The Quest for the Generalized Perfect Numbers

A. V. Lelechenko

Let n be a generalized perfect number if $\sigma_*(n) = kn$ for an integer k and a generalized sum-of-divisors function σ_*. We develop a numerical approach to estimate the lower bound of generalized perfect numbers with given properties and present an effective algorithm to find all generalized perfects below given limits. The program, implementing our algorithm, produces many new, unknown previously samples of generalized perfects.

Keywords: sum-of-divisors function, perfect number.
MSC 2010: 11Y55, 11Y16, 11N56
UDC: 511

Introduction

We say that function s is a generalized sum-of-divisors function if s is a multiplicative arithmetic function and for prime p

$$s(p^a) = \sum_{\substack{0 \le b \le a \\ \mathrm{Pr}(a,b)}} p^b,$$

where Pr is a fixed predicate. Examples of such functions include:

- $\mathrm{Pr}(a,b) = 1$ for all a and b produces $s \equiv \sigma$, which is the usual sum-of-divisors function. See [1, A000396], [2].
- $\mathrm{Pr}(a,b) = [b = a] \vee [b = 0]$ produces the sum-of-unitary-divisors function $s \equiv \sigma^*$. See [1, A002827], [3], [4].
- $\mathrm{Pr}(a,b) =$ [binary digits of b have zeros in all positions, where a's have] gives us the sum-of-infinitary-divisors function $s \equiv \sigma^\infty$. See [1, A007357], [5], [6].
- $\mathrm{Pr}(a,b) = [b$ divides $a]$ produces the sum-of-exponential-divisors function $\sigma^{(e)}$, introduced in [7]. There is also a bunch of other exponential sum-of-divisors functions, which definitions utilize different modifications of the concept of divisibility, including $\sigma^{*(e)}$ for unitary and $\sigma^{\infty(e)}$ for infinitary divisors [8].

We call integer n a generalized $\langle s, m\rangle$-perfect number, if s is a generalized sum-of-divisors function, $m > 0$ is an integer and $s(n) = mn$.

Here and below letters p and q denotes prime numbers.

Note that n is $\langle \sigma^{(e)}, m\rangle$-perfect if and only if $n/\gamma(n)$ is $\langle s', m\rangle$-perfect for function s' defined by $\mathrm{Pr}(a,b) = [b{+}1$ divides $a{+}1]$ (so-called reduced or sum-of-modified-exponential-divisors function), where $\gamma(n) = \prod_{p|n} p$ is an arithmetical kernel of n. The same reduction is valid for all other kinds of exponential sum-of-divisors functions.

Our paper is devoted to computation of generalized perfects and addresses two following questions. First: what can be said about divisibility properties of "small" (e.g., below 10^{500}) generalized perfects of a given kind? Second: how can this information be used for effective and exhaustive search of all such perfects M canonical representation of which includes only primes $< P$ in powers $< A$ for given M, P and A?

Unfortunately, we are able to deal only with a subclass of generalized sum-of-divisors functions, namely, functions s such that $s(p) = p+1$ and $\Pr(a, a-1) = 0$ for $a > 1$. However, this class contains many important and popular functions, e.g., all reduced exponential sum-of-divisors functions, sum-of-unitary-divisors functions and many others.

Divisibility Properties of Generalized Perfects

Let R be a finite set of primes. What can be said about lower bound of $\langle s, m\rangle$-perfects which are not divisible by any element of R? In this section we develop an algorithm, appropriate for automatic computation of such lower bound.

For a fixed s let B be a polynomial of finite degree such that $s(p^a)p^{-a} \le B(p^{-1})$, $a \ge 2$. Then restriction $\Pr(a, a-1) = 0$ implies that B can be chosen so that $B(x)$ has zero first-degree coefficient and unit zero-degree coefficient:

$$B(x) = 1 + O(x^2), \qquad x \to 0. \tag{1}$$

Choose function $b(t)$ such that $b(t) \ge \max_{\tau \ge t} s(2^\tau)2^{-\tau}$. Trivially one can take

$$b(t) = \begin{cases} \max(3/2, B(1/2)) & \text{if } t = 1, \\ B(1/2) & \text{else.} \end{cases}$$

Consider an $\langle s, m\rangle$-perfect number n with canonical representation

$$n = 2^t \prod_{p \in P} p \prod_{q \in Q} q^{a_q},$$

where $P \cap Q = \{\}$, $(P \cup Q) \cap R = \{\}$, $a_q \ge 2$. For any fixed integer N (it is appropriate for applications to take $1 \le N \le 10$) we split P into disjoint subsets $P = P_1 \sqcup \cdots \sqcup P_N$, where

$$P_k = \{p \in P \mid p \equiv -1 \pmod{2^k}, p \not\equiv -1 \pmod{2^{k+1}}\}, \quad k = 1, \ldots, N-1,$$

$$P_N = P \setminus P_1 \setminus \cdots \setminus P_{N-1}.$$

Denote $t_k = |P_k|$. Now since $s(p) = p + 1 \equiv 0 \pmod 2$ for $p > 2$ we have

$$s(n) = s(2^t) \prod_{k=1}^{N} \prod_{p \in P_k} (p+1) \prod_{q \in Q} s(q^{a_q}) \equiv 0 \pmod{2^{\sum k t_k}},$$

so $t + \nu_2(m) \geq \sum_{k=1}^{N} kt_k$, where $\nu_p(n)$ stands for a maximal power of p, dividing n. Since n is $\langle s, m\rangle$-perfect

$$m = \frac{s(n)}{n} \leq b\left(\sum_{k=1}^{N} kt_k - \nu_2(m)\right) \cdot \prod_{k}\prod_{p\in P_k}\left(1+p^{-1}\right)\prod_{q\in Q} B(q^{-1}) \leq$$

$$\leq b\left(\sum_{k=1}^{N} kt_k - \nu_2(m)\right) \cdot \prod_{k}\prod_{p\in P_k}\frac{1+p^{-1}}{B(p^{-1})}\prod_{q\in P\sqcup Q} B(q^{-1}). \quad (2)$$

The last product in the right-hand side is majorized by $\prod_{q\notin R} B(q^{-1})$, which due to the choice of B in (1) converges to a computable constant $C < \infty$. Every product by $p \in P_k$ is majorized by product by first t_k elements of the set of primes

$$\mathcal{P}_k = \Big\{p \mid p \equiv -1 \pmod{2^k},\ p \not\equiv -1 \pmod{2^{k+1}},$$
$$\forall r \in R\ p \not\equiv -1 \pmod{r^{1+\nu_r(m)}}\Big\}, \quad k = 1, \ldots, N-1,$$

$$\mathcal{P}_N = \Big\{p \mid p \equiv -1 \pmod{2^N},\ \forall r \in R\ p \not\equiv -1 \pmod{r^{1+\nu_r(m)}}\Big\}.$$

Thus condition (2) implies following estimate for the lower bound of n:

$$n \geq \min_{t_1,\ldots,t_N}\left\{2^{\sum_{k=1}^{N} kt_k - \nu_2(m)} \cdot \prod_{k=1}^{N}\prod_{p\in\mathcal{P}_k} p \;\middle|\; m/C \leq b\left(\sum_{k=1}^{N} kt_k - \nu_2(m)\right) \cdot \prod_{k=1}^{N}\prod_{p\in\mathcal{P}_k}\frac{1+p^{-1}}{B(p^{-1})}\right\},$$

where $\prod_{p\in\mathcal{P}_k}$ denotes a product with index restricted to first t_k elements of $\mathcal{P}_k$.

The last optimization problem can be solved numerically by brute force, running over $t_1, \ldots, t_N$.

An Algorithm for Exhaustive Search

Suppose that we are searching for $\langle s, m\rangle$-perfects and, utilizing framework from the previous section, we know, that all "small" (below given M) of them are divisible by $p \in R = \{r_1, r_2, \ldots, r_{|R|}\}$. Usually we obtain that $2 \in R$, in many cases $2, 3 \in R$ and in several cases we can show that even $2, 3, 5 \in R$. We restrict our search to the numbers built up from primes below some limit P in powers below some limit A.

Let us present the corresponding (presumably new) algorithm:

1. Build a set of "bricks", which contains all p^a for $p \leq P$, $a \leq A$ such that $s(p^a)$ does not contain prime divisors greater than P. We store bricks in an associative array *Bricks*, indexed with primes. Element *Bricks*$[p]$ is a list of pairs $(s(p^a)/p^a, p^a)$, where p^a satisfies mentioned condition.

2. Build a set of initial pillars: *Inits* is a set of pairs of form
$$\left(1/m \cdot \prod_{k=1}^{|R|} \mathit{Bricks}[r_k][a_k].\mathit{fst}, \prod_{k=1}^{|R|} \mathit{Bricks}[r_k][a_k].\mathit{snd}\right),$$
where each a_k runs from 1 to the length of $\mathit{Bricks}[r_k]$.
3. Feed each pair of *Inits* into the recursive routine *Builder*. The latter consequently checks following conditions:
   (a) If a ratio in input pair (*pair.fst*) equals to 1, then the number (*pair.snd*) is an $\langle s, m\rangle$-perfect and it is send to output.
   (b) If the numerator of *pair.fst* is not coprime with *pair.snd*, then this factor cannot be cancelled on successive steps, so $\langle s, m\rangle$-perfect can not have *pair.snd* as a unitary divisor. So nothing is sent to output.
   (c) Otherwise if the numerator of ratio *pair.fst* is not equal to 1, choose p such that $\mathit{Bricks}[p]$ is the shortest of all possible p, which divides the numerator of *pair.fst*. If there are several p with equal length of $\mathit{Bricks}[p]$, choose the smallest. Run *Builder* on each pair of form
   $$\big(\mathit{pair.fst} \cdot \mathit{Bricks}[p][a].\mathit{fst}, \mathit{pair.snd} \cdot \mathit{Bricks}[p][a].\mathit{snd}\big)$$
   for $a = 1, \ldots, \big|\mathit{Bricks}[p]\big|$.
   (d) The case when the numerator of ratio *pair.fst* equals to 1 is the trickiest one. We cannot continue building a pillar in a way described above. Instead we should look for $\langle s, 1/\mathit{pair.fst}\rangle$-perfects, coprime with *pair.snd*: each such perfect, multiplied by *pair.snd*, will produce $\langle s, m\rangle$-perfect.
   We check using the framework from the previous section whether there are $\langle s, 1/\mathit{pair.fst}\rangle$-perfects coprime with *pair.snd* below M (usually there is no). If they are then find them running *Builder* recursively with $m = 1/\mathit{pair.fst}$.

The proposed algorithm was implemented as a computer program, written in Haskell. The choice of the language was determined by appropriate abilities of standard library, concise and idiomatic syntax, which fits well for recursive algorithms, laziness by default and easy opportunity to run code on multiple cores.

All ratios in the course of algorithm are stored as two associative arrays, each of them stores canonical representation of numerator and denominator correspondingly. Such trade-space-for-speed optimization has allowed to increase the speed of computations greatly: we almost completely avoid costly factorization of large numbers, replacing them with set operations on multisets.

The method of choice of p at step (c) is extremely important: it allows to examine less number of combinations and cut branches earlier.

The source code can be found at https://bitbucket.org/Bodigrim/perfect. The source code is accompanied by many examples of found $\langle s, m\rangle$-perfects, some of which were previously unknown, for different s and m.

Conclusions

It is worth mentioning that arguments above can be easily transferred on the class of alternating sum-of-divisors functions. The simplest representative of them is β, which is a multiplicative arithmetic function with $\beta(p^a) = \sum_{b=0}^{a}(-1)^{a-b}p^b$. Numbers n with a property $\beta(n) = n/m$ are called m-imperfect. Several examples of 2-imperfects and 3-imperfects were known before (see [9]), but no examples of 4-imperfects were known. However the proposed algorithm (slightly modified for the case of imperfects) has found a bunch of 4-imperfects at once.

References

[1] N. J. A. Sloane, ed., *The on-line encyclopedia of integer sequences.* 2014.

[2] R. L. Sorli, *Algorithms in the study of multiperfect and odd perfect numbers.* Ph. D. thesis, University of technology, Sydney, 2003.

[3] C. R. Wall, "The fifth unitary perfect number," *Canad. Math. Bull.*, vol. 18, no. 1, pp. 115–122, 1975.

[4] C. R. Wall, "On the largest odd component of a unitary perfect number," *Fib. Quart.*, vol. 25, pp. 312–316, 1987.

[5] G. L. Cohen, "On an integer's infinitary divisors," *Math. Comput.*, vol. 54, pp. 395–411, jan 1990.

[6] G. L. Cohen and P. Hagis Jr., "Arithmetic functions associated with the infinitary divisors of an integer," *Int. J. Math. Math. Sci.*, vol. 16, no. 2, pp. 373–383, 1993.

[7] M. V. Subbarao, "On some arithmetic convolutions," in *The theory of arithmetical functions: Proceedings of the Conference at Western Michigan University, April 29 — May 1, 1971*, vol. 251 of *Lecture Notes in Mathematics*, (Berlin), pp. 247–271, Springer Verlag, 1972.

[8] N. Minculete and L. Tóth, "Exponential unitary divisors," *Ann. Univ. Sci. Budap. Rolando Eötvös, Sect. Comput.*, vol. 35, pp. 205–216, 2011.

[9] W. Zhou and L. Zhu, "On k-imperfect numbers," *Integers*, vol. 9, no. 1, p. A01, 2009.

Authors

Andrew Wolodymyrowych Lelechenko — the 3rd year postgraduate student, Department of Computer Algebra and Discrete Mathematics, Faculty of Mathematics, Institute of Mathematics, Economics and Mechanics, I.I. Mechnikov Odessa National University, Odessa, Ukraine; E-mail: *1@dxdy.ru*