12} + \frac{1}{20} + \frac{1}{24} + \frac{1}{30} + \frac{1}{32} + \frac{1}{98} + \frac{1}{4704} = \frac{3}{8}.$$

Combined result. Concatenating the two sub-results gives $Q = \{7, 7, 11, 19, 23, 29, 31, 97, 4703\}$, and indeed

$$\frac{1}{8} + \frac{1}{8} + \frac{1}{12} + \frac{1}{20} + \frac{1}{24} + \frac{1}{30} + \frac{1}{32} + \frac{1}{98} + \frac{1}{4704} = \frac{1}{2}.$$

The prime 7 appears twice because both sub-problems independently selected it as their first term. This is consistent with what was reported in iteration 3: the two sub-problems

are solved in isolation, without any global check on which primes are already present, so new multiplicities can be introduced — which is precisely why `ct` rises to 9 after this iteration.

3.4. **Phase 1: initial multiset M_0.** In Phase 1 we apply the beam search above to the target rational r in mode Fast (then Middle, then Deep on failure).

Example 10 (Phase 1 for the run $(r = 2,\ s = +1)$)**.** The beam search is called on target $r = 2$ in mode Fast. The function `_decompose_via_split_table` tries table entries in order; the first one that fully succeeds is the pair $[2/5,\ 3/5]$, which splits the target into two independent sub-problems:

$$\frac{2}{5}\cdot 2 \;=\; \frac{4}{5}, \qquad \frac{3}{5}\cdot 2 \;=\; \frac{6}{5}.$$

The beam solves each independently:

$$\frac{4}{5} \;\longrightarrow\; \{2,3,5,19\}, \qquad \frac{6}{5} \;\longrightarrow\; \{2,3,5,7,11,13,17,19,23,43,3467,4005539\}.$$

Concatenating the two sub-results gives the initial multiset

$$M_0^+ \;=\; \{\,2,2,3,3,5,5,7,11,13,17,19,19,23,43,3467,4005539\,\},$$

of size 16, with $\mathtt{ct}(M_0^+) = 4$. The four repeated primes are 2, 3, 5, and 19, each appearing exactly twice, one copy coming from each sub-result. One verifies in exact rational arithmetic that

$$\sum_{p\,\in\,M_0^+} \frac{1}{p+1} \;=\; \frac{2}{3}+\frac{2}{4}+\frac{2}{6}+\frac{1}{8}+\frac{1}{12}+\frac{1}{14}+\frac{1}{18}+\frac{2}{20}+\frac{1}{24}+\frac{1}{44}+\frac{1}{3468}+\frac{1}{4005540} \;=\; 2.$$

Phase 1 terminates here and hands M_0^+ to Phase 2, which will iteratively eliminate the four repeated primes.

3.5. **Phase 2: deduplication loop.** Let M denote the current multiset, initialised to $M = M_0$. Each iteration of Phase 2 performs the following step.

(1) *Pick a repeated prime.* Let $p^* = \min\{p \in M : \mu_M(p) \geq 2\}$ and let $m = \mu_M(p^*)$.
(2) *Replace.* Find a list Q of distinct primes, all $> p^*$, with

$$\sum_{q\in Q} \frac{1}{q+s} = \frac{m-1}{p^*+s}.$$

This is done by calling the beam search of Section 3.3 on the target $(m-1)/(p^*+s)$ with lower bound p^*, with each splitting table entry tried in turn.
(3) *Update.* Remove $m-1$ copies of p^* from M, and add Q. The new multiset M' satisfies $S^{(s)}(M') = S^{(s)}(M) = r$.

See Section 4 for an illustration of how this works.

3.6. **Termination.** A naive expectation is that `ct` should decrease monotonically. This is *not* the case: the replacement of $m-1$ copies of p^* by a list Q of new primes may introduce *new* multiplicities, because some primes in Q may already lie in M. Thus `ct` can rise sharply in the early iterations.

As shown in the two figures of Section 5, the trajectory of `ct` in practice follows a roughly unimodal shape: it first rises (with occasional downward fluctuations) to a peak, then decreases (with occasional upward fluctuations) to 0. This behaviour is not proved mathematically, but can be intuitively understood as follows. In the early iterations, the repeated primes are small and their replacements Q frequently collide with primes

already in M, causing `ct` to grow. As the frontier of unresolved primes moves upward, collisions become rarer and `ct` begins its descent.

The algorithm terminates successfully if and only if `ct` eventually reaches 0. Guaranteeing termination for all inputs r would provide a fully constructive proof of Theorems 1 and 2, complementing Bloom's existential argument. This remains open: we can only verify termination experimentally for specific inputs, as reported in Section 5.

4. Concrete examples from the run ($r = 2$, $s = +1$)

From Example 10, the initial multiset is

$$M_0^+ = \{\, 2, 2, 3, 3, 5, 5, 7, 11, 13, 17, 19, 19, 23, 43, 3467, 4005539 \,\},$$

with $\texttt{ct}(M_0^+) = 4$, the four repeated primes being $2, 3, 5, 19$.

Iteration 1: $p^* = 2$, $m = 2$. Phase 2 removes one copy of 2 and must find a set Q of distinct primes > 2 satisfying

$$\sum_{q \in Q} \frac{1}{q+1} \;=\; \frac{m-1}{p^*+1} \;=\; \frac{1}{3}.$$

The beam search, using the table entry [1/3, 2/3] to split the target 1/3 into sub-targets 1/9 and 2/9, returns $Q = \{5, 7, 23\}$. Indeed,

$$\frac{1}{6} + \frac{1}{8} + \frac{1}{24} \;=\; \frac{1}{3}.$$

The new multiset is

$$M_1 = \{\, 2, 3, 3, 5, 5, 5, 7, 7, 11, 13, 17, 19, 19, 23, 23, 43, 3467, 4005539 \,\}.$$

Since 5, 7, 23 were already in M_0^+, their multiplicities increase, and `ct` rises from 4 to 6. The repeated primes are now $\{3, 5, 7, 19, 23\}$ with multiplicities $\{2, 3, 2, 2, 2\}$.

Iteration 2: $p^* = 3$, $m = 2$. One copy of 3 is removed; the replacement target is

$$\frac{m-1}{p^*+1} \;=\; \frac{1}{4}.$$

The beam search returns $Q = \{5, 11\}$. Indeed,

$$\frac{1}{6} + \frac{1}{12} \;=\; \frac{1}{4}.$$

The new multiset is

$$M_2 = \{\, 2, 3, 5, 5, 5, 5, 7, 7, 11, 11, 13, 17, 19, 19, 23, 23, 43, 3467, 4005539 \,\}.$$

Adding 5 and 11 (both already present) raises `ct` further to 7. The prime 5 now has multiplicity 4.

Iteration 3: $p^* = 5$, $m = 4$. The multiplicity of 5 grew from 2 in M_0^+ to 4 in M_2 because iterations 1 and 2 each introduced an extra copy of 5. Three copies of 5 are removed, leaving one. The replacement target is

$$\frac{m-1}{p^*+1} \;=\; \frac{3}{6} \;=\; \frac{1}{2}.$$

A larger target is not a problem: it simply means the beam search must find more primes to fill it. From Example 9, the beam search returns the multiset $Q = \{7, 7, 11, 19, 23, 29, 31, 97, 4703\}$ (nine primes, with 7 appearing twice). Indeed,

$$\frac{1}{8} + \frac{1}{8} + \frac{1}{12} + \frac{1}{20} + \frac{1}{24} + \frac{1}{30} + \frac{1}{32} + \frac{1}{98} + \frac{1}{4704} \;=\; \frac{1}{2}.$$

Since five of the nine primes in Q (namely $7, 7, 11, 19, 23$) are already present in M_2, the count rises further: $\texttt{ct}(M_3) = 9$. The repeated primes in M_3 are $\{7, 11, 19, 23\}$ with multiplicities $\{4, 3, 3, 3\}$.

This illustrates why `ct` grows during the early iterations: each replacement introduces primes that are already in the multiset, creating new collisions faster than old ones are resolved. Although `ct` has increased at every step so far, the algorithm is making irreversible progress: the primes 2, 3, and 5 will never be repeated again in any subsequent multiset M_j. The frontier of unresolved primes moves strictly upward at every iteration.

5. The two runs: proofs of Theorems 3 and 4

The two theorems are obtained by running `sunconj.py` with different command-line arguments. We summarise both runs in Table 2, then comment on the trajectory of `ct` in each case.

	Theorem 3: $r = 2$, $s = +1$	Theorem 4: $r = 3$, $s = -1$
Command	`sunconj.py +1 2 false true`	`sunconj.py -1 3 false true`
$\lvert M_0 \rvert$	16	12
$\texttt{ct}(M_0)$	4	2
Phase 2 iterations	452	167
Peak `ct`	180 at $j = 191$	57 at $j = 48$
$\lvert P \rvert$	1963	698
Largest prime	181 digits	68 digits

Table 2. Summary of the two runs of `sunconj.py`

5.1. **Proof of Theorem 3: $r = 2$ via $1/(p+1)$.** Run the program `sunconj.py` on the target $r = 2$ with the command

```
python sunconj.py +1 2 false true
```

From Example 10, Phase 1 produces the multiset

$$M_0^+ = \{\, 2, 2, 3, 3, 5, 5, 7, 11, 13, 17, 19, 19, 23, 43, 3467, 4005539 \,\}, \tag{2}$$

of size 16, with $\texttt{ct}(M_0^+) = 4$ (four excess copies, namely the second copies of 2, 3, 5, and 19).

Phase 2 then runs for 452 iterations. The trajectory of `ct` is plotted in Figure 1. It is roughly unimodal: `ct` rises (with occasional downward fluctuations) from 4 at iteration 0 to a peak of 180 at iteration 191, then decreases (with occasional upward fluctuations) to 0 at iteration 451.

After 452 iterations the program prints a certificate line of the form

$$\texttt{RESGOOD} \quad r \quad [\,p_1,\, p_2,\, \ldots,\, p_k\,] \quad k,$$

whose fields are the keyword `RESGOOD`, the target rational r, the sorted list of primes, and its length k. For $(r, s) = (2, +1)$ this line is `RESGOOD 2 [2,3,5,...] 1963`, listing the elements of the set

$$P_+ = \{p_1, p_2, \ldots, p_{1963}\} \subset \mathcal{P}$$

in increasing order. The full list is contained in the supplementary file `prime_set_P_plus2.txt`. We verify three properties:

(1) *All elements are prime*, confirmed by `gmpy2` primality testing.
(2) *All elements are distinct*: the output list, sorted in increasing order, has length 1963.

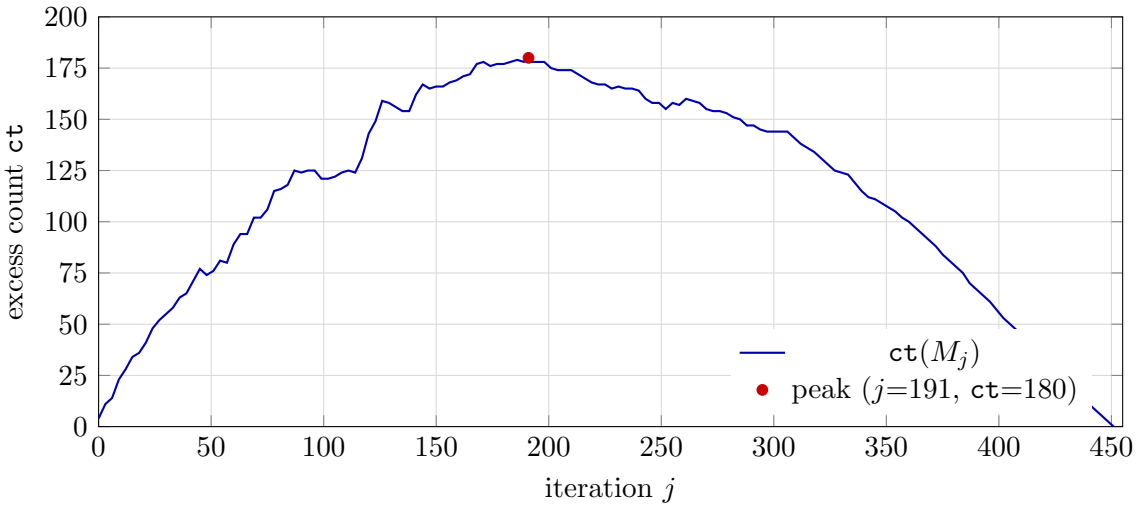


FIGURE 1. Trajectory of ct during Phase 2 for $(r, s) = (2, +1)$

(3) *The sum equals* 2 *exactly*: the sum $\sum_{p\in P_+} 1/(p+1) = 2$ is computed in arbitrary-precision rational arithmetic.

The smallest element of P_+ is 2, the largest is a prime with 181 decimal digits. This proves Theorem 3. □

5.2. **Proof of Theorem 4: $r = 3$ via $1/(p-1)$.** Run the program with the command

```
python sunconj.py -1 3 false true
```

($s = -1$, target $r = 3$). Phase 1 produces the initial multiset

$$M_0^- = \{2, 3, 3, 5, 5, 7, 11, 13, 17, 19, 37, 241\}, \tag{3}$$

of size 12, with $\mathtt{ct}(M_0^-) = 2$ (two excess copies, namely the second copies of 3 and 5). One checks directly in exact rational arithmetic that

$$S^{(-1)}(M_0^-) = \frac{1}{1} + \frac{2}{2} + \frac{2}{4} + \frac{1}{6} + \frac{1}{10} + \frac{1}{12} + \frac{1}{16} + \frac{1}{18} + \frac{1}{36} + \frac{1}{240} = 3.$$

Phase 2 runs for only 167 iterations (compared to 452 for the $s = +1$ case), peaking at $\mathtt{ct} = 57$ at $j = 48$ and then decreasing to 0 at $j = 166$. The trajectory is plotted in Figure 2.

The qualitative shape of the trajectory is the same as in Figure 1: a climb, a noisy plateau, and a monotone descent.

After 167 iterations the program prints the analogous certificate line

```
RESGOOD 3 [2,3,5,...] 698,
```

listing the elements of the set

$$P_- = \{q_1, q_2, \ldots, q_{698}\} \subset \mathcal{P}$$

in increasing order. The full list is contained in the supplementary file `prime_set_P_minus3.txt`. The same three verifications as in Section 5.1 are performed:

(1) all 698 elements are prime;
(2) the list, sorted in increasing order, has length 698;
(3) the sum $\sum_{p\in P_-} 1/(p-1) = 3$ exactly.

The smallest element of P_- is 2, and the largest is a prime with 68 decimal digits. This proves Theorem 4. □

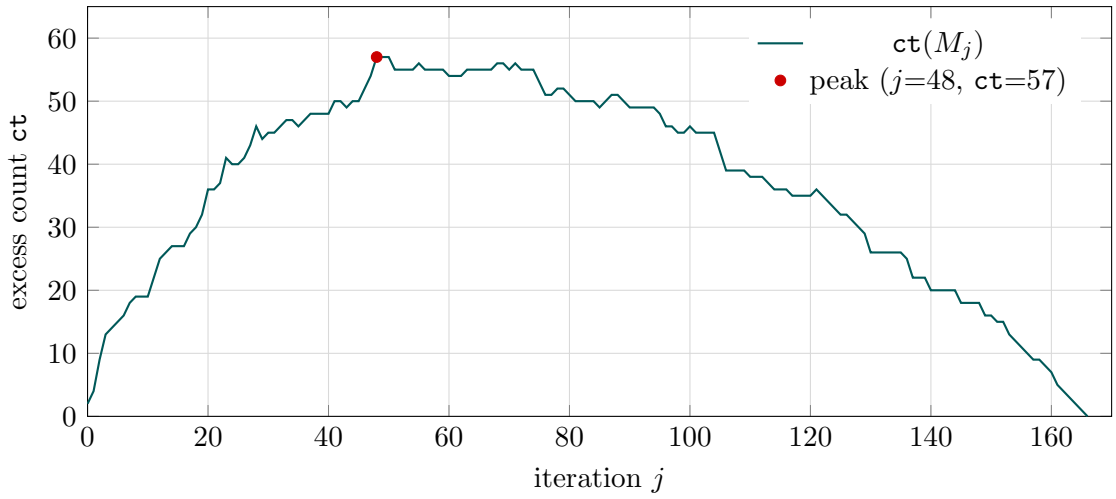


FIGURE 2. Trajectory of ct during Phase 2 for $(r, s) = (3, -1)$

5.3. **Remarks on the two sets.**

Remark 11. The sets P_+ and P_- produced by our algorithm are not claimed to be optimal in any sense. Their cardinalities depend on the splitting tables and on the choice of beam-search candidates; smaller sets certainly exist. Finding the minimum $|P|$ such that $\sum_{p\in P} 1/(p+s) = r$ is a natural open question for each given pair (r, s).

Remark 12. The largest primes in P_+ and P_- have 181 and 68 decimal digits respectively. Such large primes are not surprising: the identity (1) roughly squares the denominator at each splitting step, and several chained splittings during Phase 2 can compound this growth.

ACKNOWLEDGEMENTS

The author thanks Zhi-Wei Sun, Qing-Hu Hou and Wenjie Fang for helpful discussions.

REFERENCES

[1] T. F. Bloom, Unit fractions with shifted prime denominators, *Proc. Roy. Soc. Edinburgh* **155** (2025), 2285–2295.
[2] P. Erdős and E. G. Straus, On the irrationality of certain Ahmes series, *J. Indian Math. Soc. (N.S.)* **27** (1964), 129–133.
[3] Z.-W. Sun, Conjectures on representations involving primes, in: M. Nathanson (ed.), *Combinatorial and Additive Number Theory II*, Springer Proc. Math. Stat., Vol. 220, Springer, Cham, 2017, pp. 279–310; preprint (2012), https://arxiv.org/abs/1211.1588.
[4] The gmpy2 development team, *gmpy2: General Multi-Precision arithmetic for Python*, version 2.x, https://github.com/gmpy2/gmpy2.
[5] R. L. Graham, A theorem on partitions, *J. Austral. Math. Soc.* **3** (1963), 435–441.

I.R.M.A., UMR 7501, UNIVERSITÉ DE STRASBOURG ET CNRS, 7 RUE RENÉ DESCARTES, F-67084 STRASBOURG, FRANCE
Email address: guoniu.han@unistra.fr