EXPLORING THE TREE OF NUMERICAL SEMIGROUPS

JEAN FROMENTIN AND FLORENT HIVERT

Abstract. In this paper we describe an algorithm visiting all numerical semigroups up to a given genus using a well suited representation. The interest of this algorithm is that it fits particularly well the architecture of modern computers allowing very large optimizations: we obtain the number of numerical semigroups of genus $g \leqslant 67$ and we confirm the Wilf conjecture for $g \leqslant 60$.

Introduction

A *numerical semigroup* S is a subset of $\mathbb{N}$ containing 0, closed under addition and of finite complement in $\mathbb{N}$. For example the set

$$S_E = \{0, 3, 6, 7, 9, 10\} \cup \{x \in \mathbb{N}, x \geqslant 12\} \tag{1}$$

is a numerical semigroup. The *genus* of a numerical semigroup S, denoted by $g(S)$, is the cardinality of $\mathbb{N} \setminus S$. For example the genus of S_E is 6, the cardinality of $\{1, 2, 4, 5, 8, 11\}$.

For a given positive integer g, the number of numerical semigroups of genus g is finite and is denoted by n_g. In J.A. Sloane's *on-line encyclopedia of integer sequences* [?] we find the values of n_g for $g \leqslant 52$. These values have been obtained by M. Bras-Amorós ([?] for more details for $g \leqslant 50$). On his home page [?], M. Delgado gives the value of n_{55}.

M. Bras-Amorós used a depth first search exploration of the tree of numerical semigroups $\mathcal{T}$ up to a given genus. This tree was introduced by J.C. Rosales and al. in [?] and it is the subject of Section 1. Starting with all the numerical semigroups of genus 49 she obtained the number of numerical semigroups of genus 50 in 18 days on a pentium D runing at 3GHz. In the package `NumericalSgs` [?] of `GAP` [?], M. Delgado together with P.A. Garcia-Sanchez and J. Morais used the same method of exploration.

Here we describe a new algorithm for the exploration of the tree of numerical semigroups $\mathcal{T}$ and achieve the computation of n_g for $g \leqslant 67$. The cornerstone of our method is a combinatorial representation of numerical semigroups that is well suited and allows large code optimization essentially based on the use of vectorial instructions and parallelization. The goal of the paper is twofold: first to present our encoding of numerical semigroups and the associated algorithms, and second to present the optimization techniques which allow, for those kinds of algorithms, to get speedups by factors of hundreds and even thousands. We claim that these techniques are fairly general for those kinds of algorithms. As a support for the claim, we applied it to an algorithm of N. Borie enumerating integer vector modulo permutation groups [?] and got a speedup by a factor larger than 2000 using 8 cores.

The paper is divided as follows. In Section 1 we describe the tree of numerical semigroups and give bounds for some parameters attached to a numerical semigroup. The description of our representation of numerical semigroups is done in the second section. In Section **??** we describe an algorithm based on the representation given in Section **??** and give its complexity. Section **??** is more technical, and is devoted to the optimization of the algorithm introduced in Section **??**. In the last section we emphasize the results obtained using our algorithm.

1. The tree of numerical semigroups

We start this section with definitions and properties of numerical semigroups that will be used in the sequel. For a more complete introduction, the reader can usefully consult the book *Numerical Semigroups* by J.C. Rosales and P.A. García-Sánchez [?] or the book *The Diophantine Frobenius Problem* by J.L. Ramírez Alfonsín [?].

Date: April 10, 2014.
2010 *Mathematics Subject Classification.* 05A15,68R05,68W10.
Key words and phrases. semigroups, tree, algorithm, `SSE`, `Cilk`, optimization.