A Family of Conceptual Problems in the Automated Design of Systems Self-Assembly

N. Krasnogor and S. Gustafson

**

Abstract. Several self-assembling systems have been proposed, design and implemented in the last few years and it is becoming more evident that nanotechnology in the 21st century will depend crucially on our ability to mastermind systems self-assembly. The automated design of such systems is thus a fundamental cornerstone of nanoscience. In this paper we introduce a family of (tunable) conceptual problems in automated design of systems self-assembly with the aim of focusing researchers attention on the difficulties that will need to be overcame in order to have a robust technology for the routine *automated design* of self-assembling systems.

1 Introduction

Self-assembly is a process that creates complex hierarchical structures through the statistical exploration of alternative configurations. These processes occur without external intervention. The specific system that is self-assembled (from a given set of components) is determined by the way the statistical exploration of conformations is performed. In turn, the exploration mechanisms are constrained by the individual components that undergo self-assembly and the conditions imposed upon them by their local environment. Usually these constraints are related to the type of interactions in which the components engage. In general, components are autonomous, have no pre-programmed *master* assembly plan, and can only interact with their local environment and other components. Reif in [9] said:

> "We need improved software for designing novel DNA tiles and tiling assemblies."

Although major advances in the design of systems that exhibit self-assembly properties have been reported in the literature (e.g. [11, 15]), much less has been said about the *automated* design of self-assembly. In [3] the author indeed tackles the problem of automated design of self-assembly for a very specific class of problems which are amenable to analytical solution. However, it is unrealistic to

** Automated Scheduling, Optimisation and Planning Research Group, School of Computer Science and IT, University of Nottingham, Nottingham, United Kingdom, {nxk,smg}@cs.nott.ac.uk. This work was partly funded by EPSRC/BBSRC grants GR/T07534/01, EP/D021847/1 and BB/C511764/1.

expect that each and every system which self-assemblies through the bottom-up interaction of component parts will present properties which make them agreeable to a hand-made design. That is, we anticipate that in the near future, as the number of applications for self-assembly (and their complexity) increases, a point will be reached where humans cannot design the set of components and their interactions. Instead, as in many other industrial settings, we will need to resort to computer aided automated design of components, interaction matrices and assembly skeletons. A discipline of general systems self-assembly will thus require not only the analysis of the computational and Kolmogorov complexities (e.g. [1, 10]) of the automated design of self-assembly but also suitable algorithmic tools to deal with computationally hard cases. That is, NP-hardness results have not, in the past, deterred the advance of other branches of science and engineering. On the contrary, NP-hardness results abound, are intrinsic to complex technology and they are routinely solved by an arsenal of modern optimisation algorithms ranging from integer and linear programming, Lagrangian relaxations to sophisticated metaheuristics like tabu search[4], simulated annealing[7] and memetic evolutionary algorithms[13].

2 (Tunable) Conceptual Problems in Automated Self-Assembly Design

In what follows we present three families of conceptual problems in automated self-assembly design - related to some well known problems - in the hope that these "self-assembly computational challenges" will initiate research on the application of sophisticated algorithms for the automated design of systems self-assembly[1]

2.1 Family 1:

This family of problems is based on the well-known NK-Landscapes model[6], albeit, in its inverted version. An NK-landscape instance consists of two integer n and k representing the total number of genes n and the number of neighboring genes a gene, let say i, is epistatically related to. The values k can take are $0 \leq k \leq n-1$. Besides n and k, a $n \times 2^{k+1}$ matrix E with elements sampled randomly from the (usually) uniform distribution $U(0,1)$ is also required to completely define an instance. A solution to an NK-landscape problem instance is represented as a binary string S with length n. The fitness of S is given by $\frac{1}{n} * \sum_{i=1}^{i=n} f_i(S_i, S_{i_1}, \ldots, S_{i_k})$ where $f_i(\cdot)$ is an entry in E, S_i the value of string S at position i and S_{i_j} is the value of string S at the $j-th$ neighbor of bit i. The neighborhood structure (which is not necessarily nearest neighbors), j of bit i are part of the input as well. In the general case this problem is NP-Hard [14].

Problem Π_1: The NK-Landscape Inverse Topology Given a target binary string S and an epistatic matrix E find a neighborhood epistatic topology

[1] A longer version of this paper can be found at http://www.cs.nott.ac.uk/~nxk.

of order k such that S is the string with optimal fitness under the resulting NK-Landscape.

In this self-assembly design problem we are given S as the target system into which components must self-assemble and the interaction matrix E which gives the $k+1$-wise interaction contributions. We are asked to design the *topology* of valid $k+1$-wise interactions such that when they occur, S is the system with maximum fitness. Please note that in Π_1 the assembled structure is a linear order and components can be of only two types, 0 and 1, which when assembled into specific order contribute to the fitness of the resulting structure S based on both the designed topology and the epistatic matrix E.

Problem Π_2: The NK-Landscape Inverse Epistatic Interaction Given a target binary string S and an order k neighborhood topology find the epistatic matrix E with minimum number of non-zero entries in $0 \leq m_{i,j} \leq 1$ such that S is the optimal fitness string in the resulting NK-Landscape problem.

In Π_2 we are given the topology and the target structure and we are required to design the energetic interactions between pairs of components in such a way that S is the maximum fitness self-assembled system.

Problems Π_1 and Π_2 can be generalized to alphabets with cardinality bigger than two. For example, instead of requiring the string S to be a binary string it could be required to be a string in a 20-letters alphabet (e.g. as proteins amino acids alphabet) or a 4-letters alphabet (e.g. as genetic bases).

2.2 Family 2:

The second family of problems is concerned with inverse protein folding problems based on two popular simplified models of folding: the HP model[2] and Functional Model Proteins[5] in two and three dimensional lattices.

Problem Π_3: The Inverse HP Protein Structure Prediction Given the specification of a two/three dimensional native structure S embedded on a lattice of geometry G (e.g. square, honeycomb, face centered cubic, etc) and the pairwise energy interactions typical of the HP model, find the sequence $s \in \{H, P\}^*$ that has S as its native structure.

Problem Π_4: The Inverse Functional Model Protein Structure Prediction Given the specification of a two/three dimensional native structure S embedded on a lattice of geometry G (e.g. square, honeycomb, face centered cubic, etc) and the pairwise energy interactions typical of the Functional Protein model, find the sequence $s \in \{H, P\}^*$ that has S as its unique native structure.

Inverse protein folding (also called Protein Design) has been shown to be NP-Hard (e.g. [8]). Although this problem is a very active research area (see for example [12] and references therein), little attention has been put on its relation to the automated design of self-assembling systems. The fundamental question that problems Π_3 and Π_4 pose is whether the (partial) success of the various artificial intelligence approaches which were applied to the protein structure prediction problem could be replicated in the inverse problems, and if not, which new algorithmic technologies are required to tackle them.

2.3 Family 3:

A bi-labeled graph $G = (V, E, L_V, L_E)$, has labels $L_V : V \mapsto \Sigma_V$, and $L_E : E \mapsto \Sigma_E$, $\Sigma_V \cap \Sigma_E = \emptyset$, where V is a set of vertices, E is a set of edges such that $e \in E, e = (v_1, v_2), v_1, v_2 \in V$, and Σ_V, Σ_E are alphabets for vertices and edges. Also, when a vertex is given it can be used as a *template* for vertices families. That is, if $v \in V$ then v is a family of vertices with certain properties and we can use/build as many instances of the family as we want. In $\Pi_{5,6,7,8}$ bellow, G' is a minimum energy ensemble, that is, we defined and energy function $\mathcal{E}(G') = \sum_{i,j \in V', i \neq j} E'(i,j) * M'(i,j)$ which measures the native state energy of a graph G' as the sum of the interactions M'(i,j) of every pair of vertices i, j which share and edge (represented by the edge set E' in incidence matrix format). The next four conceptual problems also beg the question of which computational search strategy must be used to automatically design the requested entities?.

Problem Π_5: The Structure Problem Given a target graph G and the energy formulation described, design the set of vertices V' and interaction matrix $M' : V' \times V' \mapsto \mathcal{R}$ such that the unique *unlabeled* graph G' is formed, with $G' \sim G$, $\sim$ is an isomorphism between the two graphs and V' is such that every vertex is constrained to have a in/out degree of at most k.

Problem Π_6: The Structure-Content Problem Given a target graph G and the energy formulation described, design the set of vertices V', the interaction matrix $M' : V' \times V' \mapsto \mathcal{R}$ and a labeling function $L' : V' \mapsto \Sigma_V$ such that the unique *labeled* graph G' is formed, with $G' \sim G$, $\sim$ is an isomorphism between the two graphs which also respects the vertices labels and the V' is such that every vertex is constrained to have a in/out degree of at most k.

Problem Π_7: The Structure-Property Problem Given a target graph G and the energy formulation described, design the set of vertices V', the interaction matrix $M' : V' \times V' \mapsto \mathcal{R}$ and an labeling function $L'' : E' \mapsto \Sigma_E$ such that the unique *labeled* graph G' is formed, with $G' \sim G$, $\sim$ is an isomorphism between the two graphs which also respects the edges labels and V' is such that every vertex is constrained to have a in/out degree of at most k.

Problem Π_8: The Structure-Function Problem Given a target graph G and the energy formulation described, design the set of vertices V', the interaction matrix $M' : V' \times V' \mapsto \mathcal{R}$ and labeling functions $L' : V' \mapsto \Sigma_V$, $L'' : E' \mapsto \Sigma_E$ such that the unique *bi-labeled* graph G' is formed, with $G' \sim G$, $\sim$ is an isomorphism between the two graphs which also respects both edges and vertices labels and V' is such that every vertex is constrained to have a in/out degree of at most k.

3 Conclusions

In this paper we introduced three families of problems in the automated design of systems self-assembly. Any new algorithmic technique for the automated design of systems self-assembly should be tested on a set of well defined, highly idealized, complex problems as the ones we introduced here. Our future work

will concentrate in four fronts: (1) we will formally define the complexity of these eight problems, (2) we will make publicly available source code to systematically generate tunable instances of these problems, (3) we will investigate the applicability of modern search techniques to these challenging problems and finally (4) we have set up a public web-page (www.cs.nott.ac.uk/~gzt/CASA/) were researchers working in self-assembly could share with the research community their "self-assembly automated design grand-challenges".

References

1. L. Adleman, Q. Cheng, A. Goel, M. Huang, D. Kempe, P. Moisset de Espanes, and P.W.K. Rothemund. Combinatorial optimization problems in self-assembly. In *Proceedings of the Annual ACM Symposium on Theory of Computing(STOC)*. ACM Press, 2002.
2. Ken A. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24:1501, 1985.
3. E.Klavins. Automatically synthesized controllers for distributed assembly: Partial correctness. In S.Butenko, R.Murphey, and P.M.Pardalos, editors, *Cooperative Control: Models, Applications and Algorithms*. Kluwer, 2002.
4. F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.
5. J.D. Hirst. The evolutionary landscape of functional model proteins. *Protein Engineering*, 12:721–726, 1999.
6. S.A. Kauffman. *The Origins of Order, Self Organization and Selection in Evolution*. Oxford University Press, 1993.
7. S. Kirkpatrick, C.D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 no 4598:671–680, 1983.
8. N.A. Pierce and E. Winfree. Protein design in np-hard. *Protein Engineering*, 15(10):779–782, 2002.
9. J.H. Reif. Dna lattices: A method for molecular-scale patterning and computation. *Computing in Science and Engineering Magazine, IEEE Computer Society*, 4(1):32–41, 2002.
10. P. Rothemund and E. Winfree. The program-size complexity of self-assembled squares. In *Proceedings of STOC*, 2000.
11. W.K. Rothemund. Using lateral capillary forces to compute by self-assembly. *Proceedings of the National Academy of Science, USA*, 97(3):984–989, 2000.
12. S.Sun, R.Brem, H.Chan ans S.Hue, and K.A.Dill. Designing amino acid sequences to fold with good hydrophobic cores. *Protein Engineering*, 9(1), 1996.
13. N. Krasnogor W.E. Hart and J.E. Smith. *Recent Advances in Memetic Algorithms*. Studies in Fuzziness and Soft Computing Series - Springer, 2004.
14. E.D. Weinberger and A. Fassberg. Np completness of kauffman's n-k model, a tuneably rugged fitness landscape. In *Santa Fe Institute Technical Reports*, 1996.
15. G.M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295:2418–2421, 2002.