Tomasz GŁOWACKI, Adam KOZAK, Marcin BOROWSKI, Piotr FORMANOWICZ
Politechnika Poznańska

O ALGORYTMACH ASEMBLACJI DŁUGICH ŁAŃCUCHÓW PEPTYDOWYCH[1]

Streszczenie. Znajomość sekwencji długich peptydów pozwala na przewidywanie ich budowy przestrzennej, a co za tym idzie funkcjonalności. Brak bezpośrednich metod do określenia sekwencji długich peptydów powoduje, że potrzebne są metody asemblacji krótkich łańcuchów. W pracy przedstawione są dwa problemy asemblacji i zaproponowane są algorytmy do ich rozwiązania. Przeprowadzono eksperyment obliczeniowy do zbadania skuteczności zaproponowanych metod.

ON ALGORHITMS FOR LONG PEPTIDE ASSEMBLY

Summary. Knowing of peptide sequences allows predicting their structure and therefore their functionality. Lack of direct method to determine long peptides sequences is the reason that assembling methods are needed. In the paper two assembling problems are presented and algorithms to resolve them are proposed. The computational experiment was conducted to evaluate their quality.

1. Wprowadzenie

Nowoczesne nauki przyrodnicze są źródłem dużej ilości danych, które wymagają przetworzenia i interpretacji. Biolodzy i biochemicy coraz częściej zasięgają pomocy informatyków, aby poradzić sobie z ilością i złożoną naturą otrzymanych wyników. Potrzeba współpracy specjalistów nauk przyrodniczych ze specjalistami nauk obliczeniowych zrodziła nową, dynamicznie rozwijającą się naukę – biologię obliczeniową. Ważnym zagadnieniem biologii obliczeniowej jest analiza i ustalanie sekwencji związków chemicznych, ze szczególnym uwzględnieniem sekwencji DNA oraz sekwencji białkowych.

Praca koncentruje się na problemie ustalania sekwencji peptydowych. Peptydy są to wielocząsteczkowe związki składające się z 20 rodzajów aminokwasów. Aminokwasy są połączone ze sobą w długie łańcuchy wiązaniami peptydowymi. Długie peptydy o masie cząsteczkowej powyżej 10000 Daltonów nazywane są białkami. Istnieją metody analityczne, które pozwalają na poznanie krótkich sekwencji aminokwasowych. Poznanie sekwencji aminokwasowych nosi nazwę sekwencjonowania. Przykładam takich metod są degradacja Edmana i spektrometria masowa. Istnieją ograniczenia co do długości sekwencji, które mogą być poznane przy użyciu powyższych metod. W przypadku metody Edmana jest to 50 kolejnych aminokwasów. Składanie krótkich peptydów

[1]Badania częściowo finansowane ze środków projektu badawczego MNiSW nr PBZ-MNiI-2/1/2005.

w jeden łańcuch nosi nazwę asemblacji. Asemblacja łańcuchów peptydowych wymaga zdefiniowania problemów obliczeniowych i modeli kombinatorycznych.

2. Eksperyment chemiczny

Na potrzeby eksperymentu chemicznego zaproponowano wykorzystanie endopeptydaz. Są to enzymy z grupy proteaz, które katalizują trawienie białka. Łańcuch aminokwasów poddany działaniu endopeptydaz ulega podziałowi w dobrze określonych miejscach. W doświadczeniu wykorzystano trypsynę oraz chymotrypsynę[10]. Trypsyna tnie białko na wiązaniu peptydowym po wystąpieniu lizyny lub argininy. Chymotrypsyna tnie białko bezpośrednio po wystąpieniu w łańcuchu tryptofanu, fenyloaniliny lub tyrozyny. Rozkładane są wiązania peptydowe w których grupa karbonylowa należy do wymienionych powyżej aminokwasów. Powstałe krótkie łańcuchy mogą być zsekwencjonowane przy wykorzystaniu metody Edmana lub metod spektrometrycznych. Aby odtworzyć kolejność krótkich łańcuchów w oryginalnej cząsteczce wymagane są dodatkowe informacje. W eksperymencie proponuje się podział białek do dwóch naczyń a następnie w jednym z nich przeprowadza się katalityczne trawienie przy wykorzystaniu trypsyny a w drugim naczyniu katalityczne trawienie z udziałem chymotrypsyny. W ten sposób powstałe krótkie łańcuchy peptydowe częściowo się pokrywają. Składanie kolejnych, częściowo pokrywających się, łańcuchów pozwala na odtworzenie oryginalnej cząsteczki. Eksperyment chemiczny może być źródłem błędów. W przypadku idealnym zakłada się, że cięcia nastąpią w łańcuchu po wystąpieniu każdego z powyżej wymienionych aminokwasów. Taką sytuację nazywa się całkowitym trawieniem. W przypadku, gdy z powodu warunków reakcji część ze spodziewanych cięć nie zachodzi mamy do czynienia z częściowym trawieniem. Opcjonalnie eksperyment pozwala na poznanie rozkładu aminokwasów w oryginalnej cząsteczce. W tym celu przeprowadza się pełne trawienie białka do aminokwasów i mierzy ich stężenie w uzyskanym roztworze. W pracy rozważa się problem całkowitego trawienia oraz częściowego trawienia ze znanym rozkładem aminokwasów.

3. Definicja problemów

W przypadku pełnego trawienia do reprezentacji problemu asemblacji zaproponowano model grafowy G. Krótkie peptydy odpowiadają wierzchołkom tego grafu. Wierzchołki są etykietowane przez te krótkie peptydy. Istnieje łuk w grafie między dwoma wierzchołkami, gdy sufiks poprzednika jest równy prefiksowi następnika. Zaproponowany graf jest grafem dwudzielnym. Istnieje łuk między dwoma wierzchołkami tylko wtedy, gdy odpowiadające im peptydy należą do wyników doświadczeń z różnymi endopeptydazami[1][6]. Można zauważyć, że w przypadku problemu pełnego trawienia jedynie najdłuższe nałożenie dwóch etykiet wierzchołków odzwierciedla poprawne biologiczne zjawisko nakładania się dwóch krótkich peptydów w rozwiązaniu. Oznacza to, że między dowolną parą wierzchołków istnieje co najwyżej 1 łuk. Graf G jest więc 1-grafem. Dodatkowo, dla nałożenia etykiet o długości n, $(n-1)$-sza litera prefiksu powinna odpowiadać aminokwasowi, po którym następuje cięcie. Dzięki powyższej własności można zauważyć, że dla każdych dwóch wierzchołków grafu G lista ich następników jest taka sama lub jest zbiorem pustym. Graf G jest 1-grafem. Rozwiązaniem

problemu asemblacji jest ścieżka Hamiltona w grafie G.

TWIERDZENIE 1:[12]
1-graf $G' = (V, E)$ jest adjointem grafu wtedy i tylko wtedy, gdy dla każdej pary wierzchołków $x, y \in V$ spełniony jest następujacy warunek:

$$N^+(x) \cap N^+(y) \neq \emptyset \Rightarrow N^+(x) = N^+(y) \tag{1}$$

TWIERDZENIE 2:[12] Jeśli G' jest adjointem grafu H, to w H istnieje cykl/droga Eulera wtedy i tylko wtedy, gdy istnieje cykl/ścieżka Hamiltona w G'

WNIOSEK: Graf G jest adjointem. Problem asemblacji dla pełnego trawienia może być rozwiązany w czasie wielomianowym.

W przypadku częściowego trawienia należy rozważyć dowolne możliwe nałożenie dwóch krótkich peptydów. W tym wypadku krótkie peptydy można zdefiniować jako ciągi znaków nad pewnym alfabetem. Zakładając, że znany jest rozkład aminokwasów w łańcuchu peptydowym problem można zdefiniować następująco:
Instancja: Multizbiór S ciągów znaków nad alfabetem Σ (Σ-zbiór wszystkich symboli reprezentujących poszczególne aminokwasy), rozkład D symboli z alfabetu Σ, np. zbiór par (x, i) dla wszystkich symboli x z alfabetu Σ, gdzie i jest liczbą całkowitą dodatnią.
odpowiedź: Superciąg dla multizbioru ciągów S, spełniający rozkład D.

Problem jest problemem NP-trudnym[7]. Do rozwiązania problemu zaproponowano kilka metaheurystyk. W tabeli 1 przedstawiono klasyfikację problemów asemblacji ze względu na informacje otrzymane w eksperymencie chemicznym.

Tabela 1

Klasyfikacja problemów asemblacji

Nr porządkowy	Multizbiór	Wszystkie cięcia	rozkład aminokwasów	złożoność problemu
1	TAK	TAK	NIE	łatwy
2	TAK	NIE	TAK	trudny
3	TAK	TAK	TAK	otwarty
4	NIE	TAK	NIE	otwarty
5	NIE	TAK	TAK	otwarty
6	NIE	NIE	TAK	otwarty
7	NIE	NIE	NIE	otwarty
8	TAK	NIE	NIE	trudny

Praca skupia się na analizie problemu pierwszego i drugiego.

4. Algorytmy

Dla trudnej wersji problemu (problem nr 2) zaimplementowano i przetestowano następujące algorytmy: Metoda Tabu Search, Algorytm ewolucyjny dla dwóch różnych funkcji celu, algorytm zachłanny oraz metodę GRASP. Metoda Tabu Search, algorytm ewolucyjny dla funkcji celu (2) oraz metoda GRASP zostały uprzednio przez autorów opublikowane[5][3]. Algorytm ewolucyjny dla funkcji celu(3) oraz metoda zachłanna są prezentowane po raz pierwszy. W pracy zaprezentowane są wszystkie wyniki, by przeprowadzić dyskusję na temat jakości uzyskanych rozwiązań.

Oceną jakości rozwiązania jest podobieństwo uzyskanej sekwencji do oryginalnej sekwencji poddanej eksperymentowi trawienia. Podobieństwo zostało obliczone za

pomocą metody Needlemana-Wunscha[11]. Jest to metoda dynamicznego programowania oceniająca globalne podobieństwo dwóch sekwencji.

Metoda Tabu Search jest jedną z najpopularniejszych metaheurystyk do rozwiązywania problemów optymalizacyjnych. Tabu Search jest rodzajem lokalnego przeszukiwania[8][9]. Z rozwiązania i konstruowane jest rozwiązanie $j \in N(i)$, Gdzie $N(i)$ oznacza sąsiedztwo rozwiązania i. Następnie sprawdzane jest, czy rozwiązanie j spełnia warunek stopu, czy należy przeszukiwanie nadal wykonywać. Istnieją dodatkowe metody podwyższające efektywność takiego przeszukiwania, poprzez zapobieganie wpadaniu w lokalne minimum. Jedną z nich jest wykonywanie losowych ruchów, które pozwolą rozpocząć przeszukiwanie przestrzeni w innym miejscu. Inną metodą zapobiegającą wpadaniu w lokalne minimum jest lista tabu. Lista tabu zawiera ruchy zabronione. Lista ta zawiera kilka ostatnio wykonanych ruchów. Rozwiązaniem początkowym jest losowa permutacja elementów multizbioru S i losowy wybór nałożenia dla każdych dwóch sąsiadujących elementów z możliwych nałożeń[5]. Brak możliwego nałożenia pomiędzy dwoma elementami S jest traktowany jako nałożenie o wartości 0. Wykonanie ruchu polega na zamianie dwóch elementów zbioru S miejscami i wyboru nałożenia między tymi elementami a ich sąsiadami. Jako sąsiedztwo $N(i)$ rozwiązania i zdefiniowano wszystkie permutacje, w których zamieniono miejscami 2 elementy multizbioru S i wybrano ich nowe nałożenie z sąsiadami. Jako ruch wybierany jest przejście do rozwiązania ze zbioru $N(i)$ o najlepszej wartości funkcji oceny heurystycznej. Jeśli ruch znajduje się na liście Tabu, to wykonujemy go tylko wtedy, gdy powoduje przejście do lepszego rozwiązania niż aktualne. Funkcję oceny heurystycznej dla rozwiązania l zdefiniowano jako taksówkową odległość rozkładu otrzymanego łańcucha od rozkładu D:

$$f(l) = \sum_{i=1}^{20} | x_i - y_i | \tag{2}$$

gdzie x_i oznacza liczbę i-tej litery w ocenianym rozwiązaniu l a y_i oznacza liczbę i-tej litery szukanej dla dystrybucji D. Funkcja f jest minimalizowana.

Algorytm Ewolucyjny to algorytm, który symuluje zjawisko ewolucji biologicznej: krzyżowanie się osobników populacji oraz mutację. W omawianym problemie osobnikiem populacji jest permutacja elementów zbioru S wraz z nałożeniami sąsiadujących ze sobą elementów. Zauważono, że preferowanie nałożeń o niezerowej wadze wpływa na jakość rozwiązania, ponieważ nałożenia o wadze „0” nie odzwierciedlają biologicznego nakładania się. Obserwację tą wykorzystano przy tworzeniu populacji oraz w zaproponowanym operatorze krzyżowania. Poniższy pseudokod przedstawia działanie operatora:

```
Wybierz 2 osobniki X_i oraz X_j do krzyżowania według zasad
selekcji ruletkowej
T := {wszystkie wspólne podciągi X_i i X_j}
New_Solution = pobierz losowo element z T
For i:=2 to rozmiar(T) do
Begin
    New_Solution = NULL;
    Pobierz losowo element z T;
    {Wstaw go na takiej losowej pozycji j w New_Solution,
    że element j − 1 ORAZ j + 1 mają niezerowe nałożenia z
    dodawanym elementem} gdy niemożliwe to
        {Wstaw go na takiej losowej pozycji j w
```

```
        New_Solution, że istnieje niezerowe nałożenie z
        elementem j−1 lUB elementem j+1} gdy niemożliwe to
            {Wstaw go na losowej pozycji w New_Solution};
end;
```

Algorytm ewolucyjny został zbadany dla dwóch różnych funkcji oceny heurystycznej f (2) i f(3):

$$f'(l) = \sum_{i=1}^{20} x_i - \sum_{i=1}^{20} y_i \tag{3}$$

Funkcja f' ocenia jakość uzyskanego rozwiązania, przy założeniu, że optymalizowana jest jedynie długość rozwiązania, bez uwzględnienia rozkładu D. Funkcja f' jest różnicą długości szukanego rozwiązania od rozwiązania uzyskanego.

Prawdopodobieństwo wyboru do krzyżowanie $l-tego$ elementu populacji L wynosi:

$$P(l) = \frac{f_{\max} - f(l)}{\sum\limits_{m \in L} (f_{\max} - f(m))} \tag{4}$$

gdzie f_{max} oznacza ocenę najgorszego rozwiązania w populacji. Operator krzyżowania buduje listę wszystkich wspólnych podciągów dla dwóch wybranych rozwiązań. Następnie buduje się nowe rozwiązanie dodając do rozwiązania częściowego kolejne podciągi. Kolejne podciągi są dodawane losowo, z tym że preferuje się ich niezerowe nałożenia z pozostałymi elementami tworzonego rozwiązania. Mutacja jest zamianą dwóch elementów zbioru S i wybraniem losowego nałożenia tych elementów z elementami sąsiednimi.

Dla omawianego problemu zaimplementowano algorytm zachłanny. Na początku działania algorytmu rozwiązanie jest puste. Na każdym kroku dodawany jest do niego kolejny element. Tworzona jest lista najlepszych możliwych ruchów, uaktualniana na każdym etapie podejmowania decyzji. Spośród dostępnych ruchów wybierany jest jeden ruch losowo. W problemie asemblacji ruch zdefiniowano jako dodanie elementu z multizbioru S do rozwiązania na dowolnej pozycji j i wybraniu dowolnego nałożenia nowego elementu rozwiązania z ciągami znaków $j-1$. Funkcje oceny ruchu zdefiniowano jako długość uzyskanego w ten sposób nałożenia.

Dla omawianego problemu zaimplementowano metodę GRASP - Greedy Randomized Adaptive Search Procedure. GRASP bazuje na znajdowaniu dobrego rozwiązania początkowego i późniejszej lokalnej optymalizacji tego rozwiązania[4]. Dla rozwiązania uzyskanego przez powyższy algorytm zachłanny przeprowadzono lokalną optymalizację. Losowo wybierany jest ruch - zamiana dwóch elementów S miejscem i wybór nowego nałożenia miedzy tymi elementami a ich sąsiadami. Ruch jest wykonywany tylko wtedy, gdy prowadzi do polepszenia funkcji oceny heurystycznej (2). Algorytm zatrzymuje się, jeśli nie istnieje ruch, który mógłby polepszyć istniejące rozwiązanie.

5. Wyniki

Algorytmy zostały zaimplementowane w języku Java 1.5 i przetestowane na komputerze klasy PC z procesorem Intel 2xXenon 3.6 GHz z 4 GB RAM. Dla potrzeby testów wykorzystano zbiór 150 prawdziwych peptydów, który został podzielony na 15 podzbiorów o liczności 10 peptydów każdy, o długościach kolejno: 100, 150, 200, 250, 300 aminokwasów i zawierających kolejno 1,2 lub 3 błędy wynikające z braku cięć. Parametry obu Algorytmów Genetycznych dobrano eksperymentalnie:

- Wielkość populacji 10000
- Liczba iteracji 10000
- Prawdopodobieństwo mutacji 0.05

Parametry algorytmu Tabu Serach dobrano eksperymentalnie:

- Długość listy Tabu 10
- Liczba ruchów bez polepszenia rozwiązania 10
- Liczba losowych ruchów 5
- Liczba losowych serii 3
- Liczba restartów 10

Parametry algorytmu GRASP dobrano eksperymentalnie:

- RLC zawiera rozwiązania nie gorsze o więcej niż 35% od rozwiązania optymalnego
- RLC zawiera nie więcej niż 30 % wszystkich rozwiązań

Algorytmy wykonano dla każdej instancji danych 10 razy a wyniki uśredniono. Wyniki dla algorytmów ewolucyjnych i Tabu przedstawiono w Tabeli 2. Wyniki dla algorytmu zachłannego i metody GRASP opartej na tym algorytmie zostały przedstawione w Tabeli 3.

Tabela 2

Wyniki eksperymentu obliczeniowego

Sekwencja	Liczba błędów	Metoda Tabu		Algorytm Ewolucyjny(f)		Algorytm ewolucyjny (f')	
		podobieństwo[%]	czas[s]	podobieństwo[%]	czas[s]	podobieństwo[%]	czas[s]
100	1	81.33	6.07	82.58	1.87	76.51	1.002
100	2	85.73	1.76	87.19	1.93	80.17	0.993
100	3	65.66	2.11	88.92	1.89	79.21	0.986
150	1	70.22	15.39	83.54	1.73	77.17	0.992
150	2	73.89	14.73	81.17	2.02	75.14	0.964
150	3	74.19	12.62	85.67	1.98	81.37	1.034
200	1	56.67	86.49	80.19	1.86	70.18	1.087
200	2	58.91	58.58	81.12	1.92	72.86	0.995
200	3	62.41	51.43	84.56	2.01	76.59	0.918
250	1	49.22	150.46	81.86	1.89	72.18	0.892
250	2	49.54	154.40	79.34	1.99	71.28	0.944
250	3	54.23	93.04	83.27	1.94	76.15	0.908
300	1	45.88	289.12	83.40	1.96	72.36	0.928
300	2	47.98	289.48	80.96	2.02	71.67	0.919
300	3	44.58	250.75	81.02	2.11	73.18	1.122

Tabela 3

Wyniki eksperymentu obliczeniowego

Sekwencja	Liczba błędów	Algorytm zachłanny		Metoda GRASP	
		podobieństwo[%]	czas[s]	podobieństwo[%]	czas[s]
100	1	76.75	0.011	82.17	0.87
100	2	84.21	0.015	87.56	0.921
100	3	84.48	0.015	89.17	0.923
150	1	77.73	0.037	85.94	1.078
150	2	80.49	0.03	90.17	1.007
150	3	75.38	0.012	75.38	1.144
200	1	65.78	0.017	74.95	1.125
200	2	70.02	0.055	81.02	1.117
200	3	64.73	0.012	72.17	1.103
250	1	63.51	0.019	63.5136	1.435
250	2	60.14	0.01	63.14	1.489
250	3	63.95	0.09	63.95	1.397
300	1	62.33	0.019	65.18	1.642
300	2	64.72	0.017	69.29	1.598
300	3	60.82	0.018	60.93	1.572

6. Podsumowanie i wnioski

W pracy przedstawiono metodę asemblacji długich łańcuchów peptydowych. Zaproponowano wielomianowe rozwiązanie dla problemu asemblacji dla pełnego trawienia. Rozwiązaniem problemu jest ścieżka Hamiltona w zaproponowanym w pracy grafie G. Graf ten jest adjointem pewnego grafu. Szukanie ścieżki Hamiltona w takim gafie jest może być rozwiązane przez szukanie ścieżki Eulera w grafie związanym z tym grafem.

Dla problemu trudnego zaprojektowano i zaimplementowano pięć algorytmów. Metody zostały przetestowane na rzeczywistych łańcuchach peptydowych. Otrzymane wyniki pokazują, że algorytm ewolucyjny dla funkcji f znacznie przewyższa pozostałe metody.

Ciekawą obserwacją jest, że algorytm tworzący zachłanne rozwiązanie zwraca lepsze wyniki niż algorytm przeszukiwania Tabu Search. Uzyskane przez te algorytmy wyniki to kolejno 70,34% oraz 61,36%. W metodzie Tabu, która zwraca najsłabsze wyniki, wykonywane są losowe ruchy dążące do możliwie najszybciej optymalizacji rozwiązania. Metoda ta w żaden sposób nie wykorzystuje dodatkowej wiedzy o problemie. Można zauważyć, że istnienie niezerowych - a szczególnie długich nałożeń między elementami S bardzo często odzwierciedla rzeczywiste nałożenie się tych łańcuchów w oryginalnej cząsteczce. Prawdopodobieństwo istnienia nałożenia o długości n dla zbioru S, które nie odzwierciedla prawdziwego nałożenia się krótkich peptydów w odpowiadającym zbiorowi S spektrum wyraża się wzorem:

$$p(n) = \frac{1}{20^{n+1}} \tag{5}$$

Prawdopodobieństwo istnienia nałożenia o długości n między dwoma elementami jest więc równe prawdopodobieństwu, że n ostatnich liter poprzedzającego elementu jest równe n pierwszym literom kolejnego elementu, a n-1 litera poprzedzającego elementu symbolizuje aminokwas, po którym następuje cięcie.

Wykorzystanie powyższej obserwacji znacząco wpłynęło na jakość otrzymanych rozwiązań. Własność tą wykorzystano w algorytmie GRASP - poprzez zachłanny wybór jednego spośród możliwie najdłuższych nałożeń. Zaprojektowany przez autorów operator krzyżowania algorytmu ewolucyjnego także preferuje długie nałożenia się łań-

cuchów. Dodatkowo cechą, którą odziedziczają osobniki potomne w algorytmie ewolucyjnym są wspólne fragmenty łańcuchów dla obojga rodziców. Obie te własności algorytmu ewolucyjnego przyczyniają się do powstania w trakcie symulowanej ewolucji wielu wspólnych podścieżek o niezerowych nałożeniach pomiędzy kolejnymi elementami.

W trakcie eksperymentu obliczeniowego zbadano wpływ funkcji celu w algorytmie ewolucyjnym na jakość otrzymanego rozwiązania, rozumianego jako globalne podobieństwo uzyskanej sekwencji do sekwencji szukanej. Wyniki pokazują, że optymalizacja długości uzyskanego łańcucha daje słabsze wyniki niż wykorzystanie wiedzy o rozkładzie D. Uzyskano odpowiednio 75,07% oraz 82,99%.

LITERATURA

1. Błażewicz J., Borowski M., Formanowicz P., Głowacki T.: On graph theoretical models for peptide sequence assembly, Foundations of Computing and Decision Sciences 30 (2005) p. 183–191.
2. Błażewicz J., Borowski M., Formanowicz P, Głowacki T.: Genetic and tabu search algorithms for peptide assembly problem, RAIRO - Operations Research, 44 (2010) p. 153–166
3. Głowacki T., Kozak A., Formanowicz P.: Asemblacja długich łańcuchów peptydowych przy wykorzystaniu metaheurystyki GRASP, Zeszyty Naukowe Politechniki Śląskiej z. 150, 2008, p. 203–209.
4. Resende M. G. C., Ribeiro C. C.: Greedy randomized adaptive search procedures, Handbook of Metaheuristics, Kluwer Academic Publishers, 2003, p. 219-–249 Stryer L.,
5. Błażewicz J., Borowski M., Formanowicz P., Stobiecki M.: Tabu Search Method for Determining Sequences of Amino Acids in Long Polypeptides, Lecture Notes in Computer Science 3449 (2005) p. 22–32.
6. Formanowicz P.: Selected Combinatorial Aspects of Biological Sequence Analysis, Poznań, Publishing House of Poznań University of Technology 2005.
7. Gallant J. K.: The complexity of the overlap method for sequencing biopolymers, Journal of Theoretical Biology 101 (1983) p. 1–17.
8. Glover F.: Tabu Search, Part I, ORSA Journal on Computing 1 (1989) p. 190–206.
9. Glover F.: Tabu Search, Part II, ORSA Journal on Computing 1 (1990) p. 4–32.
10. Stryer L., Biochemistry, 4th edition, New York, W.H. Freeman and Company, 1995.
11. Needleman S. B., Wunsch, Ch. D.: A general method applicable to the search for similarities in the amino acid sequence of two protein Journal of Molecular Biology 48 (1970) p. 443—453.
12. Błażewicz J., Hertz A., Kobler D., de Werra D.: On some properties of DNA graphs. Discrete Applied Mathematics 98, 1999, p. 1–19.

Recenzent: Dr hab. inż. Cezary Recenzent, prof. Pol. PL.

Abstract

Peptide sequencing methods allow to recognize only short sequences. There is a need to bring together many short sequences to reconstruct the oryginal sequence. The methods for peptide sequences assembling with and without errors in digestion phase are considered. It is shown that the problem without errors in digestion phase is easy. The problem when not all supposed cuts occurred is known to be NP-hard. Three metaheuristics are proposed to resolve the NP-hard version of the problem. Tabu search method, evolution algorithm and GRASP were implemented and tested on real peptides sequences. The Greedy Algorithm for peptides assembling is also presented and tested. The experiment clearly shows that Evolution algorithm is better than others. The explanation and discussion of the results is provided.