PRZEDMIOT OBIERALNY: BIOINFORMATYKA, LABORATORIUM

Wersja pliku: v 1.5, 27 marca 2025
Autor: dr inż. Marcin Radom

PROJEKT – ZALICZENIE - OGÓLNE ZASADY

- Praca w zespole dwuosobowym
- Sprawozdanie główne:
    - Należy zacząć od opisu algorytmu. Proszę się przede wszystkim skoncentrować na opisie algorytmu, czy to jest moja wersja z pliku z materiałami, czy zmieniona (jak i gdzie), czy zupełnie inny pomysł.
        - Opisy generatora instancji można sobie podarować, chyba że jakimś cudem ktoś wymyślił tam jakąś ciekawą modyfikację, którą chce się pochwalić (doubt…)
        - Opis modułu tworzenia grafu – jak wyżej. Jak to jest tak jak opisywałem, opisu tego proszę nie robić. Chyba, że coś w tym module projektu też było zrobione „oryginalnie.
    - (Metaheurystyka) proszę o dokładny opis, np. jeśli algorytm genetyczny, to chciałbym dokładnie się dowiedzieć, jak działają krzyżowanie, mutacja, selekcja. Jeśli to Tabu Search, to znowu: jak działa lista tabu, czy jest kryterium aspiracji, czy algorytm ma jakieś sposoby radzenie sobie z optimami lokalnymi w które TS lubi wpadać, itd. Tak samo w przypadku innych algorytmów, zarówno tych typowych i takich bardziej autorskich – chciałbym mieć szansę zrozumieć, jak działa algorytm bez potrzeby dokładnej analizy linia po linii kodu programu.
    - Testy i wyniki testów w formie tabel i wykresów (o tym obszernie dalej, głównie od jakości tej części będzie zależeć wstępna opinia / ocena projektu).
- Sprawozdania w pdf/doc/odf – można wysyłać mailem oraz kod projektu (jako całość, uwaga na pliki wykonywalne w załącznikach (niech ich po prostu nie będzie) i jeśli nie potwierdzę w ciągu 24-48h maks, że dostałem maila (jak zawsze robię) to można założyć, że mail zaginął – zazwyczaj przez filtry poczty i załączniki).
- Język programowania: preferowane C++, C#, Java, Python.

TERMIN: DNI I GODZINY ZALICZENIA PROJEKTU

- W niedzielę 15.06 grupy L2 i L4 mają oczywiście priorytet, bo to ich zajęcia. 15.06 będę w pracy od około 9:00 do 16:0 / ostatniej umówionej grupy. W praktyce, gdyby wszyscy z L2 i L4 oddali sprawozdania na czas (o tym niżej) to i tak rozmowy z nimi zajmą mi mniej niż połowę przewidzianego czasu. Więc grupy L1, L3 i L5 mogą w ten dzień przychodzić, jeżeli znajdą czas pomiędzy innymi zajęciami. Zapowiedzenie się wcześniejsze mile widziane (nie dotyczy L2, L4, ci niech w mailu co najwyżej zapowiedzą, że się **nie** zjawią i chcą Zoom – patrz niżej).
- Aby rozmowa o projekcie mogła bez problemu się odbyć, chciałbym dostać sprawozdanie do 9/10.06, czyli tak do północy/do rana we wtorek 10.06.
    - Wszystkim którzy do tego czasu wyślą projekty będę najpierw pisał maila, że odebrałem (jak nie przyjdzie, to znaczy, że mail nie dotarł, bo to odpisuję priorytetowo), a po 1-2 dniach kolejny, gdzie na niedzielę 15.06 porozdzielam was czasowo co około kwadrans, żebyście bez sensu nie musieli stać w jakiejś kolejce pod salą…

    - Jeżeli dostanę sprawozdanie między 10.06 a 12.06 – to zależy. Jak zdążę przed piątkiem przeczytać, to można przychodzić w niedzielę.
- **W sobotę 14.06 – zajęcia odwołane**. <gothic mode> W związku z powyższym ogłasza się co następuje. Z woli miłościwie panującego nam Dziekanatu, oceny muszą być wpisane w usos do 22.06. </gothic> Czyli możemy „rozmowę" mieć na zoomie. W związku z tym, takie sprawozdania musze dostać **najpóźniej** do soboty 14.06 do północy. Dzięki czemu, w niedzielę zamiast rozmawiać z wami, nie będę się na PP nudzić, tylko drukować i czytać. W poniedziałek 16.06 wyślę maile z linkami do Zooma. Jak mam być szczery, to wolałbym to zunifikować. Na przykład od razu ustalmy, że dobrze byłoby rozmowę na Zoomie przeprowadzać np. w środę 18.06 lub czwartek 19.06 około wieczornej godziny 18 i później. Po prostu nie chciałbym stawać przed koniecznością generowania 15 spotkań na zoom rozsianych po wszystkich dniach i godzinach…

TESTY

Ogólnie dzielą się na dwa rodzaje: testy w ramach strojenia metaheurystyki oraz testy jakości sekwencjonowania DNA.

TESTOWANIE PARAMETRÓW ALGORYTMU

Gdy mówimy o algorytmach metaheurystycznych (czy także innych odpowiednio zaawansowanych algorytmach) to wiemy, że pojawiają się w nich liczne parametry, które muszą mieć przypisane jakieś wartości. Na przykład w przypadku algorytmu genetycznego są to prawdopodobieństwa zachodzenia mutacji i krzyżowania, reguły selekcji i inne, w przypadku tabu – długość listy tabu, pamięć długoterminowana, parametry strategii wychodzenia z optimum lokalnego, w przypadku algorytmu mrówkowego – liczba mrówek, funkcje zarządzające macierzą feromonową (to jest ich parametry), itd.

Zanim więc zaczniecie Państwo kolejnymi wykresami udowadniać jak wspaniale sekwencjonują DNA wasze heurystyki, należy choć jednym czy dwoma skromnymi wykresami udowodnić zasadność tych parametrów, żeby były mniej „wyśnione" a bardziej uzasadnione. Na przykład, dlaczego populacja w genetycznym ma akurat wielkość 42 – przydałbym się wykres uzasadniający, który udowadnia (lub właśnie nie), że dla 43 i 41 to już tragedia i lepiej&szybciej wszystko liczyć na liczydle, a 42 to super wartość. Że już nie wspomnę o takich drobnostkach jak czas pracy algorytmu, prawdopodobieństwa mutacji, krzyżowania, decyzji mrówek, długość listy tabu, itd.

Nie popadajmy jednak w przesadę, przetestować wszystkiego się nie da, nawet matematycznie to problem wielowymiarowy i wielokryterialny (tj.: testujemy parametr x, ale przyjmujemy że a,b,c,y oraz z są takie a nie inne, ale czemu takie są? Bo je ustaliliśmy wcześniej przyjmując x jakiekolwiek… I tak w kółko.). Ale choć jeden skromny wykres testujący COŚ z samego algorytmu zdziała cuda na moje przyszłe potencjalne marudzenie, czy właśnie jego brak przy sprawdzaniu.

TESTY SEKWENCJONOWANIA

Oczywiście główną rzeczą, którą testujemy jest to jak dobrze algorytm radzi sobie ze składaniem DNA. Są trzy problemy do wyboru:

1. Klasyczny problem SBH z błędami negatywnymi oraz pozytywnymi.
2. Problem SBH z informacją o położeniu oraz wszystkimi rodzajami błędów.
3. Problem SBH z informacją o powtórzeniach oraz wszystkimi rodzajami błędów.

Parametry problemów do możliwego testowania:

- Długość poszukiwanego DNA – n : między 300 a 1000 (to ostatnie dla klasycznego SBH to już bardzo wysoka liczba, zwłaszcza z błędami w spektrum).
- Długość oligonukleotydów sond – k : pomiędzy 7 a 10. Oczywiście można użyć większych wartości, ale to już takie trochę oszukiwanie (nie ma takich macierzy, nie opłaca się) – wiadomo że im dłuższe sondy, tym łatwiej potem składać DNA. Im mniejsze k, tym więcej błędów negatywnych wynikających z powtórzeń będzie się pojawiać.
- Rodzaj błędów:
    - Negatywne wynikające z powtórzeń (standard, zakładamy, że przy dłuższych DNA trudno o spektrum, gdzie choć jeden czy dwa takie się nie pojawią)
    - Negatywne „normalne”
    - Pozytywne.
- Ilość / procent błędów danego typu. Proponuję przyjąć, że 2-3% z wartości n–k+1 to absolutne minimum. 10% w przypadku negatywnych to już naprawdę sporo, 15% to już ekstrema i oczekujmy wtedy raczej słabych wyników. Co do pozytywnych (samych), to zazwyczaj jest tak, że nawet 10-15% takich błędów (jeśli nie ma negatywnych) to sporo, ale algorytm może to jeszcze „ogarnąć”.
    - Jeśli bierzemy przypadek łączony, czyli np. negatywne (10%) i pozytywne (też 10%), to też można uznać, że jest to dużo błędów i jakoś rozwiązania może kuleć.
- [Problem 2] można testować precyzję informacji o powtórzeniach. Typowe informacje to „1 lub więcej” lub „1, 2 lub więcej powtórzeń”. Raczej nie występuje sytuacja, że np. powtórzeń jest 7 i algorytm dokładnie wie, że 7. To by oczywiście ułatwiało pracę, ale w praktyce dwa schematy które przedstawiłem były najczęściej osiągalne w eksperymencie.

Oczywiście nie trzeba testować absolutnie wszystkiego, ale im więcej testów dla różnych parametrów, tym lepiej dla sprawozdania i końcowej oceny.

REPREZENTACJA DANYCH

Dane w tabelkach są ok, o ile tabele są dobrze skonstruowane. To jest od razu wiadomo co jest w każdej komórce i jak to się łączy ze sobą jako całość. Ale jak już mamy excelowe tabelki, to oczywiście można je też przerabiać na wykresy. I zaczyna się zabawa: z tym zawsze był kłopot w sprawozdaniach z przedmiotów, gdzie wykresy trzeba robić. Opis poniżej na motywach książki: Janina Bąk, “Statystycznie rzecz biorąc. Czyli ile trzeba zjeść czekolady, żeby dostać Nobla” Wydawnictwo W.A.B., 2020

1. Wykres to nie walentynka – musi być **PODPISANY**. Dotyczy to tytułu jak i osi.
2. Wszystkie elementy wykresu (np. kolory, kształty punktów, itd.) są jak dzieci – muszą być odpowiednio **NAZWANE**. Odpowiednio to znaczy tak, by nikt nie miał żadnych wątpliwości, czego dotyczą i co oznaczają (elementy wykresu, nie dzieci).
3. Pamiętajmy, że 25% dużej pizzy to nie to samo co 25% MAŁEJ pizzy – zawsze podawajcie **LICZBĘ WSZYSTKICH OBSERWACJI N**, które obejmuje wykres.
4. Wizualizacja to nie aktywność fizyczna – jej celem nie jest utrudnianie ludziom życia. Uprośćcie swoim odbiorcom zrozumienie wykresu, na przykład dzięki dodaniu **ETYKIET Z WARTOŚCIAMI**.
5. Wykres to nie pisanka – **KOLORY** stosujemy ostrożnie.
6. Myślcie z czułością o ludziach, którzy stają w płomieniach na widok liczb – pokazujcie na wykresie dokładnie tyle, ile jest niezbędne, i **PODKREŚLAJCIE** to, co najważniejsze. Wykres to nie studniówka – nie potrzebuje brokatu, udziwnień i innych fiu-bździu.

TESTOWANIE – SZCZEGÓŁY

To co przeczytaliście do tej pory, jak dotąd wystarczało (lepiej lub gorzej). Powyższy tekst powstał w okolicach 2021 roku, korekty tegoroczne dotyczą pierwszej strony i samej góry drugiej strony. Gdzieś w tak zwanym międzyczasie, w okolicach 2023-2024, dla innego przedmiotu, ale z niemal tą samą tematyką zrobiłem inny opis testów. Wklejam go tutaj poniżej. To co teraz przeczytacie jest szczegółowym powtórzeniem tej samej idei testów, tylko bardziej dokładnie. Myślę, że bardzo się przyda (będę o tym mówić na drugim laboratorium).

Podsumowanie projektu do zrobienia (szczegóły były już dokładnie tłumaczone na laboratorium):

1) Praca w grupach dwuosobowych, należy zaimplementować algorytm(y) rozwiązujące problem sekwencjonowania przez hybrydyzację, w tym w skład projektu wchodzą:
    a. Generator instancji problemu generujący spektrum hybrydyzacji, potencjalnie z błędami pozytywnymi i negatywnymi (czy i ile – zależy od przeprowadzonego testu). Typowy rozmiar DNA n > 300nt oraz n < 700nt, rozmiar oligonukleotydów k >= 7 oraz k <= 10. Zależnie od przypadku.
    b. Generator rozwiązań początkowych, zwany przeze mnie w ramach laboratoriów „generatorem rozwiązań losowych", bo to niestety dobrze oddaje jego typową „jakość" pracy.
    c. Aby wyjść powyżej progu oceny 4.0 – 4.5, należy chociaż spróbować zaimplementować algorytm metaheurystyczny, który na podstawie rozwiązania/rozwiązań, które wyszły z algorytmu z punktu wcześniejszego stara się ulepszyć rekonstrukcję DNA z elementów spektrum z instancji.
    d. Sprawozdanie, na które składa się:
        i. Opis stworzonych algorytmów
        ii. Testy algorytmów, tj. ich parametrów, jeśli jakoweś w ogóle mają. W przypadku metaheurystyki jest tam ich zawsze sporo, należy wybrać 2-3 kluczowe i przetestować ich pływ na jakość otrzymywanego rozwiązania. Np. a przypadku ACO parametrem może być liczba mrówek, współczynnik parowania feromonów, itd. W przypadku algorytmu genetycznego: procentowe szanse na krzyżowanie i osobno mutacje, parametry związane z algorytmem (pomocniczym) selekcji rozwiązań, itd. Tabu Search: długość listy ruchów zakazanych, wielkość przeszukiwanego sąsiedztwa, itd.
        iii. Testy instancji problemu, np. wpływ % błędów hybrydyzacji danego typu na jakość rozwiązań wychodzących z algorytmu mającego już jakoś ustalone i niezmienne parametry wewnętrzne (np. z testów z poprzedniego podpunktu). Wpływ długości dna (dość trywialny i oczywisty w wynikach tekst), wpływ długości parametru k, itd.
2) Na potrzeby wykresów/tabel wyników najlepiej używać miary odległości między dwoma łańcuchami znaków, np. odległość Levenshteina. Ponieważ aby stworzyć dowolną instancję problemu musimy zacząć od wygenerowania DNA, to jak już na podstawie spektrum z błędami (lub bez) nasze algorytmy jakiś łańcuch DNA odtworzą, wtedy można odwołać się do tego oryginalnego DNA i obiektywnie ocenić stopień różnicy między obydwoma – na tej podstawie mamy punkty pomiarowe do wykresów w sprawozdaniu.
3) SBH klasyczne, znamy długość DNA, długość oligonukleotydów, czyli k, liczbę błędów negatywnych, pozytywnych, wierzchołek początkowy.
4) Testy w sprawozdaniu w ogólności można podzielić na dwa rodzaje. Pierwszy typ testów bada wpływ parametrów algorytmu na jakość rozwiązań, druga kategoria testów bada wpływ wartości zmiennych opisujących instancję problemu (np. liczba błędów) na trudność znalezienia dobrego rozwiązania.
    a. Na potrzeby testowania parametrów algorytmu musimy dysponować pewnym stałym, reprezentacyjnym zbiorem instancji testowych. Powinny być do siebie względnie podobne i w liczbie min. kilkudziesięciu. Sugeruję przyjąć dla nich jeden stały parametr k, np. k= 9 lub 10,

oraz stałą, albo podobną długość DNA, czyli n (np. między 500 a 700, ta druga wartość już raczej dla k=10). Różnić się zdecydowanie powinny liczbą i rodzajem błędów hybrydyzacji. Załóżmy dla przykładu poniżej, że mamy 60 takich instancji tworzących zbiór testowy.

i. Przykład: chcemy przetestować liczbę mrówek w metaheurystyce ACO, tj. wpływ tej liczby na jakość rozwiązania. Zakładamy, że dla danych i stałych parametrów ACO, liczba mrówek i jej wpływ na jakość rozwiązania będzie badana dla wartości 50, 100, 150, 200 i 250 mrówek.
ii. Zbieranie danych do testu / wykresu wygląda następująco. Najpierw każdą z 60 instancji przeliczamy przez metaheurystykę ACO dla liczby mrówek równej 50 (a wszystkie inne zmienne rządzące działaniem ACO są ustawione na **jakieś** wartości i ich nie zmieniamy już w tym teście. Przez przeliczenie rozumiem tu uzyskanie jakieś rekonstrukcji DNA. Potem dla każdej instancji porównujemy to co wyszło z algorytmu z DNA oryginalnym dla danej instancji (miara Levenshteina). Mając 60 instancji, po 60-ciu uruchomieniach algorytmów mamy 60 miar Levenshteina. Dodajemy je do siebie i dzielimy sumę przez 60. Wychodzi nam zwykła średnia arytmetyczna jakości rozwiązań dla 60 instancji zbioru testowego i liczby mrówek równej 50.
iii. Powtarzamy punkt powyższy osobno za każdym razem dla liczby mrówek 100, 150, 200 i 250. Dostajemy łącznie 5 punktów na wykres do sprawozdania (pięć bo każdy punkt pomiarowy to oczywiście średnia miara jakości miar Levenshteina dla liczby mrówek równej 50, 10, 150, 200 i 250 w tym przykładzie).
iv. Powyższy schemat można powtórzyć dla dowolnego parametru meteheurystyki, przetestować należy 2-3 parametrów, nawet jak ich jest więcej w algorytmie.
v. Jeżeli nie dysponujemy metaheurystyką to trzeba kombinować :) i próbować znaleźć jakiekolwiek istotne parametry w naszym algorytmie naiwnym, czy jak go tam nazwiemy. Poza takim „problemem", że jest to o wiele prostszy algorytm (niż metaheurystyka) i może mieć o wiele mniej parametrów, procedura z grubsza wygląda tak samo.

b. Testowanie parametrów instancji problemu. Czyli druga klasa testów w sprawozdaniu. Należy minimum 2-3 parametry testować, np. n / k (idą w parze zazwyczaj), liczbę błędów takich czy innych. Opiszę typowy test badania wpływu błędów negatywnych.

i. Generujemy sobie zbiorek instancji testowych. Tutaj wartości n oraz k powinny być **identyczne**, załóżmy, że chcemy 20 instancji. Tworzymy więc 20 łańcuchów DNA, robimy z nich spektrum. **Musimy zapamiętać dokładnie to 20 wygenerowanych DNA** – czemu, o tym dalej, przydadzą się jeszcze później.
ii. Załóżmy, że chcemy przetestować wpływ błędów negatywnych dla wartości 2, 4, 6, 8 i 10% spektrum. Załóżmy też, że dla naszych 20 instancji testowych zawsze mamy DNA gdzie n = 600, k = 10. W ramach szukania pierwszego punktu na wykres do sprawozdania, w każdej instancji usuwamy 2% spektrum, czyli tworzymy błędy negatywne.
iii. Każdą z tych 20 instancji z błędami przeliczamy algorytmem takim jaki mamy (w algorytmie, czy to losowym czy metaheurystycznym wszystkie parametry traktujemy jako stałe i niezmienne w testach dalszych). Otrzymujemy 20 zrekonstruowanych łańcuchów DNA, każdy porównujemy z oryginalnym DNA dla danej instancji – wychodzi miara Levenshteina dla każdego. Dodajemy 20 wyników do siebie, sumę dzielimy przez 20. Otrzymana liczba to punkt na wykresie, na osi Y – wartość miary Levenshteina, na osi X oznaczamy „2% bł. negatywnych" lub po prostu „2%" bo tytuł wykresu będzie nam ogłaszać co dokładnie testujemy w danej chwili i co jest na osi X.
iv. Teraz uwaga: dla 4% błędów negatywnych **musimy użyć tych samych 20 łańcuchów DNA które sobie zachowaliśmy na początku**. DNA, tj. jego struktura wewnętrzna, tj. kolejność i powtarzalność liter, ma wpływ na jakość wyników, a chcemy testować tylko wpływ błędów. Dlatego bierzemy te same DNA. Tym razem jednak w spektrum każdego z nich usuwamy 4% elementów. Po czym powtarzamy punkt trzeci tutaj i

ostatecznie otrzymujemy teraz punkt wykresu oznaczający średnią jakość rekonstrukcji DNA według miary Levenshteina dla 4% błędów negatywnych.

v. Powtarzamy dla takich %-ów błędów negatywnych, ile i jakie chcemy testować.