
РАЗРАБОТКА И ИССЛЕДОВАНИЕ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ТРЕТИЧНОЙ СТРУКТУРЫ ПРОТЕИНА

Леонид Гуляницкий, Виталина Рудык

Аннотация: Рассматривается оптимизационная задача вычислительной биологии – прогнозирование структуры протеина по последовательности аминокислотных остатков. Описывается HP-модель, которая, не смотря на свою простоту, достаточно точно отображает процессы сворачивания в реальных молекулах. Предложены локальные и глобальные алгоритмы решения. Сравнительные характеристики методов и их комбинаций проиллюстрированы результатами вычислительного эксперимента по решению задач разной размерности.

Ключевые слова: комбинаторная оптимизация, прогнозирование третичной структуры протеина, метод ветвей и границ, стохастический локальный поиск, алгоритмы муравьиных колоний.

Conference: The paper is selected from XVth International Conference "Knowledge-Dialogue-Solution" KDS 2009, Varna, Bulgaria, June-July 2009

Введение

Задача прогнозирования третичной структуры протеина по последовательности аминокислотных остатков – одна из самых важных и сложных задач вычислительной биологии. Поскольку процессы сворачивания протеинов изучены не в полной мере, исследователями предложен ряд упрощенных моделей, которые базируются на физических свойствах молекул и приводят к задачам комбинаторной оптимизации. Но и предложенные точные алгоритмы не решают задачи за удовлетворительное время даже для последовательностей длины 50, что дает основания для разработки приближенных алгоритмов. В настоящее время для решения возникающих задач предложен ряд таких алгоритмов популяционного типа [1, 2, 3], а также алгоритмов с оценкой точности [4, 5]. Необходимость в получении более точных решений побуждает к разработке новых детерминированных и метаэвристических алгоритмов, которые и являются предметом дальнейшего исследования.

Постановка задачи

Задача прогнозирования третичной структуры протеина состоит в определении его формы в пространстве, исходя из последовательности аминокислотных остатков в его цепи. В большинстве известных моделей пространство представляется дискретной решеткой. Соседние в последовательности остатки должны располагаться в соседних узлах решетки. Тогда структура протеина задается самонепересекающимся путем в решетке – сверткой. Задача прогнозирования третичной структуры состоит в поиске свертки с минимальной энергией, которая подсчитывается согласно определенному правилу. Таким образом, для описания задачи необходимо выбрать тип решетки и функцию энергии.

В данной работе рассматривается гидрофобно-полярная модель (HP-модель) Дилла [6]. В HP-модели 20 аминокислотных остатков делятся на две группы – гидрофобные (неполярные) и гидрофильные (полярные). Тогда входную последовательность можно рассматривать как слово из алфавита $\{H, P\}$, где P обозначает полярные, а H – гидрофобные остатки. В заданной структуре контактирующие остатки определяются как остатки, которые не являются соседними в последовательности, но расположены в соседних узлах решетки. Между контактирующими гидрофобными остатками возникают связи. Энергия

каждой связи принимается за -1, а энергия структуры – это сумма энергий всех связей в ней. С биологической точки зрения это объясняется тем, что при сворачивании протеина гидрофобные остатки для предотвращения контакта с водой перемещаются внутрь молекулы, в то время как гидрофильные остаются извне. Согласно гипотезе, протеин принимает ту структуру, которая минимизирует площадь воздействия гидрофобных остатков с водой или полярными остатками. Несмотря на простоту NP-модели, процессы свертывания в ней имеют сходство с таковыми в реальных протеинах.

Будем рассматривать двумерную модель протеинов. Для нее изначально Диллом была выбрана квадратная решетка. Но одним из ее свойств является то, что два остатка могут быть соседними в решетке только тогда, когда количество символов между ними будет четным. Так, строка $(HP)^n$ в квадратной решетке не будет иметь ни одной связи, что не соответствует логическим представлениям. Кроме того, такое строгое ограничение усложняет анализ приближенных алгоритмов, определяя искусственно низкую нижнюю границу энергии, вследствие чего оценка точности алгоритма не будет иметь существенного значения для реальных задач. Поэтому в работе была выбрана треугольная решетка, в которой таких проблем не возникает, так как для любых двух позиций в строке существует свертка, в которой соответствующие остатки будут соседями по решетке.

Доказано, что задача прогнозирования третичной структуры протеина в NP-модели является NP-трудной [7, 8].

Детерминированный локальный поиск

Прежде всего, рассмотрим алгоритм детерминированного локального поиска для минимизации энергии протеина в NP-модели. В общем, он не дает удовлетворительных результатов, поскольку функция энергии имеет много локальных минимумов, но может быть использован в более сложных алгоритмах.

Пусть n – длина аминокислотной последовательности протеина. Каждую свертку в треугольной решетке можно закодировать в виде строки длины $n-2$ из алфавита $T = \{F, Fl, Fr, Bl, Br\}$, элементы которого обозначают поворот на каждом шагу – Front, Front-left, Front-right, Back-left, Back-right.

Назовем свертку допустимой, если она задает путь без самопересечений на решетке. Обозначим множество допустимых сверток D . Определим расстояние между свертками $\rho(s, v)$ как количество символов, на которое отличаются их кодировки s и v . Обозначим $O_\delta(s)$ окрестность точки s размера δ , т.е. $O_\delta(s) = \{v \mid \rho(s, v) \leq \delta\}$, а $E(s)$ – энергию свертки s . В приведенных обозначениях схему детерминированного локального поиска показано на рис. 1 [9].

```

procedure LocalSearch ( $s_0$ )
    while (окрестность  $O_1(s_0)$  не просмотрена полностью) do
         $v :=$  некоторый вариант из  $O_1(s_0)$ ;
        if  $v \in D$  and  $E(v) < E(s_0)$  then
             $s_0 := v$ ;
    end while;
    return  $s_0$ ;
end procedure.

```

Рисунок 1 Схема алгоритма детерминированного локального поиска

Таким образом, алгоритм гарантирует улучшение текущей свертки в окрестности, если оно существует. Сложности возникают в том, что условие допустимости является достаточно строгим, поэтому в окрестности $O_1(s)$ может оказаться малое количество допустимых вариантов решения (сверток) или не оказаться вообще. Поэтому предлагается в процедуре LocalSearch заменить окрестность $O_1(s)$ окрестностью большего размера $O_2(s)$, что позволяет улучшать полученные результаты, но, с другой стороны, увеличивает время выполнения. Поэтому для повышения эффективности предлагается использовать стохастические методы локального поиска.

Алгоритм имитационного отжига

Алгоритм имитационного отжига – один из широко используемых алгоритмов локальной оптимизации [10]. Его схема похожа на детерминированный локальный поиск, но за счет стохастической составляющей алгоритм может выходить из локальных минимумов, возвращая более точное значение. Основная идея состоит в том, что переход к лучшим значениям осуществляется всегда, а к "худшим" (в смысле значений целевой функции) – с определенной вероятностью, зависящей от значения целевой функции и номера итерации. Его схема показана на рис.2.

```

procedure SA ( $S_0$ )
     $S_{rec} := S_0$ ;  $T :=$  начальное значение;
    while окрестность  $O_1(s_0)$  не просмотрена полностью do
        while не достигнуто равновесие do
             $v :=$  некоторый вариант из  $O_1(s_0)$ ;
            if ( $v \in D$ ) then
                 $\Delta := E(v) - E(s_0)$ ;
                 $p := \min\{1, \frac{\Delta}{T}\}$ ;
                 $\xi := \text{random}[0, 1]$ ;
                if ( $p \geq \xi$ ) then
                     $s_0 = v$ ;
                    if ( $E(v) < E(s_{rec})$ ) then
                         $S_{rec} := v$ ;
                    end if;
                end if;
            end while;
             $T :=$  следующее значение;
        end while;
    return  $S_{rec}$ ;
end procedure.

```

Рисунок 2 Схема алгоритма имитационного отжига

Для задачи прогнозирования структуры протеина алгоритм имитационного отжига был предложен и проанализирован в работе [11], где можно найти рекомендации по выбору параметров алгоритма и особенностям реализации.

G-алгоритм

Эффективным алгоритмом комбинаторной оптимизации также является G-алгоритм [12]. В отличие от алгоритма имитационного отжига, вероятность перехода к худшему значению на каждом шаге не зависит от номера итерации, но изменяется пороговое значение, которое определяет условия отсева ухудшающих вариантов (рис. 3).

```

procedure G-algorithm (  $s_0$  )
   $s_{rec} := s_0$ ;  $\mu_0 := 0$ ;  $h := 0$ ;  $t := 0$ ;
  while (окрестность  $O_1(s_0)$  не просмотрена полностью) do
    while (не достигнуто равновесие) do
       $v :=$  некоторый вариант из  $O_1(s_0)$ ;
      if ( $v \in D$ ) then
         $\tilde{\varphi} := \varphi(s_0, v)$ ;
         $p := \min\{1, \max\{0, \tilde{\varphi}\}\}$ ;
         $\xi := \mu_t + \text{random}[0,1] \cdot (1 - \mu_t)$ ;
        if ( $p \geq \xi$ ) then
           $s_0 := v$ ;  $h := h + 1$ ;
          if ( $E(v) < E(s_{rec})$ ) then
             $s_{rec} := v$ ;
          end if;
        end if;
      end while;
       $\mu_{t+1} := G(\mu_t)$ ;  $t := t + 1$ ;
    end while;
  return  $s_{rec}$ ;
end procedure.

```

Рисунок 3 Схема G-алгоритма

В качестве функции $\varphi(x, y)$ выберем функцию $\varphi(x, y) = 1 - \frac{E(y) - E(x)}{\gamma \cdot (E(x) - E_{\min})}$,

где E_{\min} – нижняя граница энергии протеина. Для ее подсчета используется свойство, что каждый гидрофобный аминокислотный остаток, кроме первого и последнего, в последовательности в треугольной решетке может образовывать максимум четыре связи, а первый и последний остатки, если они гидрофобные -- по пять связей. Тогда, если обозначить n_H количество гидрофобных остатков в последовательности, E_{\min} можно посчитать по формуле:

$$E_{\min} = -\frac{4n_H + 2}{2} = -(2n_H + 1).$$

В качестве функции G выбрано $G(x) = (\eta_k(x) + H)^k$,

где $0 < H < 1$, $k \in \{1, 2, 3\}$ – параметры алгоритма,

$$\eta_k(x) = \begin{cases} 0, & x = 0, \\ x^{1/k}, & x > 0. \end{cases}$$

Алгоритм имитационного отжига и G-алгоритм в среднем дают более заметное улучшение допустимого решения задачи, чем детерминированный локальный поиск, при этом время расчетов увеличивается несущественно, что дает основания для их дальнейшего использования.

Алгоритм свертывания стрелкой

В [13] предложен алгоритм для решения поставленной задачи с априорной оценкой точности. Гарантированная точность – $\frac{1}{2}$ от оптимального решения. Алгоритм является детерминированным, конструктивным. Сворачивание производится таким образом, чтобы все гидрофобные остатки находились на одном из трех уровней – так называемом костяке свертки (см. рис. 4)

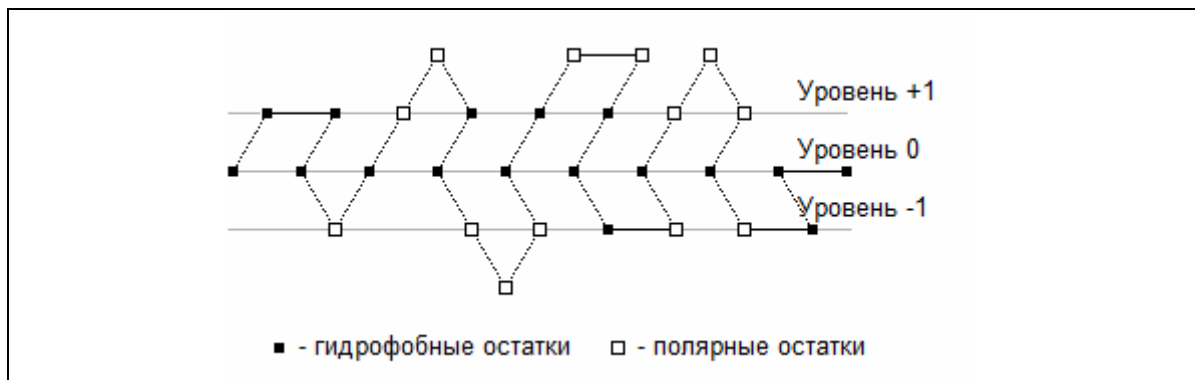


Рисунок 4 Сворачивание стрелкой для последовательности RHPHPHPHPHPHPHPHPHPHPHPHPHPHPHP

Алгоритм более эффективен для последовательностей с высоким соотношением числа гидрофобных к числу полярных остатков. Важное преимущество – линейные затраты по времени, что, во-первых, важно для практических задач больших размерностей, и, во-вторых, позволяет использовать его в других более сложных алгоритмах. Недостатком алгоритма является невозможность достигнуть желаемой точности.

Метод ветвей и границ

Для поиска точного решения поставленной задачи предлагается метод ветвей и границ, идея которого состоит в использовании двух процедур – ветвления и поиска границ. Ветвление – деление множества допустимых решений на подмножества меньших размеров, а поиск границ – поиск верхней и нижней оценки значения целевой функции на определенном множестве. При этом для задачи на минимум при условии, что нижняя оценка для подмножества допустимых решений A больше верхней оценки для подмножества B , то A не содержит в себе оптимального решения и дальше может не рассматриваться. Ветвление предлагается делать последовательно по каждой координате на пять областей – по числу возможных направлений.

Для нижней границы предлагается следующая формула:

$$m = -(b_{ee} + b_{en} + b_{mn} + 1),$$

$$b_{en} = 4 * \min\{p_f, p_n\}, \quad b_{mn} = 2 * (p_n - \min\{p_f, p_n\}),$$

где b_{ee} – количество связей между фиксированными остатками, b_{en} и b_{mn} – максимальное количество связей, которая может быть образована свободными остатками, которые свяжутся с фиксированными и свободными соответственно, p_n – количество свободных гидрофобных остатков, p_f – количество мест в решетке, в которых фиксированные остатки могут образовывать связи.

Для получения верхней оценки используется описанный выше алгоритм свертывания стрелкой. Свободные остатки располагаются соответственно алгоритму, а в том случае, когда такое положение не является допустимым – равновероятно в любом другом положении. Предусмотрено генерацию допустимого решения в том случае, если оно существует в данном подмножестве. В качестве верхней оценкой взято значение целевой функции на построенном решении.

Как и любой точный метод, предложенный алгоритм требует больших затрат по времени, что делает его непрактичным для реальных задач.

Алгоритм оптимизации муравьиными колониями

Одним из современных стохастических методов комбинаторной оптимизации является алгоритм оптимизации муравьиными колониями [14]. Его общая схема показана на рис. 5, где k – параметр алгоритма.

```

procedure ACO()
   $s_{rec} := null$ ;
  ИнициализироватьФеромоннуюМатрицу(M);
  while (не выполняется условия завершения) do
    for  $i = 1, \dots, k$  do
       $v_i := \text{ДопустимоеРешение}(M)$ ;
       $v_i := \text{ЛокальныйПоиск}(v_i)$ ;
      if ( $E(v_i) < E(s_{rec})$ ) then
         $s_{rec} := v_i$ ;
    end for;
    for  $i = 1, \dots, k$  do
      ОбновитьФеромоннуюМатрицу(M,  $v_i$ );
    end while;
  return  $s_{rec}$ ;
end procedure.

```

Рисунок 5 Схема алгоритма оптимизации муравьиными колониями

Феромонная матрица в рассматриваемой задаче имеет размерность $5 \times (n - 2)$. На каждой итерации вероятность совершить i -й поворот после $(j + 1)$ -го аминокислотного остатка в структуре пропорционален элементу матрицы с номером $[i, j]$ и оценке позиции $e(pos)$, которая считается по следующему правилу: $e(pos) = \sum_{p \in O(pos)} e_{f(pos)f(p)}$, где pos – узел решетки, на котором может находиться следующий остаток, $O(pos)$ – соседние узлы, $f(\cdot) \in \{H, P, 0\}$ – функция, которая возвращает H , если в узле находится гидрофобный остаток, P , если полярный и 0 в случае, если узел не занят. Числа $e_{HH}, e_{HP}, e_{H0}, e_{PH}, e_{PP}, e_{P0} \geq 0$ – параметры алгоритма, должны удовлетворять условиям:

$$e_{HP} \leq e_{H0} \leq e_{HH}, \quad e_{PH} \leq e_{P0} \leq e_{PP}.$$

Обновление феромонной матрицы происходит отдельно по частям последовательности, элементы которой образуют связи только внутри этой части. В качестве процедуры локального поиска может быть выбран один из трех вышеописанных алгоритмов.

Вычислительный эксперимент

Для сравнения показателей эффективности алгоритмов был проведен вычислительный эксперимент, результаты которого приведены в табл. 1 и 2.

В табл. 1 отражены результаты применения алгоритмов локального поиска – детерминированного (ДЛП), детерминированного с окрестностью радиуса 2 (ДЛП-2), имитационного отжига (АИО) и G-алгоритма. Для тестирования была случайным образом сгенерирована последовательность задач размерности от 15 до 300 с соотношением гидрофобных остатков к полярным 1:4, 1:2, 1:1, 2:1, 4:1 для каждой размерности – всего 40 задач. Все алгоритмы в качестве начального приближения получали одну и ту же свертку в каждой задаче. В этой табл. приведены усредненные результаты, сгруппированные по длине входящей последовательности – n . Здесь $\overline{f_0}$ – усредненная энергия свертки, которая поступает на вход алгоритма, \overline{t} – усредненное время выполнения в миллисекундах, $\overline{f^*}$ – усредненное значение оптимальной энергии, выданное алгоритмом, \overline{q} – усредненное значение улучшения $q = \frac{f_0 - f^*}{|f_0|} \cdot 100\%$. В контексте таких подсчетов показатель \overline{q} является более корректным, чем $\overline{f^*}$, так как чем больше в молекуле соотношение количества гидрофобных остатков к количеству полярных, тем меньше нижняя граница энергии, следственно величина $\overline{f^*}$, в отличие от \overline{q} , неравномерно взвешена по задачам. Стохастические алгоритмы для каждой задачи запускались по три раза, поэтому для них выводится также показатель $\overline{f_{\min}^*}$ – усредненное значение лучшего результата для каждой задачи.

Табл. 1

n	$\overline{f_0}$	ДЛП			ДЛП-2			АИО				G			
		$\overline{t}, \text{мс}$	$\overline{f^*}$	$\overline{q}, \%$	$\overline{t}, \text{мс}$	$\overline{f^*}$	$\overline{q}, \%$	$\overline{t}, \text{мс}$	$\overline{f^*}$	$\overline{f_{\min}^*}$	$\overline{q}, \%$	$\overline{t}, \text{мс}$	$\overline{f^*}$	$\overline{f_{\min}^*}$	$\overline{q}, \%$
15	-1,2	3	-4,2	260	16	-6,0	410	5	-5,4	-6,4	360	10	-5,4	-6,4	360
50	-10,6	34	-23,0	177	220	-29,2	328	46	-23,8	-26,2	180	61	-21,9	-28,4	175
75	-14,2	70	-28,4	156	518	-35,2	289	89	-28,8	-29,8	197	127	-26,3	-30,6	181
100	-24,8	141	-41,0	137	750	-49,6	236	191	-44,1	-46,6	177	270	-45,3	-48,8	201
150	-34,6	383	-66,2	82	2208	-71,4	125	457	-63,9	-67,6	83	546	-67,0	-69,0	89
200	-39,0	838	-79,0	140	4343	-97,2	210	790	-76,0	-81,6	134	937	-74,3	-83,0	115
250	-51,2	941	-91,8	106	5481	-115,6	156	1240	-92,6	-96,6	110	1373	-93,9	-95,0	113
300	-38,0	589	-81,6	38	3397	-84,9	41	574	-72,5	-128,6	30	872	-71,8	-133,0	29

В табл. 2 приведены результаты, полученные алгоритмами ветвей и границ (АВГ), свертывания стрелкой (АСС), свертывания стрелкой + G-алгоритм (С+G), оптимизации муравьиными колониями без локального поиска (ОМК) и с детерминированным локальным поиском (ОМК+ДЛП). Поскольку в отличие от алгоритмов локального поиска, в приведенных алгоритмах нет начального приближения, для подсчета относительного улучшения, задачи не группировались по размерности. В табл. 2 n обозначает размер последовательности, $h:p$ – соотношение количества гидрофобных остатков к количеству полярных, t – время работы алгоритма в секундах, f^* – полученное оптимальное значение. Как и в случае стохастических локальных алгоритмов, алгоритмы ОМК и ОМК+ ДЛП выполнялись по три раза для каждой

задачи, f_{\min}^* – наименьшее из полученных значений. Критерий остановки: оптимальное значение не изменялось на протяжении количества итераций, которое равняется длине входящей последовательности. Алгоритм ветвей и границ выполнялся с ограничениями по времени, в качестве которого выбиралась величина в пять раз большая, чем время, использованное алгоритмами ОМК и ОМК+ ДЛП.

Табл.2

n	$h:p$	АВГ		АСС		АСС+G		ОМК			ОМК+ ДЛП		
		t, c	f^*	t, c	f^*	t, c	f^*	t, c	\bar{f}^*	f_{\min}^*	t, c	\bar{f}^*	f_{\min}^*
15	4:1	10	-12	<1	-11	<1	-12	<1	-11,7	-12	<1	-12,0	-12
15	2:1	10	-12	<1	-9	<1	-9	<1	-11,0	-12	<1	-11,7	-12
15	1:1	10	-6	<1	-5	<1	-5	<1	-5,0	-6	<1	-6,0	-6
15	1:2	10	-6	<1	-4	<1	-4	<1	-5,3	-6	<1	-4,7	-5
15	1:4	10	-5	<1	-3	<1	-4	<1	-4,7	-5	<1	-4,3	-5
50	4:1	25	-52	<1	-44	<1	-49	2,0	-44,0	-46	5,0	-51,0	-52
50	2:1	12	-51	<1	-44	<1	-48	1,3	-48,7	-52	2,3	-52,7	-54
50	1:1	22	-34	<1	-29	<1	-31	1,3	-37,7	-41	4,3	-37,0	-38
50	1:2	15	-12	<1	-12	<1	-12	1,0	-8,3	-9	3,0	-8,3	-9
50	1:4	15	-17	<1	-15	<1	-16	1,0	-17,7	-19	3,0	-16,7	-18
100	4:1	660	-113	<1	-99	<1	-99	11,3	-98,7	-104	131,7	-112,0	-114
100	2:1	200	-97	<1	-78	1	-87	10,3	-82,7	-87	40,0	-90,0	-92
100	1:1	257	-57	<1	-52	<1	-53	8,3	-52,7	-55	51,3	-55,0	-56
100	1:2	127	-34	<1	-33	1	-34	10,3	-34,0	-35	25,3	-34,3	-35
100	1:4	65	-24	<1	-22	1	-24	2,7	-20,0	-23	13,0	-22,0	-24
300	4:1	2797	-350	<1	-294	6	-324	189,3	-300,7	-326	559,3	-342,3	-359
300	2:1	5259	-280	<1	-249	6	-261	185,3	-244,0	-260	1051,7	-272,7	-279
300	1:1	1413	-167	<1	-157	4	-164	146,0	-137,0	-151	282,7	-150,3	-154
300	1:2	1677	-98	<1	-95	7	-96	177,3	-72,3	-80	335,3	-79,3	-89
300	1:4	2035	-66	<1	-66	3	-66	160,3	-51,3	-55	407,0	-59,7	-64

Вычислительный эксперимент показал существенную зависимость показателей эффективности алгоритмов от их параметров, чем обуславливается необходимость разработки методологии их настройки в зависимости от типа решаемых задач.

Заключение

Практически каждый из "базовых" алгоритмов имеет свои преимущества при решении поставленных задач в зависимости от их характеристик. Возможность создания на их основе метаэвристических алгоритмов позволяет достичь лучших показателей эффективности.

Целью дальнейших исследований может быть: теоретическое исследование сходимости некоторых из вышеописанных алгоритмов; формализованный выбор значений параметров алгоритмов в зависимости от длины входящей последовательности и, возможно, соотношения гидрофобных и полярных остатков в ней для повышения эффективности; применение более агрегированных алгоритмов для решения поставленной задачи; обобщение полученных результатов на трехмерную решетку.

Благодарности

Статья частично финансирувана из проекта **ITHEA XXI** Института Информационных теории и Приложений FOI ITHEA и Консорциума FOI Bulgaria (www.ithea.org, www.foibg.com).

Список литературы

- [1] R.Unger, J.Moult. Genetic Algorithms for protein folding simulations // J. of Molecular Biology.–1993. – 231(1). –P.75–81.
- [2] A.Piccolboni, G.Mauri. Application of Evolutionary Algorithms to Protein Folding Prediction // Artificial Evolution. – 1997. – P. 123–136.
- [3] M.Khimasia, P.Coveney. Protein Structure Prediction as a Hard Optimization Problem: the Genetic Algorithm Approach // Molecular Simulation. – 1997. – 19. – P. 205–226.
- [4] W.Hart, S.Istrail. Fast protein folding in the hydrophobic-hydrophilic model within three-eighths of optimal // J. of Computational Biology. – 1996 –3(1). –P. 53–96.
- [5] R.Agarwala, S.Batzoglou, V.Dancik, SE.Decatur, M.Farach, S.Hannenhalli, S.Muthukrishnan, S.Skienna. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model // J. of Computational Biology. – 1997. – 4, №3 . – P. 75–96.
- [6] K.Dill, S.Bromberg, K.Yue, K.M.Fiebig, D.Yee, P.Thomas, H.Chan. Principles of protein folding - a perspective from simple exact models // Protein Science. – 1995. – 4. – P. 561– 602.
- [7] B.Berger, T.Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete // J. of Computational Biology. – 1998. –5(1). – P. 27–40.
- [8] P.Crescenzi, D.Goldman, C.Papadimitriou, A.Piccolboni, M.Yannakakis. On the complexity of protein folding // J. of Computational Biology. –1998. – 5(3) . – P. 423–465.
- [9] И.В. Сергиенко. Математические модели и методы решения задач дискретной оптимизации. – К.: Наукова думка, 1985. – 384 с.
- [10] H.Hoos, T.Stützle. Stochastic Local Search: Foundations and Applications. – San Francisco: Morgan Kaufmann Publ., 2005. – 658 p.
- [11] A. Albrecht, K. Steinhöfel. Run-time Estimates for Protein Folding Simulation in the H-P Model. – 9th Int. Symp. on Artificial Intelligence and Mathematics, Fort Lauderdale, Florida, 2006.
- [12] Гуляницкий Л.Ф. Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // Компьютерная математика. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2004. – №1. – С. 64–72.
- [13] R.Agarwala, S.Batzoglou, V.Dancik, S.E.Decatur, M.Farach, S.Hannenhalli, S.Muthukrishnan, S. Skienna. Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP model // J. of Computational Biology. – 1997 – 4.– P. 275–296.
- [14] M.Dorigo M., T. Stützle. Ant Colony Optimization. – Cambridge: MIT Press, MA, 2004. – 348 p.

Об авторах

Леонид Гуляницкий (*Hulianytskyi*) – д.т.н., ведущий научный сотрудник Института кибернетики им. В.М.Глушкова НАН Украины, пр-т Глушкова, 40, Киев, 03680, Украина. e-mail: lh_dar@hotmail.com

Виталина Рудык (*Rudyk*) – студентка, Киевский Национальный Университет им. Тараса Шевченка, ф-т кибернетики, пр-т Глушкова, 2, корп. 6, Киев, 03680, Украина. e-mail: vitalina_@rambler.ru