SAND2000-2762J

CONSTRAINED WALKS AND SELF-AVOIDING WALKS: IMPLICATIONS FOR PROTEIN STRUCTURE DETERMINATION

JEAN-LOUP FAULON*, MARK D. RINTOUL†, AND MALIN M. YOUNG‡

Abstract.

While the protein folding problem on lattices is known to be NP-hard, we prove in this paper that lattice protein structures of size n matching specific lists of $O(n)$ distance constraints can be determined in linear time on 2D honeycomb, 2D square, 3D diamond and 3D cubic lattices.

Key words. constrained self-avoiding walk.

AMS subject classifications.

1. Introduction. A protein is a heteropolymeric chain of amino acids that folds into a complex three-dimensional native structure. The structure of a protein is intrinsically related to its biological function(s). In the human cell, for example, some 100,000 proteins form the architecture of the cell and carry out its metabolism. With the amount of genomic information currently being generated, there is great interest in the development of high-throughput methods for determining the structures of the encoded protein products.

Experimentally determining the structures of all of these proteins is simply not possible for the foreseeable future even with advances in X-ray crystallography and NMR spectroscopy. Many proteins cannot be crystallized and both experimental techniques require 10-100 milligrams of pure materials and take months to years to elucidate a structure. Thus, many researchers are pursuing alternative computational approaches.

Sequence alignment, fold recognition and homology modeling approaches make predictions based on the similarity between a sequence of unknown structure and the database of known protein structures. Although these approaches are quite successful

*(Corresponding Author) Computational Biology and Materials Technology Department, Sandia National Laboratories, P.O. Box 5800, Albuquerque NM 87185-1111 (jfaulon@cs.sandia.gov).

†Computational Biology and Materials Technology Department, Sandia National Laboratories, P.O. Box 5800, Albuquerque NM 87185-1111 (rintoul@sandia.gov).

‡Biosystems Research Department, Sandia National Laboratories, P.O. Box 969, Livermore, CA 94551-9214 (mmyoung@sandia.gov).

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

at predicting and modeling structures when there is sufficient similarity to a known structure, they become unreliable in the absence of such similarity[Daune 1999]. Ab initio structure prediction techniques, because they aim to predict the structure of a protein from its amino acid sequence alone, can be used for structure prediction in situations where there are no homologs of known structure. However the accuracy of ab initio techniques, with a few notable exceptions, remains quite low[Duan 1998].

From the aforementioned observations, it becomes clear that the development of computational methods to calculate structure from sequence is more important than ever. Exhaustive search of a protein's conformational space is clearly not a feasible algorithmic strategy. The number of possible conformations is exponential in the length of the protein sequence. Instead, computational models of protein folding have been formulated to find the global minimum of a potential energy function. Thus, the protein folding problem has been formalized as a global optimization problem. Computational complexity results and approximate solutions to the protein optimization problem have been derived both in continuous and discrete spaces. In continuous space, solutions are based on distance geometry[Crippen 1988], while in discrete space, solutions have been proposed using lattice models[Fraenkel 1993]. Unfortunately, in both cases, the protein optimization problem has been proven to be strongly NP-hard[Saxe 1979, Moré 1995, Hart 1997].

Although protein folding is strongly NP-hard, this fact does not prevent the problem to be solvable in polynomial time when restrictions are applied. In fact, as shown in Crippen and Havel[Crippen 1988], if all pairwise distances between the amino acids are known and a solution exists, then protein folding can be solved in polynomial time using distance geometry. The solution involves the computation of the largest three eigenvalues and eigenvectors of a specific distance matrix. However, compiling $n(n-1)/2$ pairwise distances for a protein of size n requires an experimental effort that is as costly as to elucidate protein structures directly from NMR or X-ray crystallography data.

Recently, a rapid experimental protocol for protein structure determination has been proposed[Young 2000]. The technique, called MS3D is based on cross-linking technology and mass spectrometry. The protocol requires 3-4 days and less than $100\mu g$ of protein and provides low-resolution interatomic distance information. While current cross-linking technology limits the number of distance information to about $n/3$, experimentalists anticipate that this technology coupled with combinatorial chemistry techniques will lead in the near future to a number of distance information close to n. The obvious question is then: *Is it at all feasible to elucidate protein structures using only O(n) distance information?*

This paper demonstrates that on the usual 2D and 3D lattices protein structures can indeed be retrieved in polynomial time when at most n specific pairwise distances are provided. More precisely, we represent protein structures by self-avoiding walks on lattices, and prove in section 2 that for the 2D honeycomb, 2D square, 3D diamond and 3D cubic lattices there exist sets of distance constraints, named canonical distance lists, for which there is only one corresponding walk or self-avoiding walk. In section 3 we give algorithms for the 3D lattices that constructs walks and self-avoiding walks from given canonical distance lists. We also prove our algorithms to run in linear time. Finally, in section 4 we give algorithms for the 3D lattices that compiles canonical distance lists using a linear number of distance measurements.

2. Unique representation for walk and self-avoiding walk. In the following, L is an infinite lattice embedded in the Euclidean d-dimensional space, $\mathbb{R}$ is the set of reals, and $\mathbb{Z}$ is the set of integers. Let x be a site of L, $\omega(x)$ are the Euclidean coordinates of site x. We assume that neighboring sites are equidistant, that is, for all closest neighbors y of site x we have $|\omega(x) - \omega(y)| = 1$ where $|\ |$ denotes the Euclidean norm. Examples of lattices verifying the equidistance constraint are honeycomb, square, diamond, and d-dimensional cubic lattices.

DEFINITION 2.1. *A n-step walk, W, on L, beginning at the origin, is a sequence of vertices $\omega(0), \omega(1), \ldots, \omega(n)$ with $\omega(0) = 0$ and $|\omega(i) - \omega(i-1)| = 1$ for $1 \leq i \leq n$*

DEFINITION 2.2. *A n-step self-avoiding walk, SAW, on L, is a n-step walk on L verifying $\omega(i) \neq \omega(j)$ for all pairs $i \neq j$ in $[0, \ldots, n]$.*

DEFINITION 2.3. *Two n-step walks W_1 and W_2 on L are said to be isomorphic iff a combination, π, of translations, rotations, and reflections of L can be found such that $\pi(W_1) = W_2$. The definition naturally extends to SAW's.*

DEFINITION 2.4. *A m-distance list, D, associated to an n-step walk W is a sequence of m triplets (i, j, d_{ij}) where $\omega(i) \in W$, $\omega(j) \in W$, and $d_{ij} \in \mathbb{R}$.*

DEFINITION 2.5. *Given W a n-step walk on L and D an associated m-distance list. W is a m-constrained, n-step walk on L, iff for all triplets (i, j, d_{ij}) in D, $|\omega(i) - \omega(j)| = d_{ij}$. The definition naturally extends to SAW's.*

DEFINITION 2.6. *A m-distance list, D, is said to be W-canonical iff there is only one m-constrained n-step walk on L corresponding to D. The definition naturally extends to SAW's, where D is then said to be SAW-canonical.*

2.1. Lattice coordinates. Prior seeking a unique representation for walk and self-avoiding walk, it is necessary to define more precisely the four lattices we are considering. These are 2D honeycomb, 2D square, 3D diamond and 3D cubic. In order to compute cartesian coordinates on these lattices we arbitrarily place the origin of our Euclidean space at specific lattice sites as described in Figures 1–4. For any given lattice site x we then define some subset of the operators L, R, U, D, F, and B depending on the coordination number of the lattice. These operators return different

neighbors of x. The operators are defined next for each lattice considered.

2.1.1. L = 2D honeycomb lattice. For the honeycomb lattice we define only the functions L, R, and U. The functions are defined indirectly using the Euclidean coordinates of their images. As can be seen in Figure 1, the sites x on a honeycomb lattice fall into two distinct sets, based on the positions of their neighbors. We define set $\mathbb{A}$ as $\{x|\omega(x)\cdot(0,1) = 3k/2, k \in \mathbb{Z}\}$, and set $\mathbb{B}$ as $\{x|x \notin \mathbb{A}\}$. Using our orientation in Figure 1, set $\mathbb{A}$ contains the sites that have a neighbor directly below, and set $\mathbb{B}$ contains the sites that have a neighbor directly above. We define the operators separately for the two sets as:

- $x \in \mathbb{A}$

$\omega(L(x)) = \omega(x) + (-\sqrt{3}/2, 1/2)$; $\omega(R(x)) = \omega(x) + (\sqrt{3}/2, 1/2)$;

$\omega(U(x)) = \omega(x) + (0, -1)$.

- $x \in \mathbb{B}$

$\omega(L(x)) = \omega(x) + (-\sqrt{3}/2, -1/2)$; $\omega(R(x)) = \omega(x) + (\sqrt{3}/2, -1/2)$;

$\omega(U(x)) = \omega(x) + (0, 1)$.

2.1.2. L = 2D square lattice. For the square lattice (cf. Figure 2) we define the functions L, R, U, and D with their associated Euclidean coordinates:

$\omega(L(x)) = \omega(x) + (-1, 0)$; $\omega(R(x)) = \omega(x) + (1, 0)$;

$\omega(U(x)) = \omega(x) + (0, 1)$; $\omega(D(x)) = \omega(x) + (0, -1)$.

2.1.3. L = 3D diamond lattice. For the diamond lattice we define the functions L, R, U, and D. As one can see in Figure 3, the sites on the diamond lattice fall into two distinct sets based on their neighbor environment, just like the 2D honeycomb lattice. We define set $\mathbb{A}$ to be $\{x|\omega(x)\cdot(1,0,0) = 2k\sqrt{3}, k \in \mathbb{Z}\}$ and $\mathbb{B}$ to be $\{x|x \notin \mathbb{A}\}$. For the two sets, we define the operators as:

- $x \in \mathbb{A}$

$\omega(L(x)) = \omega(x) + (\sqrt{1/3}, 0, -\sqrt{2/3})$; $\omega(R(x)) = \omega(x) + (\sqrt{1/3}, 0, \sqrt{2/3})$;

$$\omega(U(x)) = \omega(x) + (-\sqrt{1/3}, \sqrt{2/3}, 0); \quad \omega(D(x)) = \omega(x) + (-\sqrt{1/3}, -\sqrt{2/3}, 0).$$

- $x \in \mathbb{B}$

$$\omega(L(x)) = \omega(x) + (-\sqrt{1/3}, 0, -\sqrt{2/3}); \omega(R(x)) = \omega(x) + (-\sqrt{1/3}, 0, \sqrt{2/3});$$

$$\omega(U(x)) = \omega(x) + (\sqrt{1/3}, \sqrt{2/3}, 0); \quad \omega(D(x)) = \omega(x) + (\sqrt{1/3}, -\sqrt{2/3}, 0).$$

Note that for all lattice sites x, we have: $\omega(R(L(x))) = \omega(x)$, $\omega(L(R(x))) = \omega(x)$, $\omega(U(D(x))) = \omega(x)$, and $\omega(D(U(x))) = \omega(x)$. These relations define operator inverses for all of the operators, and we will take advantage of this during later proofs. We also note that for a point in set $\mathbb{A}$, all of its neighbors belong to set $\mathbb{B}$, and vice versa.

2.1.4. L = 3D cubic lattice. For the cubic lattice (cf. Figure 4) the functions L, R, U, D, F, and B are:

$$\omega(L(x)) = \omega(x) + (-1, 0, 0); \quad \omega(R(x)) = \omega(x) + (1, 0, 0);$$

$$\omega(U(x)) = \omega(x) + (0, 1, 0); \quad \omega(D(x)) = \omega(x) + (0, -1, 0);$$

$$\omega(F(x)) = \omega(x) + (0, 0, 1); \quad \omega(B(x)) = \omega(x) + (0, 0, -1).$$

2.2. Unicity theorems. The series of theorems that follows defines canonical representations for walks and self-avoiding walks on honeycomb, square, diamond and cubic lattices.

THEOREM 2.7. *On the honeycomb lattice the following distance list is W-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 0), & \text{or} \\ (i, i-k_i, 0), & \text{or} \\ (i, i-k_i, \sqrt{3}), & \text{or} \\ (i, i-k_i, 2), & \text{or} \\ (i, i-k_i, \sqrt{7}), & \text{or} \\ \emptyset, & \text{otherwise.} \end{cases}$$

with

$k_i = \{$ *min* $k \in [3, \ldots, n]$ *s.t.* $i - k \geq 0$, $\omega(i-k) \neq \omega(i-1)$, *and* $\omega(i-k) \neq \omega(i-2)$ $\}$.

Proof. We prove by induction that there is only one walk W corresponding to D. Let us first notice that for a zero-step, or a one-step walk, $D = \emptyset$. The theorem

is true since on the honeycomb lattice there is only one non-isomorphic walk having zero or one step.

We now assume the theorem is true up to step $i-1$, that is, there is only one walk corresponding to the list $d_2, d_3, \ldots, d_{i-1}$. We need to prove that given d_i there is only one location for step i. If $d_i = (i, i-2, 0)$ then step i is uniquely located and has the same location than step $i-2$, i.e., $\omega(i) = \omega(i-2)$ (cf. Figure 5.a). All other distances use step $i-k_i$, which is the last step on the walk $\omega(0), \omega(1), \ldots, \omega(i-1)$ at a different location than steps $i-1$ and $i-2$. If $d_i = (i, i-k_i, 0)$ then step i is uniquely located and $\omega(i) = \omega(i-k_i)$ (cf. Figure 5.b). Without loss of generality, we now assume that the lattice site corresponding to step $i-1$ is an element of $\mathbb{A}$. (the reader can verify that the remaining of the proof holds true when $2/3\ (\omega(i-1)\cdot(0,1)-1) \in \mathbb{Z}$). There are three possible locations for step $i-2$, these are $\omega(i-2) = \omega(R(i-1))$, $\omega(i-2) = \omega(L(i-1))$, and $\omega(i-2) = \omega(U(i-1))$.

We first treat the case $\omega(i-2) = \omega(R(i-1))$, which is depicted in Figure 5.

If $d_i = (i, i-k_i, \sqrt{3})$, then the path between steps $i-k_i$ and i is of length 2 (cf. Figure 5.b). Since step $i-1$ is neighbor of step i and since step $i-1$ is in the path between $i-k_i$ and i, we have $\omega(i-k_i) = \omega(U(i-1)$ (as in Figure 5.b), or $\omega(i-k_i) = \omega(L(i-1))$. From the definition of k_i we cannot have $\omega(i-k_i) = \omega(i-2) = \omega(R(i-1))$. It is then obvious that $\omega(i) = \omega(L(i-1))$ if $\omega(i-k_i) = \omega(U(i-1)$, and $\omega(i) = \omega(U(i-1))$ otherwise. Note that $\omega(i) \neq \omega(R(i-1))$ since we have $\omega(i-2) = \omega(R(i-1)$ and we already have eliminated the case $\omega(i) = \omega(i-2)$ (i.e., $d_i = (i, i-2, 0)$).

If $d_i = (i, i-k_i, 2)$ then according to Figure 5.c, we have $\omega(i) = \omega(U(i-1))$ if $\omega(i-k_i) = \omega(R(i-2))$, and $\omega(i) = \omega(L(i-1))$ if $\omega(i-k_i) = \omega(U(i-2))$. For $d_i = (i, i-k_i, \sqrt{7})$, we have $\omega(i) = \omega(L(i-1))$ if $\omega(i-k_i) = \omega(R(i-2))$, and $\omega(i) = \omega(U(i-1))$ if $\omega(i-k_i) = \omega(U(i-2))$.

Finally, if $d_i = \emptyset$ none of the other values for d_i have been found. In particular, k_i was not found such that the location of step $i-k_i$ differs from the locations of

steps $i-1$ and $i-2$. Consequently the walk W up to step $i-1$ occupies only two adjacent lattice sites. Step i can be located at three different locations $\omega(U(i-1))$, $\omega(L(i-1))$, and $\omega(R(i-1))$. The case $\omega(i)=\omega(R(i-1)$ is eliminated since $\omega(i-2)=\omega(L(i-1)$ and $d_i \neq (i,i-2,0)$. It is easy to verify that the two remaining cases lead to isomorphic walks. Consequently, there is only one non-isomorphic walk corresponding to $d_i = \emptyset$.

We have thus proven that there is only one walk up to step i corresponding to the list of distances $d_2, d_3, \ldots, d_i$ in the case $\omega(i-2)=\omega(R(i-1))$. The reader can verify that a similar conclusion can be drawn for the two cases $\omega(i-2)=\omega(L(i-1))$, and $\omega(i-2)=\omega(U(i-1))$. □

COROLLARY 2.8. *On the honeycomb lattice the following distance list is SAW-canonical.* $D = d_3, d_4, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i,i-3,2), & \textit{or} \\ (i,i-3,\sqrt{7}), & \textit{or} \\ \emptyset, & \textit{otherwise.} \end{cases}$$

Proof. Let us first notice that for a zero-, one-, or two-, step walk $D=\emptyset$. The corollary is true since on the honeycomb lattice there is only one non-isomorphic SAW having zero, one, or two steps. The corollary is then proven by induction using the same technique than for Theorem 2.7. Note that d_i takes only three values here. Because sites cannot overlap with SAW we necessarily have $k_i = 3$ (cf. Theorem 2.7 for definition of k_i). Hence, $d_i = (i,i-3,2)$ correspond to $(i,i-k_i,2)$ and $d_i = (i,i-3,\sqrt{7})$ correspond to $(i,i-k_i,\sqrt{7})$. The cases $d_i = (i,i-2,0)$, $d_i = (i,i-k_i,0)$, and $d_i = (i,i-k_i,\sqrt{3})$ are not present here because they all implies non SAW. □

THEOREM 2.9. *On the square lattice the following distance list is W-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 0), & \text{or} \\ (i, i-2, 2), & \text{or} \\ (i, i-k_i, h), & \text{or} \\ (i, i-k_i, \sqrt{h^2+4}), & \text{or} \\ \emptyset, & \text{otherwise.} \end{cases}$$

with

$k_i = \{$ *min* $k \in [3, \ldots, n]$ *s.t.* $i - k \geq 0$, $|\omega(i-k+2) - \omega(i-k)| \neq 0$, *and* $|\omega(i-k+2) - \omega(i-k)| \neq 2\}$.

and $h = |\omega(i-1) - \omega(i-k_i+1)|$. *Note:* k_i *is the last step on the walk* $\omega(0), \omega(1), \ldots, \omega(i-1)$ *that does not fall onto the straight line going through* $\omega(i-2)$ *and* $\omega(i-1)$.

Proof. We prove by induction that there is only one walk W corresponding to D. Let us first notice that for a zero-step, or a one-step walk $D = \emptyset$. The theorem is true since on the square lattice there is only one non-isomorphic walk having zero or one step.

We now assume the theorem is true up to step $i-1$, that is, there is only one walk corresponding to the list $d_2, d_3, \ldots, d_{i-1}$. We need to prove that given d_i there is only one location for step i.

If $d_i = (i, i-2, 0)$ then step i is uniquely located and has the same location than step $i-2$, i.e., $\omega(i) = \omega(i-2)$ (cf. Figure 6.a).

If $d_i = (i, i-2, 2)$ then step i is uniquely located using the following formula. If A is the operator defined by $\omega(i-1) = \omega(A(i-2))$ then $\omega(i) = \omega(A(i-1))$.

The two next distances make use of step $i - k_i$. We fully develop the case corresponding to the orientation (1) of vector $[\omega(i-2), \omega(i-1)]$. This case is depicted in Figure 6.b. The reader can verify that all other cases lead to the same conclusion. The four possible locations for step i are $\omega(L(i-1))$, $\omega(R(i-1))$, $\omega(U(i-1))$, and $\omega(D(i-1))$. Positions $\omega(L(i-1))$ and $\omega(R(i-1))$ are eliminated since they correspond to the respective d_i values $(i, i-2, 0)$ and $(i, i-2, 2)$. Recall that $i - k_i$ does not fall on the straight line going through $\omega(i-2)$ and $\omega(i-1)$, therefore, $\omega(i-k_i)$ $= \omega(U(i-k_i+1))$ or $\omega(i-k_i) = \omega(D(i-k_i+1))$.

Let A be the operator defined by $\omega(i-k_i+1) = \omega(A(i-k_i))$. Then, from Figure 6.b we can see that if $d_i = (i, i-k_i, h)$, $\omega(i) = \omega(A^{-1}(i-1))$. Otherwise, if $d_i = (i, i-k_i, \sqrt{h^2+4})$, then $\omega(i) = \omega(A(i-1))$.

Finally, if $d_i = \emptyset$ none of the other values for d_i have been found. In particular, k_i was not found such that the location of step $i-k_i$ does not fall on the line going through steps $i-1$ and $i-2$. Consequently, the walk W up to step $i-1$ is limited to a straight line. If A is defined to be $\omega(i-1) = \omega(A(i-2))$, then $\omega(i) \neq \omega(A(i-1))$ or $\omega(i) \neq \omega(A^{-1}(i-1))$ since those values are covered by the cases $d_i = (i, i-2, 2)$ and $d_i = (i, i-2, 0)$, respectively. The other two directions (both perpendicular to the existing walk) lead to isomorphic walks which represent the one allowed walk corresponding to $d_i = \emptyset$.

We have thus proven that there is only one walk up to step i corresponding to the list of distances $d_2, d_3, \ldots, d_i$ in case (1) (i.e., $\omega(i-1) - \omega(i-2) = (1,0)$). The same conclusion can be drawn for the cases (2),(3) and (4). □

COROLLARY 2.10. *On the square lattice the following distance list is SAW-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 2), & \textit{or} \\ (i, i-k_i, k_i-2), & \textit{or} \\ (i, i-k_i, \sqrt{k_i^2-4k_i+8}), & \textit{or} \\ \emptyset, & \textit{otherwise.} \end{cases}$$

with

$k_i = \{$ *min* $k \in [3, \ldots, n]$ *s.t.* $i-k \geq 0$, $|\omega(i-k+2) - \omega(i-k)| \neq 0$, *and* $|\omega(i-k+2) - \omega(i-k)| \neq 2\}$.

Proof. The corollary is proven using the same technique than for Theorem 2.9, and using the fact that for SAW we have $h = k_i - 2$. Also, note that $d_i \neq (i, i-2, 0)$ since overlap are not allowed with SAW. □

THEOREM 2.11. *On the diamond lattice the following distance list is W-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 0), & or \\ (i, i-3, \sqrt{19/3}), & or \\ (i, i-k_i, h), & or \\ (i, i-k_i, \sqrt{h^2+8/3}), & or \\ (i, i-k_i, \sqrt{h^2+16/3}), & or \\ \emptyset, & otherwise. \end{cases}$$

with

$k_i = \{$ *min* $k \in [4, \ldots, n]$ *s.t.* $i-k \geq 0$, $|\omega(i-ki+2) - \omega(i-ki)| \neq 0$, *and* $|\omega(i-ki+3) - \omega(i-ki)| \neq \sqrt{19/3}$ $\}$,

and $h = |\omega(i-k_i+1) - \omega(i-1)|$.

Note: k_i *is the last step on the walk* $\omega(0), \omega(1), \ldots, \omega(i-1)$ *that does not fall onto the plane going through* $\omega(i-k_i+1)$, $\omega(i-k_i+2)$,..., $\omega(i-1)$.

Proof. We prove by induction that there is only one walk W corresponding to D. Let us first notice that for a zero-step, or a one-step walk $D = \emptyset$. The theorem is true since on the diamond lattice there is only one non-isomorphic walk having zero or one step.

We now assume the theorem is true up to step $i-1$, that is, there is only one walk corresponding to the list $d_2, d_3, \ldots, d_{i-1}$. We need to prove that given d_i there is only one location for step i. If $d_i = (i, i-2, 0)$ then step i is uniquely located and $\omega(i) = \omega(i-2)$. If $d_i = (i, i-3, \sqrt{19/3})$ then step i is uniquely located. To find the location of $\omega(i)$, we first define the operator A as $\omega(i-2) = \omega(A(i-3))$, then $\omega(i) = \omega(A(i-1))$. Physically, this is fairly clear since it just means that to travel a distance $\sqrt{19/3}$ (the maximum distance one can travel on the unit diamond lattice in three steps), one must go in the same direction as the first step.

All other distances use step $i-k_i$, which is the last step on the walk $\omega(0), \omega(1), \ldots,$ $\omega(i-1)$ that does not fall onto the plane going through $\omega(i-k_i+1)$, $\omega(i-k_i+2)$,..., $\omega(i-1)$. We divide the proof into two cases. First, for k_i even, and then for k_i odd. For both cases, we define the operator A as $\omega(i-k_i+1) = \omega(A(i-k_i))$.

An example case for k_i even is shown in Figure 7.b. All of the points shown lie

in a plane, except for i and $i-k_i$. If both points are above or below the plane, then they lie a distance h apart, while if they lie on opposite sides of the plane, they are a distance $\sqrt{h^2+8/3}$ apart. From the diagram, one can see that if $d_i = (i, i-k_i, h)$, $\omega(i) = \omega(A^{-1}(i))$, while if $d_i = (i, i-k_i, \sqrt{h^2+8/3})$, $\omega(i) = \omega(A(i))$.

A typical case for k_i odd is outlined in Figure 7.c. We have the same situation as in k_i even in which all of the points lie in the plane except for i and $i-k_i$. If both points are above or below the plane, then they lie a distance $\sqrt{h^2+8/3}$, while if they are on opposite sides of the plane, they are a distance $\sqrt{h^2+16/3}$ apart. In this case, if $d_i = (i, i-k_i, \sqrt{h^2+8/3})$, $\omega(i) = \omega(A^{-1}(i))$, while if $d_i = (i, i-k_i, \sqrt{h^2+16/3})$, $\omega(i) = \omega(A(i))$.

Finally, if $d_i = \emptyset$ none of the other values for d_i have been found. In particular, k_i was not found such that step $i-k_i$ does not fall onto the plane going through $\omega(i-k_i+1)$, $\omega(i-k_i+2)$,..., $\omega(i-1)$. This implies that all of the steps up to $i-1$ are in a plane. However, the two cases corresponding to i lying in the same plane have already been considered by the cases $d_i = (i, i-2, 0)$ and $d_i = (i, i-3, \sqrt{19/3})$. If $\omega(i-2) = \omega(A(i-3))$, then the two cases already considered correspond to the cases $\omega(i) = \omega(A^{-1}(i-1))$ and $\omega(i) = \omega(A(i-1))$, respectively. There are technically two different cases left, but they are isomorphic. They both correspond to the walk leaving the plane in either the up or the down direction relative to the plane. Since they are isomorphic, we have uniquely located point i up to isomorphisms. It is simply located at one of the two directions from $i-1$ *not* corresponding to either $A(i-1)$ or $A^{-1}(i-1)$.

We have thus proven that there is only one walk up to step i corresponding to the list of distances $d_2, d_3, \ldots, d_i$. □

COROLLARY 2.12. *On the diamond lattice the following distance list is SAW-canonical.* $D = d_3, d_4, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-3, \sqrt{19/3}), & or \\ (i, i-k_i, k_i-2), & or \\ (i, i-k_i, \sqrt{k_i^2 - 4k_i + 8/3}), & or \\ (i, i-k_i, \sqrt{k_i^2 - 4k_i + 16/3}), & or \\ \emptyset, & otherwise. \end{cases}$$

with

$k_i = \{$ *min* $k \in [4, \ldots, n]$ *s.t.* $i - k \geq 0$, $|\omega(i-ki+3) - \omega(i-ki)| \neq \sqrt{19/3}$ $\}$.

Proof. Let us first notice that for a zero-, one-, or two-, step walk $D = \emptyset$. The corollary is true since on the diamond lattice there is only one non-isomorphic SAW having zero, one, or two steps. The corollary is then proven by induction using the same technique than for Theorem 2.11. Note that with SAW we have $h = k_i - 2$ and $d_i \neq (i, i-2, 0)$. □

THEOREM 2.13. *On the cubic lattice the following distance list is W-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 0), & or \\ (i, i-2, 2), & or \\ (i, i-k_i, h_0), & or \\ (i, i-k_i, \sqrt{h_0^2 + 4}), & or \\ (i, i-l_{ki}, \sqrt{h_1^2 + h_2^2}), & or \\ (i, i-l_{ki}, \sqrt{h_1^2 + h_2^2 + 4}), & or \\ \emptyset, & otherwise. \end{cases}$$

with

$h_0 = |\omega(i-1) - \omega(i-k_i+1)|$,

$h_1 = (\omega(i-1) - \omega(i-l_{ki}+1)) \cdot (\omega(i-1) - \omega(i-2))$,

$h_2 = (\omega(i-k_i+1) - \omega(i-l_{ki}+1)) \cdot (\omega((i-k_i+1) - \omega((i-k_i))$,

$k_i = \{$ *min* $k \in [3, \ldots, n]$ *s.t.* $i - k \geq 0$, $|\omega(i-k+2) - \omega(i-k)| \neq 0$, *and* $|\omega(i-k+2) - \omega(i-k)| \neq 2$ $\}$,

and $l_{ki} = \{$ *min* $l \in [3, \ldots, n]$ *s.t.* $i - l \geq 0$, $k_i - l \geq 0$, *and* $($ $(\omega(i-k_i+2) - \omega(i-k_i+1))$ $\times$ $(\omega(i-k_i) - \omega(i-k_i+1))$ $) \cdot$ $(\omega(i-l+1) - \omega(i-l))$ $= \pm 1$ $\}$.

Note: k_i *is the last step on the walk* $\omega(0), \omega(1), \ldots, \omega(i-1)$ *that does not fall onto*

the straight line going through $\omega(i-2)$ *and* $\omega(i-1)$. l_{ki} *is the last step on the walk* $\omega(0),\omega(1),\ldots,\omega(i-1)$ *that does not fall onto the plane going through* $\omega(i-k_i+2)$, $\omega(i-k_i+1)$, *and* $\omega(i-k_i)$.

Proof. We prove by induction that there is only one walk W corresponding to D. Let us first notice that for a zero-step, or a one-step walk $D=\emptyset$. The theorem is true since on the cubic lattice there is only one non-isomorphic walk having zero or one step.

We now assume the theorem is true up to step $i-1$, that is, there is only one walk corresponding to the list $d_2, d_3, \ldots, d_{i-1}$. We need to prove that given d_i there is only one location for step i.

If $d_i = (i, i-2, 0)$ then step i is uniquely located and has the same location than step $i-2$, i.e., $\omega(i) = \omega(i-2)$ (cf. Figure 8.a).

If $d_i = (i, i-2, 2)$ then step i is uniquely located, and its position is given by $\omega(i) = \omega(A(i-1))$, where $\omega(i-1) = \omega(A(i-2))$.

The next two distances make use of step $i-k_i$. We fully develop the case corresponding to an orientation such that $\omega(i-1) = \omega(U(i-2))$, just for the sake of clarity. This case is also depicted in Figure 8.b. The reader can verify that all other cases lead to the same conclusion. The six possible locations for step i are $\omega(U(i-1))$, $\omega(D(i-1))$ $\omega(F(i-1))$, $\omega(B(i-1))$, $\omega(L(i-1))$, and $\omega(R(i-1))$. Positions $\omega(D(i-1))$ and $\omega(U(i-1))$ are eliminated since they correspond to the respective d_i values $(i, i-2, 0)$ and $(i, i-2, 2)$. Recall that $i-k_i$ does not fall on the straight line going through $\omega(i-2)$ and $\omega(i-1)$, therefore, there is four locations for $i-k_i$. If we define $\omega(i-k_i+1) = \omega(A(i-k_i))$, then we have A as either F, B, L, or R. For any of these four possibilities, we know $\omega(i) = \omega(A^{-1}(i-1)$ if $d_i = (i, i-k_i, h_0)$ and $\omega(i) = \omega(A(i-1))$ if $d_i = (i, i-k_i, \sqrt{h_0^2+4})$.

In all the above cases two positions remain unknown. For this instance, we consider the case where $\omega(i-k_i+1) = \omega(F(i-k_i))$, cannot uniquely locate site i if it is at $L(i-1)$ or $R(i-1)$. These positions are given by the distances $(i, i-l_{ki}, \sqrt{h_1^2+h_2^2})$,

and $(i, i-l_{ki}, \sqrt{h_1^2+h_2^2+4})$. These last two distances introduce step $i-l_{ki}$. Recall that $i-l_{ki}$ does not fall onto the plane going through $\omega(i-k_i+2)$, $\omega(i-k_i+1)$, and $\omega(i-k_i)$. Hence, for the direction defined by $\omega(i-l_{ki}+1) = \omega(A(i-l_{ki}))$, the only two possibilities for A are L and R. From Figure 8.c, we can see that for $d_i = (i, i-l_{ki}, \sqrt{h_1^2+h_2^2})$, $\omega(i) = \omega(A^{-1}(i-1))$. Otherwise, if $d_i = (i, i-l_{ki}, \sqrt{h_1^2+h_2^2+4})$, we have $\omega(i) = \omega(A(i-1))$.

Finally, in the case $d_i = \emptyset$, step i can be located at six different positions $\omega(D(i-1))$, $\omega(U(i-1))$ $\omega(F(i-1))$, $\omega(B(i-1))$, $\omega(L(i-1))$, and $\omega(R(i-1))$. The first two positions are eliminated because they correspond to the respective d_i values $(i, i-2, 0)$ and $(i, i-2, 2)$. Note that if $d_i = \emptyset$ none of the other values for d_i have been found. In particular, k_i was not found corresponding to the distances $d_i = (i, i-k_i, h_0)$ and $d_i = (i, i-k_i, \sqrt{h_0^2+4})$. Such a situation may rise when the walk W up to step $i-1$ is limited to the straight line going through steps $i-1$ and $i-2$, or when the walk W up to step $i-1$ is in the plane defined by steps $i-k_i+2$, $i-k_i+1$, and $i-k_i$. The former case is easy to identify, if the walk is limited to a straight line, then all distances up to step $i-1$ have the values $(i, i-2, 0)$ or $(i, i-2, 2)$. In such an instance, all the four possible locations $\omega(F(i-1))$, $\omega(B(i-1))$, $\omega(L(i-1))$, and $\omega(R(i-1))$ lead to walks that are isomorphic. The latter case is identified when there is at least one distance up to step $i-1$ taking the value $(i, i-k_i, h_0)$ or $(i, i-k_i, \sqrt{h_0^2+4})$. In this cases too, the two possible locations for step i lead to isomorphic walks.

We have thus proven that there is only one walk up to step i corresponding to the list of distances $d_2, d_3, \ldots, d_i$ in case (1.1), i.e., $\omega(i-1) - \omega(i-2) = (1, 0)$. The same conclusion can be drawn for the cases (2.1)-(2.4),(3.1)-(3.4) and (4.1)-(4.4). □

COROLLARY 2.14. *On the cubic lattice the following distance list is SAW-canonical.* $D = d_2, d_3, \ldots, d_n$, *where*

$$d_i = \begin{cases} (i, i-2, 2), & or \\ (i, i-k_i, k_i-2), & or \\ (i, i-k_i, \sqrt{k_i^2 - 4k_i + 8}), & or \\ (i, i-l_i, \sqrt{h_1^2 + h_2^2}), & or \\ (i, i-l_i, \sqrt{h_1^2 + h_2^2 + 4}), & or \\ \emptyset, & otherwise. \end{cases}$$

with

$h_1 = (\omega(i-1) - \omega(i-l_{ki}+1)) \cdot (\omega(i-1) - \omega(i-2))$,

$h_2 = (\omega(i-k_i+1) - \omega(i-l_{ki}+1)) \cdot (\omega((i-k_i+1) - \omega((i-k_i))$,

$k_i = \{$ *min* $k \in [3, \ldots, n]$ *s.t.* $i - k \geq 0$, $|\omega(i-k+2) - \omega(i-k)| \neq 0$, *and* $|\omega(i-k+2) - \omega(i-k)| \neq 2$ $\}$,

and $l_{ki} = \{$ *min* $l \in [3, \ldots, n]$ *s.t.* $i - l \geq 0$, $k_i - l \geq 0$, *and* $($ $(\omega(i-k_i+2) - \omega(i-k_i+1))$ $\times$ $(\omega(i-k_i) - \omega(i-k_i+1))$ $) \cdot (\omega(i-l+1) - \omega(i-l)) = \pm 1$ $\}$.

Proof. The corollary is proven using the same technique than for Theorem 2.13 and using the fact that for SAW we have $h_0 = k_i - 2$. Also note that for SAW, $d_i \neq (i, i-2, 0)$. □

3. Walks construction from canonical distance lists. Keeping our biological application in mind, we give in this section an algorithm that constructs walks on a diamond and cubic lattices from W/SAW-canonical distance lists. Note that proteins are 3D self-avoiding structures and note that the diamond lattice is better suited for proteins than the cubic lattice since it pictures the tetravalent character of carbon with the appropriate bond angles and the appropriate trans conformation of hydrocarbon chains. The algorithms are given next, similar algorithms can be derived for the 2D lattices of the previous section.

```
BUILD-WALK-DIAMOND(D,W)
input:  -D:  n-distance list
```

output: -W: n-step walk

local: -i, k_i: integer

begin

1. $\omega(0) = (0,0,0)$; $\omega(1) = \omega(U(0))$;

2. for $i = 2$ to n do $\omega(i)$ = COMPUTE-COORD-DIAMOND(W,D,i);

3. done

end

COMPUTE-COORD-DIAMOND(W, D, i)

input: -D: n-distance list

output: -W: n-step walk

local: -i, k_i: integer

-h: real

-A: direction operator

-$o(d_i)$, $e(d_i)$, and $r(d_i)$: functions returning the first and second element of triplet d_i

begin

1. if $d_i = \emptyset$ then

2. let A be the operator such that $\omega(i-i) = \omega(A(i-2))$

3 return any neighbor of $\omega(i-1)$ other than $\omega(A(i-1))$ or $\omega(A^{-1}(i-1))$

4. else

5. $k_i = o(d_i) - e(d_i)$; $h = |\omega(i-1) - \omega(i-k_i+1)|$

6. let A be the operator such that $\omega(i-k_i+1) = \omega(A(i-k_i))$

7. if $r(d_i) = h$ then $\omega(i) = \omega(A^{-1}(i-1))$

8. else if $r(d_i) = \sqrt{h^2 + 16/3}$ then $\omega(i) = \omega(A(i-1))$

9. else if k_i even then $\omega(i) = \omega(A(i-1))$

10. else $\omega(i) = \omega(A^{-1}(i-1))$

11. endif;

12. done

end

BUILD-WALK-CUBIC(D, W)

input: -D: n-distance list

output: -W: n-step walk

local: -i, k_i: integer

begin

1. $\omega(0) = (0,0,0)$; $\omega(1) = \omega(U(0))$;

2. for $i = 2$ to n do $\omega(i)$ = COMPUTE-COORD-CUBIC(W,D,i);

3. done

end

COMPUTE-COORD-CUBIC(W, D, i)

input: -D: n-distance list

output: -W: n-step walk

local: -i, k_i: integer

-h: real

-A: direction operator

-$o(d_i)$, $e(d_i)$, and $r(d_i)$: functions returning the first and second element of triplet d_i

begin

1. if $d_i = (i, i-2, 0)$ then $\omega(i) = \omega(i-2)$

2. else if $d_i = (i, i-2, 2)$ then

3. let A be defined by $\omega(i-i) = \omega(A(i-2))$

4. $\omega(i) = \omega(A(i-1))$

5. else

6. let k_i be the first point not on a line between $\omega(i-1)$ and $\omega(i-2)$

7. let $h_0 = |\omega(i-1) - \omega(i-k_i+1)|$

8. let A be defined by $\omega(i-k_i+1) = \omega(A(i-k_i))$

9. if $d_i = (i, i-k_i, h_0)$ then $\omega(i) = \omega(A^{-1}(i-1))$

10. else if $d_i = (i, i-k_i, \sqrt{h_0^2+4})$ then $\omega(i) = \omega(A(i-1))$

11. else

12. let l_{ki} be the first point not on a plane formed by

13. $\omega(i-k_i+2)$, $\omega(i-k_i+1)$, and $\omega(i-k_i)$

14. let $h_1 = (\omega(i-1) - \omega(i-l_{ki}+1)) \cdot (\omega(i-1) - \omega(i-2))$

15. let $h_2 = (\omega(i-k_i+1) - \omega(i-l_{ki}+1)) \cdot (\omega((i-k_i+1) - \omega((i-k_i))$

16. let A be defined by $\omega(i-l_{ki}+1) = \omega(A(i-l_{ki}))$

17. if $d_i = (i, i-l_{ki}, \sqrt{h_1^2+h_2^2})$ then $\omega(i) = \omega(A^{-1}(i-1))$

18. else if $d_i = (i, i-l_{ki}, \sqrt{h_1^2+h_2^2+4})$ then $\omega(i) = \omega(A(i-1))$

19. endif

20. endif

21. done

end

The scaling of the above algorithms is given by the two following theorems.

THEOREM 3.1. *Given D a W/SAW-canonical distance list the following is true.*

(i) The algorithm BUILD-WALK-DIAMOND(D, W) constructs a $|D|$-constrained n-

step walk, W, on the diamond lattice corresponding to the distance list D.

(ii) BUILD-WALK-DIAMOND runs in O(n) steps.

(iii) All walks on the diamond lattice corresponding to D are isomorphic to the walk generated by BUILD-WALK-DIAMOND.

(iv) If D is SAW-canonical then the walk produced by BUILD-WALK-DIAMOND is a SAW.

Proof. (i). W is a walk since for all i we have $|\omega(i) - \omega(i-1)| = 1$. It is obvious that W is composed of n-step and is subjected to $|D|$ distance constraints. W matches the distance list D since the if statements are derived from the proof of Theorem 2.11. (ii). The loop starting at line 2 comprises $n-2$ steps. (iii). W is constructed from D, which is W-canonical, consequently all walks matching D are isomorphic to W. (iv). If D is SAW-canonical then D was generated from a SAW W'. Since D is canonical W is isomorphic to W' and W is a SAW. □

THEOREM 3.2. *Given D a W/SAW-canonical distance list the following is true.*

(i) The algorithm BUILD-WALK-CUBIC(D,W) constructs a (n-2)-constrained n-step walk, W, on the cubic lattice corresponding to the distance list D.

(ii) BUILD-WALK-CUBIC runs in O(n) steps.

(iii) All walks on the diamond lattice corresponding to D are isomorphic to the walk generated by BUILD-WALK-CUBIC.

(iv) If D is SAW-canonical then the walk produced by BUILD-WALK-CUBIC is a SAW.

Proof. The proof follows exactly the same way as the previous proof using instead Theorem 2.13. □

4. Compiling canonical distance lists with a minimum number of experiments. In this section we are interested to compute canonical distance lists for protein structures. Since proteins are 3D objects, we will restrict ourself to SAW on

the diamond and cubic lattices. The Euclidean coordinates of the probed SAW are unknown but we assume that the walk exists and that we have an experimental apparatus that can measure the distance between any pair of steps on the walk. In other words, our protein structure is unresolved but we have experimental technique that can measure the distance between any pair of amino acids in the protein sequence. Techniques such as NMR and MS3D can provide such information. The algorithms given next computes the canonical distance list of a SAW with O(n) distance measurements.

```
COMPILE-DISTANCE-DIAMOND(W,D)
input:  -W:  n-step walk (Euclidean coordinates unknown)
output: -D:  n-distance list
        -W:  n-step walk (Euclidean coordinates computed)
local:  -i,k_i:  integer
        -MEASURE(W,i,j):  function returning the Euclidean
        distance between sites i and j of W
begin
1.  D = ∅; ω(0) = (0,0,0); ω(1) = ω(U(0));
2.  for i = 3 to n do
3.      δ = MEASURE(W,i,i-3);
4.      if δ = √(19/3) then D = D ∪ (i,i-3,δ);
5.      else k_i = 0;
6.          find k_i > 3 s.t.  i-k_i ≥ 0 and (i-k_i+3,i-k_i,√(19/3)) ∉ D;
7.          if k_i = 0 then goto 11 endif;
8.          δ = MEASURE(W,i,i-k_i);
9.          D = D ∪ (i,i-k_i,δ);
10.     endif
11.     ω(i) = COMPUTE-COORD-DIAMOND(D,W,i);
```

12. done

end

COMPILE-DISTANCE-CUBIC(W,D)

input: -W: n-step walk (Euclidean coordinates unknown)

output: -D: n-distance list

-W: n-step walk (Euclidean coordinates computed)

local: -i, k_i, l_{ki}: integer

-MEASURE(W, i, j): function returning the Euclidean distance between sites i and j of W

begin

1. $D = \emptyset$; $\omega(0) = (0,0,0)$; $\omega(1) = \omega(U(0))$;
2. for $i = 2$ to n do
3. δ = MEASURE($W, i, i-2$);
4. if $\delta = 2$ then $D = D \cup (i, i-2, \delta)$;
5. else $k_i = 0$;
6. find $k_i > 2$ s.t. $i - k_i \geq 0$ and $(i - k_i + 2, i - k_i, 2) \notin D$;
7. if $k_i = 0$ then goto 18 endif;
8. δ = MEASURE($W, i, i - k_i$);
9. if $\delta = k_i - 2$ or $\delta = \sqrt{k_i^2 - 4k_i + 8}$ then
10. $D = D \cup (i, i - k_i, \delta)$;
11. else $l_{ki} = 0$;
12. find $l_{ki} > 2$ s.t. $i - l_{ki} \geq 0$ and $k_i - l_{ki} \geq 0$ and $((\omega(i - k_i + 2) - \omega(i - k_i + 1)) \times (\omega(i - k_i) - \omega(i - k_i + 1)))$ $\cdot (\omega(i - l_{ki} + 1) - \omega(i - l_{ki})) = \pm 1$.
13. if $l_{ki} = 0$ then goto 18 endif;

```
14.            δ = MEASURE(W,i,i − l_ki);
15.            D = D ∪ (i,i − l_ki,δ);
16.         endif
17.      endif
18.      ω(i) = COMPUTE-COORD-CUBIC(D,W,i);
19. done
end
```

THEOREM 4.1. *Given W a SAW on the diamond lattice, the following is true.*
(i) The algorithm COMPILE-DISTANCE-DIAMOND(W,D) computes a SAW-canonical distance list for W.
(ii) The procedure MEASURE is called at most O(n) times.

Proof. (i) Proof is obvious from Corollary 2.12.
(ii) At most two measurements are carried for every i, $3 \leq i \leq n$. □

THEOREM 4.2. *Given W a SAW on the cubic lattice, the following is true.*
(i) The algorithm COMPILE-DISTANCE-CUBIC(W,D) computes a SAW-canonical distance list for W.
(ii) The procedure MEASURE is called at most O(n) times.

Proof. (i) Proof is obvious from Corollary 2.14.
(ii) At most three measurements are carried for every i, $2 \leq i \leq n$. □

5. Extension to Real Space Protein Folding Simulations. The lattice model of the protein gives us a straightforward means of producing structures conforming to a set of given constraints using established techniques of understood com-

plexity. Unfortunately, the actual protein structure exists in real space, with features that may not be adaptable to a lattice treatment. However, the same general SAW generation technique may be applicable to real space protein generation.

The idea behind SAW generation of these real space structures is that there are limitations on the different ways that individual amino acids can connect to form chains. The physically reasonable conformational space available for adding an amino acid to an existing chain can be analyzed and then broken up into discrete branching possibilities much like that of the lattice based SAW. In this case one does not have a fixed lattice but instead, the possible positions of the next amino acid. These positions depend on the most probable conformations, which in turn are determined by the local environment. Because of the limited number of amino acids, a conformational library for each of the 20 amino acids can be precomputed de novo using classical molecular dynamics and Monte Carlo techniques, or statistically, from the set of solved protein structures [Daune 1999]. Once the size of the conformational space for each amino acid is known, a decision can be made regarding which (and how many) conformations will be attempted for the next step in the SAW. Generally, there is enough global and local constraint information such that one can explore the conformational space with a reasonably small number of possibilities. Although the lattice complexity results are not rigorously extendible to the real space, many of the constraint results do apply, and we do not expect the computational complexity results to be dramatically different.

There are additional issues to consider in the real space protein folding problem. The primary one is how many constraints are generally needed to uniquely define the walk. More generally, one could ask how many constraints are needed to have a small number of walks, or perhaps $O(n)$ walks, where n is the number of peptides. A related question is what size constraints are generally the most effective. Is it better to have lots of constraints for peptides that are close to each other in space, or far, or perhaps a mix? One can also ask the same things about how close the constrained

peptides should be to one another in the peptide sequence. Finally, one must consider the effect of the uncertainty which will always be present in experimentally derived constraints.

6. Conclusion. We have demonstrated that given a lattice model of a protein containing n sites (analogous to peptides), the complete structure of the protein can be determined given $O(n)$ distance constraints for the most commonly used lattices in 2 and 3 dimensions. Furthermore, we have shown that reconstruction can be accomplished in linear time, and the canonical distance list can be compiled with a linear number of distance measurements. Future work will be directed at exploring similar questions in real space.

Acknowledgments. This work was supported by the Mathematic Information and Computer Science Program of the U.S. Department of Energy. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contrcat DE-AC04-94AL85000.

REFERENCES

[Crippen 1988] Crippen, G. M., and Havel, T. F. 1988. *Distance Geometry and Molecular Conformation*, John Wiley & Sons.

[Daune 1999] Daune, M. 1999. *Molecular Biophysics: Structures in Motion*, Oxford University Press, Oxford.

[Dill 1996] Dill, K. A. 1996. Theory for the folding and stability of globular proteins, *Biochemistry*, 24, 1501.

[Duan 1998] Duan, Y., and Kollman P.A. 1998. Pathways to a Protein Folding Intermediate Observed in a 1-microsecond Simulation in Aqueous Solution. *Science* 282, 740-744.

[Fraenkel 1993] Fraenkel, A. S. 1993. Complexity of protein folding, *Bull. Math. Bio.* 55, 1199-1210.

[Hart 1997] For NP-hardness results in $\mathbb{Z}$ cf. Hart, W. E., and Istrail, S. 1997. Robust Proofs of NP-Hardness for Protein Folding: General Lattices and Energy Potentials, *J. Computational Biology*, 4, 1-20.

[Havel 1979] Havel, T. F., Crippen, G. M., and Kuntz, I. D. 1979. *Biopolymer*, 18, 73.

[Hendrickson 1992] Hendrickson, B. A. 1992. Conditions for unique graph realizations, *SIAM J. Comput.*, 21, 65-84.

[Machta 1992] Machta, J. 1992. The computational complexity of self-avoiding walk on random lattices, *J. Phys. A: Math. Gen.*, 25, 321-327.

[Madras 1993] Madras, N., and Slade, G. 1993. *The Self-avoiding Walk*, Birkhauser, Boston.

[Moré 1995] For ϵ approximation in $\mathbb{R}$ cf. Moré, J. J., and Wu, Z. 1995. ϵ-optimal solutions to distance geometry problems via global continuation, 151-168. In Pardalos, P. M., Shalloway, D., and Xue, G., eds., *in Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, American Mathematical Society.

[Saxe 1979] For global optimization in $\mathbb{R}$ cf. Saxe, J. B. 1979. Embeddability of weighted graphs in k-space is strongly NP-hard, *Proc. 17th Allerton Conference in Communications, Control and Computing*, pp. 818-821.

[Valiant 1981] Valiant, L. G. 1981. Universality Considerations in VLSI Circuits, *IEEE Trans. Comput.*, C30, 135-140.

[Young 2000] Young, M. M. *et al.* 2000. High-Throughput Protein Fold Identification by Using Experimental Constraints Derived from Intramolecular Crosslinks and Mass Spectrometry, *PNAS* 97, 5802-5806.

Figure Captions

Figure 1: Arbitrary embedding of 2D honeycomb lattice in Euclidian spaces.

Figure 2: Arbitrary embedding of 2D square lattice in Euclidian spaces.

Figure 3: Arbitrary embedding of 3D diamond lattice in Euclidian spaces.

Figure 4: Arbitrary embedding of 3D cubic lattice in Euclidian spaces.

Figure 5: Distances in honeycomb lattice.

a) $|\omega(R(i-1)) - \omega(i-2)| = 0$,

b) $|\omega(U(i-1)) - \omega(i-k_i)| = 0$ and $|\omega(L(i-1)) - \omega(i-k_i)| = \sqrt{3}$.

c) $|\omega(U(i-1)) - \omega(i-k_i)| = 2$ and $|\omega(L(i-1)) - \omega(i-k_i)| = \sqrt{7}$.

Figure 6: Distances in square lattice.

a) $|\omega(D(i-1)) - \omega(i-2)| = 0$, and $|\omega(U(i-1)) - \omega(i-2)| = 2$.

b) $h = |\omega(i-1) - \omega(i-k_i+1)|$. $|\omega(D(i-1)) - \omega(i-k_i)| = h$, and $|\omega(U(i-1)) - \omega(i-k_i)| = \sqrt{h^2+4}$.

Figure 7: Distances in diamond lattice.

a) $|\omega(D(i-1)) - \omega(i-2)| = 0$, and $|\omega(U(i-1)) - \omega(i-3)| = \sqrt{19/3}$.

b) $h = |\omega(i-1) - \omega(i-k_i+1)|$. $|\omega(L(i-1)) - \omega(i-k_i)| = h$ if $\omega(i-k_i) = \omega(L(i-k_i))$, and $|\omega(L(i-1)) - \omega(i-k_i)| = \sqrt{h^2+8/3}$ otherwise.

$|\omega(R(i-1)) - \omega(i-k_i)| = h$ if $\omega(i-k_i) = \omega(R(i-k_i))$, and $|\omega(R(i-1)) - \omega(i-k_i)| = \sqrt{h^2+8/3}$ otherwise.

c) $h = |\omega(i-1) - \omega(i-k_i+1)|$. $|\omega(L(i-1)) - \omega(i-k_i)| = \sqrt{h^2+8/3}$ if $\omega(i-k_i) = \omega(L(i-k_i))$, and $|\omega(L(i-1)) - \omega(i-k_i)| = \sqrt{h^2+16/3}$ otherwise.

$|\omega(R(i-1)) - \omega(i-k_i)| = \sqrt{h^2+8/3}$ if $\omega(i-k_i) = \omega(R(i-k_i))$, and

$|\omega(R(i-1)) - \omega(i-k_i)| = \sqrt{h^2 + 16/3}$ otherwise.

Figure 8: Distances in cubic lattice.

a) $|\omega(D(i-1)) - \omega(i-2)| = 0$, and $|\omega(U(i-1)) - \omega(i-2)| = 2$.

b) $h_0 = |\omega(i-1) - \omega(i-k_i+1)|$. $|\omega(F(i-1)) - \omega(i-k_i)| = h_0$, and $|\omega(B(i-1)) - \omega(i-k_i)| = \sqrt{h_0^2 + 4}$.

c) $h_1 = (\omega(i-1) - \omega(i-li)) \cdot (\omega(i-1) - \omega(i-2))$,

$h_2 = (\omega(i-k_i+1) - \omega(i-li)) \cdot (\omega((i-k_i+1) - \omega((i-k_i))$,

$|\omega(L(i-1)) - \omega(i-l_i)| = \sqrt{h_1^2 + h_2^2}$, and $|\omega(R(i-1)) - \omega(i-l_i)| = \sqrt{h_1^2 + h_2^2 + 4}$.

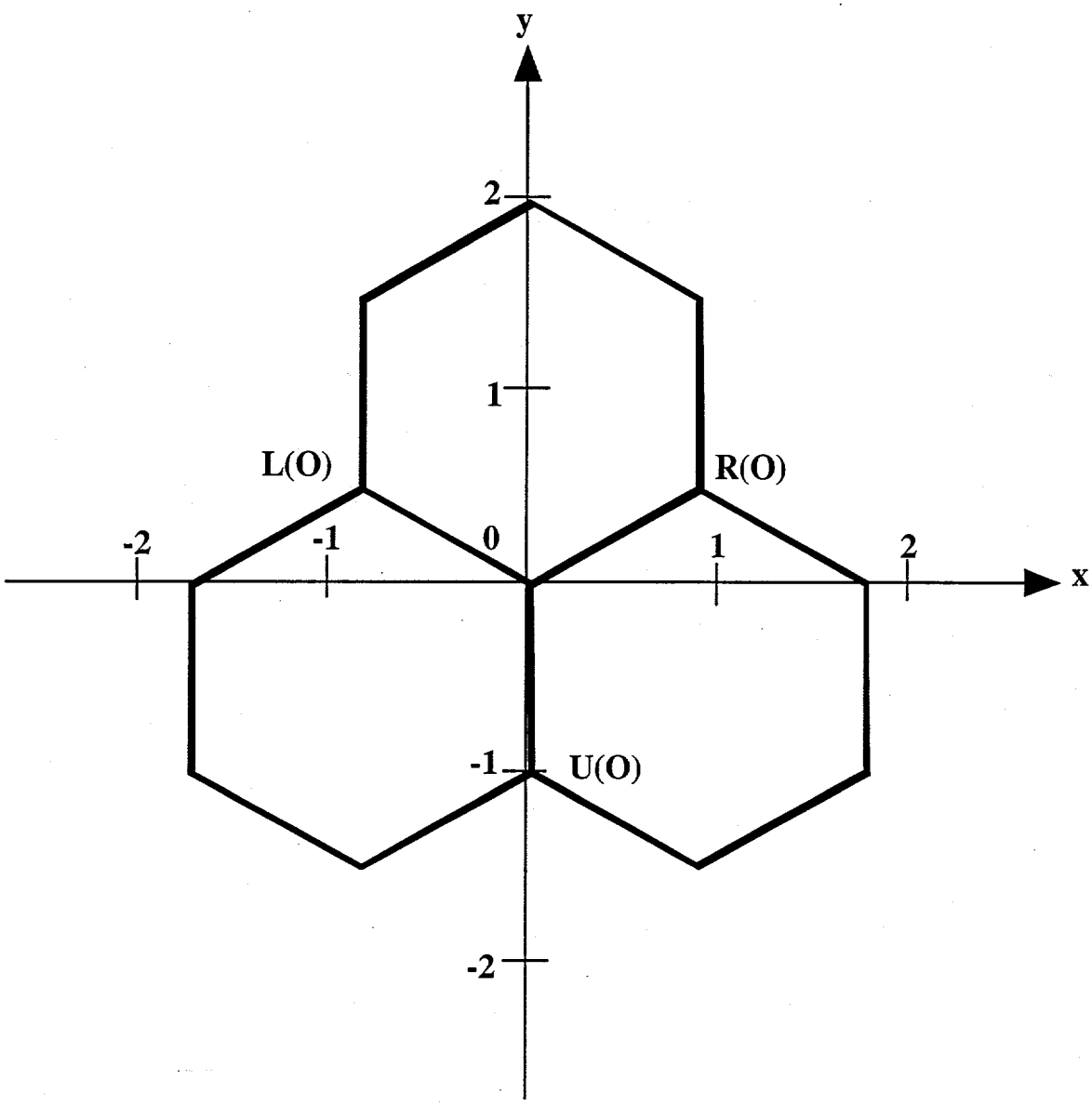


Figure 1

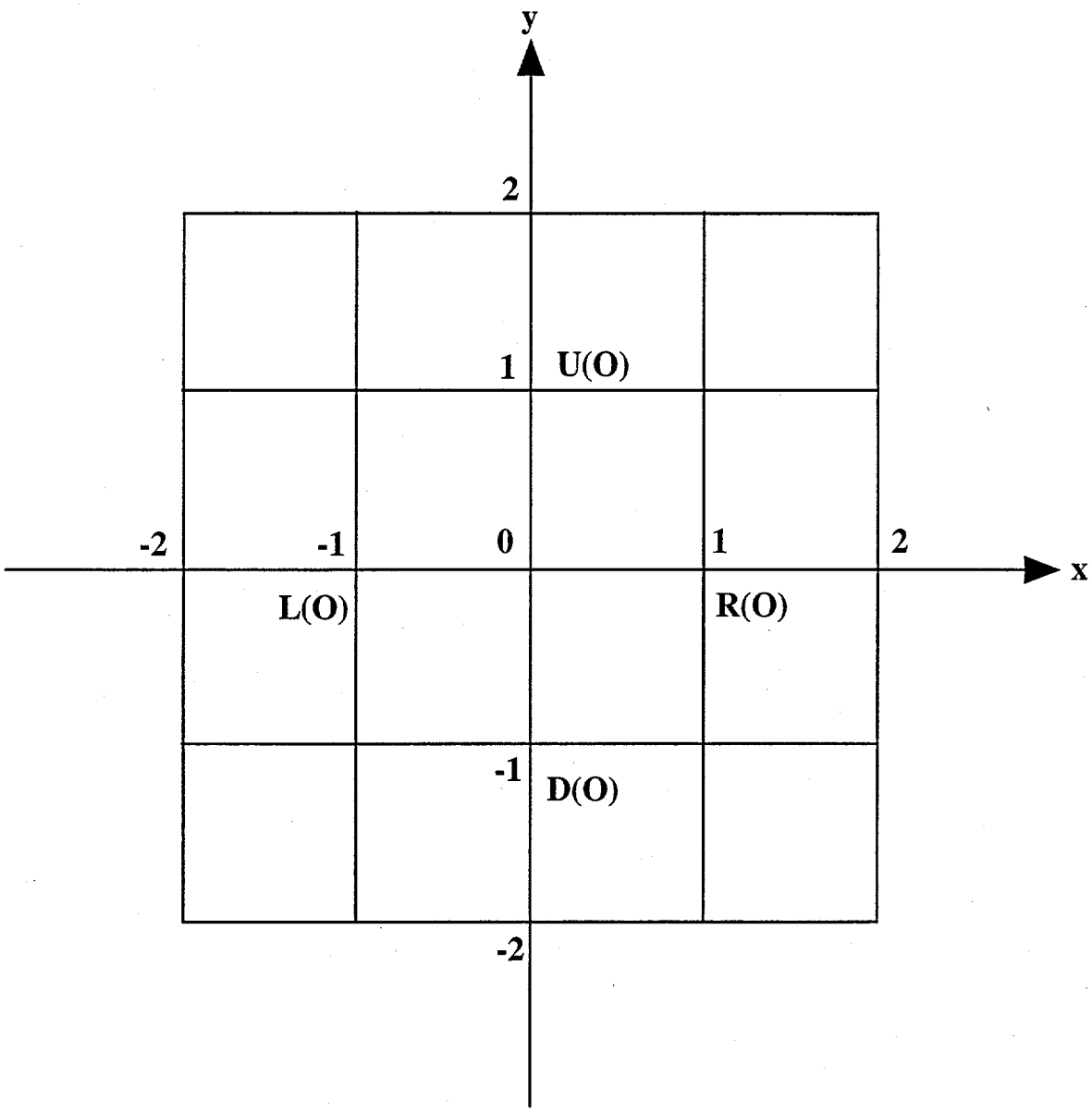


Figure 2

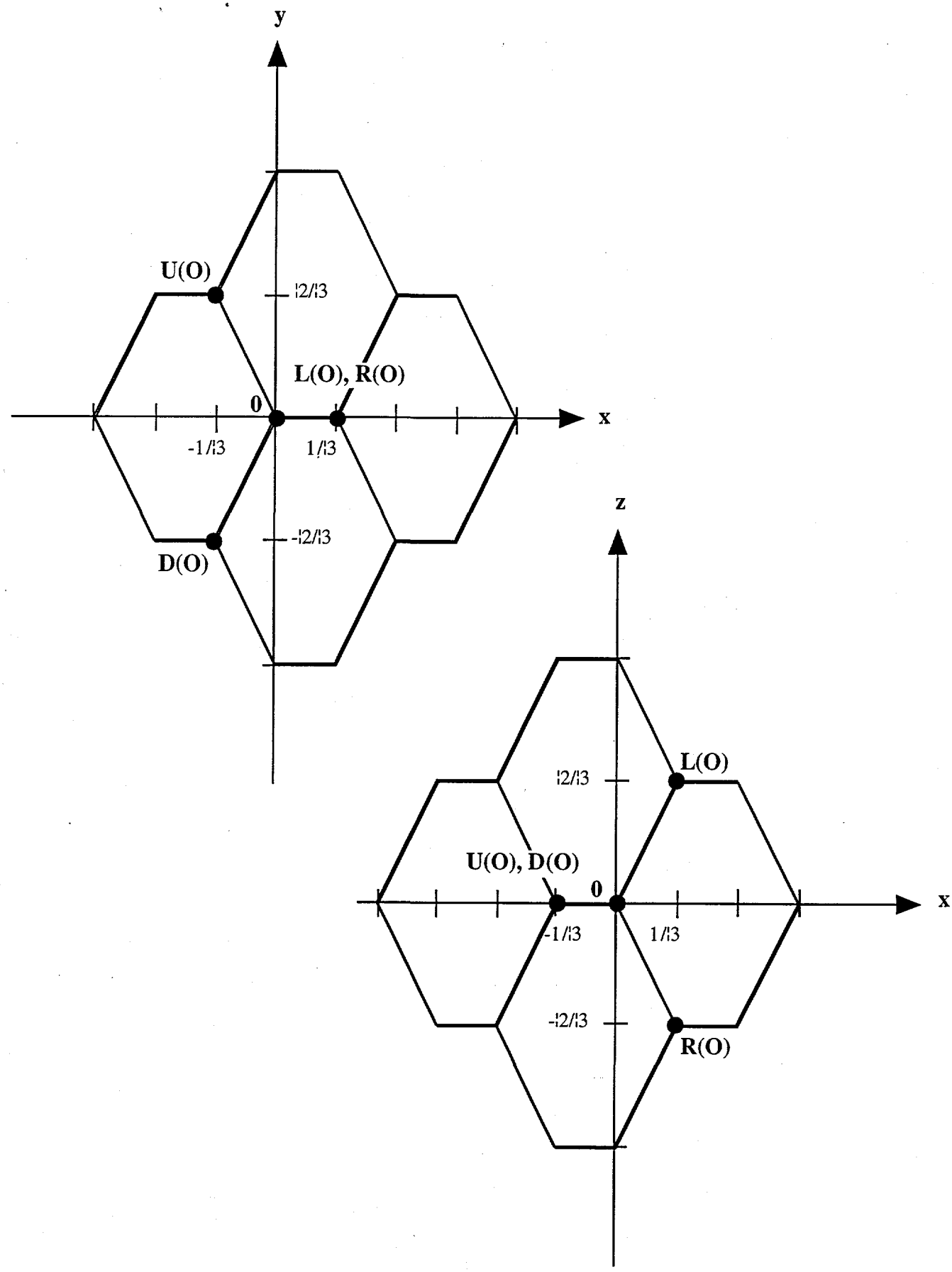


Figure 3

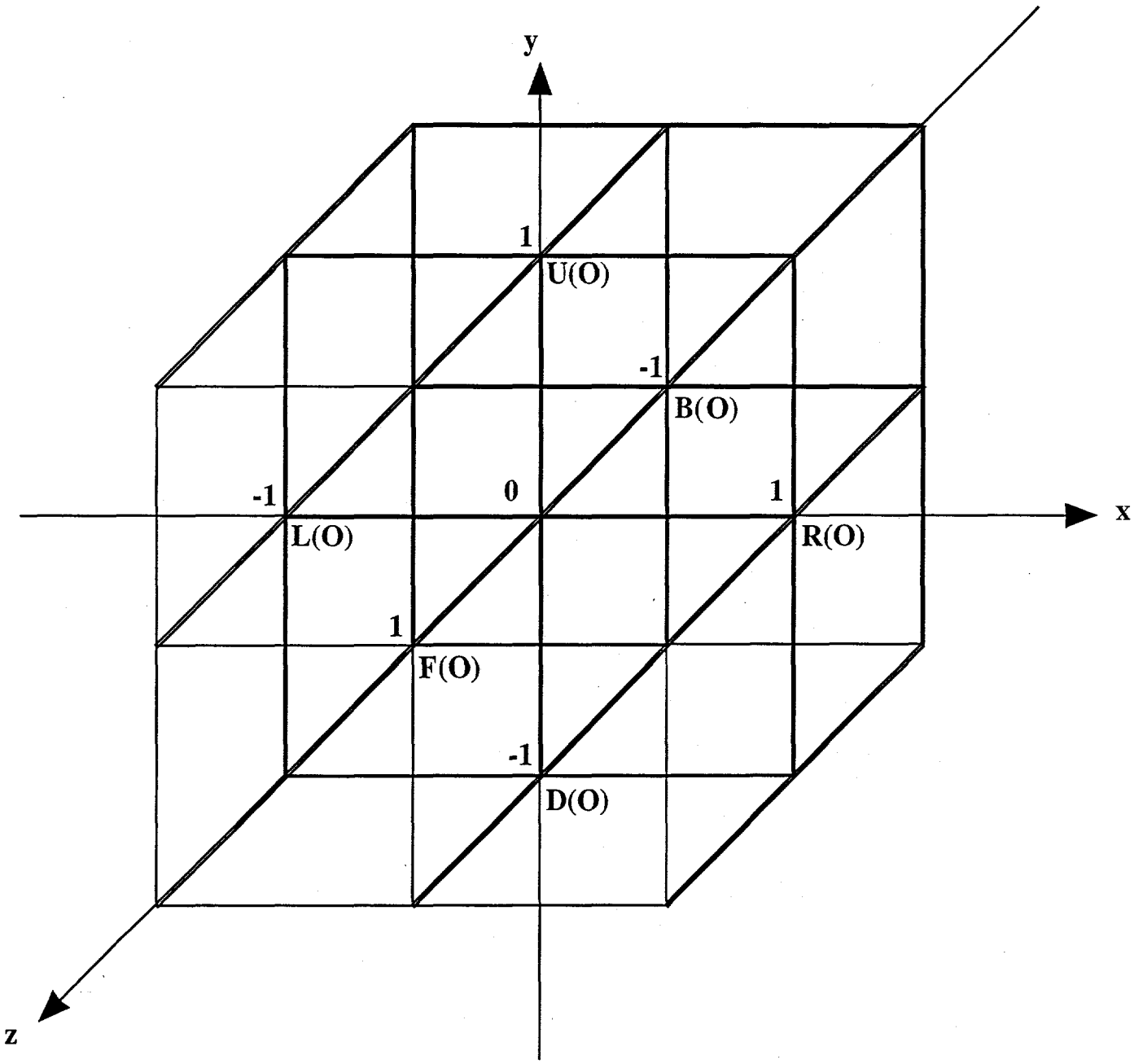


Figure 4

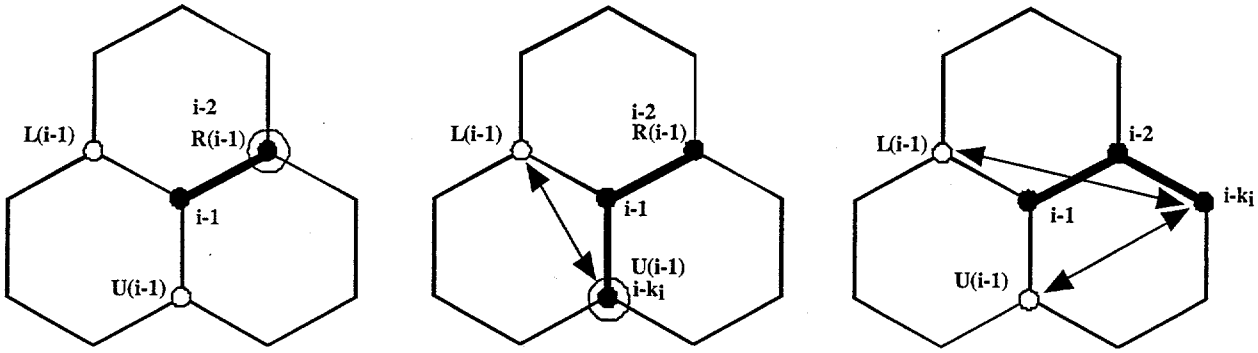
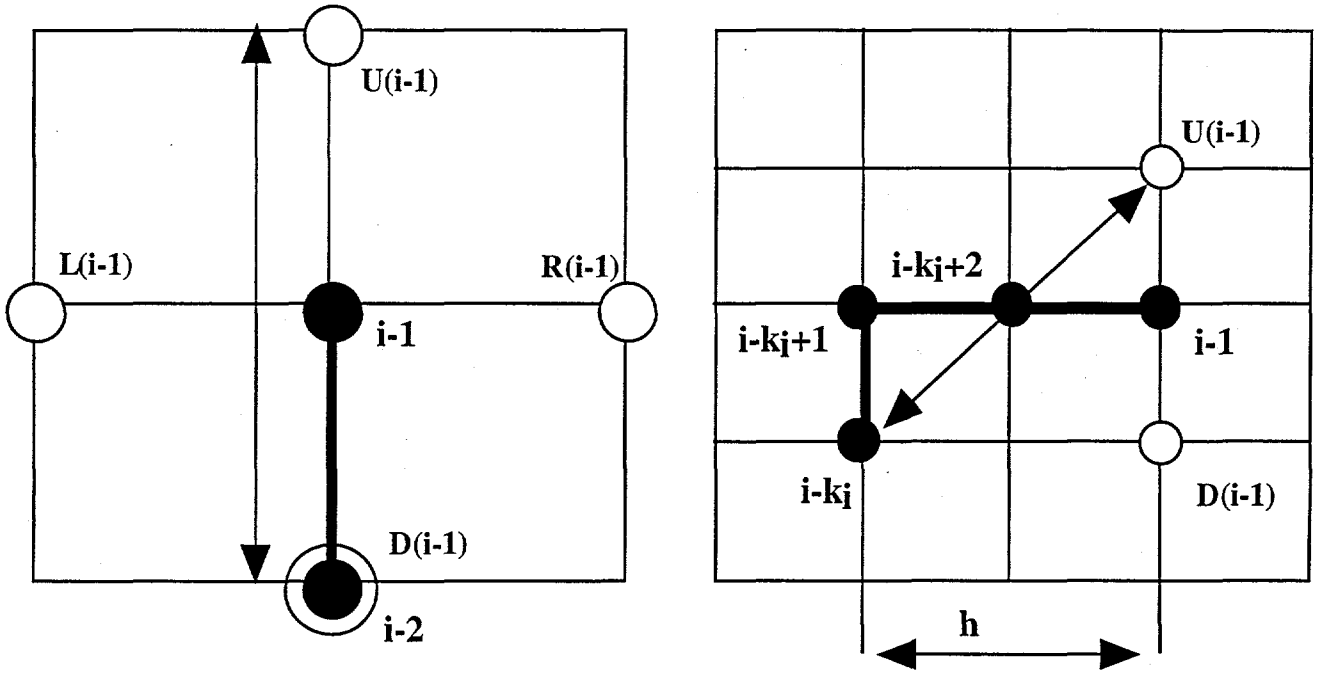


Figure 5



a)

b)

Figure 6

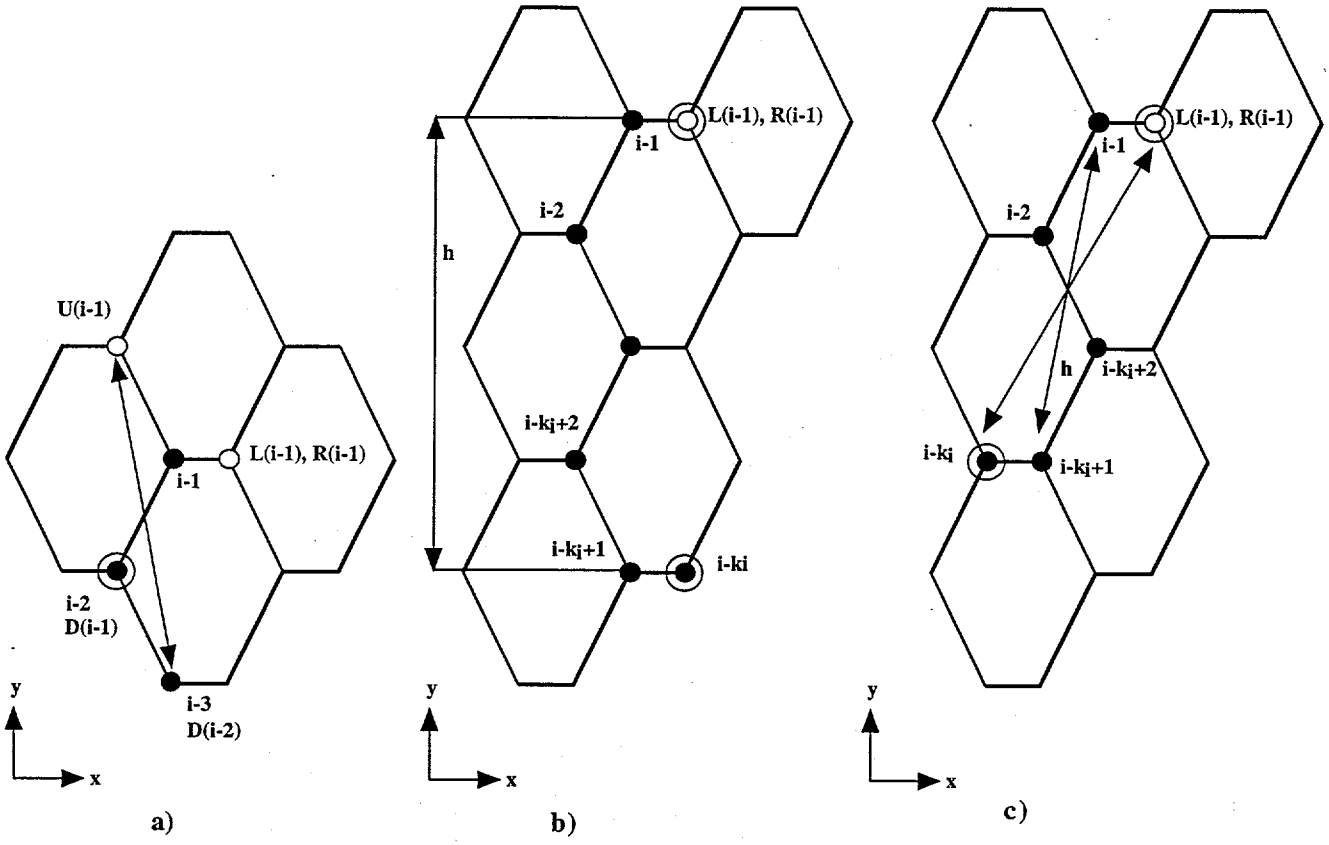
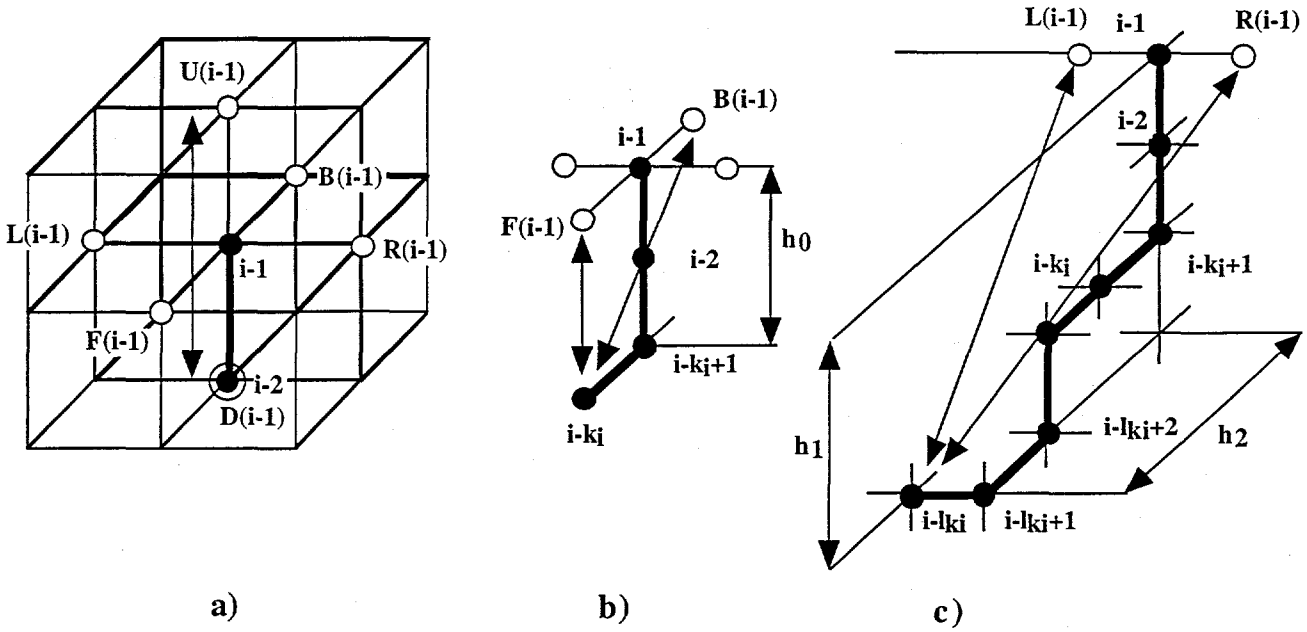

U(i-1)
L(i-1), R(i-1)
i-1
i-2
D(i-1)
i-3
D(i-2)
y
x
a)
L(i-1), R(i-1)
i-1
i-2
h
i-kj+2
i-kj+1
i-ki
y
x
b)
L(i-1), R(i-1)
i-1
i-2
h
i-kj+2
i-kj
i-kj+1
y
x
c)

Figure 7


U(i-1)
B(i-1)
L(i-1)
i-1
R(i-1)
F(i-1)
i-2
D(i-1)
B(i-1)
i-1
F(i-1)
i-2
h0
i-kj+1
i-kj
L(i-1)
i-1
R(i-1)
i-2
i-kj
i-kj+1
h1
i-lki+2
h2
i-lki
i-lki+1

a) b) c)

Figure 8