データベースシステム 129－4
（２００３．１．２３）

# 最小値関数により適合度を算出するNRA検索アルゴリズムの改善

河村一史[†] 辻 裕樹[††]
藤本典幸[††] 萩原兼一[††]

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のインデックスへのアクセス方法にある制限を置いた場合，NRA は最適なアルゴリズムであるが，計算量の多さが問題点である．本報告では，適合度算出関数を最小値関数に限定する場合，NRA の計算量を削減できることを示す．また，単語分割型 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から，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がわかった．

# An Improvement of the NRA Algorithm with the Min Function

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

The NRA is a meta search algorithm which retrieves the objects with the $k$ highest scores from a collection of objects distributed on subsystems for a given query (where $k$ is a constant). Under some restriction against the access method to indices of the subsystems, the NRA is optimal. However there is the problem that the complexity of the NRA is high. In this report, it is shown that the complexity of the NRA can be reduced if the min function is used to combine scores computed by subsystems. 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共に，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一つとして，メタ検索システム[4]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一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，検索結果をその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[6]．また，ユーザにとっては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一つとして NRA(No Random Accesses) アルゴリズム[2] 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，最適なアルゴリズムである[2]．しかし NRA には，計算量が $\Omega(b^2m)$（$b$ =Sorted Access で取得する検索結果件数，$m$ = 質問に含まれる単純質問の数）でボトルネックとなる処理があり，検索処理にかかる時間が長くなるという問題点がある．

本報告では，適合度算出において最も一般的に用いられている最小値関数[2] を使用する場合，NRA のボトルネック

† 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

となる処理を削減できることを示す．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 台構成の単語分割型 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 3.036 秒から 0.274 秒に短縮できることがわかった．

本報告では，2 章において対象とする検索システムについて述べ，3 章で既存の NRA アルゴリズムについて説明し，その問題点を示す．そして，4 章で提案する改善手法について説明する．最後に，5 章において改善手法の効果を評価するための実験を行う．

## 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け付ける質問 $q$ 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．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．文書検索システムの場合（例：$A$ = 文書に含まれる単語，$V$ = 単語 $word$），文書 $d$ 内での単語 $word$ の出現頻度から算出する値 TF と，$word$ に適合する検索対象の数から算出する値 IDF[3] を用いた TF/IDF などの方法がある．また，画像検索システムの場合（例：$A$ = 色，$V$ = 赤），画像 $d$ のデータ内容を解析し特徴量を求める方法などがある．

また，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 等が挙げられる．

ここで，$G(q_i, d) = x_i(d)$ として，すべての $i$ $(1 \leq i \leq m)$ について $x_i(d) \leq x'_i(d)$ が成り立つならば，$t(x_1(d), \cdots, x_m(d)) \leq t(x'_1(d), \cdots, x'_m(d))$ 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．適合度算出関数の例として挙げた min，sum，avg はすべて単調増加関数である．

### 2.3 検 索 結 果

2.1 節，2.2 節の定義を用いて，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

検索システムの一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，一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 $q$ を受け付ける．

(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一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二種類に分類できる．

（typeA）検索対象が持つ属性毎に，異なるサブ検索システムを用いるメタ検索システム

（typeB）検索対象が持つ，ある一つの属性に対するサブ

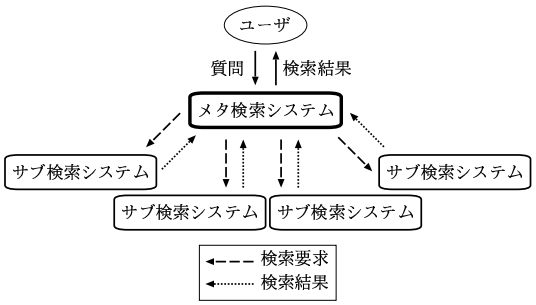


**図 1** メタ検索システム

検索システムをすべて用いるメタ検索システム

typeA のメタ検索システムに質問 $q$ 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．

typeB のメタ検索システムに質問 $q$ が入力された場合，すべてのサブ検索システムに $q$ を検索要求として送信する．

(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，編集して一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．ここで，typeB のメタ検索システムの場合には，サブ検索システムの数を $n$ とすると，$n$ 個のサブ検索結果の集合を取得することとなる．それらを一つの検索結果に統合する際には，$q$ に含まれる演算子とは別に and，or といった演算子をユーザが指定し，演算子に応じて一つの検索結果に統合する．このとき，サブ検索システムが算出する検索対象 $d$ の適合度を $G_1(q,d), G_2(q,d), \cdots, G_n(q,d)$ とすると，統合された検索結果の適合度は and，or に対応する適合度算出関数 $t'_1$，$t'_2$ を用いて $G(q,d) = t'_1(G_1(q,d), G_2(q,d), \cdots, G_n(q,d))$（and の場合）と算出される．

## 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．

ここで，NRA では演算子として and のみを考えており，演算子 and に対する適合度算出関数を $t$ とする．以降では，説明の便宜上 2.4 節の typeA のメタ検索システムを考えるが，typeB のメタ検索システムに対しても NRA は適用可能であり，提案する改善手法についても同様である．また，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．

### 3.1 NRA アルゴリズムの概要

検索システムのユーザが目的とする情報は，通常検索結果のうち適合度の高い部分集合の中に存在する場合が多い．検索システムが検索対象に対して算出する適合度の精度が良い場合ほど，ユーザが目的とする情報の適合度が高くなるためこの傾向にあると言える．そのため，より良い適合度算出手法の研究が進むにつれて，今後この傾向は更に強くなると言える．

以上の傾向をふまえて，NRA は質問 $q$ を入力として検索対象集合 $D_{1..k}(q)$ を出力とする．ここで，$D_{i..j}(q)(i \leq j)$ は質問 $q$ に適合する検索対象集合 $D(q)$ 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表す．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，適合度の順位の情報については，定数 $k$ を $1, 2, \cdots, k$ と順に増やしながら NRA を繰り返し実行することで $D_{1..k}(q)$ 内の順位を求めることができる．

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，ある検索対象 $d1$ 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．

### 3.2 Sorted Access

メタ検索システムによる各サブ検索システムのインデックスに対する，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のインデックスにアクセスする．

SA の例を**図 2** 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一つに定まり，$R(q_i)$ 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．

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，$b = n1$

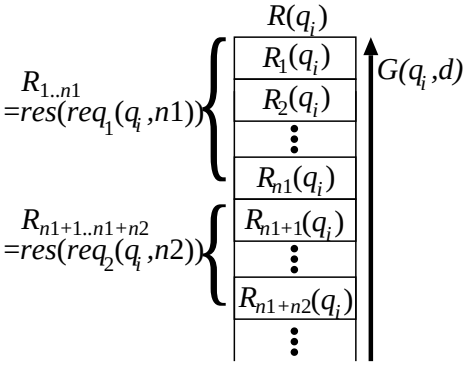


**図 2** Sorted Access

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であることを表す．

NRA は質問 $q$ に含まれるすべての単純質問 $q_i$ について，SA によってサブ検索結果 $R(q_i)$ に対する適合度の降順の連続アクセスを繰り返し行う．ここで，$q$ に対する検索対象 $d$ の適合度 $G(q, d)$ をメタ検索システムが算出するには，すべての $q_i$ に対する適合度 $G(q_i, d)$ を取得する必要がある．深さが $b$ のとき，すべての単純質問に対する適合度を取得済の検索対象については，$q$ に対する適合度を算出できるが，未取得の適合度が一つ以上ある検索対象については，$q$ に対する適合度を算出できない．このような検索対象に対して NRA では，$q$ に対する適合度の上限値と下限値を算出することでその値域を求める．そして，求めた値域から検索対象間の $q$ に対する適合度の大小関係を判定し，$q$ に適合する検索対象のうち上位 $k$ 件を求める．

### 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 $x_i(d)$ と表す．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．

また，$t$ は単調増加関数であるため，以下の式が成り立つ．

$$W^{(b)}(q, d) \leq G(q, d)$$

### 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．

ここで，すべての $R(q_i)$ に対して SA による適合度の降順の連続アクセスを行うため，未取得の適合度 $x_n(d)$ $(l+1 \leq n \leq m)$ のすべてについて $x_n(d) \leq \underline{x}_n^{(b)}$ が成り立つ．また，$t$ は単調増加関数であるから以下の式が成り立つ．

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

### 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，一回の 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 $b = b + step$ 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度に，以下の情報を更新する．
   - (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 $C = \{($ すべての $d' \in D'$ について $B^{(b)}(d') \leq M_k^{(b)})$ または (SA によって $q$ に含まれるすべての単純質問のサブ検索結果 $R(q_i)$ を取得済$)\}$ が不成立ならば，(1) へ戻る．$C$ が成立すれば，$T_k^{(b)}$ を出力として終了する．

NRA の概念を**図 3** に示す．図中の各直線はそれぞれ質問 $q$ に対する検索対象 $d(\in D(q))$ の適合度の値域を表す．$d_k$ と $d'$ の二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

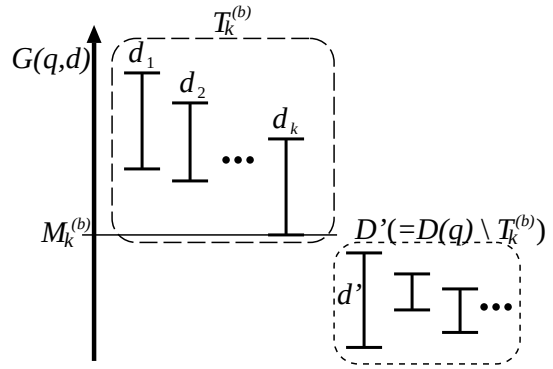


**図 3** NRA による適合度上位 $k$ 件の決定

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，深さを $b$，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) アルゴリズム終了時，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 $O(b'm)$

(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 $O(b'm)$

(b) 下限値 $W^{(b)}(q,d)$ の算出方法は 3.3 節で述べた通りである．ある単純質問について，一回の 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 $R_{b..b+step}(q_i)$ に含まれる検索対象についてのみ更新する必要がある，アルゴリズム終了時，$q_i$ のサブ検索結果は $b'$ 個取得しており，また，単純質問は $m$ 個あるので，計算量は $O(b'm)$

(c) 上限値 $B^{(b)}(q,d)$ の算出方法は 3.4 節で述べた通りである．深さが $b$ のとき，SA によって取得した各単純質問のサブ検索結果 $R_{1..b}(q_i)$ のうち，上位 $b$ 番目の適合度の集合 $\{\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)}\}$ 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 $D^{(b)}(q)$ について上限値を算出する．深さが $b$ のときに上限値の算出に用いる $\{\underline{x}_1^{(b)},\cdots,\underline{x}_m^{(b)}\}$ は，SA によって深さが変更される度に，すなわちサブ検索結果を取得する度に更新されるため，SA によってサブ検索結果を取得する度にすべての検索対象 $d \in D^{(b)}(q)$ について上限値を更新する必要がある．その計算量は文献[2] によると $\Omega(b'^2m)$

(d) $T_k^{(b)}$ の更新は，一回の SA によって取得したサブ検索結果 $R_{b..b+step}(q_i)$ 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，その計算量は $O(b'm)$ であり $M_k^{(b)}$ についても同様である．

(2) 終了条件 $C$ より，SA によって取得したすべてのサブ検索結果 $R_{1..b}(q_i)$ に含まれる検索対象集合 $D^{(b)}(q)$ のうち，すべての $d' \in D'(=D^{(b)}(q)\setminus T_k^{(b)})$ について $B^{(b)}(d') \le M_k^{(b)}$ の比較を行う必要がある．したがって，その計算量は $O(b'^2m)$

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 $\Omega(b'^2m)$ となり，(c) 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，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 $t$ を考えた場合にその適合度算出関数の性質から，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 $C$ を変更することを考える．

適合度算出関数 $t$ として最小値関数 min を考える．min は $l \in \{1,\cdots,m\}$ として，ある検索対象 $d$ とすべての $i(1 \le i \le m)$ について $x_l(d) \le x_i(d)$ が成り立つとき，$\min(x_1(d),\cdots,x_m(d)) = x_l(d)$ である適合度算出関数を表す．

$t = \min$ のとき，NRA の終了条件 $C$ を変更することにより，3.6 節で述べた (c) 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 $t$ として最小値関数 min を用いる場合，3.5 節で示した NRA の処理内容 (1-d) の後に以下の (1-e) を追加し，(2) を以下の (2') に変更する．

(1-e) $Max =$ { 質問 $q$ に含まれるすべての単純質問 $q_i(1 \le i \le m)$ 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 $C' =$ {($Max \le M_k^{(b)}$) または (SA によって $q$ に含まれるすべての単純質問のサブ検索結果 $R(q_i)$ を取得済)} が不成立ならば，(1) へ戻る．$C'$ が成立すれば，$T_k^{(b)}$ を出力として終了する．

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 $C$ でのみ参照した．しかし，改善後の終了条件 $C'$ 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 (1-c)，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．したがって，計算量を削減することができる．

表 1　改善前後の NRA における各処理の計算量

| | 改善前 | 改善後 |
|---|---|---|
| (1-c) | $\Omega(b'^2m)$ | – |
| (1-e) | – | $O(b'm)$ |
| ( 2 ) | $O(b'^2m)$ | – |
| ( 2' ) | – | $O(b'^2m)$ |

### 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．

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．改善前の処理（1-c）は改善後は省略できるため，その計算量 $\Omega(b'^2m)$ を削減できる．

また，改善によって加わった処理（1-e）の計算量は $O(b'm)$ であり，処理 ( 2 ) を変更した処理 ( 2' ) の計算量は $O(b'^2m)$ となり，処理 ( 2 ) と変らない．

### 4.4　改善後の終了条件の証明

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[2] 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 $C'$ 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，$C'$ のうち，条件 {$q$ に含まれるすべての単純質問のサブ検索結果 $R(q_i)$ を SA で取得済 } 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 $\{Max \leq M_k^{(b')}\}(Max = \max(\underline{x}_1^{(b')}, \cdots, \underline{x}_m^{(b')}))$ が成立したとき，$T_k^{(b')} = D_{1..k}(q)$ である．

**[証明]** $T_k^{(b')} = \{d_1^T, d_2^T, \cdots, d_k^T\}$，$T_k^{(b')}$ 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．(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)　$B^{(b')}(q, d') = x_j(d')\ (1 \leq j \leq l)$ のとき

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

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

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

が成立する．

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

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

が成立するのは明らかである．

(a)，(b) より，

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

が成立する．(1)，(2)，(4)，(5) より，

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

が成立する．

(i)，(ii) より，証明終わり

## 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．

### 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　単語分割型 WWW 並列全文検索システム

実験に用いた検索システムは，WWW 並列全文検索システム（**図 4**）である．検索対象 $D_{all}$ =（WWW 上に公開された文書），検索対象が持つ属性 $V$ =（文書中に記述されている単語）である．

システムは図 4 に示す通り，複数の計算機を用いて並列化している．システムを構成する計算機は，ユーザとの入出力を担当する検索ゲートウェイと，インデックスを保持

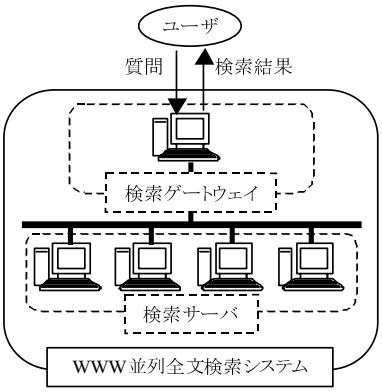

**図 4** WWW 並列全文検索システム

し，その検索を担当する検索サーバの二つに，その役割を分担する．

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する必要がある．インデックスの分割手法の一つとして，単語分割手法[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，単語集合の分割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．システムでは，検索サーバの数を $n$ とし，単語分割手法によりインデックスを $n$ 個に分割することとした．

この単語分割手法を WWW 並列全文検索システムに適用した単語分割型 WWW 並列全文検索システムを構築した．

#### 5.1.2 NRA の適用

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

単語分割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割手法により $n$ 個に分割され，検索サーバ $P_i(1 \leq i \leq n)$ 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 4 のシステムでは単語分割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 450MHz の CPU，メモリ 512MB から構成される計算機 8 台を使用し，それらを 100Mbps 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 1.8GHz の CPU，メモリ 1GB 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割型 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，100 万件の文書から作成したインデックスを用いた．システムへ入力する質問として，検索システム Metacrawler[5] に入力された質問 1 万件を用いた．評価項目は，システムが質問を受け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．

### 5.3 実験結果

**図 5** は改善前，**図 6** 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一つを表している．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，一回の SA によって取得する件数 $step$ を 1000 とした．

図 5 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 6 の改善後でも放物線のグラフであり，SA に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 5，図 6 から改善前よりも改善後のアルゴリズムの方が，性能が良いことがわかる．

$k = 1$，10，100 のときの改善前後の平均検索応答時間の内訳を**表 2**（$step = 1000$），**表 3**（$step = 10000$）に示す．表 2 と表 3 を見ると，改善前では $step$ の値が大きい方が上限値計算にかかる時間が大幅に短縮され，その結果平均検索応答時間も大幅に短縮されていることがわかる．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一回目に取得したときは $D^{(step)}(q)$，二回目は $D^{(2\times step)}(q)$，三回目は $D^{(3\times step)}(q)$ となる．したがって，$step$ を大きくすると上限値を計算する回数が減るため，上限値算出処理にかかる時間が短縮される．ここで，質問や

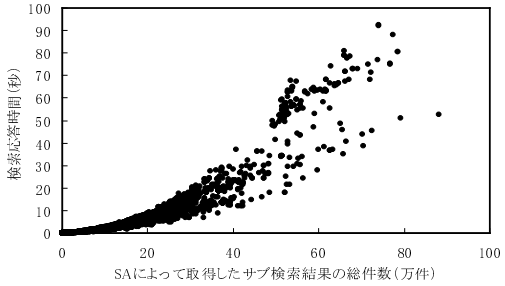


**図 5** 検索応答時間（改善前，$k = 10$）

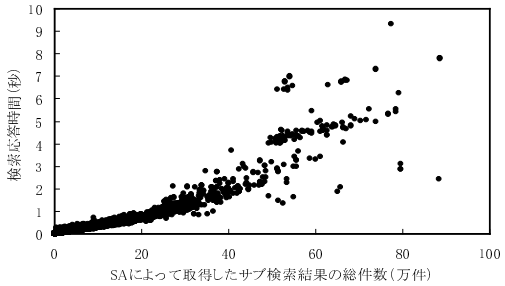


**図 6** 検索応答時間（改善後，$k = 10$）

**表 2** 平均検索応答時間の内訳（$step = 1000$）（単位：秒）

| $k$ | | 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 | 0.131 | 2.649 | 0.195 | 2.975 |
| | 改善後 | 0.098 | – | 0.154 | 0.252 |
| 10 | 改善前 | 0.137 | 2.696 | 0.204 | 3.036 |
| | 改善後 | 0.109 | – | 0.165 | 0.274 |
| 100 | 改善前 | 0.143 | 2.710 | 0.272 | 3.124 |
| | 改善後 | 0.122 | – | 0.233 | 0.355 |

**表 3** 平均検索応答時間の内訳（$step = 10000$）（単位：秒）

| $k$ | | 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 | 0.118 | 0.263 | 0.146 | 0.527 |
| | 改善後 | 0.104 | – | 0.127 | 0.231 |
| 10 | 改善前 | 0.126 | 0.277 | 0.152 | 0.555 |
| | 改善後 | 0.113 | – | 0.137 | 0.250 |
| 100 | 改善前 | 0.137 | 0.286 | 0.222 | 0.646 |
| | 改善後 | 0.124 | – | 0.202 | 0.327 |

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．

表 2 では改善前での全体に占める上限値計算処理にかかる時間の割合が，約 89%，約 88%，約 87%と高くなっている．また，表 3 においても約 50%，約 50%，約 44%と同様である．この結果から，上限値計算処理がボトルネックとなっていることがわかる．

一方，改善後では上限値計算の処理を省略できるため，改善前の上限値計算の処理時間の分だけ，平均検索応答時間を短縮できている．

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わかる．

## 6. ま と め

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 を適用した単語分割型 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 と共に提案された他のアルゴリズムと改善後の NRA との比較評価が挙げられる．

## 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., Lotem, A. and Naor, M.: Optimal Aggregation Algorithms for Middleware, *Symposium on Principles of Database Systems* (2001).
3) Frakes, W. B. and Baeza-Yates, R.: Information Retrieval : Data Structure and Algorithms, Prentice Hall (1992).
4) 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).
5) metacrawler: http://www.metacrawler.com/.
6) 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).
7) 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 (Jan 2001).