Теорія оптимальних рішень

УДК 519.8

Л.Ф. ГУЛЯНИЦЬКИЙ, С.А.ЧОРНОЖУК

ГІБРИДНИЙ АЛГОРИТМ ОПТИМІЗАЦІЇ МУРАШИНИМИ КОЛОНІЯМИ ДЛЯ ПЕРЕДБАЧЕННЯ СТРУКТУРИ БІЛКІВ

Запропоновано новий гібридний алгоритм, який дозволяє автоматично налаштовувати параметри алгоритму оптимізації мурашиними колоніями для розв'язування задачі передбачення третинної структури білка.

Вступ. Визначення просторової структури білків є необхідним етапом для встановлення взаємозв'язку між просторовою (третинною) структурою та функцією білків. Відома третинна структура білку дозволяє визначити необхідне лікування важких хвороб генетичного плану для конкретного індивідууму. В 1960-х роках американський біохімік, Нобелівський лауреат Крістіан Анфінсен спостулював термодинамічну гіпотезу, згідно з якою атоми в молекулах білка укладаються в природніх умовах у термодинамічно стабільну конформацію, що відповідає мінімуму вільної енергії системи [1]. Це дозволяє розробляти математичні моделі для передбачення просторової структури білків, однією із найпоширеніших серед яких є модель Ділла [2]. Знаходження оптимальної третинної структури білка – задача із класу *NP*, що обмежує можливість застосування точних методів. На сьогодні в комбінаторній оптимізації існує ряд підходів, що дозволяють знаходити наближені до точних розв’язки за відносно високої швидкодії. Одним із них є алгоритм оптимізації мурашиними колоніями (ОМК), вперше запропонований Марко Доріго [3, 4]. Але такі алгоритми потребують тонкого налаштування значень параметрів, що саме по собі є складною задачею. В статті запропоновано гібридний алгоритм, який дозволяє автоматично налаштовувати параметри алгоритму ОМК для конкретної послідовності амінокислот.

1. Про математичну постановку задачі. Маємо $< X, P, N, V, F >$ – кортеж, який відображає ключові аспекти задачі комбінаторної оптимізації, що подає проблему передбачення структури білка. Тут X – область допустимих значень структур, що подаються у вигляді маршрутів (ламаних без самоперетинів) у заданій ґратці [5]; P – предикат, що визначає допустимість маршруту; V та N – послідовність амінокислот, що визначає білок, та її розмірність відповідно; F – цільова функція, яку потрібно мінімізувати.

Усі формальні означення компонент кортежу $< X, P, N, V, F >$ описані в [5]. Більш детальне обговорення питань формалізації подано в [6].

2. Пропонований алгоритм розв'язування. У процесі роботи розроблено гібридний алгоритм, що поєднує у собі алгоритм детермінованого локального пошуку та ОМК [7].

Для подальшого опису алгоритму необхідно ввести декілька позначень.

Нехай $Pars = (Pars_1, Pars_2, ..., Pars_n)$ – деяка множина можливих наборів параметрів ОМК, де $Pars_i$ – певний конкретний набір параметрів ОМК.

Для значень кожного параметру $p \in Pars_i, i = \overline{1, N}$ введемо дискретну ґратку можливих значень $\delta(p, s_p, \varepsilon_p, n_p) = (s_p, s_p + \varepsilon_p, s_p + 2\varepsilon_p, ..., s_p + n\varepsilon_p)$, де s_p – початкове значення параметра p. Нехай маємо певні два значення p_1, p_2 з ґратки $\delta(p, s_p, \varepsilon_p, n_p)$. Тоді $D_{\delta(p, s_p, \varepsilon_p, n_p)}(p_1, p_2) = \dfrac{| p_2 - p_1 |}{\varepsilon}$ – відстань між двома конкретними значеннями в ґратці $\delta(p, s_p, \varepsilon_p, n_p)$.

Нехай у нас є фіксована кількість параметрів ОМК $N_{par.}$ Покладемо

$$\eta(par_1, par_2) = \sum_{i=1}^{N_{par}} D_{\delta(p_i, s_{pi}, \varepsilon_{pi}, n_{pi})}(p_{i1}, p_{i2}), p_{i1} \in par_1, p_{i2} \in par_2,$$

де $par_1, par_2 \in Pars$ відстань між двома наборами параметрів ОМК par_1, par_2.

$O_r(par_i) = \{par_j : \eta(par_i, par_j) \le r\}$ – окіл параметрів $par_i \in Pars$ з радіусом r. У практичній реалізації алгоритма будемо використовувати окіл $O_1(par_i)$ радіусу $r = 1$.

Загальна обчислювальна схема пропонованого алгоритму показана на рис. 1.

procedure *DLSACO*();
xOptimal = деякий припустимий варіант розв'язку;
parsOptimal = деякі припустимі параметри ОМК;
for each *pars* in *Pars* **do**
$xCurrent = ACO(pars)$
if $f(xCurrent) < F(xOptimal)$ **then**
$xOptimal = xCurrent$
$parsOptimal = pars$
endif

endfor
while *не вичерпано весь окіл* $O_1(parsOptimal)$ **do**
$pars = Генерація\ Чергової\ Точки\ Околу\ O_1(parsOptimal)$
$xCurrent = ACO(pars)$
if $f(xCurrent) < F(xOptimal)$ **then**
$xOptimal = xCurrent$
$parsOptimal = pars$
endif
endwhile
return $\{xOptimal, parsOptimal\}$
end

РИС. 1. Загальна схема гібридного алгоритму

3. Ключові аспекти метода оптимізації мурашиними колоніями. Загальна схема ОМК для знаходження оптимальної структури білків базується на алгоритмі з [8], але з урахуванням в функції пристосованості допоміжних компонент, що враховують евристичну інформацію.

При реалізації алгоритму кожен конкретний набір $par \in Pars$ складався з таких параметрів:

G – кількість поколінь мурах, які не знайшли покращення розв'язку, необхідних для зупинки алгоритму;

n_a – кількість мурах у поколінні;

ρ – коефіцієнт випаровування феромонного сліду;

α – коефіцієнт значущості феромонного сліду;

β – коефіцієнт значущості основної евристичної інформації;

γ – коефіцієнт значущості допоміжної евристичної інформації;

θ – поріг випадковості переходу по ребру;

τ_0 – нижня межа значення феромону на ребрі;

τ_1 – верхня межа значення феромону на ребрі;

n_e – кількість розв'язків-маршрутів у одному поколінні, що використовуються для оновлення глобальної пам'яті мурах;

n_d – кількість розв'язків-маршрутів у одному поколінні, що використовуються демоном для покращення результатів.

Сама множина наборів параметрів *Pars* визначалася парою n_a та n_e при фіксованих інших параметрах ОМК. Тобто

$$Pars = \{(12,\ 200+\Delta n_a,\ 0.1,\ 0.2,\ 0.8,\ 0.5,\ 0.6,\ 0.2,\ 0.8,\ 1+\Delta n_e,\ 2)\},$$

де $\Delta n_a = 400 * i, i = \overline{0,5}$ та $\Delta n_e = \overline{0,19}$.

Така множина визначалася емпірично і показувала гарні результати роботи алгоритму при відносно високій швидкодії.

Проаналізувавши [8], запропоновано таку схему переходу з додавання наступного ребра до ланцюга (рис. 2).

procedure *ChooseEdge*();
  for each $e_i \in ((1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1))$ **do**

$$p_{e_i,d} = \frac{(\tau_{e_i,d})^{\alpha}(\eta_{e_i,d})^{\beta}(\sigma_{e_i,d})^{\gamma}}{\sum_{e_j \in ((1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1))} (\tau_{e_j,d})^{\alpha}(\eta_{e_j,d})^{\beta}(\sigma_{e_j,d})^{\gamma}};$$

  endfor
  p = *random*[0,1];
  if p > θ **then**
    result = *Обрати ребро з найбільшим значенням* $p_{e_i,d}$;
  else then
    result = *Обрати одне з ребер, враховуючи ймовірності переходу* $p_{e_i,d}$;
  endif
  return result;
end

РИС. 2. Схема алгоритму додавання ребра до ланцюга в ОМК

Тут кожен поворот e_i [5] при фіксованій довжині ланцюга еквівалентний ребру графу задачі ОМК, описаному в [4]. Надалі будемо вважати твердження «обрати поворот e_i при фіксованій довжині ланцюга» та «обрати *i*-те ребро при фіксованій довжині ланцюга» еквівалентними.

Визначимо евристичну інформацію так: $\eta_{e_i,d} = e^{\Delta F}$, де ΔF – різниця значення *F* при умові вибору *i*-го ребра та поточного значення *F* (при довжині ланцюга *d*); $\tau_{e_i,d}$ – це значення феромонного сліду на *i*-му ребрі при поточній довжині ланцюга *d*, а

$$\sigma_{e_i,d} = \frac{d \cdot (N-d) \cdot |N - dist[\sum_{j=1}^{d} e_j + e_i(\sum_{j=1}^{d} v_j \sum_{k=1}^{j} e_j) / \sum_{j=1}^{d} v_j]|}{N \cdot (\sum_{j=d+1}^{N} v_j)}.$$

Як і в [5], $dist(x, y)$ – це евклідова відстань у просторі Z^3.

Варто зазначити, що така допоміжна евристична інформація для ОМК запропонована вперше. Мотивацією для такого підходу стала фітнес-функція для методу імітаційного відпалу [5]. Як і вищезгадана фітнес-функція, $\sigma_{e_i,d}$ слугує для того, щоб далекі ізольовані гідрофобні амінокислоти (наприклад, 110110**100000001**) не віддалялися від групи інших гідрофобних амінокислот при побудові чергового ланцюга.

Очевидно, що при запропонованій схемі вибору ребер під час конструювання ланцюга можуть виникати самоперетини. Для того, щоб справлятися з цим, запропоновано наступний алгоритм генерації ланцюга (рис. 3).

procedure *GeneratePath*();
  $visited_d$ – множина опрацьованих ребер при довжині ланцюга d ;
  $visited_d$:={};
  pnt :=(0,0,0);
  $path$:={(0,0,0)};
  while *довжина ланцюга* **do**
    $possible_d$ – множина допустимих ребер при довжині ланцюга d ;
    $possible_d$:={};
    for each $e_i \in ((1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1))$ **do**
      if $pnt + e_i$ not in path **then**
        Додати e_i у $possible_d$;
      endif
      if $possible_d$ = {} **then**
        $visited_d$:={};
        $pnt = pnt - path_d$;
        Видалити останній елемент з $path$;
        d:=d –1;
      else
        Обрати ребро e_{opt} *використовуючи процедуру* ***ChooseEdge***;
        Додати e_{opt} *у* $visited_d$;
        $pnt = pnt + e_{opt}$;
        Додати e_{opt} *у* $path$;
        d:=d +1;
      endif
    endfor
  endwhile
  return $path$;
end

РИС. 3. Схема алгоритму генерації ланцюга в ОМК

На кожній генерації мурах з ОМК отримуємо n_a різних можливих ланцюгів. На n_d найкращих ланцюгів (n_d ланцюгів з найменшим значенням F) діє демон – в нашому випадку, детермінований локальний пошук з [5]. Після дії демону для кожної мурахи вираховуємо внесок феромону по ребрах шляху, згенерованого нею.

Як і в [7], внесок феромону мурахи обраховується за формулою

$$\Delta\tau_i = \tau_0(1-\frac{f-f_{\min}}{\overline{f}-f_{\min}}),\ \ i=\overline{1,n_a}\ ,$$

де $f_{\min}$, $\overline{f}$, f – мінімальне, середнє, поточне значення цільової функції F у черговій генерації мурах. Тільки n_e найменших $\Delta\tau_i$ беруть участь в оновленні феромону. Феромон оновлюється і випаровується, так як описано у [7].

Результати та висновки. Розроблені алгоритми тестувалися на 10 реальних значеннях білків довжиною 48, які взяті із відомої бібліотеки [5]. Результати обчислювального експерименту подані в таблиці, де $F(x_{opt})$, $x_{opt} \in X$, результуюче значення цільової функції у точці розв'язку, знайденої розробленим алгоритмом.

ТАБЛИЦЯ. Результати обчислювального експерименту

Код білка	$F(x_{opt})$
1011001111011100110010110101100100000000000110	– 29
1111011011111001001100100000010010000100110011101	– 31
0101101111110010100101101010001001100110010010	– 31
0101100101110011011000111110010110100001001010	– 30
0010001011110011111011011001010100100000001101101	– 30
1110001101011011011010000000100100100010011111101	– 30
0100001011101011111011011000010100011100110110001	– 31
0110111011110011100000010110011010000110101011000	– 30
0101000010101001011111110011101001011100100001	– 30
0110000001100011101001011001001001100111111110011	– 29

Гібридизація алгоритмів детермінованого локального пошуку та ОМК дозволяє автоматично налаштувати параметри ОМК до кожної конкретної структури білка, що разом із пропонованою допоміжною евристичною інформацією дозволяє знайти кращі значення цільової функції на реальних даних середньої довжини (30 – 60). Проте цей підхід потребує і подальших досліджень, серед яких зазначимо такі.

Питання вибору множини параметрів ОМК, за якими алгоритм виконує детермінований локальний пошук, залишається відкритим. Пропонований набір параметрів визначався емпірично, але подальші дослідження та інші підходи до вибору цих значень можуть значно покращити як і отримувані результати, так і швидкодію алгоритму.

Спосіб подання графу задачі для ОМК може бути об'єктом подальших досліджень [6]. Зокрема, наприклад, можна враховувати поточне значення цільової функції при переході по ребру графа, що, можливо, дозволить зробити роботу алгоритму більш ефективною.

Важливим напрямом підвищення ефективності запропонованого алгоритму є розроблення його версій з розпаралелюванням обчислювальної схеми.

Доцільно також досліджувати питання вибору ґратки $\delta(p, s_p, \varepsilon_p, n_p), \forall p \in pars, pars \in Pars,$ яка напряму впливає на ефективність гібридного алгоритму.

Л.Ф. Гуляницкий, С.А. Черножук

ГИБРИДНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ МУРАВЬИНЫМИ КОЛОНИЯМИ ДЛЯ ПРЕДСКАЗАНИЯ СТРУКТУРЫ БЕЛКОВ

Предложен новый гибридный алгоритм, который позволяет автоматически настраивать параметры алгоритма оптимизации муравьиными колониями для решения задачи предсказания структуры белка.

L.F. Hulianytskyi, S.A. Chornozhuk

THE HYBRID ANT COLONY OPTIMIZATION ALGORITHM FOR PREDICTING THE PROTEIN STRUCTURE

A new hybrid algorithm is proposed, which allows to automatically adjust the parameters of ant colony optimization algorithm for solving protein structure prediction problem.

Список літератури

1. Simoni K.N., Hill R.D., Hill R.L. The thermodynamic hypothesis of protein folding: the work of Christian Anfinsen. *Journal of Biological Chemistry*. 2006. 281,14. 11 p.
2. Dill K.A. Theory for the folding and stability of globular proteins. *Biochemistry*. 1985. 24(6), P. 1501 – 1509.

3. Dorigo M., Stützle T. *Ant colony optimization*. Cambridge (MA): MIT Press. 2004.
4. Гуляницкий Л.Ф., Рудык В.А. 2012. Анализ алгоритмов прогнозирования третичной структуры протеина на базе метода оптимизации муравьиными колониями. *Problems of Computer Intellectualization (Eds. V. Velichko, O. Voloshin, K. Markov). Kiev-Sofia: V.M. Glushkov Institute of Cybernetics, ITHEA.* 2012. P. 152 – 159.
5. Чорножук С.А. Новий алгоритм імітаційного відпалу для передбачення структури білків. *Компьютерная математика.* 2018. С. 118 – 124.
6. Гуляницкий Л.Ф., Рудык В.А. Проблема предсказания структуры протеина: формализация с использованием кватернионов. *Кибернетика и системный анализ*. 2013. № 4. С. 130 – 136.
7. Гуляницький Л.Ф., Мулеса О.Ю. Прикладні методи комбінаторної оптимізації. *Видавничо-поліграфічний центр "Київський університет"*. 2016.
8. Shmygelska A., Hoos H.H. An ant colony optimization algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinformatics*. 2016. N 6(30). P. 30 – 52.
9. 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.

Одержано 19.03.2019