Generating pairwise disjoint families through DNA computations

Mario J. Pérez–Jiménez, Fernando Sancho–Caparrini

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
E.T.S. Ingeniería Informática. Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {marper,fsancho}@us.es

1   Introduction

The computational solvability of **NP**-complete problems in practice has achieved a quantitative improvement with the born of DNA computing at the end of 1994. The arise of the first molecular models, in the beginning of 1995, and the universality of these models (i.e., with a computational power as a Turing Machine) allow, at least in a theoretical way, to deal with the solvability of any **NP**-complete problem through DNA computations in an efficient manner.

Solving a problem, X, in a molecular model consists in the design of a program, P, within this model, that can solve it in the following sense: for each instance, E, of the problem X, the execution of the program P over an initial tube encoding all possible solutions associated with E, produces a tube (or a set of tubes) encoding the solutions to the problem P respect to the input data E. Verifying (X, P) consists in proving that program P solves, in fact, the problem X; that is, to verify (X, P) we must follow two steps: (a) prove that every molecule of the output tube encodes a correct solution of the problem (*Soundness* of the program); (b) prove that every molecule of the input tube encoding a correct solution of the problem must be in the output tube (*Completeness* of the program).

The goal of this paper is the design, analysis and formal verification of a molecular program within the sticker model solving the problem of generating pairwise disjoint subfamilies of a finite collection of finite sets. The paper begins by briefly introducing a sticker based model for DNA computations. In section 3 we outline a problem, then a molecular program is designed in this model to solve the problem and finally we analyze the time complexity (number of molecular operations) and the space complexity (size of input tube and total number of tubes used along the execution) of the program. The main contribution of this work is presented in section 4, where formal verification of designed program is given. In section 5 we apply the above results to solve two well-known **NP**-complete numerical problems: the *Exact Cover* problem and the *Set Packing* problem [1].

References

1. Garey M.R.; Johnson D.S. *Computers and intractability*, W.H. Freeman and Company, New York, 1979.
2. Pérez-Jiménez, M.J.; Sancho-Caparrini, F. Solving Knapsack Problems in a Sticker Based Model, in N. Jonoska and N. Seeman (eds.), *DNA Computing, Lecture Notes in Computer Science*, **2340** (2002), 161-171.
3. Roweis, S.; Winfree, E.; Burgoyne, R.; Chelyapov, N.; Goodman, M; Rothemund, P; and Adleman, L. A Sticker–Based Model for DNA Computation, *Journal of Computational Biology*, **5** (4) (1998), 615–629.