Combinatorial Problems on Strings with Applications to Protein Folding

Alantha Newman[1] and Matthias Ruhl[2]

[1] MIT Laboratory for Computer Science
Cambridge, MA 02139
alantha@theory.lcs.mit.edu
[2] IBM Almaden Research Center
San Jose, CA 95120
ruhl@almaden.ibm.com

Abstract. We consider the problem of protein folding in the HP model on the 3D square lattice. This problem is combinatorially equivalent to folding a string of 0's and 1's so that the string forms a self-avoiding walk on the lattice and the number of adjacent pairs of 1's is maximized. The previously best-known approximation algorithm for this problem has a guarantee of $\frac{3}{8} = .375$ [HI95]. In this paper, we first present a new $\frac{3}{8}$-approximation algorithm for the 3D folding problem that improves on the absolute approximation guarantee of the previous algorithm. We then show a connection between the 3D folding problem and a basic combinatorial problem on binary strings, which may be of independent interest. Given a binary string in $\{a, b\}^*$, we want to find a long subsequence of the string in which every sequence of consecutive a's is followed by at least as many consecutive b's. We show a non-trivial lower-bound on the existence of such subsequences. Using this result, we obtain an algorithm with a slightly improved approximation ratio of at least .37501 for the 3D folding problem. All of our algorithms run in linear time.

1 Introduction

We consider the problem of protein folding in the HP model on the three-dimensional (3D) square lattice. This optimization problem is combinatorially equivalent to folding a string of 0's and 1's, i.e. placing adjacent elements of the string on adjacent lattice points, so that the string forms a self-avoiding walk on the lattice and the number of adjacent pairs of 1's is maximized. Figure 1 shows an example of a 3D folding of a binary string.

Background. The widely-studied HP model was introduced by Dill [Dil85, Dil90]. A protein is a chain of amino acid residues. In the HP model, each amino acid residue is classified as an H (hydrophobic or non-polar) or a P (hydrophilic or polar). An optimal configuration for a string of amino acids in this model is one that has the lowest energy, which is achieved when the number of H-H contacts (i.e. pairs of H's that are adjacent in the folding but not in the string) is maximized. The *protein folding* problem in the hydrophobic-hydrophilic (HP)

M. Farach-Colton (Ed.): LATIN 2004, LNCS 2976, pp. 369–378, 2004.

model on the 3D square lattice is combinatorially equivalent to the problem we just described: we are given a string of P's and H's (instead of 0's and 1's) and we wish to maximize the number of adjacent pairs of H's (instead of 1's). An informative discussion on the HP model and its applicability to protein folding is given by Hart and Istrail [HI95].

Related Work. Berger and Leighton proved that this problem is NP-hard [BL98]. On the positive side, Hart and Istrail gave a simple algorithm with an approximation guarantee of $\frac{3}{8}OPT - \Theta(\sqrt{OPT})$ [HI95]. Folding in the HP model has also been studied for the 2D square lattice. This variant is also NP-hard [CGP+98]. Hart and Istrail gave a $\frac{1}{4}$-approximation algorithm for this problem [HI95], which was recently improved to a $\frac{1}{3}$-approximation algorithm [New02].

Our Contribution. Improving on the approximation guarantee of $\frac{3}{8}$ for the 3D folding problem has been an open problem for almost a decade. In this paper, we first present a new 3D folding algorithm (Section 2.1). Our algorithm produces a folding with $\frac{3}{8}OPT - \Theta(1)$ contacts, improving the absolute approximation guarantee. We then show that if the input string is of a certain special form, we can modify our algorithm to yield $\frac{3}{4}OPT - O(\delta(S))$ contacts, where $\delta(S)$ is the number of transitions in the input string S from sequences of 1's in odd positions in the string to sequences of 1's in even positions. This is described in Section 2.2.

In Section 3, we reduce the general 3D folding problem to the special case above, yielding a folding algorithm producing $.439 \cdot OPT - O(\delta(S))$ contacts. This reduction is based on a simple combinatorial problem for strings, which may be of independent interest.

We call a binary string from $\{a, b\}^*$ *block-monotone* if every maximal sequence of consecutive a's is immediately followed by a block of at least as many b's. Suppose we are given a binary string with the following property: every suffix of the string (i.e. every sequence of consecutive elements that ends with the last element of the string) contains at least as many b's as a's. What is the longest block-monotone subsequence of the string? It is easy to see that we can find a block-monotone subsequence with length at least half the length of the string by removing all the a's. In Section 3.1, we show that there always is a block-monotone subsequence containing at least a $(2 - \sqrt{2}) \approx .5857$ fraction of the string's elements.

Finally, we combine our folding algorithm with a simple, case-based algorithm that achieves $.375 \cdot OPT + \Omega(\delta(S))$ contacts, which is described in the full version of this paper. We thereby remove the dependence on $\delta(S)$ in the approximation guarantee and obtain an algorithm with a slightly improved approximation guarantee of .37501 for the 3D folding problem. Due to space restrictions, all proofs are omitted and can be found in the full version of this paper.