British Journal of Mathematics & Computer Science
8(5): 361-365, 2015, Article no.BJMCS.2015.169
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

SCIENCEDOMAIN

Complexity of Finding Values of the Generalized Taxicab Number

Alexander Bolotin[1*]

[1]*Department of Computer Science, Ben-Gurion University of the Negev, Beersheba, Israel.*

Article Information

DOI: 10.9734/BJMCS/2015/17549
Editor(s):
(1) Kai-Long Hsiao, Taiwan Shoufu University, Taiwan.
Reviewers:
(1) Grienggrai Rajchakit, Department of Mathematics, Maejo University, Thailand.
(2) W. Obeng-Denteh, Mathematics Department, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=1034&id=6&aid=8839

Original Research Article

Received: 17 March 2015
Accepted: 08 April 2015
Published: 15 April 2015

Abstract

This short paper demonstrates a sketch of a generic algorithm that can generate integers written as the sum of N m^{th} positive powers with any $N \geq 2$ and $m \geq 1$ in two different ways. As it is shown in the paper, such a generic algorithm is NP-hard. One can infer from this result that to generate exact values of the generalized Taxicab (m, N, j)– the smallest sum of N m^{th} positive powers expressed in j different ways – is at least as hard as the hardest problems in NP. This implies that except for small N to find the generalized Taxicab (m, N, j) would be unfeasible on any computing device.

Keywords: Generalized taxicab number; number partitioning problem; integer factorization; complexity; NP-hardness; NP-complete problems.

1 Introduction

According to the definition, the smallest integer that can be written as the sum of *two positive cubes in j different ways* is called the j^{th} taxicab number and it is denoted as Ta(j) or Taxicab $(3,2,j)$. Owing to Fermat's proof (see the Theorem 412 of Hardy & Wright [1]), it is known that Ta(j) exists for any j. Here is an example of Ta(4) found in [2]:

$$\begin{aligned} &6963472309248 = \\ &= 2421^3 + 19083^3 = \\ &= 5436^3 + 18948^3 = \\ &= 10200^3 + 18072^3 = \\ &= 13322^3 + 16630^3 \ . \end{aligned} \tag{1}$$

*Corresponding author: arkady_bolotin@hotmail.com;

Formally, the concept of taxicab numbers may well be extended to higher powers of summand positive integers and to a greater amount of such powers. The question then arises: Does the generalized taxicab number Taxicab (m, N, j) – i.e., the smallest number expressible as the sum of N m^{th} positive powers in j different ways, where $m \geq 1$, $N \geq 2$ and $j > 1$– exist?

The answer to this question is still not clear. For example, even though the smallest j-way sums of fourth powers, i.e., Taxicab $(4,3,j)$, are known [3], e.g.,

$$\begin{aligned} &\text{Taxicab}(4,3,3) = 811538 = \\ &= 29^4 + 17^4 + 12^4 = \\ &= 28^4 + 21^4 + 7^4 = \\ &= 27^4 + 23^4 + 4^4 \;, \end{aligned} \tag{2}$$

no positive integer T, which can be written as the sum of two fifth powers in more than one way, e.g.,

$$T := a_1^5 + a_2^5 = b_1^5 + b_2^5 \;, \tag{3}$$

has been found [4].

As to the theoretical proof of Ta(j) existence, it cannot be used to find the actual values of Ta(j). Neither can it be used to find values of the generalized Taxicab (m, N, j).

The present paper will demonstrate a sketch of a straight forward generic algorithm capable of generating integers T written as the sum of N m^{th} positive powers in *two different ways*:

$$T := a_1^m + \cdots + a_N^m = b_1^m + \cdots + b_N^m, \tag{4}$$

where positive integers a_i, and b_i are such that

$$\{a_1, \dots, a_N\} \cap \{b_1, \dots, b_N\} = \emptyset \;. \tag{5}$$

As it will be shown in the paper, this generic algorithm is NP-hard.

2 The Algorithm Generating Integers written as the Sum of Nm^{th} Positive Powers in two Different Ways

We will start the algorithm with solving *the problem of number partitioning* (N_{PP}). Namely, given the set S of P randomly generated positive integers x_i

$$\text{S} = \{x_1, \dots, x_P\} \;, \tag{6}$$

is there a partitioning of this set S into two disjoint subsets R and $(S - R)$ such that the sum of the elements $x_i^{'}$ in the first subset R and the sum of the elements $x_i^{''}$ in the second subset $(S - R)$ are the same?

If the answer to this question is 'yes', that is,

$$\sum_{i=1}^{P_R} x_i^{'} = \sum_{k=1}^{P_{(S-R)}} x_k^{''} \text{ where } x_i^{'} \text{ and } x_k^{''} \in \text{S}; \; P = P_R + P_{(S-R)} \;, \tag{7}$$

then at the next step the algorithm will ask: Is the number of the elements P_R equal to the number of elements $P_{(S-R)}$? In other words, does the given set S have a partition into two halves with equal sum?

Suppose that the answer is 'yes' again, that is, $P_R = P_{(S-R)} = N$:

$$x_1^{'} + \cdots + x_N^{'} = x_1^{''} + \cdots + x_N^{''} \quad . \tag{8}$$

Then the next question of the algorithm will be: Can the elements $x_i^{'}$ and $x_i^{''}$ be expressible as the same m^{th} positive powers? In other words, can every $x_i^{'}$ and $x_i^{''}$ be decomposed (factorized) into the product of m integers a_i and b_i such that

$$\forall i \in \{1, \dots, N\}: \quad \begin{cases} x_i^{'} \equiv a_i \text{mod}(a_i) \\ x_i^{''} \equiv b_i \text{mod}(b_i) \end{cases} \quad ? \tag{9}$$

It is clear that in the case, in which the answer is 'yes', the sum of the elements $x_i^{'}$ or $x_i^{''}$ will provide the value of the integer T expressible as the sum of N m^{th} positive powers in two different ways.

As follows, the sketched above generic algorithm consists of the three principal parts: (1) solving number partitioning, (2) calculating set cardinality, and (3) factoring integers. Let us evaluate difficulty and complexity of each of those parts.

3 Complexity of the Algorithm

As one can see, the runtime bottleneck of the outlined algorithm is the solution of the N_{PP}.

Indeed, to perform the part 2 of the algorithm, i.e., to calculate the number of elements in the set S is not hard – it can be done in polynomial in N numbers of steps.

To complete the part 3 does not appear to be hard as well. Let us formulate this part as the following decision problem: Given an integer $x_i^{'}$ (or $x_i^{''}$) and an integer a_i (or b_i) with $1 < a_i < x_i^{'}$ (or $1 < b_i < x_i^{''}$), does $x_i^{'}$ have a factor a_i? Since no reduction from any NP-complete problem (such as the 3SAT problem, the traveling salesman problem, or any other problem discussed in [5]) to this decision problem has been found [6,7], we do not have any reason to believe that the part 3 is hard (the informal consensus is that integer factorization is one of the "in-between" problems that are not in P and are not NP-complete).

In contrast, to accomplish the first part of the algorithm, i.e., to solve number partitioning, is supposed to be hard given that the N_{PP} is proven NP-complete. This means that all NP problems (i.e., decision problems whose 'yes' solutions can be verified in polynomial number of steps) are polynomial-time reducible to the N_{PP}. Therefore, finding an efficient (i.e., polynomial-time) solution for the N_{PP} would imply that an efficient (polynomial-time) solution could be found for all NP problems (which would entail P=NP), since any problem belonging to the class NP can be recast into any other member of this class [8,9].

Since it is generally believed that P≠NP, we expect that such a solution does not exist and thus the generic algorithm for generating integers T written as (4) would take, in the worst case, an exponential in N amount of time.

4 Concluding Remarks

It is obvious that the task of finding the generalized Taxicab(m, N, j) cannot be easier than generating integers T expressible as the sum of N m^{th} positive powers in two different ways. Consequently, it is reasonable to conclude that the generic exact algorithm for finding values of the generalized Taxicab (m, N, j) is at least supposed to be NP-hard, which means that unless P=NP to find the generalized Taxicab (m, N, j) for arbitrary N would be unfeasible (on a deterministic machine and a quantum computer alike).

This does not exclude that values of Taxicab (m, N, j) can be found for small N.

Another way to deal with NP-hardness of the generalized Taxicab (m, N, j) might be to abandon the idea of finding exact values of Taxicab (m, N, j) and instead to seek an approximate but fast heuristic algorithm that can generate estimated intervals of Taxicab (m, N, j) values.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Hardy G, Wright E. An introduction to the theory of numbers. 5th edition. 1979;330. Available:https://archive.org/details/AnIntroductionToTheTheoryOfNumbers-4thEd-G.h.HardyE.m.Wright

[2] Rosenstiel E, Dardis J, Rosenstiel C. The four least solutions in distinct positive integers of the Diophantine equation s = x3 + y3 = u3 + v3 = m3 + n3, Bulletin Inst. Math. Appl. 1991;27:155-157. Available: http://www.cix.co.uk/~rosenstiel/cubes/welcome.htm

[3] Moore D. Generalized taxicab numbers and cabtaxi numbers. Available: http://homepage.ntlworld.com/duncan-moore/taxicab/

[4] Guy R. Unsolved problems in number theory (third edition). New York, New York, USA: Springer-Science + Business Media, Inc. 2004;437. ISBN 0-387-20860-7. Contents and Preview are available: http://www.springer.com/us/book/9780387208602

[5] Karp R. Reducibility among combinatorial problems. In: Miller R, Thatcher J. (editors). Complexity of Computer Computations. New York: Plenum. 1972;85–103. Available: http://www.cs.berkeley.edu/~luca/cs172/karp.pdf

[6] Dixon J. Asymptotically fast factorization of integers. Math. Comp. 1981;36:255–260. Available: www.jstor.org/stable/2007743

[7] Fortnow L. Computational complexity blog: Complexity Class of the Week: Factoring. Available: http://weblog.fortnow.com/2002/09/complexity-class-of-week-factoring.html

[8] Garey M, Johnson D. Computers and intractability: a guide to the theory of NP-completeness. New York: Freeman & Co.; 1979. ISBN 0-7167-1045-5. MR 519066. Available:http://udel.edu/~heinz/classes/2014/4-667/materials/readings/Garey%20and%20Johnson/Chapter%20One.pdf

[9] Papadimitriou C, Steiglitz K. Combinatorial optimization: Algorithms and complexity, dover, New York; 1998. Preface and Contents available: http://tocs.ulb.tu-darmstadt.de/116091606.pdf

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=1034&id=6&aid=8839