КІБЕРНЕТИКА та КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

УДК 519.8 DOI:10.34229/2707-451X.20.3.6

С.А. ЧОРНОЖУК

НОВЕ ГЕОМЕТРИЧНЕ ПОДАННЯ ПРОСТОРУ «СТАНІВ-ДІЙ» Q-LEARNING АЛГОРИТМУ В ПРОБЛЕМІ ПЕРЕДБАЧЕННЯ ТРЕТИННОЇ СТРУКТУРИ БІЛКА

Розроблено новітнє подання простору станів та дій для алгоритму машинного навчання з підкріпленням Q-learning. Застосування Q-learning алгоритму з пропонованим поданням простору станів та дій досліджується на задачі передбачення третинної структури білків. Особливість пропонованого подання полягає в урахуванні геометричних властивостей результуючого ланцюга в кубічній ґратці. Ефективність такого підходу підтверджується експериментально на широкорозповсюдженому в світі наборі тестових даних.

Ключові слова: *просторова структура білка, комбінаторна оптимізація, відносне кодування, машинне навчання, Q-learning, рівняння Белмана, простір станів, простір дій, базис у трьохвимірному просторі.*

Вступ. Машинне навчання з підкріпленням – це актуальний апарат для розв'язування багатьох задач, в яких точний пошук оптимальних розв'язків є NP-складною задачею і тому неможливий на практиці. Машинне навчання з підкріпленням – це галузь машинного навчання, що досліджує дії, які мають виконувати програмні агенти в певному середовищі для максимізації деякого уявлення про сукупну винагороду при неможливому вичерпному пошуці в просторі, утвореному декартовим добутком множини станів на множину дій. Визначення вторинної або третинної структури білків на основі використання HP-моделі Діла [1] – важлива та актуальна проблема обчислювальної біології. Точний пошук оптимального розташування послідовності амінокислот (мономерів) у такій моделі є NP-складною задачею, а отже обчислювально неможливий вже на відносно невеликих розмірностях. Саме тому для передбачення вторинної або третинної структури білків на базі моделі Діла використовуються численні метаевристичні методи [2–8], а також й підходи машинного навчання з підкріпленням, а саме Q-learning підхід [9–12].

Варто зазначити, що левова частина методів машинного навчання з підкріпленням Q-learning використовуються саме для передбачення вторинної структури білка, адже для такої задачі простір станів та дій суттєво менший [9, 10, 12]. У цій статті ми транслювали ідеї, що вищезазначені у роботах, для постановки та розв'язування задачі передбачення саме третинної структури білка з використанням Q-learning методу, а також формалізовано нове подання простору «стани – дії» для тієї ж задачі.

Математична постановка задачі. У роботі використовується HP-модель Діла, яку представимо згідно [13]. У такій моделі кожна амінокислота (мономер) $\xi_i, i = \overline{1,n}$, яка входить до первинної структури білка, належить одному з двох типів: H – гідрофобний тип, P – полярний, а сама третинна структура подається як послідовність амінокислот, що розміщена у деякій

(просторовій) ґратці у вигляді неперервного ланцюга [14]. У цій роботі використовується кубічна ґратка. Важливо зазначити, що у такого ланцюга не має бути самоперетинів. Тоді цільова функція (енергія утвореної третинної структури) обчислюється за формулою

$$F(x) = -\sum_{1 \le i \le j-2 \le n-2} I(U(\xi_i), U(\xi_j)) h(\xi_i) h(\xi_j), \quad (1)$$

де $x \in X$ – неперервний ланцюг без самоперетинів довжиною n, X – простір усіх можливих неперервних ланцюгів довжиною n без самоперетинів, розміщених у кубічній ґратці, $U(\xi_i)$ – вузол у кубічній ґратці, який містить i-й мономер ланцюга (заданий декартовими координатами), а

$$I(U_1, U_2) = \begin{cases} 1, & \text{якщо вузли } U_1 \text{ та } U_2 \text{ сусідні в кубічній ґратці,} \\ 0, & \text{в іншому разі,} \end{cases}$$

$$h(\xi) = \begin{cases} 1, & \text{якщо } \xi = \text{H}, \\ 0, & \text{якщо } \xi = \text{P}. \end{cases}$$

Задача полягає у тому щоб знайти такий допустимий ланцюг $x_{opt} \in X$, що

$$x_{opt} = \arg\min_{x \in X} F(x). \quad (2)$$

Конкретний ланцюг із X зручно подавати не у вигляді абсолютних координат вузлів, у яких він розташований, а за допомогою внутрішнього відносного кодування [15], при якому вважається, що ланцюг починається в точці (0, 0, 0), а надалі кожен новий сегмент ланцюга задається відносним поворотом, щодо свого попереднього положення (вправо на 90° – right, вліво на 90° – left, догори на 90° – up, донизу на 90° – down, рух прямо без поворотів – front). Наприклад, ланцюг ABCDEF (рис. 1) при заданій початковій орієнтації, буде мати кодування (front, up, right, down, down).

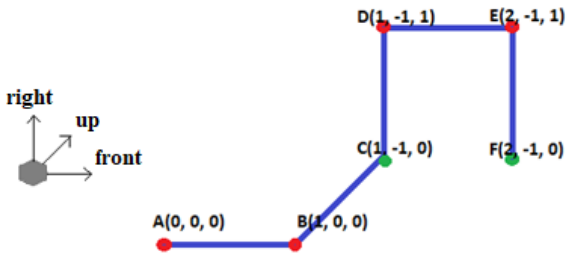


РИС. 1. Ланцюг ABCDEF має відносне кодування (front, up, right, down, down)

У роботі використовується відносне кодування, а початкова орієнтація задається за рахунок двох початкових неоднакових з точки зору абсолютного кодування поворотів. Перший з них обов'язково буде позначено front, а другий – up у рамках відносного кодування. Така початкова орієнтація робить відносне кодування інваріантним щодо будь-якого повороту, кратного 90° навколо будь-якої з координатних осей (рис. 2). Використання такого підходу зменшує простір X, а значить робить пошук x_{opt} у X більш ефективним. Нагадаємо, що кодування $Enc(U(\xi_1), U(\xi_2), ..., U(\xi_n))$ ланцюга x, який подається послідовності мономерів $\xi_1, \xi_2, ..., \xi_n$, є інваріантним щодо довільного відображення $f : \mathbb{Z}^3 \to \mathbb{Z}^3$, якщо ґратка та відношення сусідства в ній інваріантні відносно f, та $Enc(U(\xi_1), U(\xi_2), ..., U(\xi_n)) = Enc(f(U(\xi_1)), f(U(\xi_2)), ..., f(U(\xi_n)))$ [14].

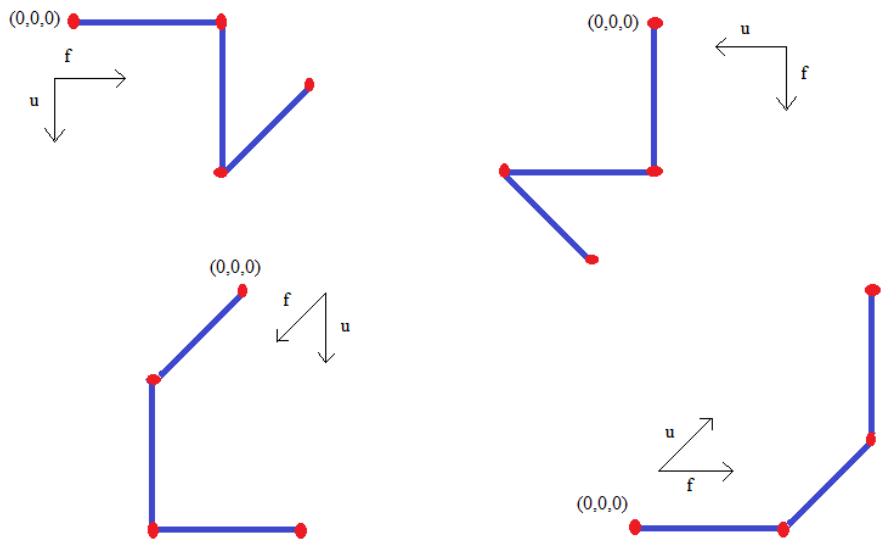


РИС. 2. Три ланцюги, утворені шляхом повороту лівого зверху на 90° навколо кожної з осей

Наприклад, використовуючи наведену орієнтацію, кожне з відносних кодувань буде трійка (front, up, right).

Загальний опис алгоритму Q-learning, який використовується для розв'язування задачі. Як показано у [9], Q-learning – це розширення традиційного підходу динамічного програмування, що розв'язує задачу, яку можна описати недетермінованим марковським процесом прийняття рішень [15]. При цьому функція розподілу ймовірностей визначає множину майбутніх потенційних станів системи, за умови виконання агентом певної фіксованої дії при певному поточному стані системи. В подальшому поняття «стан системи» буде конкретизуватися для досліджуваної проблеми. На відміну від інших підходів машинного навчання з підкріпленням, які здебільшого базуються на відповідності пари «стан-стан» певному скаляру, при підході Q-learning знаходиться відповідність пари «стан-дія» певному скалярному значенню (Q-value). Оптимальне значення Q-value – це сума підкріплень, яка може бути отримана, виконавши відповідну дію та послуговуючись оптимальній стратегії подальших дій (на кожному етапі обирати таку дію, якій відповідає найбільше значення Q-value).

У процесі навчання Q-learning алгоритм знаходить оптимальні Q-value ітеративно, використовуючи рівняння Белмана. Оновлення значень відбувається наступним чином:

$$Q(s,a) := Q(s,a) + \alpha(r(s,a) + \varphi \cdot \max_{a'}(Q(s',a'))), \tag{3}$$

де $Q_n(s,a)$ – Q-value на поточній ітерації алгоритму, $r(s,a)$ – винагорода, яку отримує агент, виконавши дію a перебуваючи в стані s, s' – стан, в який потрапить агент, перебуваючи у стані a та виконавши дію s. Для зручності введемо поняття функції переходу зі стану в стан $\delta: S \times A \to S$, яка визначає в який наступний стан перейде система, тобто $s' = \delta(s,a)$. У такому випадку рівність (3) буде іншого вигляду

$$Q(s,a) = Q(s,a) + \alpha(r(s,a) + \varphi \cdot \max_{a'}(Q(\delta(s,a),a'))), \tag{4}$$

де α – фактор навчання агента, який показує, як швидко агент реагує на нову інформацію, φ – фактор дисконтування агента, що показує, наскільки потужно агент реагує на винагороду після майбутніх своїх дій.

Необхідно ввести декілька нових параметрів алгоритму. Нехай задано I – загальна кількість ітерацій алгоритму, p – ймовірнісний поріг, який відповідає за стратегію вибору поточної дії, λ – темп зменшення p. Параметр λ відповідає за те, щоб на більш пізніх ітераціях агент обирав оптимальні дії з більшою ймовірністю, а *Softmax* – відома в машинному навчанні функція активації [15].

Важливо зазначити, що тут і надалі при адаптації відомих підходів для передбачення третинної структури білків простір дій складатиметься з можливих поворотів при відносному кодуванні ланцюга. Агентом, у свою чергу, є ітеративний процес утворення відносного кодування ланцюга. Коли говоримо, що агент виконує певну дію з простору дій, то маємо на увазі, щодо відносного кодування додається наступний поворот, що однозначно відповідає цій дії. Бінарний оператор $\oplus$ і позначає додавання чергового повороту до поточного відносного кодування.

У випадку задачі (2) схема пропонованого Q-learning алгоритма показана на рис. 3.

procedure *Qlearning*();
x_optimal := $\varnothing$;
x_current := $\varnothing$;
energy_optimal := 0;
for *i* **from** 0 **to** $I - 1$ **do**
for *j* **from** 0 **to** $n - 2$ **do**
s_current := визначити стан системи, що відповідає *x_current*;
possible_actions := $\varnothing$;
qualities := $\varnothing$;
for each *a* із множини можливих поворотів **do**:
$Q(s\_current, a) = Q(s\_current, a) + \alpha(r(s\_current, a) + \varphi \cdot \max_{a'}(Q(s\_next, a')))$
if *x_current* $\oplus$ *a* не містить самоперетинів **then**
Додати *a* в *кінець possible_actions*;
Додати $Q(s\_current, a)$ в кінець *qualities*;
endif
endfor
if *possible_actions* не пустий **then**
value := *random*[0; 1];
if *value* > *p* **then**
Агент виконує дію *a* з *possible_actions*, якому відповідає найбільше значення з *qualities* (тобто *x_current* := *x_current* $\oplus$ *a*);
else
probs := *Softmax*(*qualities*);
Агент виконує дію *a* з *possible_actions* з ймовірностями, описаними у *probs* (*x_current* := *x_current* $\oplus$ *a*);
endif
endif
endfor
energy := *F(x_current)*;
if *energy* < *energy_optimal* **then**
x_optimal := *x_current*;
energy_optimal := *energy*;
endif
p: = $\lambda \cdot p$;
endfor
return {*x_optimal, energy_optimal*};

РИС. 3. Загальна схема роботи Q-learning алгоритму

Аналіз пропонованих репрезентацій простору «стан-дія» в літературі. Як випливає із загальної схеми роботи Q-learning алгоритму, важливим є визначення стану системи, який відповідає тому чи іншому відносному кодуванню, а також обчислення винагороди, яку отримує агент виконавши ту чи іншу дію, перебуваючи в певному стані. Наприклад, у [10] простір «стан-дія» визначається наступним чином:

1) множина допустимих дій A фактично множиною поворотів при абсолютному кодуванні ланцюга. Тобто $A = \{a_1, a_2, a_3, a_4\}$, де a_1 = left, a_2 = right, a_3 = up, a_4 = down;

2) множина станів S складається з $\frac{4^n - 1}{3}$ станів, де в i-й стан можна потрапити, виконавши рівно $[\log_4(3i-2)]$ дій (поворотів). До того ж, неможливо потрапити у будь-який стан, виконавши більше однієї унікальної множини дій (поворотів). Функція переходу $\delta: S \times A \to S$ визначається наступним чином:

$$\delta(s_i, a_j) = s_{4i-3+j}, i \in \{1, 2, ..., (4^{n-1} - 1)/3\}, j \in \{1, 2, 3, 4\}.$$

Для наочності, простір станів та дій у [10] показано на рис. 4.

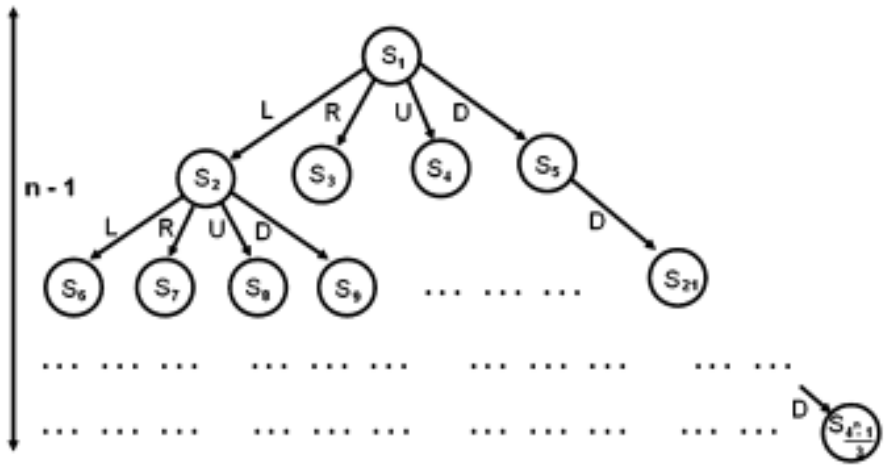


РИС. 4. Схема простору станів та дій [10]

Таке подання легко адаптується для передбачення третинної структури білка. Використовуючи відносне кодування замість абсолютного, єдиною відмінністю буде інша множина допустимих дій A та загальна кількість станів системи. Тоді $A = \{a_1, a_2, a_3, a_4, a_5\}$, де a_1 = left, a_2 = right, a_3 = up, a_4 = down, a_5 = front, а загальна кількість станів множини S тепер дорівнює $\frac{5^n - 1}{4}$.

Винагорода $r(s, a)$, яку отримує агент під час вибору тої чи іншої дії (повороту в кубічній ґратці), обчислюватиметься аналогічно до того, як вона обчислюється в квадратичній [10], а саме:

$$r(s,a) = r(a \mid s_1, a_1, a_2, ..., a_k) = \begin{cases} 0.01, \text{ якщо ланцюг } a_1 \oplus a_2 \oplus ... \oplus a_k \oplus a \text{ містить самоперетини.} \\ -F(a_1 \oplus a_2 \oplus ... \oplus a_k \oplus a), \text{ якщо } k+1 = n-1. \\ 0.1, \text{ інакше.} \end{cases} \quad (5)$$

Тут $r(a \mid s_1, a_1, a_2, ..., a_k)$ означає те саме, що і $r(s, a)$, де s – стан системи, отриманий шляхом виконання дій $a_1, a_2, ..., a_k$ на початку перебуваючи в стані s_1, а $a_1 \oplus a_2 \oplus ... \oplus a_k$ визначає ланцюг, який при цьому отримано (фактично $a_1 \oplus a_2 \oplus ... \oplus a_k$ визначає відносне кодування ланцюга, але надалі називатимемо $a_1 \oplus a_2 \oplus ... \oplus a_k$ ланцюгом). Автори у [10] зазначають, що умова $k+1 = N-1$ означає, що система перейшла в термінальний стан. Надалі будемо використовувати цей термін.

Таке подання простору станів має очевидний недолік – кількість станів є значною (для n = 48 їх кількість $\approx 8{,}88 \cdot 10^{32}$), а отже під час виконання алгоритму багато станів так і залишаться невідвіданими, що може суттєво погіршити знайдене рекордне значення цільової функції. Надалі називатимемо цю проблему надмірною інформативністю простору станів системи. Для вирішення цієї проблеми пропонуються й інші способи подання простору станів. Так, в [12] $A = \{a_1, a_2, a_3, a_4\}$, де a_1 = left, a_2 = right, a_3 = up, a_4 = down, але станом є пара, що утворюється останнім поворотом та його порядковим номером для поточної послідовності амінокислот з функцією переходу між станами

$$\delta(s_i, a_j) = \begin{cases} s_{i+(4-(i-1) \bmod 4)+j}, & \text{якщо } (i-1) \bmod 4 = 0. \\ s_{i+j}, & \text{інакше.} \end{cases}$$

Приклад подання простору станів показано на рис. 5, 6 [12]. Як і в [10], у цитованій роботі [12] розв’язується задача передбачення вторинної структури білка.

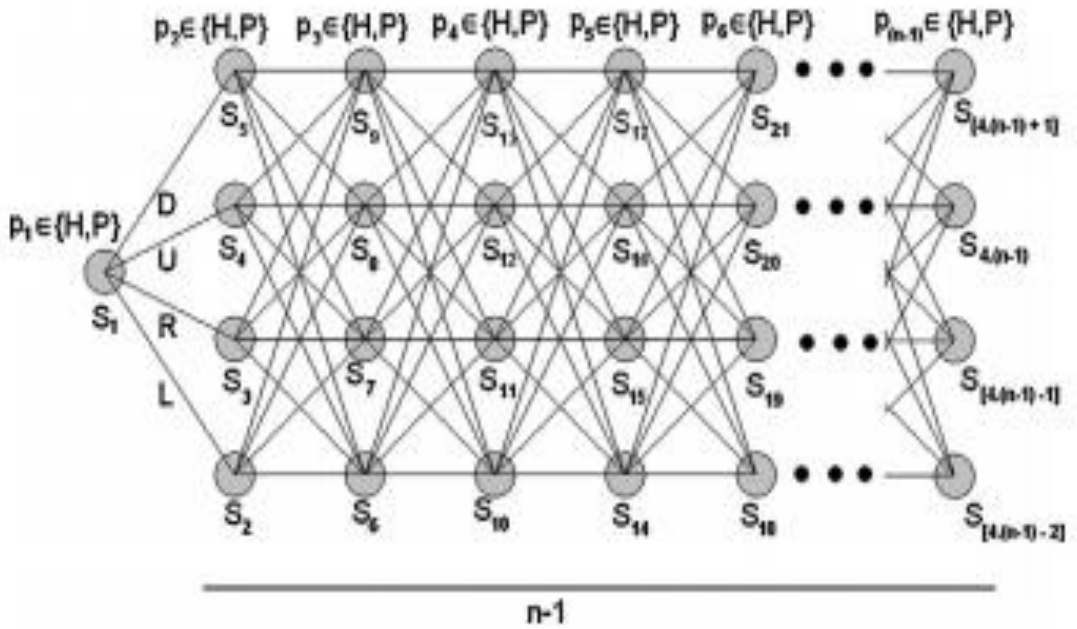


РИС. 5. Схема простору станів та дій [12]

S1 = H	L, R, U, D	S2 = HPL S3 = HPR S4 = HPU S5 = HPD
S2 = HPL S3 = HPR S4 = HPU S5 = HPD	L, R, U, D	S6 = HPHL S7 = HPHR S8 = HPHU S9 = HPHD
S6 = HPHL S7 = HPHR S8 = HPHU S9 = HPHD	L, R, U, D	S10 = HPHPL S11 = HPHPR S12 = HPHPU S13 = HPHPD
S10 = HPHPL S11 = HPHPR S12 = HPHPU S13 = HPHPD	L, R, U, D	S14 = HPHPPL S15 = HPHPPR S16 = HPHPPU S17 = HPHPPD
S14 = HPHPPL S15 = HPHPPR S16 = HPHPPU S17 = HPHPPD	L, R, U, D	S18 = HPHPPHL S19 = HPHPPHR S20 = HPHPPHU S21 = HPHPPHD

РИС. 6. Схема простору станів та дій [12] на прикладі послідовності амінокислот HPHPPH

Необхідно зазначити, що абсолютне кодування ланцюга не є вдалим для подання множини станів. Оскільки абсолютне кодування не є інваріантним щодо поворотів ґратки навколо координатних осей у тривимірному просторі та проти годинникової стрілки у двовимірному, то, наприклад, стану HPHPPHD може відповідати певний ланцюг, який при повороті ґратки на 90°, 180°, 270° проти годинникової стрілки буде вже відповідати станам HPHPPHR, HPHPPHU, HPHPPHL. Це означає, що всі термінальні стани системи будуть відповідати одній тій самій множині ланцюгів з точністю до поворотів ґратки. Отже, сумарна кількість винагороди, яку отримає агент при всіх можливих переходах до термінальних станів, буде однаковою для кожного з цих термінальних станів. Фактично це означає, що агент не буде мати достатньо інформації, щоб якісно відрізнити один стан від іншого. Надалі будемо називати цю проблему недостатньою інформативністю простору станів системи.

Недостатня інформативність простору станів системи робить Q-learning алгоритм малоефективним. Заміна абсолютного кодування відносним частково вирішує цю проблему. Тобто, розширивши множину дій A до $\{a_1, a_2, a_3, a_4, a_5\}$, де a_1 = left, a_2 = right, a_3 = up, a_4 = down, a_5 = front та замінивши абсолютне кодування відносним, стає можливим використання подання [12] для передбачення третинної структури білків.

Використання простору станів та дій, поданих у [9], разом із заміною абсолютного кодування ланцюга на відносне є компромісним підходом, що частково вирішує як проблему недостатньої, так і надлишкової інформативності простору станів системи. Тут станом є трійка, утворена координатами вузла мономера в ґратці, типом $\{H, P\}$ та порядковим номером мономера у первинній структурі білка (послідовності амінокислот). Множина дій ідентична з [12] для задачі передбачення третинної структури білка. Функція винагороди $r(s,a)$ обчислюється наступним чином:

$$r(s,a) = r(a \mid s_1, a_1, a_2, ..., a_k) = \begin{cases} -0.1, \text{ якщо ланцюг } a_1 \oplus a_2 \oplus ... \oplus a_k \oplus a \text{ містить самоперетини,} \\ \min(F(a_1 \oplus a_2 \oplus ... \oplus a_k) - F(a_1 \oplus a_2 \oplus ... \oplus a_k \oplus a), 0.1), \text{ інакше.} \end{cases}$$

У роботі [8] використовується підхід Deep Q-learning, в якому значення $Q(s, a)$ обчислюється не точно, а наближено з використанням LSTM нейронної мережі. Різниця між класичним Q-learning підходом та Deep Q-learning підходом показана на рис. 7 [9].

Важливо зазначити, що у своїй роботі ми не використовували Deep Q-learning підхід, але використали опис простору станів та дій з [9] в Q-learning алгоритмі, показаному на рис. 3.

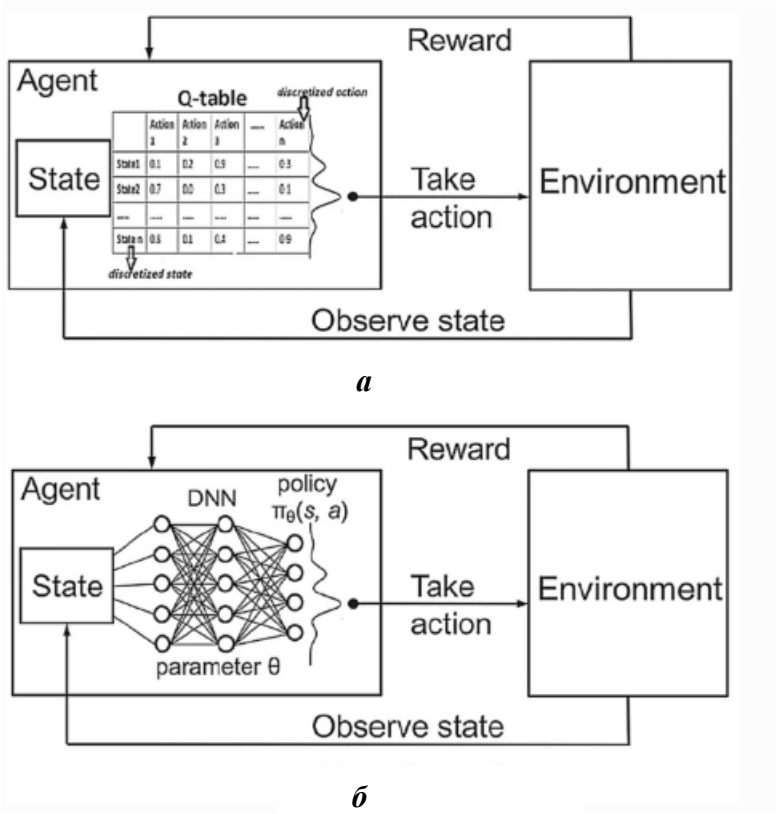


РИС. 7. Відмінність підходів: *а* – Q-learning; *б* – Deep Q-learning

Пропонований підхід до подання простору станів та дій. Очевидно, що, після виконання поворотів, якими визначено ланцюг при відносному кодуванні, результуюча орієнтація буде відрізнятися від початкової (рис. 8). Якщо в кінцевій орієнтації позначити напрямок front, наприклад, як орту (1, 0, 0), left, як орту (0, 1, 0), та up, як (0, 0, 1), то утвориться певний базис, в якому можна буде задати координати усіх мономерів у вузлах ґратки. Тоді станом системи $s_i^{l,\mu,\eta}$, $n \in \{1,2,...,n-1\}, \eta \le \mu \le l$ назвемо сукупність інформації.

1. Координати не більше, ніж μ найближчих вузлів у ґратці окрім попереднього, що містять мономери будь-якого з двох типів $\{H, P\}$ у базисі, утвореному кінцевою орієнтацією, після виконання l перших поворотів.

2. Координати не більше, ніж η найближчих вузлів у ґратці окрім попереднього, що містять H-мономери будь-якого з двох типів $\{H, P\}$ у базисі, утвореному кінцевою орієнтацією споглядання, після виконання l перших повротів.

3. Тип $(l + 1)$-го мономера у первинній структурі білка.

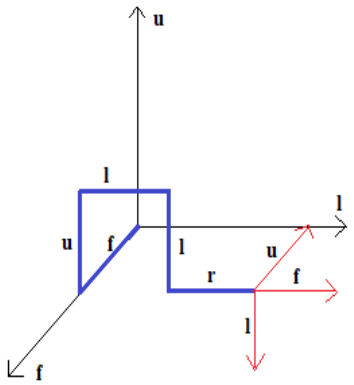


РИС. 8. Початкова та кінцева орієнтації ланцюга з відносним кодуванням forward, up, left, left, right

Множиною дій так само, як і раніше, є $A = \{a_1, a_2, a_3, a_4, a_5\}$, де a_1 = left, a_2 = right, a_3 = up, a_4 = down, a_5 = front. Тут повороти виконуються з урахуванням кінцевої орієнтації. Важливо зазначити, що кількість перших поворотів не є частиною подання стану системи, а отже $\delta(s_i^{l,\mu,\eta}, a) = s_j^{l+\Delta,\mu,\eta}$, де Δ може бути будь-яким елементом множини $\{-l, -l+1, ..., n-1-l\}$, а не тільки 1. Також необхідно вказати, що попередній вузол не розглядається, оскільки він є сусіднім у первинній структурі білка, тобто точно не впливає на можливе значення цільової функції F. Для наочності деякі приклади пропонованих станів системи з μ = 2 та η = 1 показані на рис. 9 (на рис. *а* після виконання поворотів front, up, left, left, right з первинною структурою HPHPHH отримаємо стан «**2 – 1 – 1 0 – 2 0 0 1 – 2 – 1 0 H**», на рис. *б* після виконання поворотів front, up, left, left, right, right, front, up з первинною структурою HPHPHHHHP отримаємо стан «**2 – 1 0 1 – 1 – 1 1 1 – 1 0 1 P**»).

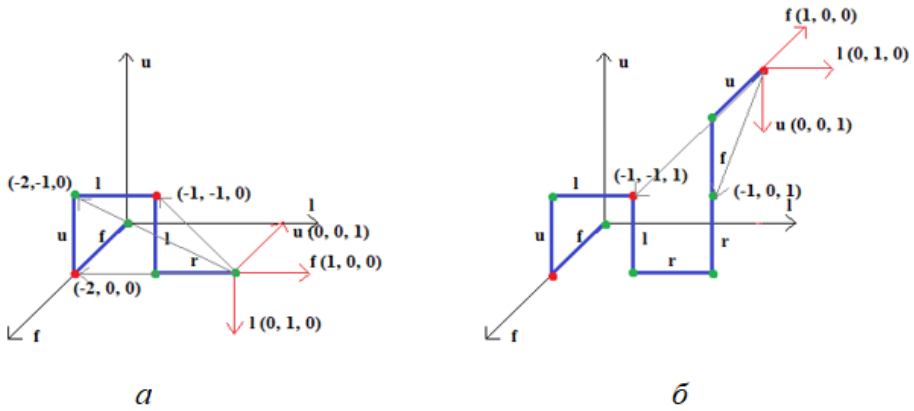


РИС. 9. Приклади пропонованих станів системи при μ = 2 та μ = 1

Таке подання станів, як і подання [9], також є компромісним підходом, що частково вирішує як проблему недостатньої, так і надлишкової інформативності простору станів, адже містить у собі інформацію лише про значущу частину ланцюга. Пропонована функція винагороди обчислюється наступним чином:

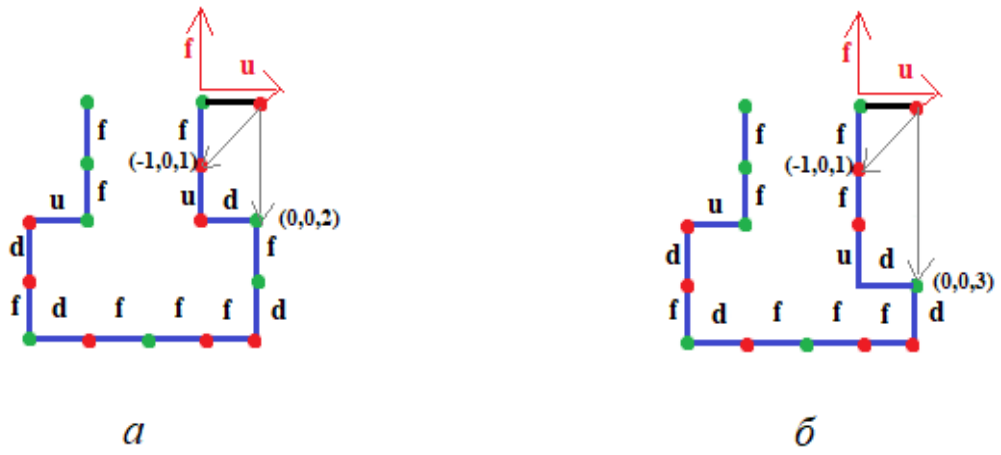
$$r(s,a)=r(a\mid s_1,a_1,a_2,...,a_k)=\begin{cases}-10, \text{ якщо ланцюг } a_1\oplus a_2\oplus...\oplus a_k\oplus a \text{ містить самоперетини,}\\ e^{2(F(a_1\oplus a_2\oplus...\oplus a_k)-F(a_1\oplus a_2\oplus...\oplus a_k\oplus a))+0.25(G(a_1\oplus a_2\oplus...\oplus a_k)-G(a_1\oplus a_2\oplus...\oplus a_k\oplus a))},\\ \text{інакше.}\end{cases}$$

Тут

$$G(a_1,a_2,...,a_k)=\frac{\sum_{i=1}^{k-1}\left\|U(\xi_{k+1})-U(\xi_i)\right\|\cdot h(\xi_i)}{\sum_{i=1}^{k-1}h(\xi_i)}.$$

Необхідно нагадати, що відносне кодування $a_1,a_2,...,a_k$ однозначно задає положення мономерів у ґратці $U(\xi_1),U(\xi_2),...,U(\xi_{k+1})$.

Слід також зазначити, що функція $\delta: S\times A\to S$ може бути недетермінованою. На рис. 10, *а*, *б* показано приклад недетермінованості функції $\delta: S\times A\to S$ при пропонованому поданні простору станів та дій. Наприклад, з параметрами $\mu = 1$ та $\eta = 1$ обидва відносних кодування front, front, up, down, front, down, front, front, front, down, front, down, up, front та front, front, up, down, front, down, front, front, front, down, down, up, front, front переведуть систему в поточний стан **«1 – 1 0 0 1 0 0 – 2 H»**. В обох випадках агент виконує поворот вгору. В першому випадку така дія переведе систему в стан **«1 – 1 0 1 1 0 0 2 P»** (рис. 10, *а*), в другому – у стан **«1 – 1 0 1 1 0 0 3 P»**. (рис. 10, *б*).



а *б*

РИС. 10. Приклад недетермінованості функції $\delta: S\times A\to S$ при пропонованому поданні простору станів та дій

Обчислювальний експеримент. Q-learning алгоритм тестувався на 10 реальних значеннях білків довжиною 48, які широко використовуються при аналізі алгоритмів розв'язування задачі, і вперше були запропоновані у [16] (табл. 1). Обчислювальний експеримент проводився на персональному комп'ютері Apple MacBook Pro Touch Bar 13 2019 з процесором 1.4 GHz Quad-Core Intel Core i5 та оперативною пам'яттю 8 GB 2133 MHz LPDDR3.

Параметри алгоритму Q-learning та параметри опису станів системи обрані на основі попередніх досліджень так:

$$\alpha = 0.01,\ \varphi = 0.95,\ p = 1.0,\ \lambda = 0.999994,\ I = 100000,\ \mu = 8,\ \eta = 8.$$

ТАБЛИЦЯ 1. Номер та код послідовності

№ послідовності	PH код послідовності
1	HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
2	HHHHPHHPHHHHHPPHPPHHPPHPPPPPPHPPHPPPHPPHHPPHHHPH
3	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP
4	PHPHHPPHPHHHPPHHPHHPPPHHHHHPPHPHHPHPHPPPPHPPHPHP
5	PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPPPPHHPHHPH
6	HHHPPPHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHPPHHHHHHPH
7	PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPHHPPHHPPPH
8	PHHPHHHPHHHHPPHHHPPPPPPHPHHPPHHPHPPPHHPHPHPHHPPP
9	PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH
10	PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPPHHHHHHHPPHH

Для кожної з 10 послідовностей зроблено 5 прогонів Q-learning алгоритму з різним поданням простору станів та дій для передбачення третинної структури білка, а саме пропонованого та наведеного у [9, 10, 12]. У рамках цих 5 прогонів для кожної з послідовностей підраховувалось найменше ($F_{\min}$) та середнє (F_{avg}) значення цільової функції (енергії).

Основні результати наведено в табл. 2, де $F_{\min}$ – рекорд із знайдених у кожному прогоні мінімальних значень енергії, F_{avg} – середнє для обраної кількості прогонів, F_{opt} – відоме оптимальне значення цільової функції для кожної з послідовностей.

ТАБЛИЦЯ 2. Результати обчислювального експерименту Q-learning алгоритму з різними *S*, *A*, *R*(*s*, *a*)

№	F_{opt}	Пропоновані *S*, *A*, *r*(*s*, *a*)		*S*, *A*, *r*(*s*, *a*) із [9]		*S*, *A*, *r*(*s*, *a*) із [10]		*S*, *A*, *r*(*s*, *a*) із [12]	
		$F_{\min}$	F_{avg}	$F_{\min}$	F_{avg}	$F_{\min}$	F_{avg}	$F_{\min}$	F_{avg}
1	– 32	– 29	– 26.8	– 20	– 19	– 12	– 11.2	– 12	– 10.6
2	– 34	– 29	– 27.2	– 20	– 19	– 11	– 10.6	– 14	– 11.8
3	– 34	– 28	– 23.8	– 19	– 18	– 12	– 11.2	– 14	– 11.4
4	– 33	– 21	– 20	– 20	– 19	– 13	– 10.8	– 11	– 10.4
5	– 32	– 24	– 21.2	– 20	– 18.8	– 14	– 11.6	– 12	– 11
6	– 32	– 21	– 19.6	– 21	– 18.8	– 12	– 10.2	– 12	– 10.4
7	– 32	– 20	– 19.6	– 18	– 17.4	– 12	– 10.6	– 12	– 10.4
8	– 31	– 27	– 25.4	– 19	– 18.2	– 11	– 10.2	– 12	– 11
9	– 34	– 26	– 25.4	– 21	– 19	– 13	– 11.2	– 13	– 11.4
10	– 33	– 28	– 25.2	– 18	– 17.2	– 12	– 10.2	– 12	– 10.2

Також для кожної ітерації серед всіх прогонів для всіх послідовностей підраховувалась середня кількість відвіданих пар (*s*, *a*), тобто тих, для яких *Q*(*s*, *a*) вже містило значення, що відрізняються від значень за замовчуванням. Результати показано на рис. 11, 12. На цих рисунках вісь абсцис – номер ітерації, вісь ординат – кількість відвіданих пар.

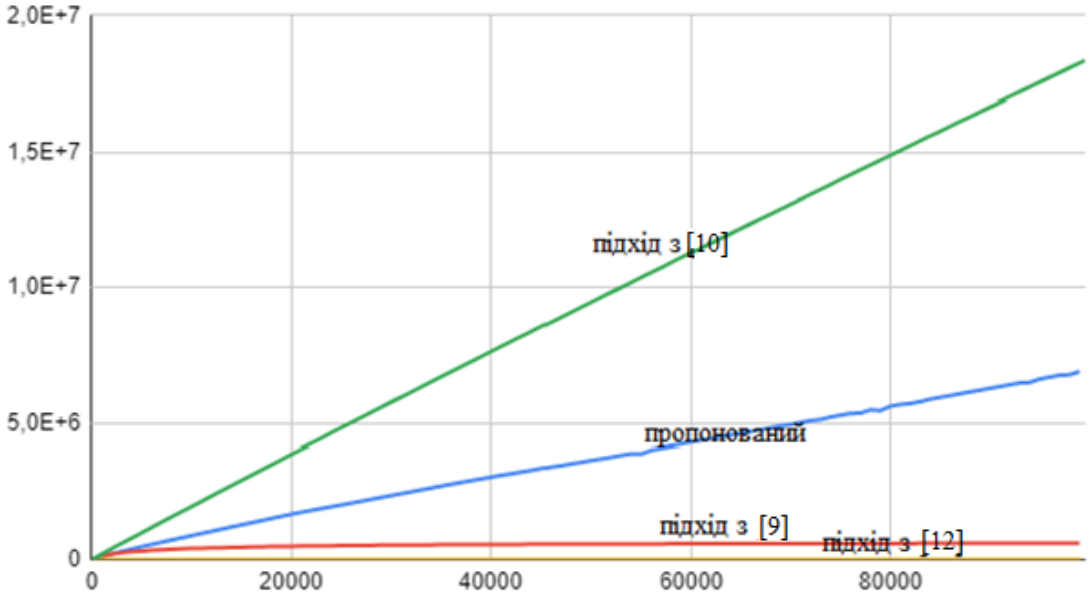


РИС. 11. Середня кількість відвіданих пар (*s*, *a*) зі звичайним масштабуванням вертикальної шкали

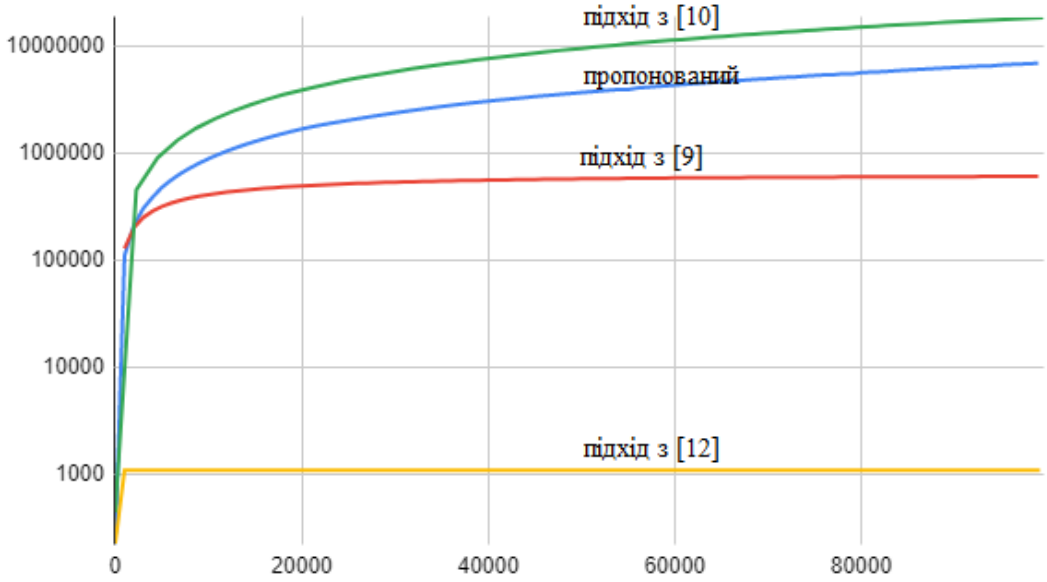


РИС. 12. Середня кількість відвіданих пар (*s*, *a*) з логарифмічним масштабуванням вертикальної шкали

Висновки. Результати обчислень показані на графіках рис. 11, 12, підтверджують те, що підходи [10, 12], адаптовані для передбачення третинної структури білків, породжують проблему надлишкової та недостатньої інформативності простору станів системи відповідно. Пропонований та підхід із [9], у цьому сенсі виглядають збалансовано, що і відображають результати у табл. 2. До того ж використання пропонованого подання простору станів та дій разом із пропонованою функцією винагороди в алгоритмі Q-learning показало кращі результати, ніж використання всіх інших подань. Отримані результати підтверджують перспективність подальших досліджень.

Серед важливих напрямів таких досліджень зазначимо.

1. Для пропонованих станів та дій функція переходу $\delta: S \times A \to S$ між станами за певною дією не є детермінованою (рис. 10, *а, б*). Перспективним виглядає дослідження щодо незначної зміни або розширення опису станів системи для того, щоб зробити функцію переходу детермінованою. Це може покращити якість розв'язків, отриманих Q-learning алгоритмом.

2. Цікавим виглядає комбінація Deep Q-learning підходу з пропонованими поданням простору станів та дій, а також функцією винагороди. Незважаючи на те, що пропонований простір станів частково вирішує проблеми і недостатньої, і надлишкової інформативності, апроксимація $Q(s, a)$ за рахунок нейронних мереж або інших евристик може суттєво покращити роботу алгоритму.

Список літератури

1. Dill K.A. Theory for the folding and stability of globular proteins. *Biochemistry*. 1985. **24** (6). P. 1501–1509. https://doi.org/10.1021/bi00327a032
2. Bazzoli A., Tettamanzi A.G.B. A Memetic Algorithm for Protein Structure Prediction in a 3D-Lattice HP Model. *Applications of Evolutionary Computing*. 2004. **3005**. P. 1–10. https://doi.org/10.1007/978-3-540-24653-4_1
3. Custodio F.L., Barbosa H.J., Dardenne L.E. A multiple minima genetic algorithm for protein structure prediction. *Applied Software Computing, Elsevier*. 2014. **15**. P. 88–99. https://doi.org/10.1016/j.asoc.2013.10.029
4. Boscovic B., Brest J. Genetic algorithm with advanced mechanisms applied to the protein structure prediction in a hydrophobic-polar model and cubic lattice. *Applied Soft Computing*. 2016. **45**. P. 61–70. https://doi.org/10.1016/j.asoc.2016.04.001
5. Morshedian A., Razmara J., Lotfi S. A novel approach for protein structure prediction based on estimation of distribution algorithm. *Software computing*. 2019. **23**. P. 4777–4788. https://doi.org/10.1007/s00500-018-3130-0
6. Nazmul R., Chetty M., Chowdhury A.R. Multimodal Memetic Framework for low-resolution protein structure prediction. *Swarm and Evolutionary Computation, Elsevier*. 2020. **52**. https://doi.org/10.1016/j.swevo.2019.100608
7. Hulianytskyi L.F., Rudyk V.O. Development and analysis of the parallel ant colony optimization algorithm for solving the protein tertiary structure prediction problem. *Information Theories and* Applications. 2014. **21** (4). P. 392–397.
8. Chornozhuk S.A. The new simulated annealing algorithm for a protein structure folding problem. *Komp'uternaa matematika*. 2018. 1. P. 118–124. http://dspace.nbuv.gov.ua/handle/123456789/161856
9. Jafari R., Javidi M.M. Solving the protein folding problem in hydrophobic-polar model using deep reinforcement learning. *SN Applied Sciences, Springer*. 2020. **2** (259). https://doi.org/10.1007/s42452-020-2012-0
10. Czibula G., Bocicor M., Czibula I. A reinforcement learning model for solving the folding problem. *Int J Computational Technology Applied*. 2011. 2. P. 171–182.
11. Li Y., Kang H., Ye K., Yin S. FoldingZero: protein folding from scratch in hydrophobic-polar model. *Deep reinforcement learning workshop (Oral) of NIPS*. 2018. https://arxiv.org/abs/1812.00967
12. Dogan B., Olmez T. A novel state space representation for the solution of 2D-HP protein folding problem using reinforcement learning methods. *Applied Soft Computing, Elsevier*. 2015. **26**. P. 213–223. https://doi.org/10.1016/j.asoc.2014.09.047
13. Гуляницький Л.Ф., Чорножук С.А. Генетичний алгоритм з жадібним стохастичним оператором схрещування для передбачення третинної структури білка. *Cybernetics and Computer Technologies*. 2020. **2**. P. 19–29. https://doi.org/10.34229/2707-451X.20.2.3

14. Гуляницкий Л.Ф., Рудык В.А. Проблема предсказания структуры протеина: формализация с использованием кватернионов. *Кибернетика и системный анализ*. 2013. **49** (4). С. 130–136. https://doi.org/10.1007/s10559-013-9546-8
15. Sutton R.S., Barto A.G. Reinforcement Learning: An Introduction. *MIT Press*, 1998. **9** (5). P. 1054. https://doi.org/10.1109/TNN.1998.712192
16. Yue K., Fiebig K.M., Thomas P.D., Chan H.S., Shakhnovich E.I., Dill K.A. A Test of Lattice Protein Folding Algorithms. *Proceedings of the National Academy of Sciences*. 1995. **92** (1). P. 325–329. https://doi.org/10.1073/pnas.92.1.325

Одержано 21.08.2020

Чорножук Сергій Анатолійович,
аспірант Інституту кібернетики імені В.М. Глушкова НАН України, Київ.
chornozhuk@gmail.com

УДК 519.8

С.А. Черножук

Новое геометрическое представление пространства «состояние-действие» Q-learning алгоритма в проблеме предсказания пространственной структуры белка

Институт кибернетики имени В.М. Глушкова НАН Украины, Киев
Переписка: chornozhuk@gmail.com

Введение. Сворачивание пространственной белковой структуры – важная и актуальная проблема вычислительной биологии. Рассматривая математическую модель задачи, можно легко сделать вывод, что поиск оптимального положения белка в трехмерной сетке является NP-сложной задачей. Следовательно, для решения проблемы можно использовать некоторые методы обучения с подкреплением, такие как Q-learning. В статье предлагается новое геометрическое представление пространства «состояние-действие», которое существенно отличается от всех альтернативных представлений, используемых для этой задачи.

Цель роботы. Анализ существующих подходов к представлению пространств состояний и действий для алгоритма Q-learning для задачи предсказания трехмерной структуры белков, выявление их преимуществ и недостатков, предложение нового геометрического представления пространства «состояние-действие». Дальше необходимо сравнить существующие и предлагаемые подходы, сделать выводы и описать возможные будущие шаги дальнейших исследований.

Результат. Работа предложенного алгоритма сравнивается с другими на основе 10 известных цепочек длиной 48, впервые предложенных в [16]. Для каждой из цепочек алгоритм Q-learning с предложенным представлением пространства «состояние-действие» превзошел тот же алгоритм Q-обучения с альтернативными существующими представлениями пространств «состояние-действие» как с точки зрения средних, так и минимальных значений энергии полученных положений белков. Более того, множество существующих представлений используется для двумерных предсказаний структуры белка. Однако в ходе экспериментов как существующие, так и предлагаемое представления были немного изменены или доработаны для решения проблемы в 3D, что является более сложной задачей с точки зрения вычислений.

Вывод. Экспериментально подтверждено качество алгоритма Q-learning с предложенным геометрическим представлением пространства «состояние-действие». Следовательно, доказано, что дальнейшие исследования перспективны. Более того, уже было предложено несколько шагов будущих исследований, таких как объединение предлагаемого подхода с методами глубокого машинного обучения.

Ключевые слова: пространственная структура белка, комбинаторная оптимизация, относительное кодирование, машинное обучение, Q-обучение, уравнение Беллмана, пространство состояний, пространство действий, базис в трехмерном пространстве.

UDC 519.8

S. Chornozhuk

The New Geometric "State-Action" Space Representation for Q-Learning Algorithm for Protein Structure Folding Problem

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv
Correspondence: chornozhuk@gmail.com

Introduction. The spatial protein structure folding is an important and actual problem in computational biology. Considering the mathematical model of the task, it can be easily concluded that finding an optimal protein conformation in a three dimensional grid is a NP-hard problem. Therefore some reinforcement learning techniques such as Q-learning approach can be used to solve the problem. The article proposes a

new geometric "state-action" space representation which significantly differs from all alternative representations used for this problem.

The purpose of the article is to analyze existing approaches of different states and actions spaces representations for Q-learning algorithm for protein structure folding problem, reveal their advantages and disadvantages and propose the new geometric "state-space" representation. Afterwards the goal is to compare existing and the proposed approaches, make conclusions with also describing possible future steps of further research.

Result. The work of the proposed algorithm is compared with others on the basis of 10 known chains with a length of 48 first proposed in [16]. For each of the chains the Q-learning algorithm with the proposed "state-space" representation outperformed the same Q-learning algorithm with alternative existing "state-space" representations both in terms of average and minimal energy values of resulted conformations. Moreover, a plenty of existing representations are used for a 2D protein structure predictions. However, during the experiments both existing and proposed representations were slightly changed or developed to solve the problem in 3D, which is more computationally demanding task.

Conclusion. The quality of the Q-learning algorithm with the proposed geometric "state-action" space representation has been experimentally confirmed. Consequently, it's proved that the further research is promising. Moreover, several steps of possible future research such as combining the proposed approach with deep learning techniques has been already suggested.

Keywords: Spatial protein structure, combinatorial optimization, relative coding, machine learning, Q-learning, Bellman equation, state space, action space, basis in 3D space.