様式Ｃ－１９

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成２５年　５月１３日現在

機関番号：11301
研究種目：基盤研究（C）
研究期間：2010～2012
課題番号：22500001
研究課題名（和文）配列アラインメントを高速に求めるための前処理に関する研究

研究課題名（英文）A study on preprocessing for time-efficiently aligning sequences

研究代表者
酒井　義文（SAKAI YOSHIFUMI）
東北大学・大学院農学研究科・准教授
研究者番号：10277361

研究成果の概要（和文）：２本の文字列のアラインメントを求める問題は、文字列間の類似度計算に多くの応用をもつ。文字列のアラインメントに関連するいくつかの問題に対して、文字列を前処理することで得られる情報を活用することで、従来の方法よりも高速に問題を解くことのできるアルゴリズムを提案した。この中には、一方の文字列のみしか前処理できないオンライ的な状況を想定した問題や､ある種の圧縮文字列のアラインメントを求める問題が含まれる。

研究成果の概要（英文）：Problems of finding a ‘good’ alignment of two strings have many applications in computing the similarity between strings. This study proposed algorithms for some of such problems. By exploiting information obtained from preprocessing, these algorithms can perform faster than the previous algorithms. The problems for which the algorithms were proposed include the problem in which only one of two strings is allowed to be preprocessed because of an one-line environment, and the problem in which strings to be aligned are compressed in a certain manner.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
2010 年度	500,000	150,000	650,000
2011 年度	500,000	150,000	650,000
2012 年度	700,000	210,000	910,000
年度			
年度			
総　計	1,700,000	510,000	2,210,000

研究分野：総合領域
科研費の分科・細目：情報学・情報学基礎
キーワード：アルゴリズム理論、文字列比較

１．研究開始当初の背景

２本の文字列のアラインメントは、対象となる各文字列に空白を表すギャップ文字を挿入して長さが同じくなるように揃えることで得られる。文字列に用いられる文字同士の間に類似性に関する得点を設定すると、アラインメントにおける各文字列の対応する位置に現れる文字対の得点の総和の最大値を、文字列間の類似度とみなすことができる。この意味で、２本の配列のアラインメントに基づく類似度を求める問題は、バイオインフォマティクスをはじめ様々な研究分野において多くの応用をもつ。

文字対の得点のもっとも単純な設定として、同じ文字同士の対の得点が１点、それ以外の文字同士の対の得点が０点である場合を考えると、アラインメントに基づく類似度を求める問題は、最長共通部分列を求める問題

とみなせる。ここで、文字列の部分列とは、その文字列のいくつかの文字を削除して得られる文字列のことであり、２本の文字列の最長共通部分列とは、それらの文字列がもつ共通かつ長さが最大の部分列のことである。これまでに、この問題を解くのに十分な漸近的計算時間の解析は、文字列の長さ m、n、最長共通部分列の長さ p、同一文字同士の文字対の個数 r、その中でも支配的とよばれる条件を満たすものの個数 d、文字列に用いられる文字の種類の個数 s になど様々な値に関する条件のもとでなされ、その計算時間で解を求めることのできる数多くのアルゴリズムが提案されている。

２．研究の目的

最長共通部分列問題や関連する問題を解くアルゴリズムの設計において、文字列に対する前処理の有無と、設計されたアルゴリズムが適用可能な環境との間には、ある種のトレードオフが存在する。なぜなら、文字列に前処理を行うためには、あらかじめ文字列全体を参照できる環境が必要であるからである。たとえば、２本の文字列双方に前処理を必要とするアルゴリズムは、一方の文字列が先頭から一文字ずつ逐次的に与えられるオンライン環境のもとでは適用できない。一方、オンライン環境を想定しないでよい場合には、文字列に対する適切な前処理によって、それをしない場合に比べてより高速に問題を解くことのできるアルゴリズムを設計できる可能性がある。

上記の観点から、本研究では、文字列に対する前処理について詳細に検討し、従来知られていたアルゴリズムよりも高速に、あるいは、より制限された環境のもとで問題を解くことのできるアルゴリズムを開発することを目的とした。

３．研究の方法

文字列のアラインメントにもとづく文字列間の類似度を求めることに関連する問題の定義を形式的に与えた後に、定義した核問題を解くためのアルゴリズムをそれぞれ提案し、それらた問題の解を正しく求めることの理論的な証明を与えた。さらに、提案した各アルゴリズムの実行における時間および領域に関する最悪な場合の計算量を理論的に見積もり、これまでに同じ問題を解くために提案されたアルゴリズムにおける計算量との比較を行った。

４．研究成果

(1) 定数種類の文字からなる文字列の最長共通部分列をオンライン環境で高速に求めることのできるアルゴリズムを提案した。

これまで、文字列の長さ m、n、最長共通部分列の長さ p、同一文字同士の文字対で支配的なものの個数 d に関して、Apostolico による O(d log m + n)時間 O(m + d)領域アルゴリズム、Chin と Poon による O(n + min(d, pm))時間 O(n + d)領域アルゴリズム、Rick による O(n + min(pm, (n - p))時間 O(n + d)領域アルゴリズムが漸近的に最速のアルゴリズムとして知られていた。ただし、文字列の長さは m ≤ n とする。この中で、Apostolico のアルゴリズムのみが、一方の文字列にのみにしか前処理をしないという意味で、オンライン環境のもとで実行可能なアルゴリズムである。残りの２つのアルゴリズムは２本の文字列双方に前処理をする必要があるため、オンライン環境で用いることは不可能であり、n が m に比較してどんなに大きな値であったとしても O(n + d)領域を要する。

上に述べた従来のアルゴリズムの漸近的な実行時間および使用領域に対し、本研究で提案したアルゴリズムは、Apostolico のアルゴリズムと同様に短い文字列のみを前処理する意味でオンライン環境で利用可能なアルゴリズムとして初めて、O(d + n)時間 O(m + d)領域での動作を実現した。漸近的な実行時間の詳細は O(min(d, p(m - q)) + n)である。ただし、q は、長さ m の文字列と長さ n の文字列の長さ m の接頭辞の最長共通部分列の長さである。提案したアルゴリズムの実行時間は、オフライン環境のみで動作する Chin と Poon のアルゴリズムや Rick のアルゴリズムが与えた漸近的な実行時間に関する上界の改良も意味する。

また、値 d に関する上界としてこれまで知られていた素直に得られる O(pm)、および、Rick による O(p(n - p))を改良し、O(p(m - q))を与えた。

(2) run 長符号化された文字列の最長共通部分列を高速に求めることのできるアルゴリズムを提案した。

文字列の run とは文字列中の同じ記号のみからなる極大な長さの区間であり、文字列は run によって分割される。文字列は各 run を構成する共通の文字と run の長さの対として表すことで run 長符号化される。文字列が長い run をもつ場合に、run 長符号化によって、その文字列は非常にコンパクトに表現される。

２本の文字列の長さを M、N とし、それぞれがもつ run の個数を m、n とする。文字列に現れる各 run をあたかも文字列に現れる一つの文字と扱っているかのような実行時間、すなわち O(MN)時間ではなく O(mn)時間で、２本の run 長符号化された文字列の最長共通部分列を求めることができるか否かは、興味深い未解決問題の一つである。run 長符号化文字列の最長共通部分文字列を求める最速

のアルゴリズムとして、これまでにApostolicoらによるO(mn log max(m, n))時間アルゴリズムと、LiuらおよびAnnらによるO(mn min(M/m, N/n))時間アルゴリズムが知られていた。M/m、N/nは各文字列におけるrunの平均長である。

Apostolicoらのアルゴリズムを、それが提案されたのちに提案されたHanによる整数ソートアルゴリズムを用いる文字列の前処理によって、O(mn loglog max(m, n))時間アルゴリズムへと改良できることに気付くのはそれほど難しいことではない。この前処理の手順をさらに注意深く設計することによって、本研究では、O(mn loglog min(m, n, M/m, N/n, X))時間で動作するアルゴリズムを提案した。ここで、Xは同じ共通文字同士のrunの対におけるrun長の差の平均値である。このアルゴリズムにより、Apostolocoのアルゴリズムで用いられている方針を用いることでもLiuらやAnnらのアルゴリズムが達成した漸近的計算時間を大幅に改良できることが示された。また、値Xに依存した計算時間の解析はこれまでなされていなかったが、すべてのrunが同じ長さをもつ場合にO(mn)時間アルゴリズムは容易に構築できるという事実から想起される、Xの値が小さければ小さいほど高速に解を求めることができるであろうという自然な直感が正しいことも、提案したアルゴリズムの時間計算量解析によって初めて示された。

(3) 特徴文字列問題を高速に解くことのできるアルゴリズムを提案した。

２本の文字列に対する特徴文字列問題は、一方の文字列の連続する区間で他方に類似度の高い連続する区間が一切存在しない最も短いものを求める問題である。ただし、類似度が高いか否かは、任意に与えられる閾値kに対して、相違度がk以下であるか否かによって判定される。この問題は、分子生物学においてDNA分子のハイブリダイゼーションを特定の位置で誤りなく行わせるためのプライマーの設計などに応用がある。

特徴文字列問題を高速に解くアルゴリズムとして、これまでに、編集距離を相違度として用いた場合のItoらのO(kmn)時間O(m + n)領域アルゴリズムが知られていた。ここで、編集距離のもとでの相違度は、同じ文字同士の文字対の得点を0点、それ以外の文字対の得点を-1 点とした場合の類似度の符号をマイナスからプラスに置き換えることで得られる値に等しい。

本研究で提案したアルゴリズムは、Ito らのアルゴリズムとはまったく異なる手法に基づくものであり、任意に与えられる相違度の閾値kに依存しない実行時間で動作するという特徴をもつ。具体的な漸近的実行時間と使用領域の大きさはそれぞれ、O(mn $\log^2$ n)とO(m log n + n)である。したがって、kとしてlog nの２次多項式を超える値が設定されると想定される場合において、提案したアルゴリズムを使用することで、従来のItoらのアルゴリズムを用いるよりも漸近的に高速に解を得ることができる。ただし、使用する領域については、Ito らのアルゴリズムと比較して対数倍程度多くを要する。

また、このアルゴリズムは、文字対の得点設定を文字の組合せに依存して任意に変更した場合であっても、同じ漸近的計算時間および使用領域で動作する。

(4) 海藻全単射を漸減的に求めることのできる高速なアルゴリズムを提案した。

海藻全単射は、すべての縦方向や横方向の辺のほかに、いくつかの任意の単位格子は左上の頂点から右下の頂点への斜め方向の辺をもつ m×n サイズの格子グラフにおける最も左あるいは最も上に位置する任意の頂点から、最も下あるいは最も右に位置する任意の頂点への最短経路の長さすべてをm + n個の添え字対としてコンパクトに表したものである。２本の文字列から定義されるある種の格子グラフにおける海藻全単射は、一方の文字列全体と他方の連続する部分区間、あるいは、一方の文字列の接頭辞と他方の接尾辞の任意の組合せにおける最長共通部分列の長さを線形時間で求めることに利用できる。このため、海藻全単射を求める問題は、文字列比較において多くの応用をもつ。

これまでに、格子グラフに単位格子を一つずつ漸増的に追加することで、それまでのグラフにおける全単射を定数時間で単位格子を一つだけ多くもつグラフの全単射に更新する素直な方法が知られていた。一方、逆に、グラフの端の位置に存在している単位格子を一つずつ漸減的に削除することで、それまでのグラフにおける全単射を定数時間で単位格子を一つだけ少なくもつグラフの全単射に更新する方法は知られていなかった。

本研究で提案したアルゴリズムは、O(mn log(m + n))時間でO(mn)領域を要する表を前処理によって作成することにより、単位格子の漸減的な削除にともなう海藻全単射の定数時間更新を可能とした。

海藻全単射の漸減構築の応用としては、任意に与えられた長さの最長共通部分列をもつ２本の文字列の連続する部分領域の対で、長さの合計が最小のものを見つける問題が挙げられる。この問題を解くことで得られる解は、２本の文字列の局所的に最も類似性の高い部分区間を与えているとみなせるため、文字列の間の共通構造の探索などに役立つ可能性がある。

(5) 文字列において各長さの連続する部分区間に現れる特定文字の最大個数に関する索引を高速に構築するアルゴリズムを提案した。この索引は、たとえば２本の文字列のギャップ文字の現れないアラインメントにおいて、局所的に最も類似度の高い連続する部分区間を探索する際に利用できる。

この索引を求めるアルゴリズムとしては、Moosa と Rahman による $O(n^2/\log n)$時間アルゴリズムが既に知られていた。ただし、n は文字列の長さである。このアルゴリズムは分割統治法に基づいて、Bremner らによる $O(m^2/\log m)$時間 m×m 行列最小値和畳込み計算を何度も実行することで、索引を構築する。

一方、本研究で提案したアルゴリズムは、構成が非常に単純であり、分割統治法に基づくことなしに、１回の最小値和畳込み計算により索引を構築する。また、O(n)時間の前処理により求めた文字列における特定文字の出現位置に関する情報を用いることによって、特定文字の個数が k であるとき、$O(n + \min(k^2/\log k, (n - k)^2/\log (n - k)))$時間で動作する。したがって、特定文字が文字列において非常に疎または密に表れる場合は、従来の Moosa と Rahman のアルゴリズムと比較して、より高速に索引を構築することが可能である。特に、k の値が $n^{1/2}$以下あるいは $n - n^{1/2}$ 以上の場合は、線形時間で索引を構築する。

特定文字にある種の条件を満たす任意の重みを与えることで、最大出現回数に関する索引を構築する問題を最大重みに関する索引を構築する問題へと自然に拡張できる。本研究において、重みがすべて任意の自然数である場合に $O(n + k^2/\log k)$時間で索引を構築するアルゴリズム、および、重みがすべて任意の整数である場合に $O(n + k \log\log k)$時間で索引を構築することのできるアルゴリズムも提案した。これらのアルゴリズムは、先に述べたギャップ文字列を含まないアラインメントにおける最も類似度の高い連続する部分区間を探索する応用において、より現実的な類似度設定のもとで解を得ることに利用できる可能性がある。

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計３件）

① Yoshifumi Sakai, Computing the longest common subsequences of two run-length encoded strings, L.-M. Chao, T.-s. Hsu, and D.-T. Lee (Eds.): ISAAC 2012, LNCS, 査読有, 7676, 2012, 197-206
DOI: 10.1007/978-3-642-35261-4_23
② Yoshifumi Sakai, A fast on-line algorithm for the longest common subsequence problem with constant alphabet, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 査読有, Vol. E95-A, No.1, 2012, 354-361
DOI: 10.1587/transfun.E95.A.354
③ Yoshifumi Sakai, A new algorithm for the characteristic string problem under loose similarity criteria, T. Asano et al. (Eds.): ISAAC 2011, LNCS, 査読有, 7074, 2011, 663-672
DOI: 10.1007/978-3-642-25591-5_68

〔学会発表〕（計７件）

① 酒井義文、部分文字列最大密度索引、2012 年度冬の LA シンポジウム、2013 年 1 月 28～30 日、京都市・京都大学
② 酒井義文、run 長符号化文字列の最強共通部分列計算、2012 年度夏の LA シンポジウム、2012 年 7 月 17～19 日、宮津市・宮津ロイヤルホテル
③ 酒井義文、海藻全単射の漸減構築、2011 年度冬の LA シンポジウム、2012 年 1 月 30 日～2 月 1 日、京都市・京都大学
④ 酒井義文、緩い類似税判定基準のもとでの特徴文字列問題アルゴリズム、2011 年度夏の LA シンポジウム、2011 年 7 月 19～21 日、湖西市・ザヴィラ浜名湖
⑤ 酒井義文、最長共通部分も配列計算における run 長の対数時間寄与、2010 年度冬の LA シンポジウム、2011 年 2 月 1～3 日、京都市・京都大学
⑥ Yoshifumi Sakai, Flexible computation of the longest common subsequence of run-length encoded strings, The 13th Japan-Korea Joint Workshop on Algorithms and Computation, 2010 年 7 月 23～24 日, 金沢市・金沢市文化ホール
⑦ 酒井義文、可能な最長部分パターンの文字列照合問題、2010 年度夏の LA シンポジウム、2010 年 7 月 20～22 日、氷見市・九殿浜温泉ひみのはな

６．研究組織
(1)研究代表者

酒井　義文（SAKAI YOSHIFUMI）
東北大学・大学院農学研究科・准教授
研究者番号：10277361