Mem. Fac. Sci. Eng. Shimane Univ.
Series A
34, pp. 65-75 (2000)

文字列探索アルゴリズムの評価方法

大段　雅典*・佐藤　匡正**
*島根大学大学院　総合理工学研究科
**島根大学　総合理工学部　数理・情報システム学科

An Evaluation Method for String Searching Algorithm

Masanori OODAN
Interdisciplinary Division for Research of Science and Engineering, Graduate School of Shimane University
Tadamasa SATOU
Department of Mathematics and Computer Science, Interdisciplinary Faculty of Science and Engineering, Shimane University

Absiract

String Searching is to find a substring specified in a given string. Some algorithms which aim to improve the speed are known, as frequent character coincidence tests in these strings are done. String Searching algorithms are generally evaluated by comparing the processing speed. The processing speed is running time, computational complexity, steps on coincidence test and so on. So far, String Searching algorithms has evaluated through theoretical values by them. There could be some problems including input (strings) classes are not considered. This presented to propose a practical evaluation method for String Searching algorithms and to establish the effective algorithms to input classes.

要　　約

文字列探索は，与えられた文字列中に，探索文字列と呼ばれる文字列と一致する部分列の有無や，その位置を求めるものである．この処理では，両方の文字列を構成している文字の一致比較が頻繁に行われるため，速度の向上を狙った様々なアルゴリズムが知られている．これらのアルゴリズムの優劣評価では，処理速度を基準とするのが一般的である．処理速度には，実行時間，計算量，文字の比較回数などがあるが，既存の評価では，それらを理論上で算出した値の比較に止まっている．これには，様々な種類の入力（文字列）に配慮されていないなどの問題点がある．そこで，より実用的な文字列探索アルゴリズムの評価方法を提案し，入力種別ごとに有効なアルゴリズムを確定した．

1. 序　　論

アルゴリズム（algorithm）とは，問題を解くための手法（解法）を定めたものである．

アルゴリズムはその適用分野によって考え方が大きく異なり，整列，文字列処理，暗号化，数理統計，並列処理，画像処理などに分類されることが多い．もともと，初期の計算機システムにおいては，利用できる計算能力や記憶域の大きさといった資源が制限されていたため，この解消を目的として様々な工夫が行われた．この整理体系化がアルゴリズム論の基本的な取り組みである．今日，ハードウェア性能の向上により，初期の頃のような資源の制約は大幅に緩和されている．しかし，一方では，計算機システムの適用拡大に伴い，高性能なアルゴリズムの要求が従前以上に強まっている．

文字列探索は，文字列処理に関するアルゴリズムの一種である．これは，コンパイラやエディタなどの言語処理プログラムにおいて，字句分けや構文解析の処理に用いられたり，あるいは生命科学において，蛋白質の特定の塩基配列を見つけるために用いられたりもしている．この文字列探索アルゴリズムは，探索文字列及び被探索文字列の二種類の文字列が用いられ，被探索文字列中に一致する探索文字列の有無やその位置を求めるものである．この処理では，文字列に含まれる文字を単位とした一致比較が頻繁に行われるため，処理速度の向上を狙って，様々な工夫を凝らしたアルゴリズムが提案されている．代表的なアルゴリズムとして，力まかせ（BF）法，クヌースーモーリスープラット（KMP）法，ボイヤームーア（BM）法，ラビンーカープ（RK）法などが知られている．

これらのうち，最も基本的なものは力まかせ法で，その名の通り全ての文字を逐一比較するが，他の方法はこの回数を減少させるべく，既知，あるいは確定している探索文字列の情報を用いたり，探索順序を逆にしたりといった工夫を行っている．

ところが，こうしたアルゴリズムの利用者の立場に立つと，多様さは選択の幅の拡大化を招く．つまり，対象とする問題に関してどのアルゴリズムを選択すべきかの基準が必要となってくる．もちろんアルゴリズムの解説書（例えば文献1）には，アルゴリズムの特徴が示されており，これによって選択するという手がある．しかし，これは必ずしも的を射ているとは限らない．筆者らの経験によれば，あるシステム開発における文字列探索処理に対して，BF法によるものを，その処理速度向上を狙ってKMP法で設計，構築し直したが，その成果が現れなかった．これは，上の評価に反することである．

そこで，本論文では文字列探索アルゴリズムに対する評価方法を提案し，この提案の評価について論ずる．

2. 評価方法の提案と実践

2.1 従来の評価方式

(1) 評価項目

アルゴリズムの評価では，一般的に省資源使用に根ざした能率の良さが基準となる．この基準に沿って考えると，次の項目が挙げられる．

- 機能の充足性
- プログラム占有メモリ量，使用メモリ量
- 処理速度（応答時間，動的命令実行数など）

- 処理能力

文字列探索探索アルゴリズムにおいては，文字の頻繁な一致比較がなされ処理時間が掛かるので，評価項目としては処理速度に重点がおかれる．そこで，ここでは探索時の文字の比較回数や実行ステップ数，計算量を基準とする．

(2) 比較値

アルゴリズムの評価項目は定められているが，比較する値には文字の比較回数や実行ステップ数，またその最大値や平均値が混在して用いられている．その具体例を示す．

例えば，探索文字列長を m，被探索文字列長を n として，次の評価が示されている[1]．

■ BF 法

文字の比較回数が約 mn 回必要となる．

■ KMP 法

必要な文字の比較回数は 2 (m+n) 回以下である．

■ BM 法

文字の比較回数の上限は m+n に比例し，入力文字列の文字種が探索文字列の長さに較べて十分多ければ，およそ n/m 回である．

■ RK 法

探索文字列と同じハッシュ値の部分列を見つけるのは，n+m に比例する時間が必要である．照合を確認するには，文字の直接比較が必要となり，理論的には依然として最悪の場合，O (nm) ステップが必要となる，ところが，実用上では，ほとんど n+m ステップである．

この違いの生じた理由は，次の様に整理できる．

- 実測値ではない
- 統一された尺度ではない

処理速度に比較回数及び計算量と，異なる尺度が用いられている．

- 入力（文字列）種別への配慮

入力となる列には，数字の列や文字の列などがあるが，これらへの配慮がない．

2.2 方法のあらまし

(1) 比較項目

同一の比較仕様に基づき，アルゴリズムを変えて作成するプログラムに対して，文字一致比較の回数を比較する．このとき，このプログラムに対する被探索文字列及び探索文字列は同一のものを用いる．また，比較回数はソース・プログラムにおける比較文の動的実行回数で表す．

(2) 比較の仕様

計測する文字ごとの比較回数の総計（以降，比較回数）とは，一致比較を行う度の一致回数，不一致回数の総計である．RK 法では，ハッシュ値の比較と文字の比較両方の合計を比較回数の総計とする．探索時の途中で完全一致の部分列が検出されても探索は続行する．つまり，探索文字列長以上の被探索文字列が存在し，被探索文字列にまだ探索の余地がある場

合，被探索文字列中の完全一致した文字列の後から引き続き探索を行う．

(3) 対象アルゴリズム

代表的な文字列探索アルゴリズムとしては4種類であるが，BM法は，MM法とBSKMP法による複合アルゴリズムと考えることができるため，対象アルゴリズムは5種類とする．

■ BF法

文字列の先頭から1文字ずつ全ての文字に対して一致比較を行う．最も単純であるが，効率は悪い．

■ KMP法

文字列の一致比較を開始して，不一致を検出した文字から前の文字に戻ることなく次の一致比較を開始する．このとき，この被探索文字列中を後戻りしないことがBF法からの改良点である．

■ MM法

常に探索文字列の最後の文字から先頭に向かって文字の一致比較を行い，不一致を検出したときの被探索文字と同じ文字が探索列中に無い場合，次の一致比較を探索文字列長分先の被探索文字から開始する．

■ BSKMP法

既に探索を行った部分列の情報を用いるが，一致比較の順序方向がMM法と同じである．このため，KMP法の逆走査版と言える．

■ RK法

ハッシュ技法を用いるが，仕様では，探索文字列が単一であるので，ハッシュ表は持たない．

2.3 入力文字列の設計

探索の速度は，基本的には与えられる入力(文字列)によるが，主として，入力列の種類，探索文字列と被探索文字列の長さ，部分一致する文字数，完全一致を検出する回数に整理することができる．これから，入力列の種類は，ビット列，数字列，英文，和文の4種類とする．

(1) 被探索文字列

表1のように入力列種ごとに被探索列を固定する．ビット列と数字列については，任意に並べた10文字の数列を繰り返し用いて最大1000文字とする．英文と和文は表中の書籍から自然な1000文字の文章を抜き出して用いる．

(2) 探索文字列

ビット列と数字列は，表2のように部分一致の割合やその開始点が異なるような条件ごとに数種の探索列を作成する．これは，探索時にアルゴリズムの特性が生かされる場合や，生かされない場合などの全ての場合について比較回数を計測するためである．英文と和文は，条件ごとに探索列を作成すると不自然な単語，文章になるため，被探索列中に存在する適当な語句を探索列とする．ビット列と数字列は20種類ずつ，和文と英文は10種類ずつと

表1　評価に用いる被探索文字列

入力列種別	被探索文字列の値
ビット列	(1111010011)†
数字列	(2098215045)†
英文	'Object Oriented Programming'の文
和文	「帝京物語　序」の文

注）†はクリーネの閉包

表2　ビット列及び数字列の探索文字列

	ビット列	数字列
一致文字無し		333
0文字目から2文字一致	110	208
6文字目から4文字一致	10010	15044
0文字目から9文字一致	1111010010	2098215044
10文字目から前に2文字一致	111	945
5文字目から前に2文字一致	00010	19882
10文字目から前に9文字一致	0111010011	1098215045

する．文字数はアルゴリズムの特性から，3以上とする．表2に数字列の探索文字列の一部を示す．

(3)　懸案事項

①　比較回数の誤差

探索列が**6文字目から4文字一致の**｛15044｝の場合，MM法とBSKMP法での探索時に，被探索列｛2098215045209821504…｝中の4と意図しない部分一致が検出される．文字種が少ないことからビット列の探索の場合，これ以上に意図しない部分一致や完全一致が検出される．この結果として比較回数の計測に誤差をきたすが，それぞれのアルゴリズムに同様の誤差が生じることから，このことは考慮に入れない．

②　RK法の特性

RK法はハッシュ技法を用いるために一文字ずつ比較することがない．従って，その比較回数は，被探索文字列長にのみ依存するため，多種の文字列を探索しても影響がほとんど無い．

③　準備処理の対処

BF法以外のアルゴリズムは，探索を開始し，一致比較を行うまでに準備処理を行う．例えば，MM法は文字列中に現れる全ての文字に対する処理が必要であり，RK法はハッシュ

値を算出する必要がある．しかし，この処理は文字列が大きければ無視できるので考慮しない．

2.4 実　　践

(1) 実測

入力列種がビット列で，{11100} を探索し，比較回数を計測した結果を表3に示す．これから，BF法以外のアルゴリズム全てがBF法と比べて比較回数が少なく，有効であるように思える．

(2) 理論的考察

探索における比較回数Cは次で与えられる．

① RK法を除くアルゴリズム

比較開始文字からの一致文字数を M_i，探索文字列の移動回数を P とすると，C は次式である．

$$\text{比較回数 } C=\sum_{i=1}^{P}(M_i+Q)$$

ここで，完全一致のとき $Q=0$，不一致のとき $Q=1$

② RK法

探索文字列長をm，被探索文字列長をnとすると，同様に，Cは次式である．

$$\text{比較回数 C}=\text{n}-\text{m}+1-\text{S}^*(\text{m}-1)$$

ここで，Sは完全一致検出回数．

3. 分　　析

仕様に基づいて作成したプログラムに，仕様に基づいて作成した被探索文字列，探索文字列全てを入力し，実行させ，計測した比較回数から評価をする．

表3　BF法によるビット列の比較回数表

	探索文字列長						
アルゴリズム	10	50	100	250	500	750	1000
BF法	17	125	260	665	1340	2015	2690
KMP法	9	69	144	369	744	1119	1494
MM法	8	48	98	248	498	748	998
BSKMP法	5	33	68	173	348	523	698
RK法	6	46	96	246	496	746	996

3.1 最小回数に基づく評価

比較回数の最小値から最大値までを，入力列種に関係なくアルゴリズム別にまとめた結果を図1に示す．これから，MM法の比較回数が最小であるので，最も有効なアルゴリズムであるように思える．

しかし，最小値や最大値，平均値をみて評価を行うのは，計算量や比較回数を算出して評価を行うのと変わらない．

3.2 比例に基づく評価

(1) 文字列種別評価

図2に，ビット列におけるBF法に対する比較回数比をアルゴリズムごとに求めた結果を示す．これから，理論どおりの評価と言える結果も現れているが，全ての探索文字列において比が1未満でないことからから，他のアルゴリズムが必ずしもBF法より有効とは言えない．以下にそれぞれの例を挙げる．

■ 探索列 {000}

この結果（x軸左から3つ目の要素）から，BSKMP法，MM法，RK法，KMP法の順に有効であることがわかる．これは，それぞれのアルゴリズムの特性が発揮された場合で，理論上の評価通りの結果が現れている．

■ 探索列 {0111010011}

この結果（x軸右端の要素）から，MM法の比較回数がBF法よりも多いことがわかる．この論拠を以下に述べる．

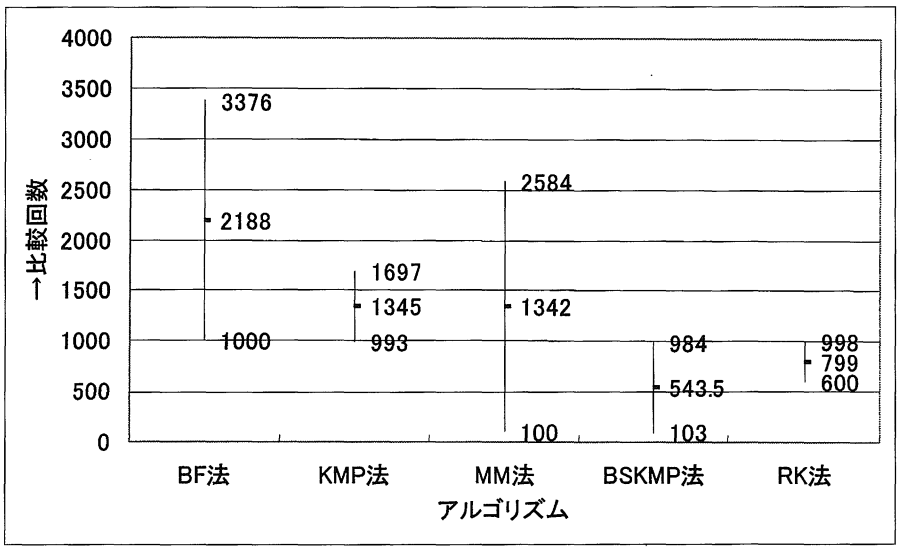

図1 アルゴリズムごとの比較回数の区間

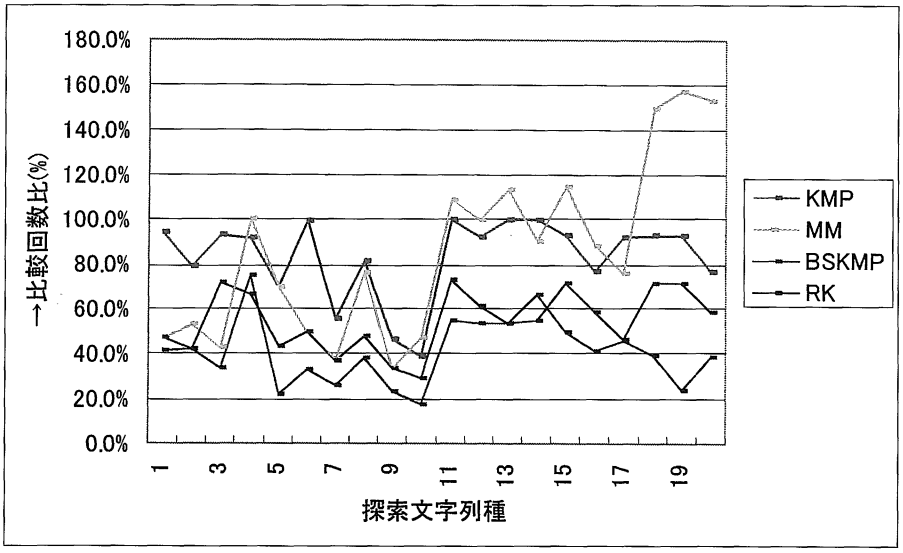

図2 ビット列における比較回数比（対 BF 法）

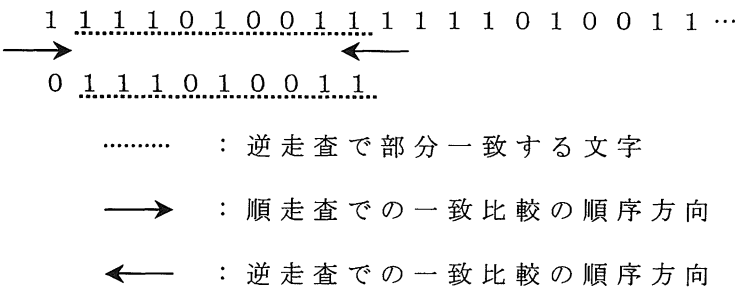

図3 順，逆走査による部分一致文字の違い

① 逆走査法の影響

BF 法とは違い，照合する走査が逆である．従って，同じ探索列でも，一度の探索で部分一致する文字数が，BF 法で探索する場合よりも多いことがある．その回数が多くなると，比較回数は MM 法が多くなる可能性が高くなる．

② ビット列の影響

入力列種がビット列では，文字が {0, 1} の 2 種類しかないために，文字列の移動回数が少ないという特性が生かされない．図 4 は，この事情を示したものである．縦方向に探索回数を，斜め方向には探索文字列の移動の様子を示して探索の様子を表す．

図において，一回目の探索では R との不一致を検出するが，探索文字列中には R は存在

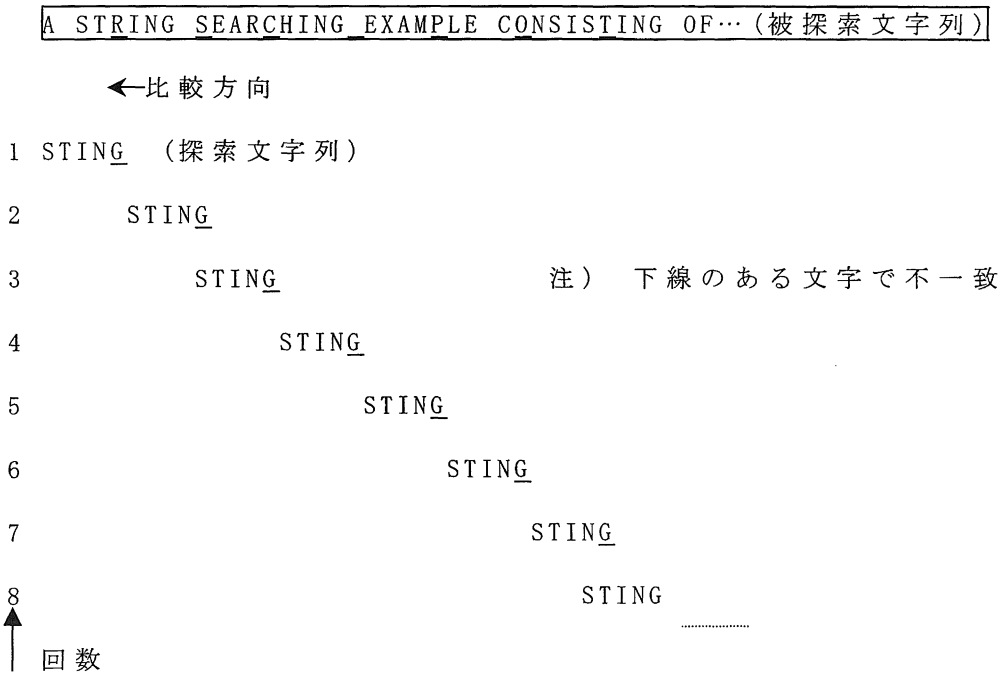

図4 MM法における探索方法

しないので，5文字移動したSから次の比較を開始する．こうして，8回目の探索で完全一致を検出し，さらに探索を続ける．

(2) 総合評価

図5に，入力文字列種別にアルゴリズムの有効性を棒グラフによって示す．アルゴリズムの「有効」とは，BF法を基準としたときに，比較回数費が0.9以下である場合とする．

これから，ビット列と数字列のでは，全てのアルゴリズムが有効性を示している．その値から，ビット列ではBSKMP法とRK法，数字列ではBSKMP法が最も有効であると言える．

英文と和文の自然言語での探索において，有効性の差が顕在化している．MM法は，ほぼ全ての場合において有効であるような値を示しているが，KMP法とRK法は全く示していない．その理由は以下の通りである．

■ KMP法

文字種の増加により，部分一致検出回数が減少している．このため，BF法からの改良点がその効果を発揮しない．図6に，部分一致が多く検出されるビット列の探索と，全く検出されない英字列の探索の様子を示す．

KMP法は，被探索文字列上を逆行し，一度比較を行った文字を再び比較することはな

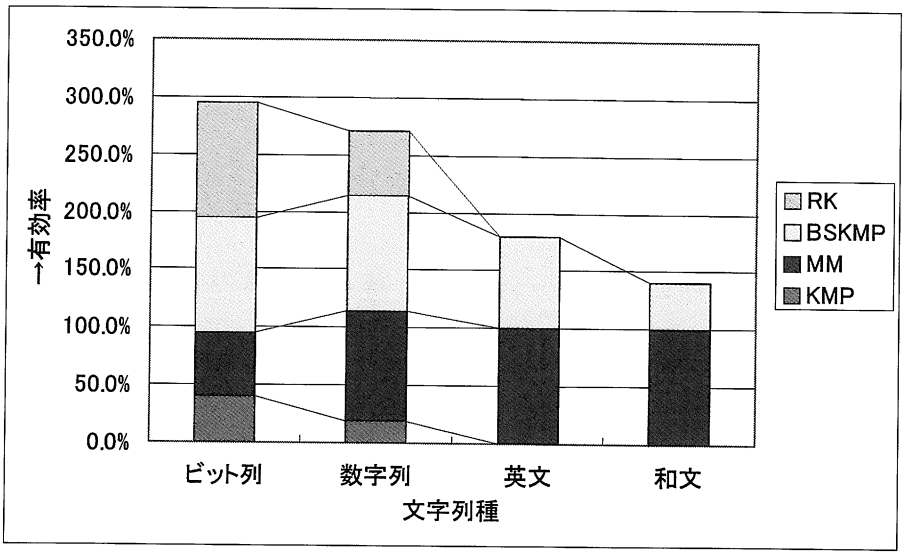

図5 文字列種別アルゴリズムの有効性

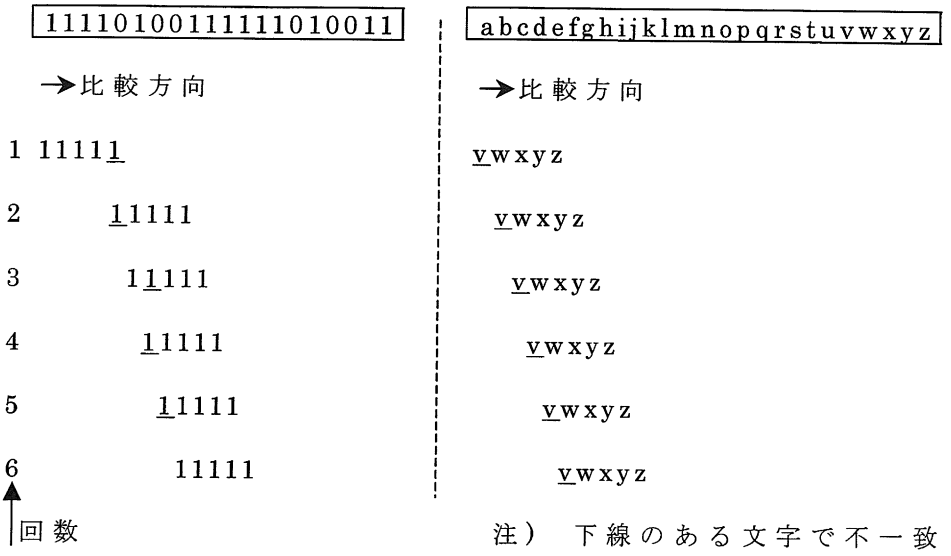

図6 KMP法における探索方法

い．従って，部分一致が多くあればあるほどBF法よりも比較回数は少なくなる．しかし，部分一致検出回数が減少すると，全ての文字に対して一致比較するBF法と違いが無くなっ

表4　アルゴリズムの評価一覧

	文字列種			
アルゴリズム	ビット列	数字列	英　文	和　文
KMP法	△	×	×	×
MM法	△	○	○	○
BSKMP法	○	○	○	△
RK法	○	△	×	×

てくることがこの図からわかる．

■ RK法

BF法の比較回数は，部分一致検出回数が減少すると，最小値である被探索文字列長の近づく．RK法は，入力する文字列種に関係なく，比較回数が被探索文字列長とほぼ同一である，このためにBF法との比較回数の差が現れにくくなる．

表4は，有効かどうかに着目して整理した表である．有効性が50%未満の場合を×，50%以上70%未満の場合を△，70%以上の場合を○で表している．

4. 結　　論

既存の評価法に比べ，現実的な評価方法を提案した．その評価結果からビット列，数字列の探索ではBSKMP法，英文，和文の文字列探索ではMM法が最も有効であることが言える．

参 考 文 献

1) Robert sedgewick "Algorithm in C", Addison-Wesley Publishing Company, Inc. (1990)（野下浩平/星　守/佐藤　創/田口　東：共訳「アルゴリズムC　第2巻」近代科学社）．
2) 大段雅典，佐藤匡正「文字列探索アルゴリズム評価法の提案」，情報処理学会　秋の全国大会講演論文集（2000）．
3) 大段雅典，佐藤匡正「文字列探索におけるクヌースーモーリスープラット法の有効性」，電気・情報関連学会中国支部連合大会講演論文集（2000）．