Comparison of reduction of the samples number algorithms

Agata Gniewek

January 27, 2016

Class: Intermediate Project
Instructor: PhD Witold Paluszyński
Department of Electronics
Wrocław University of Technology

Abstract

The report describes some algorithms used to decrease number of samples in a signal gathered from quick access data recorder. The goal was to remove as many samples as possible with the smallest error. The main assumption was that the maximal error cannot be exceeded. Five algorithms were tested, each with different parameters. Error caused by them was evaluated and the best parameter value for each of them was selected. The implemented algorithms were compared, the RDP algorithm turned out to be the best one. The removal of about 80% of samples was possible for the testing signal.

Contents

1 Introduction

1.1 Goal

The main goal of the project is decreasing samples number in the dataset gathered with the quick access data recorder [1]. The number of removed samples should be as large as possible but the error cannot exceed the given maximum. In the project few different algorithms will be tested and compared.

1.2 Assumptions

The following assumptions where introduced:

- the maximum error of each value is known, it will be provided by the airplane constructor,
- the approximation that in any point exceeds the maximum error is rejected,
- the different algorithms will be compared using the squared error.

1.3 Programming environment

Programming language: `Python 3.5`[2].
Libraries used:

- `NumPy 1.10.1`[3] – mathematical and matrix operations,
- `SciPy 0.16.1`[4] – interpolation,
- `matplotlib 1.5.0`[5] – visualization of the results,
- `rdp 0.6`[6] – Ramer–Douglas–Peucker algorithm.

2 Algorithms description

2.1 Decimation

The simplest method of decreasing the sample number is decimation [7]. The algorithm takes every n-th sample. Of course some significant samples can be omitted and this algorithm does not prevent from it. It was implemented and tested only as a reference for the others.

The algorithm takes one parameter:

- n – length of step between samples that should not be removed.

2.2 Approximation with polyline

Next, more complicated approach is approximation of the dataset with a polyline and keeping only its nodes. For this purpose the Ramer–Douglas–Peucker algorithm [8] was used.

The algorithm can be discribed in the following list of steps:

1. Take first and last sample.
2. Check if there exists sample which distance to the current polyline is greater than ε. If yes continue, else end.
3. Add sample with the greatest distance to polyline.
4. Return to step 2.

The algorithm takes one parameter:

- ε – maximum distance from polyline.

2.3 Maximum difference

This algorithm checks if the distance between two consecutive samples is not too big. The algorithm can be discribed in the following list of steps:

1. Take first sample.
2. Take the next sample. If the distance to the last kept one is smaller than maximal remove it, otherwise keep it.
3. If there are some samples left: return to step 2.

The algorithm takes one parameter:

- ε – maximum distance between two samples.

2.4 Random algorithm based on samples variance

The algorithm calculates probability that the sample should be kept based on the variance of previous n points and the current one. The probability distribution is linear and equal 0 for the variance 0 and 1 if the variance is equal or greater than the given maximum.

The algorithm can be discribed in the following list of steps:

1. Take first n samples.

2. Take the next sample. Calculate the probability of keeping the sample:

$$p = \max(1, \frac{var}{var_{max}})$$

.

3. Decide based on the probability whether to keep or remove the sample.

4. If there are some samples left: return to step 2.

The algorithm takes two parameters:

- n – number of samples to calculate variance,
- var_{max} – maximum variance.

2.5 Random algorithm based on second derivative

The algorithm calculates probability that the sample should be kept based on the second derivative of the third order spline in the given point. The probability distribution is linear and equal 0 for the derivative 0 and 1 if the derivative is equal or greater than the given maximum.

The algorithm can be discribed in the following list of steps:

1. Interpolate the dataset with the third order spline.

2. Calculate second derivative in the given points.

3. Take first sample.

4. Take the next sample. Calculate the probability of keeping the sample:

$$p = \max(1, \left|\frac{der}{der_{max}}\right|)$$

.

5. Decide based on the probability whether to keep or remove the sample.

6. If there are some samples left: return to step 2.

The algorithm takes two parameters:

- der_{max} – maximum derivative.

3 Results

3.1 Evaluation of the results

3.1.1 Interpolation

All reduced sets of data were interpolated with first and third order splines. The function `interp1d` from `SciPy` was used. The given samples were treated as the set of knots. Thanks to this we can calculate error in each point of the input dataset.

3.1.2 Maximal error

The maximal error out of all samples was calculated. The interpolation of the reduced dataset is dismissed if the maximal error exceeds the given threshold in any point. We assume that the threshold will be provided by airplane constructor and will not be a parameter of the program.

3.1.3 Square error

The sum of squared errors of interpolation in each point was calculated. The value was saved only if the maximal error was not exceeded. This value is used to compare parameters and different algorithms.

3.1.4 Length difference

Length difference of input and reduced dataset is simply the number of samples removed from the input dataset. This value is used to compare parameters and different algorithms.

3.2 Selection of parameter values

To choose the best parameter value we have to find the place where we remove the highest number of samples with the smallest possible error.

3.3 Relationship between error and parameters

In the figures 1 – 5 the relation between error and parameter values is plotted. In the plot for variance algorithm the markers size indicate the error, as there are two parameters. All curves are increasing so we have to choose as small parameter value as possible. It will be determined by number of removed samples.

3.4 Relationship between number of removed samples and parameters

In the figures 6 – 10 the relation of number of removed samples and parameter values is plottet. In the plot for variance algorithm the markers size indicate the number of removed

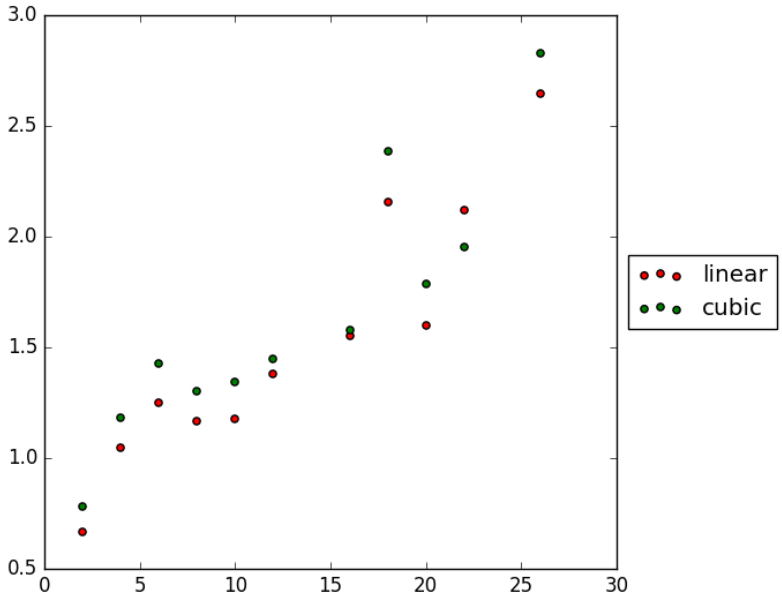

Figure 1: Decimation – relationship between error and parameter value

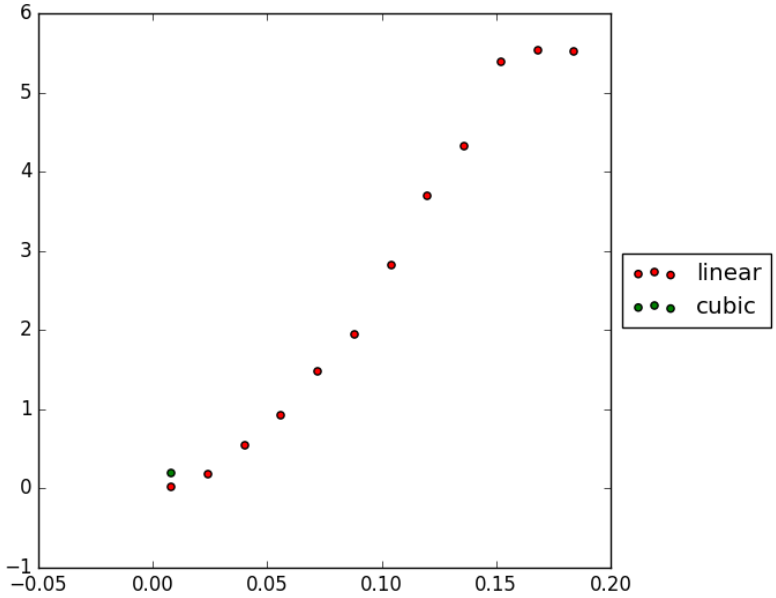

Figure 2: RDP – relationship between error and parameter value

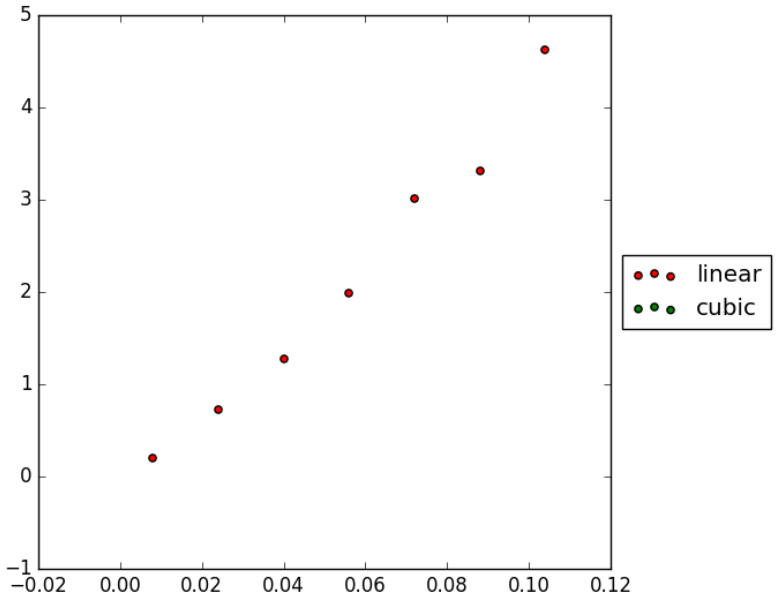

Figure 3: Maximum difference – relationship between error and parameter value

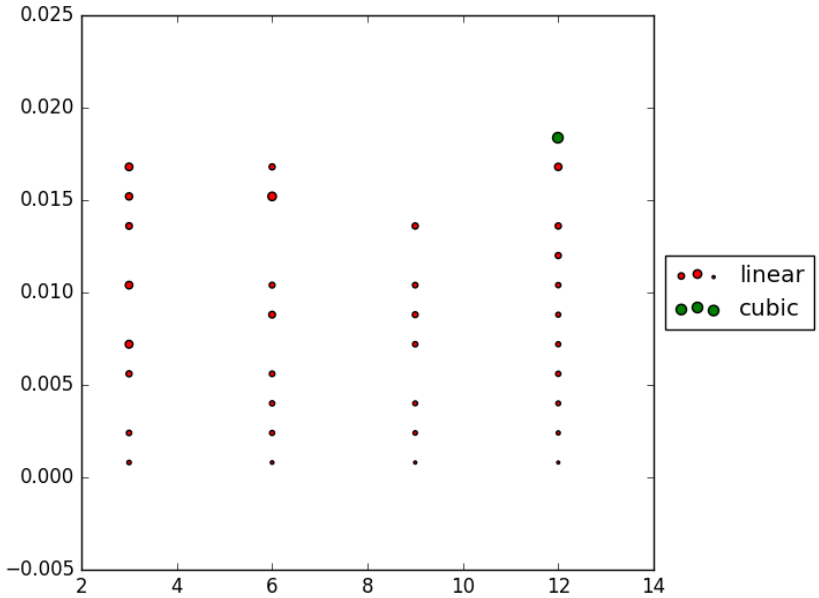

Figure 4: Variance – relationship between error and parameter values

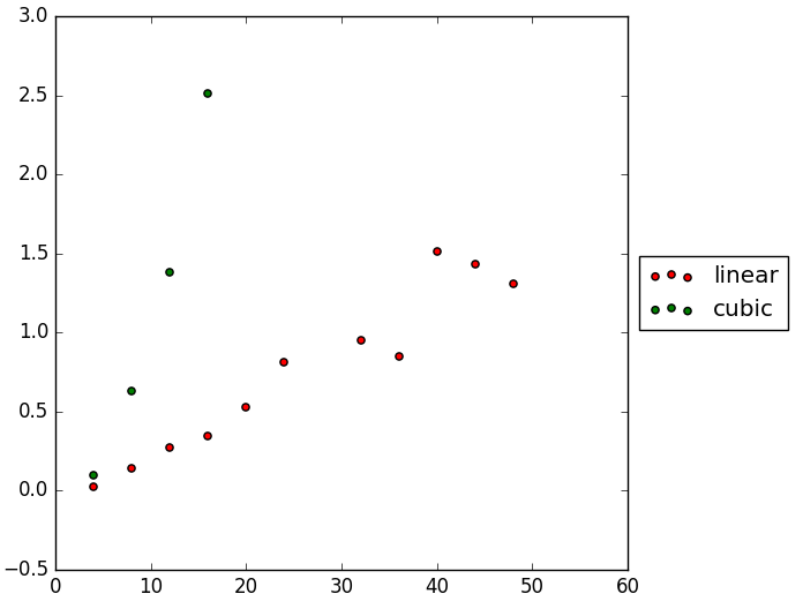

Figure 5: Derivative – relationship between error and parameter value

samples, as there are two parameters. The curves have logarithmic shape so we have to choose the place where the number of removed samples stops (or slows down) its growth. The parameter value in this place would be an optimal one.

3.5 Algorithms comparison

The plot that compares all algorithms is shown in the figure 11. We can see that for almost all algorithms (except decimation) cubic interpolation is much worse than linear one. For all algorithms the error starts to grow exponentially about 1900 removed samples.

4 Conclusions

- All algorithms behave similar.
- The RDP algorithm is slightly better than all the other algorithms – it has a little lower error.
- The selection of the algorithm or parameters might be different for another data. It has to be tested on the real data.

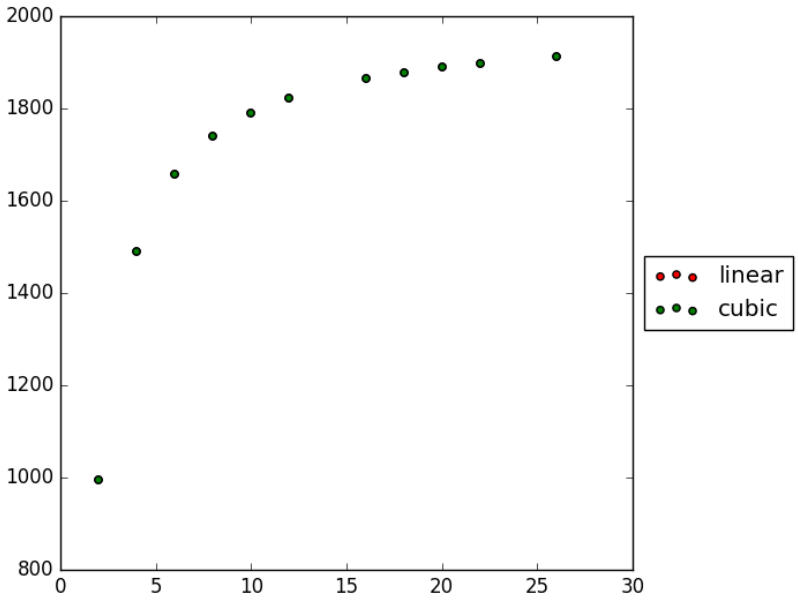

Figure 6: Decimation – relationship between number of removed samples and parameter value

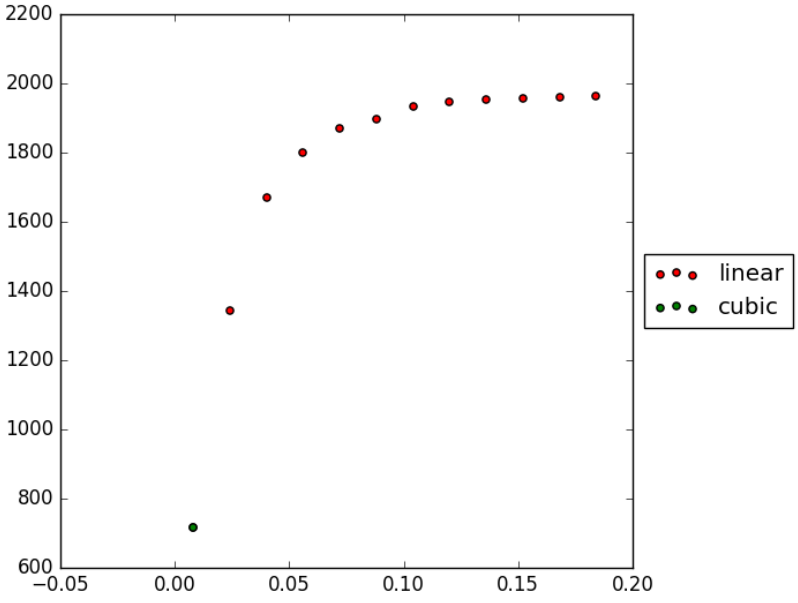

Figure 7: RDP – relationship between number of removed samples and parameter value

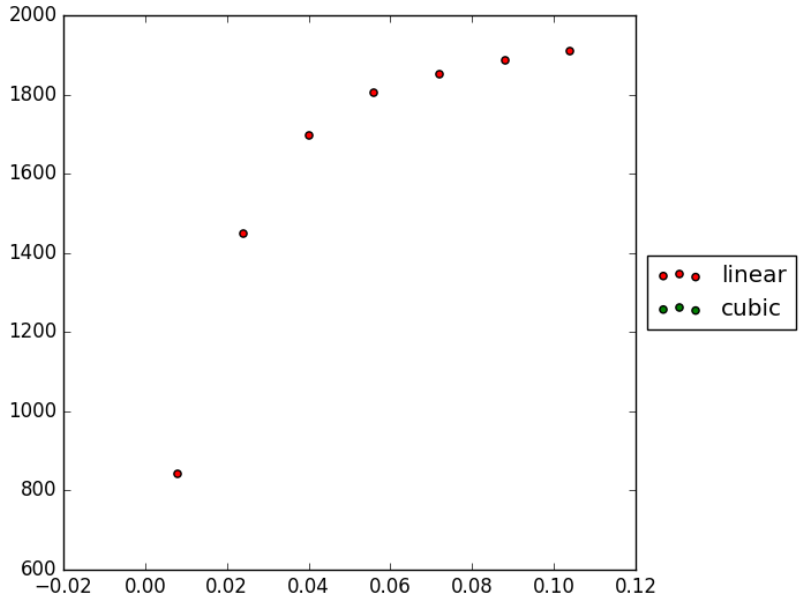

Figure 8: Maximum difference – relationship between number of removed samples and parameter value

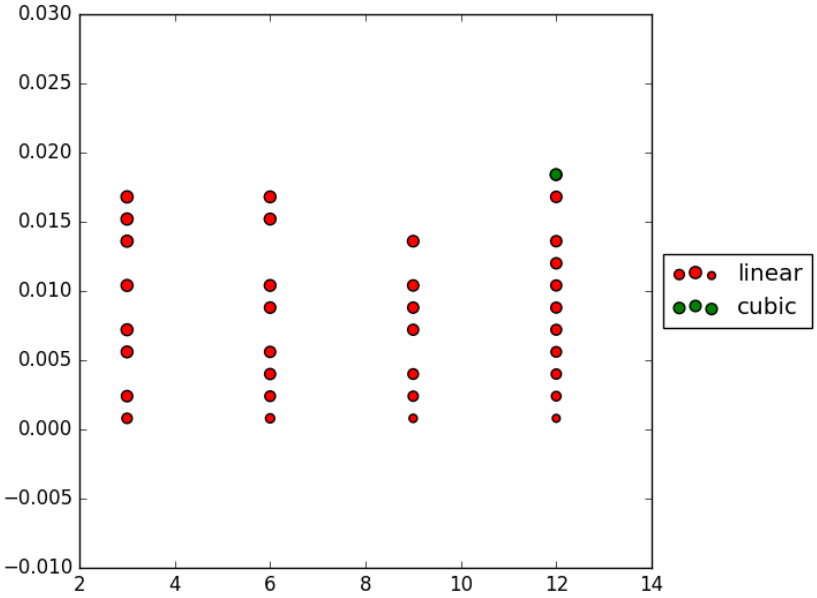

Figure 9: Variance – relationship between number of removed samples and parameter values

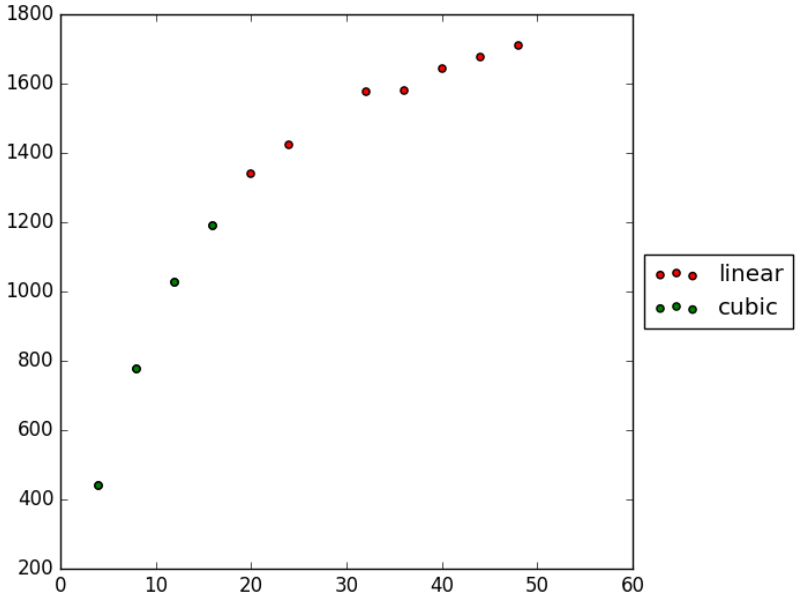

Figure 10: Derivative – relationship between number of removed samples and parameter value

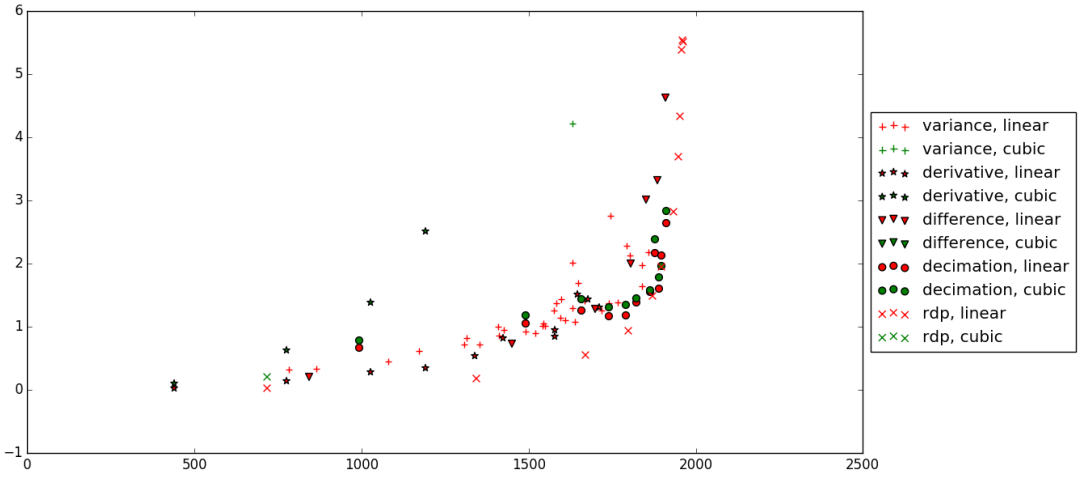

Figure 11: Comparison of all algoritms – relationship between square error and number of removed samples

References

[1] A. Gniewek. System czujników mierzący parametry lotu samolotu. Engineering project, Wrocław University of Technology, 2014.

[2] `Python 3` documentation. [online]. [access: 2016-01-25]. `https://docs.python.org/3/`.

[3] `NumPy` documentation. [online]. [access: 2016-01-25]. `http://docs.scipy.org/doc/numpy-1.10.0/reference/`.

[4] `SciPy` documentation. [online]. [access: 2016-01-25]. `http://docs.scipy.org/doc/numpy-1.10.0/reference/`.

[5] `matplotlib` documentation. [online]. [access: 2016-01-25]. `http://matplotlib.org/1.5.0/`.

[6] `rdp` documentation. [online]. [access: 2016-01-25]. `https://pypi.python.org/pypi/rdp/`.

[7] Crochiere, R.E. and Rabiner, L. Interpolation and decimation of digital signals – A tutorial review. *Proceedings of the IEEE*, 69(3):300–331, March 1981.

[8] Alan Saalfeld. Topologically consistent line simplification with the douglas-peucker algorithm. *Cartography and Geographic Information Science*, 26(1):7–18, 1999.