2J3-1

拡張Potts Modelを用いたクロスワードパズルの解き方

Solving crossword puzzles using Extended Potts Model

神保 一樹[*1] 高村 大也[*2] 奥村 学[*2]
Kazuki Jimbo Hiroya Takamura Manabu Okumura

[*1]東京工業大学 工学部情報工学科
Tokyo Institute of Technology, Department of Computer Science

[*2]東京工業大学 精密工学研究所
Tokyo Institute of Technology, Precision and Intelligence Laboratory

Crossword puzzle is one of the most famous puzzles. Solving crossword puzzles needs not only logic but also vocabulary and commonsense knowledge. That's why it is more difficult to solve it automatically than other puzzles, including sudoku. In this paper, we present Extended Potts Model, an extension of Potts Model, for solving crossword puzzles automatically. Using this model, we can solve crossword puzzles with less computational cost.

1. 研究の背景

クロスワードパズル（もしくは単に「クロスワード」）は，おそらく人々に最もよく知られたパズルのひとつであろう．

パズルといえば，近年では数独[*1]もブームとなったが，数独とクロスワードとではパズルとしての性質が異なる．数独は，いくつかの規則に反することなく数字を埋められれば正解で，この規則に曖昧性はない．つまり，規則をもとに論理的に正解を求めることができる．一方，クロスワードでは「カギ」と呼ばれるヒントから答えとなる単語を連想する必要がある．したがって，すべて論理的に解けるわけではなく，何らかの言語的処理が必要になる．

人間がクロスワードを解くときに脳の中で行っている処理を，いかにして機械に行わせるかという問題は，ある種の人工知能を作るための課題といえる．

2. 関連研究

英語のクロスワードパズルを自動で解く研究は，Keim et al.[3] により行われている．Keim et al. は，American-styleと呼ばれる英語のクロスワードパズルを解くシステムを構築した．このシステムはProverbと名付けられている．Proverbでは，言語資源やアルゴリズムを利用してカギから答え候補のリストを作る段階と，それらを組み合わせてできる最適な解を求める段階とに分けて処理が行われる．前半の段階では，5133パズルからおよそ35万ものカギと答えのペアを集めて作ったデータベースが最も強力にはたらいている．American-styleのクロスワードパズルの特徴は，すべての文字がタテとヨコの両方の単語の一部となっていることである．単語が交差することによる制約が多く利用できる代わりに，カギが短く曖昧であることが多い．また，異なる問題にまったく同一のカギと答えのペアが出現することも比較的多い．そのため，それらのカギを蓄積しておくことにより，すでに蓄積されたカギと同一のカギに対しては，ほぼ正確に答えを返すことができる．最適な解を求める方法に関しては別途論文が出されている [1]．これについての詳細は後述する．これらの処理により，Proverb は，クロスワードパズルを単語正解率[*2]95.3%で解くという性能を誇る．

佐藤 [2] は，日本語のクロスワードパズルを自動的に解く手法を提案した．この手法でも，前述の Proverb と同様，カギから答え候補のリストを作る段階と，それらを組み合わせて最適な解を求める段階とに分けて処理を行う．英語のクロスワードパズルに比べ，日本語のクロスワードパズルはカギが独創的であることが多く，異なる問題にまったく同一のカギと答えのペアが出現することはまれである．そのため，この研究では，クロスワードを集めてデータベースを作るという方針はとっていない．代わりに，カギを連想問題ととらえ，答えを導くためにはどのような連想をすればよいかによってカギを分類している．そして，分類したそれぞれに対して利用する言語資源や探索手法を変えている．一方，後半の最適な解を求める段階は，Proverb と同じ手法を利用している．この研究の手法でクロスワードパズルを解いたときの単語正解率は 44%であった．

答え候補から最適な解を探索する手法は，Shazeer et al.[1] により研究されている．Shazeer et al. のアルゴリズムは，それぞれのカギの答え候補リストに対し，単語の正解数を最大にするように候補の重みを繰り返し処理によって変更する．また一方で，カギの答え候補リストの単語の組み合わせをすべて調べ，制約を満たす解をすべて列挙しておく．そして，答え候補のスコアをもとにそれぞれの解のスコアを計算し，最適な解を選ぶ．

ところで，Shazeer et al. の方法では，答え候補として得られたすべての語の組み合わせの中から制約条件を満たす組み合わせをすべて見つけ，それらの組み合わせのうち最適なものを選ぶ，という段階がある．そのため，計算量が膨大になるうえ，制約条件を満たす組み合わせが存在しない場合には最終的な出力がまったく得られない．

一方，本研究では，制約を満たすすべての組み合わせを列挙しておくことはしないので，各カギの答え候補が多い場合にも

連絡先: 神保 一樹, 東京工業大学 工学部情報工学科, 神奈川県横浜市緑区長津田町 4259 R2-728 東京工業大学奥村研究室, 045-924-5295, 045-924-5295, jinbo@lr.pi.titech.ac.jp

*1 9 × 9 の格子状のマス目を，規則にしたがって 1–9 の数字で埋めるパズル．ナンバープレイスとも呼ばれる．

*2 クロスワードパズルのカギの答えとした単語のうち，正解した単語の割合

計算量があまり膨大にならない．また，完全に制約条件を満たす解が得られなくても，部分的に制約条件を満たす解を出力として得られる．

3. クロスワードパズルを解く手法

3.1 クロスワードパズルの自動解答タスクについて

クロスワードパズルの自動解答タスクは，大きく 2 つの段階に分けることができる．

前半の段階は，連想問題を解くタスクである．この段階では，カギに含まれる情報をもとに言語資源や Web を探索し，マス目に入るべき単語の複数の候補を，スコアとともに得る．この候補は，次の段階の最適化問題の入力となる．カギから答えを得るという点では，質問応答のタスクにも近い．

後半の段階は，制約を満たす最適な解を探す最適化問題である．ここでいう制約とは，単語と単語が交わるマスには同じ文字が入る，ということである．単語と単語が交わるマスに異なる文字が入ることはできないので，そのような場合は，交わる 2 単語のうち少なくとも一方は不正解である．この段階では，前半の段階の結果をもとに，各カギの単語の候補を組み合わせて，最適な解を求める．

3.2 カギから答え候補を得る手法

本節では，クロスワードパズルのカギから，スコア付きで答え候補を得る手法について述べる．ここで述べる手法のうち，辞書を利用するものは佐藤 [2] の手法である．一方，Web を利用するものは，佐藤のものにはない新たな手法である．

カギから答え候補を生成するために，複数の手法を併用した．具体的には，まずカギから内容語を取り出した．これらをキーワードと呼ぶ．次に，得られたキーワード一つ一つに対して各手法を適用した．最後に，それぞれの手法から得られた答え候補をすべて併合して，1 つのリストにまとめた．

以下では，これらの手法の詳細について述べる．

3.2.1 辞書を利用する手法

既存の辞書のデータを利用した手法は以下のものがある．

国語辞典　順引き：キーワードで岩波国語辞典を引き，その定義文中に出現する単語を候補として返す．スコアは，定義文中にその候補が出現した回数とする．

国語辞典　逆引き：岩波国語辞典の各見出し語の定義文を探索し，キーワードを含むものを見つける．その定義文に対応する見出し語を候補として返す．スコアは，その定義文中にキーワードが出現した回数とする．

国語辞典　穴埋めパターン探索：穴埋めパターン探索は，「馬の耳に○○○○」のように，空欄を含むカギに対して適用する手法である．この手法に限り，内容語だけを取り出すことはせず，カギの表現をパターンとみて，岩波国語辞典の見出しと本文からこのパターンにマッチする部分を取り出し，空欄に当てはまる部分を答えの候補とする．スコアは，辞書全体でマッチした回数とする．

類語辞典　反対語検索：角川類語新辞典の反対語情報を利用し，キーワードの反対語を候補として返す．スコアは，見つかった反対語情報の個数とする．なお，角川類語新辞典の反対語情報は必ずしも対称ではない．たとえば，「陰」をひくと反対語として「陽」が掲載されているが，「陽」という見出し語はない．これより，キーワードが「陰」のとき「陽」につくスコアは 1 とする．また，「陰気」をひくと反対語として「陽気」が掲載され，「陽気」をひくと反対語として「陰気」が掲載されている．これより，キーワードが「陰気」のとき「陽気」につくスコアは 2 とする．

分類語彙表：分類語彙表の中で，キーワードと同じ分類番号の単語を候補として返す．分類語彙表では，意味の似た単語に同じ分類番号が割り振られている．また，同じ分類番号の単語は，いくつかの段落に分けられ，同じ段落内ではさらに小段落に分けられている．そこで，スコアは，同一小段落の単語は 3，同一段落の異なる小段落の単語は 2，同じ分類番号で異なる段落の単語は 1 とする．

3.2.2 Web 検索を利用

本研究では，前節で説明した辞書を利用する方法の他に，Web を利用する手法を提案する．

辞書を利用する方法では，質の高い言語資源を利用して答えを得られるというメリットはあるものの，辞書に掲載されていない事柄が問われた場合に解答することができないというデメリットもある．辞書のほかに，Web 上のいろいろな情報をあわせて利用することで，より幅広い内容の問題に対しても解答可能になることが期待できる．

Web を利用する手法として，本研究では以下のものを実装して用いた．

キーワードで検索：カギごとに，キーワードをすべて並べて，Web 検索[*3] のクエリとする．結果の上位 50 件のスニペットに含まれる単語を候補として返す．含まれる単語を取り出すための形態素解析には ChaSen[*4] を用いた．スコアは，50 件のスニペット中に候補の単語が出現した回数とする．

穴埋めパターン探索：前節で説明した，国語辞典を利用した穴埋めパターン探索を，Web 検索を利用して行う．このときの Web 検索のクエリは，カギの文字列のうち空欄以外の部分とする．前の方法と同じ検索サービスを利用し，結果の上位 50 件のスニペットから，穴埋めパターンにマッチする部分を取り出す．そして，空欄に対応する部分を答えの候補とする．スコアは，マッチした回数とする．

3.2.3 答え候補の併合

ここまでに述べた各手法によって答え候補のリストを得たら，それらをすべて併合して，最終的に 1 つの答え候補リストにする．複数の手法によって同じ候補が得られた場合は，その候補のスコアは各手法で付けられたスコアの和とする．

これにより，名詞や形容詞など，多くの品詞の単語が答え候補として得られる．しかし，クロスワードパズルの答えとして使われる単語はほとんどが名詞であることから，候補語は名詞に限定し，それ以外の品詞の候補はこの時点で除外する．

3.3 答え候補から最適解を求める手法

答え候補は，一般に 1 つのカギに対して複数得られる．そして，それらのうち最もスコアの高い候補を選んで組み合わせたとき，制約を満たした解が得られるとは限らない．したがって，何らかの方法により，答え候補から解となる組み合わせを求める必要がある．

答え候補から解を求める方法として，本研究では Potts Model を変形した拡張 Potts Model を適用する方法を用いた．

Potts Model はもともと，統計力学において，結晶格子上のスピンの挙動を表現するためのモデルである．グラフを扱う問題においては，このモデルはグラフ上の各ノードの「状態」をモデル化するのに用いられている．

Shazeer et al.[1] の方法を使えば，クロスワードパズルの要素とグラフの要素とを

ノード　各カギの答えを記入する欄

*3　本研究では「Yahoo!検索 Web サービス」から「ウェブ検索」を利用した．http://developer.yahoo.co.jp/search/

*4　http://chasen.naist.jp/

状態 各カギの答えとして選ばれた語

エッジ 複数のカギの答え記入欄の交わり

と対応づけることができる．

このように対応づけができれば，クロスワードパズルを解くタスクに Potts Model を適用できそうであるが，実際にはそのままでは適用できない．それは，以下の理由からである．

まず，Potts Model では，エッジで結ばれた 2 ノードの状態が等しい場合にエネルギー関数の値が小さくなり，確率値が大きくなる．しかし，クロスワードパズルにおいては，交差する 2 語はまったく同一になるわけではない．特に，交差する 2 カギの答えの文字数が異なるときは，2 語がまったく同一になることはありえない．クロスワードパズルにおいて，交差する 2 語に関する条件は，交点にあたるマスに入る文字が同じであることだけである．

また，Potts Model では，ノードの状態に関して用いることのできる事前情報は，あるノードが特定の 1 状態であるらしい，という情報のみである．しかし，クロスワードパズルを解くタスクにおいては，各カギに対して，複数の答え候補を含む重み付きリストが与えられるため，その情報は，Potts Model で利用可能なものとは異なる．

そこで，クロスワードパズルを解くタスクに当てはめることができるよう，Potts Model をもとに，拡張 Potts Model を定義する．

まず，エネルギー関数を，以下のように定義する：

$$H_{\mathrm{cross}}(\mathbf{c}) = -\beta \sum_{ij} M_{ij}(c_i, c_j) + \alpha \sum_{i} -S_i(c_i). \quad (1)$$

ここで，M_{ij} は 2 ノード v_i，v_j に入る語が制約条件を満たすか否かを表す関数である．v_i の状態が c_i，v_j の状態が c_j のとき，交差するマスの文字が等しいかまたは交差しないならば $M_{ij}(c_i, c_j) = 1$，そうでなければ $M_{ij}(c_i, c_j) = 0$ となる．また，S_i は，ノード v_i の答え候補リストに付けられたスコアに比例する値を返す．ただし，$\forall i, \sum_c S_i(c) = 1$ である．

このエネルギー関数をもとに式変形をすると，クロスワードパズルを解くタスクにおける状態ベクトルの確率 $P_{\mathrm{cross}}(\mathbf{c})$ と自由エネルギー $F_{\mathrm{cross}}(\mathbf{c})$ は，それぞれ

$$P_{\mathrm{cross}}(\mathbf{c}) = \frac{\exp(-H_{\mathrm{cross}}(\mathbf{c}))}{Z}, \quad (2)$$

$$\begin{aligned} F_{\mathrm{cross}}(\mathbf{c}) &= \sum_{\mathbf{c}} P_{\mathrm{cross}}(\mathbf{c}) H_{\mathrm{cross}}(\mathbf{c}) \quad (3) \\ &\quad - \sum_{\mathbf{c}} -P_{\mathrm{cross}}(\mathbf{c}) \log P_{\mathrm{cross}}(\mathbf{c}) \\ &= -\alpha \sum_i \sum_{c_i} \rho_i(c_i) S_i(c_i) \\ &\quad -\beta \sum_{ij} \sum_{c_i, c_j} \rho_i(c_i) \rho_j(c_j) M_{ij}(c_i, c_j) \\ &\quad - \sum_i \sum_{c_i} -\rho_i(c_i) \log \rho_i(c_i) \quad (4) \end{aligned}$$

と表される．ここで，Z は正規化項である．また，$\rho_i(c_i)$ はノード v_i が状態 c_i をとる確率であり，$\forall i, \sum_{c_i} \rho_i(c_i) = 1$ である．

$\forall i, \sum_{c_i} \rho_i(c_i) = 1$ のもとで $F_{\mathrm{cross}}(\mathbf{c})$ を最小化すると，以下の等式が得られる：

$$\rho_i(c) = \frac{\exp(\alpha S_i(c) + \beta \sum_j \sum_{c_j} M_{ij}(c, c_j) \rho_j(c_j))}{\sum_n \exp(\alpha S_i(n) + \beta \sum_j \sum_{c_j} M_{ij}(n, c_j) \rho_j(c_j))}. \quad (5)$$

この等式を用いて ρ_i を繰り返し更新することにより，$F_{\mathrm{cross}}(\mathbf{c})$ を漸近的に最小化することができる．

日本語のクロスワードパズルの場合，語頭に「ン」「ー」の文字が入らない[*5]という日本語そのものの特徴を利用することもできる．ノード v_i の k 文字目とノード v_j の 1 文字目が交わるとき，k 文字目が「ン」または「ー」である語 c はノード v_i には入れられない．そこで，関数 $f_i(c)$ を新たに定義し，

$$f_i(c) = \begin{cases} 0 & \exists j, k, \text{『}(v_i\text{の } k \text{ 文字目}) \text{ と } (v_j\text{の 1 文字目}) \text{ が交わり，かつ } (c \text{ の } k \text{ 文字目}) \in \{ \text{ン}, \text{ー} \}\text{』} \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

とする．この $f_i(c)$ を用いれば，ρ_i の値を更新するための式は以下のようになる：

$$\rho_i(c) = \frac{f_i(c) \exp(\alpha S_i(c) + \beta \sum_j \sum_{c_j} M_{ij}(c, c_j) \rho_j(c_j))}{\sum_n f_i(n) \exp(\alpha S_i(n) + \beta \sum_j \sum_{c_j} M_{ij}(n, c_j) \rho_j(c_j))}. \quad (7)$$

実験では，この式を用いて，エネルギー関数の $H_{\mathrm{cross}}(\mathbf{c})$ の値の変化が十分小さくなるまで ρ_i の値を繰り返し更新した．値が収束した後は，それぞれのカギについて $\rho_i(c_i)$ の値が最大となった語 c_i を選び出し，これを最終的な解として出力する．このモデルを，本論文では **拡張 Potts Model** と呼ぶ．

拡張 Potts Model を用いた計算では，ρ の初期値が問題となる．本研究では，語のスコアに比例する前述の値 S_i を用いて ρ_i の初期値を $\rho_i(c_i) = S_i(c_i)$ とする方法と，$\sum_{c_i} \rho_i(c_i) = 1$ を満たすように各 $\rho_i(c_i)$ $(0 \leq \rho_i(c_i) \leq 1)$ の初期値をランダムに決定する方法の 2 つを比較した．

Potts Model に対しては，アニーリングと呼ばれる処理ができる．アニーリングは，繰り返し処理により一度 ρ の値を収束させたあと，逆温度定数 β を $\Delta\beta > 0$ だけ大きくして再び繰り返し処理を行うことである．複数回のアニーリングをするときは，ρ の値が収束するたびに β を $\Delta\beta$ だけ大きくする．拡張 Potts Model では，β を大きくすると，制約条件に重みがおかれ，結果として制約条件を満たす候補の ρ の値が大きくなりやすい．しかし，各単語のスコアがあまり加味されなくなる．一方，α を大きくすると，各単語のスコアに重みがおかれ，結果として各カギのスコアの高い単語の ρ の値が大きくなりやすい．しかし，交差部分の制約条件が無視されがちになる．そこで，まず小さい β で各単語のスコアを重視した結果を得てから，その結果を新たな初期値として，β を大きくして再び実行する．これがアニーリングである．これにより，まず各単語のスコアを加味したのち，交差部分の制約条件も重視して実行することになり，両方の値を考慮に入れたよりよい結果が得られると考えられる．

4. 評価実験・結果と考察

本研究では，提案手法の有効性を確認するため，いくつかの実験を行った．

評価には，先行研究 [1][2] と同様，盤面に入った単語のうち正解しているものの割合（単語正解率）を用いた．

*5 小さい「ッ」「ャ」なども語頭には入らないが，クロスワードパズルでは，これらは大きい「ツ」「ヤ」と同一視されるため，小さい仮名であることを制約条件として使うことはできない．

4.1 実験データ

実験に用いるデータとして，縦 7 マス×横 7 マスの大きさのクロスワードパズルを 15 問，http://cross.matrix.jp/[*6] より収集した．この 15 問はすべて，同一の作者 MAS による作品である．以降の実験は，この 15 問に対して行う．

なお，1 問のクロスワードパズルに含まれるカギの個数は，最大 22 個，最小 18 個，平均 20.5 個である．

4.2 実験

3.2 節で得られた答え候補を用いて，最適解の探索を行った．ここでは，特にことわりのない限り，Web 検索を含めたすべての手法を利用して得た答え候補を利用した．

4.2.1 パラメータを変えながら実験

拡張 Potts Model のパラメータをいろいろに変えながら実験を行った．α と β をそれぞれ 0–100000 の範囲で変化させて，アニーリングをしないときの正解率をみた．単語正解率がもっとも良くなったときの結果は，初期値がスコアのとき 14.9% ($\alpha = 1000$, $\beta = 2$)，初期値がランダムのとき 13.0% ($\alpha = 500$, $\beta = 5$ と，$\alpha = 1000$, $\beta = 20$) であった．$\alpha = 1000$ で固定して β を変化させたときの単語正解率の変化を，表 1 に示す．

表 1: $\alpha = 1000$ で β を変化させたときの単語正解率 (%)

	β								
初期値	0	1	2	5	10	20	30	40	50
スコア	12.3	13.0	**14.9**	14.0	11.7	13.3	9.1	8.8	10.1
ランダム	11.9	11.4	11.4	12.3	11.7	**13.0**	11.4	7.8	9.4

初期値がスコアのときとランダムのときとで正解率を比較すると，パラメータの値によっては初期値がランダムのほうが正解率が高い場合もあるが，全体的には初期値がスコアの方が正解率が高くなっていることが分かった．

次に，Web を利用せずに取り出した答え候補に対して同じ実験を行い，結果を比較した．Web を利用しない場合は，Web を利用した場合に比べ，著しく正解率が下がった．

4.2.2 アニーリング

前述の実験で結果のよかった $\alpha = 1000$ の場合について，アニーリングを行った場合と行わない場合の比較をした．初期値がスコアの場合とランダムの場合のそれぞれについて，1 回あたりの β の増分 $\Delta\beta$ を 0.01–5 の範囲で変化させて，$\beta = 0$ から 2 まで，および 10 までアニーリングを行い，正解率の変化を調べた．結果は，表 2 のようになった．

表 2: アニーリングを行ったときの単語正解率 (%)

		$\Delta\beta$							アニーリング
β	初期値	0.01	0.05	0.1	0.5	1	2	5	なし
2	スコア	14.0	14.0	15.3	14.0	14.0	14.0	—	14.9
2	ランダム	10.7	12.0	11.7	11.4	12.7	10.1	—	11.4
10	スコア	13.6	13.0	13.3	14.0	14.3	13.6	14.3	11.7
10	ランダム	11.4	11.4	12.7	13.0	12.3	10.7	12.7	11.7

アニーリングをすると，アニーリングをしない場合に比べて単語正解率が向上する場合が多かった．ただし，アニーリングをする場合も，パラメータによって単語正解率が上下するので，パラメータの決定方法が重要である．

*6 MAS のページ（http://www2u.biglobe.ne.jp/~MAS/）内のコンテンツ

4.2.3 先行研究の手法との比較

Shazeel et al. の手法 [1] と本研究の手法との比較実験を行った．この実験を行ったマシン環境は，CPU は 2.66GHz × 2 のデュアルコア，メモリは 2GB である．当初は正解率の比較も行うつもりであったが，Shazeel et al. の手法で処理をすると 1 問の処理がまる 3 日かかっても終わらなかったため，実行時間の比較にとどめた．

Shazeer et al. の手法では 1 問の処理に 3 日以上かかった．一方で，提案手法を用いて処理をした場合は，1 問の処理（アニーリング 10 回）にかかるユーザ時間は平均 15.47 秒であった．この結果から，拡張 Potts Model は，答え候補の数が多い場合の実行時間の短縮に大きな効果があることがわかる．また，1 カギあたりの候補数が多いと，実行時間が長くなる傾向がみられた．

5. 今後の課題

今回の実験では，拡張 Potts Model を利用することにより，比較的高速に最適化問題を解くことができた．しかし，単語正解率は最高でも 15%前後にとどまった．これは，連想問題を解く手法をあまり吟味しなかったために答え候補の正確性が乏しく，それにより最適化問題の解に影響を与えたと考えられる．

そこで，今後は，連想問題を解く精度を高めていくことが課題となる．それには，質問応答の分野で研究されている技術がいろいろと応用できそうであると考えている．

また，拡張 Potts Model にはいくつかのパラメータがあり，その値をどのように決定すればいいかという問題もある．今回は，正解データを利用して，正解率が最大になるパラメータを決定した．今後は，正解が不明な場合に最適なパラメータを決定する方法についても考えていきたい．

参考文献

[1] Noam M. Shazeer, Michael L. Littman and Greg A. Keim. Solving Crossword Puzzles as Probabilistic Constraint Satisfaction. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pp. 156–162, 1999.

[2] 佐藤理史. 日本語クロスワードパズルを解く. 情報処理学会自然言語処理研究会, NL-147-11, pp.69–76, 2002.

[3] Greg A. Keim, Noam Shazeer, Michael L. Littman, Sushant Agarwal, Catherine M. Cheves, Joseph Fitzgerald, Jason Grosland, Fan Jiang, Shannon Pollard and Karl Weinmeister. PROVERB: The Probabilistic Cruciverbalist. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pp. 710–717, 1999.