Algorytmy Dokładne i Aproksymacyjne Dla Wybranych Problemów Kombinatorycznych w Biologii Obliczeniowej

Autoreferat rozprawy doktorskiej

Tomasz Waleń

Instytut Informatyki
Uniwersytet Warszawski

2009–11–10

Wprowadzenie

W rozprawie zajmujemy się zagadnieniami algorytmicznymi inspirowanymi problemami kombinatorycznymi rozważanymi w biologii obliczeniowej. W biologii obliczeniowej kładzie się główny nacisk na takie wykorzystanie nowoczesnych metod obliczeniowych, aby umożliwiały nowe odkrycia biologiczne. Stąd w naturalny sposób szczególna uwaga jest poświęcona praktycznym zastosowaniom badań. W rozprawie skupiamy się głównie na teoretycznych aspektach rozważanych problemów, analizując ich bardzo uproszczone modele matematyczne. Można traktować takie dwa podejścia jako zupełnie przeciwstawne, jednak moim zdaniem uzupełniają się wzajemnie.

Większości problemów na które napotyka biologia obliczeniowa, nawet w przypadku zastosowania bardzo uproszczonego modelu matematycznego, pozostaje trudna do rozwiązania. Jednak bez dokładnego zbadania prostych wariantów tych problemów trudno marzyć o zdecydowanych postępach w rozwiązaniach ich bardziej złożonych wersji. Okazuje się, że badając w ten sposób problemy z jednej dziedziny, możemy przypadkowo odkryć zastosowania w zupełnie odległych dziedzinach. Przykładowo, techniki, które służą do badania podobieństwa pomiędzy sekwencjami DNA, znalazły zastosowanie w wykrywaniu podobieństw prac studenckich, artykułów prasowych, itp.

Główne wyniki rozprawy

W pierwszym rozdziale rozprawy zajmujemy się problemem wyznaczania minimalnego wspólnego podziału słów (problem MCSP). Problem ten polega na wyznaczeniu dla dwóch zadanych słów A i B, najmniej licznych podziałów A i B na bloki, tak, by każdy z podziałów zawierał ten sam multizbiór bloków. Przykładowo, dla słów $A = abaefgbcbc$ i $B = bcefgbcaba$ rozwiązaniem problemu MCSP są podziały $\mathcal{A} = (aba, efg, bc, bc)$, $\mathcal{B} = (bc, efg, bc, aba)$.

Problem ten jest blisko związany z problemem SBR (sorting by reversals), który był używany jako narzędzie pozwalające ocenić podobieństwo pomiędzy różnymi sekwencjami genów [1, 12]. Problem SBR polega na wyznaczeniu dla dwóch słów A, B, najkrótszej sekwencji operacji ρ, która przekształca A w B. Operacja $\rho(i, j)$ polega na odwróceniu kolejności symboli o indeksach $i, i+1, \ldots, j$ w słowie. Przykładowo, dla słów $A = dacb$, $B = abcd$, taka najkrótsza sekwencja to $\langle \rho(1, 2), \rho(2, 4) \rangle$, generująca następujący ciąg zmian: $dacb \rightarrow adcb \rightarrow abcd$.

Początkowo problem SBR był badany przy założeniu, że każda sekwencja wejściowa jest permutacją (czyli każdy gen jest unikalny). Niestety, nawet w przypadku permutacji problem SBR jest NP–trudny [2]. Co ciekawe, istnieją wielomianowe algorytmy dla problemu SBR, w których każdy symbol permutacji jest wyposażony w znak $+$ lub $-$, a odwrócenie zmienia znaki zamienianych symboli na przeciwne [12, 19].

Ograniczenie rozważań do permutacji okazało się być bardzo nienaturalne. Niestety usunięcie tego założenia stanowczo utrudnia efektywne rozwiązywanie problemu SBR. Nawet dla alfabetu składającego się tylko z dwóch znaków problem SBR jest NP-trudny [4]. Problem jest również trudny do aproksymacji. W ogólnym przypadku najlepszy znany algorytm daje współczynnik aproksymacji $\mathcal{O}(\log n \log^* n)$ [5, 6].

Jedną z metod przezwyciężenia tych trudności było wprowadzenie problemu k-SBR, w którym każdy symbol może pojawić się w słowach wejściowych co najwyżej k razy (czyli każdy gen może wystąpić w co najwyżej k wariantach). Kolejnym ważnym elementem było zbadanie związków pomiędzy problemami MCSP i SBR. W pracy [3] autorzy pokazali w jaki sposób można zastosować problem MCSP do rozwiązywania problemu SBR.

W pracy [13] P. Kolman podał dla problemu k-MCSP algorytm $\mathcal{O}(k^2)$–aproksymacyjny i działający w czasie $\mathcal{O}(kn)$. Po niewielkich modyfikacjach algorytm ten daje również rozwiązanie $O(k^2)$–aproksymacyjne dla problemu k–SBR. Dla pełności opisu omawiamy szczegółowo w rozprawie również to rozwiązanie.

Efektem wspólnej pracy z P. Kolmanem było podanie efektywnej, liniowej, implementacji algorytmu $\mathcal{O}(k^2)$–aproksymacyjnego dla problemu k–MCSP. Kluczowym elementem rozwiązania jest podanie struktury danych umożliwiającej efektywne znajdowanie wspólnych podsłów w dynamicznie zmieniającym się zbiorze słów. Efektywna implementacja algorytmu wymaga użycia drzewa sufiksowych [7, 10] oraz słowników opartych o strukturę umożliwiającą operowanie na zbiorach rozłącznych [11]. Wynik ten został zaprezentowany w pracy [15].

Głównym wynikiem tego rozdziału jest podanie algorytmu $\mathcal{O}(k)$–aproksymacyjnego dla problemu k-MCSP, co w konsekwencji prowadzi do uzyskania algorytmu $\mathcal{O}(k)$–aproksymacyjnego dla problemu k-SBR. Wynik ten został zaprezentowany podczas konferencji WAOA 2006 [14, 16].

W drugim rozdziale rozważamy problem MAX-NLS, który stanowi alternatywę dla typowych miar podobieństwa. Większość miar podobieństwa koncentruje się nad tekstowym opisem sekwencji (czy struktur). Okazuje się jednak, że niekiedy jest to bardzo mylące. Właściwsze wydaje się utożsamianie obiektów o podobnych kształtach. Motywacją w tym przypadku było badanie własności ncRNA (non-coding RNA), gdzie często różne sekwencje nukleotydów tworzą cząsteczki RNA o tym samym kształcie, a w konsekwencji pełnią podobną rolę. W problemie MAX-NLS wejściowe obiekty reprezentuje się jako grafy uliniowione (czyli grafy, których wierzchołki utożsamione są z kolejnymi liczbami całkowitymi), których krawędzie stanowią o kształcie obiektów. Rozwiązaniem problemu jest najliczniejszy wspólny podgraf stanowiący o wspólnych cechach wejściowych grafów. Jego rozmiar jest wyznacznikiem podobieństwa pomiędzy zadanymi grafami. W [8, 9] autorzy zaproponowali algorytm $\mathcal{O}(\log^2 m_{opt})$-aproksymacyjny dla problemu MAX-NLS (gdzie m_{opt} oznacza liczbę krawędzi w optymalnym rozwiązaniu).

Głównym wynikiem rozdziału jest algorytm $\mathcal{O}(\log m_{opt})$–aproksymacyjny rozwiązujący problem MAX-NLS. W nowym algorytmie poprawiony został również czas działania, z $\mathcal{O}(kn^5)$ do $\mathcal{O}(kn^2)$, gdzie k oznacza liczbę grafów, a n maksymalną liczbę wierzchołków w grafie. Rozwiązanie aproksymacyjne dla problemu MAX-NLS jest otrzymywane poprzez optymalne rozwiązanie problemu MAX-LLS (wersja oryginalnego problemu z bardziej restrykcyjnymi ograniczeniami dotyczącymi wspólnych podgrafów) dla wejściowych grafów. W algorytmie zastosowano standardowe techniki programowania dynamicznego. Wynik ten został zaprezentowany podczas konferencji CPM 2006 [17].

W trzecim rozdziale rozważamy problemy wyznaczania uwarunkowanych cykli Eulera. Problem ten był inspirowany jedną z metod używanych do odczytywania sekwencji DNA [18]. Rozważamy kilka problemów dotyczących cykli Eulera spełniających dodatkowe warunki. W szczególności problem obliczania, czy graf zawiera cykl Eulera zawierający (lub w innym wariancie problemu, nie zawierający) zadany zbiór ścieżek. Podajemy liniowy algorytm, który dla grafu prostego G oraz zbioru ścieżek P stwierdza, czy graf posiada cykl Eulera zawierający każdą ze ścieżek ze zbioru P, oraz dowodzimy, że problem jest NP-zupełny w przypadku multigrafów. Podajemy również liniowy algorytm, który dla prostego grafu G oraz zabronionej ścieżki π stwierdza, czy graf posiada cykl Eulera unikający ścieżki π.

Uwagi końcowe

Nowe wyniki przedstawione w rozprawie zostały zawarte w następujących artykułach:

- Petr Kolman, Tomasz Walen: *Reversal Distance for Strings with Duplicates: Linear Time Approximation using Hitting Set*. Electronic Journal of Combinatorics, 14(1), 2007.
- Petr Kolman, Tomasz Walen: *Reversal Distance for Strings with Duplicates: Linear Time Approximation Using Hitting Set*. Workshop on Approximation and Online Algorithms (WAOA), volume 4368 of LNCS, 2006.
- Petr Kolman, Tomasz Walen: *Approximating reversal distance for strings with bounded number of duplicates*. Discrete Applied Mathematics (DAM) 155(3), 2007.
- Marcin Kubica, Romeo Rizzi, *Stéphane Vialette, Tomasz Walen: Approximation of RNA Multiple Structural Alignment*. Combinatorial Pattern Matching (CPM), volume 4009 of LNCS, 2006.

Bibliografia

[1] V. Bafna, S. Muthukrishnan, and R. Ravi. Computing similarity between RNA strings. In *Proceedings of the 6th Annual Symposium on Combinatorial Pattern Matching (CPM)*, pages 1–16, 1995.

[2] A. Caprara. Sorting by reversals is difficult. In *Proceedings of the First International Conference on Computational Molecular Biology*, pages 75–83. ACM Press, 1997.

[3] X. Chen, J. Zheng, Z. Fu, P. Nan, Y. Zhong, S. Lonardi, and T. Jiang. Assignment of orthologous genes via genome rearrangement. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):302–315, 2005.

[4] D. A. Christie and R. W. Irving. Sorting strings by reversals and by transpositions. *SIAM Journal on Discrete Mathematics*, 14(2):193–206, 2001.

[5] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. In *Proceedings of the 13th Annual ACM-SIAM Symposium On Discrete Mathematics (SODA)*, pages 667–676, 2002.

[6] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Transactions on Algorithms*, 3(1), 2007.

[7] M. Crochemore and W. Rytter. *Text Algorithms*. Oxford University Press, 1994.

[8] E. Davydov and S. Batzoglou. A computational model for RNA multiple structural alignment. In *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM)*, volume 3109 of *Lecture Notes in Computer Science*, pages 254–269. Springer-Verlag, 2004.

[9] E. Davydov and S. Batzoglou. A computational model for rna multiple structural alignment. *Theoretical Computer Science*, 368(3):205–216, 2006.

[10] M. Farach. Optimal suffix tree construction with large alphabets. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 137–143, 1997.

[11] H. N. Gabow and R. E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 246–251, 1983.

[12] S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, 1999.

[13] P. Kolman. Approximating reversal distance for strings with bounded number of duplicates. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 3618 of *Lecture Notes in Computer Science*, pages 580–590, 2005.

[14] P. Kolman and T. Waleń. Reversal distance for strings with duplicates: Linear time approximation using hitting set. In T. Erlebach and C. Kaklamanis, editors, *Workshop on Approximation and Online Algorithms (WAOA)*, volume 4368 of *Lecture Notes in Computer Science*, pages 279–289. Springer, 2006.

[15] P. Kolman and T. Waleń. Approximating reversal distance for strings with bounded number of duplicates. *Discrete Applied Mathematics*, 155(3):327–336, 2007.

[16] P. Kolman and T. Waleń. Reversal distance for strings with duplicates: Linear time approximation using hitting set. *Electronic Journal of Combinatorics*, 14(1), 2007.

[17] M. Kubica, R. Rizzi, S. Vialette, and T. Waleń. Approximation of rna multiple structural alignment. In M. Lewenstein and G. Valiente, editors, *Combinatorial Pattern Matching (CPM)*, volume 4009 of *Lecture Notes in Computer Science*, pages 211–222. Springer, 2006.

[18] P. Pevzner, H. Tang, and M. Waterman. A new approach to fragment assembly in dna sequencing. *Proceedings of the 5th International Conference on Computational Molecular Biology (RECOMB)*, 2001.

[19] E. Tannier, A. Bergeron, and M.-F. Sagot. Advances on sorting by reversals. *Discrete Applied Mathematics*, 155(6-7):881–888, 2007.