# Testing k-cover, k-partition and k-packing

Mats Rydberg

September 2, 2013

# 1 Introduction

In this text, I will present some results from my work in the form of benchmarks for a few variations of the algorithms examined in this Master's Thesis. The core of the algorithm is the Fast Zeta Transform in Linear Space, see 2, which makes the algorithms look very much alike. The algorithm is presented in this paper in three versions, each solving a problem from the Set Cover family of NP-complete problems [citation needed]. These are $k$-cover, $k$-partition and $k$-packing, further explained below.

One question that this test seeks to answer, or at least make an indication toward an answer for, is whether the complexity of the arithmetics performed with polynomials in the partition and packing problems is a significant time-consuming part of the algorithm. That is, would it be worthwhile to represent the polynomials in a non-explicit way in order to perform faster multiplication and addition? In this text, I have provided two different alternatives to polynomial representation (and arithmetics).

# 2 Fast Zeta Transform in Linear Space

The core of the algorithm is the Fast Zeta Transform (FZT) in Linear Space, presented by Björklund et al in [**?**]. The FZT is directly translated into a solution for the three problems mentioned above, by a simple line of summation. But we perform tests on this core also, and try to determine its alleged linear increase in complexity as $|\mathcal{F}|$ increases.

# 3 Representation of polynomials

How to handle polynomial representation is an open question, relevant for $k$-partition and $k$-packing. Two approaches comes to mind; firstly the explicit, direct representation of polynomials as a vector of coefficients, where addition and multiplication is performed in time polynomial to the degree of the polynomial. Secondly, one could evaluate each polynomial at sufficiently many (constant number?) small integers, and recover the coefficients by interpolation and the Chinese Remainder Theorem [**?**].

## 3.1 Complexity of arithmetics

# 4 Test structure

## 4.1 FZT in Linear Space

## 4.2 $k$-cover

## 4.3 $k$-partition

## 4.4 $k$-packing

# 5 Conclusions