# Flexible Management of Large-Scale Integer Domains in CSPs

Nikolaos Pothitos  Panagiotis Stamatopoulos

Department of Informatics and Telecommunications,
University of Athens

# Introduction to CSPs

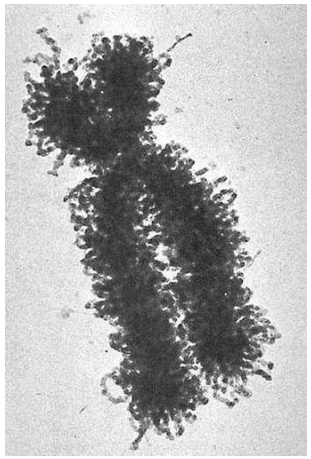
A *Constraint Satisfaction Problem* (CSP) is described by a triplet containing:

1. the *variables* of the problem, e.g. $X$, $Y$,
2. the *domains* of the variables, e.g. $D_X = \{0, 1\}$, $D_Y = \{1, 3, 4\}$ and
3. the *constraints* between the variables, e.g. $X \neq Y$.

Those problems are solved when we assign a value to each variable from its domain and the constraints are not violated.

# Management of Large Domains

In this work we focus on the manipulation of large finite integer domains (with millions of values).

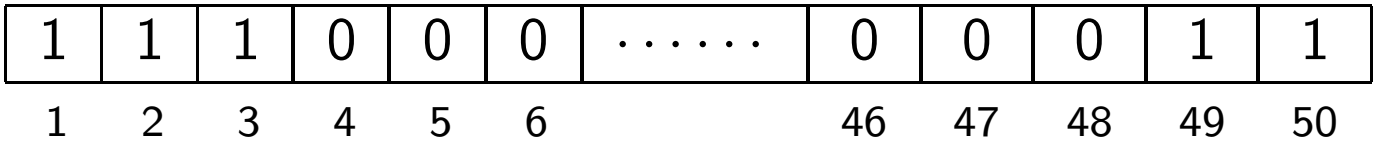


For example, in Bioinformatics the *position* of a nucleotide in a nucleotide chain can be depicted by a constrained *variable* whose *domain* is $\{1, 2, \ldots, 247200000\}$, also denoted as $[1..247200000]$.

# Using *Bit Vectors* to Store Large Domains

The wasted memory when we use bit vectors is shown.

E.g., let's store the domain [1..3 49..50] in a bit vector:




Too many zero bits are repeated in it.

# Using *Range Sequences* to Store Large Domains

Another option is to implement a binary search tree with the ranges that make up the domain.

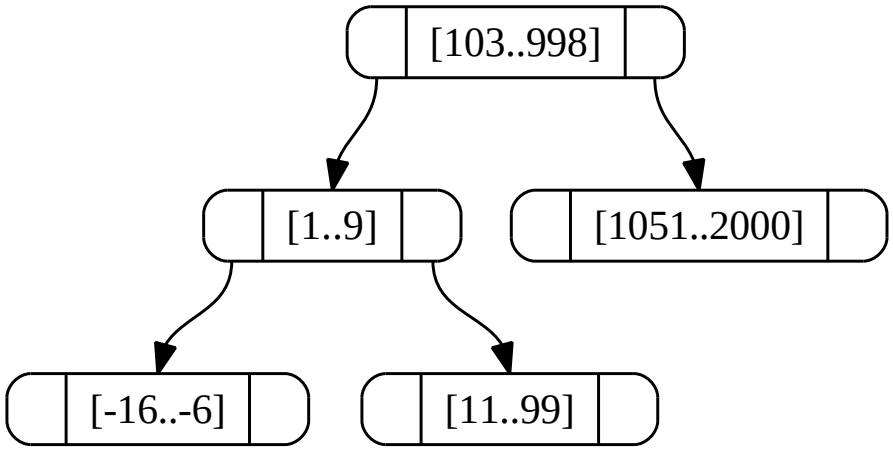



Figure: A tree with the ranges of the domain
$[-16..-6 \quad 1..9 \quad 11..99 \quad 103..998 \quad 1051..2000]$

Consider the time complexity when we have to *remove* a range from it, e.g. $[5..1500]$. Remember that the backtracking mechanism has to 'remember' every change made to the data structure.

# A New Way of Manipulating Large Domains

So we introduced a new way of storing a domain as a binary search tree containing *gaps*, that is proved to adapt better to large domains and general backtracking methodologies.

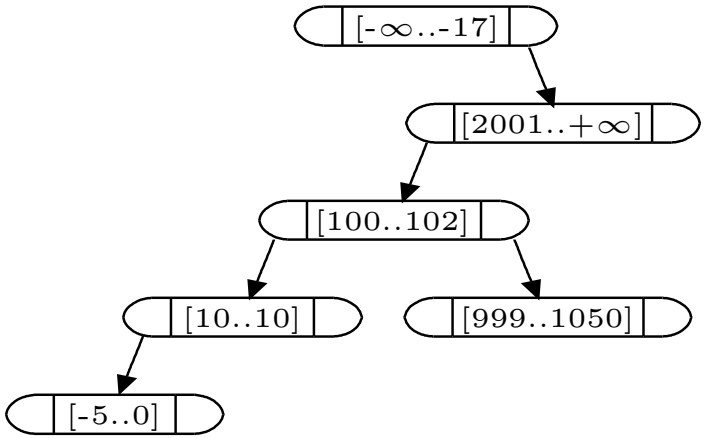
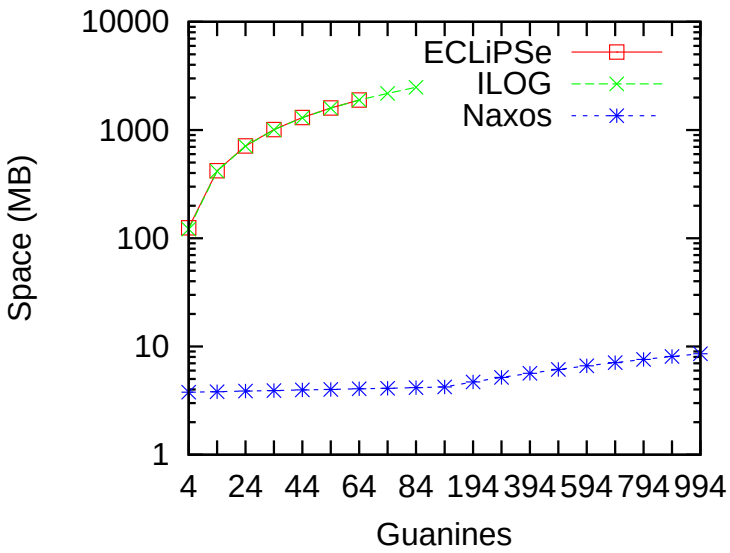
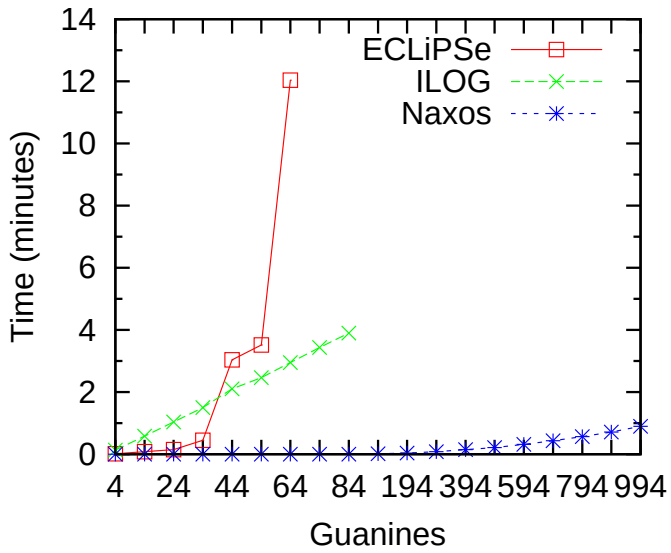



Figure: A tree with the gaps of the domain
$[-16..-6 \quad 1..9 \quad 11..99 \quad 103..998 \quad 1051..2000]$

# Empirical Results

Our data structures and algorithms are also supported by experimental results. . .




. . . that show that not only less space is allocated but also that the time is greatly reduced.

# Future Work

- Theoretical and experimental analysis of hybrid data structures.
- Variable size bit vectors may be integrated into binary tree nodes.
- Usage of various search and backtracking methodologies.
- More problems from Bioinformatics or other fields may be solved.

# References



Codognet, P., Diaz, D.:
Compiling constraints in `clp(FD)`.
*The Journal of Logic Programming* 27(3), pp. 185–226, 1996.


Schulte, C., Carlsson, M.:
Finite domain constraint programming systems.
In Rossi, F., van Beek, P., Walsh, T. (eds.) *Handbook of Constraint Programming*, chap. 14, pp. 495–526. Foundations of AI, Elsevier Science, Amsterdam 2006.


Watson, J., Baker, T., Bell, S., Gann, A., Levine, M., Losick, R.:
*Molecular Biology of the Gene*,
chap. 6. Pearson/Benjamin Cummings, 5$^{\text{th}}$ edn, 2004.


Zytnicki, M., Gaspin, C., Schiex, T.:
A new local consistency for weighted CSP dedicated to long domains.
In: *SAC'06: 2006 ACM Symposium on Applied Computing*, Dijon, pp. 394–398. ACM, New York, 2006.