COMBINATORIAL ALGORITHMS FOR BIPOLE SELF-ASSEMBLY ON LATTICES WITH APPLICATIONS TO THE LIPID BILAYER

KSHITIJ LAURIA AND SORIN ISTRAIL

ABSTRACT. Phospholipid bilayer self-assembly is modeled as a discrete optimization problem on space-filling lattices. The bipole packing problem is defined and proved NP-Complete on arbitrary graphs by a reduction from Densest k-Subgraph. Bipole packing is proved to be a combination of Densest k-Subgraph and bipartite matching. It is conjectured that the optimal configuration is a biplane in the 3D cubic and FCC lattices. This is supported by proof that the biplane configuration is a $1 - O(\frac{1}{\sqrt{n}})$ approximation to the optimal in the cubic lattice. A stronger approximation ratio of $1 - O(\frac{1}{n^2})$ is suggested by computational evidence. Integer linear programs are formulated and used to compute optimal configurations for $n \sim 150$ in the 3D cubic lattice, verifying the biplane conjecture for these sizes. A branch-and-bound algorithm is used to verify these results. A statistical mechanics model is formulated to study the biplane's thermal stability. Monte Carlo simulations are used to estimate the partition function to provide evidence for the stability of the biplane configuration in a low-temperature region. A phase transition is observed in the stability of the biplane configuration. The "shared-heads" relaxation is introduced and empirically shown to have the same optimal as bipole packing in this lattice. A related, previously open problem (Bipole Configuration Labeling Problem) is proved NP-Complete.

CONTENTS

1. Introduction

Abstracted lattice models provide mathematically precise ways to formulate questions about the structure of biomolecules [1]. Simplified descriptions of the interactions between atoms or other subunits of biomolecules have two major benefits. First, the mathematics becomes simpler. Complex force-fields with many parameters are replaced with graph-theoretic criteria, allowing the use of combinatorial arguments in finding and proving optima[4]. Provably optimal solutions can be found for small instances using combinatorial optimization algorithms. Even so, the mathematics remains hard. The paucity of results in this area is related to the great difficulty of solving low-dimensional packing problems[6], a connection that is explored in this paper and made explicit.

The second benefit of simplified models is that they are conceptually simple enough that we may pry them apart and investigate how each element in the description affects resulting structure. The freedom to experiment with different constraints and subunits allows us to tackle questions like: Why do phospholipids self-assemble into bilayers? The broad goals of such study are to discover some fundamental "vocabulary" of subunits and (local) interactions by which clusters of biomolecules self-assemble into useful structures; and to build intuitive understanding of which forces are responsible for which aspects of structure. This allows us to tackle questions like: Which structures are accessible to self-assembly? Mathematically, such analysis usually involves the study of restrictions and relaxations of the underlying optimization problem, an example of which is presented in this paper.

The model structure studied in this paper is the phospholipid bilayer. A phospholipid generally consists of a hydrophilic head and hydrophobic tails. In aqueous media phospholipids spontaneously assemble into a double layer with hydrophilic heads facing outside and hydrophobic tails sandwiched on the inside. Interactions with the polar medium stabilize the hydrophilic outer layers, while mutual hydrophobic interactions stabilize the inner. Chemical intuition can guide the construction of simplified models: hydrophobic interactions are the major driving force for the formation of lipid bilayers [7], so our model

contains only attractive hydrophobic terms in its objective function. A phospholipid is represented by a double-ended entity with a hydrophobic tail a fixed distance away from a hydrophilic head, a unit we will refer to as a bipole. Bipoles may only occupy discrete nodes of a space-filling lattice, with one end on each of two adjacent lattice points.

Analysis of the bipole packing model suggests that the optimal configuration is indeed a bilayer with tails sandwiched between heads, although proof remains elusive. A key feature of the bipole packing model is its relationship to two classic problems in computer science, Densest k-subgraph[5] and bipartite matching. These problems admit pithy expression as integer linear programs, a property that proves immensely useful for computational experimentation.

This paper begins with the development of an integer linear program for the bipole packing model. The Densest k-subgraph problem is revisited several times, representing the core computational hardness of this problem. The restriction to lattice points has some immediate implications for optimal configurations. Finally, theoretical and computational evidence supporting the biplane conjecture is presented.

In conclusion, discrete models allow us to turn problems in statistical mechanics into problems in combinatorial optimization. Whether this program of research will be fruitful is for time to tell.

Notation. The letters G and H will usually refer to arbitrary graphs. A subgraph H of G is a said to be *full* or *induced* if it has every edge that G does over the same vertex set. A set of vertices $S \subseteq V(G)$ defines an induced subgraph of G denoted by $G[S]$.

Notation	Meaning
$\|S\|$	number of items in set S
$E(G)$	edge set of graph G
$V(G)$	vertex set of graph G
$E_G(v)$	edges in $E(G)$ incident to vertex v
$E_G(S,T)$	edges in $E(G)$ joining a vertex in S to a vertex in T
$N_G(v)$	neighbors of v in G
$N_G(S)$	$\bigcup_{v\in S} N_G(v) - S$, neighborhood of S in G
$G[S]$	subgraph of G induced by S
(S,T,E)	bipartite graph $S+T$ and edges E straddling S and T
$[t,h]$	a bipole with tail t and head h (note order)

2. The Densest k-subgraph Problem

DkS is a discrete generalization of sphere-packing problems studied since before Guass [10]. It is also the natural generalization of the Max-Clique problem. While the problem is defined (and NP-Hard) for arbitrary graphs, we will only consider DkS and its variants on (finite chunks of) space-filling lattices.

Definition Given an integer k and a graph G of n vertices where $k < n$, the Densest k-subgraph problem (DkS) asks for a subgraph H of G such that H has exactly k vertices

and no other subgraph with k vertices has a greater number of edges than H. Note that H must be an induced (or full) subgraph if its edge set has maximum cardinality.

If S is the vertex set of the selected subgraph H, then $H = G[S]$ and the number of edges in H can be written as $|E(G[S])$. In pseudo-standard form, the maximization problem can be written as

$$\begin{aligned} \max \quad & |E(G[S])| \\ \text{s.t.} \quad & |S| = k \\ & S \subseteq V(G). \end{aligned}$$

Decision-problem variant: Given (G, k, E), return True if there exists a subgraph H of G with k vertices and E edges, and no other subgraph with k vertices has more than E edges.

The decision problem is NP-complete: we can solve an instance of Clique with graph G and clique-size c as an instance of DkS with graph G, subgraph vertex-set size c and edge-set size $\binom{c}{2}$.

Suppose H_1 and H_2 are two disjoint induced subgraphs of G. If we let $E(H)$ denote the set of edges of subgraph H, and $E_G(V(H_1), V(H_2)) = \{(u, v) : u \in V(H_1), v \in V(H_2), (u, v) \in E(G)\}$, the edges joining a vertex in H_1 to a vertex in H_2 (see the section "DkS + matching = BPP" for a more precise definition) then

$$|E(H_1 \cup H_2)| \leq |E(H_1)| + |E(H_2)| + |E_G(V(H_1), V(H_2))|,$$

that is to say, any edge in the union of H_1 and H_2 was either in H_1, in H_2, or the set of edges that cross between H_1 and H_2. By choosing families of subgraphs appropriately we can obtain upper bounds on the optimum using this inequality. For example, in an n-dimensional grid we can divide any set of lattice points into two across a plane. If k_1 and k_2 are the numbers of points in either subset then at most $\min(k_1, k_2)$ edges may cross the dividing plane. This leads to the well-known[3] maximin recurrence

$$f(k) = \max_{k=k_1+k_2} (f(k_1) + f(k_2) + min(k_1, k_2))$$

where the maximization occurs over all partitions of k.

3. Integer Linear Programs for DkS and matching

3.1. **ILP for DkS.** Let $x_1, x_2, ...x_n$ be binary variables representing the nodes of a graph G. x_i is 1 if node i is in the selected subgraph. An edge is in the selected subgraph iff both its endpoints are selected. A first attempt to formulate an integer linear program for DkS might be

$$\begin{aligned} \max \quad & \sum_{(u,v) \in E(G)} x_u x_v \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\ & x_i \in \{0, 1\}. \end{aligned}$$

While this formulation captures DkS, it is unfortunately not linear: the objective function consists of terms with variables multiplied with each other. However, since each term is the boolean AND of two variables, we can introduce auxiliary binary variables $y_1, y_2, ...y_m$ where $m = |E(G)|$, one for each edge of G. The variable y_i is 1 when the corresponding edge is in the selected subgraph. This constraint is expressed by requiring that the value of y_i be less than the values of both x_u and x_v, u and v are the vertices that edge i connects. Thus, y_i can be 1 only if both x_u and x_v are. Note that since each y_i appears with a positive coefficient in the objective function and this is a maximization problem, we do not need constraints to make sure $y_i \neq 0$ when $x_u = x_v = 1$. This is equivalent to the statement that any optimal subgraph must be an induced subgraph. The final integer linear program for DkS is:

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} y_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\ & y_i \leq x_u, u \in \text{edge } i \\ & x_i \in \{0,1\} \\ & y_i \in \{0,1\}. \end{aligned}$$

3.2. **Bipartite Matching.** Bipartite matching is a classic problem in computer science. Given a bipartite graph $G = G_1 \cup G_2$ we seek the largest set of edges $M \subseteq E(G)$ (called a *matching*) such that any vertex in G is incident to at most one edge in M. Intuitively we wish to pair vertices in G_1 with vertices in G_2, maximizing the number of matched pairs. An edge between two vertices indicates that they form an acceptable pair, and once a vertex is paired it may not be part of any other pair. Let $m = |E(G)|$ and $y_1, y_2, ...y_m$ be binary variables, one for each edge in G, where $y_i = 1$ iff edge i is in the matching. The sum of all y_i is the number of selected edges: this is our objective function. For any vertex v, let $E_G(v) = \{(u_1, u_2) \in E(G) : u_1 = v \text{ or } u_2 = v\}$, the set of edges incident to v. Since at most one of these edges can be in the matching, the sum of the y_i's corresponding to edges in this set must be at most one.

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} y_i \\ \text{s.t.} \quad & \sum_{i \in E_G(v)} y_i \leq 1 \text{ for every } v \in G \\ & y_i \in \{0,1\}. \end{aligned}$$

A matching is called *perfect* if every vertex has been matched, or more generally if $|M| = \min(|G_1|, |G_2|)$. Let S be some subset of vertices in G_1, and let $N_G(S)$ denote the set of vertices in G that are incident to some vertex in S (since $G = G_1 \cup G_2$ is bipartite, $N_G(S) \subseteq G_2$ whenever $S \subseteq G_1$). Then Hall's Marriage Theorem states that there exists a perfect matching in G iff for every $S \subseteq G_1$, $|S| \leq |N_G(S)|$.

Matching can be solved in polynomial time, for example by reducing it to a max-flow problem. This is mirrored by a property of the integer linear program: its constraint matrix is totally unimodular (by expressing $y_i \in \{0, 1\}$ as $0 \leq y_i \leq 1$).

4. The Bipole Packing Problem

In this section we define the Bipole Packing Problem and consider its properties on arbitrary graphs. In the following sections we study its restriction to square and cubic lattices. This model was introduced in [9], where the *biplane problem* was introduced. In [8] it was described as the *Bi-Sphere Packing Problem* and solved in the 2D square lattice.

4.1. **Setup.** We are given k bipoles to pack on a graph G. Bipoles have a head and a tail, and each end occupies one vertex in G. When two tails are adjacent in the packing, we count the edge between them as a contact. We wish to maximize the number of such tail-tail contacts. Heads don't create contacts: in this problem they act only to constrain the tails. Every tail must have its head on an adjacent vertex, and no other tail nor head can occupy that vertex.

Accommodating these heads may or may not be a severe restraint. Below we will see an example, K_{2k}, where the need to place a head next to each tail is trivially satisfied. In general this is not true and the optimal solution for DkS will not be a feasible solution for BPP.

Definition An instance of the Bipole Packing Problem (BPP) consists of a graph G and an integer $k < \frac{|V(G)|}{2}$. A *bipole* is an ordered pair $[u, v]$ of two vertices such that there is an edge between them: $(u, v) \in E(G)$. A *feasible solution* consists of k bipoles such that no vertex appears more than once among all bipoles. Equivalently, the complete set of heads and tails has $2k$ elements. The number of contacts in a feasible solution K is the number of edges in the subgraph induced by just the set of tails, $|E(G[\{u : [u, v] \in K\}])|$. An *optimal solution* is one that maximizes the number of contacts.

We will mostly refer to candidate solutions to the BPP by their tail sets. There may be several feasible solutions that have the same tail set, but we will typically be more interested in the fact that a tail set is feasible than in what those solutions are. Note that the number of contacts in a feasible solution depends only on its tail set.

For example, suppose G is the complete graph over $2k$ vertices, K_{2k}. Then any set of k vertices is a feasible tail set, since there will be $2k - k = k$ vertices left over to be heads, and any vertex not already occupied may serve as head to any other. Any set of k vertices is also optimal, since each set of k vertices induces a subgraph with exactly $\binom{k}{2}$ edges. Corresponding to any choice of k tails there are $k!$ ways to choose heads for each tail. Since there are $\binom{2k}{k}$ ways to choose a tail set, there are $\binom{2k}{k}k!$ distinct optimal solutions, each with $\binom{k}{2}$ contacts. There is only solution up to isomorphism, since any permutation of vertex labels is a graph isomorphism in K_{2k}.

4.2. **DkS + matching = BPP.** We can imagine packing bipoles as a two-step game: first put down all the tails, then find a place adjacent to each tail to put its head. The set of vertices S selected to be tails defines the number of contacts in a configuration, because

each contact is an edge joining two vertices in this set. This can be expressed succinctly as $|E(G[S])|$, the number of edges in the subgraph induced by S in G. In pseudo-standard form, this strategy can be expressed as finding a vertex set S that optimizes the following:

$$\begin{array}{rl} \max & |E(G[S])| \\ \text{s.t.} & |S| = k \\ & S \subseteq V(G) \\ & \text{there's head-space for every tail in } S. \end{array}$$

Note the similarity to the pseudo-standard form for DkS. The requirement of allocating a vertex adjacent to each tail to serve as its head is the one additional constraint. Once we have selected a set of vertices S to be tails, only vertices adjacent to vertices in S can serve as heads. In some sense once the tails have been placed we can forget about the rest of the graph and focus on a small part near S to determine whether there is place for a head next to each tail. This motivates the following definition:

Definition Let $N_G(v)$ denote the neighbors of vertex v in G. Then the *neighborhood of S in G*, denoted by $N_G(S)$, is the set of vertices in $V(G) - S$ adjacent to some vertex in S by an edge in G, given by $\bigcup_{v \in S} N_G(v) - S$.

We will also find it convenient to talk about edges that join vertices in two different sets.

Definition Let G be a graph and let S and T be two disjoint subsets of the vertex set of G. Then the *edges across S and T*, denoted by $E_G(S,T)$, are those edges that connect a vertex in S to a vertex in T, given by $\{(u,v) \in E(G) : u \in S, v \in T \text{ or } u \in T, v \in S\}$.

With these definitions we can formulate the head-space constraint in a precise way. The set of vertices S, its neighborhood $N_G(S)$ and the edges across S and $N_G(S)$ together define a bipartite graph $(S, N_G(S), E_G(S, N_G(S)))$. Allocating a head for a given tail is equivalent to matching a vertex in S to a vertex in $N_G(S)$ connected to it by an edge. In other words, the problem of allocating a head for each tail is equivalent to the problem of finding a perfect matching in this bipartite graph. Now we can refine our formulation to:

$$\begin{array}{rl} \max & |E(G[S])| \\ \text{s.t.} & |S| = k \\ & S \subseteq V(G) \\ & (S, N_G(S), E_G(S, N_G(S))) \text{ admits a perfect matching} \end{array}$$

Note the conceptual structure of this formulation for BPP: *first* a set S of k tails is selected, which defines a bipartite graph $(S, N_G(S), E_G(S, N_G(S)))$, *then* we attempt to find a perfect matching in this bipartite graph. Suppose we take some subset $T \subseteq S$ and construct its corresponding bipartite graph, only to find that there is no perfect matching. What can we then say about the existence of a perfect matching in the bipartite graph corresponding to the full set S?

Proposition 4.1. *Let G be a graph, $S \subseteq V(G)$ and $T \subseteq S$. If the bipartite graph $(T, N_G(T), E_G(T, N_G(T)))$ admits no perfect matching then neither does the bipartite graph $(S, N_G(S), E_G(S, N_G(S)))$.*

Proof. Let B_T and B_S be the two bipartite graphs in the statement of the proposition. By Hall's Marriage Theorem we conclude that there must be some subset $U \subseteq T$ such that $|N_{B_T}(U)| < |U|$. Can $N_{B_S}(U)$, the neighborhood of this offending set in the larger bipartite graph B_S, be any larger? No new vertices become adjacent to vertices already in T by the inclusion of additional vertices to the tail set, while some vertices may leave the neighborhood because they become members of the tail set. Thus, the neighborhood of this offending set in the larger bipartite graph can only be smaller, and so by another application of Hall's Theorem in the other direction there can be no perfect matching in B_S. □

This suggests a branch-and-bound algorithm for finding the optimal feasible solution to BPP. The algorithm maintains, at any stage, a candidate subset U with k or fewer elements. Branching is done upon vertices to add to U. If the bipartite graph $(U, N_G(U), E_G(U, N_G(U)))$ admits no perfect matching (a fact that can be checked in polynomial time) then that branch of the tree can be pruned. Moreover, since the objective function depends only on the tail vertices, it may be possible to derive bounds on the maximum value of the objective of any configuration containing U as a subset. The particular structure of G can inform such bounds, especially when the order in which branches are taken is chosen carefully. We will see such a branch-and-bound algorithm in the section on computational evidence, although the integer program formulation performs better in practice for the particular graphs we are interested in.

4.3. **An integer linear program for BPP.** We follow the subsection above and formulate an integer program for BPP by combining those for DkS and bipartite matching. Recall that there are m edges and n vertices in graph G, and we want to place k bipoles such that we maximize the number of edges between tails. Let us forget about bipoles and place only tails, so we are left with DkS. x_i's correspond to vertices, and y_j's correspond to edges.The variable x_i is set to 1 if there is a tail placed at vertex v, and y_j is set to 1 if both vertices connected by edge j are tails. We want to maximize the number of such edges, so we maximize the sum of y_j's over all edges. We also only want k tails, so the sum of x_i's over all vertices must be 1.

$$\begin{aligned} \max \quad & \sum_{j=1}^{m} y_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\ & y_j \leq x_u, u \in \text{edge } j \\ & x_i \in \{0,1\} \\ & y_j \in \{0,1\}. \end{aligned}$$

Now we want to attach the body and head to each tail. That is, every vertex where a tail is placed must be matched to its own non-tail vertex. The edge that connects them is called a *bipole edge*, and we introduce a z_j associated with each edge j that denotes bipole edges ($z_j = 1$ iff edge j is a bipole edge).

How do we characterize (constrain) bipole edges? There must be one adjacent to every tail vertex: if $x_i = 1$, at least one of the z_j's for the edges of x_i must be 1. This is equivalent to requiring that the sum of z_j's for edges of vertex i is greater than or equal to x_i, because all of the variables here are binary.

Bipoles occupy space: as in the ILP for bipartite matching, only one of the edges around a vertex can be a bipole edge. Thus, the sum of z_j's for edges of vertex i must be less than or equal to 1.

Note: $E_G(v)$ denotes the set of edges of vertex v in graph G.

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} y_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\ & y_j \leq x_u, u \in \text{edge } j, \text{ for every edge } j \\ & x_i, y_j, e_j \in \{0, 1\} \\ & x_i \leq \sum_{e \in E_G(i)} z_e \leq 1, \text{ for every vertex } i \end{aligned}$$

Unfortunately, the optimal solutions to this ILP sometimes assign $z_j = y_j = 1$ for the same edge j. In effect, it gets around the bipole edge constraint by denoting one of the contact edges as a bipole edge. We must enforce that a bipole edge must be between a tail vertex and a non-tail vertex. Since we merely require that every tail vertex have a bipole edge, there is no harm if an edge that is between two non-tails is designated a bipole edge. Thus, all we need to do is exclude the possibility of a bipole edge between two tails. In other words, and an edge can be either a contact or a bipole edge: $y_j + e_j \leq 1$ for every edge j. Thus, we arrive at an integer linear program for BPP on arbitrary graphs:

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} y_j \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\ & y_j \leq x_u, u \in \text{edge } j, \text{ for every edge } j \\ & x_i, y_j, e_j \in \{0, 1\} \\ & x_i \leq \sum_{e \in E_G(i)} z_e \leq 1, \text{ for every vertex } i \\ & y_j + z_j \leq 1, \text{ for every edge } j \end{aligned}$$

4.4. **BPP is NP-Complete.** A **reduction from an instance of DkS to an instance of Bipole Packing** is as follows: duplicate the input graph G, then for every vertex v add a partner vertex v' connected to it by an edge. The partner vertices are connected to no other vertices. The claim is that an optimal solution to BPP in this constructed graph allows us to recover an optimal solution to DkS on the original graph. The number of bipoles is equal to the size of the subgraph k, and, for the purposes of formulation as a decision problem, the number of contacts is exactly the same in both problems.

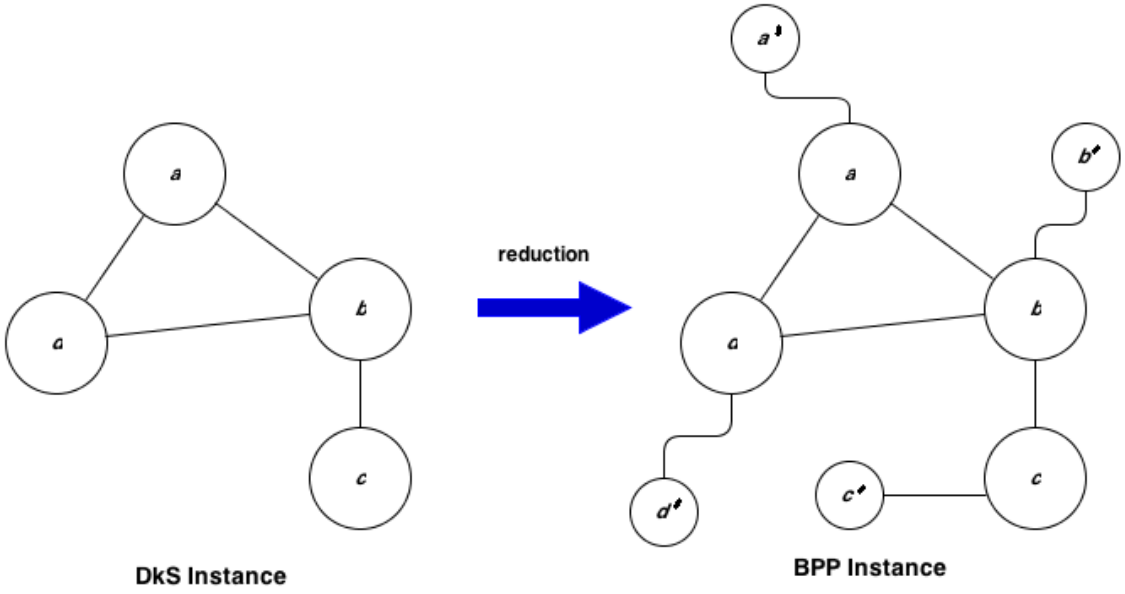

FIGURE 1. Example of reduction from DkS to BPP

Intuitively, it is clear that the added partner vertices are useless for placing tails: they are connected to only one vertex, and that's where the head must go, so they cannot form contacts. So we may assume that tails are only on vertices that were in the original graph. But now, looking at just the set of tails, we have a feasible solution to the original DkS instance, and it must be optimal, as proved below.

Lemma 4.2. *If the given instance of* DENSEST K-SUBGRAPH *is optimal, then the constructed instance of* BIPOLE PACKING *is optimal.*

Proof. This direction is easy: we simply place the tails at exactly those vertices that are chosen in a satisfying configuration for the DkS instance. By construction, there is a vertex adjacent to every tail that was not in the original graph, so no matter what the optimal/satisfying configuration was, this will be a feasible solution to BPP: simply place the heads in the partner vertices. The number of contacts is exactly the same, thus, the Bipole Packing instance is a satisfying instance (feasible, same number of contacts). □

Lemma 4.3. *If the constructed instance of* BIPOLE PACKING *is optimal, then the original instance of* DENSEST K-SUBGRAPH *must have been optimal.*

Proof. We will construct a feasible solution to the original DkS instance, using any optimal configuration to the constructed BPP instance. First, without any loss in contacts, we flip any bipoles which had their tail in a partner vertex. As argued above, this operation cannot decrease the number of contacts. The feasible solution to DkS consists of just the

tail vertices after this transformation. Since each tail is now in a position that was in the original graph, this solution is feasible.

Suppose it is not optimal. Then there must be a feasible solution to DkS with a greater number of contacts. Let us lift this solution back to the constructed BPP instance: place tails for every vertex in this better optimal solution, then put the heads in the partner vertices. This produces a feasible solution to the constructed BPP instance with a greater number of contacts than the hypothetical optimal: a contradiction. □

Thus, in a general setting, these lemmas lead us to conclude:

Theorem 4.4. Bipole Packing *is NP-Complete.*

4.5. **Shared-Heads Relaxation.** We can generalize the BPP to obtain an easier problem. Instead of requiring every tail to be attached to a head, we allow several tails to share a single head. Thus, we exclude the situations where a solution vertex is surrounded on all sides by other solution vertices: each solution vertex must have "room to breathe", but their standards are lower than the tails of bipoles.

This is an easier problem, and its integer linear program is simpler and faster to solve. This does not seem to be of much use: we have solved a different, easier problem. However, we will present evidence that in fact the solution to this problem is also a solution to the BPP in the 3D cubic lattice, a fact we can exploit to solve larger instances than by solving the BPP directly.

Definition An instance of the Shared-heads Bipole Packing Problem consists of a graph G and an integer $k < \frac{|V(G)|}{2}$. A *feasible solution* X is a set of k vertices such that every vertex in the set is adjacent to at least one vertex not in the set. The number of contacts in a feasible solution is the number of edges in the subgraph induced by the vertices in V, $|E(G[X])|$. An *optimal solution* is a feasible solution that maximizes the number of contacts.

Both the BPP and Shared-heads BPP can be solved on the same input G and k, where the tails in the BPP are associated with the vertices in the Shared-heads BPP. A solution that is feasible in both has the same number of contacts in both. Every feasible solution of the BPP on a given graph G is a feasible solution of the Shared-heads BPP on that same graph, but the reverse is not true: there are feasible solutions to the Shared-heads BPP that violate the feasibility criteria for BPP. These are solutions that do not admit a perfect matching; in other words, two more more tails in the feasible set X are adjacent to the same vertex in $V(G) - X$ and no other. Thus, these tails *share* the same head. This is why the Shared-heads BPP is a relaxation of the BPP.

Theorem 4.5. *If an optimal solution to the Shared-heads relaxation associated with a given instance of BPP is also a feasible solution to the BPP, then it is an optimal solution to that BPP instance.*

Proof. Suppose there is an optimal solution to the Shared-heads relaxation associated with an instance of BPP that is also a feasible solution to the BPP, but is not an optimal solution to the BPP. Then there must be some feasible solution to the BPP with a greater number

of contacts. But this solution must admit a perfect matching (it is feasible for BPP). For every tail vertex (associated with the solution vertices for the Shared-heads BPP) there exists a vertex adjacent to it that is not a tail: the head is such a vertex. Thus, it must also be a feasible solution to the Shared-heads BPP, contradicting the optimality of the first solution. □

Theorem 4.6. *The number of contacts in an optimal solution to the Shared-heads BPP on input G and k is an upper bound for BPP on the same input.*

Proof. The optimal solution to BPP is feasible for Shared-heads BPP. Thus, the number of contacts in the optimal solution to BPP is bounded by above by the number of contacts in the optimal solution to Shared-heads BPP. □

4.6. **An integer linear program for the Shared-heads relaxation.** We can formulate an ILP for the Shared-heads BPP by starting from the ILP for DkS. In that ILP, vertices were colored red, and an edge counted if it was between two red vertices. In the Shared-heads BPP we must ensure in addition that each red vertex has at least one non-red neighbor. That is, the number of red neighbors of any red vertex must be strictly smaller than its degree (number of neighbors). The number of red neighbors of a non-red vertex can be greater.

Theorem 4.7. *We can combine these conditions in a single constraint:* $\sum_{i \text{ neighbor } v} x_i \leq$ *degree*$(v) - x_v$.

Proof. The left-hand side of this inequality counts the number of red neighbors of the vertex v, because all variables are binary and $x_i = 1$ iff vertex i is red.

If vertex v is colored red then $x_v = 1$ and the right-hand side equals the degree of v minus 1. The inequality states that the number of red neighbors must be less than or equal to the degree minus 1, which is equivalent to the original condition ("number of red neighbors of any red vertex must be strictly smaller than its degree") because the variables are binary.

If vertex v is not red then $x_v = 0$ and the right-hand side is just the degree of v. Since the number of red neighbors cannot exceed the degree, in this case the inequality does not constrain the solution. Thus, we have constrained every red vertex to have at least one non-red neighbor and constrained no other vertex. □

Here is the ILP for Shared-heads BPP. Note that it is substantially simpler than the ILP for BPP, because we don't have to find a matching for every configuration of tails. All we need is to count the number of tail neighbors of every tail.

$$\begin{aligned}
\max \quad & \sum_{j=1}^{m} y_j \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_i = k \\
& y_j \leq x_u, u \in \text{edge } j, \text{ for every edge } j \\
& \sum_{i \text{ neighbor } v} x_i \leq \text{degree}(v) - x_v, \text{ for every vertex } v \\
& x_i, y_j \in \{0, 1\}.
\end{aligned}$$

5. DkS on integer lattices

Theorem 5.1. *The optimal solution to DkS on the 2D square lattice has* $2n - \lceil 2\sqrt{n} \rceil$ *contacts among* n *vertices.*

The proof of the above theorem is in "A closed-form expression for the densest n-subgraph of $\mathbb{Z}^3$", attached as appendix. A different proof, given in terms of extremal graphs and the inequalities governing them, is in [2]. Here, we describe the structure of the optimal solutions found in either of these two constructive proofs. Since the tails on any given plane of a configuration in the 3D cubic lattice form an instance of DkS on the 2D square lattice, this structure will be important for determining the detailed structure of biplane configurations.

Let k be the greatest integer such that $k^2 \leq n$. Then, the first k^2 vertices lie in a square formation, the rest wrapping around the corner of the square. Note that the series of optimal solutions is formed by adding one vertex to each previous optimal solution. In other words, each optimal configuration is a subset of every larger one. An important consequence is that we can stack optimal solutions in 3 dimensions and get the maximum number of contacts between them.

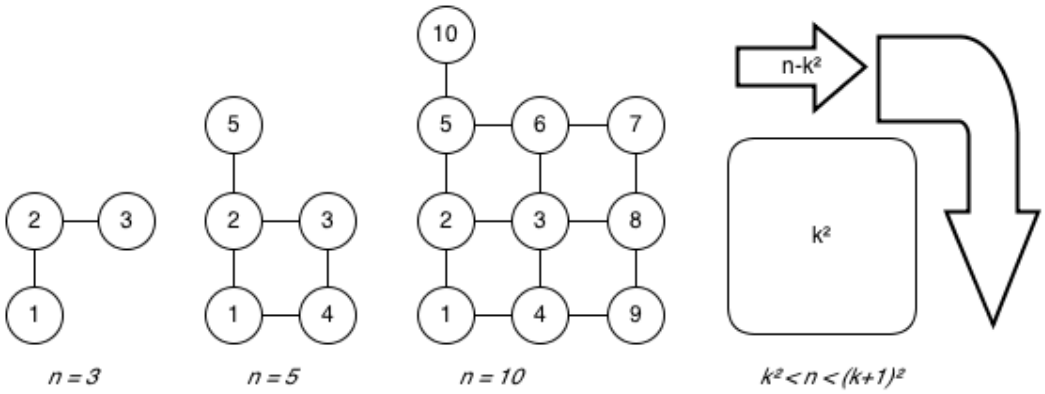

FIGURE 2. Optimal solutions to DkS in the 2D square lattice

6. BPP on lattices and the Biplane Conjecture

6.1. **Biplane Theorem on 2D square lattice.**

Theorem 6.1. *The biplane is an optimal solution to the shared-heads relaxation on the 2D square lattice.*

This theorem is proved in [8] by first showing that the number of contacts in any bipole configuration cannot be greater than $\frac{3n-4}{2}$. This is because each tail can have at most 3 tail neighbors, so there are $3n$ contacts among n bipoles and each has been counted twice. Further, there are at least 4 corner-most tails that can have at most 2 tail neighbors, so the double-count is at most $3n - 4$.

Next, it is shown by construction and counting that the biplane configuration achieves this upper bound, proving that it is an optimal solution.

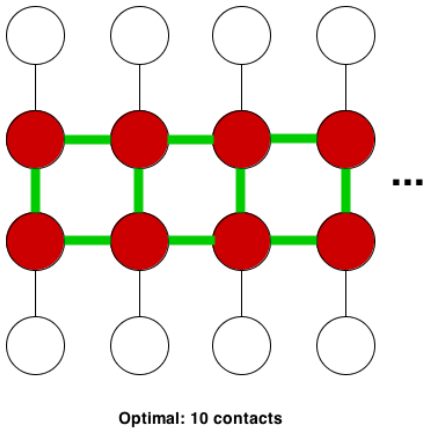

Figure 3. Optimal solution to BPP on 2D square lattice: 1D biplane

6.2. **Biplane Conjecture.**

We now have several reasons to state our suspicion: when allowed to occupy the vertices and edges defined by a space-filling lattice, the minimum energy configuration for a collection of bipoles is achieved when their tails are in two symmetrical planes facing each other. This is analogous to the 2D square situation, and we know that phospholipids in a solvent, driven mainly by hydrophobic interactions, assume the bilayer configuration.

We state several forms of the Biplane Conjecture. The first is our working definition for this paper, but a weaker form may be easier to prove, and a more general form is what we are eventually after.

Conjecture 6.1 (Biplane Conjecture)**.** *Every optimal solution to the Bipole Packing Problem on the 3D cubic lattice has its tails on one of two adjacent planes.*

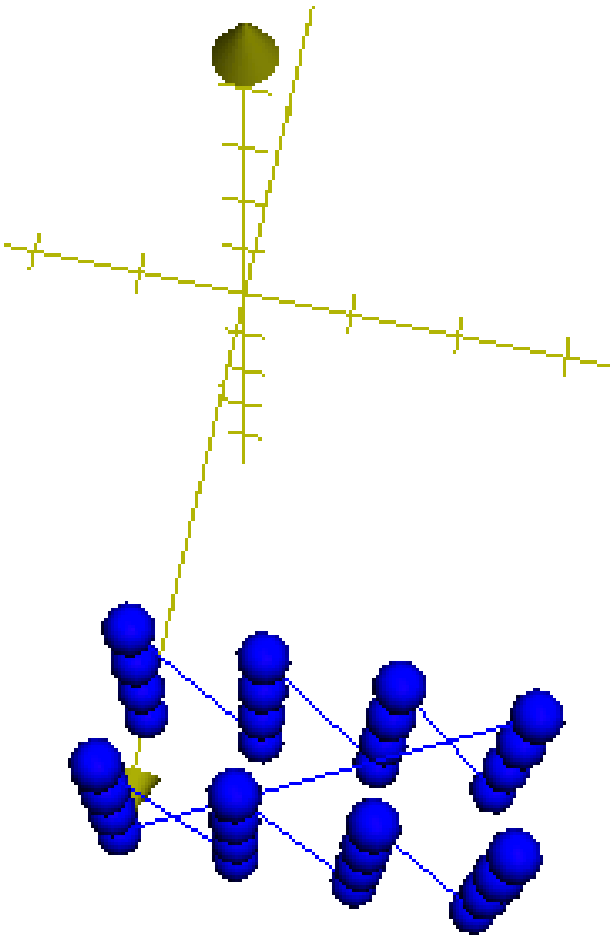

FIGURE 4. An illustration of the Biplane Conjecture: tail positions in computed optimal configuration, $n = 32$.

In the next section we will use the solution to DkS on the 2D square lattice to compute the number of contacts in an optimal biplane configuration in the 3D cubic lattice. If the biplane configuration is optimal, this must be the maximum number of contacts achievable. Thus, we may restate the conjecture as follows:

Conjecture 6.2 (Biplane Conjecture: Contacts)**.** *No configuration of* $2n$ *bipoles in the 3D cubic lattice has more than* $5n - 2\lceil 2\sqrt{n} \rceil$ *contacts.*

This is a strictly weaker statement than the first, because as we will see below, the first implies the second. Its utility is in its quantitative statement of the Biplane Conjecture, opening the door to proof using inequalities and bounds.

In the 3D cubic lattice we find computationally that the Shared-Heads BPP has the same solution as the BPP. Thus, the Biplane Conjecture can be restated in terms of solutions to the Shared-Heads BPP:

Conjecture 6.3 (Biplane Conjecture: Shared-Heads)**.** *Every optimal solution to the Shared-Heads Bipole Packing Problem on the 3D cubic lattice has its tails on one of two adjacent planes.*

Conjecture 6.4 (Biplane Conjecture: Shared-Heads Contacts)**.** *No configuration of* $2n$ *vertices in the 3D cubic lattice with an empty neighbor for each vertex has more than* $5n - 2\lceil 2\sqrt{n} \rceil$ *contacts.*

These statements are strictly stronger than their counterparts above, because optimal solutions to the Shared-Heads BPP on a graph are automatically optimal for BPP on the same graph, *provided that they are feasible solutions to BPP on that graph.*

So far, we considered the 3D cubic lattice. But there is reason to believe that bipoles self-assemble into a biplane in other lattices, particularly FCC. Thus, we generalize the conjecture to the FCC lattice.

Conjecture 6.5 (Biplane Conjecture: FCC)**.** *Every optimal solution to the Bipole Packing Problem on the FCC lattice has its tails on one of two adjacent planes.*

It is known that the FCC lattice is the densest packing of spheres among all space-filling lattices. If the Kepler Conjecture is true, the FCC lattice is the solution to the sphere-packing problem in continuous 3D space. Thus, in analogy, we expect that if the biplane configuration is optimal on the FCC lattice it is also optimal in the off-lattice model.

Conjecture 6.6 (Biplane Conjecture: Off-lattice)**.** *Bipoles free to move in continuous 3D space minimize their energy (maximize contacts) when their tails are on one of two adjacent planes.*

This is the strongest statement so far, because it implies the Biplane Conjecture on any "reasonable" space-filling lattice, that is, any lattice which can accommodate a biplane configuration.

6.3. **Contacts in a biplane configuration.** We can count the number of contacts in any given biplane configuration, but of all configurations with the bipoles lying on two facing planes, we are interested in the one with the greatest number of contacts. Forgetting the other sheet for the moment, the problem of arranging the tails in a single sheet of the biplane is exactly DkS on the 2D square lattice, because the heads are irrelevant for this problem: they are facing outward, lying in an adjacent sheet and not constraining the movement of the tails within the sheet. Further, we can appeal to the stacking property of optimal solutions to DkS in 2D to find the biplane configuration with the greatest number of contacts.

Theorem 6.2. *The optimal biplane configuration with* $2n$ *bipoles in the 3D cubic lattice has* $5n - 2\lceil 2\sqrt{n} \rceil$ *contacts.*

Proof. Suppose there are a bipoles in one the sheets of the biplane, and b bipoles in the other, where $a + b = 2n$. The contacts in this configuration may be divided into three groups: those among tails in the first sheet considered in isolation; those in the second sheet; and those between one tail in either sheet. There can be at most $\min(a, b)$ contacts of the third variety.

Each sheet is an instance of DkS on the 2D square lattice, and since an optimal solutions to DkS in this lattice is a subset of every larger optimal solution, we can optimize each sheet in isolation and be assured that we still get $\min(a, b)$ contacts between the sheets, the maximum allowed. Thus, there are $2a - \lceil 2\sqrt{a} \rceil$ contacts in the first sheet and $2b - \lceil 2\sqrt{b} \rceil$ in the second.

Now we use calculus to maximize

$$2a - \lceil 2\sqrt{a} \rceil + 2b - \lceil 2\sqrt{b} \rceil + \min(a, b)$$

by substituting $2n - a$ for b (and assuming without loss of generality that $a \leq \frac{n}{2}$), and obtain the result in the statement of the theorem. □

A corollary of this theorem is that the bipoles are evenly split between the two sheets in an optimal biplane configuration.

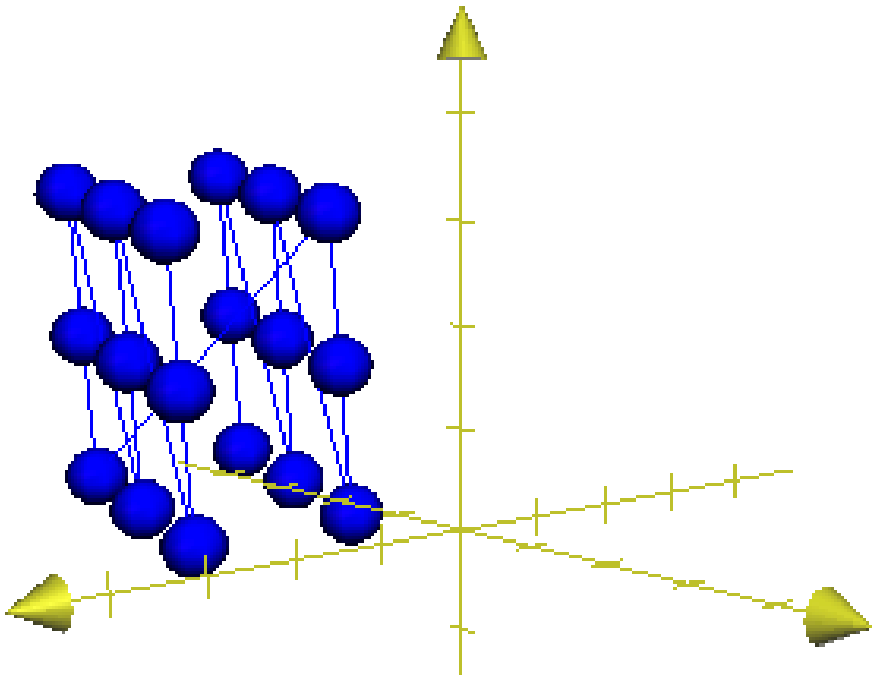

FIGURE 5. Tail positions in optimal biplane configuration, $n = 18$

6.4. **The Biplane Approximation to the Shared-heads relaxation.**

Theorem 6.3. *The biplane configuration on n bipoles is at least a $1-O(\frac{1}{\sqrt{n}})$ approximation to the optimal number of contacts in the shared-heads relaxation associated with the BPP on the 3D cubic lattice.*

Proof. In a lattice with kissing number k, each tail residue can have at most $k-1$ neighbors that are also tails: one of the spots is taken up by its attached head. Thus, there can be at most $\frac{n(k-1)}{2}$ contacts among n bipoles, where we have divided by two because each contact is counted twice (once for each tail it connects). In the 3D cubic lattice, this upper bound is $\frac{5n}{2}$.

From Theorem 6.2, the number of contacts in a biplane configuration is $\frac{5n-2\lceil 2\sqrt{n}\rceil}{2}$ (the factor of two comes because the result above is formulated for $2n$ bipoles). Dividing this count by the upper bound gives us the required approximation ratio:

$$\frac{5n-2\lceil 2\sqrt{n}\rceil}{5n} = 1 - \frac{2\lceil 2\sqrt{n}\rceil}{5n} \approx 1 - \frac{4\sqrt{n}}{5n} \approx 1 - \frac{1}{\sqrt{n}}$$

□

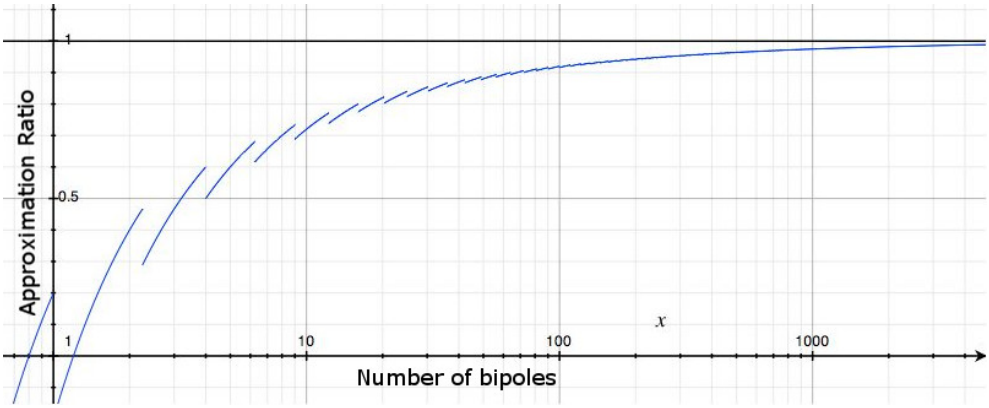

FIGURE 6. Approximation ratio approaches 1 as n increases

7. A Tighter Upper Bound: Counting External Edges

Istrail, Serafim et al. developed a tighter upper bound for the number of contacts in the 3D cubic lattice in [9]. This was done by determining a lower bound on the number of edges between tails and solvent. This line of argument can be summarized as follows: first, consider the width each plane that makes up a configuration of tails by slicing along one of the axis directions. Let $[HP]$ be the number of tail-head edges, and $[HW]$ the number of tail-solvent edges. Let H_a and W_a be the number of slices along the X and Y dimensions that have width ≥ 2. Let H_b and W_b be the number of slices with width 1. Then, for each slice:

(1) **Count exposed neighbors.** If the width of the slice is 1, then there are at least 2 exposed faces. If width ≥ 2, there are at least 4 exposed faces. Thus, in any of the three dimensions along which we can slice the configuration,

$$[HP] + [HW] \geq 2(H_a + W_a) + (H_b + W_b)$$

(2) **Don't double-count!** The count is independent up to two dimensions:

$$[HP] + [HW] \geq 4(H_a + W_a) + 2(H_b + W_b)$$

(3) **Constrain dimensions** The entire configuration must fit inside a volume defined by its height and width:

$$H_a W_a L + 1 H_b W + 1 H W_b \geq N$$

(See [9] for details of this derivation.)

We want to minimize the lower bound on $[HW] + [HP]$ so that the number of contacts, $5n - [HW] - [HP]$, is maximized. This can be formulated as an integer optimization problem:

$$\begin{array}{rl} \max & 5n - 4(H_a + W_a) - 2(H_b + W_b) \\ \text{s.t.} & H_a W_a L + 1 H_b W + 1 H W_b \geq N \\ & H_a, W_a, H_b, W_b, L > 0 \\ & L \leq H_a + H_b \leq W_a + W_b \end{array}$$

The optimal solution to the integer problem above is an upper bound to the number of contacts in the 3D cubic lattice. As before, every upper bound leads to an approximation ratio: we divide the biplane contact number by this upper bound. The tighter the upper bound is, the greater our confidence that the biplane configuration is optimal.

We were unable to solve this optimization problem analytically, but by computing the answer for small values of n ($n \leq 500$ in steps of 10) we were able to estimate the leading term in the Taylor series expansion for the approximation ratio: it is approximately $1 - O(\frac{1}{n^2})$.

8. Computational Results: Evidence for the Biplane Conjecture

We used two approaches to solve the Bipole Packing Problem on 3D cubic and FCC lattices: a direct branch-and-bound algorithm, and a commercial integer linear program solver used on the ILP developed above Both are general methods, suitable for solving BPP on arbitrary graphs. Both are exact methods, guaranteeing that the solutions they find are optimal, and that they find all optima. We verified the Biplane Conjecture on these lattices for values of n that begin to be meaningful, in the sense that the biplane is clearly developed. For the 3D cubic lattice using the integer linear program, we achieved exact solutions for sizes in the range of small globular proteins, $n = 150$ (that is to say, 300 residues).

8.1. **Branch-and-Bound Algorithm for BPP.** We decomposed the Bipole Packing Problem into an exponential part (DkS) and a polynomial part (bipartite matching). We exploited this decomposition to formulate integer linear programs for BPP, which we solved using off-the-shelf ILP solvers. To corroborate these results we used this decomposition to formulate a branch-and-bound algorithm that explicitly constructs bipole configurations by exploring a search tree.

Let T be the set of vertices selected to be tails at any point in the search, and suppose *ChooseBestDkSVertex(T,V)* is an oracle that returns the best vertex to add to the set T to obtain an optimal DkS solution, given that all choices better than and including V are inadmissible. Then the search may be written non-deterministically as follows:

Algorithm 1 Nondeterministic search for BPP solutions

```
T ← Empty
for i from 0 to n-1 do
  V ← NULL
  while T ∪ V does not admit a perfect matching do
    V ← ChooseBestDkSVertex(T,V)
  end while
  T ← T ∪ V
end for
```

Thus, the best possible DkS solution is found that is still feasible for BPP. A deterministic algorithm must branch on every choice of vertex V and count contacts when it has placed all k. Suppose at a certain node of the search tree the set T does not admit a perfect matching with vertices outside of T. Then any addition of vertices to T can only make the situation worse. Thus, the search can terminate that branch of the search tree. Thus, we do not need to construct full configurations of k tails before we can tell that a solution will be infeasible.

Algorithm 2 Deterministic search for BPP solutions. Input: T

```
if [T, V(G) − T does not admit a perfect matching then
  return Infeasible
end if
if T has k vertices then
  return Contacts in T
end if
best ← 0
for every vertex V that is not in T do
  best ← max(Search(T ∪ V), best)
end for
return best
```

We used two methods to improve this branch-and-bound algorithm. First, we avoided making symmetrical choices of V. This depends on the lattice being used. Second, we developed an upper bound on the number of contacts in a configuration S, given some subset T of it. That is to say, at any point in the search we have fixed some vertices in the set T and all children of this node must add to this set. We wish to bound from above the number of contacts in all children of this node.

To do this, we use the well-known separation argument: if one set has i elements and another set has j, they can have at most $\min(i, j)$ elements in common. In other words, if the set T has i vertices already in it, then we are going to add $k - i$ more vertices and there can only be $\min(i, k - i)$ contacts between the past and the future. Finally, we use an upper bound for the number of contacts between a set of $k - i$ tails to complete the estimate:

$$\text{Contacts(solutions with } T \text{ as a subset)} \leq \text{Contacts}(T) + \min(i, k - i) + \text{Estimate}(k - i)$$

An easy estimate is the Shared-heads BPP estimate for that lattice: it is $\frac{n(d-1)}{2}$ where d is the kissing number of that lattice, and n is number of bipoles we're interested in. So we plug in $n = k - i$ and get the upper bound we need.

An even better upper-bound can be designed if we carefully arrange the order in which vertices V are picked. If they are considered plane-by-plane, we can exploit a more powerful separation property in these lattices: the only contacts between the future and the past can happen at the last layer, so the $\min(i, k - i)$ can be replaced by $\min(l, k - i)$ where l is number of tails in the penultimate layer of T.

Algorithm 3 Bounded deterministic search for BPP solutions. Input: T

```
if [T, V(G) − T] does not admit a perfect matching then
  return Infeasible
end if
if Contacts(T) + min(l, k − i) + Estimate(k − i) ≤ best so far then
  return Bounded
end if
if T has k vertices then
  return Contacts in T
end if
best ← 0
for every vertex V that is not in T modulo symmetry classes of G, in layered order do
  best ← max(Search(T ∪ V), best)
end for
return best
```

This algorithm can be modified easily to solve the Shared-heads BPP as follows: instead of checking whether $[T, V(G) - T]$ admits a perfect matching, we need only count the number of non-red neighbors of every vertex in T; if it is smaller than 1, all children of this node are infeasible. This is an even cheaper polynomial computation than matching, but

using it has one disadvantage: since matching is a stricter condition, it eliminates more nodes of the search tree. In fact, since counting non-red neighbors is *so* cheap, we found a two-pronged approach to be ideal: we only computed a matching if a solution was feasible for Shared-heads BPP. Thus, we avoided many matching computations but eliminated just as many branches as before.

Our final optimization was to manage the internal state of the vertex set T so that a fresh matching was not generated each time. For this to work, we used the augmenting paths matching algorithm. When a new vertex is added to T, it is removed from the set of non-T vertices and added to the other side. The bipartite graph is modified, and any new augmenting paths are found. If, when all augmenting paths have had flow pushed along them, there are still unmatched vertices of T, the partial solution is infeasible (and so are all its children).

8.2. **Results.** For every value of n considered, our programs eliminated every branch of the search tree and stopped after finding all optimal solutions. Every optimal solution was consistent with the Biplane Conjecture; that is, every optimal solution has its tails in two planes facing each other.

When $n = 2k^2$ for some integer k, there is exactly one optimal solution modulo the placement of heads (and lattice translation, rotation and reflection symmetries): the tails are arranged in two squares facing each other.

When $n = 2k^2 + 1$, the additional tail can be in any position along the periphery of either square.

When $n = 2k^2 + 2$, there are many more options. The first additional tail can be in any periphery position, while the second must be adjacent to it. It can be on the same plane (for two additional contacts in that plane) or in the facing plane exactly opposite the first (one contact between planes, one in the plane).

When $n = 2k(k+1)$ for some integer k, there is again exactly one solution modulo heads and lattice symmetries: there are two rectangles facing each other, each of size k by $k + 1$.

For values of n between $2k(k+1)$ and $2(k+1)^2$, we have the same combinatorial explosion in the number of optimal solutions as between $2k^2$ and $2k(k + 1)$.

Note that this substructure is a direct consequence of the solution to DkS in the 2D square lattice, with the additional complication that the same number of contacts can often be added in two ways: by placing the next tail on the same plane, or by placing it on the facing plane.

These computations were redone using the Shared-Heads BPP constraints: instead of a full matching, we are satisfied by a single solvent neighbor next to each tail. The optimal solutions to the Shared-heads BPP were identical to those for BPP in the 3D cubic lattice. Since the constraints for Shared-Heads BPP are much simpler, we could solve them for much larger sizes: n 150. For each value of n considered, the optimal solution was a biplane configuration and thus feasible for the BPP constraints. Thus, it was the optimal solution to the BPP with the same graph and n. In this way we verified the Biplane Conjecture for the 3D cubic lattice for values of n that are physically meaningful, around the size of the smallest globular proteins (300 residues).

9. Related Topics

This is an extended "Further Research" section with some extensions to the idea of bipole packing. In keeping with the research presented so far, I have tried to combine equal amounts of computational exploration and the identification of computational problems and their complexity.

9.1. **Bipole Configuration Labeling Problem.**

So far in this paper the Bipole Packing Problem has been decomposed as first picking reds, then solving a polynomial-time matching problem. Another decomposition, in some sense the inverse of the one above, is as follows: first pick locations for bipoles, a matching of size n (that is, a set of n edges with no shared vertices); then, for each edge, decide which end will be red. Once a matching has been chosen, a series of n binary decisions completely specifies the configuration of bipoles and, consequently, the number of contacts.

The relevant problem to study is the part after a matching is chosen: given a matching M (the set of non-oriented bipoles), for each edge label one vertex tail (red) and the other vertex head (pick orientation), so that the number of red-red contacts is maximized.

Definition An instance of the Bipole Configuration Labeling Problem (BCLP) consists of a graph G, and a set M of n edges with no shared vertices (a *matching*). A *labeling* picks exactly one red vertex from each edge in the matching: this is a *feasible solution* to the BCLP. Thus, a labeling is a set K of red-tail vertices. As usual, the number of contacts in a feasible solution is the number of edges in the subgraph induced in G by just the set of reds, $|E(G[\{u : [u, v] \in K\}])|$. An *optimal solution* is one that maximizes the number of contacts.

The relevant question is, can we solve the BCLP in polynomial time? Unfortunately when the graph G and matching M are completely general, we demonstrate a reduction from the NP-Complete problem of finding a cocycle of maximum cardinality in an arbitrary graph. Worse, this problem remains NP-Complete on graphs with each vertex of degree three. As we will see in the following construction, the instances of BCLP that arise in solving the Bipole Packing Problem on the 3D cubic lattice are such graphs. To be clear: this does not rule out polynomial-time algorithms for solving the instances of BCLP that arise in solving the BPP on particular lattices, like k-dimensional cubic lattices or the FCC lattice, or on some general class of lattices that includes these.

Definition Given a graph G and some subset C of its vertices, let $d(C)$ be the set of edges with exactly one extremity in C. This set $d(C)$ will be called a *cocycle*. The cardinality of this cocycle is simply the size of the set, the number of edges in it. The problem of finding a cocycle of maximum cardinality in a graph is NP-Hard, and the corresponding decision problem is NP-Complete. This decision problem can be formulated as: given G and C as above, return True iff $d(C)$ is a cocycle of maximum cardinality.

Given an instance of the max cocycle problem, we can transform it into an instance of the bipole configuration labeling problem as follows. For each vertex v in G, create a bipole with vertices v_1 and v_2. For every edge between vertex v and u in G, draw an edge from v_1

to u_2, and from v_2 to u_1. Then the claim is, that a labeling of the bipoles with maximum contacts, gives a cocycle of maximum cardinality in the original graph G.

Theorem 9.1. *The* Bipole Configuration Labeling Problem *is* NP-Complete.

Proof. We shall consider a vertex v in the original graph to be in the subset C iff v_1 is colored red. By the construction of the bipole configuration graph, a contact is counted iff for some edge (u, v) in the original graph G, either u_1 and v_2 were labeled as red OR u_2 and v_1 were labeled as red. In other words, a contact is counted in the graph iff exactly one of its extremities has been selected to be in the subset C – that is, when the edges corresponding to contacts form a cocyle. □

9.2. **Monte Carlo and Statistical Mechanics.** The bulk of this paper is devoted to making a case for the optimality of the biplane configuration in the lattices considered. For one special lattice (2D square) we can prove this is so, while for others we must be satisfied with heuristics, simulations and lower bounds. In physical terms, we have associated an energy with particular configurations and tried to find a configuration that minimizes energy. Such energy minima occur in nature only when the temperature is zero (or close to it). At all higher temperatures, a system will divide its time among several low-energy configurations, of which the lowest may only be one. The issue is that of entropy, a force that represents the fact that the overwhelmingly large majority of configurations do not minimize energy but occur with finite probability.

This section presents a sketch of the application of statistical mechanics to problems like this one. To keep this discussion short and focused, we will use a simplified model of the Bipole Packing Problem on the 2D square lattice, a model built on our knowledge of the true lowest-energy configuration. In effect, we will zoom in on the statistical properties of configurations that are close to optimal.

Definition A Biplane Spin Configuration on the 2D square lattice is a feasible solution X to the Bipole Packing Problem obtained by taking an optimal configuration of n bipoles and flipping m of the bipoles, that is, exchanging their heads and tails.

We have constrained the bipoles to lie in their optimal biplane locations, but we let them fidget a little: they can flip their orientations in place. This is a tightly constrained but still varied and interesting subset of the phase space of all configurations, and it includes the optimal. One justification of this statistical analysis is that it is a detailed look at the phase space explored in the initial part of a Markov Chain Monte Carlo trajectory in a model where movement is slow. The number of contacts of configurations in this subset varies from the maximum $2n - \lceil 2\sqrt{n} \rceil$ to 0.

Definition Let $E(X)$ denote the energy of this configuration, where $E(X) = -Contacts(X)$.

Thus, energy is minimized when contacts are maximized, and 0 is the highest it can be. How many configurations are there for a given value of energy? There is only one optimal in this model, and that is when every tail is facing in. There are several of energy 0, when tails that face in are isolated from each other and thus do not form contacts. And for most values of energy between these extreme values, there may be several configuration with that energy.

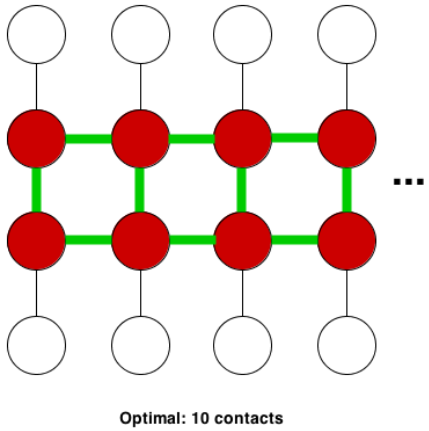

FIGURE 7. Biplane Spin Configuration with maximum contacts. Only one such configuration.

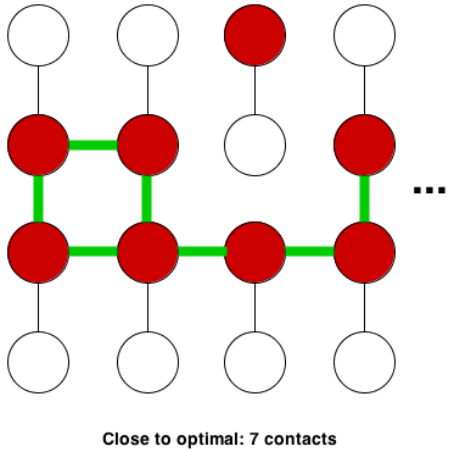

FIGURE 8. Biplane Spin Configuration with close to maximum contacts

Definition Let $g(E)$ denote the *degeneracy* of energy level E, the number of configurations with energy E.

The natural statistical ensemble for this situation is the canonical ensemble, since the number of bipoles is fixed but the energy is allowed to vary. The amount of time the system prefers to stay in a particular configuration depends on the energy of that configuration: the greater its energy, the less likely it is that the system assumes that configuration. In this ensemble the precise dependence of probability on energy is given by the *Boltzmann factors*:

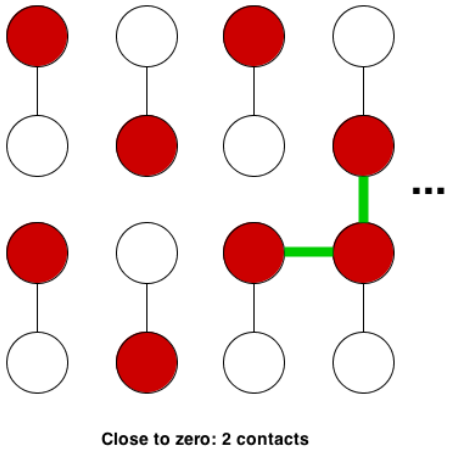

FIGURE 9. Biplane Spin Configuration with close to zero contacts

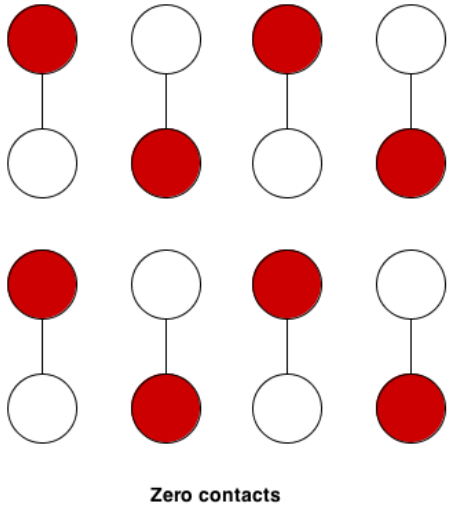

FIGURE 10. Biplane Spin Configuration with minimum (zero) contacts. Many configurations have zero contacts.

$$P(\text{Configuration with energy E}) \propto e^{-\beta E}$$

These factors are the solution to an optimization problem: maximize the number of possible states under the constraint that the total energy and size of the ensemble are fixed. β originates as a Lagrange multiplier whose value is determined by the total energy

TABLE 1. Degeneracies for the energy levels of Biplane Spin Configurations for $n = 6$

Energy level	Degeneracy
0	17
-1	15
-2	15
-3	8
-4	4
-5	4
-6	0
-7	1

of the ensemble: $\beta \propto 1/T$ is a measure of the *average energy* of the system, and this is a property set by its surroundings (which it is in thermal equilibrium with). The probabilities must sum to 1, so we introduce Z, a *normalizing factor* that we will need to consider as a *function* in its own right.

$$P(\text{Configuration with energy E}) = \frac{e^{-\beta E}}{Z}$$

What can we say from our formalism so far? If we suppose for now we somehow know the value of Z, we can compute the probability of any given configuration at a given temperature. In particular, we can compute the amount of time the system spends in its minimum energy configuration, as a function of temperature:

Thus, the system spends a significant portion of its time in the optimal configuration only near 0 temperature. Increase the temperature even a little bit, and the biplane is lost. Is this as terrible a blow as it seems to the work of this paper? A more nuanced view of the system's behavior is given by the *average* energy of the system, instead of just the likelihood of the *most probable* energy. If the average energy of the system is near that of the optimal configuration, we may be sure that the system is near the biplane configuration, deviating from it by only a few flips. If, on the other hand, we find that the average is very different from the optimal, we will know that the force of entropy has overwhelmed that of energy-minimization: the system is disordered, and the biplane has disintegrated.

How shall we compute the average energy of the system at a given temperature? We can certainly sum over all configurations, weighting the energy of each by its probability of occurrence as given by the Boltzmann factors. However, at this point it is worthwhile to look at how the normalizing factor Z is computed. As it will turn out, we can obtain all thermodynamic quantities (like average energy, entropy, or, later, the number of bipoles in a biplane configuration) by computing derivatives and logarithms of Z taken as a function: the *partition function*.

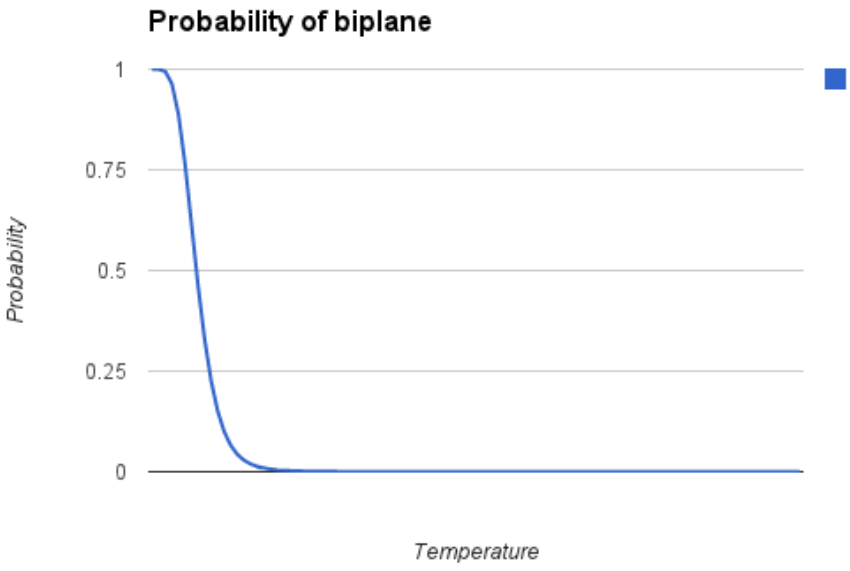

FIGURE 11. Probability of optimal (biplane) configuration vs. Temperature

Since Z is the normalizing factor that ensures the Boltzmann probabilities sum to 1, we must have

$$Z = \sum_{\text{configurations i}} e^{-\beta E_i}$$

where each configuration contributes a single term to the sum. Realizing that all that matters is the energy of a configuration, we can group configurations by energy and sum over these energy levels, multiplying the Boltzmann factor with the number of configurations in that energy level:

$$Z = \sum_{\text{energy levels j}} g(E_j) e^{-\beta E_j}$$

where $g(E_j)$ denotes the number of configurations with energy E_j, the *degeneracy* of that energy level, as defined above.

This is convenient: we have gone from an exponential number of terms (one for each configuration) to a more manageable linear number (the energy is bounded from above by $\frac{5n}{2}$). To compute Z, determine the degeneracy of each energy level. What about other sums over configurations, like the average energy?

$$\langle E \rangle = \sum_{\text{configurations i}} E_i \frac{e^{-\beta E_i}}{Z}$$

where each energy is weighted by the normalized probability of occurrence *of that configuration*. Here is the key idea: we can perform the same transformation to obtain

$$\langle E \rangle = \sum_{\text{energy levels j}} E_j g(E_j) \frac{e^{-\beta E_j}}{Z}$$

where each energy is now weighted by the normalized probability of occurrence *of that energy level*, and a linear number of terms. We are back to the problem of computing degeneracies. What about all the other thermodynamic quantities? So far we have worked with the expression for the partition function Z; in fact there is a thermodynamic function associated with each ensemble known as the *generating function* for that ensemble, and every other thermodynamic variable (and equation of state) is available from it. The generating function for the canonical ensemble is the *Helmholtz free energy*, and it too depends on the degeneracies.

In short, the entire problem has been boiled down to this: compute or estimate the degeneracy of each energy level. This may seem like hardly a simplification but we have some strategies for dealing with it in this form: we can parallelize the computations of the degeneracies; we can obtain statistical estimates of their values by simulation; and we can approximate (or even solve!) them analytically. Once these are stored, we can evaluate thermodynamic quantities at values of β.

For this paper we picked a small value, $n = 28$, and generated all Bipole Spin Configurations with n bipoles. We computed the energy of each configuration and compiled a table with the degeneracy of each energy level. This part of the computation could be parallelized easily, as each node could enumerated its portion of the phase space and reported a vector of the number of configurations it encountered in each energy level; these vectors were simply summed to produce the final degeneracies. We obtained the average energy as a function of temperature. This can be understood as a graph of "biplanarity" versus temperature: "How far from the biplane configuration does a collection of n bipoles stray at a particular temperature?"

In the figure above, the region of stability of the biplane is in a minuscule range greater than zero. One might suspect that the biplane begins to disintegrate as soon as the temperature is raised above zero. We varied the temperature in finer increments over a smaller range above $T = 0$. By looking at this region in detail we noticed that the number of contacts remains relatively constant near its maximum level for a finite range of temperatures before dropping.

Two noteworthy features of this graph:

(1) At infinitely hot temperatures, the number of contacts asymptotically approaches a finite value (7 in this case). In fact this value is always a quarter of the maximum number of contacts, a description of the system at maximum entropy.
(2) In a small region above zero temperature, the biplane or near-biplane configuration is indeed the average; we see a little "ledge" in the graph near zero temperature and it is at the maximum number of contacts, $\frac{3n-4}{2}$ (40 in this case). This suggests a phase transition in the stability of the bilayer.

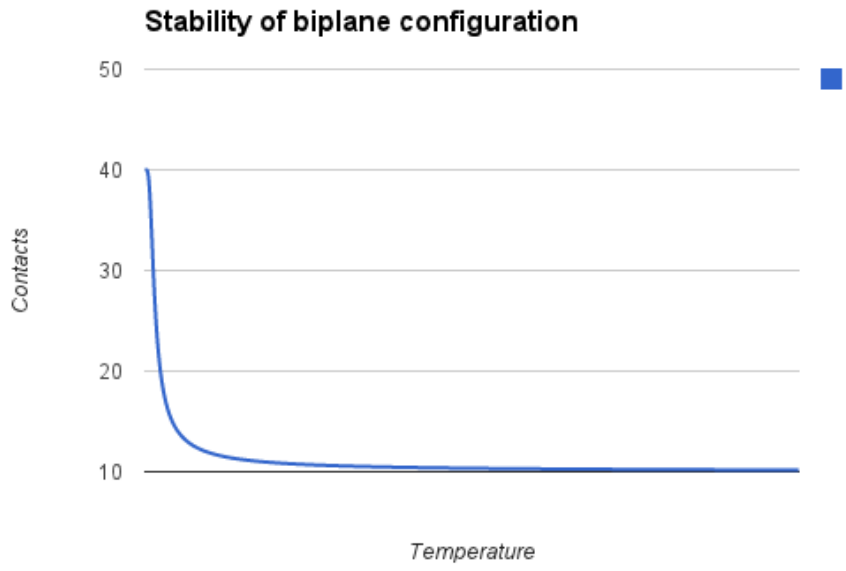

FIGURE 12. Average number of contacts vs. Temperature

So far we have ignored the possibility that bipoles can leave the biplane altogether and wander about the rest of the system. By switching to the *grand canonical ensemble* we allow the number of bipoles in the system to vary and hold the system in equilibrium with a solution of bipoles at some density. To evaluate thermodynamic quantities of interest we must now have a set of degeneracies for every value that n can take. There are opportunities for algorithms to compute this larger set efficiently.

When exhaustive enumeration of configurations is infeasible, even once, we may try estimating the degeneracies. We generate states by making random moves over and over again, accepting a move with probability 1 if it lowers the energy, and with a certain probability otherwise: this is the Metropolis-Hastings algorithm, a method that samples the probability distribution (somewhat) faithfully if our set of moves is connected, and if the acceptance probability satisfies the detailed balance condition.

Acknowledgements

My greatest thanks go to Sorin Istrail, who introduced me to this problem, advised me in my research, listened patiently to wacky ideas that could never work, and encouraged me to pursue them anyway. I owe everything to the love of Nidhi and Manu Lauria. They fed and clothed me; they paid for college. They have given me life. Paul Valiant graciously agreed to be reader. I could not have pursued this research without Kathleen Lister's daily love and reassurance, especially after

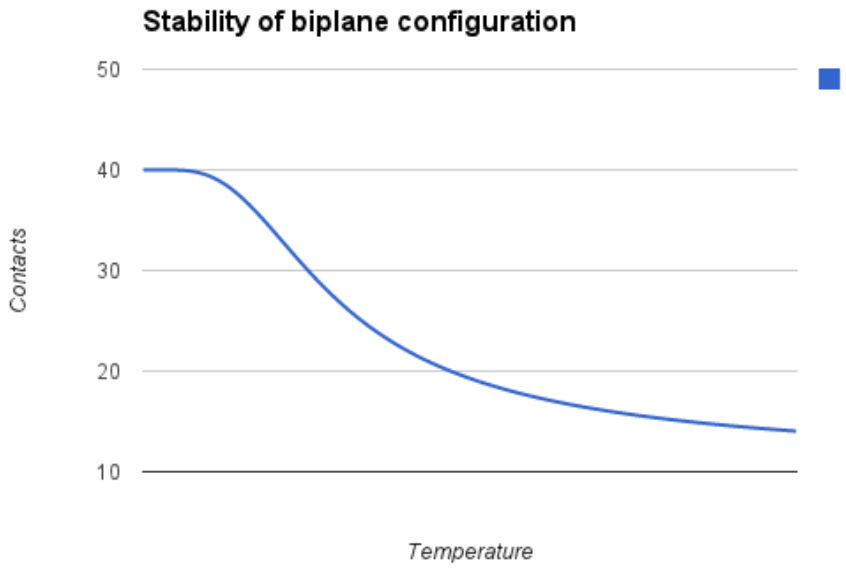

FIGURE 13. Average number of contacts vs. Temperature, zoomed in near T=0

I realized how hard it was going to be. I had many valuable conversations with Manu Lauria about the algorithms in this paper and their efficiency. Richard Stratt's stunningly clear lectures on statistical mechanics are the only reason that connection was developed. Erika DeBenedictis first pointed out to me that what I had, in fact, was a model of the lipid bilayer. The work of Huafeng Xu and my other colleagues at D. E. Shaw Research opened my eyes to a world of sampling and simulation that changed my thinking about biological models forever. Carleton Coffrin, Daniel Weinreich, Claire Mathieu, Geir Agnarsson and John Hughes each took the time to understand my research and answer questions. Tom Doeppner organized everything and put up with my confusion and questions. As for my roommates and other friends at Brown University, I was humbled and inspired by the passion I saw them casually pour into their projects every day. This work owes its existence to the environment it was nurtured in.

REFERENCES

[1] Lau, K. F., Dill, K. A.: "A lattice statistical mechanics model of the conformational and sequence spaces of proteins", *Macromolecules* 22 (10): 398697 (1989).

[2] Frank Harary; Heiko Harborth: "Extremal animals", *J. Combinatorics Information Syst. Sci.,* 1, no. 1, 1-8, (1976).

[3] Bela Bollobas and Imre Leader: "Edge-Isoperimetric Inequalities in the Grid", *Combinatorica* 11 (4) (1991).

[4] Ken A. Dill: "Theory for the folding and stability of globular proteins", *Biochemistry* 24 (6), 1501-1509 (1985).
[5] Feige, U.; Kortsarz, G.; Peleg, D.:, "The dense k-subgraph problem", *Algorithmica* 29: 410421 (1997).
[6] Keil, J.; Brecht, T.: "The complexity of clustering in planar graphs", *The Journal of Combinatorial Mathematics and Combinatorial Computing* 9: 155159 (1991).
[7] Berg JM, Tymoczko JL, Stryer L.: *Biochemistry*. 5th edition. New York: W H Freeman (2002). Section 12.4, "Phospholipids and Glycolipids Readily Form Bimolecular Sheets in Aqueous Media" Available from: http://www.ncbi.nlm.nih.gov/books/NBK22406/
[8] Sorin Istrail, Fumei Lam: "Combinatorial Algorithms for Protein Folding in Lattice Models: A Survey of Mathematical Results", *Communications in Information and Systems* vol. 9, no. 4, pp. 303-346 (2009).
[9] Sorin Istrail, Alan Hurd, Ross A. Lippert, Brian Walenz, Serafim Batzoglou, John H. Conway, Fredrick W. Peyerl: "Prediction of Self-Assembly of Energetic Tiles and Dominos: Experiments, Mathematics and Software", *Internal Technical Report*, Sandia Labs (2000).
[10] C. F. Gauss. "Recension der untersuchungen uber die eigenschaften der positiven ternaren quadratischen formen von ludwig august seeber"., *Journal fur die reine und angewandte Mathematik*, pages 312320 (1840).