Politechnika Poznańska

Wydział Informatyki
Instytut Informatyki

Rozprawa doktorska

Modele grafowe i algorytmy dla klasycznego problemu sekwencjonowania DNA przez hybrydyzację oraz dla jego odmiany z informacją o powtórzeniach

Autor:
Kamil Kwarciak

Promotor:
dr hab. inż
Piotr Formanowicz, prof. PP

22 kwietnia 2013

Serdecznie dziękuję dr. hab. inż. Piotrowi Formanowiczowi,
profesorowi PP za wskazanie ciekawego problemu badawczego oraz
pomoc udzieloną w trakcie prowadzonych badań.

Spis treści

Spis rysunków

Spis tabel

Skróty

ACO	Algorytm kolonii mrówek (ang. ***A**nt **C**olony **O**ptimization algorithm*)
ATSP	Problem komiwojażera w grafie skierowanym zwany również asymetrycznym problemem komiwojażera (ang. ***A**symmetric **T**ravelling **S**alesman **P**roblem*)
DNA	Kwas deoksyrybonukleinowy (ang. ***D**eoxyribo**N**ucleic **A**cid*)
DTM	Deterministyczna maszyna Turinga (ang. ***D**eterministic **T**uring **M**achine*)
ISBH	Izotermiczne sekwencjonowanie przez hybrydyzację (ang. ***I**sothermic **S**equencing **B**y **H**ybridization*)
ML-ACO	Wielopoziomowy algorytm kolonii mrówek (ang. ***M**ulti-**L**evel **A**nt **C**olony **O**ptimization algorithm)*
NDTM	Niedeterministyczna maszyna Turinga (ang. ***N**on-**D**eterministic **T**uring **M**achine)*
PSBH	Pozycyjne sekwencjonowanie przez hybrydyzację (ang. ***P**ositional **S**equencing **B**y **H**ybridization)*
RNA	Kwas rybonukleinowy (ang. ***R**ibo**N**ucleic **A**cid*)
SBH	Sekwencjonowanie przez hybrydyzację (ang. ***S**equencing **B**y **H**ybridization)*
SNP	Polimorfizm pojedynczego nukleotydu (ang. ***S**ingle **N**ucleotide **P**olymorphism*)
TSP	Problem komiwojażera (ang. ***T**ravelling **S**alesman **P**roblem*)

Oznaczenia

l	długość oligonukleotydów wchodzących w skład spektrum $S(Q)$
G	graf (skierowany lub nieskierowany)
Q	sekwencja DNA
$S(Q)$	zbiór słów reprezentujących spektrum sekwencji Q
$S^{(im)}(Q)$	zbiór słów reprezentujących idealne multispektrum sekwencji Q
$S^{(is)}(Q)$	zbiór słów reprezentujących idealne spektrum sekwencji Q
$S^{(m)}(Q)$	zbiór słów reprezentujących multispektrum sekwencji Q

Mojej żonie Dorocie, za cierpliwość i wsparcie.

Rozdział 1

Wprowadzenie

1.1 Kontekst prowadzonych prac

Sekwencjonowanie DNA jest jednym z najbardziej istotnych problemów biologii molekularnej i obliczeniowej. Celem jest ustalenie sekwencji nukleotydów wchodzących w skład łańcucha DNA. Zazwyczaj zapisuje się go jako sekwencję czterech liter: A, C, G i T. Reprezentują one poszczególne nukleotydy, z których składa się cząsteczka DNA i odpowiadają kolejno adeninie, cytozynie, guaninie i tyminie.

Istnieje wiele metod uzyskania tej informacji. Jedną z nich jest sekwencjonowanie przez hybrydyzację [4, 44]. Metoda ta składa się z dwóch etapów. Pierwszy z nich to eksperyment biochemiczny. Jego wynikiem jest zbiór oligonukleotydów (tj. krótkich sekwencji DNA), które stanowią fragmenty oryginalnej sekwencji DNA. Wszystkie oligonukleotydy mają tą samą długość l i są nazywane l-merami. Drugi etap (obliczeniowy) sprowadza się do rekonstrukcji analizowanej sekwencji na podstawie informacji o l-merach.

Do przeprowadzenia eksperymentu w pierwszym etapie wykorzystywane są chipy DNA [51, 53] zawierające pełną bibliotekę oligonukleotydów o długości l. Taki chip DNA podzielony jest na komórki zawierające pewną liczbę identycznych, jednoniciowych łańcuchów DNA o długości l. W ramach eksperymentu chip jest umieszczany w roztworze zawierającym wiele kopii badanej sekwencji w postaci jednoniciowego łańcucha DNA. Oligonukleotydy znajdujące się na chipie hybrydyzują z komplementarnymi fragmentami sekwencji z przygotowanego roztworu. Poszczególne kopie analizowanego DNA zostają wcześniej radioaktywnie lub fluoroscencyjnie oznakowane, dzięki czemu istnieje możliwość otrzymania dla danego chipu obrazu, który prezentuje l-mery wchodzące w skład badanej sekwencji.

W idealnym przypadku etap biochemiczny dostarcza pełnej i poprawnej informacji o oligonukleotydach wchodzących w skład analizowanej sekwencji DNA. Otrzymany zbiór l-merów nazywany jest *spektrum*. Jednak w praktyce w trakcie eksperymentu pojawiają się pewne błędy. Mogą one być dwojakiego rodzaju: pozytywne i negatywne. Błąd pozytywny występuje, gdy analizowana sekwencja ulega hybrydyzacji z oligonukleotydem na chipie, który nie jest w pełni komplementarny. W takiej sytuacji spektrum zawiera pewne dodatkowe l-mery, które nie są fragmentami badanej sekwencji. Możliwa jest również sytuacja odwrotna. Analizowana sekwencja może nie zhybrydyzować do komplementarnego oligonukleotydu na chipie. W rezultacie spektrum nie zawiera kompletnej

informacji o l-merach wchodzących w skład analizowanej sekwencji. Te brakujące oligonukleotydy to błędy negatywne. Innym źródłem błędów negatywnych są powtórzenia występujące w analizowanej sekwencji. W klasycznym podejściu spektrum jest zbiorem a nie multizbiorem, więc jeżeli w badanym DNA występują powtórzenia o długości co najmniej l, to taki przypadek nie zostanie wykryty.

Rozwiązywany w drugim etapie problem kombinatoryczny może zostać sprowadzony do grafowego problemu Orienteering [35] w grafie skierowanym. W problemie tym dany jest skierowany graf wejściowy, funkcja kosztu łuków, wartość każdego wierzchołka oraz pewien budżet. Celem jest znalezienie w grafie wejściowym skierowanej ścieżki o koszcie nie większym niż zadany budżet, której wartość wyznaczona przez sumę wartości odwiedzonych wierzchołków będzie maksymalna.

Powyższy problem kombinatoryczny jest trudny obliczeniowo. Należy do klasy problemów silnie NP-trudnych (definicja w Rozdziale 4.1), a więc nie istnieje algorytm, za pomocą którego można by go rozwiązać w czasie wielomianowym (przy założeniu, że $P \neq NP$). Dla tego typu problemów czas uzyskania rozwiązania optymalnego rośnie wykładniczo w stosunku do rozmiaru instancji problemu. W związku z tym algorytmy dokładne mają dość ograniczone zastosowanie w praktyce i zamiast nich wykorzystuje się algorytmy przybliżone, które umożliwiają uzyskanie rozwiązania w czasie wielomianowym. Szczególnym przypadkiem takiego algorytmu jest algorytm aproksymacyjny. Posiada on gwarancję jakości rozwiązania, dzięki czemu istnieje możliwość określenia o ile gorsze od optimum będzie rozwiązanie otrzymane przy jego użyciu.

Wynikiem etapu biochemicznego dla klasycznej metody SBH jest binarna informacja o oligonukleotydach będących fragmentami badanej sekwencji DNA, tj. dany oligonukleotyd jest lub nie jest częścią analizowanej sekwencji. W rezultacie powtórzenia fragmentów o długości co najmniej l prowadzą do wystąpienia błędów negatywnych. Jednakże rozwój technologii chipów DNA umożliwia wykorzystanie dodatkowej informacji pochodzącej z eksperymentu biochemicznego.

Intensywność sygnału na obrazie chipu DNA dla poszczególnych komórek jest skorelowana z liczbą wystąpień danego oligonukleotydu w sekwencji. Niestety zależność ta nie jest liniowa. Wraz ze wzrostem liczby powtórzeń danego fragmentu zmniejsza się precyzja tej informacji. O ile łatwo zaobserwować różnicę dla sygnału reprezentującego jedno i wiele wystąpień, o tyle przykładowo rozróżnienie sygnałów odpowiadających siedmiu i ośmiu powtórzeniom może być bardzo trudne lub nawet niemożliwe. Pomimo braku precyzji taka częściowa informacja o liczbie powtórzeń może być bardzo użyteczna [28–30]. Świadczyć o tym może np. wykorzystywanie intensywności sygnału z chipu do analizy ekspresji genów [61].

Wykorzystanie dodatkowej informacji o powtórzeniach nie jest jedyną koncepcją na udoskonalenie klasycznej metody SBH. Sposobem na zmniejszenie liczby błędów wynikających z hybrydyzacji jest zastosowanie chipów DNA zaprojektowanych na nieco innych zasadach. Chip DNA wykorzystywany przy klasycznym podejściu zawiera fragmenty o jednakowej długości. Mają one różne temperatury topnienia, tj. tworzą one stabilne dupleksy w różnych temperaturach otoczenia. W efekcie trudno jest uzyskać jednolite warunki, w których wszystkie elementy biblioteki oligonukleotydów o zadanej długości tworzyłyby stabilne wiązania z komplementarnymi odpowiednikami. Aby uniknąć tego problemu, zamiast wszystkich możliwych sekwencji długości l biblioteka może zawierać wszystkie sekwencje o pewnej ustalonej temperaturze topnienia [8]. Takie biblioteki nazywane są izotermicznymi, a ich zastosowanie prowadzi do zredukowania liczby błędów hybrydyzacji.

1.2 Uzasadnienie tematu badawczego

Problem Orienteering jest przedmiotem intensywnych badań, ze względu na takie praktyczne zastosowania jak planowanie ruchu robota czy problem marszrutyzacji. W związku ze złożonością obliczeniową opracowywane są algorytmy aproksymacyjne. W przypadku wersji problemu dla grafów nieskierowanych zaproponowano już algorytmy aproksymacyjne ze stałym współczynnikiem aproksymacji, które gwarantują jakość rozwiązań niezależną od danych wejściowych, a w szczególności od rozmiaru instancji problemu i wartości optymalnej [1–3, 5, 18, 19, 23, 24, 31, 32]. W przypadku wersji skierowanej istnieją jedynie algorytmy aproksymacyjne, których gwarancja aproksymacji zależy od rozmiaru problemu lub wartości optymalnej [23, 49]. Samo istnienie algorytmów ze stałym współczynnikiem aproksymacji dla problemu Orienteering w grafach skierowanych jest pytaniem otwartym [23].

Zastosowanie algorytmu aproksymacyjnego ze stałym współczynnikiem aproksymacji dla klasycznego problemu sekwencjonowania przez hybrydyzację umożliwiłoby uzyskanie rozwiązania, dla którego byłaby gwarancja wykorzystania pewnej minimalnej liczby l-merów ze spektrum. Zrekonstruowana sekwencja zawierałaby zawsze przynajmniej określony procent l-merów wchodzących w skład badanego łańcucha DNA.

Pewne problemy sekwencjonowania DNA przez hybrydyzację z dodatkową informacją o wielokrotności są już formalnie zdefiniowane [28–30]. W związku z tym, że przy wykorzystaniu aktualnie dostępnej technologii chipów DNA uzyskanie dokładnej informacji o liczbie powtórzeń jest nierealne, oprócz przypadku z precyzyjną informacją o wielokrotności zdefiniowano również bardziej praktyczne wersje problemu z uproszczonymi modelami tej informacji. Pierwszy z nich, zwany "jeden i wiele", zakłada możliwość rozróżnienia sygnału dla jednego wystąpienia oligonukleotydu od sygnału dla wielu wystąpień (więcej niż raz). Drugi model, zwany "jeden, dwa i wiele", zakłada, że istnieje możliwość określenia, że dany oligonukleotyd występuje w badanej sekwencji dokładnie jeden raz, dokładnie dwa razy lub co najmniej trzy razy. Jednak mimo potencjalnych korzyści obecnie istniejące algorytmy wykorzystują jedynie binarną informację o elementach spektrum. Istnieje więc potrzeba opracowania nowych metod, które umożliwiłby zweryfikowanie, czy faktycznie taka dodatkowa informacja pozwala na lepszą rekonstrukcję sekwencji.

Warto zauważyć, że w związku intensywnym rozwojem metod sekwencjonowania następnej generacji (opis w Rozdziale 3.2) sekwencjonowanie DNA przez hybrydyzację w swojej oryginalnej postaci raczej nie będzie wykorzystywane do sekwencjonowania nowych, nieznanych łańcuchów DNA. Metoda ta i jej rozszerzenia mogą mieć jednak zastosowanie przy resekwencjonowaniu oraz do identyfikacji SNP (ang. *Single Nucleotide Polymorphism*), tj. pojedynczych zmian nukleotydów w odpowiadających sobie sekwencjach DNA osobników tego samego gatunku. W przypadku ludzkiego DNA takie różnice mogą również występować w obrębie pary chromosomów jednego osobnika. Identyfikacja SNP dla ludzkiego DNA ma szczególne znaczenie w medycynie, bo te drobne zmiany w DNA mogą wpływać na to, jak w danym organizmie przebiega rozwój choroby czy w jaki sposób reaguje on na podane leki i szczepionki. Sama reakcja na lekarstwo może być również badana z perspektywy populacji wirusów i bakterii, których genom może wpływać na odporność na dany lek. Przykładowe zastosowanie SBH może więc dotyczyć takich aspektów jak:

- resekwencjonowanie genomu wirusów i bakterii celem określenia ich odporności na lekarstwa,

- identyfikacja SNP odpowiedzialnych za choroby genetyczne celem wsparcia diagnozowania chorób,
- realizacja koncepcji medycyny spersonalizowanej.

Szczególnie obiecującym wariantem sekwencjonowania przez hybrydyzację w kontekście opisanych powyżej zastosowań jest shotgun SBH [55]. Ogólny opis tej metody przedstawiono w Rozdziale 3.3.7.

Określono następujące założenia pracy:

1. Istnieje algorytm aproksymacyjny dla problemu Orienteering w grafach skierowanych, którego jakość aproksymacji nie zależy ani od rozmiaru instancji problemu ani od wartości optymalnej rozwiązania.
2. Wykorzystanie w problemie sekwencjonowania DNA przez hybrydyzację nawet nieprecyzyjnej informacji o powtórzeniach umożliwia uzyskanie lepszych wyników.
3. Informacja o powtórzeniach ma zastosowanie zarówno przy klasycznym sekwencjonowaniu przez hybrydyzację jak i dla wersji z bibliotekami izotermicznymi.
4. Wzrost precyzji informacji o powtórzeniach wykorzystanej w problemie sekwencjonowania DNA przez hybrydyzację prowadzi do wzrostu jakości rekonstruowanych sekwencji.

1.3 Cel i zakres pracy

Pierwsza część rozprawy związana jest z modelami grafowymi. Rozważane są potencjalne sposoby zamodelowania problemu sekwencjonowania DNA przez hybrydyzację z informacją o powtórzeniach z wykorzystaniem teorii grafów. W przypadku alternatywnych możliwości dyskutowane są konsekwencje poszczególnych rozwiązań.

Druga część pracy związana jest z klasycznym problemem sekwencjonowania DNA przez hybrydyzację i dotyczy problemu Orienteering w grafach skierowanych. Proponowane są techniki aproksymacji, które umożliwiają uzyskanie gwarancji jakości niezależnej od rozmiaru instancji problemu i wartości optymalnej. Przedstawiono również szczególny przypadek, dla którego istnieje stały współczynnik aproksymacji. Do rozwiązania głównego problemu stosuje się algorytmy aproksymacyjne dla problemów: s-t k-ścieżki w grafie skierowanym oraz k-ścieżki w grafie skierowanym. Pierwszy z nich polega na znalezieniu w grafie skierowanym najkrótszej ścieżki z wierzchołka s do wierzchołka t zawierającej co najmniej k wierzchołków. W drugim przypadku celem jest wyznaczenie najkrótszej ścieżki w grafie skierowanym zawierającej co najmniej k wierzchołków.

Ostatnia część związana jest z sekwencjonowaniem DNA przez hybrydyzację wraz z dodatkową informacją o powtórzeniach. Dla wybranego modelu grafowego definiowane są algorytmy, które umożliwiają ocenę wpływu informacji o powtórzeniach na jakość sekwencjonowania. Badania są prowadzone zarówno dla klasycznego podejścia jak i dla problemu z bibliotekami izotermicznymi.

Główne cele i zadania niniejszej pracy doktorskiej są określone następująco:

1. Opracowanie algorytmu aproksymacyjnego dla problemu s-t k-ścieżki w grafach skierowanych o stałym współczynniku aproksymacji.

2. Opracowanie algorytmu aproksymacyjnego dla problemu k-ścieżki w grafach skierowanych o stałym współczynniku aproksymacji.

3. Opracowanie algorytmu aproksymacyjnego dla problemu Orienteering w grafach skierowanych o stałym współczynniku aproksymacji.

4. Określenie potencjalnych modeli grafowych reprezentujących problem sekwencjonowania DNA przez hybrydyzację z informacją o powtórzeniach.

5. Wybór jednego z modeli grafowych, który zostanie wykorzystany do implementacji algorytmów.

6. Zaprojektowanie i implementacja algorytmów przybliżonych (zachłanny, przeszukiwanie tabu, kolonia mrówek) dla problemu sekwencjonowania DNA przez hybrydyzację z informacją o powtórzeniach. Implementacja powyższych algorytmów powinna umożliwiać wykorzystanie zarówno danych otrzymanych przy użyciu klasycznych bibliotek jak i bibliotek izotermicznych.

7. Ocena wpływu dodatkowej informacji o powtórzeniach na jakość sekwencjonowania DNA przez hybrydyzację z wykorzystaniem zaimplementowanych algorytmów.

Struktura pracy jest następująca. W kolejnym rozdziale przedstawione zostały podstawy biologii molekularnej, a w Rozdziale 3 dokonano przeglądu metod sekwencjonowania DNA. Rozdział 4 zawiera opis podstawowych zagadnień matematycznych i informatycznych. Rozdział 5 prezentuje definicje problemów sekwencjonowania DNA przez hybrydyzację oraz związane z nimi istniejące modele grafowe. Nowe modele grafowe dla problemów SBH z dodatkową informacja o powtórzeniach zostały omówione w Rozdziale 6. Następnie w Rozdziale 7 zaprezentowano algorytm aproksymacyjny dla klasycznej postaci problemu sekwencjonowania DNA przez hybrydyzację. W Rozdziale 8 przedstawiono algorytmy heurystyczne dla klasycznego i izotermicznego sekwencjonowania DNA przez hybrydyzację wykorzystujące dodatkową informację o powtórzeniach. Przedostatni rozdział zawiera rezultaty eksperymentów obliczeniowych. W ostatnim rozdziale dokonano podsumowania wyników przeprowadzonych prac badawczych.

Rozdział 2

Podstawowe zagadnienia biologii molekularnej

2.1 Wprowadzenie

Biologia molekularna jako dziedzina nauki zajmuje się badaniem życia i organizmów żywych na poziomie cząsteczek, z których są one zbudowane. Przedmiotem jej badań jest wpływ właściwości tych cząsteczek na funkcjonowanie organizmów. Zajmuje się ona zarówno zrozumieniem interakcji, które zachodzą wewnątrz komórek pomiędzy takimi cząsteczkami jak DNA, RNA i białka, jak i poznaniem mechanizmów regulujących te interakcje.

Biologia molekularna pokrywa się częściowo z innymi dziedzinami nauki, a w szczególności z genetyką i biochemią. Trudno byłoby wyznaczyć wyraźne granice pomiędzy tymi dziedzinami, jednak każda z nich skupia się na odmiennych aspektach.

- **Biochemia** studiuje substancje chemiczne i reakcje chemiczne zachodzące w organizmach żywych. Koncentruje się ona na roli, strukturze i funkcji danej cząsteczki.
- **Genetyka** zajmuje się budową genów, ich wpływem na funkcjonowanie komórek i organizmu oraz mechanizmami dziedziczenia oraz różnicowania materiału genetycznego.
- **Biologia molekularna** bada molekularne podstawy działania komórek oraz fundamenty procesów replikacji, transkrypcji i translacji (szczegóły opisane w Rozdziale 2.4).

Wiele wyników badań biologicznych, w tym biologii molekularnej, ma charakter ilościowy. Ich analiza bez zaangażowania narzędzi matematycznych czy informatycznych byłaby bardzo trudna lub wręcz niemożliwa. Rozwój i zastosowanie tych narzędzi obejmuje biologia obliczeniowa. Zajmuje się ona przykładowo takimi problemami biologii molekularnej:

- określanie sekwencji białkowych i nukleotydowych (ang. *sequencing*),
- analiza podobieństwa sekwencji białkowych i nukleotydowych (ang. *sequence alignment*),

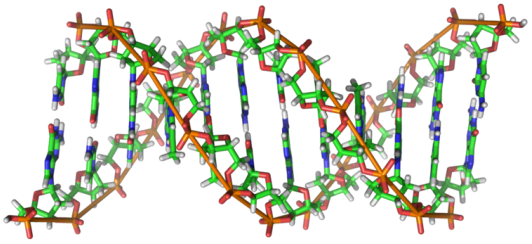
- łączenie krótszych sekwencji białkowych i nukleotydowych w dłuższe łańcuchy (ang. *sequence assembling*),
- przewidywanie ekspresji genów,
- przewidywanie struktury przestrzennej białek,
- porównywanie struktur białkowych (ang. *protein structure alignment*),
- badanie oddziaływań białko-białko,
- budowa i analiza drzew filogenetycznych.

2.2 Budowa kwasów nukleinowych

Nośnikiem informacji genetycznej organizmów żywych jest kwas deoksyrybonukleinowy (DNA). Jest on jednym z dwóch typów kwasów nukleinowych. W przypadku organizmów eukariotycznych DNA jest przechowywane głównie w jądrze komórkowym. Komórki organizmów prokariotycznych nie posiadają jądra, a DNA zlokalizowane jest bezpośrednio w cytoplaźmie komórki.

DNA jest polimerem, tj. związkiem zbudowanym z wielokrotnie powtórzonych jednostek elementarnych. Dla DNA taką jednostką elementarną (monomerem) jest nukleotyd, który składa się z pięciowęglowego cukru deoksyrybozy, reszty kwasu fosforowego oraz zasady azotowej. Istnieją cztery podstawowe rodzaje zasad azotowych wchodzących w skład DNA: adenina i guanina (zasady purynowe) oraz tymina i cytozyna (zasady pirymidynowe). Nukleotydy różnią się pomiędzy sobą jedynie rodzajem zasady azotowej, stąd do ich oznaczenia wykorzystuje się litery A, G, T i C, które reprezentują pierwszą literę nazwy zasady azotowej wchodzącej w skład danego nukleotydu.

Cząsteczka DNA składa się z dwóch łańcuchów, które owijają się wokół wspólnej

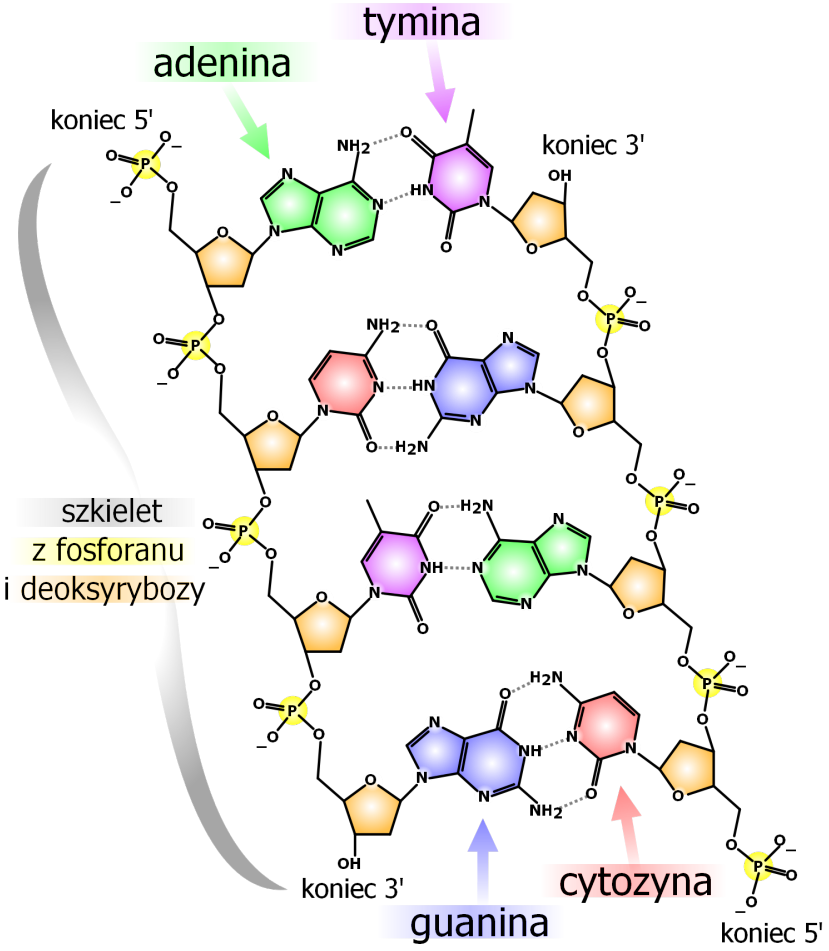


Rysunek 2.1: Podwójna helisa DNA. ©2007 Richard Wheeler, użyte na podstawie licencji Creative Commons Attribution-Share Alike 3.0 Unported

osi tworząc tzw. podwójną helisę (Rysunek 2.1). Połączone ze sobą cukry i reszty fosforanowe znajdują się na zewnątrz helisy, natomiast zasady azotowe skierowane są do wnętrza i tworzą pary z zasadami przeciwległej nici zgodnie z zasadą komplementarności [67]: adenina z tyminą, guanina z cytozyną. Każda z nici DNA ma na jednym końcu, oznaczanym jako 5', nukleotyd z wolną grupą fosforanową przy węglu 5' deoksyrybozy. Drugi koniec, oznaczany jako 3', zawiera nukleotyd z wolną grupą hydroksylową przy węglu 3' deoksyrybozy. Obie nici są splecione ze sobą w taki sposób, że jedna z nich zaczyna się od końca 3', a druga od końca 5', stąd określa się, że są one względem siebie

antyrównoległe. Schemat struktury DNA prezentuje Rysunek 2.2.

Drugim typem kwasów nukleinowych są kwasy rybonukleinowe (RNA). Istnieje wiele

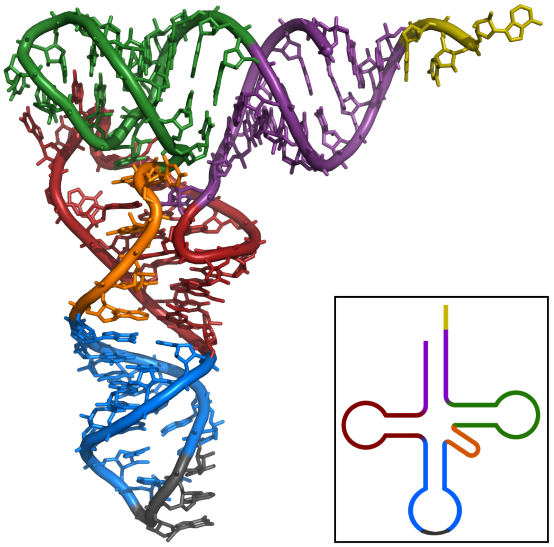


Rysunek 2.2: Struktura chemiczna DNA. Wiązania wodorowe zostały oznaczone linią przerywaną. ©2010 Marek Mazurkiewicz, użyte na podstawie licencji Creative Commons Attribution-Share Alike 3.0 Unported

odmian RNA, a każda z nich pełni określoną rolę w organizmie, np. dekodowanie informacji genetycznej zawartej w DNA, regulacja ekspresji genów. Jest to możliwe dzięki zróżnicowanej wielkości i strukturze poszczególnych typów RNA.

Kwasy rybonukleinowe, podobnie jak kwasy deoksyrybonukleinowe, są polimerami. Istnieją jednak trzy podstawowe różnice w ich budowie. RNA składa się z dużo mniejszej liczby nukleotydów i jest zazwyczaj molekułą jednoniciową. Mimo tego poszczególne nukleotydy tej samej cząsteczki mogą się ze sobą łączyć zgodnie z zasadą komplementarności [67] i tworzyć bardziej złożone struktury (Rysunek 2.3). Druga różnica dotyczy rodzaju cukru, z którego zbudowane są nukleotydy. RNA zawiera rybozę w odróżnieniu od deoksyrybozy wchodzącej w skład DNA. Trzecią istotną różnicą jest inna zasada komplementarna do adeniny. W przypadku RNA jest to uracyl a nie tymina. Podobnie jak w przypadku DNA poszczególne nukleotydy RNA różnią się pomiędzy sobą jedynie zasadą azotową i oznacza się je pierwszymi literami nazw zasad: A, C, G i U.

Poniżej przedstawiono wybrane rodzaje RNA:



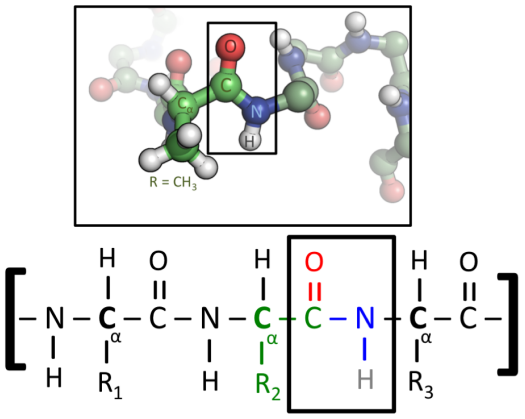
Rysunek 2.3: Struktura drugorzędowa (prezentacja w dwóch wymiarach) i struktura trzeciorzędowa (przestrzenna) tRNA, które jest odpowiedzialne za przyłączenie aminokwasu i jego dostarczenie do rybosomu, gdzie następuje budowa białka.

- matrycowe RNA (mRNA) - koduje sekwencję aminokwasów w białku,
- transportujące RNA (tRNA) - przyłącza się do wolnego aminokwasu i dostarcza go do rybosomów, gdzie następuje synteza białka na podstawie mRNA,
- rybosomalne RNA (rRNA) - wchodzi w skład rybosomów,
- interferencyjne RNA (siRNA) - reguluje ekspresję genów, tj. procesy w wyniku których informacja zakodowana w genie jest przetwarzana na konkretny produkt, np. białko, RNA.

2.3 Budowa białek

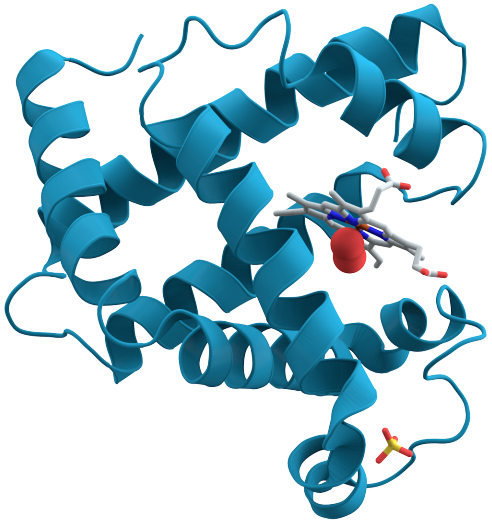
Białka są trzecim istotnym typem cząsteczek, którymi zainteresowana jest biologia molekularna. Są one polimerami zbudowanymi z połączonych sekwencyjnie aminokwasów. Aminokwasy to związki chemiczne zawierające zasadową grupę aminową -NH_2 oraz kwasową grupę karboksylową -COOH. Istnieją 22 podstawowe aminokwasy biogenne, tzn. takie, które wchodzą w skład białek organizmów żywych. Należą one do grupy α-aminokwasów ze względu na to, że zarówno grupa karboksylowa jak i aminowa przyłączone są do tego samego atomu węgla C_α. Kolejne aminokwasy w białku są połączone ze sobą wiązaniem peptydowym, które wiąże grupę aminową jednego aminokwasu z grupą karboksylową drugiego aminokwasu. Strukturę chemiczną białka przedstawia Rysunek 2.4.

Poszczególne białka różnią się od siebie sekwencją aminokwasów, która jest określona przez sekwencję nukleotydów w genie kodującym dane białko. Sekwencja aminokwasów wpływa na strukturę przestrzenną białka (Rysunek 2.5), a ta z kolei determinuje jego



Rysunek 2.4: Struktura chemiczna białka. ©2011 Protein Chemist, użyte na podstawie licencji Creative Commons Attribution-Share Alike 3.0 Unported

funkcję, np. katalizowanie reakcji, regulacja ekspresji genów, reakcja komórek na otaczające je środowisko czy transportowanie innych molekuł z jednej lokalizacji do drugiej.



Rysunek 2.5: Przykładowa struktura przestrzenna białka.

2.4 Centralny dogmat biologii molekularnej

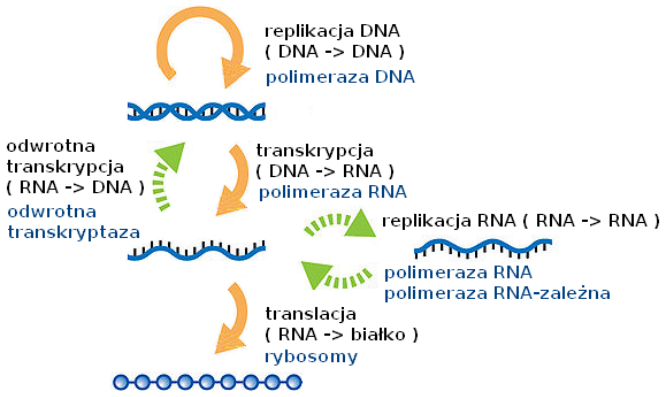
Centralny dogmat biologii molekularnej opisuje przepływ informacji genetycznej w organizmach żywych. Po raz pierwszy został przedstawiony w 1958 roku [25], a następnie opublikowany w czasopiśmie Nature [26]. Jego podstawowym założeniem jest brak możliwości odtworzenia informacji genetycznej na podstawie białka czy to do postaci kwasu nukleinowego czy też innego białka.

Dogmat przedstawia transfer informacji pomiędzy trzema podstawowymi biopolimerami (polimerami występującymi naturalnie w organizmach żywych): DNA, RNA i białkami. Istnieje 9 potencjalnych kierunków przepływu informacji pomiędzy nimi. Dogmat klasyfikuje je w trzy grupy: *ogólne* (uznawane za powszechnie występujące), *specyficzne* (znane, ale występujące tylko w ściśle określonych sytuacjach lub w warunkach laboratoryjnych) oraz *nieznane* (uznawane za niemożliwe). Klasyfikację przepływów zawiera Tabela 2.1, a graficzną prezentację Rysunek 2.6.

DNA, RNA i białka są liniowymi polimerami, co oznacza że ich monomery są po-

Tabela 2.1: Centralny dogmat biologii molekularnej - klasyfikacja przepływów informacji genetycznej.

Ogólne	Specyficzne	Nieznane
DNA → DNA	RNA → DNA	białko → DNA
DNA → RNA	RNA → RNA	białko → RNA
RNA → białko	DNA → białko	białko → białko



Rysunek 2.6: Centralny dogmat biologii molekularnej. Przepływy ogólne oznaczone są kolorem pomarańczowym, przepływy specyficzne kolorem zielonym. Dodatkowo podano nazwę przepływu, biopolimer źródłowy i docelowy oraz czynnik niezbędny do realizacji przepływu (tekst o kolorze niebieskim).

łączone sekwencyjnie, a dany monomer jest powiązany maksymalnie z dwoma innymi. Ta sekwencja monomerów koduje informację, której przepływ zgodnie z centralnym dogmatem biologii molekularnej jest deterministyczny. Sekwencja jednego biopolimeru jest wykorzystywana jako wzorzec do utworzenia innego biopolimeru, którego sekwencja jest wiernym odwzorowaniem sekwencji biopolimeru źródłowego.

Przepływy ogólne reprezentują najbardziej powszechne transfery informacji w organizmach żywych:

- **replikacja DNA** (DNA → DNA) - na podstawie DNA tworzona jest jego kopia,
- **transkrypcja** (DNA → RNA) - na podstawie DNA powstaje sekwencja mRNA,
- **translacja** (RNA → białko) - mRNA użyte jest jako wzorzec do syntezy białka.

Przepływy specyficzne występują jedynie w ściśle określonych warunkach. Odwrotna transkrypcja wykorzystywana jest przez niektóre wirusy do włączenia swojego materiału genetycznego do genomu zainfekowanej komórki. W efekcie zainfekowana komórka tworzy w procesie transkrypcji nowe mRNA, które ostatecznie prowadzi do syntezy nowych białek. Wirusy wykorzystujące proces odwrotnej transkrypcji nazywane są retrowirusami, a ich najbardziej znanym przykładem jest wirus HIV.

Drugi z przepływów specyficznych, replikacja RNA, polega na utworzeniu kopii RNA na podstawie sekwencji RNA. Jest on wykorzystywany przez wiele wirusów jako proces własnej replikacji. Trzeci przepływ specyficzny, polegający na bezpośrednim transferze informacji genetycznej z DNA na białko, udało się zaobserwować na razie jedynie w warunkach laboratoryjnych [48, 64].

Centralny dogmat biologii molekularnej zakłada brak przepływów informacji odbywających się na podstawie białek, ale warto zwrócić uwagę na następujące procesy. Po zakończeniu syntezy białka na podstawie mRNA może ono podlegać dalszym modyfikacjom posttranslacyjnym. Część takich zmian może być realizowana przez enzymy, które w większości są również białkami. Warto również zauważyć możliwość samodzielnego wydzielania się z białka podsekwencji aminokwasów zwanych inteinami [34].

Rozdział 3

Przegląd metod sekwencjonowania DNA

3.1 Podstawowe metody sekwencjonowania

3.1.1 Metoda Sangera

Nazwa metody pochodzi od nazwiska jej twórcy, Fredericka Sangera, który za to osiągnięcie [59, 60] otrzymał w 1980 roku nagrodę Nobla w dziedzinie chemii. Metoda ta była jedną z najpopularniejszych metod przez prawie ćwierć wieku. Dopiero na przełomie XX i XXI wieku udało się opracować nowe metody następnej generacji (ang. *next generation sequencing*, przykłady w Rozdziale 3.2), które wyparły metodę Sangera ze względu na znaczącą redukcję kosztów i możliwość równoległego odczytu wielu fragmentów. Mimo to metoda ta nadal jest stosowana w projektach o niewielkiej skali lub wymagających pojedynczych odczytów o dużej długości (powyżej 500 par zasad).

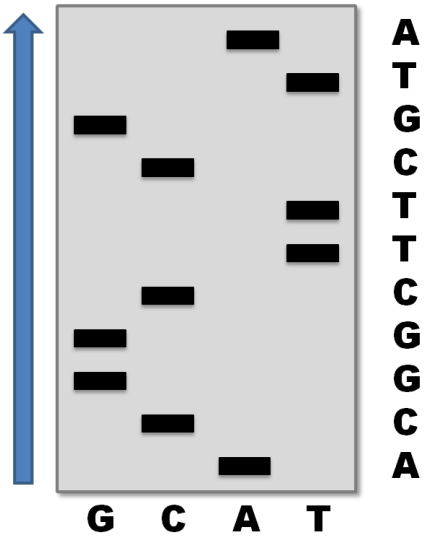
Do określenia sekwencji danego jednoniciowego łańcucha DNA wykorzystuje się fragmenty nici komplementarnej. Otrzymuje się je na podstawie wielu kopii analizowanej sekwencji, które służą jako wzorzec. Jednym z kluczowych odczynników niezbędnych do syntezy komplementarnych sekwencji są związki zwane *trifosforanami deoksynukleozydów*, które są prekursorami nukleotydów wchodzących w skład DNA. Synteza realizowana wyłącznie przy ich użyciu doprowadziłaby do pełnej odbudowy komplementarnego łańcucha. Dlatego dodatkowo wykorzystywane są one jeszcze w zmodyfikowanej postaci *trifosforanów dideoksynukleozydów*. W przypadku ich dołączenia dalsza rozbudowa nie jest możliwa i synteza danej sekwencji zostaje zatrzymana. Wykorzystywany roztwór zawiera oba rodzaje związków, przy czym stężenie trifosforanów deoksynukleozydów jest dużo większe. Synteza komplementarnych fragmentów rozpoczyna się zawsze od początku wzorca, a o zakończeniu syntezy decyduje fakt dołączenia trifosforanu dideoksynukleozydu. W efekcie powstać mogą sekwencje o różnej długości będące pewnym początkowym fragmentem nici komplementarnej. Co więcej, synteza realizowana jest niezależnie w 4 probówkach w taki sposób, że wszystkie fragmenty z danej probówki kończą się tym samym nukleotydem: A, C, G lub T.

Proces syntezy przeprowadzany jest na taką skalę, że zgodnie z prawem dużych liczb można założyć, że w każdej probówce powstają wszystkie możliwe komplementarne podsekwencje kończące się danym nukleotydem. Są one wymieszane, ale można je uporządkować względem długości poddając roztwory działaniu *elektroforezy*. Polega

ona na rozdzieleniu mieszaniny na jednorodne frakcje przez wymuszenie przemieszczania się cząsteczek pod wpływem pola elektrycznego. Zdolność do przemieszczania się jest odwrotnie proporcjonalna do wielkości cząsteczki. Im większa cząsteczka, tym mniejsze będzie jej przesunięcie. Jeżeli poszczególne cząsteczki zostaną oznakowane, przykładowo radioaktywnie, to istnieje możliwość uzyskania obrazu, przykładowo autoradiogramu, prezentującego poszczególne frakcje w postaci prążków. Im mniejsza cząsteczka, tym większe będzie jej przesunięcie względem miejsca naniesienia próbki i tym dalej pojawi się reprezentujący ją prążek.

Sanger zastosował w swojej metodzie elektroforezę żelową, która umożliwia rozróżnienie sekwencji DNA o długości różniącej się nawet jednym nukleotydem. Na płytkę z żelem nanosi się tuż obok siebie roztwory z wszystkich czterech probówek i poddaje ją działaniu pola elektrycznego. W rezultacie otrzymuje się cztery niezależne rozdziały sekwencji DNA zakończonych poszczególnymi nukleotydami. Przykładowy obraz prezentujący poszczególne frakcje przedstawia Rysunek 3.1.

Strzałka oznacza kierunek przemieszczania się cząsteczek. Najbardziej wysunięty



Rysunek 3.1: Metoda Sangera - obraz prezentujący wyniki elektroforezy. Strzałka oznacza kierunek przemieszczania się cząsteczek. Litery G, C, A i T w dolnej części reprezentują miejsca naniesienia roztworów z poszczególnych probówek. Im dalej wysunięty prążek tym mniejsza cząsteczka. Odczytując nukleotydy w odwrotnym kierunku otrzymujemy sekwencję komplementarną do badanego DNA.

prążek odpowiada najmniejszej cząsteczce składającej się z jednego nukleotydu. Na podstawie probówki, z której pochodzi ta cząsteczka, można określić jaki to nukleotyd, bowiem w tym przypadku ostatni nukleotyd jest jedynym nukleotydem wchodzącym w skład cząsteczki. Reprezentuje on pierwszy nukleotyd badanej sekwencji. Należy pamiętać, że probówki zawierały fragmenty komplementarne do badanego DNA. Pierwszym nukleotydem sekwencjonowanego DNA jest więc nukleotyd komplementarny do tego, który wynika z obrazu. Kolejny prążek reprezentuje cząsteczkę składającą się z dwóch nukleotydów. Co więcej, istnieje możliwość określenia ostatniego z nich analogicznie jak w poprzednim przypadku, tj. na podstawie źródłowej probówki. Nukleotyd ten jest komplementarny do drugiego nukleotydu badanej sekwencji. Postępując analogicznie dla kolejnych prążków możemy odczytać kolejne nukleotydy analizowanego DNA i określić kompletną sekwencję.

Metoda Sangera została w późniejszym okresie udoskonalona przez fluorescencyjne znakowanie trifosforanów dideoksynukleozydów. Cząsteczki odpowiadające poszczególnym nukleotydom oznaczone są innym kolorem. Umożliwia to realizację syntezy w jednej probówce i identyfikację nukleotydów na podstawie kolorów prążków w obrazie otrzymanym w wyniku przeprowadzenia elektroforezy.

3.1.2 Metoda Maxama-Gilberta

Metoda została opracowana przez Allana Maxama i Waltera Gilberta w latach 1976-1977 [47]. Gilbert otrzymał za to osiągnięcie (razem z Sangerem) nagrodę Nobla w dziedzinie chemii w 1980 roku. Początkowo metoda ta zdobyła dużo większą popularność niż powstała w tym samym okresie metoda Sangera. Pozwalała bowiem na bezpośrednie użycie oczyszczonego DNA, podczas gdy metoda Sangera w swojej początkowej postaci wymagała dodatkowo klonowania badanej sekwencji w ramach przygotowywania jednoniciowego DNA. Z czasem metoda Sangera została usprawniona. Spowodowało to spadek popularności metody Maxama-Gilberta, która jest dość skomplikowana technologicznie i wymaga użycia odczynników chemicznych wymagających specjalnych środków ostrożności.

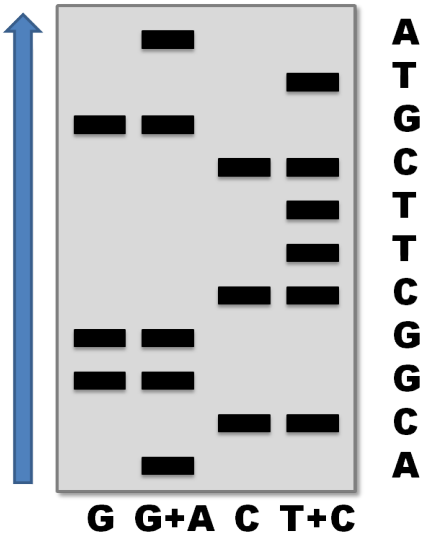
Wspomniane związki chemiczne wykorzystywane są do podziału oryginalnego DNA na mniejsze fragmenty. Są one tak dobrane, aby rozpad sekwencji następował w konkretnych miejscach. Przygotowywane są cztery niezależne roztwory. W dwóch pierwszych cięcia sekwencji następują tuż przed konkretnym nukleotydem. W dwóch pozostałych roztworach do rozpadu sekwencji może dojść przed jednym z dwóch nukleotydów. W rezultacie jeden roztwór zawiera fragmenty powstałe w wyniku przecięcia sekwencji tuż przed nukleotydem G[1]. W skład drugiego roztworu wchodzą cząsteczki utworzone w trakcie rozpadu tuż przed nukleotydem C. W trzecim roztworze zawarte są sekwencje utworzone w wyniku cięć tuż przed nukleotydem A lub nukleotydem G[2]. Ostatni roztwór składa się z fragmentów powstałych przez cięcia tuż przed nukleotydem T lub nukleotydem C. Poszczególne rodzaje cięć są w dalszej części oznaczane odpowiednio: G, C, A+G i T+C.

Do podziałów wykorzystywanych jest wiele kopii tej samej sekwencji. Jeden koniec (zazwyczaj 5') każdej z nich jest radioaktywnie oznakowany, np. radioaktywnym fosforem. Wszystkie cztery roztwory otrzymane w wyniku cięć są umieszczane obok siebie na płytce z żelem i są poddawane elektroforezie (opis metody znajduje się w Rozdziale 3.1.1). Umożliwia to rozdzielenie fragmentów o tej samej liczbie nukleotydów, a dzięki radioaktywnemu oznakowaniu danego końca istnieje możliwość otrzymania obrazu prezentującego poszczególne frakcje.

Przykładowy obraz otrzymany w wyniku elektroforezy zawiera Rysunek 3.2. Na jego podstawie, podobnie jak w przypadku metody Sangera (Rozdział 3.1.1), istnieje możliwość określenia kolejności nukleotydów w badanej sekwencji DNA. Jeżeli prążek pojawi się jedynie w dla roztworu z cięciami typu G+A, to oznacza, że cięcie nastąpiło tuż przed nukleotydem A. Jeżeli prążek pojawi się zarówno dla roztworów z cięciami typu G jak i A+G, to oznacza, że cięcie nastąpiło przed nukleotydem G. Analogicznie można ustalić, czy rozpad sekwencji nastąpił przed nukleotydem C czy T. Interpretując

[1] W oryginalnej wersji cięcia powstawały zarówno przed G jak i przed A, przy czym warunki reakcji były tak dobrane, że dominowały cięcia przed G. Oznaczano je G>A.

[2] W oryginalnej wersji cięcia powstawały zarówno przed A jak i przed G, przy czym warunki reakcji były tak dobrane, że dominowały cięcia przed A. Oznaczano je A>G.



Rysunek 3.2: Metoda Maxama-Gilberta - obraz prezentujący wyniki elektroforezy. Strzałka oznacza kierunek przemieszczania się cząsteczek. Litery G, G+A, C i T+C w dolnej części reprezentują miejsca naniesienia roztworów zawierających fragmenty otrzymane w wyniku poszczególnych cięć. Im dalej wysunięty prążek tym mniejsza cząsteczka. Odczytując nukleotydy w odwrotnym kierunku otrzymujemy sekwencję badanego DNA.

prążki w kolejności od najbardziej wysuniętych względem miejsca naniesienia próbek na płytkę można określić sekwencję analizowanego DNA.

3.2 Wybrane metody sekwencjonowania następnej generacji

3.2.1 Pirosekwencjonowanie (454)

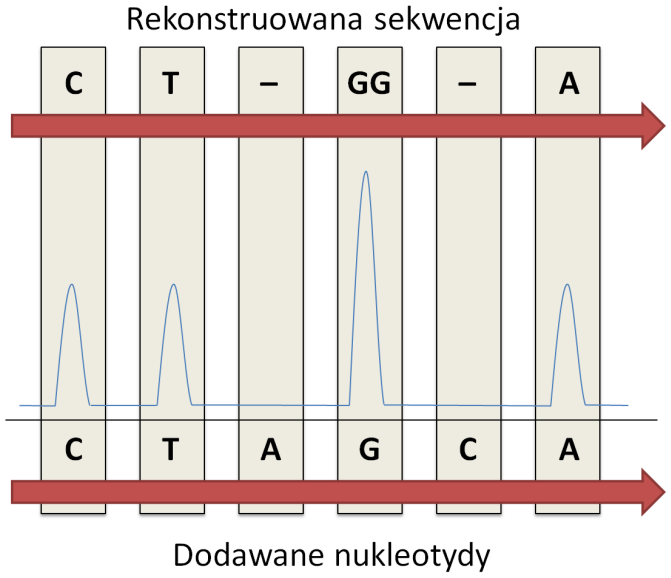
Pirosekwencjonowanie [57, 58] jest metodą sekwencjonowania bazującą na syntezie DNA. Przygotowywane jest jednoniciowe DNA stanowiące wzorzec syntetyzowanej sekwencji. Na podstawie wzorca budowany jest łańcuch komplementarny. Jest on stopniowo rozbudowywany przez dodawanie na końcu kolejnych nukleotydów. W danym kroku do bieżącej sekwencji próbuje się dołączyć po kolei każdy z czterech możliwych nukleotydów.

Jeżeli dany nukleotyd zostanie dołączony, to w wyniku zachodzącej reakcji chemicznej powstanie związek zwany difosforanem. Cały roztwór poddawany jest kolejnym reakcjom chemicznym, które w przypadku obecności difosforanu pozwalają na zaobserwowanie bioluminescencji możliwej do zarejestrowania specjalną kamerą. Próba dołączenia nukleotydu, który nie jest komplementarny do nukleotydu z wzorca, kończy się niepowodzeniem. W rezultacie difosforan nie powstaje, a zjawisko bioluminescencji nie jest obserwowane.

W przypadku występowania we wzorcu kolejno po sobie tych samych nukleotydów dołączana jest na końcu syntetyzowanej sekwencji odpowiednia liczba nowych nukleotydów komplementarnych do tych z wzorca. Powoduje to wydzielenie większej liczby

cząsteczek difosforanu, a sygnał otrzymywany w wyniku bioluminescencji jest silniejszy. Siła sygnału jest proporcjonalna do liczby przyłączonych nukleotydów. Przykładowy obraz (pirogram) zarejestrowany kamerą przedstawia Rysunek 3.3.

Firma 454 Life Sciences, która aktualnie jest częścią Roche Diagnostics, opracowała zrównoleglonąwersję powyższej metody. W jednym przebiegu możliwe jest odczytanie nawet 1 miliona pojedynczych sekwencji o długości 700 nukleotydów [43]. Umożliwiło to zastosowanie pirosekwencjonowania na szeroką skalę [46, 68].



Rysunek 3.3: Obraz otrzymywany w wyniku pirosekwencjonowania. Intensywność sygnału (wysokość piku) odzwierciedla liczbę przyłączonych nukleotydów. Brak sygnału oznacza, że dany nukleotyd nie został dołączony. Prezentowany pirogram odpowiada sekwencji CTGGA.

3.2.2 Metoda firmy Illumina/Solexa

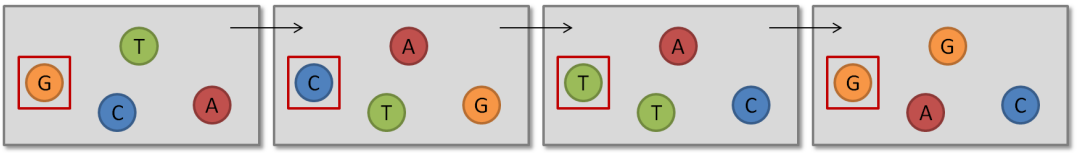
Metoda opisana w niniejszym rozdziale została opracowana przez firmę Solexa, która obecnie jest częścią firmy Illumina. W jednym przebiegu tej metody możliwe jest określenie wielu sekwencji o łącznej długości nawet 600 miliardów nukleotydów [43]. Jest to metoda opierająca się na syntezie, podobnie jak metoda Sangera i pirosekwencjonowanie. W tym przypadku również przygotowywane jest jednoniciowe DNA stanowiące wzorzec, ale sama synteza przebiega inaczej.

Do syntezy wykorzystywane są znakowane fluorescencyjnie tzw. *odwracalne terminatory* (ang. *reversible terminators*). Istnieją cztery rodzaje odwracalnych terminatorów. Każdy z nich odpowiada jednemu z czterech nukleotydów i emituje światło w innym kolorze. Co więcej, terminatory zbudowane są w taki sposób, że przyłączenie terminatora na końcu bieżącej sekwencji uniemożliwia przyłączenie kolejnego. Dzięki temu istnieje możliwość kontrolowania procesu i dołączanie do syntetyzowanej sekwencji po jednym nukleotydzie w danym kroku.

Proces rozpoczyna się od podziału sekwencji DNA i przygotowania krótkich, jednoniciowych fragmentów o długości ok. 200 nukleotydów. Stanowią one wzorce i są umieszczane na specjalnej płytce, na której przeprowadzana jest *amplifikacja*, tj. kopiowanie

danego fragmentu. W wyniku płytka zawiera tzw. *klastry* zbudowane z wielu kopii danego wzorca.

Następnie płytkę umieszcza się cyklicznie w roztworze zawierającym wszystkie cztery rodzaje odwracalnych terminatorów. W każdym cyklu dochodzi do dołączenia jednego terminatora do syntetyzowanych sekwencji we wszystkich klastrach, a pozostałe nieprzyłączone terminatory są wypłukiwane z płytki. Dzięki fluorescencyjnemu oznakowaniu terminatorów istnieje możliwość uzyskania obrazu (Rysunek 3.4) prezentującego jakie terminatory zostały przyłączone w poszczególnych klastrach. Na podstawie tego można wnioskować jaki jest kolejny nukleotyd syntetyzowanych sekwencji. Przed ponownym umieszczeniem płytki w roztworze odwracalnych terminatorów przeprowadza się jeszcze reakcje chemiczne prowadzące do usunięcia fluorescencyjnego znacznika i odblokowania możliwości przyłączenia kolejnego terminatora. W efekcie syntetyzowane sekwencje zawierają na końcu standardowe nukleotydy.



Rysunek 3.4: Metoda sekwencjonowania firmy Solexa/Illumina - sekwencja obrazów uzyskanych w kolejnych cyklach. Sekwencja nukleotydów dla zaznaczonego klastra to GCTG.

3.3 Sekwencjonowanie przez hybrydyzację

3.3.1 Podejście klasyczne

Metoda sekwencjonowania przez hybrydyzację (ang. *Sequencing By Hybridization*, SBH) [4, 44] składa się z dwóch etapów. Pierwszy z nich to eksperyment biochemiczny. Jego wynikiem jest zbiór oligonukleotydów (tj. krótkich jednoniciowych sekwencji DNA), które stanowią fragmenty oryginalnej sekwencji DNA. Wszystkie oligonukleotydy mają tą samą długość l i są nazywane l-merami. Drugi etap (obliczeniowy) sprowadza się do rekonstrukcji analizowanej sekwencji na podstawie informacji o l-merach.

Do przeprowadzenia eksperymentu w pierwszym etapie wykorzystywane są chipy DNA [51, 53] zawierające pełną bibliotekę oligonukleotydów o długości l. Taki chip DNA podzielony jest na komórki zawierające pewną liczbę identycznych, jednoniciowych cząsteczek DNA o długości l (zwanych *sondami*). W ramach eksperymentu chip jest umieszczany w roztworze zawierającym wiele kopii badanego jednoniciowego DNA. Oligonukleotydy znajdujące się na chipie hybrydyzują z komplementarnymi fragmentami cząsteczki z przygotowanego roztworu. Poszczególne kopie analizowanego DNA zostają wcześniej radioaktywnie lub fluoroscencyjnie oznakowane, dzięki czemu istnieje możliwość otrzymania dla danego chipu obrazu, który prezentuje l-mery wchodzące w skład badanej sekwencji.

W idealnym przypadku etap biochemiczny dostarcza pełnej i poprawnej informacji o oligonukleotydach wchodzących w skład analizowanej sekwencji DNA. Otrzymany zbiór

l-merów nazywany jest *spektrum*. Jednak w praktyce w trakcie eksperymentu pojawiają się pewne błędy. Mogą one być dwojakiego rodzaju: pozytywne i negatywne. Błąd pozytywny występuje, gdy analizowana sekwencja ulega hybrydyzacji z oligonukleotydem na chipie, który nie jest do niej w pełni komplementarny. W takiej sytuacji spektrum zawiera pewne dodatkowe l-mery, które nie są fragmentami badanej sekwencji. Możliwa jest również sytuacja odwrotna. Analizowana sekwencja może nie zhybrydyzować do komplementarnego oligonukleotydu na chipie. W rezultacie spektrum może nie zawierać kompletnej informacji o l-merach będących fragmentami analizowanej sekwencji. Te brakujące oligonukleotydy to błędy negatywne. Innym źródłem błędów negatywnych są powtórzenia występujące w analizowanej sekwencji. W klasycznym podejściu spektrum jest zbiorem a nie multizbiorem, więc jeżeli w badanym DNA występują powtórzenia o długości co najmniej l, to taki przypadek nie zostanie wykryty.

Przykład 3.3.1. Niech nieznaną, rekonstruowaną sekwencją będzie AACTGTAAT o długości $n = 9$, a wykorzystana biblioteka oligonukleotydów niech zawiera wszystkie l-mery o długości $l = 3$. W przypadku braku jakichkolwiek błędów wynikiem eksperymentu biochemicznego będzie idealne spektrum zawierające następujące elementy: AAC, AAT, ACT, CTG, GTA, TAA i TGT. Rekonstrukcję sekwencji dla powyższego przypadku prezentuje Rysunek 3.5.

```
AAC
 ACT
  CTG
   TGT      uporządkowane l-mery ze spektrum
    GTA
     TAA
      AAT

AACTGTAAT   zrekonstruowana sekwencja
```

Rysunek 3.5: Sekwencjonowanie DNA przez hybrydyzację - rekonstrukcja sekwencji AACTGTAAT na podstawie idealnego spektrum.

Przykład 3.3.2. Niech nieznaną, rekonstruowaną sekwencją będzie AACTGTAAT o długości $n = 9$ (ta sama co w Przykładzie 3.3.1), a wykorzystana biblioteka oligonukleotydów również niech zawiera wszystkie l-mery o długości $l = 3$. Ponadto, niech spektrum otrzymane w wyniku eksperymentu biochemicznego zawiera dwa błędy hybrydyzacji: brak oligonukleotydu AAT (błąd negatywny) oraz niebędący fragmentem analizowanej sekwencji oligonukleotyd GGT (błąd pozytywny). W skład spektrum wchodzą więc następujące l-mery: AAC, ACT, CTG, GTA, GGT, TAA i TGT. Rekonstrukcję sekwencji dla powyższego przypadku prezentuje Rysunek 3.6.

Rozwiązywany w drugim etapie problem kombinatoryczny może zostać sprowadzony do grafowego problemu *Orienteering* [35] w grafie skierowanym. W problemie tym dany jest skierowany graf wejściowy, funkcja kosztu łuków, wartość każdego wierzchołka oraz pewien budżet. Celem jest znalezienie w grafie wejściowym skierowanej ścieżki o koszcie nie większym niż zadany budżet, której wartość wyznaczona przez sumę wartości odwiedzonych wierzchołków będzie maksymalna. Problem Orienteering jest trudny obliczeniowo i należy do klasy problemów silnie NP-trudnych (definicja w Rozdziale 4.1). W praktyce oznacza to wykładniczy wzrost czasu obliczeń przy wzroście rozmiaru instancji problemu.

```
AAC
 ACT
  CTG       uporządkowane l-mery ze spektrum
   TGT      gdzie umieścić GGT?
    GTA
     TAA

AACTGTAA?   zrekonstruowana sekwencja
```

Rysunek 3.6: Sekwencjonowanie DNA przez hybrydyzację - rekonstrukcja sekwencji AACTGTAAT na podstawie spektrum z błędami. Odtworzona sekwencja AACTGTAA jest krótsza niż analizowana sekwencja. Co więcej, jeden z l-merów ze spektrum (GGT) nie został wykorzystany w utworzonym rozwiązaniu.

Rozważa się również uproszczoną wersję problemu rozwiązywanego w drugim etapie. Zakładając brak jakichkolwiek błędów negatywnych i pozytywnych problem sprowadza się do znalezienia ścieżki Hamiltona w grafie skierowanym, tj. ścieżki odwiedzającej dokładnie raz każdy z wierzchołków. W ogólności problem ten jest również silnie NP-trudny, ale w przypadku SBH może on zostać sprowadzony do znalezienia drogi Eulera w grafie skierowanym [52], tj. drogi przechodzącej przez każdy z łuków dokładnie raz. Problem znalezienia drogi Eulera jest problemem łatwym obliczeniowo i można go rozwiązać w czasie wielomianowym.

3.3.2 Sekwencjonowanie wieloetapowe

Jednym ze sposobów redukcji skali błędów negatywnych wynikających z powtórzeń w klasycznym SBH jest wykorzystanie biblioteki dłuższych l-merów. Pozwala to na rekonstrukcję potencjalnie dłuższych sekwencji. Jednak konsekwencją zwiększenia wartości l jest wykładniczy wzrost liczności biblioteki, której rozmiar wynosi 4^l. Wraz ze wzrostem liczności biblioteki wymagana jest większa powierzchnia chipu DNA, a ta jest ograniczona dostępną technologią. Aktualnie pozwala ona na konstrukcję chipów DNA zawierających pełną bibliotekę oligonukleotydów o długości l nie większej niż 10-11 nukleotydów.

Problem wykładniczego wzrostu liczności biblioteki można rozwiązać przeprowadzając kilkukrotnie eksperyment biochemiczny z wykorzystaniem za każdym razem oligonukleotydów o różnej długości. Podejście to nazywa się wieloetapowym sekwencjonowaniem przez hybrydyzację (ang. *multistage sequencing by hybridization*) [38, 45, 63].

W pierwszym etapie wykorzystywana jest pełna biblioteka krótkich oligonukleotydów, przykładowo l=4. W wyniku eksperymentu biochemicznego otrzymuje się informację o wszystkich podsewkencjach długości l. W kolejnym etapie wykorzystuje się dwukrotnie dłuższe $2l$-mery. Jeżeli dany l-mer nie jest częścią badanej sekwencji, to również nie będzie jej podsekwencją zawierający go $2l$-mer. Dzięki tej obserwacji można znacząco zredukować rozmiar wykorzystywanej biblioteki $2l$-merów. Zawiera ona jedynie kombinacje tych l-merów, które zostały zidentyfikowane w poprzednim etapie. Po ustaleniu składu biblioteki $2l$-merów i przygotowaniu odpowiedniego chipu DNA wykonywany jest kolejny eksperyment biochemiczny. W wyniku identyfikuje się wszystkie podsekwencje o długości $2l$. Następne etapy przebiegają analogicznie. Na podstawie wyników poprzedniego etapu ustala się zawartość kolejnej biblioteki oligonukleotydów, opracowuje się nowy chip i ponownie przeprowadza się proces hybrydyzacji.

Podstawowym ograniczeniem tego rozszerzenia jest brak możliwości stosowania standardowych bibliotek oligonukleotydów. O ile pierwsza biblioteka zawierająca najkrótsze oligonukleotydy może być wykorzystywana przy sekwencjonowaniu innych cząsteczek DNA, o tyle biblioteki wykorzystywane w kolejnych etapach są dedykowane rekonstrukcji konkretnej sekwencji i konieczne jest przygotowywanie nowych chipów DNA. Powoduje to nie tylko wydłużenie samego procesu sekwencjonowania, ale również podnosi koszty analizy pojedynczej sekwencji.

Należy również zauważyć, że ewentualne błędy negatywne z danego etapu propagują się na kolejne. Jeżeli w trakcie eksperymentu biochemicznego obecność pewnego oligonukleotydu nie zostanie wykryta, to nie zostanie on wykorzystany do budowy biblioteki dwukrotnie dłuższych oligonukleotydów, co uniemożliwi wykrycie kolejnych oligonukleotydów w następnych etapach.

3.3.3 Sondy zawierające nukleotydy uniwersalne i zdegenerowane

Jednym z problemów napotykanych w trakcie rekonstrukcji sekwencji na podstawie spektrum jest jednoznaczność rozwiązania. Nawet w przypadku idealnego przebiegu eksperymentu hybrydyzacyjnego i braku powtórzeń w analizowanej sekwencji jednoznaczna rekonstrukcja może być niemożliwa, gdyż dwóm różnym sekwencjom może odpowiadać dokładnie takie samo spektrum.

Maksymalna długość sekwencji, którą można jednoznacznie odtworzyć z zadanym prawdopodobieństwem, została zdefiniowana w [54]. W pracy tej zaproponowano również usprawnienie klasycznej metody przez wykorzystanie odpowiednio zmodyfikowanych nukleotydów do budowy poszczególnych sond (elementów biblioteki). Zastosowanie chipów DNA opartych na takich alternatywnych bibliotekach oligonukleotydów umożliwia zmniejszenie prawdopodobieństwa otrzymania niejednoznacznego rozwiązania, co potwierdzają badania przeprowadzone w [56].

Propozycja konstrukcji alternatywnych chipów DNA zakłada możliwość użycia dwóch dodatkowych odmian nukleotydów. Pierwsza to tzw. *nukleotydy uniwersalne* oznaczane literą X. Są one zdolne do hybrydyzacji z dowolnym z czterech podstawowych nukleotydów. Drugi typ to tzw. *nukleotydy zdegenerowane*, które potrafią hybrydyzować z więcej niż z jednym nukleotydem. Wśród nich można wyróżnić nukleotydy:

- *silne*, oznaczane literą S, zdolne do hybrydyzacji z nukleotydami C i G,
- *słabe*, oznaczane literą W, zdolne do hybrydyzacji z nukleotydami A i T,
- *purynowe*, oznaczane literą R, zdolne do hybrydyzacji z nukleotydami A i G,
- *pirymidynowe*, oznaczane literą Y, zdolne do hybrydyzacji z nukleotydami C i T.

Zaproponowano trzy rodzaje specyficznych chipów DNA [54]. Pierwszy z nich to tzw. *gapped chip*. Składa się on z dwóch części. Jedna zawiera klasyczne oligonukleotydy o zadanej długości l. Druga zwiera oligonukleotydy zbudowane ze standardowych nukleotydów oraz z nukleotydów uniwersalnych. Niech N oznacza dowolny z czterech nukleotydów A, C, G i T. Dla każdego standardowego oligonukleotydu $N_1, N_2, \ldots, N_l$ gapped chip zawiera również oligonukleotyd postaci:

$$N_1, N_2, \ldots, N_{l-1}, \underbrace{X, X, \ldots, X}_{l-1}, N_l. \tag{3.1}$$

Drugi z zaproponowanych specyficznych chipów to tzw. *alternating chip*. Zawiera on oligonukleotydy zbudowane z sekwencji następujących po sobie naprzemiennie standardowych i uniwersalnych nukleotydów. Alternating chip zawiera dwa rodzaje takich oligonukleotydów:

$$N_1, X, N_2, X, \ldots, N_{l-1}, X, N_l \text{ oraz } N_1, X, N_2, X, \ldots, N_{l-1}, N_l. \tag{3.2}$$

Ostatni z proponowanych chipów to tzw. *binary chip*. Pierwsza część chipu zawiera oligonukleotydy wykorzystujące zdegenerowane nukleotydy słabe i silne. Druga część składa się z oligonukleotydów zbudowanych ze zdegenerowanych nukleotydów purynowych i pirymidynowych. W obu przypadkach ostatnim elementem każdego oligonukleotydu jest jeden z podstawowych nukleotydów: A, C, G lub T. Oba rodzaje oligonukleotydów można opisać następująco:

$$\underbrace{\{W,S\},\{W,S\},\ldots\{W,S\}}_{l},N \text{ oraz } \underbrace{\{R,Y\},\{R,Y\},\ldots\{R,Y\}}_{l},N. \tag{3.3}$$

Zastosowanie powyższych chipów prowadzi do konieczności rozwiązania w drugim etapie odmiennych problemów obliczeniowych niż w przypadku klasycznego SBH. W ogólności problemy te są również trudne obliczeniowo (silnie NP-trudne), a ich złożoność obliczeniowa została omówiona w [56]. W pracy tej zaproponowano również konkretne algorytmy umożliwiające rekonstrukcję sekwencji na podstawie wyników eksperymentu biochemicznego wykorzystującego takie specyficzne chipy.

3.3.4 Dodatkowa informacja o pozycji oligonukleotydów ze spektrum

Innym sposobem na rozwiązanie problemu niejednoznaczności rozwiązań jest wykorzystanie dodatkowej informacji o przybliżonej pozycji w analizowanej sekwencji danego oligonukleotydu ze spektrum. Odmiana klasycznej metody wykorzystująca taką informację nazywana jest pozycyjnym SBH (ang. *Positional Sequencing By Hybridization*, PSBH) [21, 36].

Początkowo rozważano powyższy problem przy założeniu braku jakichkolwiek błędów na etapie eksperymentu hybrydyzacyjnego. Dla klasycznego SBH w takim przypadku problem kombinatoryczny sprowadza się do znalezienia drogi Eulera (opis w Rozdziale 3.3.1). Dla PSBH zdefiniowano pewien wariant tego problemu zwany problemem pozycyjnej drogi Eulera [36]. Niech $\pi(P, a)$ oznacza pozycję łuku a w skierowanej drodze P. Mając dany skierowany multigraf $G = (V, A)$ oraz interwał $I_a = \{l_a, h_a\}, l_a \leqslant h_a$ określony dla każdego łuku $a \in A$ celem jest znalezienie w grafie G takiej drogi Eulera P_E, że dla każdego łuku $a \in A$ jego pozycja $\pi(P_E, a)$ spełnia następujący warunek: $l_a \leqslant \pi(P_E, a) \leqslant h_a$.

Wykazano, że problem pozycyjnej drogi Eulera jest problemem NP-zupełnym, nawet gdy maksymalny stopień wejściowy i wyjściowy wierzchołków w grafie jest równy 2 [36]. W pracy tej zaproponowano również algorytm wielomianowy dla pozycyjnej drogi Eulera, gdy rozmiar interwałów jest ograniczony pewną stałą. Udowodniono również, że problem pozycyjnej drogi Eulera jest NP-zupełny, gdy dla wszystkich łuków zdefiniowano co najwyżej trzy możliwe pozycje, choć w przypadku określenia co najwyżej dwóch pozycji istnieje algorytm liniowy [6].

Pozycyjne sekwencjonowanie przez hybrydyzację z uwzględnieniem dowolnego typu błędów zostało przedstawione w [69]. W pracy tej zaproponowano algorytm dokładny typu podziału i ograniczeń (ang. *branch-and-bound*).

3.3.5 Zastosowanie bibliotek izotermicznych

W ramach ustalania warunków realizacji eksperymentu biochemicznego trzeba uwzględnić proces *denaturacji* DNA, tj. separacji podwójnej helisy na dwie pojedyncze nici. Proces ten zachodzi w odpowiednio wysokiej temperaturze zwanej *temperaturą topnienia* oznaczanej przez T_m. W ogólności im dłuższa jest podwójna helisa, tym więcej ciepła trzeba dostarczyć, aby doprowadzić do jej denaturacji.

Jednym ze sposobów wyznaczenia temperatury topnienia dla danej dwuniciowej cząsteczki DNA jest obliczenie sumy wag dla każdej pary nukleotydów, z których jest ona zbudowana. Waga pary nukleotydów A-T wynosi 2°C, a waga pary nukleotydów C-G wynosi 4°C. Przykładowo dla podwójnej helisy DNA, której jedna z nici składa się z sekwencji nukleotydów ATTCAGA, temperatura topnienia wg tego modelu wynosi 18°C.

W klasycznym sekwencjonowaniu przez hybrydyzację wykorzystuje się bibliotekę oligonukleotydów o równej długości, a poszczególne jej elementy mają różną temperaturę topnienia. Utrudnia to takie dobranie warunków eksperymentu biochemicznego, aby wszystkie oligonukleotydy w danej temperaturze tworzyły stabilne dupleksy z analizowaną sekwencją, co ma wpływ na liczbę występujących błędów hybrydyzacji.

Pomysłem na rozwiązanie tego problemu jest zastosowanie *bibliotek izotermicznych*, które zawierają oligonukleotydy o tej samej temperaturze topnienia [8]. Niech wartości x_A, x_C, x_G i x_T oznaczają odpowiednio liczbę wystąpień nukleotydów A, C, G i T w danym oligonukleotydzie. Biblioteka izotermiczna o temperaturze topnienia T_L zawiera oligonukleotydy spełniające następujące równanie:

$$w_A x_A + w_C x_C + w_G x_G + w_T x_T = T_L \tag{3.4}$$

gdzie wagi poszczególnych nukleotydów wynoszą $w_A = w_T = 2$ oraz $w_C = w_G = 4$.

W pracy [8] udowodniono, że do określenia sekwencji DNA wystarczy zastosowanie dwóch bibliotek o temperaturach topnienia T_L oraz $T_L + 2$°C. Wykazano również, że problem obliczeniowy odpowiadający *izotermicznemu sekwencjonowaniu przez hybrydyzację* (ang. *Isothermic Sequencing By Hybridization*, ISBH) jest problemem silnie NP-trudnym.

Należy przy tym zwrócić uwagę, że model wykorzystany w [8] jest bardzo prostym sposobem wyznaczania temperatury topnienia i daje on jedynie przybliżone wyniki. Jednak stosowanie nawet takiego uproszczonego modelu wyznaczania T_m umożliwia lepsze dobranie parametrów eksperymentu hybrydyzacyjnego niż w przypadku klasycznego SBH.

3.3.6 Częściowa informacja o powtórzeniach

Wynikiem etapu biochemicznego dla klasycznej metody SBH jest binarna informacja o oligonukleotydach będących fragmentami badanej sekwencji DNA, tj. dany oligonukleotyd jest lub nie jest częścią analizowanej sekwencji. W rezultacie powtórzenia fragmentów o długości co najmniej l prowadzą do wystąpienia błędów negatywnych. Jednakże rozwój technologii chipów DNA umożliwia wykorzystanie dodatkowej informacji pochodzącej z eksperymentu biochemicznego.

Intensywność sygnału dla danej sondy w obrazie chipu DNA zależy od liczby wystąpień odpowiadającego jej oligonukleotydu w analizowanej sekwencji. Niestety precyzja tej informacji zmniejsza się wraz ze wzrostem liczby powtórzeń danego fragmentu. O ile łatwo rozróżnić sygnał reprezentujący jedno i wiele wystąpień, o tyle przykładowo

zaobserwowanie różnicy w sile sygnału dla siedmiu i ośmiu powtórzeń może być bardzo trudne lub nawet niemożliwe. Jednak pomimo braku precyzji nawet taka częściowa informacja o liczbie wystąpień danego oligonukleotydu w analizowanej sekwencji może być bardzo użyteczna. Intensywność sygnału z chipu jest wykorzystywane przykładowo w analizie ekspresji genów [61].

Formalne definicje problemów obliczeniowych dla problemu SBH z dodatkową informacją o powtórzeniach zostały przedstawione w [28–30]. Aktualnie dostępna technologia chipów DNA nie umożliwia pozyskania dokładnej informacji o wielokrotności poszczególnych oligonukleotydów, dlatego oprócz problemu z precyzyjną informacją o powtórzeniach zostały zdefiniowane również bardziej praktyczne wersje problemu oparte o modele częściowej informacji tego typu.

Pierwszy z nich, zwany "jeden i wiele", zakłada możliwość ustalenia, czy dany oligonukleotyd występuje w analizowanej sekwencji raz czy wiele razy. Drugi model, zwany "jeden, dwa i wiele", zakłada możliwość rozróżnienia sygnałów reprezentujących jedno, dwa i przynajmniej trzy wystąpienia danego oligonukleotydu w badanej sekwencji DNA.

Problemy sekwencjonowania DNA przez hybrydyzację zdefiniowane dla modelów dodatkowej informacji o powtórzeniach są w ogólności również problemami trudnymi obliczeniowo, tj. należą do klasy problemów silnie NP-trudnych. W związku z tym projektowane są przede wszystkim algorytmy przybliżone [40, 41]. Zaproponowano również algorytm dokładny typu podziału i ograniczeń przy założeniu występowania jedynie błędów negatywnych [42]. Powyższe algorytmy zostały wykorzystane do przeprowadzenia eksperymentów obliczeniowych. Otrzymane wyniki potwierdzają, że nawet częściowa informacja o powtórzeniach ma pozytywny wpływ na rekonstrukcję sekwencji.

3.3.7 Shotgun SBH

Nieco odmienne podejście do sekwencjonowania przez hybrydyzację przedstawiono w [55]. Należy przy tym zauważyć, że jest to metoda zaprojektowana dla resekwencjonowania i wymaga znajomości sekwencji homologicznej. Została ona nazwana przez autorów terminem *shotgun SBH* (ang. *shotgun sequencing by hybridization*).

Metoda ta również wykorzystuje informację o zawartości spektrum analizowanej sekwencji, przy czym informacja ta jest uzyskiwana w odmienny sposób. Podstawową różnicą jest budowa chipu DNA. W podejściu klasycznym zawiera on sondy reprezentujące pełną bibliotekę oligonukleotydów o zadanej długości l. Taki chip jest umieszczany w roztworze zawierającym wiele kopii analizowanej sekwencji DNA. W przypadku shotgun SBH chip zawiera fragmenty analizowanej sekwencji i jest umieszczany w roztworze oznakowanych l-merów nazywanych *sondami*.

Taki chip tworzony jest w następujący sposób. Analizowana sekwencja jest dzielona na fragmenty o długości ok. 200 nukleotydów, które nazywane są *cechami* (ang. *features*). Każdy taki fragment jest łączony z pewną sekwencją o długości 50 nukleotydów (zwaną *uniwersalnym łącznikiem*, ang. *universal linker*) w taki sposób, że powstaje kolista cząsteczka DNA (koniec 5' jest łączony z końcem 3'). Taka cząsteczka DNA jest następnie mocowana do powierzchni chipu DNA za pomocą uniwersalnego łącznika i przeprowadzana jest amplifikacja za pomocą metody RCA (ang. *Rolling-Circle Amplification*). W rezultacie chip DNA zawiera sekwencje składające się z wielu następujących po sobie kopii danej cechy.

W klasycznym SBH sygnały dla wszystkich sond zbierane są w tym samym momencie. Z powodu różnej temperatury topnienia poszczególnych l-merów trudno jest dobrać odpowiednie warunki eksperymentu biochemicznego. Próbą rozwiązania tego problemu

jest wykorzystanie izotermicznych bibliotek (opis w Rozdziale 3.3.5). W przypadku shotgun SBH problem ten rozwiązano przeprowadzając eksperyment biochemiczny niezależnie dla każdej sondy, co umożliwia indywidualne dostosowanie warunków eksperymentu dla danego l-meru.

Wynikiem eksperymentu biochemicznego jest spektrum $S(F)$ dla każdej cechy F, które zawiera zarejestrowane wartości sygnałów dla poszczególnych sond. Na podstawie zawartości danego spektrum i dostępnej sekwencji homologicznej określany jest fragment analizowanej sekwencji, który jest reprezentowany przez dane spektrum. Następnie krok po kroku wyznaczane są kolejne nukleotydy analizowanej sekwencji DNA. Do określenia nukleotydu na danej pozycji wykorzystywany jest model probabilistyczny. Bazuje on na wartościach sygnałów sond, przy czym dla danej pozycji wykorzystywane są jedynie sygnały sond z tych spektr, które reprezentują fragment zawierający pozycję rozpatrywaną w danym momencie (przypisanie spektrum do danego fragmentu analizowanej sekwencji jest realizowane w korku poprzedzającym). Szczegóły obliczeń zostały opisane w [55].

Rozdział 4

Podstawy matematyczne i informatyczne

4.1 Podstawy złożoności obliczeniowej

Jednym z głównych nurtów informatyki teoretycznej jest *złożoność obliczeniowa*. Zajmuje się ona określaniem ilości zasobów (np. czas, rozmiar pamięci) niezbędnych do rozwiązania problemów obliczeniowych. Analizuje ona również przyczyny, dla których pewne problemy są niezwykle trudne do rozwiązania przez komputery (por. [50]).

Problemy obliczeniowe będące przedmiotem zainteresowania teorii złożoności obliczeniowej są w ogólności zadaniami, które mogą zostać rozwiązane za pomocą komputera lub innej maszyny liczącej. Dany problem obliczeniowy składa się z opisu danych wejściowych oraz opisu wymagań, które musi spełnić rozwiązanie.

Def. 4.1.1. Problem obliczeniowy Π jest kolekcją uporządkowanych par (I, R), gdzie I jest instancją problemu, a R jest odpowiedzią (rozwiązaniem, wynikiem) dla tej instancji. Zbiór wszystkich instancji problemu Π jest oznaczany przez D_{Π}.

Można wyróżnić następujące typy problemów obliczeniowych:

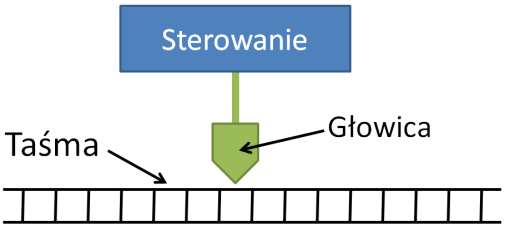
- **decyzyjny** - wymaga udzielenia odpowiedzi TAK lub NIE na zadane pytanie,
- **przeszukiwania** - sprowadza się do wyszukania dowolnego rozwiązania spełniającego warunki określone w definicji problemu lub stwierdzenia, że takie rozwiązanie nie istnieje,
- **optymalizacyjny** - jest to specyficzny problem przeszukiwania, którego celem jest znalezienie najlepszego rozwiązania wśród wszystkich możliwych rozwiązań.

W przypadku problemów optymalizacyjnych do oceny rozwiązań wykorzystywana jest *funkcja kosztu*. Jeżeli problem polega na znalezieniu rozwiązania o minimalnej wartości funkcji kosztu, to jest on nazywany *problemem minimalizacji*. W przypadku poszukiwania rozwiązania o maksymalnej wartości funkcji kosztu problem nazywany jest *problemem maksymalizacji*.

Zarówno instancja I jak i rozwiązanie R są kodowane jako *słowa* zbudowane nad pewnym *alfabetem* Σ, który jest skończonym, niepustym zbiorem symboli, a słowo to uporządkowana sekwencja symboli tego alfabetu. Długość słowa x oznaczana jest przez $|x|$.

Metodę przetworzenia słowa x kodującego instancję I na odpowiedź R precyzuje *algorytm*. Jest to szczegółowy opis, który krok po kroku określa sposób rozwiązania problemu. W przypadku problemu decyzyjnego odpowiedzią R jest TAK lub NIE. W przypadku problemu przeszukiwania i problemu optymalizacyjnego w wyniku przetworzenia słowa x zgodnie z danym algorytmem powstaje słowo y i to ono stanowi odpowiedź R.

Uniwersalnym sposobem zapisu algorytmu jest model *maszyny Turinga* (and. *Turing machine*). Dysponuje ona jedną strukturą danych - taśmą zawierającą ciąg symboli. W danym momencie może zostać odczytany lub zapisany jeden symbol. Służy do tego głowica, która może być przesuwana po taśmie w lewo lub w prawo. Sam sposób przetwarzania słowa x definiuje program sterujący (tzw. *funkcja przejścia*). Ideę maszyny Turinga prezentuje Rysunek 4.1.



Rysunek 4.1: Model maszyny Turinga.

Def. 4.1.2. Deterministyczna maszyna Turinga (ang. *Deterministic Turing Machine*, DTM) M to uporządkowana czwórka (K, Σ, δ, s), w której K jest skończonym zbiorem stanów, $s \in K$ jest stanem początkowym, Σ jest zbiorem symboli (alfabetem maszyny M), a δ jest funkcją przejścia. Σ zawiera dwa symbole specjalne: symbol pusty $\sqcup$ oraz symbol końcowy $\triangleright$. Wyróżnia się również trzy stany specjalne: stan końcowy h, stan akceptujący "tak" oraz stan odrzucający "nie". Kierunek ruchu głowicy oznaczany jest następująco: w lewo $\leftarrow$, w prawo $\rightarrow$ oraz pozostanie w miejscu $-$. Zakłada się, że K i Σ są zbiorami rozłącznymi i żaden z nich nie zawiera stanów h, "tak", "nie" ani symboli $\leftarrow$, $\rightarrow$, $-$. Funkcja przejścia δ odwzorowuje $K \times \Sigma$ w $(K \cup \{h$, "tak", "nie"$\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$ [50].

Funkcja przejścia δ definiuje dla każdej kombinacji stanu $q \in K$ i odczytanego symbolu $\sigma \in \Sigma$ trójkę (p, ρ, D), gdzie p jest następnym stanem, ρ jest symbolem zapisywanym w miejscu σ, a D określa kierunek ruchu głowicy $D \in \{\leftarrow, \rightarrow, -\}$. Ponadto symbol końcowy $\triangleright$ nie może być nadpisany innym symbolem, a w przypadku jego odczytania ruch głowicy odbywa się zawsze w prawo, tj. jeżeli dla stanów q i p spełniony jest warunek $\delta(q, \triangleright) = (p, \rho, D)$, to $\rho = \triangleright$ a $D = \rightarrow$.

Początkowo maszyna znajduje się w stanie s, taśma zawiera *słowo wejściowe* x reprezentujące instancję problemu I poprzedzone symbolem $\triangleright$, a głowica jest ustawiona na początku taśmy, tj. na symbolu $\triangleright$. Maszyna wykonuje kolejno kroki zgodnie ze zdefiniowaną funkcją przejść δ: odczytuje symbol znajdujący się aktualnie pod głowicą, zmienia stan, zapisuje odpowiedni symbol i przesuwa głowicę.

Maszyna *zatrzymuje się*, jeżeli zostanie osiągnięty stan końcowy h, stan akceptujący

"tak" lub stan odrzucający "nie". Osiągnięcie stanu "tak" oznacza, że maszyna *akceptuje* słowo wejściowe, a zakończenie przetwarzania w stanie "nie" oznacza, że maszyna *odrzuca* słowo wejściowe. Jeżeli maszyna zatrzymuje się dla słowa wejściowego x, to wynikiem $M(x)$ jest TAK jeżeli maszyna akceptuje x, NIE gdy maszyna odrzuca x lub słowo y zapisane na taśmie w momencie zatrzymania maszyny w przypadku osiągnięcia stanu h. Możliwe jest również, że maszyna M nigdy nie zakończy obliczeń dla słowa wejściowego x. W takiej sytuacji wynik oznacza się przez $M(x) = \nearrow$.

Czas wymagany do przetworzenia słowa x przez maszynę M wynosi t, jeżeli maszyna M zatrzymuje się po wykonaniu t kroków. Jeżeli $M(x) = \nearrow$, to za czas wymagany przez M dla x uznaje się ∞. Przyjmuje się, że maszyna M działa w czasie $f(n)$, jeżeli dla dowolnego słowa wejściowego x maszyna M wymaga do jego przetworzenia czasu równego co najwyżej $f(|x|)$, gdzie $|x|$ jest długością słowa wejściowego x. Funkcja $f(n)$ jest *ograniczeniem czasowym* maszyny M [50].

Ograniczenia czasowe mogą posłużyć do pogrupowania problemów w *klasy złożoności czasowej*. Do danej klasy złożoności czasowej $TIME(f(n))$ należą wszystkie te problemy, które mogą zostać rozwiązane za pomocą maszyny Turinga w czasie co najwyżej $f(n)$. W praktyce do oceny złożoności zamiast konkretnej funkcji wykorzystuje się miarę asymptotycznego tempa wzrostu funkcji. Do określenia tej miary służy notacja O. Dla dwóch funkcji $f(n)$ i $g(n)$ określonych na zbiorze liczb naturalnych $\mathbb{N}$ o wartościach w zbiorze $\mathbb{N}$ zapis $f(n) = O(g(n))$ oznacza, że istnieją takie dwie liczby naturalne c i n_0, że dla każdego $n \geqslant n_0$ zachodzi $f(n) \leqslant c \cdot g(n)$. Wskazuje to, że funkcja f rośnie najwyżej tak szybko jak funkcja g (funkcja f jest rzędu g).

Jedną z najważniejszych klas złożoności jest klasa P. Jest to klasa wszystkich tych problemów, które mogą zostać rozwiązane za pomocą deterministycznej maszyny Turinga w czasie wielomianowym. Klasa P jest więc sumą wszystkich klas złożoności czasowej $TIME(O(n^k))$, gdzie k jest pewną liczbą całkowitą większą od zera.

Sposób działania maszyny opisanej w Def. 4.1.2 jest w pełni deterministyczny. Funkcja przejścia δ zdefiniowana dla DTM jednoznacznie określa kolejną operację do wykonania. Jeżeli w roli programu sterującego zamiast funkcji δ zastosuje się *relację przejść* Δ, to w danym kroku możliwe będzie wykonanie jednej z kilku potencjalnych operacji. Maszyna, której sterowanie oparte jest na relacji Δ, nazywana jest *niedeterministyczną maszyną Turinga* (ang. *Non-Deterministic Turnig Machine*, NDTM).

Def. 4.1.3. Niedeterministyczna maszyna Turinga N to uporządkowana czwórka (K, Σ, Δ, s), w której elementy K, Σ i s są zdefiniowane identyczne jak w przypadku deterministycznej maszyny Turinga (Def. 4.1.2), tj. K jest skończonym zbiorem stanów, $s \in K$ jest stanem początkowym, Σ jest zbiorem symboli. Natomiast Δ, w odróżnieniu do funkcji przejść δ, jest relacją $\Delta \subset (K \times \Sigma) \times [(K \cup \{h,$ "tak", "nie"$\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}]$ [50].

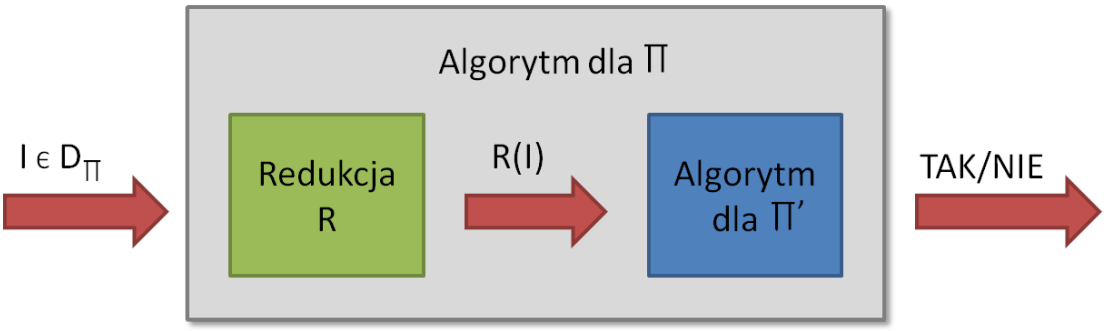
Sam proces przetwarzania słowa wejściowego przez NDTM jest praktycznie identyczny jak w przypadku DTM. Kluczową różnicą jest zdefiniowanie sposobu udzielania odpowiedzi. Niedeterministyczna maszyna Turinga *akceptuje* słowo wejściowe, jeżeli istnieje pewien ciąg niedeterministycznych (losowych) wyborów prowadzący do stanu "tak". NDTM *odrzuca* słowo wejściowe, jeżeli nie istnieje żaden ciąg wyborów prowadzących do stanu "tak".

Przyjmuje się, że niedeterministyczna maszyna Turinga działa w czasie $f(n)$, jeżeli dla dowolnego słowa wejściowego x istnieje ciąg k niedeterministycznych wyborów prowadzący do jednego ze stanów: h, "tak" lub "nie", taki że $k \leqslant f(|x|)$. W przypadku NDTM do złożoności czasowej wliczane są więc jedynie obliczenia bezpośrednio prowadzące do uzyskania rozwiązania, a łączny czas obliczeń może być wykładniczo większy.

Zbiór problemów rozwiązywalnych przez niedeterministyczną maszynę Turinga w czasie $f(n)$ tworzy klasę $NTIME(f(n))$. Najważniejszą *klasą złożoności niedeterministycznej* jest klasa NP, która jest sumą wszystkich klas $NTIME(O(n^k))$, gdzie k jest pewną liczbą całkowitą większą od zera [50]. $P \subseteq NP$, gdyż deterministyczne maszyny Turinga są specyficznym typem niedeterministycznych maszyn Turinga, dla których Δ jest funkcją. Zakłada się również, że $P \neq NP$, ale jest to wciąż otwartym problemem będącym jednym z najważniejszych zagadnień złożoności obliczeniowej.

Przynależność problemu do danej klasy złożoności wskazuje jego trudność i umożliwia pewne porównanie go z innymi problemami. Jednak bezpośrednim sposobem porównania problemów i wykazanie, że jeden z nich jest przynajmniej tak trudny jak drugi, jest zastosowanie *redukcji*. Jeżeli problem Π redukuje się do problemu Π', to problem Π' jest co najmniej tak trudny jak problem Π. Przykład redukcji przedstawia Rysunek 4.2.

Def. 4.1.4. Problem decyzyjny Π redukuje się do problemu decyzyjnego Π', jeżeli istnieje redukcja R umożliwiająca dla każdej instancji $I \in D_\Pi$ utworzenie równoważnej jej instancji $R(I) \in D_{\Pi'}$, tj. odpowiedź na pytanie sformułowane przez $R(I)$ jest jednocześnie odpowiedzią dla I [50].



Rysunek 4.2: Rozwiązanie problemu Π przez użycie redukcji do problemu Π'.

Jeżeli redukcja dowolnej instancji jednego problemu do instancji drugiego problemu może być zrealizowana przez DTM w czasie wielomianowym, to nazywana jest ona *redukcją wielomianową*. Warto również zaznaczyć, że w przypadku redukcji problemu Π do problemu Π' dopuszcza się wielokrotne rozwiązywanie problemu Π' w celu znalezienia odpowiedzi dla problemu Π. Jeżeli do uzyskania odpowiedzi dla Π wystarczy jednokrotne rozwiązanie problemu Π', to takie przekształcenie nazywa się *transformacją*. Jeżeli może ona zostać zrealizowana przez DTM w czasie wielomianowym, to nazywana jest ona *transformacją wielomianową* i jest oznaczana symbolem $\propto$.

Operacja redukcji umożliwia wydzielenie w danej klasie złożoności pewnych problemów, do których można zredukować dowolny inny problem tej klasy. Problemy te są *problemami zupełnymi* w swojej klasie złożoności.

Def. 4.1.5. Niech C będzie pewną klasą złożoności oraz niech Π będzie problemem należącym do tej klasy. Problem Π jest C-zupełny, jeżeli dowolny inny problem $\Pi' \in C$ redukuje się do Π [50].

Jedną z najistotniejszych klas problemów zupełnych jest klasa problemów NP-zupełnych.

Def. 4.1.6. Problem Π jest NP-zupełny, jeżeli dla dowolnego innego problemu $\Pi' \in NP$ istnieje transformacja wielomianowa problemu Π' do problemu Π [50].

Wśród problemów można wyróżnić specjalną kategorię *problemów liczbowych*. Są to takie problemy, których rozmiar n dowolnej instancji I nie ogranicza wielomianowo maksymalnej wielkości liczb występujących w tej instancji.

Def. 4.1.7. Problem Π jest problemem liczbowym, jeżeli dla dowolnego wielomianu p istnieje taka instancja problemu I, że $max(I) > p(n)$, gdzie $max(I)$ jest największą liczbą występującą w opisie instancji I, a n jest rozmiarem instancji I.

Wśród problemów NP-zupełnych istnieją pewne problemy liczbowe, które można rozwiązać za pomocą *algorytmu pseudowielomianowego*. Algorytm pseudowielomianowy to taki algorytm, którego czas działania jest ograniczony przez wielomian zależny od rozmiaru n instancji I i największej liczby $max(I)$ występującej w instancji I.

Niech Π będzie pewnym problemem, a Π_p będzie jego podproblemem zawierającym jedynie te instancje $I \in D_\Pi$, dla których spełnione jest $max(I) \leqslant p(n)$, gdzie $max(I)$ jest największą liczbą występującą w instancji I, n jest rozmiarem instancji I, a p jest pewnym wielomianem.

Def. 4.1.8. Problem Π jest silnie NP-zupełny, jeżeli jego podproblem Π_p jest NP-zupełny.

Kolejną istotną klasą problemów są problemy *NP-trudne*. Są to problemy, których rozwiązanie jest co najmniej tak trudne jak rozwiązanie dowolnego problemu z klasy NP, przy czym sam problem NP-trudny nie koniecznie musi należeć do klasy NP. Do wykazania NP-trudności danego problemu wykorzystywana jest *wielomianowa redukcja Turinga* oznaczana przez $\propto_T$.

Def. 4.1.9. Wielomianowa redukcja Turinga problemu Π' do problemu Π, oznaczana przez $\Pi' \propto_T \Pi$, to algorytm A rozwiązujący problem Π', który wykorzystuje pewną hipotetyczną procedurę B rozwiązującą problem Π. Co więcej, jeżeli algorytm B może zostać wykonany w czasie wielomianowym przez DTM, to również algorytm A może zostać wykonany przez DTM w czasie wielomianowym [7].

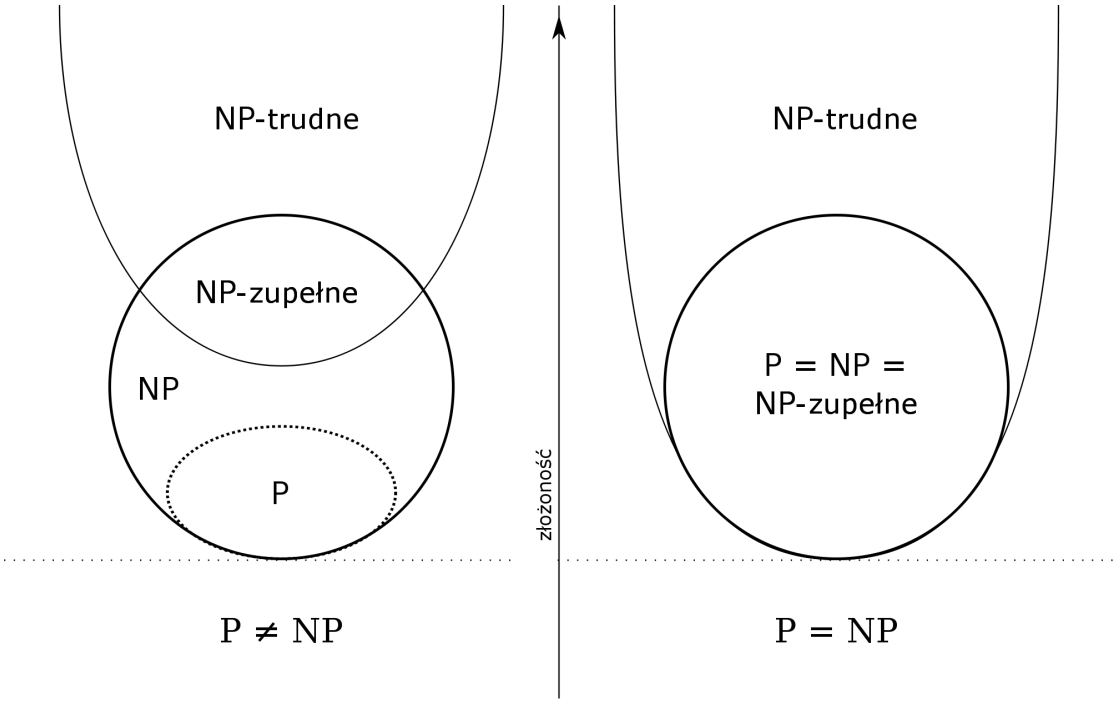
Formalna definicja problemów NP-trudnych jest następująca.

Def. 4.1.10. Problem Π jest problemem NP-trudnym jeżeli istnieje NP-zupełny problem Π', który można zredukować do problemu Π za pomocą wielomianowej redukcji Turinga, tj. $\Pi' \propto_T \Pi$ [7].

Wśród problemów NP-trudnych można jeszcze wyróżnić problemy silnie NP-trudne.

Def. 4.1.11. Problem Π jest silnie NP-trudny, jeżeli istnieje pewien wielomian p, taki że problem Π_p jest NP-trudny.

Relacje pomiędzy omawianymi powyżej klasami złożoności czasowej prezentuje Rysunek 4.3. Przedstawia on zarówno wariant zakładający $P \neq NP$ jak i $P = NP$. Gdyby udowodniono $P = NP$, to w praktyce oznaczałoby możliwość rozwiązania w czasie wielomianowym przez DTM wszystkich problemów z klasy NP, ale nie przesądzałoby to o takiej możliwości dla problemów NP-trudnych. Warto również zwrócić uwagę na to, że wykazanie $P = NP$ implikuje również $P = NP = NP$-zupełne [50].



Rysunek 4.3: Relacje pomiędzy klasami złożoności czasowej problemów P, NP, NP-zupełnych, NP-trudnych.

4.2 Podstawy teorii algorytmów

Algorytm to szczegółowy opis, który krok po kroku określa sposób rozwiązania problemu, tj. metodę przetworzenie danych wejściowych na konkretne rozwiązanie. Dziedziną nauki zajmującą się badaniem algorytmów jest *algorytmika*.

Jednym z najważniejszych podziałów algorytmów jest podział ze względu na czas obliczeń. Jeżeli maksymalny czas działania algorytmu może zostać ograniczony przez wielomian zależny od rozmiaru danych wejściowych, to nazywany jest on *algorytmem wielomianowym*. Pozostałe algorytmy są określanie mianem *algorytmów wykładniczych*.

Dla problemów liczbowych (Def. 4.1.7) można wyróżnić *algorytmy pseudowielomianowe*. Ich czas obliczeń jest ograniczony wielomianem zależnym od rozmiaru instancji rozwiązywanego problemu oraz największej liczby występującej w opisie instancji. Jednakże w ogólności są to algorytmy wykładnicze.

Wymagany czas obliczeń wpływa na możliwość praktycznego wykorzystania danego algorytmu. W przypadku algorytmów wykładniczych czas niezbędny do uzyskania odpowiedzi ogranicza ich zastosowanie jedynie do instancji problemów o niewielkim rozmiarze. Sposobem na rozwiązanie tej trudności i uzyskanie rozwiązania w czasie wielomianowym jest zastosowanie *algorytmów przybliżonych* (*heurystyk*). Nie gwarantują one uzyskania dokładnego/optymalnego rozwiązania, ale umożliwiają znaczącą redukcję czasu obliczeń. W praktyce jest to czasami jedyny sposób na uzyskanie jakiegokolwiek rozwiązania w rozsądnym czasie.

Dla algorytmów przybliżonych rozwiązujących problemy optymalizacyjne (opis w Rozdziale 4.1) istnieje możliwość określenia w pewnych przypadkach o ile gorsze od rozwiązania dokładnego będzie w najgorszym przypadku uzyskane rozwiązanie. Algorytmy takie nazywane są *algorytmami aproksymacyjnymi*.

Def. 4.2.1. Niech Π będzie problemem optymalizacyjnym (maksymalizacji lub minimalizacji), I oznacza instancję tego problemu, A będzie algorytmem rozwiązującym problem Π, $OPT(I)$ oznacza wartość funkcji kosztu dla rozwiązania optymalnego instancji I, a $A(I)$ oznacza wartość funkcji kosztu dla rozwiązania generowanego dla instancji I przez algorytm A. Jeżeli istnieje $\epsilon > 0$, taki że dla każdej instancji $I \in D_\Pi$ problemu Π spełniony jest warunek $|A(I) - OPT(I)| \leqslant \epsilon \cdot OPT(I)$, to algorytm A jest algorytmem α-aproksymacyjnym, gdzie $\alpha = 1 + \epsilon$ dla problemu minimalizacji oraz $\alpha = 1 - \epsilon$ dla problemu maksymalizacji. Wartość α nazywana jest *gwarancją dokładności* (ang. *performance guarantee*) lub *współczynnikiem aproksymacji* [62].

Jedną z najprostszych heurystyk jest *algorytm zachłanny* (ang. *greedy algorithm*). Rozwiązuje on problem iteracyjnie, a w każdym kroku dokonuje wyboru, który w danym momencie wydaje się być najlepszą możliwą opcją (lokalnie optymalną). Nie ma żadnej gwarancji, że takie podejście doprowadzi do osiągnięcia rozwiązania optymalnego, ale skupienie się przy podejmowaniu decyzji jedynie na kolejnym kroku umożliwia znaczne skrócenie czasu obliczeń w porównaniu do algorytmu dokładnego.

Wśród algorytmów przybliżonych można również wyróżnić pewną klasę ogólnych metod zwanych *metaheurystykami*. Nie rozwiązują one bezpośrednio żadnego problemu, ale opisują pewną strategię uzyskania rozwiązania i niezbędne struktury danych. Metaheurystyki wykorzystuje się głównie do rozwiązywania problemów optymalizacyjnych. Poniżej opisano jedynie kilka wybranych.

Pewną grupą metaheurystyk są algorytmy *lokalnego przeszukiwania* (ang. *local search*). Algorytmy lokalnego przeszukiwania eksplorują przestrzeń potencjalnych rozwiązań przez iteracyjne wprowadzanie drobnych modyfikacji do bieżącego rozwiązania. Kolejne rozwiązania generowane przez algorytmy lokalnego przeszukiwania są więc do siebie bardzo podobne. Proces przeszukiwania ograniczony jest zazwyczaj limitem czasu lub maksymalną liczbą wykonywanych iteracji.

Wraz z upływem czasu obliczeń proces lokalnego przeszukiwania może mieć tendencję do skupiania się na przeszukiwaniu wąskiego fragmentu przestrzeni rozwiązań i wykonywaniu modyfikacji prowadzących do ponownego tworzenia wygenerowanych już wcześniej rozwiązań. Przykładowym sposobem radzenia sobie z tym problemem jest identyfikacja i blokowanie ruchów prowadzących do takich sytuacji. Algorytmem lokalnego przeszukiwania, który realizuje tą ideę, jest *algorytm przeszukiwania tabu* (ang. *tabu search*) [33]. Korzysta on z pewnych dodatkowych struktur danych, które zapewniają szersze przeszukiwanie przestrzeni rozwiązań. Kluczową strukturą jest *lista tabu*. Zawiera ona ostatnio wygenerowane rozwiązania lub pewne ich cechy. Lista tabu jest uwzględniana przy modyfikacji bieżącego rozwiązania. Zmiany prowadzące do uzyskania rozwiązania znajdującego na liście tabu lub do rozwiązania o cechach znajdujących się na tej liście są zabronione.

Zachowanie się mrówek poszukujących pożywienia stało się inspiracją do opracowania metaheurystyki zwanej algorytmem kolonii mrówek (ang. *Ant Colony Optimization*, ACO) [27]. ACO jest algorytmem iteracyjnym, a w każdej iteracji tworzone są rozwiązania dwojakiego rodzaju: generowane *od początku* i generowane *od końca*. Rozwiązanie generowane *od początku* to rozwiązanie zawierające inicjalnie jedynie pierwszy element rozwiązania, które jest rozbudowywane przez dodawanie na końcu kolejnych elementów. Rozwiązanie generowane *od końca* to rozwiązanie zawierające inicjalnie jedynie ostatni element rozwiązania, które jest rozbudowywane przez wstawianie na jego początku kolejnych dodawanych elementów. Do konstrukcji obu typów rozwiązań jest wykorzystywany *model feromonów*, który koduje wiedzę zebraną w trakcie dotychczasowych poszukiwań. Model feromonów zawiera dla każdego elementu rozwiązania informację o tym, jak

bardzo preferowane jest jego występowanie w rozwiązaniu bezpośrednio przed innym elementem. Informacja ta jest aktualizowana na koniec każdej iteracji na podstawie najlepszych rozwiązań znalezionych w trakcie dotychczasowych obliczeń.

4.3 Podstawy teorii grafów

4.3.1 Definicje

Teoria grafów to dział matematyki, którego przedmiotem zainteresowania są własności grafów. Grafy pozwalają przedstawić informacje w uporządkowany sposób, dlatego są wykorzystywane w wielu innych dziedzinach, np. informatyce, chemii, lingwistyce, socjologii. Poniżej przedstawiono formalne definicje podstawowych terminów wykorzystywanych w teorii grafów.

Def. 4.3.1. Graf nieskierowany G to uporządkowana para $G = (V, E)$, gdzie V jest zbiorem wierzchołków, a E jest zbiorem krawędzi, które są dwuelementowymi podzbiorami V: $E \subseteq \{\{u, v\} : u, v \in V\}$.

Def. 4.3.2. Graf skierowany (digraf) G to uporządkowana para $G = (V, A)$, gdzie V jest zbiorem wierzchołków, a A jest zbiorem łuków, które są uporządkowanymi parami wierzchołków z V: $A \subseteq V \times V$.

Def. 4.3.3. Graf ważony to graf, w którym każdej krawędzi/łukowi przypisano wagę będącą pewną liczbą (zwykle rzeczywistą).

Def. 4.3.4. Multigraf to taki graf, w którym dozwolone jest połączenie danej pary wierzchołków więcej niż jedną krawędzią/łukiem. W przypadku multigrafu zbiór krawędzi E/łuków A jest multizbiorem.

Def. 4.3.5. Graf pełny to graf nieskierowany zawierający krawędź pomiędzy każdą parą wierzchołków.

Def. 4.3.6. Graf spójny to graf, w którym dla każdego wierzchołka istnieje ścieżka do dowolnego innego wierzchołka.

Def. 4.3.7. Dla nieskierowanego grafu $G = (V, E)$ grafem liniowym $L(G)$ jest taki graf, w którym każdy wierzchołek reprezentuje pewną krawędź $e \in E$, a dwa wierzchołki w $L(G)$ są połączone krawędzią wtedy i tylko wtedy, gdy odpowiadające im krawędzie $e_1, e_2 \in E$ mają wspólny koniec w postaci wierzchołka $v \in V$.

Def. 4.3.8. Dla skierowanego grafu $G = (V, A)$ skierowanym grafem liniowym $L(G)$ jest taki graf, w którym każdy wierzchołek reprezentuje pewien łuk $a \in A$, a dwa wierzchołki w $L(G)$ są połączone łukiem wtedy i tylko wtedy, gdy dla odpowiadających im łuków $a_1, a_2 \in A$ wierzchołek $v \in V$ stanowiący koniec pierwszego łuku jest jednocześnie wierzchołkiem początkowym drugiego łuku.

Def. 4.3.9. Ścieżka to taki uporządkowany zbiór wierzchołków $(v_1, v_2, \ldots, v_n)$, że dla każdego $1 \leqslant i \leqslant n-1$ istnieje krawędź $\{v_i, v_{i+1}\} \in E$ w przypadku grafu nieskierowanego lub łuk $(v_i, v_{i+1}) \in A$ w przypadku grafu skierowanego. Wartość n wyznacza długość tej ścieżki.

Def. 4.3.10. Koszt ścieżki w grafie ważonym to suma wag krawędzi/łuków pomiędzy kolejnymi wierzchołkami ścieżki.

Def. 4.3.11. Ścieżka prosta to taka ścieżka, w której żaden wierzchołek się nie powtarza.

Def. 4.3.12. Ścieżka zamknięta to ścieżka kończąca się w wierzchołku początkowym.

Def. 4.3.13. Ścieżka Hamiltona to ścieżka przechodząca przez każdy z wierzchołków w grafie dokładnie raz.

Def. 4.3.14. Droga to taki uporządkowany zbiór krawędzi $(e_1, e_2, \ldots, e_n)$ w grafie nieskierowanym lub łuków $(a_1, a_2, \ldots, a_n)$ w grafie skierowanym, że dla każdego $1 \leqslant i \leqslant n-1$ wierzchołek końcowy krawędzi e_i/łuku a_i jest jednocześnie wierzchołkiem początkowym krawędzi e_{i+1}/łuku a_{i+1}.

Def. 4.3.15. Droga prosta zawiera każdą z krawędzi w grafie co najwyżej raz.

Def. 4.3.16. Droga Eulera przechodzi przez każdą z krawędzi w grafie dokładnie raz.

Def. 4.3.17. Cykl to taka droga prosta, której początkowy wierzchołek pierwszej krawędzi jest jednocześnie wierzchołkiem końcowym ostatniej krawędzi w drodze.

Def. 4.3.18. Cykl Eulera to cykl zawierający każdą z krawędzi grafu dokładnie raz.

Def. 4.3.19. Cykl Hamiltona to cykl przechodzący przez każdy z wierzchołków grafu dokładnie raz.

Def. 4.3.20. Graf acykliczny to taki graf, który nie zawiera cyki.

Def. 4.3.21. Graf eulerowski to taki graf, który zawiera cykl Eulera.

Def. 4.3.22. Graf semieulerowski to taki graf, który zawiera drogę Eulera.

Def. 4.3.23. Drzewo to taki graf, który jest spójny i acykliczny, tj. istnieje ścieżka pomiędzy dowolnymi dwoma wierzchołkami i jest to jedyna możliwa droga pomiędzy nimi.

Def. 4.3.24. Drzewo rozpinające grafu $G = (V, E)$ to drzewo zawierające wszystkie wierzchołki ze zbioru V, a zbiór krawędzi drzewa jest podzbiorem zbioru E.

Def. 4.3.25. Krawędź e jest *incydentna* z wierzchołkiem v, jeżeli wierzchołek v jest jednym z końców krawędzi e. Łuk a jest *incydentny* z wierzchołkiem v, jeżeli wierzchołek v jest wierzchołkiem początkowym lub wierzchołkiem końcowym krawędzi a.

Def. 4.3.26. Wierzchołek v *sąsiaduje* z wierzchołkiem u, jeżeli istnieje krawędź $e = \{u, v\}$ w przypadku grafu nieskierowanego lub łuk $a = (u, v)$ w przypadku grafu skierowanego.

Def. 4.3.27. Stopień wierzchołka v w przypadku grafu nieskierowanego oznacza liczbę krawędzi incydentnych z v. W przypadku grafu skierowanego można wyróżnić stopień wejściowy i stopień wyjściowy wierzchołka v. Oznaczają one odpowiednio liczbę krawędzi kończących się w v i mających początek w v.

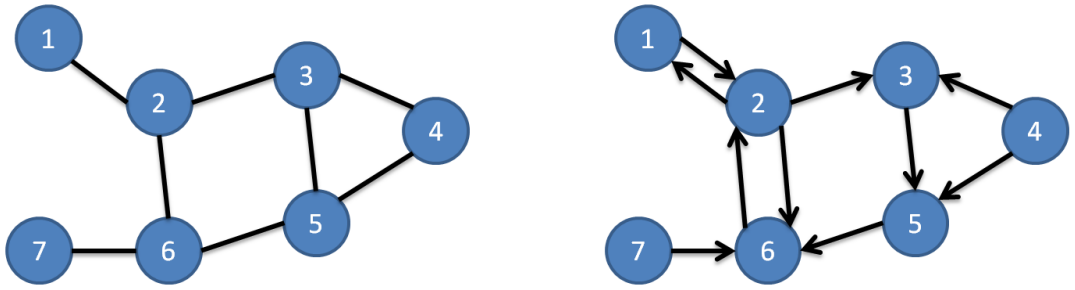
Def. 4.3.28. Skojarzenie M dla nieskierowanego grafu $G = (V, E)$ to taki podzbiór krawędzi, że żadne dwie krawędzie należące do M nie mają wspólnego końca.

Def. 4.3.29. Skojarzenie M dla nieskierowanego grafu $G = (V, E)$ nazywamy doskonałym, jeżeli kojarzy wszystkie wierzchołki z V.

4.3.2 Sposoby reprezentacji grafów

Istnieje wiele sposobów reprezentacji grafów. Jedną z nich jest reprezentacja graficzna. Składa się ona z punktów lub okręgów odpowiadających wierzchołkom oraz łączących je linii odpowiadającym krawędziom. W przypadku grafów skierowanych kierunek łuku jest oznaczany strzałką. Dodatkowo wierzchołki i krawędzie mogą być etykietowane literami lub liczbami. Dwa przykładowe grafy prezentuje Rysunek 4.4.

Grafy mogą zostać również przedstawione przy użyciu różnych struktur danych.



Rysunek 4.4: Reprezentacja graficzna grafów. Po lewej przykładowy graf nieskierowany. Po prawej przykładowy graf skierowany.

Poniżej zaprezentowano jedynie kilka z nich. Jedną z najprostszych struktur umożliwiających zdefiniowanie grafu jest *macierz sąsiedztwa* A (ang. *adjacency matrix*). Jest to macierz o rozmiarach $|V| \times |V|$. Wierzchołki są ponumerowane ($V = \{v_1, v_2, \ldots, v_n\}$), a wartość na przecięciu i-tego wiersza i j-tej kolumny A_{ij} odpowiada liczbie krawędzi pomiędzy wierzchołkami v_i i v_j w przypadku grafu nieskierowanego oraz liczbie łuków z v_i do v_j w przypadku grafu skierowanego. Warto zauważyć, że w przypadku grafów nieskierowanych macierz sąsiedztwa jest macierzą symetryczną. Macierze sąsiedztwa dla grafów przedstawionych na Rysunku 4.4 prezentują Równania 4.1. Macierz A odpowiada grafowi nieskierowanemu, a macierz A' definiuje graf skierowany.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad A' = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.1}$$

Kolejną macierzową strukturą danych umożliwiająca zdefiniowanie grafu jest *macierz incydencji* M (ang. *incidence matrix*). Jest to macierz o rozmiarze $|V| \times |E|$. Poszczególne wierzchołki i krawędzie są ponumerowane. W przypadku grafu nieskierowanego wartości M_{ij} są zdefiniowane następująco:

$$M_{ij} = \begin{cases} 1 & \text{jeżeli krawędź } e_j \text{ jest incydentna z wierzchołkiem } v_i \\ 0 & \text{w przeciwnym razie} \end{cases} \tag{4.2}$$

W przypadku grafu skierowanego macierz incydencji jest zdefiniowana wzorem:

$$M_{ij} = \begin{cases} -1 & \text{jeżeli wierzchołek } v_i \text{ jest wierzchołkiem początkowym łuku } a_j \\ 1 & \text{jeżeli wierzchołek } v_i \text{ jest wierzchołkiem końcowym łuku } a_j \\ 0 & \text{w przeciwnym razie} \end{cases} \tag{4.3}$$

Macierze incydencji dla przykładowych grafów zaprezentowanych na Rysunku 4.4 przedstawiają Równanie 4.4 i Równanie 4.5. Macierz M odpowiada grafowi nieskierowanemu, a macierz M' definiuje graf skierowany.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

$$M' = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.5}$$

Powyższe struktury zawierają jedynie informację, które wierzchołki w grafie są połączone. W przypadków grafów ważonych konieczne jest również zakodowanie informacji o wagach poszczególnych krawędzi. Umożliwia to *macierz odległości* D (ang. *distance matrix*). Jest to macierz o rozmiarach $|V| \times |V|$. Wierzchołki są ponumerowane ($V = \{v_1, v_2, \ldots, v_n\}$), a wartość na przecięciu i-tego wiersza i j-tej kolumny D_{ij} odpowiada wadze krawędzi/łuku pomiędzy wierzchołkami v_i i v_j. Warto zauważyć, że dla grafów nieskierowanych macierz D jest macierzą symetryczną. Dla odróżnienia macierz D dla grafów skierowanych nazywana jest *asymetryczną macierzą odległości*.

Graf może zostać również opisany poprzez struktury danych oparte na listach jednokierunkowych. Taką reprezentacją są *listy sąsiedztwa* (ang. *adjacency list*). Dla każdego wierzchołka określona jest lista sąsiadujących z nim wierzchołków. Listy sąsiedztwa dla przykładowych grafów przedstawionych na Rysunku 4.4 prezentuje Rysunek 4.5.

v_1	$\rightarrow$	v_2			v_1	$\rightarrow$	v_2		
v_2	$\rightarrow$	v_1	v_3	v_6	v_2	$\rightarrow$	v_1	v_3	v_6
v_3	$\rightarrow$	v_2	v_4	v_5	v_3	$\rightarrow$	v_5		
v_4	$\rightarrow$	v_3	v_5		v_4	$\rightarrow$	v_3	v_5	
v_5	$\rightarrow$	v_3	v_4	v_6	v_5	$\rightarrow$	v_6		
v_6	$\rightarrow$	v_2	v_5	v_7	v_6	$\rightarrow$	v_2		
v_7	$\rightarrow$	v_6			v_7	$\rightarrow$	v_6		

Rysunek 4.5: Reprezentacja grafów zaprezentowanych na Rysunku 4.4 w postaci list sąsiedztwa. Po lewej listy sąsiedztwa dla grafu nieskierowanego. Po prawej listy sąsiedztwa dla grafu skierowanego.

4.3.3 Wybrane problemy grafowe

Formalną definicję problemu obliczeniowego przedstawiono w Rozdziale 4.1 (Def. 4.1.1). Poniżej zaprezentowano definicje wybranych problemów obliczeniowych związanych z grafami.

Problem 4.3.1 - problem drogi Eulera
Mając dany graf G (nieskierowany lub skierowany) należy określić, czy istnieje w nim droga Eulera.

Problem 4.3.2 - problem minimalnej s-t ścieżki
Mając dany ważony graf G (nieskierowany lub skierowany) należy wskazać w nim ścieżkę P^{st} z wierzchołka s do wierzchołka t o minimalnym koszcie, tj. ścieżkę z s do t, której koszt jest najmniejszy spośród wszystkich możliwych ścieżek z s do t.

Problem 4.3.3 - problem minimalnej k-ścieżki
Mając dany ważony graf G (nieskierowany lub skierowany) należy wskazać w nim ścieżkę P_k o minimalnym koszcie zawierającą co najmniej k wierzchołków, tj. ścieżkę zawierającą co najmniej k wierzchołków, której koszt jest najmniejszy spośród wszystkich ścieżek o przynajmniej k wierzchołkach.

Problem 4.3.4 - problem minimalnej s-t k-ścieżki
Mając dany ważony graf G (nieskierowany lub skierowany) należy wskazać w nim ścieżkę P_k^{st} z wierzchołka s do wierzchołka t o minimalnym koszcie zawierającą przynajmniej k wierzchołków.

Problem 4.3.5 - problem cyklu Eulera
Mając dany graf G (nieskierowany lub skierowany) należy określić, czy istnieje w nim cykl Eulera.

Problem 4.3.6 - problem komiwojażera
Mając dany ważony graf G (nieskierowany lub skierowany) należy wskazać w nim cykl Hamiltona C_H o minimalnym koszcie. Problem komiwojażera nazywany jest również TSP (ang. *Travelling Salesman Problem*).

Problem 4.3.7 - problem minimalnego drzewa rozpinającego
Mając dany ważony, nieskierowany graf $G = (V, E)$ należy wskazać drzewo rozpinające T grafu G o minimalnym koszcie.

Problem 4.3.8 - problem minimalnego k-drzewa rozpinającego
Mając dany ważony, nieskierowany graf $G = (V, E)$ należy wskazać w nim drzewo rozpinające T_k o minimalnym koszcie zawierające przynajmniej k wierzchołków ze zbioru V.

Problem 4.3.9 - problem Orienteering
Mając dany ważony graf G (nieskierowany lub skierowany), określoną funkcję wartości wierzchołków $\psi : V \to \mathbb{R}$ oraz pewien budżet B należy wskazać w grafie G ścieżkę o koszcie nie przekraczającym B, której wartość Ψ wyznaczona przez sumę wartości odwiedzonych wierzchołków będzie maksymalna.

4.4 Algorytm aproksymacyjny dla problemu komiwojażera w grafie skierowanym

Opracowanie algorytmu aproksymacyjnego dla danego problemu w grafie nieskierowanym nie jest równoznaczne z możliwością jego zastosowania dla analogicznego problemu

w grafie skierowanym. W pewnych przypadkach konieczne jest przygotowanie nowego algorytmu. Może on wykorzystywać jako podprocedurę istniejący algorytm dla problemu w grafie nieskierowanym. W takim przypadku na podstawie danego grafu skierowanego tworzony jest graf nieskierowany, poszukiwane jest rozwiązanie w grafie nieskierowanym, a następnie na jego podstawie konstruowane jest rozwiązanie dla problemu w grafie skierowanym.

Rozwiązanie problemu komiwojażera w grafie skierowanym (ang. *Asymmetric Traveling Salesman Problem*, ATSP) wykorzystujące algorytm aproksymacyjny dla problemu komiwojażera w grafie nieskierowanym zostało zaproponowane w [39]. Kosztem tego podejścia jest dwukrotne zwiększenie rozmiaru instancji problemu. Poniżej zaprezentowano jedynie ogólną ideę. Szczegóły można znaleźć we wspomnianej pracy.

Niech $G = (V, A)$ będzie grafem skierowanym ze zdefiniowaną asymetryczną macierzą odległości D o rozmiarze $n \times n$, gdzie $n = |V|$. Zakłada się, że macierz D spełnia asymetryczną nierówność trójkąta, tj. $D_{ij} \leqslant D_{ik} + D_{kj}$ dla dowolnych wartości $i \neq j \neq k$, gdzie $1 \leqslant i, j, k \leqslant n$. Wartości d_{min} i d_{max} oznaczają odpowiednio najmniejszą i największą wagę (koszt) w macierzy D i spełniają następujące ograniczenie $-\infty < d_{min} \leqslant d_{max} < \infty$. Istnieje możliwość skonstruowania macierzy odległości D' o takim samym rozmiarze, dla której $\frac{d'_{max}}{d'_{min}} < \frac{4}{3}$.

$$D'_{ij} = \begin{cases} 0 & \text{jeżeli } i = j \\ D_{ij} & \text{jeżeli } 4d_{min} - 3d_{max} > 0, i \neq j \\ D_{ij} + 3d_{max} - 4d_{min} + \epsilon & \text{w przeciwnym razie} \end{cases} \tag{4.6}$$

gdzie ϵ jest małą liczbą większą od 0.

Na podstawie asymetrycznej macierzy odległości D' tworzona jest symetryczna macierz odległości $\overline{D}$ o rozmiarze $2n \times 2n$, która również spełnia asymetryczną nierówność trójkąta.

$$\overline{D} = \left[\begin{array}{c|c} \infty & (D')^T \\ \hline D' & \infty \end{array}\right] \tag{4.7}$$

Dla uproszczenia niech $[i]$ oznacza $i + n$, gdzie $1 \leqslant i \leqslant n$, np. $[1] = 1 + n, [n] = 2n$. Wierzchołki v_i i $v_{[i]}$ są nazywane *komplementarną* parą wierzchołków. Wierzchołek v_i jest nazywany *wierzchołkiem rzeczywistym*, a wierzchołek $v_{[i]}$ nazywany jest *wierzchołkiem wirtualnym*. Dla dowolnej pary i, j spełnione są następujące równania.

$$\overline{D}_{ij} = \overline{D}_{[i][j]} = \infty; \quad \overline{D}_{[i]j} = \overline{D}_{j[i]} = D'_{ij}; \quad \overline{D}_{[i]i} = \overline{D}_{i[i]} = 0 \tag{4.8}$$

Cykl Hamiltona $\overline{C}_H$ dla grafu o macierzy odległości $\overline{D}$ jest nazywany *dopuszczalnym*, jeżeli wyznaczona przez niego sekwencja wierzchołków zawiera naprzemiennie wierzchołki rzeczywiste i wirtualne oraz gdy w sekwencji tej po wystąpieniu danego wierzchołka kolejnym wierzchołkiem jest jego wierzchołek komplementarny.

$$\overline{C}_H = (v_{i_1}, v_{[i_1]}, v_{i_2}, v_{[i_2]}, \ldots, v_{i_n, v_{[i_n]}}) \tag{4.9}$$

Na podstawie cyklu Hamiltona $\overline{C}_H$ o postaci zdefiniowanej przez Równanie 4.9 konstruowany jest cykl Hamiltona C_H dla oryginalnego grafu skierowanego zdefiniowanego przez macierz odległości D.

$$C_H = (v_{i_1}, v_{i_2}, \ldots, v_{i_n}) \tag{4.10}$$

Niech $\mathcal{H}^*_D$ oznacza zbiór optymalnych rozwiązań dla grafu skierowanego o asymetrycznej macierzy odległości D, a $\mathcal{H}^*_{\overline{D}}$ oznacza zbiór optymalnych rozwiązań dla grafu

nieskierowanego o symetrycznej macierzy odległości $\overline{D}$. Udowodniono, że $C_H \in \mathcal{H}_D^*$ wtedy i tylko wtedy, gdy $\overline{C}_H \in \mathcal{H}_{\overline{D}}^*$ [39].

Rozdział 5

Problemy obliczeniowe i istniejące modele grafowe związane z sekwencjonowaniem przez hybrydyzację

5.1 Problemy obliczeniowe dla sekwencjonowania przez hybrydyzację

Sekwencjonowanie DNA przez hybrydyzację składa się z dwóch etapów. Pierwszy z nich sprowadza się do przeprowadzenia eksperymentu biochemicznego, którego wynikiem jest zbiór oligonukleotydów. Ten zbiór krótkich sekwencji stanowiących fragmenty oryginalnej sekwencji DNA nazywany jest spektrum. Poszczególne oligonukleotydy wchodzące w skład spektrum mają tą samą długość l i są nazywane l-merami. Drugi etap to rekonstrukcja analizowanej sekwencji na podstawie spektrum. Szczegóły metody zostały opisane w Rozdziale 3.3.1, a niniejszy podrozdział skupia się na przedstawieniu definicji problemów obliczeniowych dla jej wariantów rozpatrywanych w niniejszej pracy.

Niech $S(Q)$ reprezentuje spektrum sekwencji Q. Zbiór $S(Q)$ zawiera słowa s_i zbudowane nad alfabetem {A, C, G, T} reprezentujące poszczególne oligonukleotydy ze spektrum sekwencji Q. Przy założeniu braku jakichkolwiek błędów (negatywnych i pozytywnych) problem obliczeniowy dla klasycznego sekwencjonowania przez hybrydyzację można zdefiniować następująco.

Problem 5.1.1 - klasyczne SBH bez błędów [15]
INSTANCJA: zbiór $S(Q)$ zawierający słowa s_i o długości l zbudowane nad alfabetem {A, C, G, T}, długość n sekwencji Q, taka że $|S(Q)| = n - l + 1$.
ODPOWIEDŹ: sekwencja Q' o długości n zawierająca wszystkie elementy zbioru $S(Q)$.

Powyższy problem może zostać sprowadzony do problemu drogi Eulera w grafie skierowanym [52] i może zostać rozwiązany w czasie wielomianowym. Jednakże w praktyce otrzymane spektrum zazwyczaj zawiera błędy. Zakładając możliwość występowania błędów dowolnego typu konieczne jest rozwiązanie następującego problemu, dla którego udowodniono, że należy do klasy problemów silnie NP-trudnych [15].

Problem 5.1.2 - klasyczne SBH z błędami dowolnego typu [15]
INSTANCJA: zbiór $S(Q)$ zawierający słowa s_i o długości l zbudowane nad alfabetem {A, C, G, T} , długość n sekwencji Q.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n zawierająca maksymalną liczbę elementów zbioru $S(Q)$, przy czym Q' może zawierać ponadto pewne słowa o długości l nie będące częścią $S(Q)$.

SBH z częściową informacją o powtórzeniach (szczegóły w Rozdziale 3.3.6) zakłada możliwość pozyskania w trakcie eksperymentu biochemicznego pewnej dodatkowej informacji, na podstawie której można by wnioskować o przybliżonej liczbie wystąpień danego l-meru w badanej sekwencji DNA. Aby móc precyzyjnie zdefiniować problemy obliczeniowe dla tej wersji sekwencjonowania przez hybrydyzację konieczne jest wprowadzenie dodatkowych typów spektrum [28, 29].

Niech $S(Q)$ nadal reprezentuje spektrum sekwencji Q, a $S^{(m)}(Q)$ reprezentuje multispektrum sekwencji Q. Oba zbiory $S(Q)$ oraz $S^{(m)}(Q)$ zawierają słowa zbudowane nad alfabetem {A, C, G, T}. Spektrum jest zbiorem, więc każde słowo s_i reprezentujące pewien oligonukleotyd może występować w $S(Q)$ co najwyżej raz. Multispektrum jest multizbiorem, więc poszczególne słowa mogą występować w $S^{(m)}(Q)$ wielokrotnie. Liczba wystąpień danego słowa s_i w $S^{(m)}(Q)$ jest oznaczana przez m_i. Niech $S^{(is)}(Q)$ reprezentuje idealne spektrum sekwencji Q. Zbiór ten zawiera wszystkie i tylko te słowa, które reprezentują oligonukleotydy będące fragmentami analizowanej sekwencji. Co więcej, zbiór $S^{(is)}(Q)$ zawiera tylko jedno wystąpienie danego słowa. Wszystkie wystąpienia poszczególnych oligonukleotydów zawiera idealne multispektrum sekwencji Q, które jest reprezentowane przez zbiór $S^{(im)}(Q)$. Należy zwrócić uwagę, że liczba wystąpień dowolnego słowa w zbiorze $S^{(im)}(Q)$ odpowiada liczbie powtórzeń odpowiadającego mu oligonukleotydu w sekwencji Q. Na zawartość multispektrum mogły wpłynąć błędy hybrydyzacji, więc wartość m_i dla danego słowa może się różnić od liczby rzeczywistych wystąpień w sekwencji Q odpowiadającego mu oligonukleotydu. Będzie ona mniejsza w przypadku błędów negatywnych lub większa w przypadku błędów pozytywnych.

W zależności od precyzji dostępnej informacji o wielokrotności l-merów problem obliczeniowy sekwencjonowania przez hybrydyzację z błędami negatywnymi i pozytywnymi można zdefiniować następująco. Wszystkie trzy poniższe problemy należą do klasy problemów silnie NP-trudnych [28, 29].

Problem 5.1.3 - SBH z błędami dowolnego typu oraz informacją o powtórzeniach typu "jeden i wiele" [28, 29]
INSTANCJA: zbiór $S(Q)$ zawierający słowa o długości l zbudowane nad alfabetem {A, C, G, T}, długość n sekwencji Q, parametr $m_i \in \{1, 2\}$ dla każdego słowa $s_i \in S(Q)$.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n zawierająca maksymalną liczbę elementów zbioru $S(Q)$, przy czym Q' zawiera co najwyżej jedno wystąpienie s_i, gdy $m_i = 1$ oraz co najmniej jedno, jeżeli $m_i = 2$; sekwencja Q' może zawierać również pewne słowa o długości l nie będące częścią $S(Q)$.

Problem 5.1.4 - SBH z błędami dowolnego typu oraz informacją o powtórzeniach typu "jeden, dwa i wiele" [28, 29]
INSTANCJA: zbiór $S(Q)$ zawierający słowa o długości l zbudowane nad alfabetem {A, C, G, T}, długość n sekwencji Q, parametr $m_i \in \{1, 2, 3\}$ dla każdego słowa $s_i \in S(Q)$.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n zawierająca maksymalną liczbę elementów zbioru $S(Q)$, przy czym Q' zawiera co najwyżej jedno wystąpienie s_i, gdy $m_i = 1$, jedno lub dwa wystąpienia, jeżeli $m_i = 2$ oraz co najmniej dwa wystąpienia, jeżeli $m_i = 3$; sekwencja Q' może dodatkowo zawierać pewne słowa o długości l nie będące częścią $S(Q)$.

Problem 5.1.5 - SBH z błędami dowolnego typu oraz dokładną informacją o powtórzeniach [28, 29]
INSTANCJA: zbiór $S(Q)$ zawierający słowa o długości l zbudowane nad alfabetem {A, C, G, T}, długość n sekwencji Q, parametr $m_i \in \{1, 2, \ldots, n - l + 1\}$ dla każdego słowa $s_i \in S(Q)$.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n zawierająca maksymalną liczbę elementów zbioru $S(Q)$, przy czym każde słowo $s_i \in S(Q)$ występuje w sekwencji Q' dokładnie m_i lub $m_i - 1$ razy; sekwencja Q' może dodatkowo zawierać pewne słowa o długości l nie będące częścią $S(Q)$.

W przypadku izotermicznego SBH (ISBH), które zostało szczegółowo opisane w Rozdziale 3.3.5, wykorzystywane są biblioteki zawierające oligonukleotydy o różnej długości ale tej samej temperaturze topnienia T_m. Umożliwia to lepsze dobranie warunków eksperymentu biochemicznego i redukuje liczbę błędów hybrydyzacji. Wykazano, że do przeprowadzenia sekwencjonowania konieczne i wystarczające jest wykorzystanie dwóch takich bibliotek oligonukleotydów o temperaturach topnienia T_L oraz $T_L + 2$℃ [8, 28]. W efekcie spektrum sekwencji składa się z oligonukleotydów o tych dwóch temperaturach topnienia.

W związku z wykorzystaniem dwóch bibliotek o różnych temperaturach mogą się zdarzyć takie przypadki, w których jeden z oligonukleotydów ze spektrum będzie się całkowicie zawierał w innym. Przykładowo oligonukleotyd AAG o $T_m = 8$ zawiera się w oligonukleotydzie AAGT o $T_m = 10$. Zastosowanie przy tym takiej samej funkcji celu jak w klasycznym SBH mogłoby prowadzić do większej liczby błędów negatywnych, tj. fragmentów w zrekonstruowanej sekwencji o temperaturze topnienia T_L lub $T_L + 2$℃, które nie były obecne w spektrum. W związku z tym zaproponowano nową funkcję celu minimalizującą łączną liczbę błędów pozytywnych i negatywnych [8].

Niech α będzie liczbą oligonukleotydów ze spektrum (słów ze zbioru $S(Q)$) będących częścią zrekonstruowanej sekwencji Q', β będzie liczbą oligonukleotydów o temperaturze topnienia równej T_L lub $T_L + 2$℃, które można wyróżnić w sekwencji Q', a $|S(Q)|$ oznacza liczbę oligonukleotydów wchodzących w skład spektrum (słów zawartych w zbiorze $S(Q)$). Te trzy wartości umożliwiają określenie liczby błędów dla zrekonstruowanej sekwencji Q'. Liczba błędów negatywnych wynosi $\beta - \alpha$, a liczba błędów pozytywnych jest równa $|S(Q)| - \alpha$. Łączna liczba błędów wynosi $\beta + |S(Q)| - 2\alpha$ i w przypadku izotermicznego sekwencjonowania przez hybrydyzację celem jest minimalizacja właśnie tej wartości. Poniżej przedstawiono formalną definicję problemu ISBH.

Problem 5.1.6 - izotermiczne SBH [8]
INSTANCJA: zbiór $S(Q)$ zawierający słowa zbudowane nad alfabetem {A, C, G, T} reprezentujące oligonukleotydy o temperaturach topnienia T_L oraz $T_L + 2$℃, długość n sekwencji Q.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n oraz o minimalnej wartości $\beta + |S(Q)| - 2\alpha$.

Możliwe jest również jednoczesne zastosowanie obu omawianych powyżej modyfikacji klasycznego sekwencjonowania przez hybrydyzację. Można pozyskać częściową informację o powtórzeniach stosując w trakcie eksperymentu biblioteki izotermiczne. Poniżej zaprezentowano formalne definicje problemów izotermicznego sekwencjonowania DNA przez hybrydyzację z błędami dowolnego rodzaju oraz częściową informacją o powtórzeniach typu "jeden i wiele" oraz "jeden, dwa i wiele". Oba poniższe problemy należą do klasy problemów silnie NP-trudnych [30].

Problem 5.1.7 - izotermiczne SBH z błędami dowolnego typu oraz informacją o powtórzeniach typu "jeden i wiele" [30]
INSTANCJA: zbiór $S(Q)$ zawierający słowa zbudowane nad alfabetem {A, C, G, T} reprezentujące oligonukleotydy o temperaturach topnienia T_L oraz $T_L + 2$°C, długość n sekwencji Q, parametr $m_i \in \{1, 2\}$ dla każdego słowa $s_i \in S(Q)$.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n oraz o minimalnej wartości $\beta + |S(Q)| - 2\alpha$, przy czym Q' zawiera co najwyżej jedno wystąpienie s_i, gdy $m_i = 1$ oraz co najmniej jedno, jeżeli $m_i = 2$; sekwencja Q' może zawierać również pewne słowa reprezentujące oligonukleotydy o temperaturze topnienia T_L lub $T_L + 2$°C nie będące częścią $S(Q)$.

Problem 5.1.8 - izotermiczne SBH z błędami dowolnego typu oraz informacją o powtórzeniach typu "jeden, dwa i wiele" [30]
INSTANCJA: zbiór $S(Q)$ zawierający słowa zbudowane nad alfabetem {A, C, G, T} reprezentujące oligonukleotydy o temperaturach topnienia T_L oraz $T_L + 2$°C, długość n sekwencji Q, parametr $m_i \in \{1, 2, 3\}$ dla każdego słowa $s_i \in S(Q)$.
ODPOWIEDŹ: sekwencja Q' o długości nie większej niż n oraz o minimalnej wartości $\beta + |S(Q)| - 2\alpha$, przy czym Q' zawiera co najwyżej jedno wystąpienie s_i, gdy $m_i = 1$, jedno lub dwa wystąpienia, jeżeli $m_i = 2$ oraz co najmniej dwa wystąpienia, jeżeli $m_i = 3$; sekwencja Q' może dodatkowo zawierać pewne słowa reprezentujące oligonukleotydy o temperaturze topnienia T_L lub $T_L + 2$°C nie będące częścią $S(Q)$.

Możliwe jest rozszerzenie instancji każdego z powyższych problemów sekwencjonowania przez hybrydyzację o opcjonalną informację o tym, który z oligonukleotydów stanowi początek rekonstruowanej sekwencji. Zazwyczaj do przygotowania wielu kopii analizowanej sekwencji wykorzystuje się łańcuchową reakcję polimerazy (ang. *Polymerase Chain Reaction*, PCR), która wymaga jego znajomości. Stąd korzystanie z tej dodatkowej informacji ma praktyczne uzasadnienie.

5.2 Przegląd istniejących modeli grafowych dla SBH

5.2.1 Podejście klasyczne bez błędów

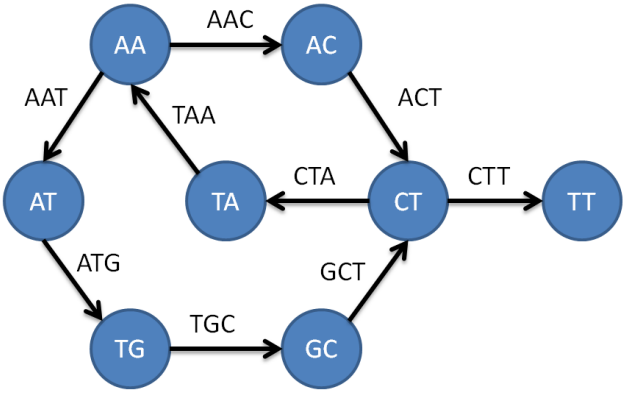
Problem obliczeniowy dla klasycznego SBH bez błędów (Problem 5.1.1) może zostać sprowadzony do problemu skierowanej drogi Eulera (Def. 4.3.1) [52]. Instancja problemu grafowego jest konstruowana w następujący sposób.

Tworzony jest graf skierowany $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada łuk $a_i \in A$. Łuk ma swój początek w wierzchołku etykietowanym pierwszymi $l-1$ symbolami s_i a jego koniec jest w wierzchołku etykietowanym ostatnimi $l-1$ symbolami s_i. Znalezienie drogi Eulera w grafie G jest równoważne znalezieniu kolejności l-merów w sekwencji Q'. Jeżeli instancja problemu precyzuje, który z l-merów stanowi początek sekwencji Q, to wymagane jest, aby droga Eulera rozpoczynała się od krawędzi reprezentującej dany oligonukleotyd.

Przykład 5.2.1. Niech nieznaną, rekonstruowaną sekwencją będzie AACTAATGCTT o długości $n = 11$, a wykorzystana biblioteka oligonukleotydów zawiera wszystkie l-mery o długości $l = 3$. W przypadku braku jakichkolwiek błędów wynikiem eksperymentu biochemicznego będzie idealne spektrum, a zbiór $S(Q)$ zawierać będzie następujące słowa: AAC, AAT, ACT, ATG, CTA, CTT, GCT, TAA i TGC.

Graf G skonstruowany zgodnie z opisanymi powyżej zasadami przedstawia Rysunek 5.1. Istnieją w nim dwie następujące drogi Eulera (przedstawione w postaci sekwencji odwiedzanych wierzchołków).
AA → AC → CT → TA → AA → AT → TG → GC → CT → TT
AA → AT → TG → GC → CT → TA → AA → AC → CT → TT

Każda droga Eulera w grafie G odpowiada jednemu możliwemu rozwiązaniu dla problemu SBH, tj. sekwencji o długości n zawierającej wszystkie elementy ze spektrum. Dwóm powyższym drogom odpowiadają odpowiednio sekwencje: AACTAATGCTT oraz AATGCTAACT. Wykorzystanie informacji o pierwszym l-merze analizowanej sekwencji może umożliwić zredukowanie liczby dopuszczalnych rozwiązań. W podanym przykładzie jedynie pierwsze rozwiązanie spełnia to ograniczenie.



Rysunek 5.1: Graf skierowany odpowiadający instancji SBH bez błędów przedstawionej w Przykładzie 5.2.1.

5.2.2 Podejście klasyczne z błędami dowolnego typu

Problem obliczeniowy dla klasycznego sekwencjonowania przez hybrydyzację z błędami dowolnego typu może zostać zamodelowany za pomocą problemu Orienteering w grafie skierowanym (Def. 4.3.9). Poniższy model został zaprezentowany w [11], przy czym warto zwrócić uwagę na inną nazwę problemu grafowego we wspomnianej pracy. Autorzy wykorzystują pewien wariant problemu selektywnego komiwojażera. Niezależnie powstało wiele prac skupiających swą uwagę konkretnie na tym wariancie, w których nazywany był on terminem Orienteering [18, 23, 24, 35].

Tworzony jest ważony, graf skierowany $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada wierzchołek $v_i \in V$. Każdy z wierzchołków ma taką samą wartość równą 1, a budżet $B = n - l$, gdzie n jest długością sekwencji Q, a l jest długością słów należących do zbioru $S(Q)$. Koszt (waga) c_{ij} łuku z wierzchołka v_i do v_j zależy od tego, jak bardzo słowa s_i i s_j nakładają się na siebie. Niech o_{ij} oznacza liczbę końcowych symboli (długość sufiksu) słowa s_i, które są jednocześnie początkowymi symbolami (prefiksem) słowa s_j, wtedy $c_{ij} = l - o_{ij}$. Zakłada się przy tym, że $s_i \neq s_j$, co przy równej długości wszystkich oligonukleotydów prowadzi do $0 \leqslant o_{ij} \leqslant l - 1$ oraz $1 \leqslant c_{ij} \leqslant l$.

Przykładowo koszt łuku z wierzchołka reprezentującego słowo CGCTTA do wierzchołka reprezentującego słowo GCTTAT wynosi 1, bo mają wspólną sekwencję GCTTA o długości równej 5. Warto również zauważyć, że dla dwóch słów może istnieć więcej niż

jedno możliwe nałożenie. W takim przypadku pomiędzy dwoma wierzchołkami odpowiadającymi tym słowom utworzonych zostanie kilka łuków, tj. po jednym dla każdego możliwego nałożenia słów. Graf G jest więc multigrafem.

Rozwiązanie problemu Orienteering w grafie G, tj. znalezienie ścieżki prostej o maksymalnej wartości i koszcie nie przekraczającym B, jest równoważne znalezieniu kolejności l-merów w sekwencji Q'. Jeżeli instancja problemu precyzuje, który z l-merów stanowi początek sekwencji Q, to dodatkowo wymagane jest, aby ścieżka ta rozpoczynała się od wierzchołka reprezentującego dany oligonukleotyd. Co więcej, dzięki relacji pomiędzy budżetem B a długością sekwencji n sekwencja Q' nigdy nie będzie dłuższa niż n, a maksymalizacja sumarycznej wartości odwiedzonych wierzchołków gwarantuje maksymalne wykorzystanie elementów ze zbioru $S(Q)$.

Przykład 5.2.2. Niech nieznaną, rekonstruowaną sekwencją będzie AACTAACGC o długości $n = 9$, a wykorzystana biblioteka oligonukleotydów zawiera wszystkie l-mery o długości $l = 3$. Fragment AAC występuje w sekwencji dwukrotnie, ale spektrum będzie zwierać tylko jedno wystąpienie tego l-meru (błąd negatywny wynikający z powtórzenia). Dodatkowo niech w spektrum brakuje oligonukleotydu ACT (negatywny błąd hybrydyzacji) oraz niech występuje w nim oligonukleotyd CGA nie będący fragmentem analizowanej sekwencji (pozytywny błąd hybrydyzacji). Zbiór $S(Q)$ zawierać więc będzie następujące słowa: AAC, ACG, CGC, CGA, CTA, TAA.

Graf G skonstruowany zgodnie z opisanymi powyżej zasadami przedstawia Rysunek 5.2. Istnieją w nim dwie następujące ścieżki o maksymalnej możliwej wartości 6, których koszt nie przekracza budżetu $B = 6$.
CTA → TAA → AAC → ACG → CGC → CGA
CGC → CTA → TAA → AAC → ACG → CGA

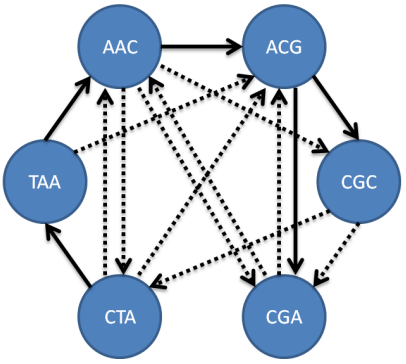
Każda ścieżka w grafie G będąca rozwiązaniem dla postawionego problemu Orienteering odpowiada jednemu możliwemu rozwiązaniu dla problemu SBH, tj. sekwencji nie dłuższej niż n zawierającej maksymalną możliwą liczbę elementów ze spektrum. Dwóm powyższym ścieżkom odpowiadają odpowiednio sekwencje: CTAACGCGA oraz CGCTAACGA. Jeżeli dodatkowo częścią instancji problemu byłaby informacja o pierwszym l-merze rekonstruowanej sekwencji, to rozwiązaniem problemu Orienteering byłyby następujące ścieżki. Odpowiadają one odpowiednio sekwencjom AACGCTAA, AACTAACGC oraz AACTAACGA.
AAC → ACG → CGC → CTA → TAA
AAC → CTA → TAA → ACG → CGC
AAC → CTA → TAA → ACG → CGA

5.2.3 Izotermiczne SBH bez błędów

Problem obliczeniowy dla izotermicznego problemu sekwencjonowania przez hybrydyzację przy założeniu braku jakichkolwiek błędów może zostać zamodelowany za pomocą problemu skierowanej drogi Eulera. Poniżej przedstawiono jedynie ogólny opis. Szczegóły są dostępne w [37].

Tworzony jest skierowany graf $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada wierzchołek $v_i \in V$. Łuki pomiędzy wierzchołkami są tworzone zgodnie z następującymi zasadami:

- jeżeli słowo s_i jest prefiksem s_j, to dodawany jest łuk z v_i do v_j i zabronione jest tworzenie jakichkolwiek innych łuków wychodzących z v_i i wchodzących do v_j,



RYSUNEK 5.2: Graf skierowany odpowiadający instancji SBH z błędami przedstawionej w Przykładzie 5.2.2. Linią ciągłą oznaczono łuki o koszcie równym 1. Linią przerywaną oznaczono łuki o koszcie równym 2. Dla czytelności pominięto łuki o koszcie równym 3.

- jeżeli słowo s_i jest sufiksem s_j, to dodawany jest łuk z v_j do v_i i zabronione jest tworzenie jakichkolwiek innych łuków wychodzących z v_j i wchodzących do v_i,
- jeżeli słowa s_i i s_j mają równą długość i nakładają się na siebie z przesunięciem jednego symbolu (s_i jest pierwsze), to tworzony jest łuk z v_i do v_j, przy założeniu że nie doprowadzi to do występowania w odtwarzanej sekwencji Q' fragmentu o temperaturze topnienia T_L lub T_L+2℃, który nie jest reprezentowany przez żadne ze słów w zbiorze $S(Q)$.

Zbiór łuków utworzony wg. powyższych wymagań jest następnie modyfikowany. Zakłada się, że znany jest pierwszy i ostatni oligonukleotyd rekonstruowanej sekwencji. Na podstawie tej informacji (możliwej do pozyskania w ramach dodatkowych eksperymentów biochemicznych) usuwane są wszelkie łuki wchodzące do wierzchołka reprezentującego pierwszy oligonukleotyd oraz wszelkie łuki wychodzące z wierzchołka reprezentującego ostatni oligonukleotyd. Dodatkowo usuwane są pewne łuki (szczegóły w [37]) celem zapewnienia, aby graf G był skierowanym grafem liniowym dla pewnego innego grafu.

Skierowany liniowy graf G jest następnie transformowany do oryginalnego grafu H, w którym poszczególne słowa ze zbioru $S(Q)$ (oligonukleotydy) są reprezentowane przez łuki. Ostatecznie w grafie H szuka się drogi Eulera. Kolejność łuków wyznaczana przez znalezioną w H drogę Eulera odpowiada kolejności oligonukleotydów w konstruowanej sekwencji Q'.

Rozdział 6

Modele grafowe dla problemów sekwencjonowania DNA przez hybrydyzację z dodatkową informacją o powtórzeniach

6.1 SBH z klasycznymi bibliotekami oligonukleotydów

6.1.1 Dodatkowe wierzchołki w grafie

Modelując problem sekwencjonowania DNA przez hybrydyzację z dodatkową informacją o powtórzeniach za pomocą problemu grafowego należy uwzględnić wielokrotne występowanie oligonukleotydów w multispektrum. Zakładając możliwość pozyskania dokładnej informacji o powtórzeniach (Problem 5.1.5) najprostszym podejściem jest utworzenie m_i wierzchołków w grafie dla każdego słowa s_i ze zbioru $S(Q)$. W takiej sytuacji problem można zamodelować za pomocą pewnego wariantu problemu Orienteering, w którym dodatkowo wymagane jest odwiedzenie przynajmniej $m_i - 1$ wierzchołków reprezentujących dane słowo s_i.

Tworzony jest ważony graf skierowany $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada zbiór V_i zawierający m_i wierzchołków $v_{i_1}, v_{i_2}, \ldots, v_{i_{m_i}}$ ze zbioru V. Każdy z wierzchołków ma taką samą wartość równą 1 ($\forall_{v \in V} \psi(v) = 1$), a budżet $B = n - l$, gdzie n jest długością sekwencji Q, a l jest długością słów należących do zbioru $S(Q)$. Koszt (waga) $c_{i_p j_q}$ łuku z wierzchołka v_{i_p} do v_{j_q} ($1 \leqslant p \leqslant m_i, 1 \leqslant q \leqslant m_j$) zależy od tego, jak bardzo słowa s_i i s_j nakładają się na siebie. Niech o_{ij} oznacza liczbę końcowych symboli (długość sufiksu) słowa s_i, które są jednocześnie początkowymi symbolami (prefiksem) słowa s_j, wtedy $c_{i_p j_q} = l - o_{ij}$. Łuki rozpoczynające i kończące się w tym samym wierzchołku są zabronione, ale dopuszcza się możliwość występowania łuków pomiędzy dwoma różnymi wierzchołkami reprezentującymi to samo słowo s_i pod warunkiem, że koszt takiego łuku jest większy od zera (nie są tworzone łuki dla $o_{ij} = l$, tj. pełnego nałożenia). W konsekwencji $1 \leqslant c_{ij} \leqslant l$.

W powyższym grafie G poszukiwana jest ścieżka P o koszcie nie większym niż B, maksymalnej wartości zwartych wierzchołków Ψ, która dodatkowo zawiera przynajmniej $m_i - 1$ wierzchołków dla każdego zbioru V_i. Niech $P = \{v_{i_1}, v_{i_2}, \cdots, v_{i_\Psi}\}$ oznacza taką

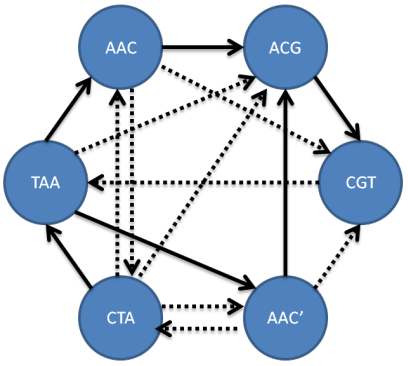
ścieżkę. W związku z tym, że każdy z wierzchołków ma taką samą wartość równą 1, to liczba wierzchołków wchodzących w skład P jest równa Ψ. Sekwencja wierzchołków określona przez P definiuje kolejne słowa, tj. $\{s_{i_1}, s_{i_2}, \cdots, s_{i_\Psi}\}$, składające się na sekwencję Q'.

Uwaga 6.1. Jeżeli instancja problemu precyzuje, który z l-merów stanowi początek sekwencji Q, to wymagane jest również, aby ścieżka rozpoczynała się od wierzchołka reprezentującego dany oligonukleotyd.

Przykład 6.1.1. Niech nieznaną, rekonstruowaną sekwencją będzie AACTAACG o długości $n = 8$, a wykorzystana biblioteka oligonukleotydów zawiera wszystkie l-mery o długości $l = 3$. Niech w spektrum brakuje oligonukleotydu ACT (negatywny błąd hybrydyzacji) oraz niech występuje w nim oligonukleotyd CGT nie będący fragmentem analizowanej sekwencji (pozytywny błąd hybrydyzacji). Zbiór $S(Q)$ zawierać więc będzie następujące słowa: AAC, ACG, CGG, CTA, TAA. Przyjmując dokładny model informacji o powtórzeniach parametr m_i dla słowa AAC będzie równy 2, a dla pozostałych słów będzie równy 1.

Graf G skonstruowany zgodnie z opisanymi powyżej zasadami przedstawia Rysunek 6.1. Dla słowa AAC utworzono dwa wierzchołki o etykietach AAC i AAC'. Poniżej przedstawiono przykładowe ścieżki będące rozwiązaniem dla zdefiniowanego problemu grafowego. Wartość ścieżek Ψ wynosi 5, a ich koszt nie przekracza budżetu $B = 5$.
AAC → CTA → TAA → AAC' → ACG
AAC → ACG → CGT → TAA → AAC'
CTA → TAA → AAC → ACG → CGT

Każda ścieżka w grafie G będąca rozwiązaniem dla postawionego problemu grafowego odpowiada jednemu możliwemu rozwiązaniu dla problemu SBH. Trzem powyższym ścieżkom odpowiadają odpowiednio sekwencje: AACTAACG, AACGTAAC oraz CTAACGT.



Rysunek 6.1: Graf skierowany dla Przykładu 6.1.1 - instancja SBH z błędami dowolnego typu oraz dokładną informacją o powtórzeniach. Powtarzający się oligonukleotyd zamodelowano za pomocą dwóch wierzchołków AAC i AAC'. Linią ciągłą oznaczono łuki o koszcie równym 1. Linią przerywaną oznaczono łuki o koszcie równym 2. Dla czytelności pominięto łuki o koszcie równym 3.

Podstawową zaletą powyższego modelu grafowego jest jego podobieństwo do modelu wykorzystywanego dla klasycznego sekwencjonowania przez hybrydyzację. Konieczne jest rozwiązanie wariantu problemu Orienteering, w którym rozwiązania dopuszczalne

muszą spełniać jedynie pewne dodatkowe wymaganie. Jednak modelowanie wielokrotnych oligonukleotydów za pomocą kolejnych wierzchołków powoduje wzrost rozmiaru grafu. Przy dodaniu nowego wierzchołka do zbioru wierzchołków o aktualnej liczności $|V|$ konieczne jest utworzenie $2(|V|-1)$ nowych łuków w zbiorze A.

Należy również zauważyć, że powyższy model jednoznacznie określa maksymalną liczbę wystąpień poszczególnych l-merów ze spektrum w rekonstruowanej sekwencji. Dla danego słowa s_i tworzonych jest m_i wierzchołków i rekonstruowana sekwencja nigdy nie będzie zawierać więcej niż m_i powtórzeń danego l-meru. W przypadkach częściowej informacji o powtórzeniach (Problem 5.1.3 oraz Problem 5.1.4) pewne oligonukleotydy mogą wystąpić dowolną liczbę razy. Powyższy model grafowy umożliwia więc odwzorowanie jedynie problemu sekwencjonowania przez hybrydyzację z pełną informacją o powtórzeniach (Problem 5.1.5), który ma obecnie wartość wyłącznie teoretyczną (szczegóły w Rozdziale 3.3.6).

6.1.2 Dodatkowe etykiety wierzchołków

Do wiernego odwzorowania problemów SBH z częściową informacją o powtórzeniach zaproponowano następujące podejście. Opiera się ono na dodatkowych etykietach wierzchołków. Dla każdego wierzchołka zdefiniowana jest minimalna i maksymalna liczba jego wystąpień w konstruowanej ścieżce.

Tworzony jest ważony graf skierowany $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada wierzchołek $v_i \in V$. Każdy z wierzchołków ma taką samą wartość równą 1 ($\forall_{v \in V} \psi(v) = 1$), a budżet $B = n - l$, gdzie n jest długością sekwencji Q, a l jest długością słów należących do zbioru $S(Q)$. Koszt (waga) c_{ij} łuku z wierzchołka v_i do v_j zależy od tego, jak bardzo słowa s_i i s_j nakładają się na siebie. Niech o_{ij} oznacza liczbę końcowych symboli (długość sufiksu) słowa s_i, które są jednocześnie początkowymi symbolami (prefiksem) słowa s_j, wtedy $c_{ij} = l - o_{ij}$. Łuki rozpoczynające i kończące się w tym samym wierzchołku są dozwolone ($s_i = s_j$) pod warunkiem, że koszt takiego łuku jest większy od zera (nie są tworzone łuki dla $o_{ij} = l$, tj. pełnego nałożenia). W konsekwencji $1 \leqslant c_{ij} \leqslant l$.

Dla każdego wierzchołka $v_i \in V$ określona jest również minimalna i maksymalna liczba jego wystąpień w konstruowanej ścieżce. Precyzują to odpowiednio wartości min_i oraz max_i. Są one ustalane na podstawie parametru m_i, a konkretne wartości dla poszczególnych modeli częściowej informacji o powtórzeniach przedstawia Tabela 6.1.

W powyższym grafie G poszukiwana jest ścieżka P o maksymalnej wartości odwie-

Tabela 6.1: Model grafowy dla sekwencjonowania DNA przez hybrydyzację z częściową informacją o powtórzeniach - minimalna (min_i) i maksymalna (max_i) liczba odwiedzin danego wierzchołka v_i w zależności od typu informacji o powtórzeniach.

	"jeden i wiele"		"jeden, dwa i wiele"		
	$m_i = 1$	$m_i = 2$	$m_i = 1$	$m_i = 2$	$m_i = 3$
min_i	0	1	0	1	2
max_i	1	∞	1	2	∞

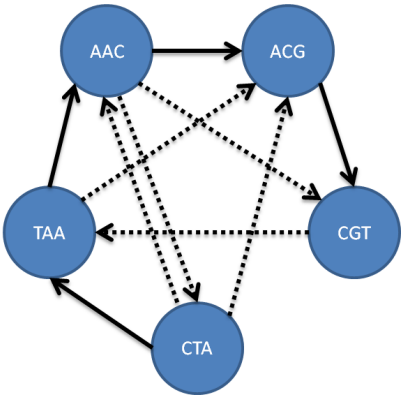
dzonych wierzchołków Ψ, której koszt nie przekracza B oraz w której każdy z wierzchołków występuje zgodnie z wartościami min_i i max_i. Dodatkowo jeżeli podany został pierwszy oligonukleotyd rekonstruowanej sekwencji, to wymagane jest, aby pierwszym wierzchołkiem ścieżki P był wierzchołek reprezentujący ten oligonukleotyd. Poszczególne

wierzchołki mogą być odwiedzone wielokrotnie (w przypadku Orienteering co najwyżej raz), a każde odwiedzenie danego wierzchołka powoduje dodane jego wartości do sumarycznej wartości ścieżki. Sekwencja wierzchołków określona przez P definiuje kolejne słowa składające się na sekwencję Q'.

Przykład 6.1.2. Niech nieznaną, rekonstruowaną sekwencją będzie AACTAACG o długości $n = 8$, a wykorzystana biblioteka oligonukleotydów zawiera wszystkie l-mery o długości $l = 3$. Niech w spektrum brakuje oligonukleotydu ACT (negatywny błąd hybrydyzacji) oraz niech występuje w nim oligonukleotyd CGT nie będący fragmentem analizowanej sekwencji (pozytywny błąd hybrydyzacji). Zbiór $S(Q)$ zawierać więc będzie następujące słowa: AAC, ACG, CGG, CTA, TAA. Przyjmując model informacji o powtórzeniach typu "jeden i wiele" parametr m_i dla słowa AAC będzie równy 2, a dla pozostałych słów będzie równy 1.

Graf G skonstruowany zgodnie z opisanymi powyżej zasadami przedstawia Rysunek 6.2. Rysunek ten prezentuje również wartości min_i oraz max_i dla poszczególnych wierzchołków. Poniżej przedstawiono przykładowe ścieżki będące rozwiązaniem dla zdefiniowanego problemu grafowego. Wartość ścieżek Ψ wynosi 5, a ich koszt nie przekracza budżetu $B = 5$. Co więcej, każda ze ścieżek odwiedza przynajmniej raz wierzchołek reprezentujący słowo AAC oraz co najwyżej raz pozostałe wierzchołki.
AAC → CTA → TAA → AAC → ACG
AAC → ACG → CGT → TAA → AAC
CTA → TAA → AAC → ACG → CGT

Każda ścieżka w grafie G będąca rozwiązaniem dla postawionego problemu grafowego odpowiada jednemu możliwemu rozwiązaniu dla problemu SBH. Trzem powyższym ścieżkom odpowiadają odpowiednio sekwencje: AACTAACG, AACGTAAC oraz CTAACGT.



	min_i	max_i
AAC	1	∞
ACG	0	1
CGT	0	1
CTA	0	1
TAA	0	1

Rysunek 6.2: Graf skierowany dla Przykładu 6.1.2 - instancja SBH z błędami dowolnego typu oraz informacją o powtórzeniach typu "jeden i wiele". Tabela obok prezentuje wartości min_i oraz max_i dla poszczególnych wierzchołków. Linią ciągłą oznaczono łuki o koszcie równym 1. Linią przerywaną oznaczono łuki o koszcie równym 2. Dla czytelności pominięto łuki o koszcie równym 3.

6.2 SBH z bibliotekami izotermicznymi

Zaproponowany model grafowy dla problemu izotermicznego SBH z błędami dowolnego typu oraz częściową informacją o powtórzeniach bazuje na modelu przedstawionym w

Rozdziale 6.1.2. Wymaga on jednak pewnych modyfikacji ze względu na różną długość oligonukleotydów w spektrum i wykorzystywanie odmiennej funkcji celu w problemach ISBH (szczegóły w Rozdziale 5.1).

Tworzony jest ważony graf skierowany $G = (V, A)$. Każdemu słowu $s_i \in S(Q)$ odpowiada wierzchołek $v_i \in V$. Każdy wierzchołek v_i ma taką samą wartość równą 1 ($\forall_{v \in V} \psi(v) = 1$) oraz ma określone wartości min_i i max_i. Wartości te zależą od parametru m_i słowa s_i oraz wykorzystywanego modelu częściowej informacji o powtórzeniach. Są one wyznaczane analogicznie jak w przypadku klasycznych bibliotek oligonukleotydów (szczegóły w Tabeli 6.1).

Każdy wierzchołek ma również przypisany koszt $c_{v_i} = |s_i|$, gdzie $|s_i|$ jest długością słowa s_i. Budżet $B = n - c_{v_p}$, gdzie n jest długością sekwencji Q, a v_p jest pierwszym wierzchołkiem skonstruowanej ścieżki. Koszt (waga) c_{ij} łuku z wierzchołka v_i do v_j zależy od tego, jak bardzo słowa s_i i s_j nakładają się na siebie. Niech o_{ij} oznacza liczbę końcowych symboli (długość sufiksu) słowa s_i, które są jednocześnie początkowymi symbolami (prefiksem) słowa s_j, wtedy $c_{ij} = |s_j| - o_{ij}$. Łuki rozpoczynające i kończące się w tym samym wierzchołku są dozwolone ($s_i = s_j$) pod warunkiem, że koszt takiego łuku jest większy od zera (nie są tworzone łuki dla $o_{ij} = |s_i| = |s_j|$, tj. pełnego nałożenia).

Należy zauważyć, że w przypadku zastosowania izotermicznych bibliotek oligonukleotydów możliwe są sytuacje, gdy słowo s_j będzie sufiksem słowa s_i, tj. $o_{ij} = |s_j|$. Wtedy koszt łuku c_{ij} (zakładając $s_i \neq s_j$) będzie równy 0 i jest to minimalny możliwy koszt łuku w grafie G. Maksymalny koszt łuku w tworzonym grafie G zależy od najdłuższego słowa w zbiorze $S(Q)$. Jego długość można wyznaczyć na podstawie temperatur topnienia stosowanych bibliotek. Zakładając wykorzystanie dwóch bibliotek o temperaturach topnienia T_L i $T_L + 2$℃ najdłuższe możliwe słowo będzie miało długość $(T_L + 2)/2$.

Oprócz kosztu c_{ij} każdy łuk ma przypisaną jeszcze wartość β_{ij}, która jest równa liczbie oligonukleotydów o temperaturze topnienia T_L lub $T_L + 2$℃, które można wyróżnić w słowie powstałym z nałożenia s_i i s_j.

W powyższym grafie G poszukiwana jest ścieżka P o minimalnej wartości $\beta + |S(Q)| - 2\alpha$, której koszt nie przekracza B oraz w której każdy z wierzchołków występuje zgodnie z wartościami min_i i max_i, gdzie α jest sumaryczną wartością odwiedzonych wierzchołków, β jest sumą wartości β_{ij} łuków pomiędzy kolejnymi wierzchołkami ścieżki, a $|S(Q)|$ jest liczbą słów wchodzących w skład zbioru $S(Q)$. Ponadto, jeżeli określono pierwszy oligonukleotyd analizowanej sekwencji, to odpowiadający mu wierzchołek musi być pierwszym wierzchołkiem ścieżki P.

6.3 Modelowanie problemu SBH za pomocą modelu grafowego dla izotermicznego SBH

Problemy sekwencjonowania DNA przez hybrydyzację z błędami dowolnego typu oraz częściową informacją o powtórzeniach zostały zdefiniowane w Rozdziale 5.1. W Rozdziale 6.1 przedstawiono dla nich dwa modele grafowe. W niniejszym rozdziale zostanie zaprezentowana możliwość zastosowania dla nich również takiego samego modelu grafowego jak dla izotermicznemu SBH z błędami dowolnego typu oraz częściową informacją o powtórzeniach, który został przedstawiony w Rozdziale 6.2.

W przypadku ISBH funkcja celu określona jest jako minimalizacja $\beta + |S(Q)| - 2\alpha$, a dla problemu z klasycznymi bibliotekami celem jest maksymalizacja α. Jednak w przypadku wykorzystania bibliotek klasycznych $\beta = n - l + 1$ i maksymalizacja α jest tożsama z minimalizacją $\beta + |S(Q)| - 2\alpha$.

Stosując model grafowy opisany Rozdziale 6.2 dla problemów zdefiniowanych dla bibliotek klasycznych warto zwrócić uwagę na następujące konsekwencje. Równa długość wszystkich l-merów prowadzi do: $B = n - l$, $c_{ij} = l - o_{ij}, 1 \leqslant c_{ij} \leqslant l - 1$. Dodatkowo dla każdego łuku ustawiane są wartości $\beta_{ij} = c_{ij} + 1$. Wartości min_i oraz max_i wyznaczane są zgodnie z danymi w Tabeli 6.1, a wartość każdego wierzchołka jest równa 1 ($\forall_{v \in V} \psi(v) = 1$).

W otrzymanym grafie G poszukiwana jest ścieżka o minimalnej wartości $\beta + |S(Q)| - 2\alpha$, której koszt nie przekracza B oraz w której każdy z wierzchołków występuje zgodnie z wartościami min_i i max_i. W związku z tym, że przy wykorzystaniu standardowych bibliotek minimalizacja $\beta + |S(Q)| - 2\alpha$ jest jednoznaczna z maksymalizacją liczby elementów ze zbioru $S(Q)$, to znalezienie ścieżki spełniającej opisane powyżej wymagania jest równoznaczne z określeniem kolejności l-merów w sekwencji Q'.

Rozdział 7

Algorytm aproksymacyjny dla klasycznego problemu sekwencjonowania DNA przez hybrydyzację

7.1 Proste przekształcenie problemu TSP w grafie skierowanym do problemu w grafie nieskierowanym

Zaproponowany w dalszej części algorytm dla problemu komiwojażera w grafie skierowanym wykorzystuje algorytm dla TSP w grafie nieskierowanym. Na podstawie danego grafu skierowanego konstruowany jest graf nieskierowany, rozwiązywany jest problem w grafie nieskierowanym, a otrzymane rozwiązanie jest następnie wykorzystywane do utworzenia rozwiązania dla oryginalnego problemu w grafie skierowanym. Jedno z możliwych przekształceń opisano w Rozdziale 4.4. Poniżej przedstawiono alternatywne podejście.

Niech $G = (V, A)$ będzie danym skierowanym grafem, a D oznacza jego macierz odległości spełniającą nierówność trójkąta. Nieskierowany graf $\widetilde{G} = (V, E)$ tworzony jest na podstawie G następująco. Dowolne dwa łuki (v_i, v_j) i (v_j, v_i) należące do zbioru A $(v_i, v_j \in V)$ o wadze (koszcie) odpowiednio c_{ij} oraz c_{ji} są zastępowane pojedynczą krawędzią $\{v_i, v_j\}$ o koszcie równym $max(c_{ij}, c_{ji})$. Macierz odległości $\widetilde{D}$ dla grafu $\widetilde{G}$ nadal spełnia nierówność trójkąta i może zostać zdefiniowana następująco:

$$\widetilde{D}_{ij} = \widetilde{D}_{ji} = max(D_{ij}, D_{ji}) \tag{7.1}$$

Następnie rozwiązywany jest problem komiwojażera dla grafu $\widetilde{G}$. Koszt optymalnego rozwiązania $\widetilde{C}^*_H$ dla grafu $\widetilde{G}$ jest w najgorszym przypadku większy od kosztu optymalnego rozwiązania dla grafu G o $\frac{d_{max}}{d_{min}}$ razy, gdzie d_{min} i d_{max} są odpowiednio minimalną i maksymalną wartością w macierzy odległości D oraz $-\infty < d_{min} \leqslant d_{max} < \infty$. Co więcej, odwiedzanie wierzchołków w oryginalnym grafie G zgodnie z kolejnością wyznaczaną przez $\widetilde{C}^*_H$ nie wiąże się z żadnym dodatkowym kosztem.

Niech α oznacza gwarancję dokładności dla pewnego algorytmu aproksymacyjnego A rozwiązującego problem komiwojażera w grafie nieskierowanym. Algorytm aproksymacyjny wykorzystujący opisane powyżej przekształcenie oraz algorytm A będzie miał gwarancję dokładności równą $\alpha \cdot \frac{d_{max}}{d_{min}}$.

7.2 Algorytm dla problemu minimalnej s-t k-ścieżki

Problem znalezienia minimalnej s-t k-ścieżki zdefiniowano w Rozdziale 4.3.3. Do jego rozwiązania zaproponowano algorytm wykorzystujący jako podprocedurę algorytm Chaudhuri do konstrukcji drzewa rozpinającego zawierającego przynajmniej k wierzchołków, w tym wyróżnione wierzchołki s i t [22]. Koszt takiego drzewa w najgorszym przypadku jest $(1+\delta)$ razy większy w porównaniu do kosztu najkrótszej s-t k-ścieżki, gdzie $\delta > 0$.

7.2.1 Zastosowanie prostego przekształcenia

Algorytm 1 Algorytm aproksymacyjny dla problemu minimalnej s-t k-ścieżki w grafie skierowanym wykorzystujący proste przekształcenie grafu opisane w Rozdziale 7.1

Instancja: macierz odległości D spełniająca asymetryczną nierówność trójkąta o wymiarach $n \times n$ dla grafu skierowanego $G = (V, A)$, taka że $n = |V|$, $-\infty < d_{min} \leqslant d_{max} < \infty$; początkowy wierzchołek $s = v_s$; końcowy wierzchołek $t = v_t$; minimalna liczba wierzchołków do odwiedzenia k.

1: Utwórz graf nieskierowany $\widetilde{G}$ o macierzy odległości $\widetilde{D}$ zgodnej z równaniem (7.1).
2: Znajdź w $\widetilde{G}$ k-drzewo rozpinające $\widetilde{T}_k$ zawierające wierzchołki v_s i v_t.
3: Skopiuj (duplikuj) wszystkie krawędzie w utworzonym drzewie $\widetilde{T}_k$ z wyjątkiem tych należących do ścieżki z v_s do v_t.
4: Znajdź drogę Eulera $\widetilde{E}_k$ w grafie powstałym w wyniku duplikacji krawędzi.
5: Kolejność odwiedzanych wierzchołków zdefiniowana przez drogę $\widetilde{E}_k$ odpowiada pewnej v_s-v_t k-ścieżce P_k^{st} w oryginalnym grafie skierowanym G.
6: Ścieżka P_k^{st} odwiedza pewne wierzchołki wielokrotnie. Usuń z sekwencji odwiedzanych przez nią wierzchołków każde kolejne wystąpienie odwiedzonego już wcześniej wierzchołka (przejdź bezpośrednio do kolejnego nieodwiedzonego wierzchołka).

Odpowiedź: v_s-v_t k-ścieżka P_k^{st} w grafie G.

7.2.1.1 Gwarancja dokładności

Przekształcenie asymetrycznej macierzy odległości D do symetrycznej macierzy odległości $\widetilde{D}$ w kroku 1 prowadzi w najgorszym przypadku do zwiększenia kosztu rozwiązania o $\gamma = \frac{d_{max}}{d_{min}}$ razy. Koszt k-drzewa rozpinającego $\widetilde{T}_k$ otrzymanego w kroku 2 jest maksymalnie $(1+\delta)$ razy większy od kosztu najkrótszej s-t k-ścieżki P_k^{st*} w grafie $\widetilde{G}$ (dla dowolnej wartości $\delta > 0$) [22]. W kolejnym kroku krawędzie są dublowane, co może doprowadzić do podwojenia kosztu rozwiązania. Odwiedzanie wierzchołków w grafie G zgodnie z kolejnością wyznaczoną przez drogę Eulera $\widetilde{E}_k$ nie powoduje wzrostu kosztu rozwiązania, bo przy konstruowaniu macierzy $\widetilde{D}$ jako koszt krawędzi użyto większy z kosztów łączonych łuków. Niech $c_D(P)$ oznacza koszt ścieżki P w grafie zdefiniowanym przez macierz odległości D. Uwzględnienie wszystkich powyższych czynników prowadzi do uzyskania następującej gwarancji dokładności dla Algorytmu 1.

$$c_D(P_k^{st}) \leqslant 2 \cdot (1+\delta) \cdot \gamma \cdot c_D(P_k^{st*}) = (2\gamma + \delta) c_D(P_k^{st*}), \qquad \text{gdzie } \gamma = \frac{d_{max}}{d_{min}} \tag{7.2}$$

Jeżeli wartość γ jest ograniczona pewną stałą, to prowadzi to do uzyskania dla problemu minimalnej s-t k-ściezki algorytmu aproksymacyjnego o współczynniku aproksymacji $O(1)$.

7.2.1.2 Złożoność czasowa

Algorytm 1 wykonuje sekwencyjnie każdy z korków dokładnie raz, więc złożoność czasowa algorytmu zależy od najbardziej złożonego kroku. W tym przypadku jest nim krok 2, w ramach którego tworzone jest k-drzewo za pomocą algorytmu Chaudhuri, a jego złożoność czasowa wynosi $O(n^3 \log n)$ [22].

7.2.2 Zastosowanie przekształcenia grafu Kumar-Le

Algorytm 2 dla problemu minimalnej s-t k-ścieżki w grafie skierowanym bazuje na algorytmie dla ATSP [39], w ramach którego dany graf skierowany jest przekształcany do grafu nieskierowanego. Pewne własności tego przekształcenia udowodnione w [39] w kontekście poszukiwania cyklu Hamiltona o minimalnym koszcie są również prawdziwe w kontekście poszukiwania minimalnej s-t k-ścieżki, co zostało udowodnione w dalszej części niniejszego rozdziału. Umożliwia to konstrukcję optymalnej s-t k-ścieżki w oryginalnym grafie G na podstawie optymalnej s-t $2k$-ścieżki w grafie $\overline{G}$.

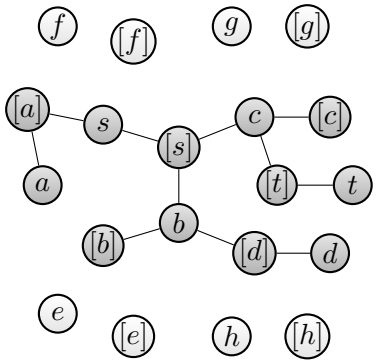
Oryginalny algorytm został zmodyfikowany w następujący sposób. Przede wszystkim zamiast tworzenia minimalnego drzewa rozpinającego jest konstruowane k-drzewo rozpinającego o minimalnym koszcie przy użyciu algorytmu Chaudhuri [22] (Rysunek 7.1 (a)). Należy przy tym zauważyć, że liczba wierzchołków ulega podwojeniu przy konstrukcji symetrycznej macierzy odległości $\overline{D}$, co prowadzi do konieczności poszukiwania minimalnego $2k$-drzewa rozpinającego w $\overline{G}$.

Zmodyfikowano również etap konstrukcji grafu Eluera. W związku z poszukiwaniem ścieżki (w oryginale poszukiwany jest cykl) konstruowany jest graf semieulerowski o dwóch wierzchołkach nieparzystego stopnia s i t.

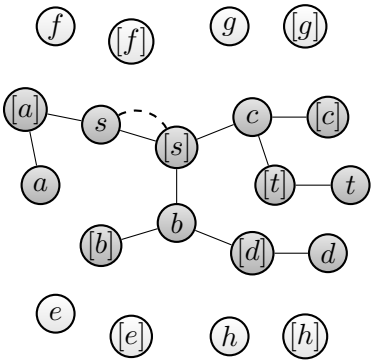
Nieparzysty stopień tych wierzchołków zapewniają kroki 3-5. W kroku 3 w razie potrzeby dodawane są krawędzie o zerowym koszcie: $\{v_s, v_{[s]}\}$ i/lub $\{v_t, v_{[t]}\}$ (Rysunek 7.1 (b)), gdzie $v_s = s, v_t = t$. Dodawanie kolejnych krawędzi o końcu w v_s lub v_t w dalszych krokach jest zabronione. Są one pomijane zarówno w kroku 4 przy dodawaniu krawędzi celem zbalansowania liczby wierzchołków rzeczywistych i wirtualnych o nieparzystym stopniu jak i w kroku 5 przy szukaniu doskonałego skojarzenia o minimalnym koszcie.

Należy zauważyć, że aby poszukiwane skojarzenie doskonałe nie zwierało krawędzi o nieskończonym koszcie (tj. krawędzi pomiędzy dwoma wierzchołkami rzeczywistymi lub dwoma wirtualnymi) konieczne jest wcześniejsze zbalansowanie liczby wierzchołków rzeczywistych i wirtualnych o nieparzystym stopniu (krok 4). W związku z wykluczeniem wierzchołków v_s i v_t z poszukiwania doskonałego skojarzenia wymagane jest spełnienie równania $|V_{odd}| = |[V]_{odd}| + 2$ (Rysunek 7.1 (c) i (d)).

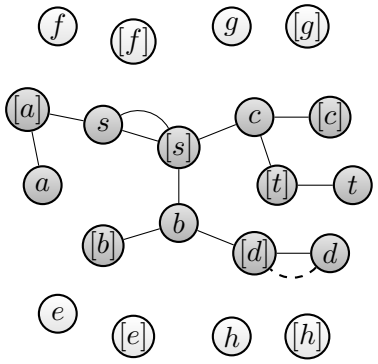
W przedostatnim kroku ze ścieżki $\overline{P}^{st}_{2k}$ usuwane są kolejne wystąpienia odwiedzonych już wcześniej wierzchołków. W trakcie tego procesu niektóre wierzchołki mogą zostać pominięte, ale ostatecznie otrzymana ścieżka zwiera zawsze przynajmniej jeden wierzchołek z każdej komplementarnej pary. Zapewnia to możliwość odtworzenia ścieżki P^{st}_k w oryginalnym grafie G (szczegóły w [39]).



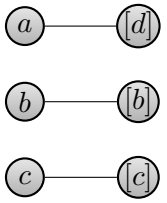
(a) $2k$-drzewo $\overline{T}_{2k}$ zawierające wierzchołki s i t



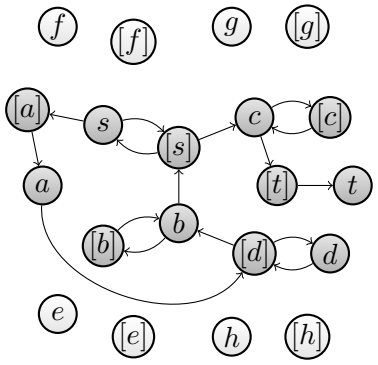
(b) Zapewnienie nieparzystego stopnia dla s



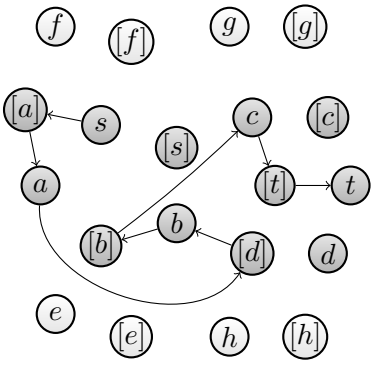
(c) Zapewnienie $|V_{odd}| = |[V]_{odd}| + 2$



(d) Skojarzenie doskonałe $\overline{M}^*$



(e) Droga Eulera $\overline{E}_{2k}$ z s do t



(f) s-t $2k$-ścieżka $\overline{P}^{st}_{2k}$

Rysunek 7.1: Konstrukcja s-t $2k$-ścieżki $\overline{P}^{st}_{2k}$

Algorytm 2 Algorytm aproksymacyjny dla problemu minimalnej s-t k-ścieżki w grafie skierowanym wykorzystujący przekształcenie grafu Kumar-Le opisane w Rozdziale 4.4

Instancja: macierz odległości D spełniająca asymetryczną nierówność trójkąta o wymiarach $n \times n$ dla grafu skierowanego $G = (V, A)$, taka że $n = |V|$, $-\infty < d_{min} \leqslant d_{max} < \infty$; początkowy wierzchołek $s = v_s$; końcowy wierzchołek $t = v_t$; minimalna liczba wierzchołków do odwiedzenia k.

1: Utwórz graf nieskierowany $\overline{G}$ o macierzy odległości $\overline{D}$ o wymiarach $2n \times 2n$ zgodnej z równaniem (4.7).
2: Znajdź w $\overline{G}$ $2k$-drzewo rozpinające $\overline{T}_{2k}$ zawierające wierzchołki v_s i v_t. Niech to drzewo będzie podstawą do zbudowania grafu $\overline{G}'$.
3: Jeżeli wierzchołek v_s jest parzystego stopnia, to dodaj w $\overline{G}'$ dodatkową krawędź o zerowym koszcie pomiędzy wierzchołkami v_s i $v_{[s]}$, aby uzyskać nieparzysty stopień wierzchołka v_s. W analogiczny sposób zapewnij w $\overline{G}'$ nieparzystość stopnia dla wierzchołka v_t.
4: Niech V_{odd} i $[V]_{odd}$ będą odpowiednio zbiorami rzeczywistych i wirtualnych wierzchołków o nieparzystym stopniu w $\overline{G}'$. Dodaj w $\overline{G}'$ dodatkowe krawędzie o koszcie równym 0 pomiędzy komplementarnymi wierzchołkami w taki sposób, aby $|V_{odd}| = |[V]_{odd}| + 2$, przy czym dodawanie kolejnych krawędzi pomiędzy v_s i $v_{[s]}$ oraz v_t i $v_{[t]}$ jest zabronione.
5: Znajdź skojarzenie doskonałe $\overline{M}^*$ o minimalnym koszcie dla zbioru pozostałych wierzchołków w $\overline{G}'$ o nieparzystym stopniu (z wyjątkiem wierzchołków v_s i v_t) i dodaj krawędzie $\overline{M}^*$ do $\overline{G}'$.
6: Znajdź drogę Eulera $\overline{E}_{2k}$ w grafie $\overline{G}'$.
7: Kolejność odwiedzanych wierzchołków zdefiniowana przez drogę $\overline{E}_{2k}$ odpowiada pewnej v_s-v_t $2k$-ścieżce $\overline{P}^{st}_{2k}$ w grafie $\overline{G}$.
8: Ścieżka $\overline{P}^{st}_{2k}$ może odwiedzać pewne wierzchołki wielokrotnie. Usuń z sekwencji odwiedzanych przez nią wierzchołków każde kolejne wystąpienie odwiedzonego już wcześniej wierzchołka (przejdź bezpośrednio do kolejnego nieodwiedzonego wierzchołka). Aby zapewnić naprzemienne odwiedzanie wierzchołków rzeczywistych i wirtualnych niezbędne może być również usunięcie jednego wierzchołka z komplementarnej pary.
9: Na podstawie ścieżki $\overline{P}^{st}_{2k}$ skonstruuj v_s-v_t k-ścieżkę P^{st}_k w oryginalnym skierowanym grafie G.

Odpowiedź: v_s-v_t k-ścieżka P^{st}_k w grafie G.

7.2.2.1 Konstrukcja grafu semieulerowskiego

W niniejszym rozdziale wykazano, że proces utworzenia grafu półeulerowskiego zdefiniowany przez Algorytm 2 (kroki 2-5) jest możliwy dla dowolnego grafu zdefiniowanego przez macierz odległości $\overline{D}$ utworzoną zgodnie z równaniem (4.7).

Wniosek 7.1. *Niech $\overline{T}^*_{2k}$ oznacza minimalne $2k$-drzewo rozpinające w grafie zdefiniowanym przez macierz odległości $\overline{D}$ utworzoną zgodnie z równaniem (4.7). Wtedy spełnione są następujące warunki.*

1. *$\overline{T}^*_{2k}$ nie zawiera krawędzi pomiędzy dwoma rzeczywistymi ani pomiędzy dwoma wirtualnymi wierzchołkami.*
2. *$\overline{T}^*_{2k}$ zawiera krawędzie pomiędzy wierzchołkami komplementarnymi.*

Dowód. Pierwszy punkt wynika z faktu, że koszt dowolnej krawędzi pomiędzy dwoma rzeczywistym/wirtualnymi wierzchołkami wynosi nieskończoność, a $d_{max} < \infty$. Drugi punkt wynika z faktu, że koszt krawędzi pomiędzy komplementarnymi wierzchołkami z definicji jest równy 0, a z definicji macierzy D' (równanie (4.6)) i $\overline{D}$ (równanie (4.7)) koszt krawędzi pomiędzy dowolną parą wierzchołków: rzeczywistego i i wirtualnego $[j]$ $(i \neq j)$ jest większy od 0. □

Lemat 7.2. *$\overline{T}^*_{2k}$ zawiera albo oba wierzchołki z danej komplementarnej pary albo nie zawiera żadnego z nich.*

Dowód. Niech $\overline{T}^*_{2k}$ hipotetycznie zawiera pewne rzeczywiste lub wirtualne wierzchołki, których komplementarne wierzchołki nie są częścią $\overline{T}^*_{2k}$. W związku z tym, że drzewo zawiera parzystą liczbę wierzchołków $(2k)$, to muszą istnieć przynajmniej dwa takie wierzchołki, których komplementarne odpowiedniki nie należą do $\overline{T}^*_{2k}$. Co więcej, możliwe jest dodanie tych wierzchołków do drzewa przez połączenie krawędziami o zerowym koszcie z komplementarnymi wierzchołkami będącymi już częścią drzewa. Prowadzi to do powstania $(2k+2)$-drzewa bez wzrostu kosztu. Usuwając z niego dowolną parę komplementarnych wierzchołków powstaje $2k$-drzewo rozpinające o niższym koszcie. Zaprzecza to optymalności $\overline{T}^*_{2k}$. □

Lemat 7.3. *$\overline{T}^*_{2k}$ zawiera przynajmniej jeden taki wierzchołek o nieparzystym stopniu, że jego komplementarny wierzchołek jest stopnia parzystego.*

Dowód. Z Lematu 7.2 wynika, że $2k$-drzewo $\overline{T}^*_{2k}$ może zostać utworzone przez iteracyjne dodawanie kolejnych par komplementarnych wierzchołków. Niech v_i $v_{[i]}$ będzie ostatnią dodaną parą przez połączenie krawędzią wierzchołka v_i z pewnym wirtualnym wierzchołkiem należącym do drzewa. W takiej sytuacji stopień wierzchołka v_i jest równy 2, a $v_{[i]}$ jest liściem (stopień jest równy 1). Jeżeli powyższa para została dołączona do drzewa przez krawędź pomiędzy wierzchołkiem $v_{[i]}$ i pewnym rzeczywistym wierzchołkiem należącym do drzewa, to wtedy stopień $v_{[i]}$ jest równy dwa, a v_i jest liściem. Z Wniosku 7.1 wynika, że nie ma innego sposobu włączenia nowej pary do drzewa. □

Lemat 7.4. *Istnieje możliwość zbalansowania w $\overline{T}^*_{2k}$ liczby rzeczywistych i wirtualnych wierzchołków nieparzystego stopnia przez dodanie krawędzi o zerowym koszcie pomiędzy komplementarnymi wierzchołkami.*

Dowód. Z Lematu 7.2 wynika, że możliwe jest zbudowanie $2k$-drzewa rozpinającego $\overline{T}^*_{2k}$ przez iteracyjne dodawanie do bieżącego rozwiązania kolejnych par komplementarnych wierzchołków.

Początkowo rozwiązanie zawiera jedną z par. Niech tą parą będzie v_p i $v_{[p]}$. Rozwiązanie początkowe składa się więc z jednego rzeczywistego i jednego wirtualnego wierzchołka połączonych krawędzią $\{v_p, v_{[p]}\}$. Stopień obu wierzchołków jest równy 1. Liczba rzeczywistych i wirtualnych wierzchołków nieparzystego stopnia jest tak sama, a warunek ten jest niezmiennikiem. Po dodaniu kolejnej pary wierzchołków nadal jest on spełniony lub możliwe jest utworzenie dodatkowej krawędzi celem przywrócenia balansu pomiędzy liczbą rzeczywistych i wirtualnych wierzchołków nieparzystego stopnia.

W każdym kroku dodawana jest kolejna para komplementarnych wierzchołków v_i i $v_{[i]}$ połączonych krawędzią $\{v_i, v_{[i]}\}$. Z Wniosku 7.1 wynika, że istnieją dwie możliwości dodania danej pary. Jeżeli zostanie ona dołączona do wierzchołka o nieparzystym stopniu, to jego stopień staje się parzysty, ale równocześnie dodawany jest liść tego samego

rodzaju (odpowiednio rzeczywisty lub wirtualny). Taka operacja nie narusza więc wspomnianego niezmiennika.

Możliwe jest również, że nowa para zostanie dołączona do wierzchołka o stopniu parzystym, co spowoduje naruszenie niezmiennika. Co więcej, do drzewa zostanie dodany liść tego samego rodzaju (rzeczywisty lub wirtualny). Spełnienie warunku niezmiennika może zostać przywrócone poprzez utworzenie dodatkowej krawędzi pomiędzy wierzchołkami z dodawanej właśnie pary komplementarnych wierzchołków. □

Uwaga 7.5. Nie ma potrzeby tworzenia nowych krawędzi po dodaniu każdej nowej pary komplementarnych wierzchołków, ponieważ pewne operacje mogą się wzajemnie zbilansować. Ewentualne utworzenie dodatkowych krawędzi można zrealizować na samym końcu po dodaniu wszystkich par.

Niech $\overline{T}_{2k}^{B*}$ będzie grafem otrzymanym w wyniku dodania do drzewa $\overline{T}_{2k}^{*}$ dodatkowych krawędzi celem zbalansowania liczby rzeczywistych i wirtualnych wierzchołków nieparzystego stopnia, tj. $\overline{T}_{2k}^{B*}$ spełnia warunek $|V_{odd}| = |[V]_{odd}|$. Należy zauważyć, że $\overline{T}_{2k}^{B*}$ może nie być drzewem.

Wniosek 7.6. *Niech graf $\overline{T}_{2k}^{B*}$ zawiera taką komplementarną parę wierzchołków v_i i $v_{[i]}$, że stopień $v_{[i]}$ jest nieparzysty, a stopień v_i jest parzysty. Wtedy możliwe jest utworzenie dodatkowej krawędzi pomiędzy v_i i $v_{[i]}$ celem otrzymania pewnego grafu, którego łączna suma kosztów krawędzi będzie taka sama jak dla $\overline{T}_{2k}^{B*}$ oraz spełniony będzie warunek $|V_{odd}| = |[V]_{odd}| + 2$.*

Dowód. Dodanie w $\overline{T}_{2k}^{B*}$ krawędzi $\{v_i, v_{[i]}\}$ zmniejsza liczbę wirtualnych wierzchołków o stopniu nieparzystym o jeden i jednocześnie zwiększa liczbę rzeczywistych wierzchołków o stopniu nieparzystym o jeden. Krawędź jest tworzona pomiędzy wierzchołkami komplementarnymi, więc jej koszt z definicji jest równy 0 i nie wpływa na łączny koszt wszystkich krawędzi w grafie $\overline{T}_{2k}^{B*}$. □

Lemat 7.7. *Niech $\overline{T}_{2k}^{B*}$ będzie grafem zawierającym taką samą liczbę rzeczywistych i wirtualnych wierzchołków o stopniu nieparzystym. Dodatkowo niech będą wyróżnione dwa rzeczywiste wierzchołki tego grafu: v_s i v_t. Możliwe jest otrzymanie pewnego grafu o takim samym łącznym koszcie wszystkich krawędzi jak dla $\overline{T}_{2k}^{B*}$, w którym v_s i v_t są stopnia nieparzystego i spełniony jest warunek $|V_{odd}| = |[V]_{odd}| + 2$. Taki graf uzyskać można w następujący sposób.*

1. *Jeżeli stopień wierzchołka v_s jest parzysty, to należy dodać dodatkową krawędź o zerowym koszcie pomiędzy v_s i $v_{[s]}$. W razie potrzeby należy wykonać analogiczną operację dla wierzchołka v_t.*
2. *Należy utworzyć dodatkowe krawędzie pomiędzy komplementarnymi wierzchołkami celem zapewnienia $|V_{odd}| = |[V]_{odd}| + 2$, przy czym dodawanie nowych krawędzi pomiędzy v_s i $v_{[s]}$ oraz v_t i $v_{[t]}$ jest zabronione.*

Dowód. Koszt nowego grafu nie ulega zmianie, bo dodawane są jedynie krawędzie o zerowym koszcie pomiędzy komplementarnymi wierzchołkami.

Poniżej zostanie wykazane, że utworzenie dodatkowych krawędzi celem zapewnienia

nieparzystego stopnia dla wierzchołka v_s nie zmieni balansu pomiędzy liczbą rzeczywistych i wirtualnych wierzchołków o stopniu nieparzystym lub jest możliwe jego przywrócenie (analogicznie można to wykazać dla wierzchołka v_t).

Z Lematu 7.2 wynika, że $\overline{T}_{2k}^{B*}$ zawiera również wierzchołki $v_{[s]}$ i $v_{[t]}$. Jeżeli stopień obu wierzchołków v_s i $v_{[s]}$ jest parzysty, to utworzenie dodatkowej krawędzi pomiędzy nimi nie naruszy balansu. Jeżeli stopień wierzchołka v_s jest parzysty, a wierzchołka $v_{[s]}$ nieparzysty, to musi istnieć inna para komplementarnych wierzchołków v_i i $v_{[i]}$ taka, że stopień v_i jest nieparzysty, a stopień $v_{[i]}$ jest parzysty, aby wspomniany balans był zachowany. Jeżeli tą parą jest v_t i $v_{[t]}$, to utworzenie dodatkowej krawędzi $\{v_s, v_{[s]}\}$ doprowadzi do otrzymania grafu o pożądanych właściwościach ($|V_{odd}| = |[V]_{odd}| + 2$) i żadne dalsze akcje nie są wykonywane. W przeciwnym przypadku dodawana jest krawędź $\{v_i, v_{[i]}\}$ celem przywrócenia balansu.

Po zrealizowaniu pierwszego kroku w otrzymanym grafie stopień wierzchołków v_s i v_t jest nieparzysty, a liczba wierzchołków rzeczywistych i wirtualnych o stopniu nieparzystym jest taka sama (z wyjątkiem kiedy otrzymany graf spełnia oczekiwane własności i dalsze modyfikacje nie są konieczne).

W kroku drugim konieczne jest zapewnienie $|V_{odd}| = |[V]_{odd}| + 2$. Aby skorzystać z Wniosku 7.6 konieczne jest wykazanie, że w grafie istnieje taka para komplementarnych wierzchołków v_i i $v_{[i]}$ różna od v_s i $v_{[s]}$ oraz różna od v_t i $v_{[t]}$, że stopień v_i jest parzysty, a stopień $v_{[i]}$ jest nieparzysty.

Niech hipotetycznie stopień przynajmniej jednego z wierzchołków $v_{[s]}$ lub $v_{[t]}$ jest parzysty. Po wykonaniu pierwszego kroku stopień obu wierzchołków v_s i v_t jest nieparzysty, a balans jest zachowany, więc graf musi zawierać przynajmniej jedną taką parę wierzchołków v_i i $v_{[i]}$, że stopień v_i jest parzysty, a stopień $v_{[i]}$ jest nieparzysty.

Jeżeli jednak stopień wszystkich czterech wierzchołków v_s, $v_{[s]}$, v_t i $v_{[t]}$ jest nieparzysty, to z Lematu 7.3 wynika, że niezależnie musi istnieć w grafie taki wierzchołek stopnia parzystego, iż jego komplementarny wierzchołek jest stopnia nieparzystego. Wcześniejsze tworzenie dodatkowych krawędzi pomiędzy wierzchołkami z komplementarnej pary nie wpływa na poprawność tego wniosku, bo taka nowa krawędź wpływa jednocześnie na zmianę parzystości stopnia obu wierzchołków z pary. Niech taką parą będzie v_i i $v_{[i]}$. Graf ma identyczną liczbę wierzchołków rzeczywistych i wirtualnych o stopniu nieparzystym, więc musi również istnieć inna para komplementarnych wierzchołków v_j i $v_{[j]}$, aby zapewnić balans. Jedna z tych par (v_i i $v_{[i]}$ lub v_j i $v_{[j]}$) musi być więc taką parą, że stopień wierzchołka rzeczywistego jest parzysty, a stopień wierzchołka wirtualnego jest nieparzysty. Dowodzi to możliwości skorzystania z Wniosku 7.6 i kończy cały dowód. □

Uwaga 7.8. Czasami może być możliwe przywrócenie balansu przez usunięcie utworzonej wcześniej dodatkowej krawędzi pomiędzy wierzchołkiem rzeczywistym i wirtualnym. Taka krawędź może zostać usunięta (zamiast tworzenia nowej) celem przywrócenia balansu pomiędzy liczbą rzeczywistych i wirtualnych wierzchołków nieparzystego stopnia. Umożliwia to ograniczenie liczby dodatkowo tworzonych krawędzi.

Twierdzenie 7.9. *Niech będzie dany graf $\overline{G}$ określony przez macierz odległości $\overline{D}$ zdefiniowaną zgodnie z równaniem (4.7), $2k$-drzewo rozpinające $\overline{T}_{2k}$ w grafie $\overline{G}$ zawierające dwa wierzchołki v_s i v_t. Wtedy możliwe jest utworzenie grafu semieulerowskiego $\overline{G}'$ zawierającego drogę Eulera z wierzchołka v_s do v_t w następujący sposób.*

1. *Początkowo graf $\overline{G}'$ jest równy drzewu $\overline{T}_{2k}$.*

2. *Jeżeli stopień wierzchołka v_s jest parzysty, to aby uzyskać nieparzysty stopień tego wierzchołka utwórz w $\overline{G}'$ dodatkową krawędź o zerowym koszcie pomiędzy wierzchołkami v_s i $v_{[s]}$. Nieparzysty stopień wierzchołka v_t w $\overline{G}'$ można zapewnić analogicznie.*

3. *Dodaj w $\overline{G}'$ dodatkowe krawędzie o zerowym koszcie pomiędzy wierzchołkami komplementarnymi, aby spełnić $|V_{odd}| = |[V]_{odd}| + 2$, przy czym tworzenie kolejnych krawędzi pomiędzy parami v_s i $v_{[s]}$ oraz v_t i $v_{[t]}$ jest zabronione.*

4. *Znajdź w grafie $\overline{G}'$ uzupełnionym w poprzednich krokach o dodatkowe krawędzie doskonałe skojarzenie $\overline{M}^*$ o minimalnym koszcie. Do poszukiwania skojarzenia wykorzystaj jedynie podgraf składający się z rzeczywistych i wirtualnych wierzchołków o stopniu nieparzystym z pominięciem wierzchołków v_s and v_t.*

5. *Dodaj do $\overline{G}'$ krawędzie należące do skojarzenia $\overline{M}^*$. Otrzymany graf $\overline{G}'$ jest grafem semieulerowskim.*

Dowód. Na podstawie Lematu 7.7 w korku 2 i 3 tworzone są dodatkowe krawędzie, aby w otrzymanym grafie stopień wierzchołków v_s i v_t był nieparzysty oraz spełnione było równanie $|V_{odd}| = |[V]_{odd}| + 2$. Następnie szukane jest doskonałe skojarzenie $\overline{M}^*$. Wierzchołki v_s i v_t nie są przy tym brane pod uwagę, więc liczba rzeczywistych i wirtualnych wierzchołków w skojarzeniu $\overline{M}^*$ jest taka sama. Dzięki temu skojarzenie $\overline{M}^*$ nie zawiera krawędzi o nieskończonym koszcie, a po dodaniu krawędzi z $\overline{M}^*$ do $\overline{G}'$ nieparzysty stopień wierzchołków v_s i v_t nie ulegnie zmianie. Co więcej, dodanie krawędzi skojarzenia $\overline{M}^*$ do $\overline{G}'$ doprowadzi do tego, że stopień wszystkich wierzchołków z wyjątkiem v_s i v_t będzie parzysty, co gwarantuje możliwość znalezienia drogi Eulera z wierzchołka v_s do v_t. Otrzymany graf jest więc grafem semieulerowskim. □

7.2.2.2 Gwarancja dokładności

Konstrukcja rozwiązania rozpoczyna się od znalezienia w grafie $\overline{G}$ (określonym przez macierz odległości $\overline{D}$) $2k$-drzewa rozpinającego $\overline{T}_{2k}$ zawierającego wierzchołki s i t. Niech $c_{\overline{D}}(\overline{T}_{2k})$ oznacza koszt tego drzewa (tj. sumę kosztów jego krawędzi), $\overline{P}^{st*}_{2k}$ będzie minimalną s-t ścieżką w $\overline{G}$ zawierającą przynajmniej $2k$ wierzchołków, a $c_{\overline{D}}(\overline{P}^{st*}_{2k})$ oznacza koszt tej ścieżki. Wtedy spełnione jest następujące równanie.

$$c_{\overline{D}}(\overline{T}_{2k}) \leqslant (1+\delta) c_{\overline{D}}(\overline{P}^{st*}_{2k}) \tag{7.3}$$

gdzie δ jest dowolną wartością większą od zera [22].

W kroku 3 i 4 tworzone są dodatkowe krawędzie w grafie $\overline{G}'$ celem zapewnienia, aby $\overline{G}'$ był grafem półeulerowskim. Krawędzie są tworzone jedynie pomiędzy komplementarnymi wierzchołkami i koszt tych krawędzi z definicji jest równy 0. Nie powodują one więc wzrostu całkowitego kosztu wszystkich krawędzi w $\overline{G}'$.

W kroku 5 poszukiwane jest skojarzenie doskonałe $\overline{M}^*$ o minimalnym koszcie dla zbioru wierzchołków, których stopień jest nieparzysty. Kumar i Le udowodnili w [39], że w kontekście poszukiwania optymalnego (o minimalnym koszcie) cyklu Hamiltona $\overline{C}^*_H$ w grafie $\overline{G}$ koszt tego skojarzenia doskonałego jest ograniczony równaniem $c_{\overline{D}}(\overline{M}^*) < \frac{2}{3} c_{\overline{D}}(\overline{C}^*_H)$. W analogiczny sposób można udowodnić następującą zależność.

$$c_{\overline{D}}(\overline{M}^*) < \frac{2}{3} c_{\overline{D}}(\overline{P}^{st*}_{2k}) \tag{7.4}$$

Semieulerowski graf $\overline{G}'$ powstaje z połączenia $\overline{T}_{2k}$, $\overline{M}^*$ i dodatkowych krawędzi o zerowym koszcie, więc górne ograniczenie kosztu drogi Eulera $\overline{E}_{2k}$ w grafie $\overline{G}'$ jest następujące.

$$c_{\overline{D}}(\overline{E}_{2k}) < \left(\frac{5}{3} + \delta\right) c_{\overline{D}}(\overline{P}^{st*}_{2k}) \tag{7.5}$$

Na podstawie drogi Eulera $\overline{E}_{2k}$ konstruowana jest s-t $2k$-ścieżka $\overline{P}^{st}_{2k}$. Spełnianie przez macierz odległości $\overline{D}$ asymetrycznej nierówności trójkąta gwarantuje, że pominięcie odwiedzonych już wcześniej wierzchołków nie prowadzi do wzrostu kosztu ścieżki, więc ostatecznie po wykonaniu kroku 8 spełniona jest nierówność $\overline{P}^{st}_{2k} \leqslant \overline{E}_{2k}$, a górne ograniczenie kosztu $\overline{P}^{st}_{2k}$ jest następujące.

$$c_{\overline{D}}(\overline{P}^{st}_{2k}) < \left(\frac{5}{3} + \delta\right) c_{\overline{D}}(\overline{P}^{st*}_{2k}) \tag{7.6}$$

W ostatnim kroku na podstawie $\overline{P}^{st}_{2k}$ tworzona jest s-t k-ścieżka P^{st}_k w oryginalnym grafie G. W trakcie tego procesu może zostać wykorzystany przeciwny łuk niż ten, którego koszt został uwzględniony przy konstrukcji $\overline{P}^{st}_{2k}$, co może prowadzić do wzrostu kosztu P^{st}_k. W związku z tym, że dla macierzy D' spełniony jest warunek $\frac{d'_{max}}{d'_{min}} < \frac{4}{3}$, to dla dowolnych jej indeksów i, j $(i \neq j)$ spełnione jest $d'_{ij} < \frac{4}{3} d'_{ji}$. Prowadzi to do spełnienia następującego równania.

$$c_{D'}(P^{st}_k) < \frac{4}{3} c_{\overline{D}}(\overline{P}^{st}_{2k}) \tag{7.7}$$

Z równań (7.6) i (7.7) wynika poniższa zależność.

$$c_{D'}(P^{st}_k) < \left(\frac{4}{3}\right)\left(\frac{5}{3} + \delta\right) c_{\overline{D}}(\overline{P}^{st*}_{2k}) = \left(\frac{20}{9} + \delta'\right) c_{\overline{D}}(\overline{P}^{st*}_{2k}) \tag{7.8}$$

gdzie $\delta' = \frac{4}{3}\delta$. W związku z tym, że δ jest dowolną liczbą większą od 0, to w dalszej części stały współczynnik został pominięty i zamiast δ' wykorzystywany jest symbol δ.

Na podstawie równania (4.6) można obliczyć zależność kosztu ścieżki $\overline{P}^{st}_{2k}$ względem kosztu ścieżki P^{st}_k.

$$c_{\overline{D}}(\overline{P}^{st}_{2k}) = \begin{cases} c_D(P^{st}_k) & \text{jeżeli } 4d_{min} - 3d_{max} > 0 \\ c_D(P^{st}_k) + (k'-1)(3d_{max} - 4d_{min} + \epsilon) & \text{w przeciwnym razie} \end{cases} \tag{7.9}$$

gdzie k' jest liczbą wierzchołków odwiedzanych przez k-ścieżkę P^{st}_k $(k' \geqslant k)$.

Twierdzenie 7.10. *Niech będzie dany skierowany graf G określony przez macierz odległości D spełniającą asymetryczną nierówność trójkąta o wymiarach $n \times n$, taką że $n = |V|$ oraz $-\infty < d_{min} \leqslant d_{max} < \infty$. Niech $\gamma = \frac{d_{max}}{d_{min}}$. Wtedy koszt s-t k-ścieżki P^{st}_k skonstruowanej zgodnie z Algorytmem 2 jest następujący.*

$$c_D(P^{st}_k) \leqslant \begin{cases} \left(\frac{20}{9} + \delta\right) c_D(P^{st*}_k) & \textit{jeżeli } \gamma < \frac{4}{3} \\ \left(\frac{11\gamma - 8}{3} + \delta\right) c_D(P^{st*}_k) & \textit{w przeciwnym razie} \end{cases} \tag{7.10}$$

Dowód. Poniższy dowód jest analogiczny do dowodu Kumar i Le z pracy [39], gdzie wykazano gwarancję dokładności algorytmu aproksymacyjnego dla ATSP.

W pierwszej kolejności zostanie wykazana prawdziwość powyższego twierdzenia dla

przypadku, gdy $\gamma < \frac{4}{3}$. Wtedy z równania (4.6) wynika, że $c_{D'}(P_k^{st}) = c_D(P_k^{st})$. Co więcej, z równania (7.9) wynika, że $c_{\overline{D}}(\overline{P}_{2k}^{st*}) = c_D(P_k^{st*})$. Łącząc powyższe z równaniem (7.8) otrzymuje się następujące zależności.

$$c_D(P_k^{st}) = c_{D'}(P_k^{st}) < \left(\frac{20}{9} + \delta\right) c_{\overline{D}}(\overline{P}_{2k}^{st*}) = \left(\frac{20}{9} + \delta\right) c_D(P_k^{st*})$$

Jeżeli $\gamma \geqslant \frac{4}{3}$, to niech $\lambda = 3d_{max} - 4d_{min} + \epsilon = (3\gamma - 4 + \frac{\epsilon}{d_{min}})d_{min}$. Wtedy z równania (4.6) wynika, że $c_{D'}(P_k^{st}) = c_D(P_k^{st}) + (k'-1)\lambda$. Jednocześnie z równania (7.9) wynika, że $c_{\overline{D}}(\overline{P}_{2k}^{st*}) = C_D(P_k^{st*}) + (k'-1)\lambda$. Z połączenia z równaniem (7.8) wynikają następujące zależności.

$$c_D(P_k^{st}) + (k'-1)\lambda = c_{D'}(P_k^{st}) < \left(\frac{20}{9} + \delta\right) c_{\overline{D}}(\overline{P}_{2k}^{st*}) = \left(\frac{20}{9} + \delta\right) \left(c_D(P_k^{st*}) + (k'-1)\lambda\right)$$

Powyższa nierówność może zostać uproszczona do następującej postaci.

$$c_D(P_k^{st}) < \left(\frac{20}{9} + \delta\right) c_D(P_k^{st*}) + \left(\frac{11}{9} + \delta\right) (k'-1)\lambda$$

Po podstawieniu $\lambda = (3\gamma - 4 + \frac{\epsilon}{d_{min}})d_{min}$ otrzymuje się następującą nierówność.

$$c_D(P_k^{st}) < \left(\frac{20}{9} + \delta\right) c_D(P_k^{st*}) + \left(\frac{11}{9} + \delta\right) (3\gamma - 4 + \frac{\epsilon}{d_{min}})(k'-1)d_{min} \tag{7.11}$$

W związku z tym, że ϵ może być dowolnie małą liczbą dodatnią można pominąć element $\frac{\epsilon}{d_{min}}$ i zastąpić ostrą nierówność przez nierówność nieostrą. Co więcej, $(k'-1)d_{min}$ jest dolnym ograniczeniem kosztu s-t k-ścieżki, więc $(k'-1)d_{min} \leqslant c_D(P_k^{st*})$. Uwzględnienie obu powyższych obserwacji w równaniu (7.11) prowadzi do drugiej części wzoru opisującego gwarancję dokładności.

$$c_D(P_k^{st}) \leqslant \left(\frac{20}{9} + \delta\right) c_D(P_k^{st*}) + \left(\frac{11}{9} + \delta\right) (3\gamma - 4) c_D(P_k^{st*}) = \left(\frac{11\gamma - 8}{3} + \delta\right) c_D(P_k^{st*})$$

□

7.2.2.3 Złożoność czasowa

Najbardziej złożony czasowo krok algorytmu to konstrukcja k-drzewa rozpinającego za pomocą algorytmu zaproponowanego przez Chaudhuri, który wymaga czasu rzędu $O(n^3 \log n)$ [22]. Poszczególne kroki wykonywane są sekwencyjnie dokładnie raz, więc złożoność konstrukcji k-drzewa rozpinającego determinuje złożoność całego algorytmu.

7.3 Algorytm dla problemu minimalnej k-ścieżki

Algorytm 1 może zostać wykorzystany do znalezienia rozwiązania dla problemu minimalnej k-ścieżki. Mając dany graf skierowany G opisany przez macierz odległości D można znaleźć minimalną s-t k-ścieżkę dla wszystkich możliwych kombinacji s i t i wybrać najkrótszą z nich.

Gwarancja dokładności Algorytmu 3 jest taka sama jak w przypadku Algorytmu 1

Algorytm 3 Algorytm aproksymacyjny dla problemu minimalnej k-ścieżki w grafie skierowanym.

Instancja: macierz odległości D spełniająca asymetryczną nierówność trójkąta o wymiarach $n \times n$ dla grafu skierowanego $G = (V, A)$, taka że $n = |V|$, $-\infty < d_{min} \leqslant d_{max} < \infty$; minimalna liczba wierzchołków do odwiedzenia k.

1: Dla każdej możliwej kombinacji wierzchołków s i t ($s \neq t$) wykonaj:
2: znajdź minimalną s-t k-ścieżkę (Algorytm 1),
3: zaktualizuj zapamiętaną ścieżkę (jeżeli znaleziono krótszą niż zapamiętana).
4: Wynikiem jest zapamiętana ścieżka.

Odpowiedź: k-ścieżka w G.

i jest określona równaniem (7.2). Istnieje $(n \times (n-1))$ możliwych kombinacji wierzchołków s i t, więc niezbędne jest $O(n^2)$ wywołań Algorytmu 1. W konsekwencji złożoność czasowa Algorytmu 3 jest rzędu $O(n^5 \log n)$.

Uwaga 7.11. Jeżeli $\frac{10}{9} < \frac{d_{max}}{d_{min}} \leqslant \frac{4}{3}$, to zamiast Algorytmu 1 można wykorzystać Algorytm 2. Zapewni to lepszą gwarancję dokładności: $\frac{20}{9} + \delta$.

7.4 Algorytm dla problemu Orienteering

Kluczowym czynnikiem dla problemu minimalnej k-ścieżki jest liczba wierzchołków, które należy odwiedzić. W przypadku problemu Orienteering takim czynnikiem jest budżet, który ogranicza maksymalny koszt ścieżki. Dodatkowo w przypadku Orienteering należy uwzględnić wartość poszczególnych wierzchołków. W przypadku problemu minimalnej k-ścieżki wszystkie wierzchołki mają taką samą wartość równą 1, a w przypadku Orienteering pewne wierzchołki mogą być bardziej wartościowe niż pozostałe.

Zaprezentowany poniżej algorytm dla skierowanego grafu $G = (V, A)$ zakłada, że wartość $\psi(v)$ dowolnego wierzchołka $v \in V$ należy do zakresu $[1, n^2]$, gdzie $n = |V|$. Aby to założenie było spełnione, można wykorzystać technikę zaproponowaną przez Blum w [18]. Należy przeskalować poszczególne wartości w taki sposób, aby maksymalna wartość była równa dokładnie n^2, a następnie zaokrąglić je w dół do najbliższej liczby całkowitej. Powoduje to utratę w najgorszym przypadku całkowitej wartości otrzymanej ścieżki o n. Należy również zauważyć, że po przeskalowaniu wartości poszczególnych wierzchołków minimalna wartość rozwiązania wynosi n^2, gdyby ścieżka zawierała jeden wierzchołek o maksymalnej wartości. Istnieje n wierzchołków, więc maksymalna wartość rozwiązania jest równa n^3.

Problem Orienteering może zostać rozwiązany z wykorzystaniem Algorytmu 3 dla problemu minimalnej k-ścieżki w następujący sposób. Dany graf G jest modyfikowany celem uwzględnienia wartości poszczególnych wierzchołków [18]. Dla każdego wierzchołka $v \in V$ o wartości $\psi(v)$ dodawanych jest $\psi(v) - 1$ wierzchołków (tzw. *liści*), które łączone są z oryginalnym wierzchołkiem za pomocą łuków o zerowym koszcie. Następnie "zgadywana" jest optymalna wartość Ψ^* należąca do przedziału $[n^2, n^3]$, tzn. jest ona wyznaczana za pomocą przeszukiwania binarnego. Optymalna wartość Ψ^* jest wykorzystywana jako parametr Algorytmu 3, tj. k $= \Psi^*$. Wynikiem jest ścieżka o łącznej wartości co najmniej Ψ^* i koszcie co najwyżej $\alpha \cdot c(P_k^*)$, gdzie α jest gwarancją dokładności Algorytmu 3, a $c(P_k^*)$ nie przekracza podanego budżetu B. Otrzymana ścieżka może być podzielona na α fragmentów o koszcie co najwyżej $c(P_k^*)$, a jako wynik może zostać wybrany fragment o największej sumarycznej wartości.

Fragment ścieżki wybrany w ostatnim kroku ma łączną wartość Ψ, której dolne

Algorytm 4 Algorytm aproksymacyjny dla problemu Orienteering w grafie skierowanym.

Instancja: macierz odległości D spełniająca asymetryczną nierówność trójkąta o wymiarach $n \times n$ dla grafu skierowanego $G = (V, A)$, taka że $n = |V|$, $-\infty < d_{min} \leqslant d_{max} < \infty$; budżet B; algorytm aproksymacyjny A dla problemu minimalnej k-ścieżki w grafie skierowanym o gwarancji dokładności α.

1: Przeskaluj wartości wierzchołków tak, aby maksymalna wartość wierzchołków była równa n^2, a następnie zaokrąglij otrzymane wartości wierzchołków w dół do najbliższej liczby całkowitej.
2: Dla każdego wierzchołka $v \in V$ o wartości $\psi(v)$ dodaj w G $\psi(v) - 1$ liści.
3: "Zgadnij" optymalną wartość rozwiązania Ψ^*.
4: Użyj algorytmu A do znalezienia k-ścieżki P_k o koszcie co najwyżej αB dla $k = \Psi^*$.
5: Podziel otrzymaną k-ścieżkę P_k na α fragmentów o koszcie co najwyżej B i wybierz fragment o największej wartości.

Odpowiedź: ścieżka w G o koszcie nie przekraczającym B.

ograniczenie wynosi Ψ^*/α, więc Algorytm 4 jest algorytmem $(1/\alpha)$-aproksymacyjnym. Podstawiając pod α gwarancję dokładności dla Algorytmu 3 dla problemu minimalnej k-ścieżki w grafie skierowanym (równanie 7.2) otrzymuje się następującą gwarancję dokładności dla Algorytmu 4.

$$\Psi \geqslant \frac{\Psi^*}{2\gamma + \delta}, \qquad \text{gdzie } \gamma = \frac{d_{max}}{d_{min}} \text{ oraz } \delta > 0$$

Uwaga 7.12. Jeżeli $\frac{10}{9} < \frac{d_{max}}{d_{min}} \leqslant \frac{4}{3}$, to w trakcie poszukiwania minimalnej k-ścieżki jako podprocedurę można wykorzystać Algorytm 2 zamiast Algorytmu 1. Umożliwi to uzyskanie lepszej gwarancji dokładności: $1/(\frac{20}{9} + \delta)$.

Najbardziej złożoną czasowo częścią powyższego algorytmu jest proces poszukiwania optymalnej wartości Ψ^* i uzyskanie minimalnej k-ścieżki dla poszczególnych wartości Ψ w ramach przeszukiwania binarnego. Niezbędnych jest $O(\log n)$ uruchomień algorytmu znajdującego minimalną k-ścieżkę, co prowadzi do łącznej złożoności czasowej $O(n^5 \log^2 n)$ (przy założeniu wykorzystania jako podprocedury Algorytmu 3).

7.4.1 Gwarancja dokładności dla klasycznego problemu SBH

W przypadku zastosowania Algorytmu 4 do rozwiązania problemu obliczeniowego dla klasycznego sekwencjonowania przez hybrydyzację z błędami dowolnego typu możliwe jest określenie wartości γ. Zależy ona od długości l-merów, tj. $\gamma = l$. Prowadzi to do następującej gwarancji dokładności.

$$\Psi \geqslant \frac{\Psi^*}{2l + \delta}, \qquad \text{gdzie } \delta > 0$$

gdzie Ψ reprezentuje liczbę oligonukleotydów ze spektrum w otrzymanym rozwiązaniu, a Ψ^* reprezentuje optymalną liczbę oligonukleotydów (tj. maksymalną liczbę l-merów ze spektrum, które ułożone kolejno reprezentują sekwencję nie dłużą niż długość n analizowanej sekwencji).

Rozdział 8

Heurystyki dla problemów sekwencjonowania DNA przez hybrydyzację z częściową informacją o powtórzeniach

8.1 Wprowadzenie

Algorytmy dokładne dla problemów trudnych obliczeniowo mają ograniczone zastosowanie w praktyce. Czas niezbędny do uzyskania rozwiązania optymalnego rośnie wykładniczo wraz ze wzrostem rozmiaru instancji problemu. Pewnym sposobem na rozwiązanie tego problemu i otrzymanie rozwiązania w czasie wielomianowym jest wykorzystanie algorytmu przybliżonego. Nie ma gwarancji osiągnięcia optimum, ale czas obliczeń może zostać znacząco zredukowany.

Złożoność obliczeniowa różnych wariantów problemu SBH została omówiona w Rozdziale 5.1. Większość z nich należy do klasy problemów silnie NP-trudnych, co uzasadnia opracowywanie heurystyk. W niniejszym rozdziale przedstawiono: algorytm zachłanny (Rozdział 8.2), algorytm przeszukiwania tabu (Rozdział 8.3), algorytm kolonii mrówek (Rozdział 8.4) oraz wielopoziomowy algorytm kolonii mrówek (Rozdział 8.5).

8.1.1 Wykorzystywany model grafowy

Zaproponowane algorytmy rozwiązują problemy obliczeniowe sekwencjonowania DNA przez hybrydyzację z błędami dowolnego typu oraz częściową informacją o powtórzeniach typu "jeden i wiele" oraz "jeden, dwa i wiele". Algorytmy zostały zaprojektowane w taki sposób, że umożliwiają rozwiązanie zarówno problemów zakładających wykorzystanie klasycznych bibliotek oligonukleotydów jak i wariantów opartych na bibliotekach izotermicznych. Bazują one na następującym modelu grafowym.

Podstawę modelu dla danego problemu obliczeniowego SBH z błędami dowolnego typu i częściową informacją o powtórzeniach stanowią modele opisane w Rozdziale 6.2 oraz Rozdziale 6.3. Zostały one uproszczone w następujący sposób. Jako funkcję celu

przyjęto kryterium dla klasycznego sekwencjonowania przez hybrydyzację, tj. maksymalizowana jest łączna liczba wykorzystanych oligonukleotydów ze spektrum. Z perspektywy problemu grafowego przekłada się to na maksymalizację łącznej wartości odwiedzonych wierzchołków. W związku z uproszczeniem funkcji celu wartości β_{ij} nie są wykorzystywane, więc zostają pominięte przy tworzeniu grafu. Pozostałe aspekty modelów pozostają bez zmian.

Należy również zauważyć, że opisane w dalszej części algorytmy rozwiązują problem grafowy, ale ze względu na inspirację problemem biologicznym zostały one opisane przy użyciu terminów charakterystycznych dla sekwencjonowania DNA przez hybrydyzację. Dla uproszczenia opisów zamiast terminu *oligonukleotyd reprezentowany przez słowo* s_i wykorzystywana będzie zamiennie również skrócona forma *oligonukleotyd* s_i.

8.1.2 Opis danych wejściowych

Wszystkie opisane w dalszej części algorytmy wykorzystują takie same dane wejściowe. Składają się one ze zbioru $S(Q)$ reprezentującego multispektrum otrzymane w trakcie eksperymentu biochemicznego oraz długości n analizowanej sekwencji Q. Dla każdego słowa $s_i \in S(Q)$ określony jest również parametr m_i. Jego maksymalna wartość zależy od stosownego modelu informacji o powtórzeniach i jest równa 2 w przypadku wykorzystania modelu typu "jeden i wiele" lub jest ona równa 3 w przypadku modelu typu "jeden, dwa i wiele". Samo multispektrum może być zarówno multizbiorem zawierającym oligonukleotydy o jednakowej długości (l-mery) jak i oligonukleotydy otrzymane przy użyciu bibliotek izotermicznych o temperaturach T_L i $T_L + 2$℃. Do określenia kosztów łuków wykorzystywana jest długość oligonukleotydów oraz ich nałożenie, co powoduje, że rodzaj użytych bibliotek jest nieistotny. Opcjonalnie jako element danych wejściowych może zostać również wskazany pierwszy oligonukleotyd sekwencji Q.

8.2 Algorytm zachłanny

Początkowo rozwiązanie zawiera jedynie pierwszy oligonukleotyd. Jeżeli informacja o pierwszym oligonukleotydzie nie jest częścią instancji problemu, to można uruchomić algorytm wielokrotnie rozpoczynając budowę rozwiązania za każdym razem od innego oligonukleotydu i wybrać najlepsze otrzymane rozwiązanie.

Algorytm rozbudowuje iteracyjnie bieżące rozwiązanie poprzez dołączanie na jego końcu kolejnych oligonukleotydów. Proces ten kończy się, gdy dodanie kolejnego oligonukleotydu spowodowałoby przekroczenie podanej długości n rekonstruowanej sekwencji.

Kolejny oligonukleotyd s_i jest wybierany ze zbioru *dostępnych* elementów spektrum. Dla każdego oligonukleotydu zapamiętana jest aktualna liczba wystąpień w bieżącym rozwiązaniu. Jeżeli liczba ta osiągnie maksymalną wartość wynikającą z parametru m_i, to dany oligonukleotyd zostaje wykluczony z procesu wyboru następnego elementu rozwiązania.

Przy wyborze kolejnego oligonukleotydu do dołączenia obliczane są koszty dodania każdego z dostępnych elementów spektrum i wybierany jest ten o najniższym koszcie. Niech ostatnim oligonukleotydem dołączonym do bieżącego rozwiązania w kroku t będzie oligonukleotyd s_t, rozważany kandydat to s_i, a jego najlepszy następnik (dostępny oligonukleotyd o największym nałożeniu względem s_i) to s_j. Wtedy do oceny oligonukleotydu s_i wykorzystywany jest łączny koszt $c_{ti} + c_{ij}$.

Parametr m_i określa również minimalną liczbę wystąpień danego oligonukleotydu w rekonstruowanej sekwencji. Aby zrealizować to wymaganie obliczana jest *rezerwa*. Reprezentuje ona wzrost długości tworzonej sekwencji spowodowany dołączeniem na końcu bieżącego rozwiązania wszystkich tych oligonukleotydów, dla których minimalna liczba wystąpień nie została jeszcze osiągnięta. Przy liczeniu rezerwy zakłada się następującą kolejność dodawania takich oligonukleotydów. Pierwszy z nich jest wybierany losowo. Kolejność pozostałych determinuje nałożenie, tj. w danym momencie wybierany jest oligonukleotyd o najlepszym nałożeniu względem ostatnio dodanego. Rezerwa jest aktualizowana jedynie wtedy, gdy do bieżącego rozwiązania zostanie dołączony oligonukleotyd, którego minimalna liczba wystąpień jest większa od 0. Umożliwia to aktualizację rezerwy jedynie wtedy, kiedy faktycznie ulegnie ona zmianie.

W rzeczywistości proces dodawania kolejnych oligonukleotydów zostaje przerwany już wtedy, kiedy dołączenie kolejnego elementu spowodowałoby powstanie sekwencji dłuższej niż długość n pomniejszona o obliczoną rezerwę. Ostatnim krokiem algorytmu jest dołączenie tych oligonukleotydów, dla których minimalna liczba wystąpień określona przez dany model informacji o powtórzeniach nie została jeszcze osiągnięta. Dodawane są one w takiej samej kolejności jaka została określona przy liczeniu rezerwy.

8.3 Algorytm tabu

Algorytm przeszukiwania tabu jest metaheurystyką opartą na lokalnym przeszukiwaniu [33]. Jego idea została wykorzystana do opracowania algorytmu dla problemu klasycznego SBH z błędami dowolnego typu [10]. Algorytm ten został następnie rozbudowany o mechanizm wielokrotnego uruchamiania dla różnych rozwiązań początkowych, które są tworzone na podstawie kolekcji najlepszych rozwiązań znalezionych w trakcie dotychczasowych obliczeń [12]. Poniższy algorytm bazuje na tej rozszerzonej wersji. Nowy algorytm uwzględnia częściową informację o powtórzeniach. Wprowadzono również pewne dodatkowe usprawnienia, które zostały opisane w Rozdziale 8.3.1. Pozostałe elementy algorytmu zostały zaprezentowane wcześniej w [12].

Globalną funkcją oceny rozwiązań jest liczba zawartych oligonukleotydów z multispektrum w danym rozwiązaniu. Dany oligonukleotyd s_i może być wykorzystany w rozwiązaniu więcej niż raz, zgodnie z parametrem m_i. Każde wystąpienie danego oligonukleotydu zwiększa całkowitą ocenę rozwiązania. Celem jest maksymalizacja globalnej funkcji oceny przy jednoczesnym zapewnieniu, że zrekonstruowana sekwencja będzie nie dłuższa niż n.

Przeszukiwanie przestrzeni rozwiązań rozpoczyna się od rozwiązania początkowego, które uzyskuje się za pomocą algorytmu zachłannego opisanego w Rozdziale 8.2. Jeżeli informacja o pierwszym oligonukleotydzie nie została podana, to algorytm zachłanny jest uruchamiany wielokrotnie. Za każdym razem budowa rozwiązania rozpoczyna się od innego oligonukleotydu i wybierane jest najlepsze otrzymane rozwiązanie (wg globalnej funkcji oceny).

Bieżące rozwiązanie jest reprezentowane przez sekwencję oligonukleotydów. Parametr m_i determinuje zarówno minimalną (min_i) jak i maksymalną (max_i) liczbę wystąpień w rozwiązaniu oligonukleotydu s_i. Wartości te zależą od wykorzystywanego modelu częściowej informacji o powtórzeniach i zostały zaprezentowane w Tabeli 6.1. Niech $użycie_i$ oznacza aktualną liczbę wystąpień oligonukleotydu s_i w bieżącym rozwiązaniu. Wtedy dla każdego oligonukleotydu s_i zawsze spełniony jest następujący warunek: $min_i \leqslant użycie_i \leqslant max_i$. Dodatkowo na każdym etapie obliczeń sekwencja DNA odpowiadająca bieżącemu rozwiązaniu nie może być dłuższa niż n. W konsekwencji w trakcie

przeszukiwania przestrzeni rozwiązań rozpatrywane są jedynie takie ruchy, które nie naruszają powyższych ograniczeń. Są one nazywane *ruchami dopuszczalnymi* (ang. *feasible moves*). Gwarantuje to, że każde rozwiązanie wygenerowane w trakcie obliczeń jest rozwiązaniem akceptowalnym.

Wykorzystywane są trzy podstawowe typy ruchów: *wstawienie* (ruch polegający na wstawieniu oligonukleotydu do rozwiązania na pewnej pozycji w bieżącej sekwencji oligonukleotydów), *usunięcie* (ruch polegający na usunięciu pewnego oligonukleotydu z rozwiązania) oraz *przesunięcie* (ruch polegający na zmianie pozycji pewnego oligonukleotydu wewnątrz rozwiązania). Należy przy tym zauważyć, że dany oligonukleotyd może występować wielokrotnie w rozwiązaniu, więc wstawienie sprowadza się do dodania kolejnego wystąpienia danego oligonukleotydu. Analogicznie usunięcie polega na usunięciu tylko danego wystąpienia oligonukleotydu. Celem zwiększenia efektywności wprowadzono ruchy *klastrów*. Klaster jest taką grupą sąsiadujących ze sobą oligonukleotydów rozwiązania, że dla każdych dwóch kolejnych oligonukleotydów s_i i s_j wewnątrz klastra koszt $c_{ij} \leqslant 1$. Wykonanie ruchu może wpłynąć na liczbę/zawartość klastrów, więc lista klastrów musi być aktualizowania po każdym ruchu. Każdy klaster jest reprezentowany przez parę pozycji wewnątrz bieżącego rozwiązania. Pierwsza pozycja wskazuje początek klastra, a druga określa jego koniec. Podsumowując, lista możliwych ruchów składa się z: wstawienia oligonukleotydu, usunięcia oligonukleotydu lub klastra oraz przesunięcia oligonukleotydu lub klastra. Ruchy te są ograniczone przez następujące reguły:

- klaster może zostać przesunięty tylko wtedy, gdy nie powoduje to uszkodzenia innego klastra,
- klaster może zostać usunięty tylko wtedy, gdy dla każdego oligonukleotydu s_i wchodzącego w skład klastra $użycie_i > min_i$,
- oligonukleotyd s_i może zostać wstawiony tylko wtedy, gdy $użycie_i < max_i$,
- wstawienie oligonukleotydu nie może uszkodzić żadnego klastra,
- oligonukleotyd może zostać przesunięty tylko wtedy, gdy nie jest częścią żadnego klastra i jego przesunięcie nie spowoduje uszkodzenia żadnego klastra,
- oligonukleotyd s_i może zostać usunięty tylko wtedy, gdy $użycie_i > min_i$ oraz nie jest on częścią klastra lub ewentualnie stanowi jeden ze skrajnych oligonukleotydów w klastrze,
- żaden z ruchów nie może zmienić pozycji pierwszego oligonukleotydu sekwencji rozwiązania (jeżeli informacja o pierwszym oligonukleotydzie jest częścią danych wejściowych).

Powyższe ograniczenia zapewniają, że wykonywane będą wyłącznie ruchy dopuszczalne. Dodatkowo część z nich ma również na celu zredukowanie czasu obliczeń poprzez eliminację ruchów nie rokujących znalezieniem dobrego rozwiązania.

Ruchy są również ograniczone *listą tabu*. Wstawione, przesunięte i usunięte oligonukleotydy są zapamiętywane na tej liście przez pewną liczbę iteracji. Lista tabu jest wykorzystywana przy określaniu następnego ruchu. Jeżeli dany oligonukleotyd znajduje się na tej liście, to jego usunięcie, przesunięcie lub wstawienie jest zabronione. Taki oligonukleotyd może być jednak przesunięty lub usunięty będąc częścią przesuwanego lub usuwanego klastra. Jednakże takie przesunięcie/usunięcie w ramach ruchu klastra nie powoduje usunięcia oligonukleotydu z listy tabu. Celem jest eliminacja cyklicznego

wykonywania tych samych ruchów. Oligonukleotyd znajdujący się na liście tabu może być również usunięty, jeżeli nie ma innego dopuszczalnego ruchu. W takim przypadku wybierany jest taki element rozwiązania, który najczęściej się znajdował na liście tabu. Zapewnia to szerokie przeszukiwanie przestrzeni rozwiązań poprzez usunięcie z bieżącego rozwiązania tego oligonukleotydu, którego ruchy wykonywano najczęściej.

Celem globalnej funkcji oceny jest maksymalizacja liczby oligonukleotydów zwartych w rozwiązaniu. Funkcja ta oczywiście preferuje wstawienia. Wykorzystywanie wyłącznie tej funkcji przy wyborze kolejnego ruchu doprowadziłoby do tego, że przesunięcia i usunięcia byłyby wykonywane sporadycznie. W związku z tym wprowadzone zostało dodatkowe kryterium *kondensacji* (ang. *condensation*). Jest to iloraz liczby oligonukleotydów z multispektrum wchodzących w skład rozwiązania do liczby oligonukleotydów o długości l, które można wyróżnić w sekwencji reprezentowanej przez dane rozwiązanie. Preferowana jest jak najwyższa wartość kondensacji. Jeżeli istnieje kilka potencjalnych ruchów o takiej samej wartości kondensacji, to wybierany jest ruch prowadzący do użycia w rozwiązaniu większej liczby elementów z multispektrum. Maksymalizacja kondensacji prowadzi do konstrukcji rozwiązania składającego się z fragmentów dobrze nałożonych na siebie oligonukleotydów. Jednakże wykorzystanie tylko tego kryterium skutkowałoby otrzymaniem rozwiązania zbudowanego z jednego klastra reprezentującego sekwencję znacząco krótszą niż n. W związku z tym obie funkcje oceny (globalna i kondensacja) są wykorzystywane jednocześnie. Celem pierwszej jest dążenie do wzrostu liczby elementów multispektrum zawartych w bieżącym rozwiązaniu, a druga zapewnia dobre nakładanie się oligonukleotydów wewnątrz rozwiązania.

Przy wyborze jednego z pięciu opisanych powyżej typów ruchów (tj. wstawienie oligonukleotydu, przesunięcie oligonukleotydu lub klastra, usunięcie oligonukleotydu lub klastra) wykorzystywane jest kryterium kondensacji, a ruchy te nazywane są *ruchami kondensującymi* (ang. *condensing moves*). Jeżeli przez zadaną liczbę iteracji nie zostanie utworzone lepsze rozwiązanie (wg globalnej funkcji oceny) niż najlepsze rozwiązanie znalezione do tej pory, to wykonywane są *ruchy rozszerzające* (ang. *extending moves*), które są częścią *strategii dywersyfikacji*. Są to ruchy dopuszczalne wybierane na podstawie *częstotliwości użycia* (ang. *frequency-based memory*). Jest to struktura danych zawierająca informację o tym, ile razy dany oligonukleotyd wchodził w skład rozwiązań wygenerowanych do tej pory. Przykładowo, oligonukleotyd będący częścią każdego rozwiązania utworzonego do tej pory ma częstotliwość użycia równą liczbie wykonanych iteracji. Analogicznie, element multispektrum, który do tej pory nie był użyty w żadnym z rozwiązań, ma częstotliwość użycia równą 0.

Istnieją dwa typy ruchów rozszerzających: wstawienie oligonukleotydu oraz usunięcie oligonukleotydu. Preferowanym ruchem jest wstawienie. Do bieżącego rozwiązania wstawiany jest oligonukleotyd o najmniejszej częstotliwości użycia. Jeżeli nie istnieje dopuszczalne wstawienie rozszerzające, to wtedy oligonukleotyd o największej częstotliwości użycia jest usuwany. Jeżeli istnieje więcej niż jeden oligonukleotyd o minimalnej (w przypadku wstawienia) lub maksymalnej (w przypadku usunięcia) częstotliwości użycia, to wybierany jest losowo jeden z nich. Należy zauważyć, że ruchy rozszerzające są ruchami dopuszczalnymi, więc nie mogą naruszyć ograniczeń dotyczących minimalnej i maksymalnej liczby wystąpień danego oligonukleotydu w aktualnym rozwiązaniu. Wykonanie ruchów rozszerzających jest również zapamiętywane na liście tabu, aby zapobiec cyklicznemu wykonywaniu tych samych ruchów. Dodatkowo w miarę możliwości usunięcie rozszerzające nie powinno uszkodzić żadnego istniejącego klastra, więc o ile to możliwe usuwany jest oligonukleotyd o najwyższej częstotliwości użycia nie będący częścią żadnego z klastrów lub stanowiący początek/koniec pewnego klastra. Po wykonaniu ustalonej liczby ruchów rozszerzających algorytm ponownie realizuje ruchy w oparciu o

kryterium kondensacji.

Strategia dywersyfikacji została również zaimplementowana poprzez następującą procedurę *restartu* algorytmu. W ustalonej liczbie cykli naprzemiennego wykonywania ruchów kondensujących i rozszerzających tworzony jest *zbiór referencyjny*. Zawiera on pewną liczbę najlepszych rozwiązań znalezionych do tej pory. Zbiór ten jest wykorzystywany do utworzenia nowego rozwiązania początkowego w ramach procedury restartu. Przed jego użyciem najgorsze rozwiązanie ze zbioru jest usuwane, a w jego miejsce jest dodawane nowe rozwiązanie otrzymane za pomocą algorytmu zachłannego. W ogólności powinno ono również mieć dość wysoką notę z perspektywy globalnej funkcji oceny i powinno się znacząco różnić od pozostałych rozwiązań wchodzących w skład zbioru referencyjnego. Celem tego kroku jest dodatkowa dywersyfikacja.

Nowe rozwiązanie początkowe jest konstruowane przy użyciu algorytmu zachłannego opisanego w Rozdziale 8.2, przy czym w tym przypadku algorytm wykorzystuje ograniczony zbiór możliwych połączeń (łuków) pomiędzy oligonukleotydami. Brane są pod uwagę jedynie połączenia występujące w rozwiązaniach ze zbioru referencyjnego. Zezwala się na wyjątek od tej reguły tylko wtedy, gdy przy użyciu ograniczonego zbioru połączeń ostatni element w bieżącym rozwiązaniu nie ma żadnego dostępnego następnika. W takim przypadku wybierany jest jeden z dostępnych oligonukleotydów przy użyciu pełnego zbioru połączeń. Warto zauważyć, że graf skonstruowany na podstawie zbioru referencyjnego będzie zawierał znacząco mniej łuków w porównaniu do oryginalnego grafu wykorzystywanego przy budowie pierwszego rozwiązania początkowego.

Jeżeli pierwszy oligonukleotyd rekonstruowanej sekwencji nie został określony, to po kolei wszystkie oligonukleotydy ze spektrum są rozważane jako pierwsze. Jako rozwiązanie początkowe do kolejnego uruchomienia wybierane jest rozwiązanie o największej wartości liczonej przy użyciu globalnej funkcji oceny. Ostatnim krokiem procedury restartu jest reset większości wykorzystywanych zmiennych. Oczywiście najlepsze rozwiązanie znalezione w dotychczasowych obliczeniach nie jest usuwane. Nie są również resetowane wartości częstotliwości użycia, aby wzmocnić dywersyfikację i skupić przeszukiwanie na tych obszarach przestrzeni rozwiązań, które były mniej eksplorowane do tej pory.

Ostatecznie wynikiem algorytmu przeszukiwania tabu jest to rozwiązanie (znalezione w trakcie całości obliczeń), które zawiera największą liczbę oligonukleotydów ze spektrum. Kolejność oligonukleotydów w tym rozwiązaniu określa kolejność oligonukleotydów w sekwencji Q'.

8.3.1 Usprawnienia w porównaniu z poprzednią wersją algorytmu

Opisany powyżej algorytm przeszukiwania tabu bazuje na heurystyce zaproponowanej w [12]. Oprócz rozszerzenia polegającego na uwzględnieniu częściowej informacji o powtórzeniach wprowadzono pewne dodatkowe usprawnienia opisane poniżej.

1. Algorytm można zastosować zarówno do rozwiązania problemów zakładających wykorzystanie standardowych bibliotek oligonukleotydów o równej długości jak i problemów wykorzystujących biblioteki izotermiczne.

2. Ruch oligonukleotydu zawsze powoduje jego zapisanie na liście tabu. Rodzaj ruchu (wstawienie, usunięcie, przesunięcie) ani wykorzystywana aktualnie funkcja oceny (ruch rozszerzający lub kondensujący) nie mają znaczenia. Zapobiega to cyklicznemu wykonywaniu tych samych ruchów, np. usunięty oligonukleotyd nie jest wstawiany na tej samej pozycji w kolejnym ruchu.

3. Wstawienie kondensujące oligonukleotydu nie może uszkodzić istniejącego klastra, aby nie uszkodzić dobrego fragmentu bieżącego rozwiązania. Jednak nadal jest to możliwe w trakcie wstawiania rozszerzającego, bo celem tego ruchu jest dywersyfikacja rozwiązania.

4. Oligonukleotyd usunięty w ramach usuwania klastra nie jest usuwany z listy tabu. Celem jest eliminacja cyklicznego wykonywania tych samych ruchów.

5. Częstotliwość użycia nie jest resetowana w ramach procedury restartu. Celem jest wzmocnienie dywersyfikacji.

Wprowadzono również pewne usprawnienia w algorytmie zachłannym wykorzystywanym do konstrukcji rozwiązań początkowych.

1. Jeżeli dwa lub więcej oligonukleotydów ma taki sam koszt wynikający z nałożenia, to algorytm wybiera losowo jeden z nich. Celem jest wzmocnienie dywersyfikacji i otrzymanie innego rozwiązania przy kolejnych uruchomieniu algorytmu. Dzięki temu w ramach restartów do zbioru referencyjnego dodawane jest za każdym razem inne rozwiązanie.

2. Poprzednia wersja algorytmu wykorzystuje albo oryginalny graf albo zredukowany graf utworzony na podstawie zbioru referencyjnego (nigdy oba równocześnie). W przypadku użycia zredukowanego grafu, gdy ostatni element bieżącego rozwiązania nie ma dostępnego następnika, to dołączany jest pierwszy dostępny oligonukleotyd. Usprawniony algorytm zachłanny wykorzystuje w takim przypadku oryginalny graf do ustalenia najlepszego dostępnego następnika. Prowadzi to do wzrostu jakości rozwiązań początkowych.

Wydajność opisanego powyżej usprawnionego algorytmu przeszukiwania tabu została zweryfikowana za pomocą narzędzia zwanego *profiler*, które umożliwia określenie czasu wykonania poszczególnych fragmentów algorytmu (funkcji programu). Na podstawie wyników wprowadzono również następujące optymalizacje.

1. Poprzednia wersja określa najlepsze możliwe wstawienie rozszerzające oligonukleotydu w następujący sposób. W pierwszej kolejności weryfikowana jest dopuszczalność wstawienia rozszerzającego każdego oligonukleotydu, który nie jest wykorzystany w rozwiązaniu bieżącym. W najgorszym przypadku, gdy taki ruch dla danego oligonukleotydu nie istnieje, wymaga to obliczenia kosztu wstawienia danego oligonukleotydu na każdej możliwej pozycji aktualnego rozwiązania. W praktyce weryfikacja kończy się, gdy zostanie znalezione pierwsze dopuszczalne wstawienie rozszerzające danego oligonukleotydu. Następnie wśród tych oligonukleotydów, dla których znaleziono dopuszczalne wstawienie, wybierany jest oligonukleotyd o najmniejszej częstotliwości użycia. Ostatecznie wyznaczana jest dla niego pozycja wstawienia o najniższym koszcie. Zoptymalizowane podejście wykorzystywane przez nową wersję algorytmu jest następujące. Faza weryfikacji istnienia dopuszczalnego wstawienia rozszerzającego dla poszczególnych oligonukleotydów zostaje pominięta. Dla danego oligonukleotydu od razu określana jest pozycja wstawienia o najniższym koszcie, przy czym jest ona wyznaczana tylko wtedy, gdy dany oligonukleotyd nie ma wyższej częstotliwości użycia, niż aktualny najlepszy kandydat na wstawienie rozszerzające. Umożliwia to zredukowanie łącznej liczby obliczeń kosztu wstawienia oligonukleotydów na poszczególnych pozycjach.

2. Lista tabu jest wspierana przez dodatkową tablicę o wartościach logicznych PRAWDA lub FAŁSZ. Zawiera ona informację, czy dany oligonukleotyd znajduje się na liście tabu czy nie. Dzięki temu nie ma potrzeby iterowania po liście za każdym razem, kiedy trzeba to sprawdzić.

3. Najlepszym możliwym ruchem kondensującym jest wstawienie oligonukleotydu o zerowym koszcie (nie prowadzące do wydłużenia sekwencji reprezentowanej przez rozwiązanie). W trakcie poszukiwania kolejnego ruchu kondensującego w pierwszej kolejności weryfikowane są wstawienia oligonukleotydów. Jeżeli znalezione zostanie dopuszczalne wstawienie o zerowym koszcie, to jest ono wykonywane bez oceny potencjalnych ruchów kondensujących innego rodzaju.

4. Po wykonaniu każdego ruchu konieczna jest aktualizacja listy klastrów. Poprzednia wersja algorytmu usuwała z listy wszystkie określone wcześniej klastry i tworzyła nową listę od podstaw. Nowy algorytm wykorzystuje istniejącą listę i w razie potrzeby wprowadza odpowiednie aktualizacje. Wykonywane ną następujące operacje na liście klastrów:

   - usunięcie z listy klastra usuniętego z rozwiązania,
   - aktualizacja pozycji początkowej i końcowej istniejących klastrów,
   - połączenie dwóch klastrów w jeden klaster,
   - rozszerzenie klastra (jeżeli dodano nowy element tuż przed/za klastrem),
   - skrócenie klastra (jeżeli pierwszy/ostatni element klastra został usunięty),
   - utworzenie nowego klastra (jeżeli dwa kolejne oligonukleotydy rozwiązania nakładają się na siebie z kosztem co najwyżej 1, ale żaden z nich nie jest częścią innego klastra),
   - usunięcie z listy klastra, który składał się z dwóch elementów, a jeden z tych oligonukleotydów został usunięty lub składał się z trzech elementów i usunięto ten środkowy,
   - podział jednego klastra na dwa klastry (jeżeli klaster zawierał więcej niż trzy elementy i usunięto oligonukleotyd wewnątrz klastra).

8.4 Algorytm kolonii mrówek

Zaprezentowany poniżej algorytm kolonii mrówek (ACO) jest rozszerzeniem algorytmu przedstawionego w [20], który został opracowany dla klasycznego sekwencjonowania DNA przez hybrydyzację z błędami dowolnego rodzaju. Oprócz uwzględniania częściowej informacji o powtórzeniach zaproponowany poniżej algorytm może być wykorzystany zarówno do rozwiązywania problemów zdefiniowanych dla klasycznych bibliotek oligonukleotydów o równej długości jak i problemów zdefiniowanych dla bibliotek izotermicznych (zmieniono sposób obliczania oceny nałożenia - szczegóły w Rozdziale 8.4.1). Pozostałe elementy algorytmu pochodzą z [20].

Algorytm kolonii mrówek to probabilistyczne, iteracyjne poszukiwanie rozwiązania dla problemu sprowadzającego się do znalezienia pewnej ścieżki w grafie. W każdej iteracji konstruowane są rozwiązania dwojakiego typu: generowane *od początku* i generowane *od końca*. Rozwiązanie generowane *od początku* to rozwiązanie rozpoczynające się od pierwszego wierzchołka ścieżki, która jest rozbudowywana przez dodawanie na końcu kolejnych odwiedzanych wierzchołków. Rozwiązanie generowane *od końca* to rozwiązanie

rozpoczynające się od ostatniego wierzchołka ścieżki, która jest rozbudowywana przez dodawanie na początku kolejnych odwiedzanych wierzchołków.

Probabilistyczne tworzenie rozwiązań bazuje na tak zwanym *modelu feromonów* F, który jest zbiorem liczb kodujących wiedzę zebraną w trakcie dotychczasowych obliczeń. Zawiera on wartość feromonu τ_{ij} dla każdej pary oligonukleotydów $s_i, s_j \in S(Q)$. Im większa wartość feromonu τ_{ij} tym bardziej pożądane jest występowanie oligonukleotydu s_j bezpośrednio po oligonukleotydzie s_i w konstruowanych rozwiązaniach. Dodatkowo F zawiera wartości τ_{0i} oraz τ_{i0} dla każdego $s_i \in S(Q)$ odzwierciedlające jak bardzo oczekiwane jest, że dany oligonukleotyd jest odpowiednio pierwszym i ostatnim elementem rozwiązania.

Aktualizacja modelu feromonów odbywa się na końcu każdej iteracji przy użyciu najlepszych znalezionych rozwiązań (szczegóły opisano w Rozdziale 8.4.2). Po aktualizacji następuje ocena, czy zaktualizowany model feromonów nie prowadzi do *zbieżności* algorytmu, tj. do konstrukcji zbyt podobnych rozwiązań. Jeżeli algorytm osiągnie zbieżność, to następuje jego restart, w ramach którego wartości modelu feromonów są ustawiane do wartości początkowych. Warto również dodać, że wartości modelu feromonów są ograniczone zakresem $[\tau_{min}, \tau_{max}]$, więc opisywane podejście to MMAS (Max-Min Ant System).

Algorytm wykorzystuje również następujące struktury danych:

- *najlepsze rozwiązanie iteracji* P_{ib}: najlepsze rozwiązanie wygenerowane w bieżącej iteracji,
- *najlepsze rozwiązanie od restartu* P_{rb}: najlepsze rozwiązanie wygenerowane od czasu restartu algorytmu,
- *najlepsze rozwiązanie do tej pory* P_{bs}: najlepsze rozwiązanie wygenerowane od początku uruchomienia algorytmu,
- *współczynnik zbieżności* $\lambda, 0 \leqslant \lambda \leqslant 1$: miara zbieżności algorytmu,
- binarna zmienna *zbieżny*: ustawiana wartością PRAWDA, gdy algorytm osiągnie zbieżność.

Na samym początku inicjowane są wartości poszczególnych zmiennych, w szczególności $\lambda = 0$, a wartości feromonów są równe 0,5. W każdej iteracji konstruowanych jest n_f rozwiązań *od początku* oraz n_b rozwiązań *od końca*. Najlepsze z nich jest zapamiętywane jako P_{ib}. Dodatkowo, jeżeli rozwiązania P_{rb} lub P_{bs} są puste lub gorsze niż rozwiązanie P_{ib}, to są one aktualizowane na podstawie rozwiązania P_{ib}.

Do oceny rozwiązań stosowana jest następująca funkcja. Preferowane jest rozwiązanie zawierające większą liczbę oligonukleotydów (ścieżka zawierająca większą liczbą wierzchołków). Ponadto, jeżeli dwa rozwiązania zwierają tą samą liczbę oligonukleotydów, to wyżej oceniane jest to, które reprezentuje krótszą sekwencję DNA (ścieżka o niższym koszcie). Taka preferencja krótszej sekwencji zwiększa szansę późniejszej rozbudowy rozwiązania przy nie przekroczeniu oczekiwanej długości sekwencji n.

W kolejnym kroku iteracji aktualizowany jest model feromonów F i *współczynnik zbieżności*. Jeżeli *współczynnik zbieżności* przekroczy 0,9999, a zmienna *zbieżność* ma wartość FAŁSZ, to zmienna *zbieżność* jest ustawiana na PRAWDA i realizowana jest kolejna iteracja. Jeżeli *współczynnik zbieżności* przekroczy 0,9999, a zmienna *zbieżność* ma wartość PRAWDA, to następuje restart algorytmu. Wartości modelu feromonów F ustawiane są ponownie na 0,5. Rozwiązanie P_{rb} jest ustawiane jako puste, a zmienna

zbieżność ma wartość FAŁSZ.

Algorytm kończy obliczenia, jeżeli przez zadaną liczbę iteracji nie zostanie znalezione lepsze rozwiązanie niż rozwiązanie P_{bs}. Wtedy *najlepsze rozwiązanie do tej pory* P_{bs} zostaje przekazane jako ostateczny wynik.

8.4.1 Konstrukcja rozwiązań

Konstrukcja rozwiązań *od początku* odbywa się z wykorzystaniem algorytmu zachłannego wykorzystującego model feromonów. Stanowi on połączenie algorytmu opisanego w Rozdziale 8.2 oraz algorytmu zaprezentowanego w [20]. Algorytm rozpoczyna od początkowego oligonukleotydu i sukcesywnie rozbudowuje rozwiązanie przez dodawanie na końcu kolejnych oligonukleotydów. Dla każdego oligonukleotydu zapamiętywana jest liczba jego powtórzeń w bieżącym rozwiązaniu. Jeżeli wartość ta osiągnie maksimum wynikające z parametru m_i oraz stosowanego modelu częściowej informacji o powtórzeniach (konkretne wartości prezentuje Tabela 6.1), to dany oligonukleotyd jest pomijany w trakcie wyboru kolejnego elementu rozwiązania. W dalszym opisie $\hat{S} \subseteq S(Q)$ oznacza zbiór słów reprezentujących te oligonukleotydy, dla których maksymalna liczba wystąpień nie została jeszcze osiągnięta i mogą zostać dodane do bieżącego rozwiązania. Proces kończy się, gdy dodanie kolejnego oligonukleotydu spowodowałoby utworzenie rozwiązania reprezentującego sekwencję DNA dłuższą od oczekiwanej długości n.

Procedura wyboru kolejnego oligonukleotydu w poszczególnych iteracjach przebiega następująco. Niech s_t oznacza ostatnio dodany oligonukleotyd do bieżącego rozwiązania na koniec kroku t. Dla każdego oligonukleotydu $s_i \in \hat{S}$ obliczana jest jego *wartość* $\mu_{ti} = \tau_{ti} \cdot \eta_{ti}$, gdzie $\eta_{ti} = (|s_i| - c_{ti})/|s_i|$ jest *oceną nałożenia*. Z definicji *ocena nałożenia* $\eta_{ti} \in [0, 1]$ i przyjmuje ona tym wyższe wartości im lepsze jest nałożenie oligonukleotydów s_t i s_i.

Jako następny oligonukleotyd do dołączenia wybierany jest z pewnym prawdopodobieństwem $q \in [0, 1)$ oligonukleotyd $s_i \in \hat{S}$ o najwyższej wartości μ_{ti}. Parametr q zwany jest *współczynnikiem determinizmu*. Kolejny element może zostać również wybrany z prawdopodobieństwem $1 - q$ w sposób losowy. Przygotowywana jest wtedy *ograniczona lista kandydatów* S^{rcl} o predefiniowanej liczności r zawierająca te oligonukleotydy, dla których $\mu_{tj} \geqslant \mu_{ti}$, gdzie $s_i, s_j \in \hat{S}, s_j \in S^{rcl}$. Następny oligonukleotyd jest wybierany na zasadzie ruletki, gdzie prawdopodobieństwo wyboru danego oligonukleotydu s_j z *ograniczonej listy kandydatów* S^{rcl} obliczane jest w następujący sposób.

$$p_{tj} = \frac{\mu_{tj}}{\sum_{j' \in S^{rcl}} \mu_{tj'}} \tag{8.1}$$

Jeżeli pierwszy oligonukleotyd rekonstruowanej sekwencji nie został określony w danych wejściowych, to jest on wybierany w analogiczny sposób, przy czym *wartość* oligonukleotydu s_i obliczana jest wtedy jako $\mu_{0i} = \tau_{0i} \cdot \eta_{0i}$, a *ocena nałożenia* obliczana jest następująco.

$$\eta_{0i} = \frac{|s_i| - o_{pre(i)\,i} + o_{i\,suc(i)}}{2|s_i|} \tag{8.2}$$

Wartości $pre(i)$ i $suc(i)$ to odpowiednio najlepszy poprzednik i najlepszy następnik oligonukleotydu s_i, które są wyznaczane w następujący sposób.

$$pre(i) = \arg\max_j \{o_{ji} | i, j \in \hat{S}, i \neq j\} \tag{8.3}$$

$$suc(i) = \arg\max_j \{o_{ij} | i, j \in \hat{S}, i \neq j\} \tag{8.4}$$

Ocena nałożenia pierwszego oligonukleotydu preferuje takie oligonukleotydy, które mają bardzo złego (małe nałożenie) najlepszego poprzednika i jednocześnie bardzo dobrego (duże nałożenie) najlepszego następnika. Najprawdopodobniej to właśnie taki oligonukleotyd będzie stanowił początek rekonstruowanej sekwencji.

Parametr m_i określa również minimalną liczbę wystąpień danego oligonukleotydu w rekonstruowanej sekwencji. Aby zrealizować to wymaganie obliczana jest *rezerwa*. Reprezentuje ona wzrost długości tworzonej sekwencji spowodowany dołączeniem na końcu bieżącego rozwiązania wszystkich tych oligonukleotydów, dla których minimalna liczba wystąpień nie została jeszcze osiągnięta. Przy liczeniu rezerwy zakłada się następującą kolejność dodawania takich oligonukleotydów. Pierwszy z nich jest wybierany losowo. Kolejność pozostałych determinuje nałożenie, tj. w danym momencie wybierany jest oligonukleotyd o najlepszym nałożeniu względem ostatnio dodanego. Rezerwa jest aktualizowana jedynie wtedy, gdy do bieżącego rozwiązania zostanie dołączony oligonukleotyd, którego minimalna liczba wystąpień jest większa od 0. Umożliwia to aktualizację rezerwy jedynie wtedy, kiedy faktycznie ulegnie ona zmianie.

W rzeczywistości proces dodawania kolejnych oligonukleotydów zostaje przerwany już wtedy, kiedy dołączenie kolejnego elementu spowodowałoby powstanie sekwencji dłuższej niż długość n pomniejszona o obliczoną rezerwę. Ostatnim krokiem jest dołączenie tych oligonukleotydów, dla których minimalna liczba wystąpień określona przez dany model informacji o powtórzeniach nie została jeszcze osiągnięta. Dodawane są one w takiej samej kolejności jaka została określona przy liczeniu rezerwy.

Konstrukcja rozwiązania *od końca* w ogólności przebiega analogicznie jak tworzenie rozwiązania *od początku*. Podstawową różnicą jest rozpoczęcie budowy rozwiązania od ostatniego oligonukleotydu i sukcesywne dodawanie kolejnych elementów ze spektrum na początku bieżącego rozwiązania. W konsekwencji bieżące rozwiązanie ma postać $\{s_t, s_{t-1}, \ldots, s_2, s_1\}$, a *wartość* kolejnego elementu liczona jest dla μ_{it} (w przeciwieństwie do μ_{ti} dla rozwiązania *od początku*). Ponadto, jeżeli określony został pierwszy oligonukleotyd analizowanej sekwencji, to *rezerwa* dodatkowo bierze pod uwagę koszt jego dołączenia na samym początku rozwiązania. Po dodaniu wielokrotnych oligonukleotydów na sam początek rozwiązania wstawiany jest jeszcze oligonukleotyd wskazany jako początkowy.

8.4.2 Aktualizacja modelu feromonów

Tabela 8.1: Wagi rozwiązań κ_{ib}, κ_{rb} i κ_{bs} w zależności od współczynnika λ i zmiennej *zbieżny*.

	zbieżny = FAŁSZ				*zbieżny* = PRAWDA
	$\lambda < 0,7$	$\lambda \in [0,7;\ 0,9)$	$\lambda \in [0,9;\ 0,95)$	$\lambda \geqslant 0,95$	
κ_{ib}	1	2/3	1/3	0	0
κ_{rb}	0	1/3	2/3	1	0
κ_{bs}	0	0	0	0	1

Aktualizacja wartości modelu feromonów odbywa się na podstawie ważonej kombinacji rozwiązań P_{ib}, P_{rb} i P_{bs}. Wagi poszczególnych rozwiązań, odpowiednio κ_{ib}, κ_{rb} i κ_{bs}, zależą od *współczynnika zbieżności* λ oraz binarnej zmiennej *zbieżny*. Ich konkretne wartości zostały zaprezentowane w Tabeli **??**. Wartość feromonu jest aktualizowana zgodnie

z poniższym wzorem.

$$\tau_{ij} = \tau_{ij} + \rho \cdot (\omega_{ij} - \tau_{ij}), \forall \tau_{ij} \in F \tag{8.5}$$

gdzie $\rho \in (0, 1]$ jest *współczynnikiem uczenia*, a parametr ω_{ij} jest obliczany następująco.

$$\omega_{ij} = (\kappa_{ib} \cdot \delta_{ij}(P_{ib})) + (\kappa_{rb} \cdot \delta_{ij}(P_{rb})) + (\kappa_{bs} \cdot \delta_{ij}(P_{bs})) \tag{8.6}$$

Jeżeli $i \neq 0$ i $j \neq 0$, to dla rozwiązania P wynikiem funkcji $\delta_{ij}(P)$ jest 1, jeżeli oligonukleotyd s_j jest bezpośrednim następnikiem oligonukleotydu s_i w P lub 0 w przeciwnym razie. Jeżeli $i = 0$, to wynikiem $\delta_{0j}(P)$ jest 1, gdy oligonukleotyd s_j jest pierwszym elementem P lub 0 w przeciwnym razie. Analogicznie, jeżeli $j = 0$, to wynikiem $\delta_{i0}(P)$ jest 1, gdy oligonukleotyd s_i jest ostatnim elementem P lub 0 w przeciwnym razie.

8.4.3 Obliczanie współczynnika zbieżności

Współczynnik zbieżności jest obliczany na podstawie wartości *modelu feromonów* F wg wzoru (8.7). Kiedy wartości feromonów są inicjalizowane lub resetowane wartością 0,5, to *współczynnik zbieżności* λ jest równy 0. Gdy algorytm jest zupełnie zbieżny, tj. wszystkie wartości feromonów są równe τ_{min} lub τ_{max}, to $\lambda = 1$. W każdym innym przypadku $\lambda \in (0, 1)$.

$$\lambda = 2 \left(\left(\frac{\Sigma_{\tau_{ij} \in F} \max(\tau_{max} - \tau_{ij}, \tau_{ij} - \tau_{min})}{|F| \cdot (\tau_{max} - \tau_{min})} \right) - 0,5 \right) \tag{8.7}$$

8.5 Wielopoziomowy algorytm kolonii mrówek

Pomysł wielopoziomowego podejścia do rozwiązywania problemów kombinatorycznych został przedstawiony w [65]. Został on wykorzystany do opracowania wielopoziomowej wersji algorytmu ACO (ang. *Multi-Level Ant Colony Optimization*, ML-ACO) dla klasycznego sekwencjonowania DNA przez hybrydyzację z błędami dowolnego typu [20]. Algorytm zaprezentowany w niniejszym rozdziale jest jego rozszerzoną wersją umożliwiającą wykorzystanie częściowej informacji o powtórzeniach. Dodatkowo pozwala on na rozwiązywanie problemów SBH zdefiniowanych zarówno dla klasycznych bibliotek oligonukleotydów jak i bibliotek izotermicznych. Pozostałe różnice zostały omówione w dalszej części niniejszego rozdziału.

Wielopoziomowe podejście do rozwiązywania problemów sprowadza się do realizacji następujących kroków. Oryginalna instancja jest rekurencyjnie upraszczana aż do momentu spełnienia określonych warunków stopu. W wyniku powstaje hierarchia uproszczonych instancji. Instancja następnego poziomu jest zawsze mniejsza niż instancja poprzedniego poziomu. Rozwiązanie początkowe jest tworzone dla najbardziej uproszczonej instancji (najwyższego poziomu), a następnie jest ono udoskonalane. Otrzymane rozwiązanie jest później iteracyjnie transformowane do rozwiązania dla instancji poprzedniego poziomu i jest ono udoskonalane na poprzednim poziomie. Ostatecznie rozwiązanie jest transformowane do rozwiązania dla instancji oryginalnego problemu i również następuje jego udoskonalanie.

Powyższa idea została wykorzystana w następujący sposób. W pierwszej kolejności tworzona jest hierarchia uproszczonych instancji poprzez łączenie nakładających się oligonukleotydów w dłuższe sekwencje. Im wyższy poziom upraszczania, tym mniejsze jest wymagane nałożenie oligonukleotydów, aby można było je połączyć (szczegóły w Rozdziale 8.5.1). Rozwiązanie początkowe dla instancji najwyższego poziomu jest tworzone

z wykorzystaniem algorytmu ACO opisanego w Rozdziale 8.4. Na tym poziomie dalsze udoskonalanie rozwiązania nie jest już wykonywane. Następnie wykonywany jest iteracyjny proces transformacji i udoskonalania rozwiązania. Rozwiązanie P^i dla poziomu i jest transformowane do rozwiązania P^{i-1} dla poziomu $i-1$ (szczegóły w Rozdziale 8.5.2) i jest ono udoskonalane przy użyciu tego samego algorytmu ACO (opisanego w Rozdziale 8.4), przy czym początkowo najlepsze rozwiązanie znalezione do tej pory P_{bs} nie jest puste i jest równe P^{i-1}. Iteracyjny proces transformacji i udoskonalania rozwiązania zatrzymuje się po zakończeniu udoskonalania rozwiązania P^0, które reprezentuje rozwiązanie dla oryginalnego problemu. Wynikiem wielopoziomowego algorytmu kolonii mrówek jest ostatecznie udoskonalone rozwiązanie P^0.

Wielopoziomowy algorytm ACO opisany w [20] ogranicza czas obliczeń na danym poziomie do pewnej ustalonej wartości i stanowi to dodatkowe kryterium stopu dla ACO. Jednakże zrealizowana implementacja rozszerzonego algorytmu opisywanego w niniejszym rozdziale umożliwia rekonstrukcję najdłuższych testowanych sekwencji DNA o długości 509 nukleotydów w czasie ok. 2-3 sekund (spektra uzyskane przy użyciu klasycznych bibliotek oligonukleotydów) bez żadnych dodatkowych ograniczeń czasowych na poszczególnych poziomach. W związku z tym algorytm został uproszczony i ograniczenie maksymalnego czasu obliczeń na danym poziomie nie zostało wykorzystane.

Podprocedura upraszczania instancji zaprezentowana w [20] nie zapamiętuje żadnej informacji o łączonych oligonukleotydach, która mogłaby następnie być wykorzystana do transformacji rozwiązania do rozwiązania poprzedniego poziomu. Opis algorytmu nie precyzuje również dokładnie w jaki sposób taką transformację należy przeprowadzić. Uniemożliwia to szczegółowe porównanie obu algorytmów.

8.5.1 Upraszczanie instancji

Podstawą podejścia wielopoziomowego jest stopniowe upraszczanie instancji. Celem tego procesu jest wygenerowanie hierarchii coraz to mniejszych instancji problemu. Dana instancja jest stopniowo upraszczania poprzez łączenie nakładających się oligonukleotydów w dłuższe sekwencje (tzw. *połączone oligonukleotydy*), które najprawdopodobniej są fragmentami analizowanej sekwencji DNA. Rozmiar spektrum jest stopniowo redukowany co jednoczenie prowadzi do redukcji tworzonego grafu, bo wierzchołki reprezentujące kilka łączonych oligonukleotydów są zastępowane przez wierzchołek reprezentujący połączoną sekwencję (połączony oligonukleotyd).

Niech zbiór $\bar{S}(Q)$ reprezentuje spektrum zawierające połączone oligonukleotydy, $\bar{x} \in \bar{S}(Q)$ i $\bar{y} \in \bar{S}(Q)$ oznaczają sekwencje słów odpowiadające sekwencjom połączonych oligonukleotydów, $s_i \in S(Q)$ będzie ostatnim słowem w sekwencji $\bar{x}$, a $s_j \in S(Q)$ będzie pierwszym słowem w sekwencji $\bar{y}$. Koszt $\bar{c}_{xy}$ łuku z wierzchołka reprezentującego połączony oligonukleotyd $\bar{x}$ do wierzchołka reprezentującego połączony oligonukleotyd $\bar{y}$ w uproszczonym grafie jest równy c_{ij}, tj. jest równy kosztowi łuku z wierzchołka reprezentującego s_i do wierzchołka reprezentującego s_j w oryginalnym grafie. Niech $\bar{x} = \{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$, a $len(\bar{x})$ oznacza długość sekwencji DNA reprezentowanej przez połączony oligonukleotyd $\bar{x}$.

$$len(\bar{x}) = |s_{i_1}| + \sum_{k=2}^{k=t} c_{i_{k-1} i_k} \tag{8.8}$$

Długość sekwencji DNA powstałej z połączenia oligonukleotydów zależy od tego, jak bardzo kolejno łączone oligonukleotydy nakładają się na siebie.

Należy również zdefiniować termin najlepszego następnika i najlepszego poprzednika dla $\bar{x}$.

$$pre(\bar{x}) = \arg\min_{\bar{y}}\{\bar{c}_{yx}|\bar{x}, \bar{y} \in \bar{S}(Q), \bar{x} \neq \bar{y}\}, \tag{8.9}$$

$$suc(\bar{x}) = \arg\min_{\bar{y}}\{\bar{c}_{xy}|\bar{x}, \bar{y} \in \bar{S}(Q), \bar{x} \neq \bar{y}\}. \tag{8.10}$$

Niech $\bar{S}_{pre(\bar{x})}$ i $\bar{S}_{suc(\bar{x})}$ oznaczają odpowiednio zbiór najlepszych poprzedników i zbiór najlepszych następników dla $\bar{x}$. Zbiory te są zdefiniowane następująco.

$$\bar{S}_{pre(\bar{x})} = \{\bar{y} \in \bar{S}(Q)|\bar{c}_{yx} = \bar{c}_{pre(\bar{x})\,x}\}, \tag{8.11}$$

$$\bar{S}_{suc(\bar{x})} = \{\bar{y} \in \bar{S}(Q)|\bar{c}_{xy} = \bar{c}_{x\,suc(\bar{x})}\}. \tag{8.12}$$

Podprocedurę upraszczania instancji prezentuje Algorytm 5. Początkowo każda sekwencja połączonych oligonukleotydów zawiera jeden oligonukleotyd z oryginalnego spektrum. Dwie sekwencje oligonukleotydów reprezentowane przez połączone oligonukleotydy $\bar{x}$ i $\bar{y}$ są ze sobą łączone, jeżeli spełnione są następujące warunki:

- $\bar{x}$ jest unikalnym najlepszym poprzednikiem $\bar{y}$, tj. $pre(\bar{y}) = \bar{x}$ i $|\bar{S}_{pre(\bar{y})}| = 1$,
- $\bar{y}$ jest unikalnym najlepszym następnikiem $\bar{x}$, tj. $suc(\bar{x}) = \bar{y}$ i $|\bar{S}_{suc(\bar{x})}| = 1$,
- $\bar{x}$ i $\bar{y}$ występują raz w analizowanej sekwencji Q, tj. $m_{\bar{x}} = 1$ and $m_{\bar{y}} = 1$,
- $\bar{x}$ i $\bar{y}$ nie reprezentują sekwencji zawierającej pierwszy oligonukleotyd w Q (jeżeli został podany w danych wejściowych).

Dodatkowo przy łączeniu $\bar{x}$ i $\bar{y}$ brany jest pod uwagę koszt $\bar{c}_{xy}$. Rekurencyjny proces upraszczania wymaga początkowo kosztu równego 0 i jest on zwiększany o jeden przy każdym kolejnym poziomie. Proces upraszczania jest zatrzymywany, gdy osiągnięty zostanie maksymalny koszt łuku z oryginalnego grafu lub gdyby w wyniku połączenia oligonukleotydów powstała sekwencja DNA dłuższa niż n.

Jeżeli $\bar{x}$ i $\bar{y}$ mogą zostać połączone, to wykonywane są dwie kluczowe akcje. Po pierwsze sekwencja słów z $\bar{y}$ jest dołączana na końcu sekwencji $\bar{x}$. Drugim istotnym krokiem jest zapamiętanie informacji o połączonych sekwencjach w zmiennej $skład_{\bar{x}}^{poziom}$. Zostanie ona wykorzystana później przy transformacji rozwiązania dla uproszczonej instancji do rozwiązania poprzedniego poziomu (szczegóły w Rozdziale 8.5.2). Należy zauważyć, że zmienne *skład* są zapamiętywane niezależnie dla każdego poziomu.

Jeżeli dla danego kosztu jakiekolwiek dwie sekwencje oligonukleotydów zostały połączone ($zmiana$ = PRAWDA), to tworzona jest instancja kolejnego poziomu. Jeżeli w wyniku połączenia powstała sekwencja DNA o długości przekraczającej n ($stop$ = PRAWDA), to łączenie oligonukleotydów jest przerywane, a procedura upraszania instancji się kończy. Proces kończy się najpóźniej po połączeniu oligonukleotydów dla maksymalnego kosztu nałożenia oligonukleotydów pomniejszonego o 1.

8.5.2 Transformacja rozwiązania

Wynikiem Algorytmu 5 jest hierarchia uproszczonych instancji oraz zbiór zmiennych *skład*. Zmienne te przechowują informację o tym jakie sekwencje słów zostały ze sobą połączone celem uproszczenia instancji na danym poziomie. Są one wykorzystywane przy transformacji rozwiązania i-tego poziomu P^i do rozwiązania poprzedniego poziomu P^{i-1}.

Algorytm 5 Wielopoziomowy algorytm ACO - procedura upraszczania instancji.

Instancja: zbiór $S(Q)$ zawierający słowa zbudowane nad alfabetem {A, C, T, G} reprezentujące oligonukleotydy ze spektrum klasycznego lub izotermicznego, długość n sekwencji Q, parametr m_i dla każdego słowa $s_i \in S(Q)$, słowo $s_f \in S(Q)$ reprezentujące pierwszy oligonukleotyd sekwencji Q (opcjonalnie)

1: $instancja^0 := (S(Q), m_i$ dla każdego $s_i \in S(Q), n)$
2: $\bar{S}(Q) := \{\{s_i\} | s_i \in S(Q))\}; m_{\bar{x}} = m_i : \bar{x} = \{s_i\}; stop :=$ FAŁSZ; $poziom := 1$
3: **dla** $koszt = 0$ **do** $koszt = max(c_{ij}) - 1$ **wykonaj**
4: $\quad zmiana :=$ FAŁSZ
5: $\quad skład_{\bar{x}}^{poziom} := \{\bar{x}\}$ dla każdego $\bar{x} \in \bar{S}(Q)$
6: $\quad$ **dopóki** $\exists \bar{x}, \bar{y} \in \bar{S}(Q) : \bar{c}_{xy} = koszt \ \& \ suc(\bar{x}) = \bar{y} \ \& \ pre(\bar{y}) = \bar{x} \ \& \ |\bar{S}_{suc(\bar{x})}| = 1 \ \& \ |\bar{S}_{pre(\bar{y})}| = 1 \ \& \ m_{\bar{x}} = 1 \ \& \ m_{\bar{y}} = 1 \ \& \ \bar{x} \neq \{s_f\} \ \& \ \bar{y} \neq \{s_f\} \ \& \ stop =$ FAŁSZ **wykonaj**
7: $\quad\quad zmiana :=$ PRAWDA
8: $\quad\quad \bar{x} := \bar{x}|\bar{y}$
9: $\quad\quad skład_{\bar{x}}^{poziom} := skład_{\bar{x}}^{poziom} | skład_{\bar{y}}^{poziom}$
10: $\quad\quad \bar{S}(Q) := \bar{S}(Q) \setminus \{\bar{y}\}$
11: $\quad\quad$ **jeżeli** $len(\bar{x}) > n$ **wtedy**
12: $\quad\quad\quad stop :=$ PRAWDA
13: $\quad\quad$ **koniec jeżeli**
14: $\quad$ **koniec dopóki**
15: $\quad$ **jeżeli** $stop =$ FAŁSZ $\ \& \ zmiana =$ PRAWDA **wtedy**
16: $\quad\quad instancja^{poziom} := (\bar{S}(Q), m_{\bar{x}}$ dla każdego $\bar{x} \in \bar{S}(Q), n)$
17: $\quad\quad poziom := poziom + 1$
18: $\quad$ **koniec jeżeli**
19: **koniec dla**

Odpowiedź: hierarchia uproszczonych instancji: $instancja^0$, $instancja^1$, ..., $instancja^{max(c_{ij})-1}$; skład połączonych sekwencji słów reprezentujących połączone oligonukleotydy na każdym poziomie upraszczania (kolekcja zmiennych $skład_{\bar{x}}^{poziom}$).

Sposób wykonania tej transformacji definiuje Algorytm 5. Połączone oligonukleotydy występujące w P^i w postaci $\bar{x}$ są zastępowane przez sekwencję oligonukleotydów zapisaną w zmiennej $skad_{\bar{x}}^i$.

Algorytm 6 Wielopoziomowy algorytm ACO - transformacja rozwiązania.

Instancja: rozwiązanie i-tego poziomu $P^i = \{\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{t^i}\}$, skład połączonych oligonukleotydów na i-tym poziomie $skład_{\bar{x}}^i$ dla każdego połączonego oligonukleotydu i-tego poziomu $\bar{x}$

1: $P^{i-1} := \varnothing$
2: **dla** $j = 1$ **do** $j = t^i$ **wykonaj**
3: $\quad P^{i-1} := P^{i-1} | skład_{\bar{x}_j}^i$
4: **koniec dla**

Odpowiedź: rozwiązanie $(i-1)$-tego poziomu P^{i-1}

Rozdział 9

Wyniki eksperymentów obliczeniowych

9.1 Wprowadzenie

Wszystkie eksperymenty opisane w tym rozdziale zostały przeprowadzone na komputerze klasy PC z procesorem Intel Core 2 Duo (T8100 2,1 GHz), pamięcią 3 GB RAM i systemem operacyjnym Windows XP. Algorytmy zostały zaimplementowane w języku C# i do uruchomienia wymagają .NET framework 3.5.

Etap biochemiczny SBH wymaga dużej liczby kopii analizowanej sekwencji DNA. Zazwyczaj przygotowuje się je za pomocą reakcji PCR, która wymaga znajomości początkowego oligonukleotydu powielanej sekwencji. W związku z tym w większości eksperymentów założono, że informacja o pierwszym oligonukleotydzie jest częścią danych testowych. Jeżeli tak nie jest, to jest to podkreślone w opisie danego eksperymentu.

9.1.1 Kryteria porównywania algorytmów

Do porównywania algorytmów wykorzystywanych jest kilka różnych kryteriów. Pierwszym z nich jest *jakość rozwiązania*, która dla danego rozwiązania reprezentuję liczbę wykorzystanych oligonukleotydów ze spektrum. Dla danej instancji problemu można wyznaczyć *oczekiwaną jakość rozwiązania*. W przypadku algorytmów nie wykorzystujących dodatkowej informacji o powtórzeniach wartość ta jest równa liczności idealnego spektrum danej sekwencji pomniejszonej o liczbę błędów negatywnych obu rodzajów (powtórzenia i brak hybrydyzacji). W przypadku algorytmów biorących pod uwagę częściową informację o powtórzeniach przy obliczeniu oczekiwanej jakości brane są pod uwagę jedynie negatywne błędy hybrydyzacji wynikające z niedoskonałości eksperymentu biochemicznego, a oczekiwana jakość jest równa liczności idealnego multispektrum pomniejszonej o liczbę negatywnych błędów hybrydyzacji.

Warto przy tym zauważyć, że przy wykorzystaniu klasycznych bibliotek oligonukleotydów o jednakowej długości l do obliczenia oczekiwanej jakości wystarczy sama długość sekwencji i liczba błędów. Przykładowo dla sekwencji o długości 109 nukleotydów, $l = 10$ i łącznej liczbie wszystkich negatywnych błędów równej 20 oczekiwana jakość wynosi 109-10+1-20=80. Przy stosowaniu bibliotek izotermicznych do wyznaczenia oczekiwanej jakości niezbędna jest analiza danej sekwencji. Oligonukleotydy biblioteki izotermicznej

mają różną długość, co powoduje, że dwie sekwencje o tej samej długości mogą mieć idealne spektra o różnej liczności. Dlatego w przypadku wyników dotyczących instancji dla bibliotek izotermicznych podawana jest uśredniona wartość oczekiwanej jakości dla wszystkich sekwencji o danej długości.

Drugim stosowanym kryterium jest liczba otrzymanych rozwiązań, których *jakość jest równa jakości oczekiwanej*. Dla sekwencji zawierających powtórzenia zaobserwowano, że w szczególnych przypadkach możliwe jest otrzymanie rozwiązania, którego jakość jest wyższa od oczekiwanej. W związku z tym wprowadzono dodatkowe kryterium prezentujące liczbę wygenerowanych rozwiązań, których *jakość jest wyższa od oczekiwanej*. Warto przy tym zauważyć, że pomimo lepszej oceny z perspektywy przyjętej funkcji celu, takie rozwiązania na pewno będą różne od sekwencji rzeczywistej. Liczbę rozwiązań, które są identyczne z analizowaną sekwencją, prezentuje kryterium *idealnej rekonstrukcji*.

Kolejnym kryterium jest *podobieństwo* zrekonstruowanej sekwencji względem sekwencji rzeczywistej. Jako funkcja oceny podobieństwa została wykorzystana miara globalnego dopasowania (ang. *global alignment*). Do obliczeń zastosowano algorytm Needlemana-Wunscha [66] sparametryzowany w następujący sposób: dopasowanie +1, brak dopasowania -1, wstawienie/usunięcie -1. Rozwiązanie identyczne z rzeczywistą sekwencją otrzymywało najwyższą ocenę podobieństwa równą liczbie nukleotydów w sekwencji. Podobieństwo przedstawiane jest w zarówno w skali punktowej jak i procentowej. W przypadku podobieństwa procentowego ocena wynosi 100%, jeżeli zrekonstruowana i analizowana sekwencja są dokładnie takie same. Podobieństwo procentowe liczone jest zgodnie z następującym wzorem.

$$podobieństwo\,procentowe = \frac{podobieństwo\,punktowe - min\,podobieństwo}{max\,podobieństwo - min\,podobieństwo}$$

Minimalne i maksymalne podobieństwo wyznaczane są następująco:

$$min\,podobieństwo = \begin{cases} |Q'| \cdot w_n + (|Q| - |Q'|) \cdot w_u & \text{jeżeli } |Q'| \leqslant |Q| \\ |Q| \cdot w_n + (|Q'| - |Q|) \cdot w_u & \text{w przeciwnym razie} \end{cases}$$

$$max\,podobieństwo = \begin{cases} |Q| \cdot w_d & \text{jeżeli } |Q'| \leqslant |Q| \\ |Q'| \cdot w_d & \text{w przeciwnym razie} \end{cases}$$

gdzie $|Q|$ jest długością analizowanej sekwencji, $|Q'|$ jest długością otrzymanej sekwencji, w_n jest wagą braku dopasowania równą -1, w_u jest wagą wstawienia/usunięcia równą -1, a w_d jest wagą dopasowania równą 1.

Ostatnie dwa wykorzystywane kryteria to *długość otrzymanej sekwencji* oraz *czas obliczeń*.

9.1.2 Testowe zestawy sekwencji

Algorytmy zaprezentowane w Rozdziale 8 zostały przetestowane z wykorzystaniem kilku różnych zestawów sekwencji. Pierwszy z nich, nazywany A, jest zbiorem użytym w [12]. Składa się on z sekwencji ludzkiego DNA kodującego białka. Sekwencje mają długość od 109 do 509 nukleotydów, a dla każdej długości zbiór zawiera 40 różnych sekwencji. Sekwencje nie zawierają powtórzeń. Spektra zostały wygenerowane jedynie dla bibliotek klasycznych i zawierają l-mery o długości $l = 10$. Ponadto spektra zawierają 20% błędów negatywnych i 20% błędów pozytywnych.

Drugi zestaw sekwencji, nazwany B, zawiera rzeczywiste sekwencje DNA pobrane z GenBank o długości 200, 400 i 600 nukleotydów. Został on wykorzystany do testów algorytmu genetycznego dla izotermicznego SBH [17]. Zestaw zawiera 40 sekwencji dla każdej długości, a spektra zostały przygotowane dla bibliotek izotermicznych o temperaturach topnienia 26°C i 28°C oraz dla biblioteki oligonukleotydów o długości $l = 10$. Dla każdej sekwencji przygotowano dwa spektra różniące się liczbą losowo wprowadzonych błędów symulujących niedoskonałość eksperymentu biochemicznego. Zawierają one odpowiednio następującą liczbę negatywnych i pozytywnych błędów hybrydyzacji (liczba błędów obu rodzajów jest taka sama): 5% i 20%. Sekwencje zostały dobrane w taki sposób, że spektra nie zawierają błędów negatywnych wynikających z powtórzeń. W dalszej części wykorzystuje się następujące nazewnictwo zestawów spektr: B.1 - spektra wygenerowane dla bibliotek klasycznych, B.2 - spektra wygenerowane dla bibliotek izotermicznych.

Trzeci zestaw, nazwany C, zawiera 59 rzeczywistych sekwencji DNA o długości 509 nukleotydów. Zwierają one od 1 do 32 naturalnych powtórzeń. Spektra zostały przygotowane dla oligonukleotydów o długości $l = 10$ i nie zawierają żadnych dodatkowych błędów (brak jakichkolwiek błędów hybrydyzacji). Ten zbiór danych został po raz pierwszy wykorzystany w [13] celem porównania dwóch algorytmów opracowanych dla klasycznego SBH: hybrydowego algorytmu genetycznego (ang. *hybrid genetic algorithm*) [16] oraz poprzedniej wersji algorytmu przeszukiwania tabu [12]. Oryginalnie spektra nie zawierały żadnej informacji o powtórzeniach. Zostały one zmodyfikowane poprzez dodanie częściowej informacji o powtórzeniach zgodnej z modelem "jeden, dwa i wiele", aby można było zaobserwować jej wpływ na proces rekonstrukcji sekwencji.

Czwarty zestaw sekwencji, nazwany D, został użyty w [14] do oceny usprawnionego hybrydowego algorytmu genetycznego. Zawiera on 40 sekwencji DNA kodującego ludzkie białka o długości 600 nukleotydów, a każda z sekwencji zawiera pewną liczbę naturalnie powtarzających się oligonukleotydów. Dla tych sekwencji wygenerowano spektra zarówno w oparciu o biblioteki klasyczne jak i izotermiczne. W przypadku klasycznych bibliotek oligonukleotydów ich długość wynosi $l = 10$, a w przypadku bibliotek izotermicznych zastosowano biblioteki o temperaturach topnienia 26°C i 28°C. W obu przypadkach wygenerowano dwa typy spektr. Pierwsze zawierały jedynie błędy wynikające z powtórzeń w sekwencjach. W drugich wprowadzono dodatkowo 5% losowych błędów pozytywnych oraz pewną liczbę losowych negatywnych błędów hybrydyzacji, aby łączna liczba wszystkich błędów negatywnych stanowiła 5% liczności idealnego spektrum. Liczba powtórzeń w przypadku stosowania klasycznych bibliotek wynosiła od 1 do 17 (średnio ok. 4), a w przypadku stosowania bibliotek izotermicznych wahała się od 4 do 30 (średnio ok. 16). Wszystkie powyższe spektra zostały rozszerzone tak, aby zawierały dodatkową informację o powtórzeniach zgodną z modelem "jeden, dwa i wiele". W dalszej części stosuje się następujące nazewnictwo poszczególnych zestawów spektr: D.1 - spektra dla bibliotek klasycznych zawierające jedynie błędy wynikające z powtórzeń, D.2 - spektra dla bibliotek klasycznych zawierające dodatkowo losowe błędy hybrydyzacji, D.3 - spektra dla bibliotek izotermicznych zawierające jedynie błędy wynikające z powtórzeń, D.4 - spektra dla bibliotek izotermicznych zawierające dodatkowo losowe błędy hybrydyzacji.

Ostatni zestaw, nazwany E, został opracowany specjalnie na potrzeby oceny wpływu dodatkowej informacji o powtórzeniach na wyniki generowane przez algorytmy opisane w niniejszej pracy. Zawiera on sekwencje ludzkiego DNA pochodzące z GenBank o długości 109, 209, 309, 409 i 509 nukleotydów. Spektra zostały przygotowane zarówno dla bibliotek klasycznych ($l = 10$) jak i izotermicznych ($T_L = 26$°C i $T_L + 2$°C$= 28$°C). Zawierają one 10% losowych pozytywnych błędów hybrydyzacji oraz 10% losowych negatywnych błędów hybrydyzacji, przy czym losowe usunięcie oligonukleotydu ze spektrum mogło

dotyczyć również oligonukleotydu występującego wielokrotnie w analizowanej sekwencji. Dodatkowo każda sekwencja zawiera powtórzenia prowadzące do kolejnych 5% błędów negatywnych dla bibliotek klasycznych. W przypadku bibliotek izotermicznych liczba negatywnych błędów wynikających z powtórzeń jest nawet kilkakrotnie większa. Aby uzyskać tą samą liczbę błędów wynikających z powtórzeń dla bibliotek izotermicznych należałoby użyć innych sekwencji, ale wtedy nie byłoby możliwości bezpośredniego porównania obu podejść. Zestaw zawiera 40 sekwencji dla każdej wspomnianej długości, a dla każdej sekwencji wygenerowano 10 różnych spektr (za każdym razem wprowadzono losowo inne błędy symulujące błędy hybrydyzacji). W związku z tym zestaw zawiera 400 instancji dla każdej długości sekwencji. Warto również dodać, że każde spektrum zostało posortowane alfabetycznie, aby usunąć informację o oryginalnej kolejności oligonukleotydów w rekonstruowanym DNA.

9.1.3 Parametry algorytmów

Wartości parametrów algorytmu przeszukiwania tabu zostały ustawione następująco: liczba restartów = 15, liczba cykli naprzemiennego wykonywania ruchów kondensujących i rozszerzających = 300, rozmiar zbioru referencyjnego = 8, minimalna liczba ruchów przed dodaniem kolejnego rozwiązania do zbioru referencyjnego = 10, rozmiar listy tabu = 10, liczba ruchów kondensujących bez poprawy najlepszego rozwiązania, po których wykonywane są ruchy rozszerzające = 2 i liczba wykonywanych ruchów rozszerzających = 4. Wartości są identyczne jak te użyte w [12].

Wartości parametrów algorytmu kolonii mrówek (ACO) zostały ustawione następująco: liczba tworzonych (w jednej iteracji) rozwiązań *od początku* $n_f = 3$, liczba tworzonych (w jednej iteracji) rozwiązań *od końca* $n_b = 3$, rozmiar ograniczonej listy kandydatów $r = 10$, współczynnik determinizmu $q = 0{,}9$, współczynnik uczenia $\rho = 0{,}1$, maksymalna liczba iteracji bez poprawy rozwiązania $P_{bs} = 100$, minimalna wartość feromonu $\tau_{min} = 0{,}01$, maksymalna wartość feromonu $\tau_{max} = 0{,}99$. Wartości są identyczne jak te użyte w [20].

9.2 Wyniki dla sekwencji DNA bez powtórzeń

9.2.1 Biblioteki klasyczne

9.2.1.1 Zestaw A

Celem tego eksperymentu było porównanie poprzedniej wersji algorytmu przeszukiwania tabu dla klasycznego SBH z błędami dowolnego typu [12] z nową wersją opisaną w Rozdziale 8.3. Dodatkowo oba algorytmy przeszukiwania tabu zostały porównane z algorytmem ACO opisanym w Rozdziale 8.4 i ML-ACO opisanym w Rozdziale 8.5.

W eksperymencie opisanym w [12] nie wykorzystywano informacji o pierwszym oligonukleotydzie analizowanej sekwencji. Aby móc rzetelnie porównać obie wersje algorytmu przeszukiwania tabu, w eksperymencie opisywanym w niniejszym rozdziale zdecydowano się również nie wykorzystywać tej informacji.

Pewne kroki algorytmów opisanych w Rozdziale 8 mają charakter losowy, więc obliczenia przeprowadzono 10 razy dla każdej sekwencji, a poniżej zaprezentowano wartości

TABELA 9.1: Wyniki dla zestawu A (biblioteki klasyczne) - sekwencje o długości od 109 do 509 nukleotydów nie zawierające powtórzeń, spektra zawierają 20% błędów negatywnych i 20% błędów pozytywnych.

Rozmiar spektrum/ocz. jakość	100/80	200/160	300/240	400/320	500/400
Poprzednia wersja algorytmu przeszukiwania tabu [12][a]					
Średnia jakość	80,00	159,90	239,20	318,10	396,40
Rozwiązania o ocz. jakości #	40,00/40	38,00/40	31,00/40	21,00/40	18,00/40
Idealna rekonstrukcja #	-	-	-	-	-
Średnie podobieństwo [pkt]	108,40	207,60	273,70	323,90	361,40
Średnie podobieństwo [%]	99,70	99,70	94,30	89,60	85,50
Średni czas obliczeń [ms][b]	14600	61700	178300	265700	474500
Algorytm przeszukiwania tabu opisany w Rozdziale 8.3					
Średnia jakość	80,00	159,89	239,55	319,38	398,89
Rozwiązania o ocz. jakości #	40,00/40	37,90/40	35,20/40	30,90/40	25,20/40
Idealna rekonstrukcja #	30,00/40	28,20/40	23,40/40	19,60/40	20,40/40
Średnie podobieństwo [pkt]	108,40	206,40	289,49	374,92	436,62
Średnie podobieństwo [%]	99,72	99,38	96,84	95,83	92,89
Średni czas obliczeń [ms]	1151	6331	20148	48724	101352
Algorytm kolonii mrówek opisany w Rozdziale 8.4					
Średnia jakość	79,91	159,53	239,25	318,35	396,59
Rozwiązania o ocz. jakości #	38,40/40	33,80/40	29,70/40	21,80/40	13,50/40
Idealna rekonstrukcja #	29,30/40	27,20/40	25,30/40	21,00/40	22,60/40
Średnie podobieństwo [pkt]	107,91	202,09	290,79	372,03	436,50
Średnie podobieństwo [%]	99,50	98,35	97,05	95,48	92,88
Średni czas obliczeń [ms]	105	555	1340	2863	4815
Wielopoziomowy algorytm kolonii mrówek opisany w Rozdziale 8.5					
Średnia jakość	80,00	159,86	239,98	319,68	398,99
Rozwiązania o ocz. jakości #	40,00/40	38,30/40	39,30/40	36,90/40	32,90/40
Idealna rekonstrukcja #	30,00/40	28,70/40	27,30/40	24,90/40	27,80/40
Średnie podobieństwo [pkt]	108,40	205,24	299,53	394,30	473,86
Średnie podobieństwo [%]	99,72	99,10	98,47	98,20	96,55
Średni czas obliczeń [ms]	84	327	692	1429	2430

[a] wyniki eksperymentów zaprezentowane w [12]

[b] obliczenia wykonane na komputerze PC z procesorem Pentium II 300MHz, 256 MB pamięci RAM i systemem operacyjnym Linux

uśrednione. W przypadku poprzedniej wersji algorytmu przeszukiwania tabu wykorzystano wartości przedstawione w [12] (gdy dana wartość jest niedostępna użyto znaku −).

Należy przy tym zauważyć, że poprzednie wersje algorytmów ACO i ML-ACO opisane w [20] zostały przetestowane dla tego samego zestawu sekwencji i spektr, ale eksperyment został przeprowadzony w inny sposób. Obliczenia również wykonano 10 razy dla każdej sekwencji, ale wyniki nie zostały uśrednione, tylko został wybrany najlepszy uzyskany rezultat. W konsekwencji większość wyników jest nieporównywalna. Niestety programy implementujące te algorytmy nie są dostępne, co uniemożliwia rzetelne porównanie nowych algorytmów z ich poprzednimi wersjami. Pewną próbą jest ocena złożoności czasowej, którą zrealizowano poprzez znormalizowanie czasu obliczeń względem czasu obliczeń dla najkrótszych sekwencji o długości 109 nukleotydów.

Tabela 9.2: Wyniki dla zestawu A (biblioteki klasyczne) - znormalizowane czasy obliczeń względem czasów dla najkrótszych sekwencji o długości 109 nukleotydów, spektra zawierają 20% błędów negatywnych i 20% błędów pozytywnych.

Rozmiar spektrum	100	200	300	400	500
Poprzednia wersja tabu [12][b]	1,00	4,23	12,21	18,20	32,50
Nowa wersja tabu (Rozdział 8.3)	1,00	5,50	17,50	42,33	88,06
Poprzednia wersja ACO [20][a]	1,00	13,29	36,36	112,29	273,79
Nowa wersja ACO (Rozdział 8.4)	1,00	5,29	12,76	27,27	45,86
Poprzednia wersja ML-ACO [20][a]	1,00	82,00	82,00	994,00	1570,00
Nowa wersja ML-ACO (Rozdział 8.5)	1,00	3,89	8,24	17,01	28,93

[a] obliczone na podstawie wyników eksperymentów zaprezentowanych w [20]
[a] obliczone na podstawie wyników eksperymentów zaprezentowanych w [12]

Wyniki eksperymentu przedstawiają Tabela 9.1 oraz Tabela 9.2. Pierwsza z nich zawiera następujące wartości: średnia jakość otrzymanego rozwiązania, liczba rozwiązań o jakości równej jakości oczekiwanej, liczba idealnie zrekonstruowanych sekwencji, średnie podobieństwo oraz średni czas obliczeń. Zestaw składa się z 40 sekwencji, więc jest to maksymalna wartość kryteriów, które prezentują pewną liczbę rozwiązań. W przypadku podobieństwa maksymalne wartości punktowe wynoszą w zależności od długości sekwencji od 109 do 509.

Tabela 9.2 zawiera znormalizowane czasy obliczeń dla wszystkich wspomnianych powyżej algorytmów. Każdy wiersz prezentuje wartości dla jednego algorytmu. Normalizacja została zrealizowana przez podzielenie czasu obliczeń dla danej długości sekwencji przez czas obliczeń dla sekwencji o długości 109 nukleotydów. Umożliwia to zaobserwowanie jak rośnie czas obliczeń przy rosnącym rozmiarze spektrum.

Wielopoziomowy algorytm kolonii mrówek (ML-ACO) zaprezentowany w Rozdziale 8.5 przewyższa oba algorytmy przeszukiwania tabu jak i algorytm kolonii mrówek opisany w Rozdziale 8.4. Liczba uzyskanych rozwiązań o jakości równej jakości oczekiwanej oraz liczba idealnie zrekonstruowanych sekwencji są znacząco większe. Średnia jakość oraz średnie podobieństwo sekwencji są również wyższe. Jednakże wyniki dla ACO nie są takie spektakularne jak opisano w [20]. Liczba rozwiązań o jakości równej oczekiwanej jakości jest najniższa spośród wszystkich czterech algorytmów. Średnie podobieństwo i średnia liczba idealnie zrekonstruowanych sekwencji dla ACO jest podobna do wyników uzyskanych przez nowy algorytm przeszukiwania tabu.

Czas obliczeń dla nowego algorytmu kolonii mrówek jest o rząd wielkości mniejszy w porównaniu z nowym algorytmem przeszukiwania tabu. Co więcej, zastosowanie pomysłu wielopoziomowego udoskonalania rozwiązania prowadzi do dalszej redukcji czasu obliczeń o ok. 42% w porównaniu z ACO (średnia obliczona dla wszystkich rozmiarów spektr). Wyniki zaprezentowane w [12], [20] i niniejszej pracy zostały uzyskane przy wykorzystaniu maszyn o różnych mocach obliczeniowych, dlatego zostały one znormalizowane, aby umożliwić bezpośrednie porównanie. ML-ACO jest również najlepszym algorytmem biorąc pod uwagę to kryterium. Warto przy tym zauważyć ogromną różnicę pomiędzy algorytmami kolonii mrówek zaprezentowanymi w [20] i odpowiadającymi im algorytmami opisanymi w niniejszej pracy. Mimo że ogólna idea jest taka sama, to różnica jest ogromna. Najprawdopodobniej muszą być jakieś istotne różnice w sposobie implementacji tych algorytmów. Niestety kod źródłowy programów implementujących algorytmy z [20] nie jest dostępny, co uniemożliwia wykonanie szczegółowych analiz.

TABELA 9.3: Wyniki dla zestawu B.1 (biblioteki klasyczne) - sekwencje o długości 200, 400 i 600 nukleotydów nie zawierające powtórzeń, spektra zawierają po 5% i 20% błędów hybrydyzacji obu rodzajów.

Długość sekwencji	200		400		600	
Liczba błędów	±5%	±20%	±5%	±20%	±5%	±20%
Poprzednia wersja algorytmu przeszukiwania tabu [12][a]						
Idealna rekonstrukcja #	40/40	36/40	32/40	21/40	32/40	15/40
Średnie podobieństwo [pkt]	–	–	–	–	–	–
Średnie podobieństwo [%]	99,9	97,9	95,3	89,4	95,2	80,5
Algorytm przeszukiwania tabu opisany w Rozdziale 8.3						
Idealna rekonstrukcja #	40,0/40	40,0/40	37,7/40	27,1/40	36,5/40	17,6/40
Średnie podobieństwo [pkt]	200,00	200,00	390,72	344,84	583,46	436,27
Średnie podobieństwo [%]	100,00	100,00	98,84	93,11	98,62	86,36
Wielopoziomowy algorytm kolonii mrówek opisany w Rozdziale 8.5						
Idealna rekonstrukcja #	40,0/40	40,0/40	40,0/40	40,0/40	39,2/40	36,4/40
Średnie podobieństwo [pkt]	200,00	200,00	400,00	400,00	599,96	587,68
Średnie podobieństwo [%]	100,00	100,00	100,00	100,00	100,00	98,97

[a] wyniki eksperymentów zaprezentowane w [17]

9.2.1.2 Zestaw B

Celem tego eksperymentu było porównanie poprzedniej wersji algorytmu przeszukiwania tabu opracowanego dla bibliotek klasycznych z nową wersją opisaną w Rozdziale 8.3 oraz porównanie obu tych algorytmów z ML-ACO, który został zaprezentowany w Rozdziale 8.5.

Wyniki przedstawiono w Tabeli 9.3 (zestaw B.1). Zawiera ona dla każdego z powyższych algorytmów dwie wartości: liczbę idealnie zrekonstruowanych sekwencji oraz średnie podobieństwo punktowe i procentowe otrzymanych rozwiązań. Zostały one podane dla poszczególnych długości sekwencji dla spektr z różną liczbą błędów hybrydyzacji. Zapis ±5% oznacza, że dane spektrum zawiera 5% błędów pozytywnych oraz 5% błędów negatywnych. Analogicznie ±20% oznacza, że spektrum zawiera 20% błędów pozytywnych oraz 20% błędów negatywnych. W przypadku algorytmów opisanych w niniejszej pracy każda z sekwencji została zrekonstruowana 10 razy, a tabela zawiera wartości uśrednione. Wartości dla poprzedniej wersji algorytmu przeszukiwania tabu dla bibliotek o oligonukleotydach równej długości pochodzą z [17] (gdy dana wartość jest niedostępna użyto znaku −).

Nowy algorytm przeszukiwania tabu opisany w Rozdziale 8.3 umożliwia uzyskanie wyraźnie lepszych rezultatów niż jego poprzednia wersja przedstawiona w [12]. Zarówno liczba idealnie zrekonstruowanych sekwencji jak i średnie podobieństwo są znacząco wyższe. Co więcej, dla najkrótszych sekwencji z zestawu B (tj. 200 nukleotydów) nowa wersja umożliwiła idealne zrekonstruowanie wszystkich sekwencji również dla spektr z większą liczbą błędów.

Jednakże najlepsze wyniki uzyskano dla wielopoziomowego algorytmu kolonii mrówek. Dla sekwencji o długości 200 i 400 nukleotydów korzystając z ML-ACO udało się idealnie odtworzyć wszystkie sekwencje zarówno dla spektr z mniejszą (±5%) jak i większą (±20%) liczbą błędów hybrydyzacji. Dla najdłuższych sekwencji o długości

Tabela 9.4: Wyniki dla zestawu B.2 (biblioteki izotermiczne) - sekwencje o długości 200, 400 i 600 nukleotydów nie zawierające powtórzeń, spektra zawierają po 5% i 20% błędów hybrydyzacji obu rodzajów.

Długość sekwencji	200		400		600	
Liczba błędów	±5%	±20%	±5%	±20%	±5%	±20%
Poprzednia wersja algorytmu przeszukiwania tabu dla ISBH [9][a]						
Idealna rekonstrukcja #	8/40	3/40	2/40	0/40	2/40	0/40
Średnie podobieństwo [pkt]	–	–	–	–	–	–
Średnie podobieństwo [%]	85,2	78,7	75,9	69,7	76,5	68,8
Algorytm genetyczny dla ISBH [17][a]						
Idealna rekonstrukcja #	39/40	37/40	38/40	36/40	36/40	32/40
Średnie podobieństwo [pkt]	–	–	–	–	–	–
Średnie podobieństwo [%]	99,9	99,2	99,2	99,2	98,0	98,0
Algorytm przeszukiwania tabu opisany w Rozdziale 8.3						
Idealna rekonstrukcja #	36,9/40	32,4/40	26,0/40	17,7/40	19,4/40	5,5/40
Średnie podobieństwo [pkt]	194,12	185,31	348,38	306,33	466,45	320,13
Średnie podobieństwo [%]	98,53	96,33	93,55	88,29	88,87	76,68
Wielopoziomowy algorytm kolonii mrówek opisany w Rozdziale 8.5						
Idealna rekonstrukcja #	38,4/40	38,9/40	37,4/40	34,6/40	35,8/40	30,4/40
Średnie podobieństwo [pkt]	199,13	199,00	396,53	394,56	580,82	554,40
Średnie podobieństwo [%]	99,78	99,75	99,57	99,32	98,40	96,20

[a] wyniki eksperymentów zaprezentowane w [17]

600 nukleotydów i spektr z 5% błędów pozytywnych i 5% negatywnych ML-ACO idealnie zrekonstruował prawie wszystkie sekwencje, a podobieństwo otrzymanych rezultatów jest tak wysokie, że średnie podobieństwo procentowe zaokrąglone do dwóch miejsc po przecinku wynosi 100,00%. W przypadku spektr z większą liczbą błędów (±20%) średnie podobieństwo jest również bardzo wysokie, a analizowaną sekwencję udało się idealnie odtworzyć w 91% przypadków.

9.2.2 Biblioteki izotermiczne

9.2.2.1 Zestaw B

Celem tego eksperymentu było porównanie poprzedniej wersji algorytmu przeszukiwania tabu opracowanego dla bibliotek izotermicznych z nową wersją opisaną w Rozdziale 8.3 oraz porównanie obu tych algorytmów z innymi algorytmami dla ISBH. Dlatego przy porównywaniu wzięto pod uwagę również algorytm genetyczny dla ISBH [17] oraz wielopoziomowy algorytm kolonii mrówek, który został przedstawiony w Rozdziale 8.5. Dodatkowo otrzymane wyniki porównano z wynikami Rozdziału 9.2.1.2, gdzie dla tych samych sekwencji przedstawiono rezultaty uzyskane przy użyciu algorytmów dla bibliotek klasycznych.

Wyniki przedstawiono w Tabeli 9.4 (zestaw B.2). Zawiera ona dla każdego z powyższych algorytmów dwie wartości: liczbę idealnie zrekonstruowanych sekwencji oraz średnie podobieństwo punktowe i procentowe otrzymanych rozwiązań. Zostały one podane

dla poszczególnych długości sekwencji dla spektr z różną liczbą błędów hybrydyzacji. Zapis ±5% oznacza, że dane spektrum zawiera 5% błędów pozytywnych oraz 5% błędów negatywnych. Analogicznie ±20% oznacza, że spektrum zawiera 20% błędów pozytywnych oraz 20% błędów negatywnych. W przypadku algorytmów opisanych w niniejszej pracy każda z sekwencji została zrekonstruowana 10 razy, a tabela zawiera wartości uśrednione. Wartości dla poprzedniej wersji algorytmu przeszukiwania tabu dla bibliotek izotermicznych oraz dla algorytmu genetycznego pochodzą z [17] (gdy dana wartość jest niedostępna użyto znaku −).

Najsłabsze wyniki generuje poprzednia wersja algorytmu tabu dla ISBH [9]. Wykorzystanie nowej wersji algorytmu przeszukiwania tabu prowadzi do uzyskania większej liczby idealnie zrekonstruowanych sekwencji, a otrzymane sekwencje są bardziej podobne do analizowanych sekwencji. Najlepsze rezultaty otrzymano dla algorytmu genetycznego [17] oraz ML-ACO. Wyniki obu tych algorytmów są dość zbliżone. Liczba idealnie zrekonstruowanych sekwencji jest w ogólności nieco wyższa (o ok. 1 sekwencję) dla algorytmu genetycznego, a dla wielopoziomowego algorytmu kolonii mrówek w 4 z 6 przypadków otrzymano wyższe podobieństwo sekwencji.

Jednakże algorytmy opisane w niniejszej pracy umożliwiają uzyskanie lepszych rezultatów w przypadku stosowania klasycznych bibliotek oligonukleotydów o równej długości. Podobną zależność można również zauważyć dla poprzednich wersji algorytmu przeszukiwania tabu. Poprzednia wersja algorytmu dla bibliotek klasycznych umożliwia uzyskanie dużo lepszych wyników niż poprzednia wersja dla ISBH.

Różnice pomiędzy wynikami dla bibliotek klasycznych i izotermicznych są bardzo wyraźne. ML-ACO przy stosowaniu klasycznych bibliotek umożliwił idealną rekonstrukcję wszystkich sekwencji o długości 200 i 400 nukleotydów dla spektr zarówno z mniejszą (±5%) jak i większą (±20%) liczbą błędów. Wszystkich sekwencji nie udało się odtworzyć żadnemu algorytmowi dla ISBH, nawet dla najkrótszych sekwencji i ±5% błędów hybrydyzacji. Dla najdłuższych sekwencji (tj. 600 nukleotydów) ML-ACO przy użyciu klasycznych bibliotek umożliwił idealną rekonstrukcję w 98% przypadków dla ±5% błędów hybrydyzacji i w 91% przypadków dla ±20% błędów hybrydyzacji. Najlepsze wyniki dla algorytmów dla ISBH wynoszą odpowiednio 90% i 80%.

9.3 Wyniki dla sekwencji DNA zawierających naturalne powtórzenia

9.3.1 Biblioteki klasyczne

9.3.1.1 Zestaw C

Pewne wyniki dotyczące wpływu występowania powtórzeń w analizowanych sekwencjach na proces rekonstrukcji zostały przedstawione przy wykorzystaniu algorytmów dla klasycznego SBH [13]. W pracy tej przeprowadzono eksperymenty dla hybrydowego algorytmu genetycznego [16] oraz poprzedniej wersji algorytmu przeszukiwania tabu [12]. Celem niniejszego eksperymentu jest porównanie wyników dla tych dwóch algorytmów z wynikami algorytmów opisanych w Rozdziale 8.

Wyniki eksperymentu prezentuje Tabela 9.5. Przedstawione zostały następujące wartości: średnia jakość otrzymanych rozwiązań, średnia oczekiwana jakość rozwiązań, liczba

Tabela 9.5: Wyniki dla zestawu C (biblioteki klasyczne) - sekwencje o długości 509 nukleotydów z naturalnymi powtórzeniami, spektra zawierają jedynie błędy wynikające z powtórzeń.

Algorytm	Hybrydowy genetyczny [16]	Poprzednia wersja tabu [12]	Nowa wersja tabu	Nowa wersja ML-ACO
Średnia jakość	493,60	495,40	499,97	498,56
Średnia oczekiwana jakość	496,20[a]	496,20[a]	500,00[b]	500,00[b]
Rozwiązania o ocz. jakości #	26,00/59	52,00/59	58,40/59	45,20/59
Idealna rekonstrukcja #	–	–	41,80/59	37,10/59
Średnie podobieństwo [pkt]	–	–	469,32	454,70
Średnie podobieństwo [%]	–	–	96,10	94,67
Średni czas obliczeń [ms]	– [c]	– [c]	3335	2221

[a] rozmiar idealnego spektrum
[b] rozmiar idealnego multispektrum
[c] nie porównywalne ze względu na obliczenia na innej maszynie

rozwiązań o jakości równej jakości oczekiwanej, liczba idealnie zrekonstruowanych sekwencji, średnie podobieństwo punktowe i procentowe oraz średni czas obliczeń. Wartość oczekiwanej jakości została uśredniona, ponieważ poszczególne sekwencje zawierają różną liczbę powtórzeń oligonukleotydów, przez co mogą one mieć różną jakość oczekiwaną (dotyczy algorytmów nie wykorzystujących dodatkowej informacji o powtórzeniach).

Ze względu na to, że pewne kroki algorytmów przedstawionych w niniejszej rozprawie mają charakter losowy, obliczenia przeprowadzono 10 razy dla każdej sekwencji i zaprezentowano wartości uśrednione. Wartości dla pozostałych dwóch algorytmów pochodzą z [13] (gdy dana wartość jest niedostępna użyto znaku −), gdzie każda z sekwencji została zrekonstruowana raz. Czas obliczeń dla tych algorytmów został pominięty, ze względu na przeprowadzenie obliczeń na innej maszynie.

Najlepsze rozwiązania uzyskano dla nowego algorytmu przeszukiwania tabu opisanego w Rozdziale 8.3. ML-ACO generuje lepsze rozwiązania niż hybrydowy algorytm genetyczny, ale gorsze niż oba algorytmy przeszukiwania tabu. Zaletą ML-ACO jest mniejszy czas obliczeń. Nowy algorytm przeszukiwania tabu wymaga ok. 50% więcej czasu niż ML-ACO.

Warto przy tym zauważyć, że spektra nie zawierały żadnych innych błędów oprócz błędów wynikających z powtórzeń. Najprawdopodobniej jest to powodem lepszych ocen algorytmów przeszukiwania tabu, które umożliwiają uzyskanie bardzo dobrych wyników dla małej liczby błędów hybrydyzacji (wniosek wyciągnięty na podstawie wyników pozostałych eksperymentów).

9.3.1.2 Zestaw D

Kolejne wyniki dotyczące wpływu występowania powtórzeń na rekonstrukcję sekwencji zostały zaprezentowane w [14]. W pracy tej przedstawiono wyniki dla usprawnionego algorytmu genetycznego oraz poprzedniej wersji algorytmu przeszukiwania tabu [12]. Celem niniejszego eksperymentu było porównanie tych algorytmów z wybranymi algorytmami zaprezentowanymi w Rozdziale 8. Do porównania wybrano nową wersję przeszukiwania tabu oraz ML-ACO.

Wyniki zostały przedstawione w Tabeli 9.6 (zestaw D.1) oraz Tabeli 9.7 (zestaw

Tabela 9.6: Wyniki dla zestawu D.1 (biblioteki klasyczne) - sekwencje o długości 600 nukleotydów z naturalnymi powtórzeniami, spektra zawierają jedynie błędy wynikające z powtórzeń.

Algorytm	Poprzednia wersja tabu [12]	Uspr. hybryd. genetyczny [14]	Nowa wersja tabu	Nowa wersja ML-ACO
Średnia jakość	585,53[a]	586,35[a]	591,00	589,85
Średnia oczekiwana jakość	586,35[b]	586,35[b]	591,00[c]	591,00[c]
Rozwiązania o ocz. jakości #	–	–	39,90/40	28,80/40
Idealna rekonstrukcja #	14,00/40	18,00/40	23,40/40	18,50/40
Średnie podobieństwo [pkt]	–	–	518,91	497,54
Średnie podobieństwo [%]	88,45	90,99	93,24	91,46
Średni czas obliczeń [ms]	– [d]	– [d]	4845	3067

Tabela 9.7: Wyniki dla zestawu D.2 (biblioteki klasyczne) - sekwencje o długości 600 nukleotydów z naturalnymi powtórzeniami, spektra zawierają 5% błędów pozytywnych oraz łącznie 5% błędów negatywnych obu rodzajów.

Algorytm	Poprzednia wersja tabu [12]	Uspr. hybryd. genetyczny [14]	Nowa wersja tabu	Nowa wersja ML-ACO
Średnia jakość	558,48[a]	560,94[a]	564,47	565,01
Średnia oczekiwana jakość	561,00[b]	561,00[b]	565,55[c]	565,55[c]
Rozwiązania o ocz. jakości #	–	–	32,20/40	33,40/40
Idealna rekonstrukcja #	10,00/40	18,00/40	17,30/40	18,90/40
Średnie podobieństwo [pt]	–	–	484,52	505,24
Średnie podobieństwo [%]	82,63	92,60	90,38	92,10
Średni czas obliczeń [ms]	– [d]	– [d]	11784	2914

[a] oszacowane na podstawie procentowego użycia oligonukleotydów ze spektrum przedstawionego w [14] oraz średniej oczekiwanej jakości
[b] rozmiar idealnego spektrum pomniejszony o liczbę losowych negatywnych błędów hybrydyzacji
[c] rozmiar idealnego multispektrum pomniejszony o liczbę losowych negatywnych błędów hybrydyzacji
[d] nie porównywalne ze względu na obliczenia na innej maszynie

D.2). Obie tabele zawierają następujące informacje: średnia jakość zrekonstruowanych sekwencji, średnia oczekiwana jakość rozwiązań, liczba otrzymanych rozwiązań o jakości równej oczekiwanej, liczba idealnie zrekonstruowanych sekwencji, średnie podobieństwo punktowe i procentowe oraz średni czas obliczeń. Są to te same kryteria, które zaprezentowano w Tabeli 9.5 dla zestawu C. Obliczenia dla algorytmów prezentowanych w niniejszej pracy również wykonano dla każdej sekwencji 10 razy, a tabela zawiera wartości uśrednione. Wyniki dla pozostałych algorytmów pochodzą z [14] (gdy dana wartość jest niedostępna użyto znaku −), gdzie obliczenia dla danej sekwencji wykonano raz.

Dla zestawu D.1 najlepsze rozwiązania uzyskano przy użyciu algorytmu przeszukiwania tabu zaprezentowanego w Rozdziale 8.3. Generuje on największą liczbę rozwiązań o jakości równej oczekiwanej. Średnie podobieństwo otrzymanych sekwencji i liczba zrekonstruowanych przez niego sekwencji identycznych z analizowanymi są również największe. Wielopoziomowy algorytm kolonii mrówek opisany w Rozdziale 8.5 dostarcza jedynie nieznacznie lepsze wyniki niż usprawniony hybrydowy algorytm genetyczny z [14], tj. średnie podobieństwo jest większe o 0,47%, a uśredniona liczba idealnie zrekonstruowanych sekwencji jest większa o 0,5. W przypadku zestawu D.2 algorytm ML-ACO otrzymuje najwyższe oceny prawie wg wszystkich kryteriów. Jedynie średnie podobieństwo uzyskane przez usprawniony hybrydowy algorytm genetyczny jest wyższe o 0,5%.

Warto przy tym zauważyć, że wprowadzone losowe błędy pozytywne i negatywne praktycznie nie wpłynęły na wyniki generowane przez ML-ACO. Oceny wg poszczególnych kryteriów są dla ML-ACO praktycznie takie same zarówno w przypadku zestawu D.1 jak i D.2. Podobne właściwości ma usprawniony hybrydowy algorytm genetyczny. W przypadku algorytmu przeszukiwania tabu dodatkowe błędy w spektrach doprowadziły do uzyskania gorszych wyników. Liczba rozwiązań o jakości równej oczekiwanej oraz liczba idealnie zrekonstruowanych sekwencji uległy zmniejszeniu, a czas obliczeń został podwojony.

9.3.1.3 Zestaw E

Eksperymenty opisane powyżej miały na celu przede wszystkim porównanie istniejących algorytmów z algorytmami zaproponowanymi w niniejszej pracy. Eksperyment wykorzystujący zestaw E został przeprowadzony wyłącznie po to, aby można było ocenić przydatność częściowej informacji o powtórzeniach.

Wyniki zostały zaprezentowane w Tabeli 9.8 (algorytm zachłanny), Tabeli 9.9 (algorytm przeszukiwania tabu), Tabeli 9.10 (algorytm kolonii mrówek - ACO) oraz Tabeli 9.11 (wielopoziomowy algorytm kolonii mrówek - ML-ACO). Wszystkie powyższe tabele składają się z trzech części. W pierwszej zaprezentowano wyniki dla przypadku braku dostępności informacji o powtórzeniach, w drugiej przedstawiono wyniki dla częściowej informacji o powtórzeniach typu "jeden i wiele", a ostatnia zawiera wyniki dla modelu częściowej informacji o powtórzeniach typu "jeden, dwa i wiele". Każda z części zawiera następujące wartości: średnią jakość otrzymanych sekwencji, oczekiwaną jakość, liczbę rozwiązań o jakości równej oczekiwanej, liczbę rozwiązań o jakości wyższej niż oczekiwana, liczbę idealnie zrekonstruowanych rozwiązań, średnie podobieństwo punktowe i procentowe, średnią długość otrzymanej sekwencji oraz średni czas obliczeń.

Pewne kroki algorytmów opisanych w niniejszej pracy mają charakter losowy, więc obliczenia przeprowadzono 5 razy dla każdego spektrum danej sekwencji, a tabele zawierają wartości uśrednione. Dla danej długości zestaw E zawiera 40 różnych sekwencji, dla każdej z nich przygotowano 10 spektr, więc tabele przedstawiają uśrednione wyniki dla 2000 uruchomień danego algorytmu. Maksymalna wartość oceny dla kryteriów prezentujących pewną liczbę rozwiązań wynosi 400.

Otrzymane wyniki dla wszystkich algorytmów jednoznacznie potwierdzają pozytywny wpływ dodatkowej informacji o powtórzeniach, mimo że wykorzystywane modele nie dostarczają dokładnej informacji o liczbie wystąpień poszczególnych oligonukleotydów w analizowanej sekwencji. Praktycznie wg wszystkich kryteriów oceny są znacząco wyższe niż w przypadku braku dostępności informacji o powtórzeniach, a czas obliczeń jest praktycznie taki sam. Wyjątkiem jest liczba rozwiązań o jakości równej oczekiwanej. W pewnych przypadkach ocena wg tego kryterium jest wyższa dla braku informacji o powtórzeniach, ale warto zauważyć, że jest ona wtedy liczona względem rozmiaru idealnego spektrum, który jest mniejszy niż rozmiar idealnego multispektrum. Warto również zwrócić uwagę, że w przypadku braku informacji o powtórzeniach bardzo często dochodzi do uzyskania rozwiązań o jakości wyższej niż oczekiwana. Można to zaobserwować zarówno dla średniej jakości jak i liczby rozwiązań o jakości wyższej niż oczekiwana. Oznacza to wykorzystanie przy rekonstrukcji takich oligonukleotydów, które z pewnością nie są częścią analizowanej sekwencji.

Tabela 9.8: Wyniki dla zestawu E (biblioteki klasyczne) - algorytm zachłanny.

Długość sekwencji	109	209	309	409	509
Brak informacji o powtórzeniach					
Średnia jakość	84,06	169,57	254,58	338,74	422,03
Oczekiwana jakość	85,00	170,00	255,00	340,00	425,00
Rozwiązania o ocz. jakości #	167,40	80,00	58,20	37,20	33,20
Jakość większa niż oczekiwana #	108,80	200,60	198,80	194,20	166,60
Idealna rekonstrukcja #	39,60	0,00	0,00	0,00	0,00
Średnie podobieństwo [pkt]	92,89	163,49	229,18	279,37	316,64
Średnie podobieństwo [%]	92,61	89,11	87,08	84,15	81,10
Średnia dł. otrzymanej sekwencji	105,60	206,42	306,21	405,80	506,23
Średni czas obliczeń [ms]	2	13	35	77	139
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	86,89	173,93	260,59	347,87	436,54
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	172,80	48,40	26,40	32,80	27,40
Jakość większa niż oczekiwana #	0,00	0,00	0,00	0,00	0,00
Idealna rekonstrukcja #	143,00	30,80	13,60	13,20	6,00
Średnie podobieństwo [pkt]	97,12	170,50	234,20	286,97	331,66
Średnie podobieństwo [%]	94,55	90,79	87,90	85,08	82,58
Średnia dł. otrzymanej sekwencji	108,08	206,57	306,78	406,98	507,24
Średni czas obliczeń [ms]	2	13	36	78	141
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	87,36	174,54	262,29	349,05	437,55
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	243,60	71,80	35,60	40,00	31,60
Jakość większa niż oczekiwana #	0,00	0,00	0,00	0,00	0,00
Idealna rekonstrukcja #	202,00	43,40	14,00	15,60	5,80
Średnie podobieństwo [pkt]	98,16	171,69	236,86	287,70	333,57
Średnie podobieństwo [%]	95,03	91,07	88,33	85,17	82,77
Średnia dł. otrzymanej sekwencji	108,50	207,33	307,10	407,34	507,16
Średni czas obliczeń [ms]	2	13	36	78	144

Wykorzystanie dodatkowej informacji o powtórzeniach prowadzi do szczególnie spektakularnego wzrostu liczby idealnie zrekonstruowanych sekwencji. Dla najkrótszych sekwencji stosując model typu "jeden, dwa i wiele" udało się idealnie zrekonstruować analizowaną sekwencję w ok. 75% przypadków, przy braku informacji o powtórzeniach jedynie w mniej niż 25%. Dla sekwencji o długości 209 nukleotydów wartości te wynoszą odpowiednio ok. 50% i ok. 4%.

Co więcej, wykorzystywanie dokładniejszego modelu prowadzi do dalszego wzrostu ocen, choć różnice pomiędzy obydwoma modelami informacji o powtórzeniach nie są już takie duże i zmniejszają się wraz ze wzrostem długości sekwencji (w pewnych przypadkach dla najdłuższych sekwencji uzyskano nawet nieco lepsze wyniki w przypadku stosowania mniej dokładnego modelu). Użycie dokładniejszego modelu pozwala przede

Tabela 9.9: Wyniki dla zestawu E (biblioteki klasyczne) - algorytm przeszukiwania tabu.

Długość sekwencji	109	209	309	409	509
Brak informacji o powtórzeniach					
Średnia jakość	85,46	171,15	256,77	342,21	426,84
Oczekiwana jakość	85,00	170,00	255,00	340,00	425,00
Rozwiązania o ocz. jakości #	235,40	48,60	38,80	28,60	24,80
Jakość większa niż oczekiwana #	162,60	331,60	347,20	352,00	309,00
Idealna rekonstrukcja #	92,20	3,40	0,00	0,00	0,00
Średnie podobieństwo [pkt]	101,14	183,17	257,93	318,97	368,69
Średnie podobieństwo [%]	96,39	93,82	91,74	88,99	86,22
Średnia dł. otrzymanej sekwencji	105,87	207,52	307,41	407,17	507,45
Średni czas obliczeń [ms]	498	1675	3589	6389	10372
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	90,25	179,97	269,03	357,51	446,40
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	341,40	226,20	184,20	136,20	130,40
Jakość większa niż oczekiwana #	58,40	113,00	90,80	47,40	35,60
Idealna rekonstrukcja #	282,00	180,00	112,00	45,00	24,80
Średnie podobieństwo [pkt]	105,01	194,58	272,89	339,72	402,73
Średnie podobieństwo [%]	98,17	96,55	94,16	91,53	89,56
Średnia dł. otrzymanej sekwencji	108,91	208,77	308,71	408,54	508,59
Średni czas obliczeń [ms]	528	1836	3938	6970	11067
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	90,02	179,94	269,29	357,75	446,58
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	393,00	286,80	205,40	158,40	143,60
Jakość większa niż oczekiwana #	6,60	64,00	81,80	35,80	17,60
Idealna rekonstrukcja #	310,40	207,80	117,40	55,20	25,00
Średnie podobieństwo [pkt]	106,56	197,24	276,70	338,67	398,68
Średnie podobieństwo [%]	98,88	97,19	94,77	91,40	89,16
Średnia dł. otrzymanej sekwencji	108,89	208,79	308,74	408,55	508,60
Średni czas obliczeń [ms]	508	1811	3884	6800	10585

Tabela 9.10: Wyniki dla zestawu E (biblioteki klasyczne) - algorytm kolonii mrówek.

Długość sekwencji	109	209	309	409	509
Brak informacji o powtórzeniach					
Średnia jakość	85,35	170,88	256,21	341,14	424,67
Oczekiwana jakość	85,00	170,00	255,00	340,00	425,00
Rozwiązania o ocz. jakości #	226,20	73,60	50,00	44,20	35,60
Jakość większa niż oczekiwana #	150,60	289,20	307,20	278,20	209,40
Idealna rekonstrukcja #	89,80	13,20	0,20	0,20	0,00
Średnie podobieństwo [pkt]	100,71	182,58	260,91	321,91	368,81
Średnie podobieństwo [%]	96,20	93,68	92,22	89,35	86,23
Średnia dł. otrzymanej sekwencji	105,75	206,55	306,11	406,01	506,52
Średni czas obliczeń [ms]	112	544	1394	2891	4760
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	89,54	177,46	264,18	351,36	438,58
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	277,40	88,80	29,00	17,20	14,80
Jakość większa niż oczekiwana #	14,80	23,40	0,80	1,80	0,00
Idealna rekonstrukcja #	281,60	83,00	28,20	31,00	11,20
Średnie podobieństwo [pkt]	106,35	198,33	277,23	342,41	402,51
Średnie podobieństwo [%]	98,79	97,45	94,86	91,86	89,54
Średnia dł. otrzymanej sekwencji	108,77	207,25	307,21	407,48	507,26
Średni czas obliczeń [ms]	121	577	1393	2793	4558
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	89,56	178,31	265,75	352,75	440,20
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	318,20	150,20	58,40	34,60	25,60
Jakość większa niż oczekiwana #	0,00	11,60	3,20	0,00	0,60
Idealna rekonstrukcja #	299,00	161,80	53,60	39,40	12,00
Średnie podobieństwo [pkt]	106,77	199,36	280,61	343,33	404,22
Średnie podobieństwo [%]	98,98	97,69	95,41	91,97	89,71
Średnia dł. otrzymanej sekwencji	108,74	208,11	307,46	407,47	507,41
Średni czas obliczeń [ms]	119	568	1379	2769	4484

wszystkim zredukować liczbę otrzymanych rozwiązań o jakości przewyższającej oczekiwaną, przy jednoczesnym wzroście liczby rozwiązań o jakości równej jakości oczekiwanej. Najprawdopodobniej jest to główna przyczyna dalszego wzrostu liczby idealnie zrekonstruowanych sekwencji.

Poszczególne algorytmy można porównać również pomiędzy sobą. Algorytm zachłanny jest bardzo prostą heurystyką. Przewaga pozostałych algorytmów nad nim jest oczywista, więc nie będzie on wykorzystywany w poniższym porównaniu.

Algorytmem generującym rozwiązania o najwyższej jakości jest algorytm przeszukiwania tabu. Rozwiązania otrzymane za jego pomocą zawierają największą liczbę elementów ze spektrum. Łączna liczba rozwiązań o jakości równej lub większej niż oczekiwana jest również największa w jego przypadku. Jest to algorytm, który najlepiej rozwiązuje zdefiniowany problem grafowy.

Tabela 9.11: Wyniki dla zestawu E (biblioteki klasyczne) - wielopoziomowy algorytm kolonii mrówek.

Długość sekwencji	109	209	309	409	509
Brak informacji o powtórzeniach					
Średnia jakość	85,47	171,30	257,00	342,77	427,73
Oczekiwana jakość	85,00	170,00	255,00	340,00	425,00
Rozwiązania o ocz. jakości #	236,20	43,60	14,80	4,60	11,00
Jakość większa niż oczekiwana #	163,00	354,20	382,80	392,40	356,40
Idealna rekonstrukcja #	96,00	17,80	0,00	0,00	0,00
Średnie podobieństwo [pkt]	101,51	192,07	274,82	345,82	398,48
Średnie podobieństwo [%]	96,57	95,95	94,47	92,28	89,14
Średnia dł. otrzymanej sekwencji	105,81	206,82	306,50	406,40	506,72
Średni czas obliczeń [ms]	77	287	651	1256	2135
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	89,95	178,28	266,20	354,88	443,69
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	353,40	151,00	110,60	118,40	127,80
Jakość większa niż oczekiwana #	17,80	33,40	11,20	25,00	20,80
Idealna rekonstrukcja #	293,80	123,00	49,00	53,20	20,00
Średnie podobieństwo [pkt]	106,95	202,73	287,09	362,54	433,47
Średnie podobieństwo [%]	99,06	98,50	96,45	94,32	92,58
Średnia dł. otrzymanej sekwencji	108,85	207,32	307,25	407,67	507,80
Średni czas obliczeń [ms]	84	291	651	1284	2124
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	89,97	179,24	267,57	355,75	444,46
Oczekiwana jakość	90,00	180,00	270,00	360,00	450,00
Rozwiązania o ocz. jakości #	392,00	244,80	158,20	150,00	161,40
Jakość większa niż oczekiwana #	0,00	18,00	11,80	3,80	2,00
Idealna rekonstrukcja #	307,40	203,80	90,40	55,40	17,40
Średnie podobieństwo [pkt]	107,19	204,66	290,80	365,80	432,93
Średnie podobieństwo [%]	99,17	98,96	97,06	94,72	92,53
Średnia dł. otrzymanej sekwencji	108,87	208,24	307,68	407,85	507,65
Średni czas obliczeń [ms]	94	342	782	1480	2437

Jednakże należy pamiętać, że głównym celem jest rozwiązanie problemu sekwencjonowania DNA przez hybrydyzację. Zastosowanie algorytmu przeszukiwania tabu umożliwia uzyskanie największej liczby idealnie zrekonstruowanych sekwencji, ale w przypadku algorytmu ACO średnie podobieństwo otrzymanych sekwencji jest nieco wyższe. Co więcej, czas obliczeń dla ACO wynosi mniej niż połowę czasu potrzebnego dla przeszukiwania tabu. Zastosowanie w ML-ACO idei wielopoziomowego udoskonalania rozwiązań prowadzi do znaczącego wzrostu ocen w porównaniu do ACO i do dalszej redukcji czasu obliczeń. W przypadku ML-ACO liczba idealnie zrekonstruowanych sekwencji oraz łączna liczba rozwiązań o jakości równej lub większej niż oczekiwana są już tylko nieco mniejsze niż w przypadku algorytmu przeszukiwania tabu. Co więcej, średnie podobieństwo otrzymanych za pomocą ML-ACO sekwencji jest już wyraźnie większe. Podsumowując, wielopoziomowy algorytm kolonii mrówek jest najszybszym algorytmem. Liczba idealnie odtworzonych przez niego sekwencji jest nieco niższa niż w przypadku algorytmu przeszukiwania tabu, ale ML-ACO generuje rozwiązania o najwyższym podobieństwie względem analizowanej sekwencji.

9.3.2 Biblioteki izotermiczne

9.3.2.1 Zestaw D

W pracy [14] oprócz wyników dla algorytmów przygotowanych dla klasycznego SBH przedstawiono również wyniki dla hybrydowego algorytmu genetycznego opracowanego dla bibliotek izotermicznych [17]. Niniejszy eksperyment ma na celu zarówno porównanie wyników tego algorytmu z wynikami algorytmów zaproponowanych w niniejszej pracy jak i porównanie z odpowiednimi wynikami dla bibliotek klasycznych, które przedstawiono w Rozdziale 9.3.1.2.

Wyniki zostały przedstawione w Tabeli 9.12 (zestaw D.3) oraz Tabeli 9.13 (zestaw D.4). Obie tabele zawierają następujące informacje: średnia jakość zrekonstruowanych sekwencji, średnia oczekiwana jakość rozwiązań, liczba otrzymanych rozwiązań o jakości równej oczekiwanej, liczba idealnie zrekonstruowanych sekwencji, średnie podobieństwo punktowe i procentowe oraz średni czas obliczeń. Są to te same kryteria, które zaprezentowano w tabelach dla zestawu D.1 i D.2 w Rozdziale 9.3.1.2. Obliczenia dla algorytmów prezentowanych w niniejszej pracy również wykonano dla każdej sekwencji 10 razy, a tabela zawiera wartości uśrednione. Wyniki dla hybrydowego algorytmu genetycznego pochodzą z [14] (gdy dana wartość jest niedostępna użyto znaku −), gdzie obliczenia dla danej sekwencji wykonano raz.

Dla zestawu D.3 najlepsze rozwiązania uzyskano przy użyciu algorytmu przeszukiwania tabu zaprezentowanego w Rozdziale 8.3. Generuje on największą liczbę rozwiązań o jakości równej oczekiwanej. Średnie podobieństwo otrzymanych sekwencji jest najwyższe, a liczba idealnie zrekonstruowanych sekwencji jest minimalnie niższa niż w przypadku ML-ACO opisanego w Rozdziale 8.5. Dla zestawu D.4 sekwencje o najwyższym podobieństwie otrzymano używając hybrydowego algorytmu genetycznego, dla którego liczba idealnie zrekonstruowanych sekwencji była tylko nieco niższa niż w przypadku ML-ACO.

Jednak najistotniejszą obserwacją jest to, że zastosowanie bibliotek izotermicznych dla tych samych sekwencji prowadzi do dużo gorszych wyników niż w przypadku stosowania klasycznych bibliotek o oligonukleotydach równej długości. W przypadku algorytmów opisanych w niniejszej pracy średnie podobieństwo spadło o kilkanaście procent, liczba idealnie zrekonstruowanych sekwencji z niecałych 20 spadła do zaledwie kilku, a

Tabela 9.12: Wyniki dla zestawu D.3 (biblioteki izotermiczne) - sekwencje o długości 600 nukleotydów z naturalnymi powtórzeniami, spektra zawierają jedynie błędy wynikające z powtórzeń.

Algorytm	Hybryd. genet. dla ISBH [17]	Nowa wersja tabu	Nowa wersja ML-ACO
Średnia jakość	756,30[a]	770,93	764,78
Średnia oczekiwana jakość	756,38[b]	771,20[c]	771,20[c]
Rozwiązania o ocz. jakości #	–	32,90/40	5,90/40
Idealna rekonstrukcja #	2,00/40	3,90/40	4,00/40
Średnie podobieństwo [pkt]	–	356,71	334,44
Średnie podobieństwo [%]	78,34	79,73	77,87
Średni czas obliczeń [ms]	– [d]	7168	12105

Tabela 9.13: Wyniki dla zestawu D.4 (biblioteki izotermiczne) - sekwencje o długości 600 nukleotydów z naturalnymi powtórzeniami, spektra zawierają 5% błędów pozytywnych oraz łącznie 5% błędów negatywnych obu rodzajów.

Algorytm	Hybryd. genet. dla ISBH [17]	Nowa wersja tabu	Nowa wersja ML-ACO
Średnia jakość	732,68[a]	739,39	742,48
Średnia oczekiwana jakość	732,68[b]	747,50[c]	747,50[c]
Rozwiązania o ocz. jakości #	–	2,50/40	9,80/40
Idealna rekonstrukcja #	3,00/40	2,50/40	3,10/40
Średnie podobieństwo [pkt]	–	279,92	334,29
Średnie podobieństwo [%]	79,12	73,33	77,86
Średni czas obliczeń [ms]	– [d]	21482	15249

[a] oszacowane na podstawie procentowego użycia oligonukleotydów ze spektrum przedstawionego w [14] oraz średniej oczekiwanej jakości
[b] rozmiar idealnego spektrum pomniejszony o liczbę losowych negatywnych błędów hybrydyzacji
[c] rozmiar idealnego multispektrum pomniejszony o liczbę losowych negatywnych błędów hybrydyzacji
[d] nie porównywalne ze względu na obliczenia na innej maszynie

czas obliczeń wydłużył się kilkukrotnie. Przyczyniają się do tego najprawdopodobniej dwa czynniki. Pierwszy to znacząco większa średnia liczba powtórzeń w przypadku wykorzystania bibliotek izotermicznych. Dla bibliotek klasycznych średnio wynosiła ona ok. 4, a w przypadku bibliotek izotermicznych ok. 16. Drugim istotnym czynnikiem jest dużo większy rozmiar idealnego multispektrum. Dla bibliotek klasycznych idealne multispektrum zawierało 591 oligonukleotydów, podczas gdy dla bibliotek izotermicznych średnia wynosiła 771,20.

9.3.2.2 Zestaw E

Ostatni z przeprowadzonych eksperymentów miał na celu ocenę wpływu dodatkowej informacji o powtórzeniach na rekonstrukcję sekwencji w przypadku stosowania bibliotek izotermicznych. Wyniki zostały zaprezentowane w Tabeli 9.14 (algorytm zachłanny), Tabeli 9.15 (algorytm przeszukiwania tabu), Tabeli 9.16 (algorytm kolonii mrówek - ACO)

Tabela 9.14: Wyniki dla zestawu E (biblioteki izotermiczne) - algorytm zachłanny.

Długość sekwencji	109	209	309	409	409
Brak informacji o powtórzeniach					
Średnia jakość	97,14	198,48	306,88	409,84	511,66
Oczekiwana jakość	109,20	223,58	352,70	474,13	596,50
Rozwiązania o ocz. jakości #	4,80	0,00	0,00	0,00	0,00
Jakość większa niż oczekiwana #	6,20	0,00	0,00	0,00	0,00
Idealna rekonstrukcja #	20,80	6,00	2,00	0,20	0,00
Średnie podobieństwo [pkt]	83,23	131,81	188,67	228,80	258,10
Średnie podobieństwo [%]	88,18	81,53	80,53	77,97	75,35
Średnia dł. otrzymanej sekwencji	106,60	206,94	307,09	407,30	507,22
Średni czas obliczeń [ms]	3	21	61	128	221
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	100,65	202,85	312,66	419,18	529,76
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	0,60	0,00	0,00	0,00	0,00
Jakość większa niż oczekiwana #	0,00	0,00	0,00	0,00	0,00
Idealna rekonstrukcja #	59,60	31,40	12,80	15,20	3,00
Średnie podobieństwo [pkt]	85,47	134,19	195,53	241,17	275,90
Średnie podobieństwo [%]	89,20	82,10	81,64	79,48	77,10
Średnia dł. otrzymanej sekwencji	107,81	207,39	307,52	407,65	507,77
Średni czas obliczeń [ms]	4	22	63	130	231
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	100,83	202,09	313,13	419,46	529,32
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	0,60	0,00	0,00	0,00	0,00
Jakość większa niż oczekiwana #	0,00	0,00	0,00	0,00	0,00
Idealna rekonstrukcja #	71,00	29,40	16,40	14,60	3,40
Średnie podobieństwo [pkt]	86,14	134,38	196,19	241,43	277,87
Średnie podobieństwo [%]	89,51	82,15	81,75	79,51	77,30
Średnia dł. otrzymanej sekwencji	108,23	207,62	307,71	407,79	507,85
Średni czas obliczeń [ms]	4	21	62	129	227

oraz Tabeli 9.17 (wielopoziomowy algorytm kolonii mrówek - ML-ACO). Wszystkie powyższe tabele składają się z trzech części. W pierwszej zaprezentowano wyniki dla przypadku braku dostępności informacji o powtórzeniach, w drugiej przedstawiono wyniki dla częściowej informacji o powtórzeniach typu "jeden i wiele", a ostatnia zawiera wyniki dla modelu częściowej informacji o powtórzeniach typu "jeden, dwa i wiele". Każda z części zawiera następujące wartości: średnią jakość otrzymanych sekwencji, oczekiwaną jakość, liczbę rozwiązań o jakości równej oczekiwanej, liczbę rozwiązań o jakości wyższej niż oczekiwana, liczbę idealnie zrekonstruowanych rozwiązań, średnie podobieństwo punktowe i procentowe, średnią długość otrzymanej sekwencji oraz średni czas obliczeń.

Pewne kroki algorytmów opisanych w niniejszej pracy mają charakter losowy, więc obliczenia przeprowadzono 5 razy dla każdego spektrum danej sekwencji, a tabele zawierają wartości uśrednione. Dla danej długości zestaw E zawiera 40 różnych sekwencji,

Tabela 9.15: Wyniki dla zestawu E (biblioteki izotermiczne) - algorytm przeszukiwania tabu.

Długość sekwencji	109	209	309	409	409
Brak informacji o powtórzeniach					
Średnia jakość	110,11	224,89	354,09	475,23	595,51
Oczekiwana jakość	109,20	223,58	352,70	474,13	596,50
Rozwiązania o ocz. jakości #	131,00	48,60	44,40	32,20	22,60
Jakość większa niż oczekiwana #	248,80	309,60	294,80	277,60	211,20
Idealna rekonstrukcja #	76,00	9,20	7,00	0,40	0,00
Średnie podobieństwo [pkt]	91,86	149,14	213,17	265,69	299,86
Średnie podobieństwo [%]	92,14	85,68	84,49	82,48	79,46
Średnia dł. otrzymanej sekwencji	107,87	208,03	308,25	408,32	508,47
Średni czas obliczeń [ms]	714	2571	5925	10931	17511
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	117,72	237,89	370,68	497,64	624,93
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	319,00	118,40	81,40	65,40	46,00
Jakość większa niż oczekiwana #	35,40	60,00	52,00	33,60	15,60
Idealna rekonstrukcja #	253,80	97,60	75,80	49,60	12,80
Średnie podobieństwo [pkt]	100,74	156,12	224,10	283,19	331,10
Średnie podobieństwo [%]	96,21	87,35	86,26	84,62	82,52
Średnia dł. otrzymanej sekwencji	108,89	208,75	308,74	408,70	508,74
Średni czas obliczeń [ms]	816	2971	6920	12663	20748
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	117,59	237,68	370,85	497,76	624,87
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	344,40	162,80	97,00	68,20	44,20
Jakość większa niż oczekiwana #	10,40	31,40	40,20	26,80	6,00
Idealna rekonstrukcja #	260,00	117,60	85,60	48,80	13,60
Średnie podobieństwo [pkt]	101,46	159,79	224,80	280,66	328,46
Średnie podobieństwo [%]	96,54	88,23	86,37	84,31	82,27
Średnia dł. otrzymanej sekwencji	108,90	208,78	308,75	408,71	508,68
Średni czas obliczeń [ms]	792	2864	6649	12095	19570

Tabela 9.16: Wyniki dla zestawu E (biblioteki izotermiczne) - algorytm kolonii mrówek.

Długość sekwencji	109	209	309	409	409
Brak informacji o powtórzeniach					
Średnia jakość	109,68	224,25	351,36	469,26	585,42
Oczekiwana jakość	109,20	223,58	352,70	474,13	596,50
Rozwiązania o ocz. jakości #	126,20	67,80	43,60	29,60	13,60
Jakość większa niż oczekiwana #	205,80	247,00	159,40	88,80	50,40
Idealna rekonstrukcja #	80,40	11,60	9,40	0,80	0,00
Średnie podobieństwo [pkt]	93,50	150,95	224,42	275,78	325,85
Średnie podobieństwo [%]	92,89	86,11	86,31	83,71	82,01
Średnia dł. otrzymanej sekwencji	106,93	206,57	306,90	406,86	507,12
Średni czas obliczeń [ms]	238	1220	3498	6217	10282
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	115,95	232,77	361,32	484,10	606,55
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	135,40	16,80	2,20	0,00	0,00
Jakość większa niż oczekiwana #	5,80	1,00	1,20	0,00	0,00
Idealna rekonstrukcja #	206,20	81,40	22,40	32,60	14,00
Średnie podobieństwo [pkt]	99,85	160,94	232,37	289,37	349,92
Średnie podobieństwo [%]	95,80	88,50	87,60	85,37	84,37
Średnia dł. otrzymanej sekwencji	108,48	208,05	307,87	407,90	507,96
Średni czas obliczeń [ms]	247	1195	3470	6626	11468
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	116,25	233,77	362,34	485,45	608,99
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	180,40	34,40	1,60	0,00	0,00
Jakość większa niż oczekiwana #	4,40	2,20	0,00	0,00	0,00
Idealna rekonstrukcja #	225,80	95,60	40,20	43,60	14,60
Średnie podobieństwo [pkt]	100,52	167,06	232,09	288,03	348,61
Średnie podobieństwo [%]	96,11	89,97	87,55	85,21	84,24
Średnia dł. otrzymanej sekwencji	108,68	207,88	307,77	407,87	508,01
Średni czas obliczeń [ms]	241	1139	3222	5981	10753

Tabela 9.17: Wyniki dla zestawu E (biblioteki izotermiczne) - wielopoziomowy algorytm kolonii mrówek.

Długość sekwencji	109	209	309	409	409
Brak informacji o powtórzeniach					
Średnia jakość	110,26	225,57	355,50	477,54	599,79
Oczekiwana jakość	109,20	223,58	352,70	474,13	596,50
Rozwiązania o ocz. jakości #	138,00	36,80	26,60	27,40	20,00
Jakość większa niż oczekiwana #	259,60	359,60	365,20	354,20	325,80
Idealna rekonstrukcja #	89,00	12,20	9,80	3,20	2,00
Średnie podobieństwo [pkt]	94,79	160,78	234,39	294,83	354,16
Średnie podobieństwo [%]	93,48	88,46	87,93	86,04	84,79
Średnia dł. otrzymanej sekwencji	107,00	206,75	307,06	407,05	507,36
Średni czas obliczeń [ms]	242	1096	2997	5913	10591
Informacja o powtórzeniach typu "jeden i wiele"					
Średnia jakość	117,30	236,23	368,65	495,47	623,10
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	275,00	78,80	41,00	31,40	21,00
Jakość większa niż oczekiwana #	16,20	13,40	13,20	3,80	4,20
Idealna rekonstrukcja #	240,60	98,40	38,80	38,60	17,40
Średnie podobieństwo [pkt]	102,05	169,90	245,90	308,25	382,96
Średnie podobieństwo [%]	96,81	90,65	89,79	87,68	87,62
Średnia dł. otrzymanej sekwencji	108,60	208,08	308,00	408,03	508,05
Średni czas obliczeń [ms]	282	1344	3381	6720	12089
Informacja o powtórzeniach typu "jeden, dwa i wiele"					
Średnia jakość	117,53	237,02	369,74	496,78	624,78
Oczekiwana jakość	117,85	239,88	374,95	504,48	634,63
Rozwiązania o ocz. jakości #	333,60	123,40	75,00	67,80	50,60
Jakość większa niż oczekiwana #	6,00	6,80	0,60	3,00	0,00
Idealna rekonstrukcja #	262,00	110,20	56,20	51,20	21,60
Średnie podobieństwo [pkt]	102,26	173,74	244,63	308,62	383,11
Średnie podobieństwo [%]	96,91	91,57	89,58	87,73	87,63
Średnia dł. otrzymanej sekwencji	108,81	207,85	307,99	407,92	508,12
Średni czas obliczeń [ms]	287	1245	3419	6854	12141

dla każdej z nich przygotowano 10 spektr, więc tabele przedstawiają uśrednione wyniki dla 2000 uruchomień danego algorytmu. Maksymalna wartość oceny dla kryteriów prezentujących pewną liczbę rozwiązań wynosi 400.

Otrzymane wyniki potwierdzają pozytywny wpływ na rekonstrukcję sekwencji nawet częściowej informacji o powtórzeniach. Dla wszystkich algorytmów zastosowanie jej prowadzi do otrzymania znacząco lepszych wyników. Podobnie jak w przypadku bibliotek klasycznych najbardziej zauważalny jest ogromny wzrost liczby idealnie zrekonstruowanych sekwencji. Stosowanie dokładniejszego modelu w przypadku bibliotek izotermicznych również prowadzi do uzyskania lepszych rezultatów, choć podobnie jak w przypadku bibliotek klasycznych różnice pomiędzy rozpatrywanymi modelami są znacząco mniejsze niż w porównaniu do sytuacji bez dostępnej informacji o powtórzeniach i maleją wraz ze wzrostem długości sekwencji.

Algorytmem najlepiej rozwiązującym problem obliczeniowy jest również algorytm przeszukiwania tabu. Pozwala on na osiągnięcie najwyższej jakości rozwiązań, a łączna liczba rozwiązań o jakości równej lub wyższej niż oczekiwana jest dla niego największa.

Temperatury topnienia bibliotek izotermicznych zostały tak dobrane ($T_L = 26$℃ i $T_L + 2$℃$= 28$℃), aby uzyskać łączną liczbę oligonukleotydów w obu bibliotekach zbliżoną do liczby elementów w bibliotece l-merów dla $l = 10$. Pozwala to również porównać otrzymane wyniki z rezultatami dla klasycznych bibliotek, które zostały zaprezentowane w Rozdziale 9.3.1.3. Niestety, wyniki dla bibliotek izotermicznych są znacząco gorsze niż w przypadku bibliotek zawierających oligonukleotydy o równiej długości. Stosując klasyczne biblioteki można uzyskać lepszą rekonstrukcję sekwencji. Obserwacja ta jest spójna z wnioskami wyciągniętymi z eksperymentów opisanych w Rozdziale 9.2.2.1 Rozdziale 9.3.2.1.

Rozdział 10

Podsumowanie

W niniejszej rozprawie rozważano jedną z metod sekwencjonowania DNA, tj. sekwencjonowanie przez hybrydyzację. Metoda ta składa się z dwóch etapów. Pierwszy z nich to eksperyment biochemiczny, w ramach którego uzyskuje się spektrum. Drugi etap sprowadza się do rozwiązania pewnego problemu obliczeniowego. Jednym z podstawowych problemów związanych z tą metodą są błędy pojawiające się w trakcie eksperymentu biochemicznego. Jednym ze źródeł błędów są powtarzające się oligonukleotydy w analizowanej sekwencji.

Aktualnie nie ma technologicznych możliwości pozyskania dokładnej informacji o liczbie powtórzeń poszczególnych oligonukleotydów wchodzących w skład badanej sekwencji, ale można próbować określić ją w sposób przybliżony. Rozważono dwa modele częściowej informacji o powtórzeniach. Pierwszy z nich zakłada, że istnieje możliwość ustalenia, czy dany oligonukleotyd występuje w sekwencji raz czy wiele razy. Drugi model przyjmuje możliwość określenia, czy dany oligonukleotyd występuje w analizowanej sekwencji dokładnie raz, dokładnie dwa razy czy też przynajmniej trzy razy.

Dla obu powyższych modeli częściowej informacji o powtórzeniach zaproponowano następujące algorytmy: zachłanny, przeszukiwania tabu oraz dwa algorytmy inspirowane funkcjonowaniem kolonii mrówek. Zostały one zaprojektowane w taki sposób, aby umożliwić rozwiązywanie problemów zarówno dla klasycznych bibliotek oligonukleotydów jak i bibliotek izotermicznych. Zaproponowane algorytmy zostały przetestowane w pierwszej kolejności przy użyciu istniejących zestawów sekwencji. Wyniki eksperymentów wykazały, że nowe algorytmy umożliwiają uzyskanie przynajmniej tak dobrych, a w wielu przypadkach nawet znacząco lepszych wyników w porównaniu do istniejących algorytmów przygotowanych dla klasycznego lub izotermicznego SBH.

Zaproponowane algorytmy zostały również przetestowane przy użyciu specjalnie przygotowanego nowego zbioru rzeczywistych sekwencji zawierających naturalne powtórzenia. Przeprowadzone badania wykazały, że wykorzystanie nawet częściowej informacji o powtórzeniach prowadzi do znacząco lepszej rekonstrukcji sekwencji zarówno w przypadku stosowania klasycznych bibliotek oligonukleotydów jak i bibliotek izotermicznych. Co więcej, wykorzystanie dokładniejszego modelu informacji prowadzi do jeszcze lepszych rezultatów, ale różnice pomiędzy rozpatrywanymi modelami nie są już tak znaczące, jak w przypadku porównania z podejściami bez jakiejkolwiek informacji o powtórzeniach.

Do oceny wykorzystano wiele kryteriów, ale szczególnie spektakularne wyniki otrzymano dla liczby odtworzonych sekwencji, które były identyczne z analizowaną sekwencją. W przypadku stosowania klasycznych bibliotek oligonukleotydów dla najkrótszych

sekwencji o długości 109 nukleotydów dzięki wykorzystaniu dodatkowej informacji o powtórzeniach udało się dokładnie odtworzyć co trzecią sekwencję, a dla sekwencji o długości 209 nukleotydów co drugą. Przy braku informacji o powtórzeniach udawało się to odpowiednio jedynie dla mniej niż 20% i ok. 4% sekwencji. Aktualnie metoda SBH nie jest wykorzystywana na szeroką skalę, ale przedstawione wyniki mogą zachęcić do jej dalszego rozwijania i wykorzystywania w resekwencjonowaniu.

W kontekście powyższych badań rozpatrywano również modele grafowe reprezentujące problemy obliczeniowe sekwencjonowania DNA przez hybrydyzację z dodatkową informacją o powtórzeniach. Wybrany model będący podstawą opracowanych algorytmów przybliżonych umożliwia rozwiązywanie zarówno problemów zdefiniowanych dla bibliotek klasycznych jak i bibliotek izotermicznych.

W pracy rozważano również algorytm aproksymacyjny dla klasycznego problemu sekwencjonowania przez hybrydyzację z błędami dowolnego typu, który odpowiada grafowemu problemowi Orienteering. Wykazano, że dla problemu Orienteering w grafie skierowanym istnieje algorytm aproksymacyjny o współczynniku aproksymacji, który nie zależy ani od rozmiaru instancji problemu ani od wartości optymalnej. Co więcej, gdy dla danego problemu iloraz największego kosztu łuku do najmniejszego kosztu łuku w grafie jest ograniczony pewną stałą, to istnieje dla niego algorytm aproksymacyjny o stałym współczynniku aproksymacji. Przykładowo takim problemem jest właśnie klasyczne SBH, dla którego maksymalna wartość tego ilorazu jest równa l, tj. długości oligonukleotydów wykorzystywanej biblioteki. Niestety, gwarancja dokładności w przypadku SBH jest na tyle mała, że algorytm aproksymacyjny ma dla tego problemu zastosowanie wyłącznie teoretyczne. Dalsze prace badawcze w tym obszarze miałyby na celu polepszenie gwarancji dokładności.

W przypadku SBH z dodatkową informacją o powtórzeniach kolejne prace mogą być związane zarówno z dalszym rozwojem algorytmów jak i weryfikacją opisanych w rozprawie algorytmów za pomocą danych otrzymanych w rzeczywistych eksperymentach biochemicznych. Zastosowanie rzeczywistych danych pochodzących z chipów DNA pozwoliłoby na empiryczne potwierdzenie przydatności dodatkowej informacji o powtórzeniach. W przypadku resekwencjonowania możliwe jest wykorzystanie ustalonej już wcześniej sekwencji homologicznej. Rozwój algorytmów miałby na celu uwzględnianie tej dodatkowej informacji.

Bibliografia

[1] E. M. Arkin, J. S. B. Mitchell, G. Narasimhan. Resource-constrained geometric network optimization. *Proceedings of ACM SoCG*, strony 307–316, 1998.

[2] S. Arora, G. Karakostas. A $2 + \epsilon$ approximation algorithm for the k-mst problem. *Proceedings of ACM-SIAM SODA*, strony 754–759, 2000.

[3] B. Awerbuch, Y. Azar, A. Blum, S. Vempala. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262, 1999.

[4] W. Bains, G. C. Smith. A novel method for nucleic acid sequence determination. *Journal of Theoretical Biology*, 135(3):303–307, 1988.

[5] N. Bansal, A. Blum, S. Chawla, A. Meyerson. Approximation algorithms for deadline-tsp and vehicle routing with time-windows. *Proceedings of ACM STOC*, strony 166–174, 2004.

[6] A. Ben-Dor, I. Pe'er, R. Shamir, R. Sharan. On the complexity of positional sequencing by hybridization. *Journal of Computational Biology*, 8(4):361–371, 2002.

[7] J. Błażewicz. *Złożoność obliczeniowa problemów kombinatorycznych.* Biblioteka Inżynierii Oprogramowania. Wydawnictwa Naukowo-Techniczne, 1988.

[8] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz. Sequencing by hybridization with isothermic oligonucleotide libraries. *Discrete Applied Mathematics*, 145(1):40–51, 2004.

[9] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, A. Świercz. Tabu search algorithm for DNA sequencing by hybridization with isothermic libraries. *Computational Biology and Chemistry*, 28(1):11–19, 2004.

[10] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, J. Weglarz. Tabu search for DNA sequencing with false negatives and false positives. *European Journal of Operational Research*, 125(2):257–265, 2000.

[11] J. Błażewicz, P. Formanowicz, M. Kasprzak, W. T. Markiewicz, J. Węglarz. DNA sequencing with positive and negative errors. *Journal of Computational Biology*, 6(1):113–123, 1999.

[12] J. Błażewicz, F. Glover, M. Kasprzak. DNA sequencing–tabu and scatter search combined. *INFORMS Journal on Computing*, 16(3):232–240, 2004.

[13] J. Błażewicz, F. Glover, M. Kasprzak. Evolutionary approaches to DNA sequencing with errors. *Annals of Operations Research*, 138(1):67–78, 2005.

[14] J. Błażewicz, F. Glover, M. Kasprzak, W. T. Markiewicz, C. Oguz, D. Rebholz-Schuhmann, A. Świercz. Dealing with repetitions in sequencing by hybridization. *Computational Biology and Chemistry*, 30(5):313–320, 2006.

[15] J. Błażewicz, M. Kasprzak. Complexity of DNA sequencing by hybridization. *Theoretical Computer Science*, 290(3):1459–1473, 2003.

[16] J. Błażewicz, M. Kasprzak, W. Kuroczycki. Hybrid genetic algorithm for DNA sequencing with errors. *Journal of Heuristics*, 8(5):495–502, 2002.

[17] J. Błażewicz, C. Oguz, A. Świercz, J. Węglarz. DNA sequencing by hybridization via genetic search. *Operations Research*, 54(6):1185–1192, 2006.

[18] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, M. Minkoff. Approximation algorithms for orienteering and discounted-reward TSP. *SIAM Journal on Computing*, 37(2):653–670, 2007.

[19] A. Blum, R. Ravi, S. Vempala. A constant-factor approximation algorithm for the k-MST problem. *Journal of Computer and System Sciences*, 58(1):101–108, 1999.

[20] C. Blum, M. Y. Vallès, M. J. Blesa. An ant colony optimization algorithm for DNA sequencing by hybridization. *Computers and Operations Research*, 35(11):3620–3635, 2008.

[21] N. E. Broude, T. Sano, C. L. Smith, C. R. Cantor. Enhanced DNA sequencing by hybridization. *Proceedings of the National Academy of Sciences of the USA*, (91):3071–3076, 1994.

[22] K. Chaudhuri, B. Godfrey, S. Rao, K. Talwar. Paths, trees, and minimum latency tours. *Proceedings of IEEE FOCS*, strona 36, 2003.

[23] C. Chekuri, N. Korula, M. Pál. Improved algorithms for orienteering and related problems. *Proceedings of ACM-SIAM SODA*, strony 661–670, 2008.

[24] K. Chen, S. Har-Peled. The orienteering problem in the plane revisited. *Proceedings of ACM SoCG*, strony 247–254, 2006.

[25] F. Crick. On protein synthesis. *Symposia of the Society for Experimental Biology*, XII:139–163, 1958.

[26] F. Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970.

[27] M. Dorigo. *Optimization, Learning and Natural Algorithms*. Praca doktorska, Politecnico di Milano, 1992.

[28] P. Formanowicz. *Selected combinatorial aspects of biological sequence analysis*. Publishing House of Poznan University of Technology, Poznan, 2005.

[29] P. Formanowicz. DNA sequencing by hybridization with additional information available. *Computational Methods in Science and Technology*, 11(1):21–29, 2005.

[30] P. Formanowicz. Isothermic sequencing by hybridization problems with information about repetitions. *Przegląd Elektrotechniczny*, 84(9):103–107, 2008.

[31] N. Garg. A 3-approximation for the minimum tree spanning k vertices. *Proceedings of IEEE FOCS*, strona 302, 1996.

[32] N. Garg. Saving an epsilon: a 2-approximation for the k-MST problem in graphs. *Proceedings of ACM STOC*, strony 396–402, 2005.

[33] F. Glover, M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Boston, Masachusetts, 1997.

[34] J. P. Gogarten, A. G. Senejani, O. Zhaxybayeva, L. Olendzenski, E. Hilario. Inteins: Structure, function, and evolution. *Annual Review of Microbiology*, 56(1):263–287, 2002.

[35] B. L. Golden, L. Levy, R. Vohra. The orienteering problem. *Naval Research Logistics*, 34(3):307–318, 1987.

[36] S. Hannenhalli, W. Feldman, H. F. Lewis, S. Skiena, P. A. Pevzner. Positional sequencing by hybridization. *Computer Applications in the Biosciences*, 12(1):19–24, 1996.

[37] M. Kasprzak. An algorithm for isothermic DNA sequencing. *Bulletin of the Polish Academy od Sciences, Technical Sciences*, 52(1):31–35, 2004.

[38] S. Kruglyak. Multistage sequencing by hybridization. *Journal of Computational Biology*, 5(1):165–171, 1998.

[39] R. Kumar, H. Li. On asymmetric TSP: Transformation to symmetric TSP and performance bound. *http://www.ece.iastate.edu/∼rkumar/PUBS/atsp.pdf*, 2002.

[40] K. Kwarciak, P. Formanowicz. Tabu and scatter search algorithm for DNA sequencing by hybridization with multiplicity information available. *w druku*.

[41] K. Kwarciak, P. Formanowicz. A greedy algorithm for the DNA sequencing by hybridization with positive and negative errors and information about repetitions. *Bulletin of the Polish Academy od Sciences, Technical Sciences*, 51(1):111–115, 2011.

[42] K. Kwarciak, M. Radom, P. Formanowicz. DNA sequencing with negative errors and information about repetitions. *Zeszyty Naukowe Politechniki Śląskiej*, 151:215–222, 2008.

[43] L. Liu, Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu, M. Law. Comparison of next-generation sequencing systems. *Journal of Biomedicine & Biotechnology*, 2012:1–11, 2012.

[44] Y. P. Lysov, V. L. Florentiev, A. A. Khorlin, K. KR, V. V. Shik, A. D. Mirzabekov. Determination of the nucleotide sequence of DNA using hybridization with oligonucleotides. a new method. *Doklady Academy Nauk SSSR*, 303(6):1508–1511, 1988.

[45] D. Margaritis, S. S. Skiena. Reconstructing strings from substrings in rounds. *Proceedings of IEEE FOCS*, strony 613–620, 1995.

[46] M. Margulies, M. Egholm, W. E. Altman, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[47] A. M. Maxam, W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, 74(2):560–4, 1977.

[48] B. J. McCarthy, J. J. Holland. Denatured DNA as a direct template for in vitro protein synthesis. *Proceedings of the National Academy of Sciences of the United States of America*, 54(3):880–886, 1965.

[49] V. Nagarajan, R. Ravi. Poly-logarithmic approximation algorithms for directed vehicle routing problems. *Proceedings of APPROX*, strony 257–270, 2007.

[50] C. Papadimitriou. *Złożoność obliczeniowa*. Klasyka Informatyki. Wydawnictwa Naukowo-Techniczne, 2002.

[51] A. C. Pease, D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes, S. P. Fodor. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proceedings of the National Academy of Science of the USA*, wolumen 91, strony 5022–5026, 1994.

[52] P. A. Pevzner. l-Tuple DNA sequencing: computer analysis. *Journal of Biomolecular Structure and Dynamics*, 7(1):63–73, 1989.

[53] P. A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. The MIT Press, Cambridge, Massachusetts, 2000.

[54] P. A. Pevzner, R. J. Lipshutz. Towards DNA sequencing chips. *Proceedings of the 19th International Symposium on Mathematical Foundations of Computer Science*, strony 143–158. Springer, 1994.

[55] A. Pihlak, G. Baurén, E. Hersoug, P. Lönnerberg, A. Metsis, S. Linnarsson. Rapid genome sequencing with short universal tiling probes. *Nature Biotechnology*, 26:676–684, 2008.

[56] M. Radom. *Kombinatoryczne aspekty nieklasycznego sekwencjonowania przez hybrydyzację*. Praca doktorska, Wydział Informatyki Politechniki Poznańskiej, Poznań 2012.

[57] M. Ronaghi, S. Karamohamed, B. Pettersson, M. Uhlén, P. Nyrén. Real-time DNA sequencing using detection of pyrophosphate release. *Analytical Biochemistry*, 242(1):84–89, 1996.

[58] M. Ronaghi, M. Uhlén, P. Nyrén. A Sequencing Method Based on Real-Time Pyrophosphate. *Science*, 281(5375):363–365, 1998.

[59] F. Sanger, A. R. Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, 94(3):441–448, 1975.

[60] F. Sanger, S. Nicklen, A. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of The National Academy of Sciences of The United States Of America*, 74(12):5463–5467, 1977.

[61] M. Schena. *Microarray Analysis*. Wiley-Liss, Hoboken, New Jersey, 2003.

[62] P. Schuurman, G. J. Woeginger. Approximation schemes - a tutorial. Wstępna wersja rozdziału książki "Lectures on Scheduling", R.H. Möhring, C.N. Potts, A.S. Schulz, G.J. Woeginger, L.A. Wolsey (red.),
http://www.win.tue.nl/∼gwoegi/papers/ptas.pdf.

[63] S. Skiena, G. Sundaram. Reconstructing strings from substrings. *Journal of Computational Biology*, 2(2):333–353, 1995.

[64] T. Uzawa, A. Yamagishi, T. Oshima. Polypeptide synthesis directed by DNA as a messenger in cell-free polypeptide synthesis by extreme thermophiles. *The Journal of Biochemistry*, 131(6):849–53, 2002.

[65] C. Walshaw. Multilevel refinement for combinatorial optimisation problems. *Annals of Operations Research*, 131:325–372, 2004.

[66] M. S. Waterman. *Introduction to computational biology - maps, sequences, and genomes: interdisciplinary statistics.* CRC Press, 1995.

[67] J. D. Watson, F. H. Crick. Genetical implications of the structure of deoxyribonucleic acid. *Nature*, 171(4361):964–967, 1953.

[68] D. A. Wheeler, M. Srinivasan, M. Egholm, et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, 2008.

[69] J.-H. Zhang, L.-Y. Wu, Y.-Y. Zhao, X.-S. Zhang. An optimization approach to the reconstruction of positional DNA sequencing by hybridization with errors. *European Journal of Operational Research*, 182(1):413–427, 2007.