

KOMPUTERY DNA

Piotr Formanowicz

Wprowadzenie

Komputery DNA (obliczenia oparte na DNA, DNA computing) są alternatywną metodą przeprowadzania obliczeń.

Oparta jest ona na obserwacji, że możliwe jest zaprojektowanie **serii eksperymentów biochemicznych** wykonywanych na **cząsteczkach DNA**, która jest równoważna wykonaniu pewnych obliczeń.

W klasycznych komputerach elektronicznych elementami przetwarzającymi i przechowującymi informacje są bramki logiczne. Wykonanie odpowiedniej sekwencji elementarnych operacji przetwarzania i przechowywania informacji jest równoważne przeprowadzeniu pewnych obliczeń.

W przypadku komputerów DNA jest podobnie, z tym że elementami przetwarzającymi i przechowującymi informację nie są elektroniczne bramki logiczne, lecz **cząsteczki DNA**.

Oznacza to, że zbiór podstawowych operacji w komputerach DNA jest inny niż w komputerach elektronicznych, ale otrzymywane wyniki mogą być podobne.

Ponadto, (niektóre) podejścia do obliczeń wykonywanych za pomocą cząsteczek DNA charakteryzują się naturalną masywną równoległością, która może prowadzić do metod umożliwiających rozwiązywanie problemów trudnych obliczeniowo.

Pierwszą osobą, która rozważała przeprowadzanie obliczeń na poziomie molekularnym był **Richard Feynman**.

Jego idee musiały czekać około 20 lat na realizację i rozwinęły się w dwóch zasadniczych kierunkach, tj. **komputerów DNA i komputerów kwantowych**.

Badania w dziedzinie komputerów DNA zostały zapoczątkowane przez Leonarda Adlemana w 1994 roku publikacją w *Science* pt. “*Molecular Computation of Solutions to Combinatorial Problems*”, w której przedstawił oparty na DNA algorytm rozwiązujący problem znajdowania **skierowanej ścieżki Hamiltona między dwoma wierzchołkami**.

Podejście zaproponowane przez Adlemana charakteryzują się bardzo dużą masywną równoległością, ponieważ olbrzymia liczba cząsteczek DNA kodujących różne “propozycje” rozwiązań jest przetwarzana

równoległe.

Wkrótce po publikacji Adlemana zostały zaproponowane algorytmy rozwiązujące wiele innych problemów kombinatorycznych.

Idea komputerów DNA

Natura cząsteczek DNA ma zasadnicze znaczenie dla możliwości wykorzystania ich do przeprowadzania obliczeń.

Cząsteczka DNA składa się z dwóch nici, z których każda może być postrzegana jako słowo nad alfabetem $\Sigma_{DNA} = \{A, C, G, T\}$.

Jedną z najważniejszych własności cząsteczek DNA, zarówno dla ich funkcji pełnionych w naturze, jak i w komputerach DNA, jest zdolność nukleotydów do łączenia się za pomocą wiązań wodorowych.

Powstają komplementarne pary nukleotydów A-T połączone za pomocą dwóch wiązań wodorowych oraz pary C-G połączone za pomocą trzech wiązań.

W ramach jednej nici kolejne nukleotydy połączone są za pomocą wiązań fosfodiesterowych, natomiast komplementarne pary nukleotydów z dwóch łańcuchów tworzących cząsteczkę DNA połączone są za pomocą wiązań wodorowych, które są słabsze niż wiązania fosfodiesterowe.

Wynika stąd, że dwie jednoniciowe cząsteczki 5'-TTGCATTAACGAC-3' i 3'-AACGTAATTGCTG-5' mogą utworzyć cząsteczkę dwuniciową

5'-TTGCATTAACGAC-3'
3'-AACGTAATTGCTG-5'

(Cząsteczki DNA mają orientację.)

W komputerach DNA instancja jest zakodowana w jednoniciowych krótkich cząsteczkach DNA, a rozwiązanie jest zakodowane w utworzonych z nich dłuższych cząsteczkach dwuniciowych.

Koncepcja komputerów DNA zaproponowana przez Adlemana jest do pewnego stopnia zbliżona do zasady funkcjonowania **niedeterministycznej maszyny Turinga**.

Na początku tworzona jest bardzo duża liczba (dwuniciowych) cząsteczek kodujących wszystkie możliwe rozwiązania danego problemu (nie tylko optymalne) oraz bardzo dużo cząsteczek, które nie kodują żadnego rozwiązania.

W następnych krokach algorytmu z początkowego zbioru cząsteczek kolejno są eliminowane te, które nie kodują żadnego rozwiązania oraz te, które nie kodują

rozwiązania optymalnego.

Na końcu działania algorytmu w zbiorze przetwarzanych cząteczek pozostają tylko te, które kodują rozwiązanie optymalne.

Z takiego schematu działania wynika siła komputerów DNA – **(prawie) wszystkie sekwencje kodujące są przetwarzane równolegle.**

Eliminacja “złych” sekwencji jest na ogół przeprowadzana w ten sposób, że algorytm sprawdza kolejno, czy sekwencje ze zbioru wejściowego posiadają wszystkie własności, które musi posiadać sekwencja kodująca rozwiązanie optymalne. Jeżeli jakaś sekwencja nie posiada przynajmniej jednej z tych własności, jest eliminowana.

W przypadku komputerów DNA, inaczej niż w przypadku klasycznych komputerów elektronicznych, nie ma ścisłego oddzielenia sprzętu od oprogramowania.

Ponadto, dla każdego rozwiązywanego problemu i tworzonego algorytmu musi być zaprojektowany nowy komputer DNA (dotąd nie została zaproponowana standardowa uniwersalna architektura komputerów DNA).

Wynika stąd m.in., że bardzo istotny staje się problem zaprojektowania odpowiednich jednoniciowych cząteczek DNA kodujących instancję rozwiązywanego problemu, takich, że po umieszczeniu ich w jednym roztworze utworzą cząsteczki dwuniciowe kodujące jego rozwiązanie.

Ogólny przykład

Rozważmy hipotetyczny (ogólny) problem kombinatoryczny, w którym poszukiwana jest pewna konkretna permutacja elementów. Załóżmy, dla uproszczenia, że instancja zawiera tylko cztery elementy, tj. x, y, w, z , a rozwiązaniem optymalnym jest permutacja $xwzy$.

Jeden z możliwych sposobów zakodowania tej instancji jest następujący:

$x : 5' - \text{ACCGATTA} - 3', y : 5' - \text{GTACCATT} - 3',$
 $w : 5' - \text{TGAACCTA} - 3', z : 5' - \text{AATTCGCG} - 3'.$

Ten zbiór oligonukleotydów nie wystarcza do tego, by uformowała się dwuniciowa cząsteczka kodująca rozwiązanie. Potrzebne są jeszcze dodatkowe oligonukleotydy, które będą mogły hybrydyzować do prawej połowy pewnych spośród wymienionych oligonukleotydów i do lewej połowy innych spośród nich.

Oligonukleotydy te będą łączyć oligonukleotydy kodujące elementy instancji w dwuniciowej cząsteczce DNA kodującej potencjalne rozwiązanie.

Dla przedstawionego wcześniej zbioru oligonukleotydów takie “łączące” oligonukleotydy powinny mieć następującą postać:

$x - y : 3' - \text{TAATCATG} - 5',$
 $x - w : 3' - \text{TAATACTT} - 5',$
 $x - z : 3' - \text{TAATTTAA} - 5',$
 $y - x : 3' - \text{GTAATGGC} - 5',$
 $y - w : 3' - \text{GTAAACTT} - 5',$
 $y - z : 3' - \text{GTAATTAA} - 5',$
 $w - x : 3' - \text{GGATTGGC} - 5',$
 $w - y : 3' - \text{GGATCATG} - 5',$
 $w - z : 3' - \text{GGATTTAA} - 5',$
 $z - x : 3' - \text{GCGCTGGC} - 5',$
 $z - y : 3' - \text{GCGCCATG} - 5',$
 $z - w : 3' - \text{GCGCACTT} - 5'$

Gdy wszystkie wymienione oligonukleotydy (tj. kodujące instancję i “łączące”) zostaną umieszczone w roztworze w bardzo wielu kopiach, zaczną ze sobą hybrydyzować i będą powstawać cząsteczki dwuniciowe o różnych długościach, tj. zawierające różne liczby oligonukleotydów.

Wśród nich będą następujące przykładowe duplekisy:

(yx)	$5' - \text{GTACCATTACCGATTA} - 3'$ $3' - \text{GTAATGGC} - 5'$
$(xzw x)$	$5' - \text{ACCGATTAAATTCGCGTGAACCTAACCGATTA} - 3'$ $3' - \text{TAATTTAAGCGCACTTGGATTGGC} - 5'$
$(xwzy)$	$5' - \text{ACCGATTATGAACCTAAATTCGCGGTACCATT} - 3'$ $3' - \text{TAATACTTGGATTTAAGCGCCATG} - 5'$
(wzy)	$5' - \text{TGAACCTAAATTCGCGGTACCATT} - 3'$ $3' - \text{GGATTTAAGCGCCATG} - 5'$
$(zxw yx)$	$5' - \text{AATTCGCGACCGATTATGAACCTAGTACCATTACCGATTA} - 3'$ $3' - \text{GCGCTGGCTAATACTTGGATCATGGTAATGGC} - 5'$
(wxz)	$5' - \text{TGAACCTAACCGATTAAATTCGCG} - 3'$ $3' - \text{GGATTGGCTAATTTAA} - 5'$

Powyższe duplekisy (i wiele innych) są tworzone w pierwszym kroku algorytmu.

W kolejnych krokach eliminowane są duplekisy, które nie mogą kodować rozwiązania.

Usunięte muszą zostać duplekisy składające się z mniej niż czterech oligonukleotydów, ponieważ nie mogą one kodować permutacji wszystkich elementów instancji.

Następnie należy usunąć duplekisy, które zawierają więcej niż cztery oligonukleotydy, ponieważ muszą

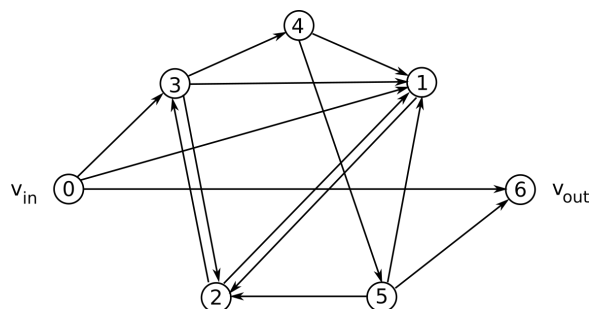
one zawierać co najmniej jedno powtórzenie elementu instancji, co jest niedopuszczalne.

Należy też usunąć duplekсы, które zawierają powtórzenia, ale mają właściwą długość.

Ponadto, w ogólności musi istnieć możliwość wykrycia i usunięcia dupleksów kodujących rozwiązania, które nie maksymalizują (lub minimalizują) wartości funkcji celu występującej w danym problemie. Ten krok (kroki) jest na ogół silnie zależny od konkretnego problemu, który jest rozwiązywany.

Algorytm Adlemana

Algorytm zaproponowany przez Adlemana rozwiązuje problem znalezienia skierowanej ścieżki Hamiltona między dwoma zadanymi wierzchołkami.



Rysunek 1: Instancja problemu rozwiązywanego za pomocą algorytmu Adlemana

ALGORYTM ADLEMANA:

1. Wygeneruj wszystkie ścieżki w grafie.
2. Sprawdź czy dana ścieżka zaczyna się w wierzchołku v_{in} i kończy w wierzchołku v_{out} . Jeżeli nie, to ją usuń.
3. Sprawdź czy dana ścieżka przechodzi przez dokładnie 7 wierzchołków. Jeżeli nie, to ją usuń.
4. Sprawdź czy dana ścieżka zawiera każdy wierzchołek grafu. Jeżeli nie, to ją usuń.
5. Jeżeli w zbiorze pozostała jakaś ścieżka, jest to poszukiwana ścieżka Hamiltona. Jeżeli nie, to dla danej instancji rozwiązanie nie istnieje.

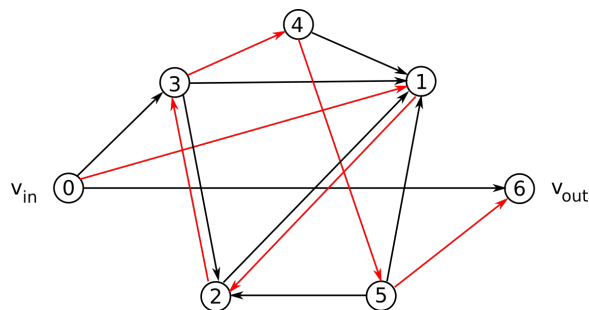
Pierwszy krok algorytmu realizowany jest przez umieszczenie w roztworze wszystkich oligonukleotydów kodujących wierzchołki grafu oraz oligonukleotydów "łączących", które w tym przypadku kodują luki w grafie. Do utworzenia trwałych powiązań między kolejnymi oligonukleotydami w jednej nici DNA wykorzystywana jest ligaza.

Drugi krok jest realizowany za pomocą reakcji PCR ze starterami odpowiadającymi wierzchołom v_{in} i v_{out} .

Trzeci krok może być zrealizowany za pomocą elektroforezy na żelu, która umożliwia zidentyfikowanie cząsteczek o długości odpowiadającej siedmiu wierzchołkom.

Krok czwarty może być zrealizowany za pomocą sond komplementarnych do oligonukleotydów kodujących poszczególne wierzchołki.

Krok piąty (stwierdzenie, czy w roztworze pozostały jakiegokolwiek cząsteczki DNA) może być zrealizowany za pomocą reakcji PCR i elektroforezy.



Rysunek 2: Rozwiązanie problemu rozwiązywanego za pomocą algorytmu Adlemana

Uwagi końcowe

W praktyce maszynowa równoległość jest ograniczona szybko rosnącą liczbą cząsteczek potrzebnych do utworzenia wszystkich łańcuchów kodujących.

Istotna jest konieczność odpowiedniego zaprojektowania oligonukleotydów kodujących instancję, by uniknąć nieporządkanych hybrydyzacji.

Każdy algorytm oparty na DNA jest jednocześnie specjalizowanym komputerem DNA przeznaczonym do rozwiązania jednego problemu.

Istnieją inne modele obliczeń opartych na DNA, np. układy samoskładające się i automaty Shapiro.

Perspektywy – komputery uniwersalne czy raczej nanoautomaty przeznaczone do wykonywania ściśle określonych operacji?

Literatura

- [1] L. M. Adleman. Molecular Computation of Solutions to Combinatorial Problems. *Science*, 266, 1994, 1021–1024.
- [2] P. Formanowicz. DNA Computing. *Computational Methods in Science and Technology*, 11, 2005, 11–20.

