# Теорія оптимальних рішень

УДК 519.21

В.А. РУДЫК

## ПРЕДСТАВЛЕНИЕ СТРУКТУРЫ БЕЛКА В ТРЕХМЕРНЫХ ДИСКРЕТНЫХ РЕШЕТКАХ ПРОИЗВОЛЬНОГО ТИПА

*Предлагается и исследуется подход к формализации понятий абсолютной и относительной кодировок, которые применяются при решении известной в вычислительной биологии задачи прогнозирования третичной структуры протеинов. Рассмотрены трехмерные решетки общего вида. Сформулированы и доказаны основные свойства этих кодировок, предлагаются конструктивные алгоритмы их построения.*



**Введение.** Одной из самых исследуемых в последнее время задач комбинаторной оптимизации в вычислительной биологии является задача прогнозирования третичной структуры молекулы белка. Ее суть состоит в том, чтобы исходя из линейной последовательности элементов, составляющих молекулу, определить ее пространственную форму. Решением задачи является некоторая неразрывная трехмерная кривая без самопересечений. Большинство моделей структуры белка дискретны, поскольку форма молекулы представляется как путь в некоторой дискретной решетке. Чаще всего для упрощения модели рассматриваются двумерные или трехмерные кубические решетки [1-4], хотя в контексте поставленной проблемы они обладают рядом недостатков. При переходе к более сложным решеткам возникает необходимость закодировать путь в виде некоторого математического объекта, максимально отображающего его характеристики, среди этих объектов и будет проходить поиск с помощью различных алгоритмов. В работе предлагается два подхода к кодировке в трехмерных решетках, обладающих различными свойствами с точки зрения алгоритмов их построения. Предложенные кодировки могут быть использованы не только для моделирования и прогнозирования структуры белковых молекул, но и в других задачах, где необходимо исследовать пространство некоторых трехмерных кривых в дискретной решетке.

**1. Кодирование в дискретной решетке.** Рассмотрим дискретную решетку $L = \{\sum_{i=1}^{3} k_i e_i : e_i \in \mathbb{R}^3, k_i \in \mathbb{Z}\}$ ($e_1, e_2, e_3$ линейно независимы, $\mathbb{R}$ и $\mathbb{Z}$ – множества действительных и целых чисел соответственно) в трехмерном евклидовом пространстве. Пусть на ней определено отношение соседства $R$, для которого выполняется следующее свойство:

$$c_1, c_2 \in L,\ (c_1, c_2) \in R \Rightarrow (c, c + (c_2 - c_1)) \in R\ \ \forall c \in L.$$

**Определение 1.** Множество $V = \{v_1, ..., v_s\},\ v_i \in L$ называется множеством векторов соседства решетки $L$ с отношением соседства $R$, если $v \in V \Leftrightarrow \exists c : (c, c + v) \in R,\ v, c \in L$.

Будем считать, что система векторов $V$ полная, т. е. любой вектор трехмерного пространства можно представить как линейную комбинацию векторов из множества $V$; если это не так, рассматриваемая проблема сводится к двумерному случаю.

Можно показать, что если для отношения $R$ выполняется вышеуказанное свойство, то $v \in V \Leftrightarrow \forall c : (c, c + v) \in R,\ v, c \in L$.

Назовем решетку ориентированной, если $\exists v \in V : -v \notin V$, и неориентированной, если $v \in V \Rightarrow -v \in V$.

Из определения следует, что в неориентированной решетке отношение соседства симметрично.

Будем понимать под путем длиной $n\ (n > 1)$ в дискретной решетке последовательность соседних $n$ узлов: $c_{(n)} = c_1 c_2 ... c_n : (c_i, c_{i+1}) \in R,\ i = \overline{1, n-1}$. Обозначим $F$ множество всех путей в решетке, а $F_n$ – множество всех путей длиной $n$.

Два пути $c = c_1 c_2 ... c_n \in F_n$ и $\tilde{c} = \tilde{c}_1 \tilde{c}_2 ... \tilde{c}_n \in F_n$ эквивалентны ($c \sim \tilde{c}$), если существует изометрическое преобразование $\phi : L \to L$, такое что выполняется условие: $\phi(c_i) = \tilde{c}_i,\ i = \overline{1, n}$.

Пути, равные с точностью до параллельного переноса, будем обозначать $c \simeq \tilde{c}$. Таким образом, $c \simeq \tilde{c} \Rightarrow c \sim \tilde{c}$.

Чтобы представить путь в решетке в виде математического объекта для использования, например, в алгоритмах комбинаторной оптимизации, введем функцию кодировки.

**Определение 2.** Функция $Enc : F \to \tilde{V},\ \tilde{V} = \bigcup_{i=0}^{\infty} V^i$ – функция кодировки, если она обладает следующими свойствами:

1. $Enc(c) = a,\ c \in F_n \Rightarrow a \in V^{n-1}$.

2. Существует обратная функция декодирования $Dec : \tilde{V} \to F$, такая что $\forall c \in F\ \ Dec(Enc(c)) \sim c$.

3. $\forall c \in F_n \ \exists \tilde{c} \in F_n : Enc(c) = Enc(\tilde{c}),\ Dec(Enc(\tilde{c})) = \tilde{c}$.

4. $Enc(c_1c_2...c_n) = a_1a_2...a_{n-1} \Rightarrow Dec(a_ia_{i+1}...a_j) \sim c_ic_{i+1}...c_{j+1} \ \ \forall i, j : 1 \le i < j \le n-1$.

5. $Enc(c_1c_2...c_n) = a_1a_2...a_{n-1} \Leftrightarrow Enc(c_1c_2...c_{n-1}) = a_1a_2...a_{n-2}$.

Первые три условия формируют соответствия между путями в решетке и их кодовыми представлениями, а из двух последних вытекает, что "похожие" пути кодируются "похожими" последовательностями и наоборот.

**Определение 3.** Функцией абсолютной кодировки $aEnc$ называется такая функция кодировки в дискретной решетке, для которой из условия $aEnc(c) = aEnc(\tilde{c})$ $(c, \tilde{c} \in F)$ следует, что $c \simeq \tilde{c}$.

Функцию декодирования функции $aEnc$ будем обозначать $aDec$.

В ряде приложений, использующих абсолютную кодировку, полезно следующее свойство.

**Теорема 1.** Если $aEnc$ – функция абсолютного кодирования, то $aEnc(aDec(a)) = a$ $(a = a_1a_2...a_{n-1},\ a_i \in V)$.

*Доказательство***.** Разобьем множество всех путей длиной $n$ на классы эквивалентности по отношению $" \simeq "$. Таких классов будет $|V|^{n-1}$. Из условия 2 определения кодировки и определения абсолютной кодировки, в каждом таком классе существует элемент $\tilde{c}$, такой, что $aDec(aEnc(\tilde{c})) = \tilde{c}$. Отсюда $aEnc(aDec(aEnc(\tilde{c}))) = aEnc(\tilde{c})$, и, если обозначить $\tilde{a} = aEnc(\tilde{c})$, получим $aEnc(aDec(\tilde{a})) = \tilde{a}$. Из определения абсолютной кодировки, если не выполняется условие $c \simeq \tilde{c}$, то $aEnc(c) \ne aEnc(\tilde{c})$. Таким образом доказано, что существует $|V|^{n-1}$ кодов, для которых выполняется лемма. Можно посчитать, что число всех возможных кодов путей длиной $n$ – $|V|^{n-1}$, значит, лемма выполняется для каждого кода. Теорему доказано.

Часто используемым примером функции абсолютного кодирования является функция $\overline{aEnc}(c_1c_2...c_n) = a_1a_2...a_{n-1}$, где $a_k = c_{k+1} - c_k$, $k = \overline{1, n-1}$ [5].

Рассмотрим решетки, для которых выполняется следующее условие: для произвольных $v, v' \in V$ существует функция поворота (здесь и далее подразумевается поворот вокруг центра координат) $\varphi : V \to V$, такая что $\varphi(v) = v'$. Одним из свойств таких решеток есть равенство длин всех векторов соседства. Примерами являются кубическая и трехмерная треугольная решетки (рисунок) [6–9].

**Определение 4.** Назовем функцию кодировки $rEnc$ функцией относительной кодировки, если для нее выполняется условие

$$\forall c, \tilde{c} \in F : c \sim \tilde{c} \Rightarrow rEnc(c) = rEnc(\tilde{c}).$$

Обозначим $rDec$ функцию декодирования $rEnc$.

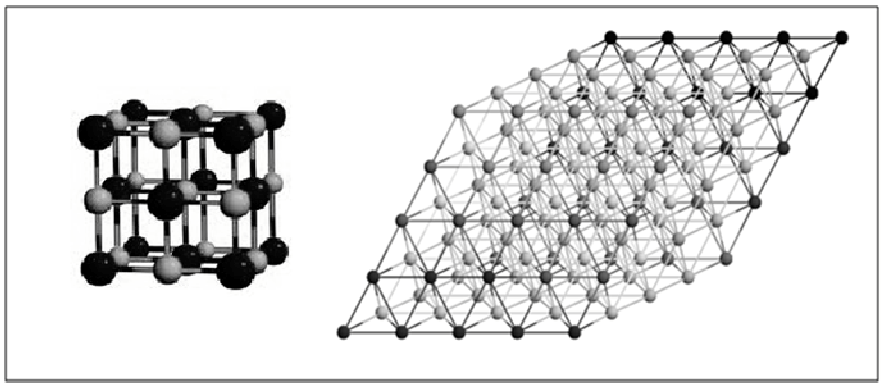

РИСУНОК. Кубическая и трехмерная треугольная решетки

Приведем пример построения относительной кодировки для неориентированных решеток. Выберем произвольный вектор $v' \in V$. Пусть $\Psi[r] = \{\varphi : V \to V \ (r \in V) \mid \varphi(r) = v'\}$. В трехмерном пространстве для фиксированных $r, v' \in V$ такое множество содержит больше одного элемента, поскольку поворот пространства однозначно не определяется поворотом одного вектора. Определим функцию $\varphi[r_1, r_2] : V \to V$, $(r_1, r_2 \in V,\ r_1 \neq r_2,\ r_1 \neq -r_2)$ как функцию поворота, удовлетворяющую следующим условиям:

1) $\varphi[r_1, r_2](r_2) = v'$;

2) $\varphi[r_1, r_2](r_1) = v_j$, где $j = \min\limits_{\exists \varphi \in \Psi[r_2] : \varphi(r_1) = v_i} i$.

Для $r_1 \neq r_2,\ r_1 \neq -r_2$ такая функция единственная и однозначно задает поворот в трехмерном пространстве. Можно построить аналитическую непрерывную функцию $\tilde{\varphi}[r_1, r_2] : \mathbb{R}^3 \to \mathbb{R}^3$, равную функции $\varphi[r_1, r_2]$ во всех точках множества $V$, но, учитывая, что $\varphi[r_1, r_2]$ определяется на конечном множестве точек, удобнее использовать алгоритм на основе перебора. Можно показать, что сложность алгоритма – $o(|V|^4)$, что приемлемо, поскольку процедура построения $\varphi[r_1, r_2]$ выполняется для фиксированной решетки один раз.

Алгоритм основывается на двух свойствах функции поворота.

1) Обозначим $d(a_1, a_2)$ $(a_1, a_2 \in V)$ евклидово расстояние между векторами $a_1$ и $a_2$. $d(a_1, a_2) = d(\varphi[r_1, r_2](a_1), \varphi[r_1, r_2](a_2))$.

2) Пусть функция $rightside(a_1, a_2, a_3)$ определяет, правосторонняя ли тройка векторов $(a_1, a_2, a_3 \in V)$. Тогда

$$rightside(a_1, a_2, a_3) = rightside(\varphi[r_1, r_2](a_1), \varphi[r_1, r_2](a_2), \varphi[r_1, r_2](a_3)) \ .$$

В алгоритме происходит перебор всех возможных поворотов, которые переводят множество $V$ в себя, и выбирается тот, для которого $r_2$ переходит в $v'$ и

при этом сохраняются все попарные расстояния между векторами и характеристики правосторонности троек векторов.

На основе преобразования $\varphi[r_1, r_2]$ предлагается следующий алгоритм построения относительной кодировки $\overline{rEnc}$.

**Алгоритм.** Пусть $c_1c_2...c_n$ – некоторый путь в заданной решетке, $a_1a_2...a_{n-1}=\overline{aEnc}(c_1c_2...c_n)$. Построим новую последовательность $\tilde{a}_1\tilde{a}_2...\tilde{a}_{\gamma(n-1)}$, удалив из $a_1a_2...a_{n-1}$ последовательно повторяющиеся и противоположные элементы:

$$\gamma(1) = 1;$$

$$\gamma(i) = s \Leftrightarrow (a_{\gamma(i-1)} \neq a_s \;\&\; a_{\gamma(i-1)} \neq -a_s) \;\&\; (a_{\gamma(i-1)} = a_l \vee a_{\gamma(i-1)} = -a_l \;\; \forall l \;\; \gamma(i-1) < l < s).$$

$$\tilde{a}_i = a_{\gamma(i)}.$$

Определим последовательность $\tilde{r}_1\tilde{r}_2...\tilde{r}_k$ таким образом:

$$\tilde{r}_1 = v',$$

$$\tilde{r}_2 = \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_2),$$

$$\tilde{r}_i = \varphi[\tilde{a}_{i-2}, \tilde{a}_{i-1}](\tilde{a}_i).$$

Отметим, что при $i > 1$ $\tilde{r}_i \neq v', \tilde{r}_i \neq -v'$.

На места элементов, которые были удалены из начальной кодировки $a_1a_2...a_{n-1}$, поставим элемент $v'$, получим последовательность $r_1r_2...r_{n-1}$:

$$r_1 = \tilde{r}_1,$$

$$r_i = \begin{cases} \tilde{r}_s, \text{ если } \gamma(s) = i; \\ v', \text{ если } a_{i-1} = a_i; \\ -v', \text{ если } a_{i-1} = -a_i. \end{cases}$$

То, что случаи $\exists s : \gamma(s) = i$, $a_{i-1} = a_i$ и $a_{i-1} = -a_i$ являются дополняющими друг друга, следует из определения функции $\gamma$.

Далее будет показано, что последовательность $r_1r_2...r_{n-1}$ и будет относительной кодировкой последовательности $c_1c_2...c_n$.

Опишем также алгоритм декодирования. Пусть $r_1r_2...r_{n-1}$ – исходная кодировка. Строится последовательность $\tilde{\tilde{r}}_1\tilde{\tilde{r}}_2...\tilde{\tilde{r}}_k$ путем удаления из $r_1r_2...r_{n-1}$ всех элементов, равных $v'$ и $-v'$ кроме первого:

$$\eta(1) = 1;$$

$$\eta(i) = s \Leftrightarrow (r_s \neq v' \;\&\; r_s \neq -v') \;\&\; (r_l = v' \vee r_l = -v' \;\; \forall l : \eta(i-1) < l < s)$$

$$\tilde{\tilde{r}}_i = r_{\gamma(i)}.$$

По построению видно, что последовательность $\tilde{\tilde{r}}_1\tilde{\tilde{r}}_2...\tilde{\tilde{r}}_k$ совпадает с последовательностью $\tilde{r}_1\tilde{r}_2...\tilde{r}_k$, а функция $\eta$ – с функцией $\gamma$.

Абсолютную кодировку для $\tilde{\tilde{r}}_1\tilde{\tilde{r}}_2...\tilde{\tilde{r}}_k$ определим так:

$$\tilde{\tilde{a}}_1 = v',$$
$$\tilde{\tilde{a}}_2 = \tilde{r}_2,$$
$$\tilde{\tilde{a}}_i = (\varphi[\tilde{\tilde{a}}_{i-2}, \tilde{\tilde{a}}_{i-1}])^{-1}(\tilde{r}_i).$$

Важно отметить, что при таком построении $\tilde{\tilde{a}}_i \neq \tilde{\tilde{a}}_{i-1}$ и $\tilde{\tilde{a}}_i \neq -\tilde{\tilde{a}}_{i-1}$.

Последний шаг – восстановление элементов, соответствующих всем $r_i$ из исходной последовательности, равным $v'$:

$$\hat{a}_1 = \tilde{\tilde{a}}_1,$$
$$\hat{a}_i = \begin{cases} \tilde{\tilde{a}}_s, \text{ если } \eta(s) = i; \\ \hat{a}_{i-1}, \text{ если } r_i = v'; \\ -\hat{a}_{i-1}, \text{ если } r_i = -v'. \end{cases}$$

Из определения функции $\eta$ следует, что случаи $\exists s : \eta(s) = i$, $r_i = v'$ и $r_i = -v'$ – взаимно дополняющие.

Декодировав последовательность $\hat{a}_1\hat{a}_2...\hat{a}_{n-1}$ с абсолютной кодировки, получим искомый путь в решетке: $\hat{c}_1\hat{c}_2...\hat{c}_n = \overline{aDec}(\hat{a}_1\hat{a}_2...\hat{a}_{n-1})$ .

**Теорема 2.** Определенная вышеописанным способом кодировка подходит под определение относительной.

*Доказательство*. Будет использоваться следующая лемма**.**

**Лемма 1.** $\varphi[\varphi[r_1, r_2](a_1), \varphi[r_1\, r_2](a_2)] = \varphi[a_1, a_2] \circ (\varphi[r_1, r_2])^{-1}$

*Доказательство*. Покажем, что для функции $\varphi[a_1, a_2] \circ (\varphi[r_1, r_2])^{-1}$ выполняются оба условия из определения.

Проверим первое условие:

$$\varphi[a_1, a_2]((\varphi[r_1, r_2])^{-1}(\varphi[r_1, r_2](a_2)) = \varphi[a_1, a_2](a_2) = v'.$$

Докажем, что условие 2 тоже выполняется. Из определения $\varphi[\varphi[r_1, r_2](a_1), \varphi[r_1, r_2](a_2)](\varphi[r_1, r_2](a_1)) = v_j$, где $j = \min\limits_{\exists\varphi\in\Psi[\varphi[r_1, r_2](a_2)]:\, \varphi(\varphi[r_1, r_2](a_1))=v_i} i$.

Условия $\exists\varphi \in \Psi[\varphi[r_1, r_2](a_2)] : \varphi(\varphi[r_1, r_2](a_1)) = v_i$ и $\exists\psi \in \Psi[a_2] : \psi(a_1) = v_i$ равносильны, что можно показать, выбрав $\varphi = \psi \circ (\varphi[r_1, r_2])^{-1}$ и $\psi = \varphi \circ \varphi[r_1, r_2]$. Следовательно,

$$j = \min_{\exists\varphi\in\Psi[\varphi[r_1, r_2](a_2)]:\varphi(\varphi[r_1, r_2](a_1))=v_i} i = \min_{\exists\varphi\in\Psi[a_2]:\psi(a_1)=v_i} i,$$
$$\varphi[a_1, a_2]((\varphi[r_1, r_2])^{-1}(\varphi[r_1, r_2](a_1))) = \varphi[a_1, a_2](a_1) = v_j.$$

Учитывая, что композиция поворотов тоже определяет некоторый поворот, а образы двух неколлинеарных векторов в трехмерном пространстве однозначно задают поворот, делается вывод, что

$$\varphi[\varphi[r_1, r_2](a_1), \varphi[r_1, r_2](a_2)] = \varphi[a_1, a_2] \circ (\varphi[r_1, r_2])^{-1}.$$

Лемма доказана.

Используя введенные в алгоритме обозначения приведем схему доказательства теоремы:

1. Показать, что существует поворот $\varphi: V \to V$, такой что $\varphi(\tilde{a}_i) = \tilde{\tilde{a}}_i$, $i = \overline{1, \gamma(n-1)}$.

2. Показать, что существует поворот $\varphi: V \to V$, такой что $\varphi(a_i) = \hat{a}_i$, $i = \overline{1, n-1}$.

3. Показать, что существует изометрия $\phi: L \to L$, такая что $\phi(c_i) = \hat{c}_i$ , $i = \overline{1, n-1}$.

Рассмотрим последовательно каждый этап.

Покажем, что $\varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_i) = \tilde{\tilde{a}}_i$ , $i = \overline{1, \gamma(n-1)}$ методом математической индукции. Проверим базу индукции для первых двух элементов:

$$\varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_1) = v' = \tilde{\tilde{a}}_1 \ , \ \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_2) = \tilde{r}_2 = \tilde{\tilde{a}}_2 .$$

Осуществим индукционный переход. Пусть $\varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_i) = \tilde{\tilde{a}}_i \quad \forall i \le k$. Покажем, что $\varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_{k+1}) = \tilde{\tilde{a}}_{k+1}$ . Действительно, воспользовавшись леммой 2 получим

$$\begin{aligned}
\tilde{\tilde{a}}_{k+1} &= (\varphi[\tilde{\tilde{a}}_{k-1}, \tilde{\tilde{a}}_k])^{-1}(\tilde{r}_{k+1}) = (\varphi[\tilde{\tilde{a}}_{k-1}, \tilde{\tilde{a}}_k])^{-1}(\varphi[\tilde{a}_{k-1}, \tilde{a}_k](\tilde{a}_{k+1})) = \\
&= (\varphi[\varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_{k-1}), \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_k)])^{-1}(\varphi[\tilde{a}_{k-1}, \tilde{a}_k](\tilde{a}_{k+1})) = \\
&= (\varphi[\tilde{a}_{k-1}, \tilde{a}_k] \circ (\varphi[\tilde{a}_2, \tilde{a}_1])^{-1})^{-1}(\varphi[\tilde{a}_{k-1}, \tilde{a}_k](\tilde{a}_{k+1})) = \\
&= \varphi[\tilde{a}_2, \tilde{a}_1]((\varphi[\tilde{a}_{k-1}, \tilde{a}_k])^{-1}(\varphi[\tilde{a}_{k-1}, \tilde{a}_k](\tilde{a}_{k+1})) = \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_{k+1}),
\end{aligned}$$

что и нужно было показать.

Перейдем ко второму этапу. Покажем, что $\varphi[\tilde{a}_2, \tilde{a}_1](a_i) = \hat{a}_i$ . Для первого элемента ($i = 1$) $\hat{a}_1 = \tilde{\tilde{a}}_1 = \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_1) = \varphi[\tilde{a}_2, \tilde{a}_1](a_1)$ , для каждого следующего рассмотрим три случая (пользуемся равенством последовательностей $\tilde{\tilde{r}}_1\tilde{\tilde{r}}_2...\tilde{\tilde{r}}_k$ и $\tilde{r}_1\tilde{r}_2...\tilde{r}_k$ и функций $\eta$ и $\gamma$):

1. $\exists s: i = \eta(s) \Rightarrow \hat{a}_i = \tilde{\tilde{a}}_s = \varphi[\tilde{a}_2, \tilde{a}_1](\tilde{a}_s) = \varphi[\tilde{a}_2, \tilde{a}_1](a_{\gamma(s)}) = \varphi[\tilde{a}_2, \tilde{a}_1](a_i)$ .

2. $r_i = v' \Rightarrow (a_{i-1} = a_i \ \& \ \hat{a}_i = \hat{a}_{i-1}) \Rightarrow \hat{a}_i = \hat{a}_{i-1} = \varphi[\tilde{a}_2, \tilde{a}_1](a_{i-1}) = \varphi[\tilde{a}_2, \tilde{a}_1](a_i)$.

3. $r_i = -v' \Rightarrow (a_{i-1} = -a_i \ \& \ \hat{a}_i = -\hat{a}_{i-1}) \Rightarrow \hat{a}_i = -\hat{a}_{i-1} = -\varphi[\tilde{a}_2, \tilde{a}_1](a_{i-1}) = \varphi[\tilde{a}_2, \tilde{a}_1](a_i)$.

Следующая лемма доказывает последнее, третье утверждение.

**Лемма 2.** Пусть $\overline{aEnc}(c_1c_2...c_n) = a_1a_2...a_{n-1}$ , а $\varphi: V \to V$ – некоторый поворот. Обозначим $\hat{c}_1\hat{c}_2...\hat{c}_n = \overline{aDec}(\varphi(a_1)\varphi(a_2)...\varphi(a_{n-1}))$ . Тогда существует изометрия $\phi: L \to L$ , такая что $\phi(c_i) = \hat{c}_i$ , $i = \overline{1, n-1}$.

*Доказательство.* Система векторов $V$ полная, значит, поворот $\varphi: V \to V$ – линейная функция, определенная на некотором базисе пространства. Тогда на его основе можно построить линейный оператор $\tilde{\varphi}: L \to L$, такой что $\tilde{\varphi}(v) = \varphi(v)\ \forall v \in V$ Следует отметить, что $\tilde{\varphi}$ аддитивен, поскольку является поворотом вокруг центра координат.

Рассмотрим $\phi: L \to L,\ \ \phi(c) = \tilde{\varphi}(c - c_1) + \hat{c}_1$ . Заданная функция изометрическая как комбинация поворота и параллельного переноса. Убедимся, что $\phi(c_i) = \hat{c}_i$. При $i = 1\ \ \phi(c_1) = \tilde{\varphi}(0) + \hat{c}_1 = \hat{c}_1$ , для каждого следующего элемента $\phi(c_i) = \tilde{\varphi}(c_i - c_1) + \hat{c}_1 = \tilde{\varphi}(c_i - c_{i-1}) + \tilde{\varphi}(c_{i-1} - c_1) + \hat{c}_1 = \tilde{\varphi}(a_i) + \hat{c}_{i-1} = \hat{c}_i$ , что доказывает лемму.

Таким образом показано, что приведенный алгоритм удовлетворяет условию абсолютной кодировки.

Теорему доказано.

**Заключение.** Абсолютная и относительная кодировки часто используются при исследовании процесса сворачивания белков в плоских решетках. В работе предложен подход к формальному определению этих понятий, на основании которых предлагаются алгоритмы построения кодировок в пространственных решетках с некоторыми свойствами. Преимущества относительной кодировки в ее устойчивости к поворотам, абсолютной – в большей (по сравнению с относительной) схожести путей с похожими кодами. Эти и другие особенности следует учитывать при выборе между ними в процессе разработки алгоритмов.

В дальнейшем исследования могут быть направлены на анализ результатов применения кодировок в различных алгоритмах для решения поставленной задачи. Важно отметить, что методы построения кодировок не привязаны к конкретной решетке, это расширяет разнообразие дискретных моделей.

*В.О. Рудик*

МЕТОДИ ПОДАННЯ СТРУКТУРИ БІЛКА В ТРИВИМІРНИХ ДИСКРЕТНИХ РЕШІТКАХ

Пропонується і досліджується підхід до формалізації понять абсолютного та відносного кодування, які використовуються при розв'язанні задачі прогнозування третинної структури протеїнів. Розглядається загальний випадок тривимірної решітки. Сформульовано і доведено основні властивості цих кодувань, пропонуються алгоритми їх побудови.

*V.O. Rudyk*

PROTEIN STRUCTURE NOTATION METHODS ON 3D DISCRETE LATTICES

The approach to formalize concepts of absolute and relative encodings used while solving protein tertiary structure prediction problem is proposed and studied. General case of discrete lattice is examined. Main characteristics of these encodings are formulated and proved, algorithms for their construction are proposed.

1. *Cutello V., Nicosia G., Pavone M., Timmis J.* An Immune Algorithm for Protein Structure Prediction on Lattice Models // IEEE Transactions on Evolutionary Computation. – 2007. – N 11(1). – P.101-117.
2. *Shmygelska A, Hoos H.* An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem // BMC Bioinformatics. – 2005. – N 6(30). – P.30–52.
3. *Wei W., Yanlin T.* A new algorithm for 2D hydrophobic-polar model: An algorithm based on hydrophobic core in square lattice // Pak. J. Biol. Sci. – 2008. – N 11. – P.1815–1819.
4. *Fidanova S., Lirkov I.* Ant Colony System Approach for Protein Folding // Int. Conf. Multiconference on Computer Science and Information Technology. – 2008. – P.887–891.
5. *Krasnogor N., Hart W., Smith J., Pelta .D.* Protein Structure Prediction With Evolutionary Algorithms // Proceedings of GACC. – 1999. – P.1596–1601.
6. *Mann M., Will S., Backofen R.* CPSP-tools - Exact and Complete Algorithms for High-throughput 3D Lattice Protein Studies // BMC Bioinformatics. – 2008. – N 9. –P.230–238.
7. *Mann M., Smith C., Rabbath M., Edwards M., Will S., Backofen R.* CPSP-web-tools: a server for 3D lattice protein studies // Bioinformatics. – 2009. – N 25(5). – P. 676–677.
8. *Local* rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model *Agarwala R., Batzoglou S., Dancik V* et al. // J. of Computational Biology. – 1997. – N 4. – P. 275–296.
9. *Гуляницкий Л., Рудык В.* Моделирование свертывания протеина в пространстве // Компьютерная математика. – 2010. – №1. – С. 128–138.

Получено 28.03.2011