Solving Diophantine Equations on a Network of Workstations

Rizos Sakellariou

Department of Computer Science, University of Manchester,
Oxford Road, Manchester M13 9PL, U.K.

1 Introduction

In 1769, Euler, by generalising Fermat's Last Theorem, conjectured that "it is impossible to exhibit three fourth powers whose sum is a fourth power", "four fifth powers whose sum is a fifth power, and similarly for higher powers". The first counterexample to the conjecture, was found in 1966 after a systematic computer search: $27^5+84^5+110^5+133^5 = 144^5$. More recently, Euler's conjecture was also disproved for fourth powers, and a method of generating an infinity of solutions to the equation $A^4 + B^4 + C^4 = D^4$ is described in [1]. It is not known whether there are any counterexamples to Euler's conjecture for powers higher than the fifth. The general case of the problem of finding equal sums of like powers, that is, solving the diophantine equation

$$x_1^k + x_2^k + \ldots + x_m^k = y_1^k + y_2^k + \ldots + y_n^k, \quad 1 \leq m \leq n, \tag{1}$$

for positive integers x_i, y_j, has been studied for various k, m, n [3, Problem D1]. This paper is concerned with the particular problem of finding the least n for which the above equation, for given k, m, is solvable. A computational attack, for $7 \leq k \leq 10$, is made, and several hitherto unpublished solutions are reported. Our approach exploits the *embarassingly parallel* [2] nature of the problem on a network of Sun workstations.

2 Methodology and Results

In order to find solutions of (1), a simple, general-purpose subroutine that can find a representation of a given integer, z, as a sum of n kth powers, if such a representation exists, has been developed. The algorithm is based on the recursive selection of potential y_j, in decreasing order, by excluding those values that cannot sum up to z. The subroutine is applied exhaustively to a range of values for x_i. Using idle time on a network of approximately 100 Sun workstations, a series of Unix shell scripts were written to distribute overnight tasks based on input from a list of 'most demanded' x_i, ranked according to estimated performance requirements. As a starting basis for our search we used the results collected by [4]; searches were initially conducted for the least n for which a solution was known (until a solution for a smaller n was found), and for $7 \leq k \leq 10$.

k, m, n	x_i	y_j
7, 2, 7	128, 5	122, 106, 69, 60, 33, 31, 6
7, 3, 5	96, 41, 17	86, 77, 77, 68, 56
	153, 43, 14	140, 137, 59, 42, 42
8, 4, 6	47, 29, 12, 5	45, 40, 30, 26, 23, 3
9, 1, 14	66	63, 54, 51, 49, 38, 35, 29, 24, 21, 12, 10, 7, 2, 1
9, 3, 9	38, 38, 3	41, 23, 20, 20, 18, 13, 13, 12, 9
9, 4, 9	38, 31, 12, 2	36, 32, 32, 30, 15, 13, 8, 4, 3
9, 5, 7	35, 26, 15, 15, 12	33, 32, 24, 16, 14, 8, 6
10, 1, 22	33	30, 30, 26, 26, 23, 21, 19, 18, 13, 13, 12, 12, 10, 10, 10, 10, 10, 9, 9, 7, 6, 3
	44	43, 36, 30, 30, 28, 26, 25, 24, 24, 24, 20, 20, 18, 15, 15, 15, 10, 9, 9, 8, 7, 2
10, 2, 15	35, 3	33, 32, 24, 21, 20, 20, 13, 13, 13, 12, 11, 9, 7, 1, 1
10, 3, 14	30, 28, 4	31, 23, 20, 20, 17, 17, 16, 10, 9, 9, 9, 5, 2, 2
	38, 37, 11	39, 34, 29, 27, 27, 22, 21, 18, 17, 15, 13, 9, 9, 8
10, 4, 13	33, 24, 22, 6	31, 29, 26, 25, 23, 23, 23, 21, 18, 13, 9, 8, 4
10, 5, 15	20, 11, 8, 3, 1	18, 18, 17, 16, 10, 7, 7, 4, 4, 4, 4, 4, 4, 2, 2

Table 1. New solutions of (1) for various k, m, n.

	k								
m	2	3	4	5	6	7	8	9	10
1	2	3	3	4	7	8	11	14	22
2		2	2	3	7	7	9	12	15
3					3	5	8	9	14
4						4	6	9	13
5							5	7	15
6								6	27
7									7

Table 2. Least n for which a solution to (1) is known.

Several new solutions of (1) have been found over a six-month period and after approximately 100,000 CPU hours; some of the smallest solutions are presented in Table 1 (more results are given in [5]). A survey, showing the least n for which a solution to (1) is known, is given in Table 2. Further improvement of these results would be better achieved through special programs developed for given k, m, n. The problem has overwhelming computational needs but with a high degree of parallelism that can be exploited over a large computer network.

References

1. N. D. Elkies. On $A^4 + B^4 + C^4 = D^4$. *Mathematics of Computation*, 51, 1988, pp. 825–835.
2. I. Foster. *Designing and Building Parallel Programs.* Addison-Wesley, 1995.
3. R. K. Guy. *Unsolved Problems in Number Theory.* Springer-Verlag, 1994.
4. L. J. Lander, T. R. Parkin and J. L. Selfridge. A Survey of Equal Sums of Like Powers. *Mathematics of Computation*, 21, 1967, pp. 446–459.
5. R. Sakellariou. On Equal Sums of Like Powers, unpublished manuscript. See also `http://www.cs.man.ac.uk/cnc/EqualSums/`