Unbiased Evaluation of Handwritten Mathematical Expression Recognition*

Francisco Álvaro, Joan-Andreu Sánchez, José-Miguel Benedí
Instituto Tecnológico de Informática. Universitat Politècnica de València.
{*falvaro,jandreu,jbenedi*}@*dsic.upv.es*

Abstract

Several approaches have been proposed to tackle the problem of mathematical expression recognition, and automatic methods for performance evaluation are required. Mathematical expressions are usually encoded as a LaTeX string or a tree (MathML) for evaluation purpose, but these formats do not enforce uniqueness. Consequently, given that there can be several representations syntactically different but semantically equivalent, the automatic performance evaluation of mathematical expressions can be biased. Given a mathematical expression recognition tree and its ground-truth tree, the error is usually computed by comparing them. In this paper we propose to obtain a new tree, equivalent to the ground-truth tree, according to the model representation criteria. Then, we can compute an error by comparing the recognized tree with the obtained by using the model, both with the same bias. Several experiments were carried out in order to evaluate this approach and results showed that representation criteria had a significative effect in the evaluation results.

1. Introduction

Mathematical notation is an essential part of information in science documents and many other fields. Handling mathematical expressions and introducing them into computers usually requires special notation like LaTeX. However, lately there has been a great increasing of pen-based interfaces and tactile devices that allow users to provide handwritten data as input. This is a more natural way of introducing mathematical notation. Introducing the mathematical expressions in this natural way requires, in turn, developing systems that are able to recognize them. Recognition systems for mathematical expressions depend on the application [4]: online recognition systems for handwritten mathematical expressions, and offline recognition systems for handwritten or printed mathematical expressions. In online recognition, the system input is usually a set of strokes that have geometric and temporal information. The system can take profit of temporal information in online recognition that is not present in offline recognition. This paper will be focused in online recognition of handwritten mathematical expressions.

The recognition of handwritten mathematical expressions can be divided into two major steps [4]: symbol recognition and structural analysis. Symbol recognition involves segmentation of the input strokes into mathematical characters and symbol classification of these hypotheses. Structural analysis deals with finding out the structure of the expression according to the symbols arrangement.

Symbol recognition can be evaluated by reporting both symbol segmentation rate and symbol recognition rate. However, evaluation and comparison of the structural analysis in mathematical expression recognition has hitherto been difficult because: first, most of the proposals used private data, and second, there has been a lack of standard performance evaluation measures [11]. Lately, some public large corpora have been developed [14] and also new metrics have been proposed [17, 20, 1]. Recent open competitions on mathematical expression recognition [15] have brought attention to the problem of automatic evaluation of the structural analysis.

One of the main problems in mathematical expression evaluation is that the same expression can be properly encoded in different ways. Given the recognized tree (t_R) of a mathematical expression and its ground-truth tree (t_G), the error is usually computed by comparing t_R with t_G. Hence, evaluation metrics based on this methodology must deal with the problem that t_G could not be unique. A good solution would be to define a canonical representation for every mathematical expres-

*Work supported by the EC (FEDER/FSE) and the Spanish MEC/MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018), the MITTRAL (TIN2009-14633-C03-01) project, the FPU (AP2009-4363) grant, and by the Generalitat Valenciana under grant Prometeo/2009/014.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.287

sion. However, this is not an easy task because many different people work developing recognition systems and corpora following different representation criteria, and there could even be differences due to semantic interpretations.

In this paper, given a certain mathematical expression and its ground-truth t_G, we propose to obtain a new tree representation (t_P) according to the model. This tree could be syntactically different but semantically equivalent to t_G. Thereby, we were able to compute an error by comparing the recognition results t_R with t_P, both with the same bias, instead of the usual comparison between t_R and t_G that can be biased in a different way.

The remainder of the paper is organized as follows. A review of related works is given in Section 2. The online handwritten mathematical expressions recognition system is described in Section 3. Experiments and results are presented in Section 4. Finally, conclusions and future work are discussed in Section 5.

2. Background

Recognition of online handwritten mathematical expressions can be divided into two major steps [4]: symbol recognition and structural analysis. Symbol recognition comprises, in turn, segmentation of the input strokes and subsequent symbol classification of these hypotheses. Structural analysis copes with the problem of building the structure of the expression according to the spatial arrangement of the mathematical symbols.

The segmentation problem has been tackled by computing connected components [2], applying the projection profile cutting method [16] or more sophisticated techniques [13]. These segmentation hypotheses are then classified. Several methods have been proposed to solve this problem, such as HMM [9], Elastic Matching [3] or Support Vector Machines [10]. Furthermore, some of these proposals combine online and offline information to perform hybrid classification and improving recognition results [10].

Several approaches have been studied to deal with the structural analysis problem, mainly based on trees [21], graphs [7] or grammars [6]. Definite clause grammars [5] or graph grammars [12] have been used to solve this problem. Chou [6] proposed to use Stochastic Context-Free Grammars (SCFG) in order to recognize printed mathematical expressions, and several studies have also used this formalism [19, 2]. In this paper, we developed an online handwritten mathematical expression parser based on SCFG and a statistical formulation based on the Cocke-Younger-Kasami (CYK) algorithm.

The automatic evaluation of mathematical expression recognition is not an easy task [11] and this fact has made difficult the definition of widely accepted evaluation measures. Several research studies have introduced different recognition techniques for mathematical expression and most of the times each study has used a different method for evaluation [11]. Thereby, it is difficult to properly compare different approaches to the problem of mathematical expression recognition.

In the past, several metrics have been proposed to report performance of mathematical expression recognition systems. There are metrics such as symbol segmentation rate and symbol recognition rate [18] that can be computed if the ground-truth is available. However, these values only take into account the evaluation of a specific part of the recognition problem. Regarding the evaluation of the structural analysis of mathematical expressions, it is often reported the expression recognition rate [5, 21]. However, it does not provide any information about errors; it only determines whether or not an expression is perfectly recognized, and sometimes it is manually calculated.

Given that the previous methods only report partial errors, several global measures have also been presented. Chan and Yeung [5] proposed an integrated performance measure, which was a simple combination of symbol recognition and operator recognition rates. Garain and Chaudhuri [8] presented a global performance index that combined symbol and structural errors according to the complexity of the mathematical expression. Sain *et al.* [17] presented EMERS, a tree matching-based performance evaluation measure that computes an edit distance between two trees. Recently, Zanibbi *et al.* [20] have developed a set of performance metrics based on a bipartite graph representation that seems to be canonical, but it is not detailed in the article and no experimentation is reported. Finally, we developed a measure based on image representation to compare two expressions because this representation solves many of the ambiguity problems [1].

Despite these global measures are being developed, nowadays there is no standard metric reported by most of mathematical expression recognition studies. Researchers usually report symbol segmentation rate, symbol recognition rate and expression recognition rate. However, either the expression recognition rate is calculated manually or the existence of equivalent representations remains a problem. In past competition CROHME, this three measures were reported [15]. Every measure was computed automatically, and the expression recognition rate was calculated by comparing the labeled tree t_G and the recognized tree t_R represented in MathML format.

Provided that tree representation do not enforce uniqueness, given a recognized tree t_R and its ground-truth information t_G the error is usually computed by comparing them. However, this comparison is incorrectly biased because t_R and t_G can be structurally different but semantically equivalent. We propose to perform a constrained parsing of a mathematical expression in order to obtain a new tree t_P equivalent to t_G but following the model representation criteria. Then, for the evaluation of a mathematical expression recognition experiment we had 3 sets of trees (Figure 1), and instead of computing the error between t_R and t_G, in this approach we compare t_R with t_P.

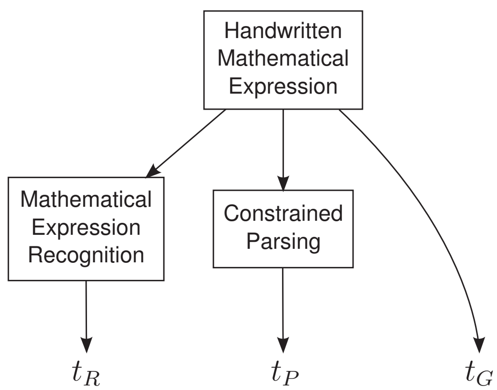

Figure 1. Different trees are obtained from a mathematical expression: recognized tree (t_R), constrained parsed tree (t_P) and ground-truth tree (t_G).

3. Handwritten mathematical expression recognition

The handwritten mathematical expression recognition system used for experimentation is detailed in the following section. It is based on two-dimensional stochastic context-free grammars. Thereafter, we describe how we used the annotated information of a certain mathematical expression in order to guide this parser to recognize it.

3.1. Parser description

SCFG are a powerful formalism that has been widely used for string patterns. Mathematical expressions are two-dimensional structures, hence, SCFG must be extended to properly model these type of information. In this work, we used a two-dimensional extension of SCFG that has been used in mathematical expression recognition [6, 19, 2]. This extension basically adds a spatial relation constraint to every production of the grammar. Thus, representing the grammar in Chomsky normal form, it results in two type of rules: binary rules with spatial relation constraint ($A \xrightarrow{r} B\ C$) and terminal rules ($A \to m$). There are 5 possible spatial relations r: horizontal, below, subscript, superscript and inside.

We developed an online handwritten mathematical expression recognition system based on parsing two-dimensional SCFG through a CYK algorithm for this type of grammars. This algorithm performs two steps. First, the parsing table is initialized by using a handwritten mathematical symbol classifier and the terminal productions of the grammar ($A \to m$). Then, the algorithm builds new subproblems of increasing size according to the binary rules of the grammar ($A \xrightarrow{r} B\ C$).

Formally, let $\mathcal{G}$ be a two-dimensional SCFG, and let $\mathcal{C}$ be the set of segmentation hypotheses representing the input expression such that $\mathcal{C} = \{\ c_i^z \,|\, i : 1 \ldots C\ \}$ where c_i^z represents the segmentation hypothesis located at a certain spatial region z. This set is provided to the parser after a previous step. In this case, the segmentation hypotheses are obtained as the connected components of the expression. This algorithm is essentially a dynamic programming method, which is based on the construction of a parsing table $\mathcal{L}$. Each element in $\mathcal{L}$ is defined according to the following concepts. Value $e^z[A]$ is the probability that A is solution of the mathematical expression contained in the region z. Analogously, $e_l^z[A]$ is the probability that A is solution of the mathematical subexpression contained in the region z, considering l segmentation hypotheses. Finally, $\mathcal{L}_l = \{e_l^z[A]\}$ is a stochastic parse structure where each element $e_l^z[A]$ is composed from l segmentation hypotheses.

This process is divided into two steps. First, the initialization begins building the set $\mathcal{L}_1$ from the set of segmentation hypotheses defined in $\mathcal{C}$:

$$\mathcal{L}_1 \;=\; \mathcal{L}_1 \,\cup\, \{\ e_1^z[A]\ \} \qquad\qquad \forall i : 1 \ldots C$$

such that:

$$e_1^z[A] = \max_m \{p(A \to m)\ q_m(c^z)\}$$

where m is a particular mathematical symbol, for which there is a specific recognizer. Value $q_m(c^z)$ is the probability provided by a HMM symbol classifier of this class m for segmentation hypothesis c^z. Next, the parsing process continues calculating new subproblems of increasing size. Formally, the general case is computed as

$$\mathcal{L}_l \;=\; \mathcal{L}_l \,\cup\, \{\ e_l^z[A]\ \} \qquad\qquad \forall\, l : 2 \ldots C$$

with:

$$e_l^z[A] = \max_{B,C} \max_r \max_{k:1\ldots l-1} \max_{\substack{e_k^{z_B}[B] \\ e_{l-k}^{z_C}[C]}}$$

$$(p(A \xrightarrow{r} B\, C)\ \ e_k^{z_B}[B]\ e_{l-k}^{z_C}[C]\ p_r(z_B, z_C))$$

where a new subproblem $e_l^z[A]$ is created from two subproblems of minor size $e_k^{z_B}[B]$ and $e_{l-k}^{z_C}[C]$ taking into account both the syntactic constraints, defined by $p(A \xrightarrow{r} B\, C)$, and the following constraints: first, r represents the spatial relation, and second, $p_r(z_B, z_C)$ is the probability that both regions were arranged according to the spatial relation r. We modeled the probability $p_r(z_B, z_C)$ with a SVM classifier by representing the relation between regions z_B and z_C as a numeric vector composed by several geometric features extracted from relative positions of these regions.

3.2. Constrained parsing

Using the previously detailed handwritten mathematical expression parser and an annotated expression, we wanted to constrain the parsing process to recognize it perfectly. There were two parts of the recognition process to take into account. First, symbol segmentation and recognition is completely solved provided that given an expression, its annotated information identifies each symbol (class m) and the strokes that compose them (c^z). Thus, using this information the parsing table was initialized such that for each symbol in $\mathcal{C}$ the probability of belonging to its labeled class was set to one ($q_m(c^z) = 1.0$).

On the other hand, the structural analysis is not straightforward due to that several trees can represent the same expression. Given a handwritten mathematical expression and its ground-truth information, we had available a set of spatial relations between subsets of symbols of the expression. The MathBrush corpus [14] has these relations explicitly labeled, but the parser could split the expression in a different way producing a different tree. In order to guide the parser to recognize a certain expression, we provided it with the list of annotated spatial relations. Hence, when two hypotheses B and C were combined during the parsing process, the probability that they were arranged according to a spatial relation r is set to one ($p_r(z_B, z_C) = 1.0$) if that relation is labeled in the reference. As it is likely that the model builds a different tree of a mathematical expression compared to the reference tree, we provided the parser with more spatial relations than the annotated ones. These extra relations between subsets of symbols were obtained thanks to transitive properties of spatial relations.

4. Experiments

In this section we used the MathBrush corpus [14] which contains $4,654$ annotated online handwritten mathematical expressions written by 20 different writers. The number of mathematical symbols is 26K and they are distributed in 100 different classes. In order to compare the expression trees we used EMERS [17], a well-defined tree edit distance evaluation measure. This metric is not a normalized distance, but it calculates the set of edit operations to transform a tree into another such that if both trees are identical EMERS is equal to zero.

Several experiments were performed to validate our proposal by using trees obtained through different sources (Figure 1). First, a constrained parsing of all the expressions in the database was performed to dispose the ground-truth represented following the model criteria (t_P). Then, a handwritten mathematical expression recognition experiment was carried out and our evaluation approach was analyzed by using these results (t_R) and both references (t_P and t_G).

4.1. Ground-truth constrained parsing

For every labeled expression of the MathBrush corpus, we obtained a new reference tree t_P as explained in Section 3.2 and we wanted to know if these recognized representations were equivalent to t_G. For that reason, we compared the recognized LaTeX expression with its corresponding annotated LaTeX by using IMEGE [1]. We performed the comparison (t_P, t_G) by using IMEGE, an image-based measure error such that if it was equal to zero we were sure that the expression was perfectly recognized. Otherwise, we manually checked the remaining constrained parsed output. As a result, we obtained that 99.05% of the corpus was properly parsed.

The 0.95% remaining expressions were discarded and the constrained parsing errors had different causes. Some expressions were not perfectly recognized because the model could not account for it due to the spatial search space restriction during the parsing process. For example, in Figure 2a, the subexpressions "$-j$" and "∞]" are not parsed properly because when performing an horizontal concatenation of two regions we require that the second one is after the first in left-to-right order. Other errors were caused because the model produced an expression structure such that the spatial relation information provided was not enough to guide the parser, hence, some decisions relied on the model which failed. For example, in Figure 2b, the relation between S and i was not provided to the parser and the probability of the spatial relation classifier is higher for subscript relation. Finally, a few expressions had annotation errors. We also noticed that the constrained parsing is useful to check if a model is able to account for a given set of expressions.

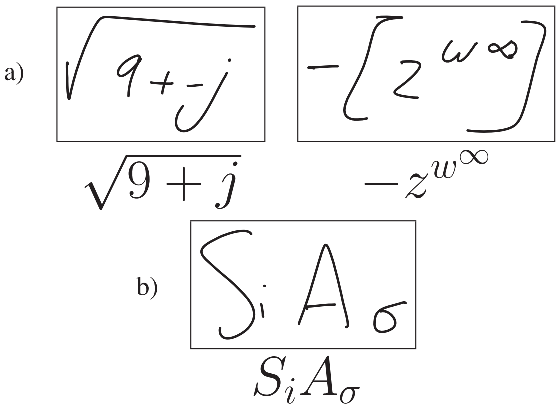

Figure 2. Example of constrained parsing errors.

After discarding the wrong parsed expressions, we compared the trees generated by the model with the labeled trees of the corpus (t_P, t_G) using EMERS. Thus, we observed that 62% of the expression trees were different. Most of the differences in the representation were due to horizontal relations, because an horizontal concatenation of symbols can be split in several ways (Figure 3a). The model produced binary trees because the SCFG used was in Chomsky Normal Form, whereas the reference had tree nodes with more than two children. Moreover, other differences were caused by codification criteria (Figure 3b). Therefore, results showed that the structural analysis of the expression in the constrained parsing process is not straightforward and that tree representation criteria will affect the evaluation results.

4.2. Mathematical expression evaluation

We used the handwritten mathematical expression recognition system described in Section 3.1 to perform a simple experiment over the MathBrush corpus. We randomly split the database into a training set and a test set. The training set was composed of 70% of the samples (3227 expressions) and the remaining 30% were selected as test set (1383 expressions). The expressions in the training set were used in order to train the HMM symbol recognition classifier and the SVM spatial relations classifier.

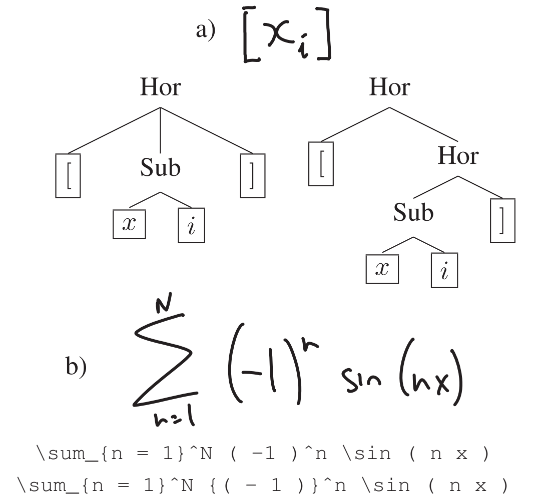

Figure 3. Examples of differences in tree representation between t_P and t_G.

For each recognized expression, we measured the results by using EMERS regarding both the ground-truth tree (t_R, t_G) and the tree obtained through constrained parsing (t_R, t_P). Table 1 shows the average results of the mathematical expression experiment, where reported measures are: symbol segmentation rate (SYM_{seg}), symbol recognition rate for well segmented symbols (SYM_{rec}), expression recognition rate (EXP_{rec}) and EMERS for both scenarios. Expressions perfectly recognized were those whose EMERS value was equal to zero.

Table 1. Mathematical expression recognition results regarding two different references.

	SYM_{seg}	SYM_{rec}	EXP_{rec}	EMERS
(t_R, t_G)	84.86%	79.82%	17.71%	4.06
(t_R, t_P)			25.31%	3.43

The segmentation recognition rate was about 85% and symbol recognition rate was about 80%. Symbol recognition could be improved, but we performed a simple writer independent experiment and this work is more interested in the structural analysis evaluation. Results show that the EMERS average value regarding the labeled tree information is higher than the value obtained regarding the tree obtained using the model, as it was expected. The representation criteria had a significative effect in the expression recognition rate, where this value differed in 7.60% depending on the reference. Thus, it can be seen that the comparison between t_R and t_G is biased in a very different way.

5. Conclusions

In this work, we proposed a method to obtain an unbiased evaluation of a recognition experiment over an online handwritten mathematical expressions database [14]. First, we used a mathematical expression recognition parser based on two-dimensional SCFG to obtain a new reference (t_P) of the corpus through constrained parsing. Thereby, we obtained the representation produced by the model for each expression in the database. Afterwards, we performed a mathematical expression recognition experiment and the results were evaluated comparing the recognized tree (t_R) versus two different ground-truth: the corpus labeled tree (t_G) and the tree obtained from the model (t_P).

Finally, we conclude that representation criteria has a significative effect in the evaluation of mathematical expression recognition if trees are compared directly, and that the comparison between t_R and t_G is biased. Several experiments were performed to measure this influence and we saw that 62% of the expression trees of the database were different than those obtained by the model (t_P,t_G). Furthermore, in a recognition experiment, the rate of expressions perfectly recognized varied by 7.60% depending on the reference. Constrained parsing is also useful to check if a model is able to account for a certain set of expressions.

For future work we are interested in applications of constrained parsing for providing automatic ground-truth information. Also it would be interesting the application of constrained parsing to printed mathematical expressions in order to be able to do post-edition or interaction of recognition results.

References

[1] F. Álvaro, J. Sánchez, and J. Benedí. IMEGE: Image-based Mathematical Expression Global Error. *DSIC-PRHLT Technical Report, UPV*, 2011.

[2] F. Álvaro, J. Sánchez, and J. Benedí. Recognition of printed mathematical expressions using two-dimensional stochastic context-free grammars. In *Proceedings of ICDAR*, 2011.

[3] K. Chan and D. Yeung. Elastic structural matching for online handwritten alphanumeric character recognition. In *ICPR*, volume 2, pages 1508 –1511 vol.2, aug 1998.

[4] K. Chan and D. Yeung. Mathematical expression recognition: a survey. *IJDAR*, 3:3–15, 2000.

[5] K. Chan and D. Yeung. Error detection, error correction and performance evaluation in on-line mathematical expression recognition. *Pattern Recognition*, 34(8):1671 – 1684, 2001.

[6] P. A. Chou. Recognition of equations using a two-dimensional stochastic context-free grammar. In W. A. Pearlman, editor, *Visual Communications and Image Processing IV*, volume 1199 of *SPIE Proceedings Series*, pages 852–863, 1989.

[7] Y. Eto and M. Suzuki. Mathematical formula recognition using virtual link network. In *Proceedings of ICDAR*, pages 762–767, Washington, DC, USA, 2001.

[8] U. Garain and B. Chaudhuri. A corpus for OCR research on mathematical expressions. *IJDAR*, 7:241–259, 2005.

[9] L. Hu and R. Zanibbi. HMM-based recognition of online handwritten mathematical symbols using segmental K-means initialization and a modified pen-up/down feature. In *Proceedings of ICDAR*, September 2011.

[10] B. Keshari and S. Watt. Hybrid mathematical symbol recognition using support vector machines. In *Proceedings of ICDAR*, volume 2, pages 859 –863, sept. 2007.

[11] A. Lapointe and D. Blostein. Issues in performance evaluation: A case study of math recognition. In *ICDAR*, pages 1355–1359, Washington DC, USA, 2009.

[12] S. Lavirotte and L. Pottier. Mathematical formula recognition using graph grammar. In *In Proc. of the SPIE*, volume 3305, pages 44–52, 1998.

[13] Z. Luo, Y. Shi, and F. Soong. Symbol graph based discriminative training and rescoring for improved math symbol recognition. In *Proceedings of ICASSP*, pages 1953–1956, 2008.

[14] S. MacLean, G. Labahn, E. Lank, M. Marzouk, and D. Tausky. Grammar-based techniques for creating ground-truthed sketch corpora. *IJDAR*, 14:65–74, 2011.

[15] H. Mouchère, C. Viard-Gaudin, U. Garain, D. Kim, and J. Kim. CROHME2011: Competition on recognition of online handwritten mathematical expressions. In *Proceedings of ICDAR*, September 2011.

[16] M. Okamoto and B. Miao. Recognition of mathematical expressions by using the layout structures of symbols. *Proceedings of ICDAR*, pages 242–250, 1991.

[17] K. Sain, A. Dasgupta, and U. Garain. EMERS: a tree matching–based performance evaluation of mathematical expression recognition systems. *IJDAR*, pages 1–11, 2010.

[18] Y. Takiguchi, M. Okada, and Y. Miyake. A fundamental study of output translation from layout recognition and semantic understanding system for mathematical formulae. *ICDAR*, 0:745–749, 2005.

[19] R. Yamamoto, S. Sako, T. Nishimoto, and S. Sagayama. On-line recognition of handwritten mathematical expressions based on stroke-based stochastic context-free grammar. *IEIC Technical Report*, 2006.

[20] R. Zanibbi, P. Amit, H. Mouchère, C. Viard-Gaudin, and D. Blostein. Stroke-Based Performance Metrics for Handwritten Mathematical Expressions. In *ICDAR*, Beijing, China, Sept. 2011.

[21] R. Zanibbi, D. Blostein, and J. Cordy. Recognizing mathematical expressions using tree transformation. *IEEE Trans. on PAMI*, 24(11):1–13, 2002.