Tomasz Dzięcioł 132216 tomasz.dzieciol@student.put.poznan.pl

Ewa Fengler 132219 ewa.fengler@student.put.poznan.pl

Bioinformatyka, laboratoria środa godz. 11:45 – 13:15

Sprawozdanie 2

Zaimplementowanie i przetestowanie heurystyki

# **Wybór metody inicjalizowania początkowych szans dla wyboru kolejnych wyrazów.**

Zaprezentowany przez nas w pierwszej części sprawozdania algorytm został zaimplementowany zgodnie z przedstawioną ideą. Zanim został on ostatecznie przystosowany do zwracania jak najlepszych wyników, algorytm ten musiał przejść szereg testów parametryzujących. Pierwszym problemem wymagającym rozstrzygnięcia był szeroko opisywany wcześniej sposób początkowego przypisania wag prawdopodobieństwa dla wyboru kolejnych wyrazów podczas tworzenia sekwencji.

W obu strategiach wykorzystany miał być pomysł k-krotnie większej szansy wyboru dla kolejnego wyrazu przyłączanego do sekwencji, który to by miał wspólny początek z końcem sekwencji l jednym znaku dłuższym. Tak oto skonstruowana wartość prawdopodobieństwa wyboru w pierwszym przypadku miała zostać przypisana każdej parze wyrazów odpowiednio w zależności od długości wspólnych końcówek. Drugie podejście brało dodatkowo pod uwagę liczbę możliwości doboru kolejnego słowa o danej długości wspólnej końcówki. Wówczas dane prawdopodobieństwo nie było przypisywane, lecz rozdzielane równomiernie na wszystkie kombinacje tworzące zazębiające się końcówki o danej długości.

Kwestia ustalenia początkowych szans wyboru jest znacząca dla całego przetwarzania heurystyki. Ich wpływ najlepiej powinien być widoczny w jakościach otrzymywanych rozwiązań na przestrzeni kolejnych iteracji. W związku z tym zostanie przeprowadzone zebranie średnich wartości rozwiązań otrzymywanych w każdej kolejnej iteracji. Do eksperymentu posłuży duża instancja z pozytywnymi losowymi błędami – 59.500+200. Tak duży zbiór wyrazów powinien uwypuklić ewentualne różnice między opisanymi strategiami.

Algorytm w trakcie tego badania będzie wykonywać 100 iteracji, próbując w każdej z nich 100 razy odtworzyć sekwencję. Pozostałe parametry, które zostaną jeszcze przeanalizowane, są ustawione na domyślne wartości uznane za odpowiednie na etapie opisu stosowanej heurystyki. Ważne jest też, aby dla danej metody inicjalizacji wag nie wykonywać algorytmu mrówkowego tylko raz, który swoje działanie opiera na losowości. Aby móc zauważyć dane właściwości i móc wyciągnąć z nich poprawne wnioski, należy powtarzać całą heurystykę wielokrotnie. Dopiero po uśrednieniu otrzymywanych wyników można pozwolić sobie na analizę wykorzystanej wersji algorytmu. Taka metoda wielokrotnego powtarzania algorytmu będzie również zastosowana przy doborze odpowiednich wartości parametrów.

Na zaprezentowanych powyżej wykresach wyraźnie widać, która metoda inicjalizacji wag rozkładu prawdopodobieństwa działa lepiej. To dla drugiej metody przeciętnie otrzymywane wyniki w kolejnych iteracjach są zdecydowanie wyższe. Widać to zarówno w średniej liczbie wykorzystanych słów we wszystkich próbach, jak i w najlepszej próbie z użyciem największej ilości wyrazów. Już po pierwszych iteracjach druga strategia równomiernego rozdzielenia danej wagi prawdopodobieństwa daje najlepsze wyniki na poziomie przekraczającym , podczas gdy dla pierwszej metody poziom ten jest osiągany dopiero razy później. Otrzymane wyniki wyraźnie pokazują jak znacznie lepsza jest druga metoda przypisania początkowych szans prawdopodobieństwa wyboru kolejnego wyrazu.

# **Dobór odpowiednich wartości parametrów.**

W zaimplementowanej przez nas wersji algorytmu mrówkowego można wyróżnić kilka parametrów, którym można by nadać różne wartości. Od ich wartości będzie zależeć przebieg działania algorytmu – odpowiednie ich dobranie skutkować będzie otrzymywaniem jak najlepszych rozwiązań. Parametry te to:

* *iterations\_number* - liczba iteracji,
* *attempts\_number* - liczba prób w iteracji,
* *k\_set\_starting\_wages* - wartość k stosowana przy k-krotnie większej szansie wyboru kolejnego wyrazu,
* *part\_of\_the\_best\_sequences* - część zbioru najlepszych rozwiązań podlegający nagrodzeniu (wartość domyślna: ),
* *best\_price\_for\_sequence* - wartość zwielokrotnienia szansy wyboru połączenia par wyrazów będących częścią nagradzanych rozwiązań (wartość domyślna: ),
* *compensation\_coef* - współczynnik dominacji jednego połączenia nad pozostałymi, określający moment przeprowadzania wygładzania wag prawdopodobieństwa (wartość domyślna: ),
* *base\_of\_logarithm* - podstawa logarytmu używanego we wzorze wygłądzającym dominację najlepszego połączenia (wartość domyślna: ).

Kolejnym etapem jest więc ustalenie najlepszych wartości dla tych parametrów. Pierwsze dwa z nich są dość specyficzne.. Od liczby iteracji i ilości powtórzeń zależy bowiem długość działania algorytmu. Jednocześnie przy dłuższej pracy programu losowe szukanie rozwiązania doprowadzi kiedyś do coraz to lepszych rozwiązań. Mając to na uwadze można dojść do wniosku, że aby uzyskać najlepsze wykorzystanie wyrazów w sekwencji, powinno uruchomić program na nieskończenie długi czas. W związku z tą zależnością przyjęty został następujący plan działania algorytmu, który można uznać za kompromis pomiędzy pozwoleniem na jak najdłuższą pracę programu, a koniecznością podania ostatecznych wyników w rozsądnym czasie. Ustalona została bowiem stała liczba powtórzeń prób utworzenia sekwencji w danej iteracji, podczas gdy sam algorytm zostaje przerwany, gdy od czasu wykonywania pewnej stałej liczby iteracji nie zostaje otrzymane lepsze rozwiązanie. Wystarczająca wydaje się być liczba iteracji bez wzrostu funkcji celu, która to wyznaczać będzie koniec pracy algorytmu mrówkowego.

Dzięki wprowadzonym ograniczeniom pozostaje pięć parametrów wymagających ustalenia na podstawie analizy otrzymywanych rezultatów algorytmu. Sposób przeprowadzenia parametryzacji jest kluczowy w kwestii przeprowadzenia go w skończonym czasie. Chcąc bowiem sprawdzić jak dział heurystyka przy różnych wartościach dla każdego z parametrów, trzeba by zebrać wyniki dla wszystkich kombinacji przypisania parametrom wartości. W świetle wykonywania jednorazowego przebiegu algorytmu w czasie liczonym w sekundach, przeprowadzenie algorytmu dla wszystkich kombinacji dla jednej instancji testowej staje się praktycznie nie do wykonania.

Konieczne okazuje się więc dokonywanie testów wszystkich rozpatrywanych wartości danego parametru przy domyślnych wartościach pozostałych niezbadanych jeszcze współczynników. Wówczas wybierana jest wartość parametru, dla którego uzyskiwane są najlepsze wyniki, po czym można przejść do badania kolejnego parametru. Tym sposobem powinno zostać odkryte zestawienie parametrów składających się na bardzo dobre sprawowanie się algorytmu.

Aby móc wykonać omówiony proces parametryzacji, należy jeszcze tylko obrać instancje problemu, przy której dokonywana będzie analiza otrzymywanych przeciętnie wyników. Ze względu na wielokrotnie wykonywanie całego algorytmu, nie można pozwolić na obranie zbyt dużej instancji wykorzystującej co najmniej 500 wyrazów do odtworzenia wynikowej sekwencji. Odpowiednią instancją do tego zadania powinien być problem z rodziny 300+100 – problem z błędami pozytywnymi. Dodatkowym atutem takiej instancji oprócz nie dużego rozmiaru jest stosunkowo największa trudność w odnalezieniu oryginalnej sekwencji. Własność ta została zaobserwowana przy wstępnie wykonywanych obserwacjach uzyskiwanych wyników algorytmu mrówkowego w zależności od rodzaju instancji problemu. Trenowanie parametrów pod trudny przypadek powinno przynieść również lepsze rezultaty w późniejszym rozwiązywaniu potencjalnie łatwiejszych problemów.

1. Wartość k stosowana przy k-krotnie większej szansie wyboru kolejnego wyrazu.

Pierwszy wykres przedstawiający średnią jakość otrzymywanych rozwiązań pokazuje, że dla wartości parametrów większego niż heurystyka znajduje już praktycznie najlepsze rozwiązanie. Co więcej, najlepsze wyniki zostają znalezione już w około dwudziestej iteracji. Odstającym przypadkiem jest tu wartość współczynnika równa , dla której otrzymywane wyniki są bardzo słabe. Taka wartość parametru oznacza równe szanse połączenia danych par wyrazów, bez względu na długość zachodzących na siebie końcówek. Przypadek ten pokazuje więc, że nie uwzględnienie długości wspólnego sufiksu-prefiksu skazuje jakiekolwiek próby poszukiwania rozwiązania na całkowitą porażkę. Najlepszą wartością dla tego współczynnika będzie , dla której znajdywane wcześnie rozwiązania osiągały w trakcie testów zawsze maksymalną wartość.

1. Część zbioru najlepszych rozwiązań podlegający nagrodzeniu.

Dzięki wybraniu już odpowiedniego współczynnika w poprzednim badaniu, teraz podczas obserwowania działania algorytmu otrzymywane wyniki niemal zawsze były maksymalne. Warto tutaj zatem skupić swoją uwagę na innych pomiarach - czas obliczeń oraz iterację uzyskania najlepszego wyniku. Słusznie można tutaj zauważyć, że wczesne znalezienie najlepszego ułożenia wyrazów skutkuje szybkim zakończeniem pracy heurystyki. Związane jest to z poczynionym założeniem odnośnie liczby wykonywanych iteracji, które są dopiero kończone, gdy po wykonaniu kolejnych kolejek algorytmu nie następuje poprawa funkcji celu. Uzyskane wykresy wskazują na wybieranie najlepszych utworzonych sekwencji, które posłużą do nagrodzenia wykorzystywanych w nich połączeń par wyrazów.

1. Wartość zwielokrotnienia szansy wyboru połączenia par wyrazów będących częścią nagradzanych rozwiązań.

Podczas dobierania odpowiedniej wartości czynnika zwielokratniającego należy zwrócić uwagę na średnią wartość uzyskiwanych najlepszych rozwiązań. Praktycznie są one niemal jednakowe, bliskie maksymalnej wartości wykorzystania trzystu wyrazów. Wymagając jednak od algorytmu rekonstrukcji najlepszych sekwencji, trzeba postawić na niskie wartości badanego parametru. Najlepszym wówczas rozwiązaniem staje się wartość .

1. Współczynnik dominacji jednego połączenia nad pozostałymi, określający moment przeprowadzania wygładzania wag prawdopodobieństwa.

|  |  |  |  |
| --- | --- | --- | --- |
| compensation\_coef | czas obliczeń | największa liczba użytych wyrazów | iteracja uzyskania najlepszego wyniku |
| 0,5 | 53,73755 | 300 | 16,4 |
| 0,7 | 52,90251 | 300 | 15,8 |
| 0,8 | 52,51591 | 300 | 15,3 |
| 0,85 | 51,92842 | 300 | 14,8 |
| 0,9 | 48,75428 | 300 | 12,3 |
| 0,92 | 47,51673 | 299,5 | 11,2 |
| 0,95 | 53,36137 | 300 | 16,2 |
| 0,97 | 54,85823 | 300 | 17,4 |
| 0,98 | 49,74286 | 300 | 13 |
| 0,99 | 48,92409 | 300 | 12,3 |

Wyniki badań dla współczynnika dominacji zostały przedstawione w powyższej tabeli. Różnice pomiędzy poszczególnymi wartościami współczynnika są prawie identyczne. Wyróżnić można warianty, dla których praca algorytmu zakończyła się przed upływem sekund. To dla nich odnalezienie najlepszego rozwiązania miało miejsce średnio już w pierwszych kilkunastu iteracjach. Wartość parametru compensation\_coef= budzi zastrzeżenia ze względu na nie osiąganie maksymalnego użycia wyrazów. W tych okolicznościach najbardziej atrakcyjną staje się wartość .

1. Podstawa logarytmu używanego we wzorze wygłądzającym dominację najlepszego połączenia.

|  |  |  |  |
| --- | --- | --- | --- |
| base\_of\_logarithm | czas obliczeń | największa liczba użytych wyrazów | iteracja uzyskania najlepszego wyniku |
| 2 | 53,84044 | 300 | 16,5 |
| 3 | 47,52177 | 300 | 11,2 |
| 4 | 51,02738 | 300 | 14,2 |
| 5 | 47,49789 | 300 | 11,1 |
| 7 | 50,53964 | 300 | 13,9 |
| 10 | 51,42452 | 300 | 14,6 |
| 13 | 50,05657 | 299,6 | 13,4 |
| 16 | 50,33539 | 299,6 | 13,8 |
| 20 | 51,31465 | 300 | 14 |
| 25 | 47,77991 | 299,2 | 11,3 |

Zmiana ostatniego z parametrów nie wprowadza drastycznych zmian w procesie przetwarzania heurystyki. Jednakże, zbyt wysoki dobór podstawy logarytmu potrafi przeszkadzać w przeciętnym osiąganiu maksymalnego rozwiązania. W celu ustalenia ostatecznego zestawu wartości parametrów, wartość ostatniego współczynnika zostanie dopasowany do czwartego wariantu: .

# **Działanie algorytmu na obowiązkowych zbiorach instancji testowych.**

Po przeprowadzeniu szerokich testów parametrów, stworzony algorytm jest gotowy do zastosowania go na zbiorze instancji testowych. Wyniki dla poddania poszczególnych zestawów działaniu stworzonego algorytmu mrówkowego prezentują się w poniższej tabeli.