最小値関数を用いて適合度を算出するNRA検索アルゴリズムの改善

河村一史[†] 藤本典幸[††]
辻 裕樹[††] 萩原兼一[††]

Fagin らの NRA アルゴリズムはメタ検索システムにおいて，ユーザからの質問に適合する検索結果のうち，適合度の上位 k 件のみを出力するアルゴリズムである（k は定数）．メタ検索システムからサブ検索システムへのアクセス方法を Sorted Access に限定した場合，NRA の計算量は漸近的に最適である．しかし，さらに適合度算出関数を最小値関数に限定する場合には NRA の計算量を係数レベルで改善できる可能性が Fagin らにより指摘されていた．本論文では，適合度算出関数を最小値関数に限定して，NRA の計算量を削減したアルゴリズムを示す．また，単語分散型 WWW 並列全文検索システムの内部に NRA を適用して行った評価実験により，提案する改善アルゴリズムを用いれば平均検索応答時間を短縮できることを示す．

An Improvement of the NRA Algorithm Using the Min Function

Kazufumi Kawamura,[†] Noriyuki Fujimoto,[††] Hiroki Tsuji[††]
and Ken-ichi Hagihara[††]

Fagin et al. developed a meta search algorithm, named NRA, which retrieves for a given query the objects with the k highest scores from a collection of objects distributed over subsystems (where k is a given constant). Under the restriction that only the sorted access is available as the access method to the subsystems, NRA is asymptotically optimal. However, Fagin, et al. pointed out that the factor of the complexity could be improved under the additional restriction that only the min function is permitted to combine scores computed by subsystems. In this paper, we show that the complexity of NRA can be reduced in such a case. The result of our experiment on WWW parallel full-text search system with term partitioning shows that the response time is shortened by our improved algorithm.

1. はじめに

膨大な数の情報の中から，目的の情報を取得することを支援するために検索システムが開発されている．検索システムの例として，World Wide Web（以下 WWW と呼ぶ）上に公開された文書を検索対象とする WWW 全文検索システムなどがある．検索システムのユーザは，検索対象が持つ複数の属性に対して目的の情報を特徴付ける属性値を指定することで，システムへ検索要求を行う．このような検索システムへのユーザからの入力を質問と呼ぶ．ここで，質問に含まれる属性とその属性値の組を単純質問と呼び，その数を m とする．すなわち，質問は単純質問 m 個と，and や or などの演算子 $(m-1)$ 個を組み合わせたものである．入力された質問に対して検索システムは，検索対象が質問に対してどの程度適合しているかを実数値で表した適合度とともに，質問に適合する検索対象を出力する．このような検索システムの種類の 1 つとして，メタ検索システム[9] がある．メタ検索システムは複数の検索システムの検索結果を統合して 1 つの検索結果として表示する検索システムである．メタ検索システムには，複数の検索システムを用いることで検索対象の数を増やしたり，同種の検索対象に対して異なる属性に関する検索を行う検索システムを用いたりすることで，その検索結果を統合できるといった利点がある．

検索システムは通常，検索結果を適合度の降順に出力するが，多くの場合ユーザが目的とする情報は適合度の高い検索結果の部分集合の中に存在する[11),13)]．また，ユーザにとって検索結果の中での適合度の順位は重要であるが，適合度の値そのものは重要でないと考えられる．そこで，質問に適合する検索結果をすべ

† 大阪大学大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University
†† 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

て求めるのではなく，質問に対する適合度が上位 k 件の検索対象集合のみを求める（このとき各検索対象の適合度は求めない）ことで，検索処理を高速化できる（k は定数）．このようなアルゴリズムの 1 つとして Fagin らの NRA（No Random Accesses）アルゴリズム[3] がある．

NRA はメタ検索システムにおいて，適合度が上位 k 件の検索対象集合のみを求めるアルゴリズムである．NRA は各検索システムから受信した検索結果を統合する際，統合後の適合度の下限値と上限値を算出し，算出した値を用いて検索対象間の適合度の順位を定める．メタ検索システムから各検索システムへの検索要求として，Sorted Access のみが許される条件の下では，NRA は漸近的には最適なアルゴリズムである[3]．NRA では，上限値計算の計算量がボトルネックとなるが，Fagin らにより，適合度算出関数が最小値関数である場合は計算量の係数を改善できる可能性が指摘されていた[3]．

本論文では，適合度算出において一般的に用いられている関数の 1 つである最小値関数[3] を使用する場合，NRA のボトルネックとなる処理を削減したアルゴリズムを提案し，提案したアルゴリズムによって検索結果のうち適合度の上位 k 件を求めることができることを示す．また，複数の計算機によって構成される並列検索システムの内部において，計算機間に NRA を適用した検索システムを構築し，評価実験によって提案手法を実際のシステムに実装した場合の計算量削減の効果を示す．評価実験により，検索対象を 100 万件の文書とした，計算機 9 台構成の単語分散型 WWW 並列全文検索システムにおいて $k = 10$ としたとき，平均検索応答時間を 0.120 秒から 0.078 秒に短縮できることが分かった．

2. 検索システム

検索対象集合 D_{all} の中から，質問 q に適合する検索対象集合 $D(q)$ ($\subseteq D_{all}$) を求めるシステムを，検索システムと呼ぶ．

2.1 質　　問

検索システムがユーザからの入力として受付ける質問 q を，以下のように定義する．ここで，検索対象が持つ属性を A，A のある属性値を V とする．

(1) (A, V) は質問である．

(2) 質問 q_1，q_2 と演算子 and，or から構成される $(q_1 \text{ and } q_2)$ と $(q_1 \text{ or } q_2)$ はそれぞれ質問である．

質問 q に含まれる，(1) のように演算子 and，or を含まない質問を単純質問と呼び $q_i = (A_i, V_i)$ と表す．また，q に含まれる単純質問の数を m とする．すなわち，演算子を op と表すと，質問 q は一般的に $q = q_1\ op\ q_2\ op \cdots op\ q_m$ と表せる．

このように定義された質問 q に適合する検索対象集合 $D(q)$ を以下のように定義する．

- $D(A, V)$ は検索対象集合 D_{all} のうち，属性 A が属性値 V であるような検索対象の集合．
- $D(q_1 \text{ and } q_2) = D(q_1) \cap D(q_2)$
- $D(q_1 \text{ or } q_2) = D(q_1) \cup D(q_2)$

2.2 適　合　度

検索システムは質問に適合する検索対象集合 $D(q)$ の出力順を決定するため，質問 q に対する検索対象 $d \in D(q)$ の適合度 $G(q, d)$ を算出する．ここで，適合度 $G(q, d)$ を以下の性質を持つように定義するものとする：

$$\begin{cases} G(p, d) > 0 & \text{if } d \in D(q) \\ G(p, d) = 0 & \text{if } d \notin D(q) \end{cases}$$

質問 q が単純質問である場合は，属性 A と属性値 V から何らかの方法で $G(q, d)$ を求める．たとえば，WWW 文書検索システムの場合は，$A =$ 文書に含まれる単語，$V =$ 単語として，$G(q, d)$ は文書 d 内での単語の出現頻度から算出する値 TF と，単語に適合する検索対象の数から算出する値 IDF[4] を用いた TF·IDF 値などから求める．

演算子 and，or を用いた質問 q に対する適合度 $G(q, d)$ を以下のように定義する．ここで，t_1，t_2 を適合度算出関数と呼ぶ．

- $G(q_1 \text{ and } q_2, d) = t_1(G(q_1, d), G(q_2, d))$
- $G(q_1 \text{ or } q_2, d) = t_2(G(q_1, d), G(q_2, d))$

適合度算出関数の例としては，最小値関数 min，合計関数 sum，平均関数 avg，最大値関数 max や CombANZ，CombMNZ[8),12)] などがあげられる．

ここで，すべての i $(1 \leq i \leq m)$ について $x_i(d) \leq x'_i(d)$ ならば $t(x_1(d), \cdots, x_m(d)) \leq t(x'_1(d), \cdots, x'_m(d))$ が成り立つような適合度算出関数 t を単調増加関数と呼ぶ．例としてあげた min，sum，avg，max，CombMNZ は単調増加関数であり，CombANZ は単調増加関数ではない．

2.3 検 索 結 果

2.1 節，2.2 節の定義を用いると，質問 q に対して検索システムが出力する検索結果の集合 $R(q)$ は以下のように表せる．

$$R(q) = \{(d, G(q, d)) \mid d \in D(q)\}$$

すなわち，検索結果 $R(q)$ は質問 q に適合する検索対

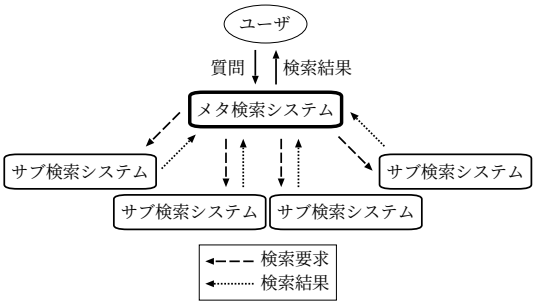

図 1 メタ検索システム
Fig. 1 A metasearch system.

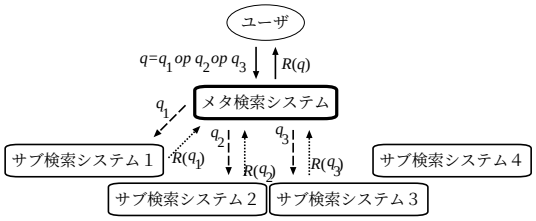

図 2 メタ検索システム（typeA）の検索処理
Fig. 2 Query processing of a type A metasearch system.

象 d ($\in D(q)$) とその適合度 $G(q,d)$ の組である．検索システムは検索結果 $R(q)$ を適合度 $G(q,d)$ の降順にユーザに出力する．

2.4 メタ検索システム

検索システムの 1 つにメタ検索システムと呼ばれるシステムがある．メタ検索システムの一般的な構成を**図 1** に示す．メタ検索システムはインデックスを保持せず，複数のサブ検索システムがそれぞれのインデックスを用いて求めた検索結果を加工，編集し，1 つの検索結果に統合してユーザに出力するシステムである．ここで，インデックス[1] とは，検索対象集合 D_{all} から，単純質問 (A,V) に適合する検索対象集合 $D(A,V)$ ($\subseteq D_{all}$) を取得するために必要な情報を，構造化して保存したものである．メタ検索システムの利点としては，複数のサブ検索システムが出力する検索結果を用いることで検索対象の数を増やすことができる点や，同種の検索対象に対して異なる属性に対する検索を行う複数のサブ検索システムを用いることで，異なる属性に対する適合度を統合できるといった点があげられる．

メタ検索システムが質問を処理するアルゴリズムを以下に示す．

(1) メタ検索システムは入力として，ユーザから質問 q を受付ける．
(2) メタ検索システムは q に基づいて，サブ検索システムに検索要求を送信する．
(3) サブ検索システムはメタ検索システムからの検索要求に対する検索結果を求め，メタ検索システムに送信する．
(4) メタ検索システムは検索要求を送信したすべてのサブ検索システムから検索結果を受信し，受信した検索結果を 1 つの検索結果に統合して，ユーザに出力する．

(2) では，メタ検索システムの種類によってサブ検索システムへの検索要求の方法が異なる．メタ検索システムは大きく分けて以下の 2 種類に分類できる．

（typeA）1 つの属性に対応するサブ検索システムは 1 つだけであるメタ検索システム
（typeB）1 つの属性に対応するサブ検索システムが 2 つ以上あるメタ検索システム

以降では，説明の便宜上 typeA のメタ検索システムについてのみ説明する．typeA のメタ検索システム（**図 2**）に質問 q が入力された場合，q に含まれる属性 A_1 に関する単純質問を検索要求としてサブ検索システム 1，属性 A_2 に関する単純質問を検索要求としてサブ検索システム 2 というように，q に含まれる単純質問単位でサブ検索システムに対する検索要求を送信する．図 2 は，質問 q に含まれる単純質問 q_1，q_2，q_3 がそれぞれ属性 A_1，A_2，A_3 に関する単純質問である場合の例である．A_i ($i \in \{1,2,3\}$) はサブ検索システム i が管理している．入力された質問に属性 A_4 に関する単純質問が含まれていなかったため，メタ検索システムは検索要求をサブ検索システム 4 には送信しない．したがって，サブ検索システム 4 は検索処理に参加しないこととなる．このように，メタ検索システムは各単純質問をそれぞれの属性に従って，あるサブ検索システム 1 つに送信して検索要求を行う．そのため，入力される質問によって検索処理に使用するサブ検索システムが決まる．

(4) では，サブ検索システムから検索結果を受信する．ここで，サブ検索システムから受信する検索結果をサブ検索結果と呼ぶ．サブ検索システムから受信したすべてのサブ検索結果を，適合度や順位を元に加工，編集して 1 つの検索結果に統合する．このとき，適合度算出関数 t_1，t_2 を用いて適合度 $G(q,d)$ を算出する．

3. NRA アルゴリズム

本章では，既存手法である NRA について説明する．適合度算出関数を適切に選ぶことにより，NRA は and 検索，or 検索のどちらにでも対応できる．以降では，サブ検索システムがメタ検索システムに送

信するサブ検索結果は，適合度と検索対象の組の集合 $R(q_i) = \{(d, G(q_i, d)) \mid d \in D(q_i)\}$ とする．サブ検索結果に適合度の順位情報のみが含まれており，適合度の値そのものの情報が付加されていない場合でも，順位情報からメタ検索システムの方で適合度を決定することで NRA を適用できる．

NRA と同じ問題を扱うアルゴリズムとして “Stream-Combine” アルゴリズム[5] があるが，以下の 2 点で NRA が優れている[2]．NRA は計算量において漸近的に最適なアルゴリズムであるが，“Stream-Combine” は最適ではない．また，“Stream-Combine” ではある検索対象について，質問に含まれるすべての単純質問に対する適合度を取得しなければ，統合後の検索結果のうち適合度の上位 k 件に含まれるかどうか判断できないが，NRA では適合度が未取得である単純質問がある場合でも，上位 k 件に含まれると判断できる状況が起こりうる．

3.1 NRA アルゴリズムの概要

NRA はメタ検索システムにおいて適合度算出関数 t として単調増加関数（2.2 節参照）を用いる場合，ユーザが入力した質問 q に適合する検索対象のうち適合度の上位 k 件（k は定数）である $D_{1..k}(q)$ を求めるアルゴリズムである．NRA は q に対して $D(q)$ に属する検索対象の適合度の下限値（3.3 節参照）と上限値（3.4 節参照）を算出することで，適合度の値域を求める．そして，検索対象 $d1$ の適合度 $G(q, d1)$ の下限値と検索対象 $d2$ の適合度 $G(q, d2)$ の上限値の大小関係から，$G(q, d1)$ と $G(q, d2)$ の大小関係を判定する．これにより各検索対象について $d \in D_{1..k}(q)$ かどうかを判定して $D_{1..k}(q)$ を求める．

以降では，$D(q)$ のうち，適合度 $G(q, d)$ $(d \in D(q))$ が上位 i 番目から上位 j 番目の検索対象集合を $D_{i..j}(q)(i \leq j)$ と書く．NRA の出力には各検索対象の適合度や，出力する検索対象集合内での適合度の順位の情報は含まれていないが，必要であれば，これらの情報は，定数 k を $1, 2, \cdots, k$ と順に増やしながら NRA を繰り返し実行すれば求められる．

3.2 Sorted Access

メタ検索システムによる各サブ検索システムに対する，適合度の降順の連続アクセスを Sorted Access（以下，SA）と呼ぶ．NRA は SA のみを用いて各サブ検索システムにアクセスする．

SA の例を**図 3** に示す．ここで，単純質問 q_i に対するサブ検索結果 $R(q_i)$ のうち，適合度 $G(q_i, d)(d \in D(q_i))$ が上位 j 番目から上位 n 番目までのサブ検索結果を $R_{j..n}(q_i)(j \leq n)$ と表す．また，$R(q_i)$ のうち適合度が上位 j 番目のサブ検索結果を $R_j(q_i)$ と表す．2.4 節の typeA のメタ検索システムを考えているので，q_i を処理するサブ検索システムはただ 1 つに定まり，$R(q_i)$ 中での適合度による検索結果の順位付けが可能となっている．

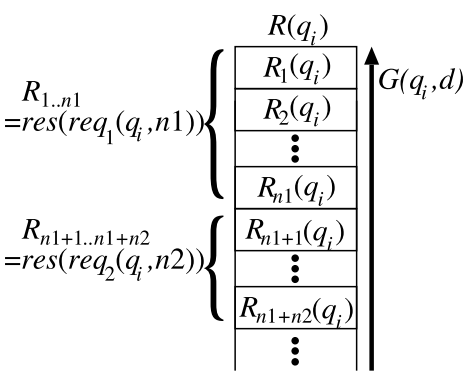

図 3 Sorted Access
Fig. 3 Sorted Access.

メタ検索システムからサブ検索システムへ単純質問 q_i と件数 $n1$ を検索要求 $req_1(q_i, n1)$ として送信する．検索要求を受信したサブ検索システムは，検索要求に対する検索応答 $res(req_1(q_i, n1))$ として，サブ検索結果 $R_{1..n1}(q_i)$ をメタ検索システムへ送信する．ここで，SA でそれまでに取得したある単純質問のサブ検索結果の総件数を深さ b と呼ぶ．この段階では，$b = n1$ である．

続いて，検索要求 $req_2(q_i, n2)$ を送信した場合，$b = n1$ なので $R_{n1+1}(q_i)$ から数えて $n2$ 件のサブ検索結果を検索応答 $res(req_2(q_i, n2))$ としてメタ検索システムへ送信する．すなわち，$res(req_2(q_i, n2)) = R_{n1+1..n1+n2}(q_i)$ であり，$b = n1 + n2$ となる．

このように，深さが b であることは，質問 q に含まれるすべての単純質問 $q_i(1 \leq i \leq m)$ のサブ検索結果 $R_{1..b}(q_i)$ をメタ検索システムが取得済みであることを表す．

3.3 適合度の下限値

質問 q に含まれる単純質問 $q_i(1 \leq i \leq m)$ に対する検索対象 d の適合度 $G(q_i, d)$ を $x_i(d)$ と書く．また，すべての q_i に対する d の適合度のうち，深さが b の段階で取得済みの適合度の集合を $S^{(b)}(d) = \{x_{i_1}(d), \cdots, x_{i_l}(d) | 1 \leq i_1, \cdots, i_l \leq m, l \leq m\}$ とする．

適合度算出関数 t が単調増加関数（2.2 節参照）であるならば，$G(q, d)$ の下限値 $W^{(b)}(q, d)$ は，未取得の適合度 $x_i(d)(1 \leq i \leq m, i \notin \{i_1, \cdots, i_l\})$ を 0 で補完することで t によって算出される．す

なわち，$i_n = n(1 \leq n \leq l)$ ならば $W^{(b)}(q,d) = t(x_1(d), x_2(d), \cdots, x_l(d), 0, \cdots, 0)$ となる．

3.4 適合度の上限値

深さが b のとき SA によって取得した単純質問 q_i のサブ検索結果 $R_{1..b}(q_i)$ のうち，上位 b 番目の適合度を $\underline{x}_i^{(b)}$ とする．すなわち，適合度が上位 b 番目の検索対象を d_b とすると，$\underline{x}_i^{(b)} = G(q_i, d_b)(d_b \in D(q_i))$ である．

適合度算出関数 t が単調増加関数（2.2 節参照）であるならば，質問 q に対する検索対象 d の適合度 $G(q,d)$ の上限値 $B^{(b)}(q,d)$ は，未取得の適合度 $x_i(d)(1 \leq i \leq m, i \notin \{i_1, \cdots, i_l\})$ を $\underline{x}_i^{(b)}$ で補完することで適合度算出関数 t によって算出される．すなわち，$i_n = n(1 \leq n \leq l)$ ならば $B^{(b)}(q,d) = t(x_1(d), x_2(d), \cdots, x_l(d), \underline{x}_{l+1}^{(b)}, \cdots, \underline{x}_m^{(b)})$ となる．

3.5 NRA アルゴリズムの詳細

本節では，NRA の処理内容を説明する．ここで，q に含まれるすべての単純質問 $q_i(1 \leq i \leq m)$ のサブ検索結果 $R(q_i)$ のうち，深さ b の段階で SA によって取得済みのサブ検索結果 $R_{1..b}(q_i)$ に含まれるすべての検索対象の集合を $D^{(b)}(q)$ とする．また，1 回の SA によって取得するある単純質問のサブ検索結果の件数を step とし，アルゴリズムの初期状態では深さ $b = 0$ とする．

(1) $j = b + 1$，深さ $b = b + \text{step}$ として，質問 q に含まれるすべての単純質問 q_i のサブ検索結果 $R(q_i)(1 \leq i \leq m)$ に対する SA により，$R_{j..b}(q_i)$ を取得する．SA によるサブ検索結果へのアクセスのたびに，深さ b における以下の情報を保持するために必要な計算を行う．
  (a) すべての $d(\in D^{(b)}(q))$ について，すべての単純質問に対する適合度のうち，取得済みの適合度の集合 $S^{(b)}(d)$
  (b) すべての $d(\in D^{(b)}(q))$ について，q に対する適合度の下限値 $W^{(b)}(q,d)$
  (c) すべての $d(\in D^{(b)}(q))$ について，q に対する適合度の上限値 $B^{(b)}(q,d)$
  (d) $T_k^{(b)} = \{D^{(b)}(q)$ のうち，$W^{(b)}(q,d)$ の上位 k 件の集合$\}$，$M_k^{(b)} = \{T_k^{(b)}$ のうち k 番目に大きい $W^{(b)}(q,d)$ の値$\}$

(2) 検索対象集合 $D' = D^{(b)}(q) \setminus T_k^{(b)}$ として，終了条件 $C_1 = \{$(SA によって取得した一意な検索対象の数が k 個以上) かつ (すべての $d' \in D'$ について $B^{(b)}(d') \leq M_k^{(b)})\}$ とする．また，終了条件 $C_2 = \{q$ に含まれるすべての単純質問 q_i のサブ検索結果 $R(q_i)(1 \leq i \leq m)$ を SA によってすべて取得 $\}$ とする．C_1，C_2 がともに不成立ならば，(1) へ戻る．C_1 または C_2 が成立すれば，$T_k^{(b)}$ を出力として終了する．

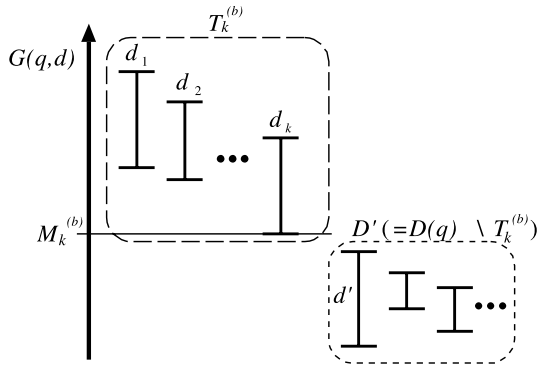

図 4 NRA による適合度上位 k 件の決定
Fig. 4 Decision of the documents with top k scores by NRA.

NRA の概念を**図 4** に示す．図中の各線分はそれぞれ質問 q に対する検索対象 $d(\in D(q))$ の適合度の値域を，線分の上端が上限値，下端が下限値をそれぞれ表す．d_k と d' の 2 つの検索対象のように，質問に対する適合度の値域が重ならない場合，適合度の大小関係が判定できる．したがって，集合 $T_k^{(b)}$ の中で最小の下限値 $M_k^{(b)}$ が，検索対象集合 $D' = D(q) \setminus T_k^{(b)}$ に含まれるすべての検索対象の適合度の上限値を上回ると，$D(q)$ の中で q に対する適合度の上位 k 件が $T_k^{(b)}$ と判定できる．

3.6 NRA アルゴリズムの問題点

3.5 節で述べた NRA の各処理の計算量は，アルゴリズム終了時の深さを b，質問 q に含まれる単純質問の数を m とすると以下のようになる．ここで，ある単純質問についての SA の計算量は取得するサブ検索結果の数に比例するものとする．

(1) アルゴリズムが終了するまでに，SA によって取得するサブ検索結果は $R_{1..b}(q_i)$ となる．SA は q に含まれる単純質問 q_i すべてについて行うため，その計算量は $O(bm)$ である．
  (a) SA によって取得したサブ検索結果 $R_{1..b}(q_i)$ の各検索結果について適合度を格納するので，その計算量は $O(bm)$ である．
  (b) 深さ l における下限値 $W^{(l)}(q,d)$ の算出方法は 3.3 節で述べたとおりである．深さ l において，各単純質問 q_i について SA によって step 個のサブ検索結果を取得したとき，下限値の算出は step 個

のサブ検索結果 $R_{l..l+\mathrm{step}}(q_i)$ に含まれる検索対象についてのみ更新する必要がある，すなわち，下限値の更新の必要がある検索対象の数は最大で $\mathrm{step} \times m$ 個となる．アルゴリズムが終了するまでには，各単純質問のサブ検索結果をそれぞれ b 個取得しているので，下限値算出に要する計算量は $O(bm)$ である．

(c) 深さ l における上限値 $B^{(l)}(q,d)$ の算出方法は 3.4 節で述べたとおりである．深さが l のとき，SA によって取得した各単純質問のサブ検索結果 $R_{1..l}(q_i)$ のうち，上位 l 番目の適合度の集合 $\{\underline{x}_1^{(l)},\cdots,\underline{x}_m^{(l)}\}$ を用いて，SA によって取得したすべてのサブ検索結果に含まれる検索対象集合 $D^{(l)}(q)$ について上限値を算出する．深さが l のときに上限値の算出に用いる $\{\underline{x}_1^{(l)},\cdots,\underline{x}_m^{(l)}\}$ は，SA によって深さが変更されるたびに，すなわちサブ検索結果を取得するたびに更新されるため，SA によってサブ検索結果を取得するたびにすべての検索対象 $d \in D^{(l)}(q)$ について上限値を更新する必要がある．つまり深さ l のとき，上限値の更新が必要な $D^{(l)}(q)$ の要素の数は最大で $l \times m$ 個となる．アルゴリズムが終了するまでに，上限値算出に要する計算量は文献 3) によると $O(b^2m)$ である．

(d) 深さ l における $T_k^{(l)}$ の更新は，1 回の SA で取得したサブ検索結果 $R_{l..l+\mathrm{step}}(q_i)$ に含まれるすべての検索対象について，更新する必要があるかどうか判断する．したがって，アルゴリズムが終了するまでに $T_k^{(b)}$ の更新に要する計算量は $O(bm)$ であり $M_k^{(l)}$ についても同様である．

(2) 終了条件 C_1，C_2 より，SA によって取得したすべてのサブ検索結果 $R_{1..l}(q_i)$ に含まれる検索対象集合 $D^{(l)}(q)$ のうち，すべての $d' \in D'(= D^{(l)}(q) \setminus T_k^{(l)})$ について $B^{(l)}(d') \leq M_k^{(l)}$ の比較を行う必要がある．したがって，アルゴリズムが終了するまでに終了条件 C_1，C_2 の評価に要する計算量は $O(b^2m)$ である．

NRA の各処理の計算量は上記のようになる．したがって，NRA 全体の計算量は $O(b^2m)$ となり，(1c) の上限値算出処理の計算量がボトルネックとなる．

4. 改善手法

本章では，適合度算出関数が最小値関数 min であり，かつ，質問が and 検索である場合に NRA の問題点を解決する改善手法を提案する．

4.1 改善方針

改善の方針としては，特定の適合度算出関数 t を考えた場合に t の性質から，求める検索結果の上位 k 件を変えないように NRA の終了条件 C_1 を変更することを考える．

t として最小値関数 min（2.2 節参照）を考える．

$t = \min$ のとき，NRA の終了条件 C_1 を変更することにより，3.6 節で述べた (1c) の上限値算出計算を省略し，NRA の計算量削減を目指す．

4.2 NRA アルゴリズムの改善

本節では，最小値関数 min を用いる場合の NRA の改善について述べる．

適合度算出関数 t として min を用いる場合，3.5 節で示した NRA の処理内容 (1d) の後に以下の（1e）を追加し，(2) を以下の (2') に変更する．

（1e） Max ={ 質問 q に含まれるすべての単純質問 $q_i(1 \leq i \leq m)$ について，サブ検索結果 $R_{1..b}(q_i)$ のうち，上位 b 番目の適合度 $\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)}$ の最大値 }

（2′） 終了条件 C_1' ={(SA によって取得した一意な検索対象の数が k 個以上) かつ $(Max \leq M_k^{(b)})$} とする．また，終了条件 C_2 ={q に含まれるすべての単純質問 q_i のサブ検索結果 $R(q_i)(1 \leq i \leq m)$ を SA によってすべて取得 } とする．C_1'，C_2 がともに不成立ならば，(1) へ戻る．C_1' または C_2 が成立すれば，$T_k^{(b)}$ を出力として終了する．

改善後の NRA の概念図を**図 5** に示す．

改善前，質問 q に対する検索対象 d の適合度の上限値 $B^{(b)}(q,d)$ は終了条件 C_1 でのみ参照した．しかし，改善後の終了条件 C_1' では，$B^{(b)}(q,d)$ を参照しないため，3.5 節で述べた NRA の処理内容 (1c)，すなわち $B^{(b)}(q,d)$ を算出する処理を省略することができる．

4.3 改善後の計算量

本節では，4.2 節で示した改善前後の NRA の計算量について述べる．

(2') は以下の 3 つの計算からなる：

(p1) SA によって取得した一意な検索対象の数の計算

(p2) $Max \leq M_k^{(b)}$ の判定

(p3) 終了条件 C2 の計算

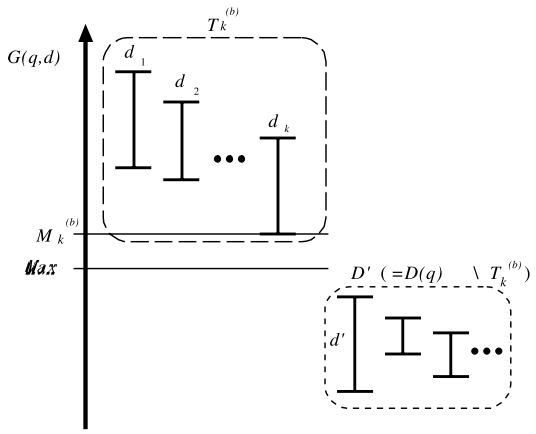

図 5 改善後の NRA による適合度上位 k 件の決定
Fig. 5 Decision of the documents with top k scores by improved NRA.

表 1 改善前後の NRA における各処理の計算量
Table 1 The time complexity of the existing method and the proposed method.

処理	改善前	改善後
(1c)	$O(b^2m)$	–
(1e)	–	$O(bm)$
(2)	$O(b^2m)$	–
(2')	–	$O(b^2m)$

深さが a のときの (1e)，(p1)，(p2)，(p3) の計算量を考える．(1e) は明らかに $O(m)$ 時間で計算できる．(p1) は $O(am)$ 時間で計算できる．$M_k^{(b)}$ と Max はそれぞれ (1d) と (1e) で計算済みなので，(p2) は $O(1)$ 時間で計算できる．(p3) は明らかに $O(m)$ 時間で計算できる．ゆえにアルゴリズム終了までの (1e) と (2') の全体の計算量は $\sum_{a=1}^{b} O(am) = O(b^2m)$ である．

NRA の処理のうち，改善前後で変化があった処理の計算量を**表 1** に示す．ここで，アルゴリズム終了時の深さを b とする．表から，改善後の NRA 全体の計算量は $O(b^2m)$ となり，改善前の計算量 $O(b^2m)$ とオーダレベルでは同じであることが分かる．ただし，改善前のボトルネックである処理 (1c) に比べて，追加されている処理は簡単であるため，改善によってアルゴリズムの計算量が係数レベルでは削減されている．これは 5.3.2 項の実験結果からも分かる．

4.4 改善後の終了条件の正しさの証明

改善前の NRA によって，質問に適合する検索対象のうち適合度の上位 k 件を正しく求められることは，文献 3) で証明されている．

本節では，NRA において適合度算出関数 t として最小値関数 min を用いた場合，4.2 節で示した終了条件 C_1' によって，質問 q に適合する検索対象の集合 $D(q)$ の適合度の上位 k 件 $D_{1..k}(q)$ を正しく求められることを証明する．

ここで，終了条件 C_2 が成立してアルゴリズムが終了する場合，$D_{1..k}(q)$ が正しく求められることは自明である．

［定理］ 質問 q に適合する検索対象集合 $D(q)$ の適合度の上位 k 件を $D_{1..k}(q)$ とすると，適合度算出関数 t として最小値関数 min を用いた NRA において，深さ b で終了条件 C_1' ={(SA によって取得した一意な検索対象の数が k 個以上) かつ $(Max \leq M_k^{(b)})$} が成立したとき，$T_k^{(b)} = D_{1..k}(q)$ である．

［証明］ 終了条件 C_1' より，SA によって一意な検索対象を k 個以上取得済みであるので，定義より $T_k^{(b)}$ に含まれる要素数は必ず k 個である．$T_k^{(b)}$ ={$d_1^T, d_2^T, \cdots, d_k^T$}，$T_k^{(b)}$ に含まれない任意の検索対象を d' として，すべての $i(1 \leq i \leq k)$ について $G(q,d') \leq G(q,d_i^T)$ が成立することを証明する．

定義より，

$$G(q,d') \leq B^{(b)}(q,d') \tag{1}$$

$$M_k^{(b)} \leq W^{(b)}(q,d_i^T) \leq G(q,d_i^T) \tag{2}$$

が成り立つ．ここで，適合度算出関数 t として最小値関数 min を用いているので，$1 \leq l \leq m$ とすると，

$$B^{(b)}(q,d') = \min(x_1(d'), \cdots, x_l(d'), \underline{x}_{l+1}^{(b)}, \cdots, \underline{x}_m^{(b)})$$

である．

(i) $l = m$ のとき
質問 q に含まれるすべての単純質問に対する検索対象 d' の適合度を取得済みなので，$B^{(b)}(q,d') = W^{(b)}(q,d')(= G(q,d'))$ が成り立つ．$M_k^{(b)}$ の定義 $W^{(b)}(q,d') \leq M_k^{(b)}$ より，

$$B^{(b)}(q,d') \leq M_k^{(b)} \tag{3}$$

が成り立つ．式 (1)，(2)，(3) より，
$G(q,d') \leq G(q,d_i^T)$
が成立する．

(ii) $1 \leq l < m$ のとき
仮定より，

$$\max(\underline{x}_1^{(b)}, \cdots, \underline{x}_m^{(b)}) \leq M_k^{(b)} \tag{4}$$

が成立する．ここで，

$$B^{(b)}(q,d') \leq \max(\underline{x}_1^{(b)}, \cdots, \underline{x}_m^{(b)})$$

が成立することを示す．

(a) $B^{(b)}(q,d') = x_j(d')$ $(1 \le j \le l)$ のとき

$$\begin{aligned} &B^{(b)}(q,d') \\ &= \min(x_1(d'),\cdots,x_l(d'),\underline{x}_{l+1}^{(b)},\cdots,\underline{x}_m^{(b)}) \\ &= x_j(d') \ (1 \le j \le l) \end{aligned}$$

であるので，すべての n $(l+1 \le n \le m)$ について $x_j(d') \le \underline{x}_n^{(b)}$ が成立する．したがって，

$$B^{(b)}(q,d') \le \max(\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)})$$

が成立する．

(b) $B^{(b)}(q,d') = \underline{x}_j^{(b)}$ $(l+1 \le j \le m)$ のとき

$$B^{(b)}(q,d') \le \max(\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)})$$

が成立するのは明らかである．

(a)，(b) より，

$$B^{(b)}(q,d') \le \max(\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)}) \quad (5)$$

が成立する．式 (1)，(2)，(4)，(5) より，

$G(q,d') \le G(q,d_i^T)$

が成立する．

(i)，(ii) より，$G(q,d') \le G(q,d_i^T)$ が成り立つ．∎

5. 評価実験

本章では，4 章で述べた改善による効果を評価するために行った実験とその結果について述べる．実験では，適合度算出関数として最小値関数 min，質問として and 検索を用いて実験を行った．

5.1 実験に用いたシステム

本節では，評価実験の際に NRA を実装した検索システムについて述べる．

5.1.1 単語分散型 WWW 並列全文検索システム

実験に用いた検索システムは，単語分散型 WWW 並列全文検索システムである．このシステムは**図 6** に示すとおり，以下の 2 つの役割を果たす複数の計算機によって構成される．

- 検索ゲートウェイ
  検索システムとユーザ間の入出力と，検索サーバへの検索要求，検索サーバが出力する検索結果の受信を担当
- 検索サーバ
  インデックスを保持し，受信した検索要求に応じた検索を担当

並列化されたシステムでは検索に用いるインデックスを分割し，システムを構成する計算機に配置

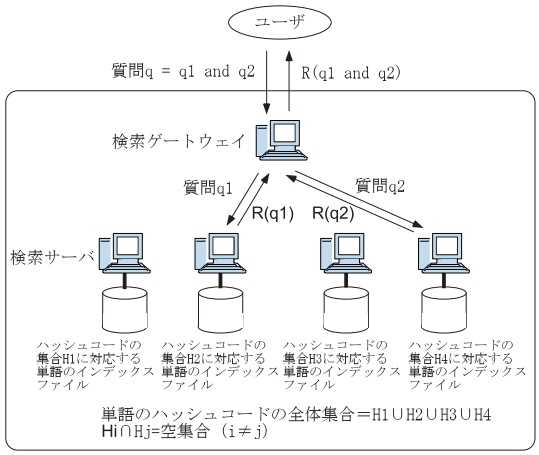

図 6 単語分散型 WWW 並列全文検索システム
Fig. 6 A WWW parallel full-text search system with term partitioning method.

する必要がある．インデックスの分割手法の 1 つとして，単語分散手法[6),7)] がある．これは検索対象となるすべての文書集合 D_{all} 中に記述されているすべての単語集合 $W(D_{all})$ に関するインデックスを $I(W(D_{all}))$ とすると，$W(D_{all})$ を n 個の部分集合 $W_1(D_{all}),\cdots,W_n(D_{all})$ に分割し，分割された単語集合に対応するようにインデックスを $I_1(W_1(D_{all})),\cdots,I_n(W_n(D_{all}))$ の n 個に分割するという手法である．単語分散型 WWW 並列全文検索システムは，単語分散手法を適用した WWW 並列全文検索システムである．

5.1.2 NRA の適用

本項では，5.1.1 項で述べたシステムの内部への NRA の適用について述べる．

単語分散手法を用いてインデックスを分割した並列検索システムでは，検索ゲートウェイが図 1 のメタ検索システムに相当し，検索サーバが図 1 のサブ検索システムに相当する．すなわち，検索ゲートウェイにおいて NRA を実行し，検索ゲートウェイから SA によって検索サーバのインデックスへの適合度の降順の連続アクセスを行う．検索サーバの数を n とすると，インデックスは単語分散手法により n 個に分割され，検索サーバ $P_i(1 \le i \le n)$ がインデックス $I_i(W_i(D_{all}))$ を保持している．検索対象が持つ属性 V =(文書中に記述された単語) として，単純質問 $(V, word)$ に対して検索ゲートウェイは，$word \in W_i(D_{all})$ であった場合，検索サーバ P_i のインデックス $I_i(W_i(D_{all}))$ に対する SA によって，サブ検索結果 $R((V, word))$ を取得する．

ここで，SA による適合度の降順の連続アクセスを

表 2 入力した全質問における単純質問の個数
Table 2 Classification of input queries based on the number of the included words.

単純質問の個数	質問の数	出現割合
1	12,413	19.93%
2	24,809	39.83%
3	15,987	25.67%
4	5,922	9.51%
5	1,986	3.19%
6 以上	1,173	1.88%
全質問	62,290	100.00%

行うには，ある単語のサブ検索結果の集合について適合度による順位付けが必要となる．図 6 のシステムでは単語分散手法を用いているため，ある単語に関するインデックスはある単一の検索サーバのみが保持することになる．したがって，ある単語のサブ検索結果の集合内での適合度による順位付けが可能となる．

検索ゲートウェイで NRA を実行し，検索サーバのインデックスへの SA を繰り返しながら，質問に適合する検索対象のうち適合度の上位 k 件を検索ゲートウェイが求めユーザへ出力する．

5.2 実験環境

検索サーバとして PentiumII 450 MHz の CPU，512 MB の主記憶から構成される計算機 8 台を使用し，それらを 100 Mbps のイーサネットスイッチングハブで接続した．また，検索ゲートウェイとして Pentium4 1.8 GHz の CPU，1 GB の主記憶で構成される計算機 1 台を使用した．これらの計算機を用いて単語分散型 WWW 並列全文検索システムを構築し，評価実験を行った．

検索システムが検索対象とする文書は WWW から収集した文書 100 万件とした．これらの文書は，我々が作成した収集ロボットを用いて無作為に収集した．収集した文書には英語以外の文書も含まれる．この 100 万件の文書から作成したインデックスを実験に用いた．入力する質問として，検索システム MetaCrawler [10] に入力された質問を用いた．MetaCrawler が質問を公開している Web ページ（リアルタイム更新）を数秒ごとに取得し，英単語からなる質問のみを収集した．実験に用いた質問が含む単純質問の個数を**表 2** に示す．システムの構成とアルゴリズムの性質上，NRA が処理する質問は複数の単純質問を含む質問のみであり，その個数は 49,877 個である．評価項目は，システムが質問を受付けてから検索結果を出力するまでの時間である検索応答時間とした．

表 3 終了条件 C_1，C_1' が成立する質問の数とその割合
Table 3 Number of queries such that condition C_1 (resp. C_1') terminates the algorithm.

k	1	10	100
改善前	22,076（44.26%）	17,263（34.61%）	11,953（23.96%）
改善後	21,704（43.52%）	17,232（34.55%）	11,948（23.95%）

5.3 実験結果と考察

5.3.1 終了条件の成立

改善前（改善後）の NRA で終了条件 C_1（C_1'）が成立してアルゴリズムが終了する場合と，C_2 が成立して終了する場合を比べると，前者の方が処理時間は短くなる．すなわち，C_1，C_1' が成立した質問については NRA が有効であったといえる．本項では，入力された質問のうち，C_1，C_1' が成立して終了する質問の割合を調べた実験の結果を示す．

$k = 1$，10，100 のそれぞれの場合について，入力した質問の中で終了条件 C_1，C_1' が成立した質問の数 n と，NRA の処理対象である複数の単純質問からなる質問の数（49,877 個）に対する n の割合を**表 3** に示す．実験に用いたシステムの仕様と NRA の性質上，単純質問 1 つで構成される質問は NRA の処理対象ではなく，複数の単純質問を含む質問のみが NRA により処理される．表から，入力された質問のうち約 25%から約 45%の質問に NRA が有効であることが分かる．また k の値が小さいほど，C_1，C_1' が成立する質問の数の割合が大きい傾向にあることが分かる．

また，改善前では C_1 が成立して終了したが，改善後では C_1' が成立せず C_2 が成立して終了した質問がある．これらの質問は改善後より改善前の方がアルゴリズムが早く終了するが，表 3 から分かるとおり，このような質問の数は少ない．

5.3.2 検索応答時間

図 7 は改善前，**図 8** は改善後，それぞれの検索応答時間を表すグラフである．横軸は質問に含まれるすべての単語のサブ検索結果について SA によって取得したサブ検索結果の総件数を，縦軸に検索応答時間を表し，グラフ中の各点が質問 1 つを表している．適合度算出関数 t は min とし，$k = 10$ とした．また，あるサブ検索結果について，1 回の SA によって取得する件数 $step$ を 5000 とした．

図 7 の改善前では放物線のグラフとなっており，SA によって取得したサブ検索結果の総件数が多い質問ほど検索応答時間が長くなり，性能が悪化している．一方，図 8 の改善後でも放物線のグラフであり，SA に

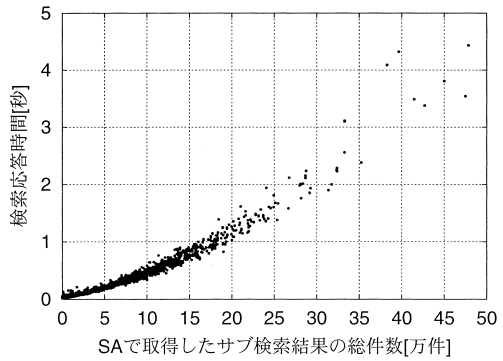

図 7 改善前の検索応答時間，$k = 10$，step=5000
Fig. 7 Search response time by the existing method: k=10, step=5000.

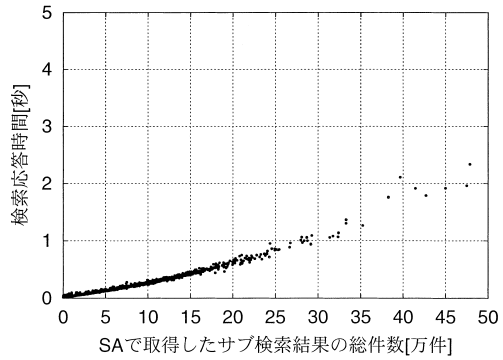

図 8 改善後の検索応答時間，$k = 10$，step=5000
Fig. 8 Search response time by the proposed method: k=10, step=5000.

よって取得した総件数が多くなるにつれて性能は悪化しているが，その度合は改善前ほどではない．

図 7，図 8 から改善前よりも改善後のアルゴリズムの方が，性能が良いことが分かる．

次に，**表 4**，**表 5**，**表 6** にそれぞれ step = 1000，5000，10000 について $k = 1$，10，100 における改善前後の平均検索応答時間とその内訳を示す．

表から，改善前では step が小さい場合ほど，平均検索応答時間に占める上限値計算に要する時間の割合が大きくなっていることが分かる．適合度の上限値は深さ b のとき，すべての単純質問 $q_i(1 \leq i \leq m)$ のサブ検索結果 $R(q_i)$ のうち，SA によって取得済みのサブ検索結果 $R_{1..b}(q_i)$ に含まれるすべての検索対象の集合 $D^{(b)}(q)$ について更新する必要がある．すなわち，上限値を計算する必要のある検索対象の集合は，SA によってサブ検索結果の集合を 1 回目に取得したときは $D^{(\mathrm{step})}(q)$，2 回目は $D^{(2\times\mathrm{step})}(q)$，3 回目は $D^{(3\times\mathrm{step})}(q)$ となる．したがって，step を小さくすると上限値を計算する回数が増えるため，上限値算出処理にかかる時間が長くなる．ここで，質問やシステムによってはアルゴリズム終了時の深さの値が小さい場合があり，step が小さい方が検索応答時間が短くなる場合があることを付け加えておく．

一方，改善後では上限値計算が省略可能であり，下限値計算やその他の処理にかかる時間は改善前とほぼ同じである．したがって，改善前において上限値計算に要する時間の分だけ，改善後では平均検索応答時間を短縮できている．また，改善後の NRA では上限値計算を省略できるため，平均検索応答時間に占める下限値計算に要する時間の割合が大きくなっている．下限値計算の計算量は，アルゴリズム終了時の深さを b，単純質問の個数を m とすると $O(bm)$（3.6 節参照）であるため，step の変化による下限値計算に要する時間の変化は小さく，平均検索応答時間の変化も小さい．

表 4 平均検索応答時間の内訳（step=1000）（単位：秒）
Table 4 A breakdown of the average response time: step=1000.

k		下限値計算	上限値計算	その他	平均検索応答時間
1	改善前	0.026	0.085	0.033	0.144
	改善後	0.027	—	0.034	0.060
10	改善前	0.037	0.151	0.038	0.227
	改善後	0.037	—	0.039	0.076
100	改善前	0.057	0.291	0.080	0.429
	改善後	0.057	—	0.079	0.136

表 5 平均検索応答時間の内訳（step=5000）（単位：秒）
Table 5 A breakdown of the average response time: step=5000.

k		下限値計算	上限値計算	その他	平均検索応答時間
1	改善前	0.033	0.024	0.030	0.086
	改善後	0.033	—	0.030	0.063
10	改善前	0.043	0.039	0.039	0.120
	改善後	0.043	—	0.036	0.078
100	改善前	0.062	0.069	0.076	0.207
	改善後	0.062	—	0.076	0.138

表 6 平均検索応答時間の内訳（step=10000）（単位：秒）
Table 6 A breakdown of the average response time: step=10000.

k		下限値計算	上限値計算	その他	平均検索応答時間
1	改善前	0.040	0.018	0.031	0.089
	改善後	0.040	—	0.031	0.071
10	改善前	0.049	0.026	0.037	0.112
	改善後	0.049	—	0.037	0.086
100	改善前	0.067	0.042	0.078	0.187
	改善後	0.067	—	0.078	0.145

この結果から，4 章で述べた改善手法により，検索応答時間を削減できていることが分かる．

6. まとめと今後の課題

メタ検索システムにおいて，ユーザが入力した質問に適合する検索対象のうち適合度の上位 k 件を出力する NRA アルゴリズムについて，適合度算出関数として最小値関数を用いる場合，アルゴリズムを改善しボトルネックとなる処理を削除できることを示した．また，システム内部に NRA を適用した単語分散型 WWW 並列全文検索システムを構築し，評価実験を行った．実験結果から，提案する改善手法により検索応答時間を短縮できることを示した．

今後の課題として，最小値関数以外の適合度算出関数を用いた場合の NRA の改善や，NRA とともに提案された他のアルゴリズムと改善後の NRA との比較評価があげられる．また，本論文の実験では，NRA を並列化されたシステムの内部に適用して評価実験を行ったが，メタ検索システムへ NRA を適用して実験することが考えられる．

謝辞 本研究の一部は，日本学術振興会未来開拓学術研究推進事業（JSPS-RFTF99I00903），科学研究費補助金基盤研究（C）（2）（14580374），NEC ネットワークス開発研究所および栢森情報科学振興財団の補助による．

参 考 文 献

1) Baeza-Yates, R.A. and Ribeiro-Neto, B.A.: *Modern Information Retrieval*, ACM Press/ Addison-Wesley (1999).
2) Fagin, R.: Combining Fuzzy Information: An Overview, *SIGMOD*, Vol.31, No.2, pp.109–118 (2002).
3) Fagin, R., Lotem, A. and Naor, M.: Optimal Aggregation Algorithms for Middleware, *Symposium on Principles of Database Systems* (2001).
4) Frakes, W. and Baeza-Yates, R.A.: *Information Retrieval: Data Structure and Algorithms*, Prentice Hall (1992).
5) Günter, U., Balke, W.-T. and Kießling, W.: Towards Efficient Multi-Feature Queries in Heterogeneous Environments, *Proc. IEEE International Conference on Information Technology: Coding and Computing* (*ITCC 2001*), pp.622–628 (2002).
6) 速水賢史，竹野 浩，永瀬智哉，藤本典幸，萩原兼一：スケーラビリティのある WWW 並列全文検索システム構築法の提案と評価，情報処理学会研究報告，2001-DBS-123, pp.45–52 (2001).
7) Jeong, B.-S. and Omiecinski, E.: Inverted File Partitioning Schemes in Multiple Disk Systems, *IEEE Trans. Parallel and Distributed Systems*, Vol.6, No.2, pp.142–153 (1995).
8) Lee, J.H.: Analyses of Multiple Evidence Combination, *Proc. 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.267–276 (1997).
9) Meng, W., Yu, C.T. and Liu, K.-L.: Building efficient and effective metasearch engines, *ACM Computing Surveys*, Vol.34, No.1, pp.48–89 (2002).
10) metacrawler. http://www.metacrawler.com/
11) Orlando, S., Perego, R. and Silvestri, F.: Design of a Parallel and Distributed Web Search Engine, *The 2001 Parallel Computing Conference* (*ParCo 2001*), pp.197–204 (2001).
12) Shaw, J.A. and Fox, E.A.: Combination of multiple searches, *Proc. 2nd Text REtrieval Conference* (*TREC-2*), *National Institute of Standards and Technology Special Publication 500-215*, pp.243–252 (1994).
13) Silverstein, C., Henzinger, M.R., Marais, H. and Moricz, M.: Analysis of a Very Large Web Search Engine Query Log, *SIGIR Forum*, Vol.33, No.1, pp.6–12 (1999).

(平成 14 年 12 月 27 日受付)
(平成 15 年 7 月 9 日採録)

（担当編集委員 定兼 邦彦）

河村 一史

平成 13 年大阪大学基礎工学部情報科学科卒業．平成 15 年同大学院基礎工学研究科博士前期課程修了．同年，株式会社東芝入社．並列処理，情報検索等に興味を持つ．

藤本 典幸（正会員）

平成 4 年大阪大学基礎工学部情報工学科卒業．平成 6 年同大学院基礎工学研究科博士前期課程修了．平成 9 年同大学院基礎工学研究科博士後期課程単位取得退学．同年大阪大学大学院基礎工学研究科助手．平成 14 年より大阪大学大学院情報科学研究科助教授．工学博士．並列処理，Web 検索，組合せ最適化，近似アルゴリズム等に興味を持つ．

辻 裕樹

平成 14 年大阪大学基礎工学部情報科学科中退．現在，同大学大学院情報科学研究科博士前期課程在学中．Web コミュニティに関する研究に従事．

萩原 兼一（正会員）

昭和 49 年大阪大学基礎工学部情報工学科卒業．昭和 54 年同大学院基礎工学研究科博士課程修了．工学博士．同大学基礎工学部助手，講師，助教授を経て，平成 5 年奈良先端科学技術大学院大学教授．平成 6 年より大阪大学教授．平成 4 年〜5 年文部省在外研究員（米国メリーランド大学）．現在，並列処理の基礎および応用に興味を持っている．