# Computational Algorithms and the Complexity of Link Polynomials

Bruce EWING and Kenneth C. MILLETT

## Introduction

In this paper we describe the fundamental combinatorial nature of one of the knot and link invariants and an effective computational algorithm to determine it in a significant range of mathematically interesting cases. A brief introduction to the fundamental concepts of classical knot theory, to issues concerning the representation of knots and links and to the recursive approach to the calculation of the polynomial invariants, is provided in the first section. In addition, various theoretical issues concerning the complexity of these calculations are discussed. The basic structures of several recursive algorithms are described and discussed in the second section of the paper, including the recent algorithm of Robert J. Jenkins, jr. The third section of the paper is devoted to an evaluation of the observed growth in complexity associated to the algorithms and their implementations. This evaluation is of fundamental importance since it is known that the calculation of these polynomial invariants is NP-hard, [Ja1, 2] and [Th2]. The fourth, and final, section gives a summary of our conclusions and some remarks on outstanding questions.

The invariants discussed in this paper are Laurent polynomials associated to oriented knots and links in Euclidian space. The first was developed by J.W. Alexander [A] in about 1926 and further elaborated upon by Conway [Con] in the 1960's, while the second was discovered in the spring of 1984 by V.F.R. Jones [Jon1 & 2]. The first generalization, the oriented polynomial, was discovered almost immediately following the announcement of Jones discovery by four sets of authors [FYHLMO]. Subsequently, another distinctly different generalization was discovered by Brandt, Lickorish, and Millett [BLM] and Ho [Ho]. This, in turn, was immediately extended by Kauffman [Kau3] to give the definition of the semioriented polynomial. Although we focus this paper on the oriented polynomial, the effective calculation of both the oriented and semioriented polynomials is one of the goals of our project. A fuller account of the oriented polynomial and a description of the case of the semioriented polynomial is found in an article describing our earlier efforts, [EM]. The Jenkins algorithm is described in [Je].

One of the principal vehicles to define these invariants is a recursive formula relating the polynomial associated to one knot or link to those associated to simpler knots or links. The recursive approach has exponential growth in the number of cases that have to be calculated. Significant effort is devoted to the search for alternative mathematical theories which would provide a comprehensive conceptual approach to the invariants. Despite the recent theoretical advances via Yang-Baxter statistical mechanical state models derived from quantum mechanics, [Jon3, Tu1], or the functional integral formulation from quantum field theory described by Ed Witten in 1988, [W1 & 2], these recursive methods remain the most effective calculational method for general presentations of knots and links. In addition, Witten described a family of numerical invariants of closed 3-manifolds inspiring Reshetikhin and Turaev, [RT] to define another version of Witten's 3-manifold invariants via the theory of quantum groups. An alternative approach has been developed by Lickorish [L2,3] using a summation of variants of the Jones polynomials associated to an explicit family of links constructed from a surgery description of the 3-manifold These links are cables on the subsets of the components of the link on which the surgery is done to construct the 3-manifold Because the cabling construction rapidly increases the number of crossings in a minimal presentation of the link, the ability to do extremely rapid calulations for rather large numbers of crossings is required in order to do numerical studies of the invariants associated to the 3-manifolds. Some general formulae are known for the cases associated to some roots of unity in terms of classical invariants, [KM], but the general structure of these numerical invariants remains quite mysterious

## 1 A brief introduction to combinatorial knot theory

The purpose of this section is to provide a brief survey of the fundamental aspects of the study of knots and links in three-space and the "oriented" spatial invariants associated to them. Thus, we are concerned with the spatial analysis of disjoint simple closed curves, whereby one allows these "strings" to move about from one position to another so long as no portion passes through another Knots or links which can be moved one to the other, are said to be equivalent or " to be the same knot or link " Fuller accounts of the theory of knots and links are found in the recent books of Burde and Zieschang, [BZ], and Kauffman [Kau2] as well as those of Crowell and Fox, [CrF], and Rolfsen, [Rol], and the expository articles of Kauffman [Kau3], Lickorish [L1], Lickorish and Millett [LM2], Fox, [F], or Thistlethwaite [Th1].

One fundamental goal of this theory is to develop mathematical methods which allow one to distinguish between different knots or links, as opposed to different presentations of the same knot or link Thus one is lead to try to associate numbers or algebraic structures to presentations of knots or links which are to be unchanged when a spatial movement occurs taking one configuration to another equivalent one A traditional way in which to begin is to consider regular projections of the knots or links onto a fixed plane and to study the allowable changes of the presentation which are sufficient to describe all changes from one presentation of a knot or link to another presentation of the same knot or link. The projections are broken so as to indicate places where one strand crosses over another strand Examples of two distinct knots are shown in Figure 1 1, and non-generic behavior, such as shown in Figure 1.2, is not permitted. The first case shows a point of tangency, not a crossing, of the segments

We have developed a notation for the presentation designed for the purposes of our computational algorithms. We assume that all components are oriented.

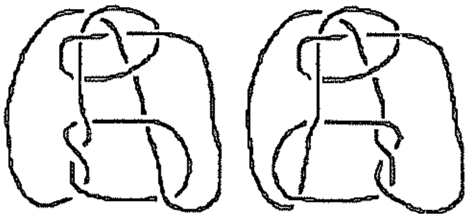


Figure 1.1. The Conway and Kinoshito-Terasaka Knots

An example (
the crossings is assig
the directions occurr
on the over crossing

and the rema
a clockwise directio
following the conver

data associat(
of the sign of the cr
directions of the cro:
1 4.

Among the p
nition of the existe
deletion of crossing
ious possible crossi
of arbitrary knots ar
the Reidemeister m(
Planar motio
connection relations

ım groups An alter-
on of variants of the
m a surgery descrip-
nponents of the link
:abling construction
ıe link, the ability to
quired in order to do
general formulae are
:al invariants, [KM],
terious

damental aspects of
nvariants associated
mple closed curves,
nother so long as no
to the other, are said
theory of knots and
Kauffman [Kau2] as
xpository articles of
=], or Thistlethwaite

ıethods which allow
ent presentations of
gebraic structures to
al movement occurs
which to begin is to
› study the allowable
om one presentation
ojections are broken
mples of two distinct
in Figure 1 2, is not
e segments
the purposes of our

ıots

An example of our coding scheme is shown in Figure 1 3 Here one sees that each of the crossings is assigned a specific number and at each of the crossings letters are assigned to the directions occurring at the crossing according to the rule : the outward pointing direction on the over crossing segment is assigned the letter "a"

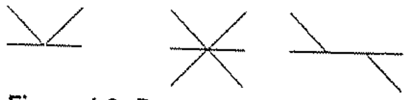


**Figure 1.2. Degenerate projections**

and the remaining directions are assigned the letters "b", "c", and "d", proceeding in a clockwise direction from "a" Furthermore, each crossing is assigned a sign "+" or "-", following the convention shown in Figure 1 6 The

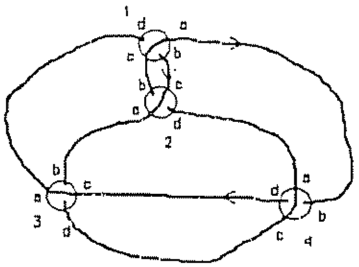


**Figure 1.3. The coding of the figure 8 Knot**

data associated to a given presentation consists of the ordered list, for each crossing, of the sign of the crossing followed by the connection points of the "a"", "b", "c", and "d" directions of the crossing Thus, data for the example shown in Figure 1,3, are given in Table 1 4

| | | | | |
|---|---|---|---|---|
| - | 4b | 2c | 2b | 3a |
| - | 3b | 1c | 1b | 4a |
| + | 1d | 2a | 4d | 4c |
| + | 2d | 1a | 3d | 3c |

**Table 1.4. Figure 8 knot data**

Among the principal benefits of our presentation are that it allows very rapid recognition of the existence of Reidemeister moves of types I and II, it provides for the rapid deletion of crossings and it supports the rapid evaluation of the potential benefits of the various possible crossing changes These modifications allow one to simplify the presentations of arbitrary knots and links as described below This ease of recognition explains the use of the Reidemeister moves in our modification of the standard ascender algorithm

Planar motions or modifications of the pictures which do not change the crossing and connection relationships of the strands will take a presentation of a knot or link to a presentation of an equivalent knot or link because these can be understood as shadows of permissible spatial movements It can be shown that sequences of only three additional elementary local alterations, called Reidemeister moves of types I, II and III, of the presentations are sufficient to characterize the equivalence of any two presentations of the same knot or link

These Reidemeister moves are shown in Figure 1 5

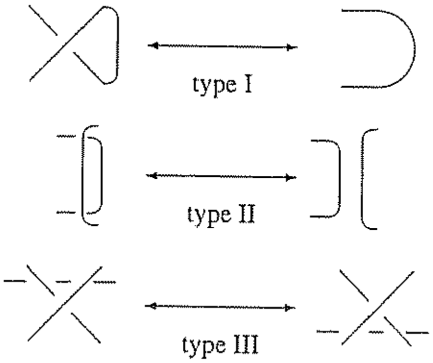


**Figure 1.5. Reidemeister Moves**

As mentioned in the introduction, the polynomials can be calculated by means of a simple recursive process Indeed, this process can be employed to define the polynomials The fundamental fact that makes this work is that one may change a link to the unlink with the same number of components by simply reversing some of the crossings in a given presentation. The polynomials to be calculated are Laurent polynomials with two variables and integer coefficients (Laurent polynomials allow both negative and positive powers of the variables). The following theorem describes the fundamental properties of the oriented polynomial discovered following the first announcement of the Jones polynomial which it generalizes.

**Theorem 1.1.** *The oriented polynomial, $P_L(\ell, m)$, of an oriented knot or link, $L$, is the unique invariant which satisfies the following fundamental formulae* ·

*(i) if $U$ denotes the standard unknotted circle in the plane, then $P_U(\ell, m) = 1$.*

*(ii) if $L_+, L_-$, and $L_0$ are planar pictures of oriented links in each of which we have identified a small circular region of the picture containing either a single crossing or, in the last case, no crossing at all, and such that outside these small circular regions where they behave as shown in Figure 1 6, the planar pictures are* **exactly** *the same, then* ·

$$\ell P_{L_+}(\ell, m) + \ell^{-1} P_{L_-}(\ell, m) + m P_{L_0}(\ell, m) = 0$$

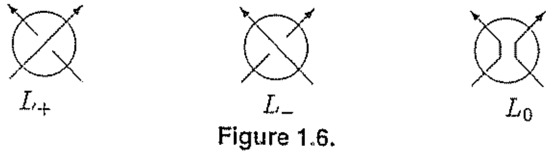


**Figure 1.6.**

To demonstrate the
recursion formula in the de
the general method that we
may calculate the oriented
configurations shown in F
case of a general presenta
order to significantly simpl
especially algorithmically
an unknot. Indeed, $L_-$ is
each used once.

These moves may b
of the crossings to be chan
of the subsidiary knots o
possible, in general, to sel
reductions The overall go
in the greatest possible ov

By definition, the p
the polynomial associated
which associates the desi
method that we employ fo
further reduced by means
the recursion

procedure can be
link. Here, this is accomp

We discover that
the trivial link with two c
by using the Reidemeiste
determine the polynomia
shown in Figure 1 9

ɔod as shadows of permissible
ree additional elementary local
the presentations are sufficient
same knot or link.

)

calculated by means of a
ɔ define the polynomials.
link to the unlink with the
gs in a given presentation.
variables and integer co-
ers of the variables). The
d polynomial discovered
eneralizes.

*knot or link, $L$, is the*

*ı $P_U(\ell, m) = 1$.*
*each of which we have*
*gle crossing or, in the*
*ar regions where they*
*e, then*

To demonstrate the complexity of the calculations that one encounters in the use of the recursion formula in the determination of this polynomial invariant and to make more concrete the general method that we shall describe in the next section, we shall briefly review how one may calculate the oriented polynomial associated to the right-handed trefoil. Consider the configurations shown in Figure 1.7. Here we find that the trefoil is the case $L_+$. In the case of a general presentation of a knot or link, the selection of a crossing to be changed in order to significantly simplify the resulting presentation is an important and difficult question, especially algorithmically. In the case of this trefoil knot, any crossing change will result in an unknot. Indeed, $L_-$ is a trivial knot. This is seen by use of Reidemeister moves I and II, each used once

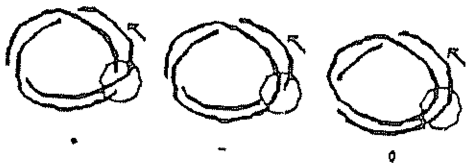


**Figure 1.7.**

These moves may be exploited, in more general cases, to indicate the optimal selection of the crossings to be changed so as to insure that the number of crossings in the presentations of the subsidiary knots or links can be significantly reduced by their application. It is not possible, in general, to select crossing changes which provide such immediate and dramatic reductions. The overall goal of the recursive method is to select crossing changes which result in the greatest possible overall reduction of complexity.

By definition, the polynomial associated to $L_-$ is equal to 1. Next we must compute the polynomial associated to $L_0$. For simple presentations it is possible to develop a table which associates the desired polynomial with the presentation. Indeed, this is exactly the method that we employ for presentations having six or fewer crossings and which can not be further reduced by means of the Reidemeister moves. In the present situation we show that the recursion procedure can be used to reduce all configurations to those involving a trivial knot or link. Here, this is accomplished by using the set of configurations shown in Figure 1.8.

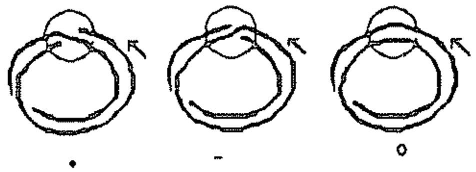


**Figure 1.8.**

We discover that $L_0$ is a trivial knot, again by using the Reidemeister move I, $L_-$ is the trivial link with two components which it is convenient to denote by $U^2$. This is shown by using the Reidemeister move II. Thus, in order to complete the calculation, we need only determine the polynomial associated to $U^2$. For this we employ the set of configurations shown in Figure 1.9

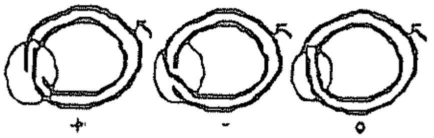


Figure 1.9.

Here we find that $L_0 = U^2$ and that the other configurations are trivial knots As a consequence of the fundamental formula (ii) of the Theorem, we have :

$$\ell P_U(\ell,m) + \ell^{-1}P_U(\ell,m) + mP_{U^2}(\ell,m) = 0$$

so that

$$P_{U^2}(\ell,m) = -(\ell + \ell^{-1})m^{-1}$$

To compute the polynomial associated to the $L_+$ depicted in Figure 1 8 we have :

$$\ell P_{L_+}(\ell,m) + \ell^{-1}P_{U^2}(\ell,m) + mP_U(\ell,m) = 0$$

so that

$$P_{L_+}(\ell,m) = \ell^{-2}(\ell + \ell^{-1})m^{-1} - \ell^{-1}m$$

Finally to complete the calculation of the polynomial associated to the right-handed trefoil, $T$, we return to the set of configurations in Figure 1 7, from which we obtain, using the fact that the $P_{L_+}(\ell,m)$ just computed is the $P_{L_0}(\ell,m)$ in that situation :

$$\ell P_T(\ell,m) + \ell^{-1}P_U(\ell,m) + m[\ell^{-2}(\ell + \ell^{-1})m^{-1} - \ell^{-1}m] = 0$$

so that

$$\begin{aligned} P_T(\ell,m) &= -\ell^{-2} - \ell^{-1}m[\ell^{-2}(\ell + \ell^{-1})m^{-1} - \ell^{-1}m] \\ &= -2\ell^{-2} - \ell^{-4} + \ell^{-2}m^2 \end{aligned}$$

This sample calculation contains many of the elementary aspects that are employed to achieve a recursive calculation in the general case The first principle is that one can change a crossing in a diagram in order to relate a given knot or link to a simpler one at the cost of having to consider an additional knot or link The altered knot or link and the auxiliary one are, in some measurable way, simpler than the initial case and the calculations of their polynomials can be made completely independently so long as one remembers how to reassemble the results to calculate the polynomial of the desired knot or link Many of the auxiliary knots and links which arise in the course of the calculation are actually equivalent To the extent that one can easily recognize these equivalent cases and combine their calculations so as to reduce the number of redundant calculations, one can significantly improve the complexity of the calculation of the invariant This is one of the underlying principles of the recursive method that we have developed

One can see that t
employed to calculate the
initially could have signi
is a fundamental concern
calculational algorithms a
The problem of d
its generalizations is, in g
between them and variou
Alexander polynomial or
one-variable determinant
polynomial time as a fun
The oriented and
Jaeger, [Ja1], and Thistle
Although it was c
of both the oriented and s
Vertigan and Welsh, [Ja2
special class of alternati
eight special values of tl
terms of classical (polync
Jones polynomial at any
$P$-hard problems is a cla
class

## 2 The algorithms

The development
section is based upon se
proposed to study, i e pr
such as closed braid forr
barrier to their calculatio
speed of the fundamental
configurations appearing
structure should provide
appearing in the fundam
for fast evaluation of the
benefit is related to the
of these moves These g
effective method of calc
research efforts

Briefly, at each st
three) of associated subs
problems is measurably
more readily than for the
measure this reduction c

ɜ trivial knots As a

we have :

:ight-handed trefoil,
btain, using the fact

ɿ] = 0

ɿ]

hat are employed to
ɩat one can change a
at the cost of having
ɩuxiliary one are, in
ɔf their polynomials
ɿ to reassemble the
the auxiliary knots
ɜlent. To the extent
:alculations so as to
ove the complexity
ɔles of the recursive

One can see that there may be many alternative computational strategies that may be employed to calculate the invariant. Specifically, the choice of which crossing is to be changed initially could have significant impact upon the complexity of the resulting calculation. This is a fundamental concern to be addressed in the next section where we describe the variaous calculational algorithms and their underlying philosophical approaches.

The problem of deciding whether or not the calculation of the Jones polynomial or its generalizations is, in general, a feasible calculation was answered by making connections between them and various versions of the chromatic polynomials of graphs. On one hand, the Alexander polynomial or the associated Conway polynomial, being computable as a "small" one-variable determinant, has been determined to be 'feasible' in that it can be calculated in polynomial time as a function of the number of crossings in its presentation.

The oriented and the semioriented polynomials have been shown to be NP-hard by Jaeger, [Ja1], and Thistlethwaite, [Th2].

Although it was considered possible that the Jones polynomial, being a specialization of both the oriented and semioriented polynomials, could be computationally simpler Jaeger, Vertigan and Welsh, [Ja2], have shown that this is not the case by proving that, even for the special class of alternating presentations, the calculation is $P$-hard. Moreover, except for eight special values of the Jones polynomial which are either trivial or can be described in terms of classical (polynomial time computable) knot and link invariants, the evaluation of the Jones polynomial at any specific value of the variable is a $P$-hard problem [Ve]. The class of $P$-hard problems is a class of enumeration problems playing a role analogous to the NP-hard class.

## 2 The algorithms

The development of the load balanced algorithm which we shall describe later in this section is based upon several fundamental premises : First, within the range of problems we proposed to study, i.e. presentations with no more than 150 crossings and no special structure such as closed braid form, the space required to store their descriptions would not provide a barrier to their calculation. Second, the data structure should be designed so as to optimize the speed of the fundamental elements of the algorithm such as the recognition of those elementary configurations appearing within the presentation which we wish to exploit. Third, the data structure should provide for fast changes such as crossing switches and removals of the sort appearing in the fundamental recursion relation : Fourth, the data structure should provide for fast evaluation of the benefit associated with a variety of crossing change strategies. The benefit is related to the use of elementary Reidemeister moves and suitable generalizations of these moves. These generalizations were developed and studied in an effort to produce an effective method of calculation of the polynomials of links which arise in the course of other research efforts.

Briefly, at each stage of the process one generates a small collection (often only two or three) of associated subsidiary problems of the same type. The complexity of these subsidiary problems is measurably simpler and their individual solutions can therefore be calculated more readily than for the original problem, [LM1 & 2]. There are two basic ways in which we measure this reduction of complexity. The first is simply the total number of crossings in the presentation while the second involves the identification of a sequence of crossing changes which would result in a significant change in the knot or link. A fundamental quality typical of these subsidiary problems is that, in addition to being structurally simpler than the original problem, they can be solved completely independently of one another. The solutions found for these subproblems can then be combined to yield the desired result. In most cases this reassembly is easy, with a few simple algebraic operations yielding the correct answer. The actual implementation of the process requires a prior estimate of the size of the final answer and traditional methods to insure that the calculation is accomplished efficiently.

The critical issue is that the recursive application of this reduction process results in a regular, exponential growth, computational tree with nodes representing individual instances of the calculations, edges representing the reduction process, and the root node representing the solution to the original problem. Several recursive algorithms have been developed. We will describe the general approaches and the specific implementations which we have developed in this section of the paper. In the subsequent section we shall describe the special attributes of the load balanced algorithm.

First it is useful to consider one of the algorithms derived from the article of J. H. Conway and which played a fundamental role in one of the theoretical approaches to the definition of the polynomial invariants. This method is described implicitly in Conway [Con] and is exploited in the papers of Hoste [Hos] and Lickorish and Millett [LM1]. The basic concept is that of the "standard ascending position". Actually Hoste utilizes a standard descending position, but this is an equivalent method. Consider a specific projection of a given link into a plane, to be thought of as the floor of the room, and imagine taking a length of rope (or several lengths, if one is looking at the projection of a link of several components) which is exactly long enough to cover the projection. Beginning at any point in the projection which is neither an over or under crossing, and proceeding in the direction given by the orientation, cover the projection with the rope. When returning to the starting point, join the two ends together. Proceed with any other component in exactly the same fashion. This physical process introduces under and over crossings which may differ from those of the given presentation and thereby defines the standard ascending position. One can easily observe that the standard ascending position always represents a trivial knot or link. The specific representation is, however, dependent upon the choice of starting point or points and their order, if there are several components.

The standard ascender algorithm is described as follows : Proceed in the direction determined by the orientation from any starting point, chosen as in the definition of the standard ascending position, change crossings as necessary to achieve the standard ascending position. With each crossing change, the new knots or links defined by removing crossings are added to a list of cases to be resolved (to each case is attached the information needed to calculate its specific contribution to the final polynomial) and one proceeds to change crossings until the standard ascending position is reached. At this point one updates a polynomial table and continues with the last unresolved case by means of the same procedure until there are no further cases. The recursion formulae have thereby provided the means to calculate the polynomial by reducing a knot or link to the associated standard ascending position and computing other terms, each of which involves knots or links having fewer crossings in their presentations. The number of cases on the list need never be more than half the total number of

cossings. The polynomia
only on the number of co
determined and stored b
independent of the choic
the theoretical developm
The modified stan
and II and, in addition, th
be described later. They a
under consideration and
algorithm exploits two o
searches the given presen
I or II or their generaliz
moves as soon as they are
identified and the present
Since these moves reduc
appearing in the applicat
of the calculation. When
in a given case, the stand
The ancillary cases are a
again.

The fundamental
one subsidary example i
consequence one expects
the effect of the reductio
running time of the impl

Before giving a
describe the underlying
in complexity of the typi
first step is to search fo
crutial step, is the evalu
of the computational tre
across the two ancillary
by selecting crossing ch
complexity. A simple e
reduction of complexity
where $n$ is the size of t
oriented and semioriente
the observed running ti
type. This appears to be
when pruning is attempt

Suppose that a re
giving an auxiliary calc
of type $(n, 1)$ Suppos
and gives an auxiliary c

e of crossing changes
mental quality typical
npler than the original
. The solutions found
ılt. In most cases this
e correct answer. The
ize of the final answer
efficiently
on process results in a
ıg individual instances
t node representing the
en developed. We will
ch we have developed
e the special attributes

m the article of J. H.
cal approaches to the
citly in Conway [Con]
ett [LM1]. The basic
ite utilizes a standard
ecific projection of a
and imagine taking a
n of a link of several
beginning at any point
eding in the direction
turning to the starting
nt in exactly the same
which may differ from
ling position. One can
a trivial knot or link
starting point or points

oceed in the direction
the definition of the
the standard ascending
by removing crossings
information needed to
eds to change crossings
tes a polynomial table
ocedure until there are
means to calculate the
scending position and
ewer crossings in their
half the total number of

cossings. The polynomials of the standard ascender which can arise in the calculation depend only on the number of components and the number of crossings in the presentation. They are determined and stored beforehand. Showing that this process leads to a single polynomial, independent of the choices made in its recursive definition is one of the fundamental tasks of the theoretical development of the polynomial invariant theory.

The modified standard ascender algorithm employs the Reidemeister moves of types I and II and, in addition, their generalizations developed as part of the load balanced algorithm to be described later. They are employed to reduce the complexity of the knot or link presentation under consideration and, thereby, reduce the overall complexity of the calculation. This algorithm exploits two of the fundamental combinatorial aspects of the problem. First, one searches the given presentation of a knot or link for instances of Reidemeister moves of types I or II or their generalizations and immediately reduces the presentation according to these moves as soon as they are discovered. With our data structure, instances of these moves can be identified and the presentation modified to represent the result of the move extremely rapidly. Since these moves reduce the number of crossings in the presentations of all subsequent links appearing in the applications of the recursion, there are significant savings in the complexity of the calculation. When there are no further moves of types I and II or their generalizations in a given case, the standard ascender algorithm is applied to select a crossing to be changed. The ancillary cases are added to the list and the process of seeking type I or II moves is begun again.

The fundamental issue with respect to standard ascender algorithms is the fact that one subsidary example is created with each crossing change invoked in the algorithm. As a consequence one expects roughly exponential growth in complexity proportional to $2^n$. Thus the effect of the reduction in the number of crossings has significant potential for reducing the running time of the implementation if this can be accomplished rapidly.

Before giving a technical description of the load balanced algorithm we shall first describe the underlying reasons why this approach is expected to yield important reductions in complexity of the typical calculation. As in the modified standard ascender algorithm, the first step is to search for applications of the first two Reidemeister moves. The next, and crutial step, is the evaluation of each crossing change for potential simplication and pruning of the computational tree through the reduction of the number of crossings. Load balancing across the two ancillary cases is used to minimize the exponential growth of the calculation by selecting crossing changes which would yield the greatest *weighted* reduction in the total complexity. A simple example illustrates the nature of this evaluation and the potential for reduction of complexity : Suppose that the complexity of a calculation is proportional to $\rho\alpha^n$, where $n$ is the size of the data. We believe this to be a reasonable assumption since, in the oriented and semioriented knot polynomial calculation algorithms and their implementations, the observed running time and number of evaluations are both complexity measures of this type. This appears to be a consequence of the exponential growth of the calculation tree, even when pruning is attempted.

Suppose that a reduction is possible which reduces the resulting complexity to $\rho$ while giving an auxiliary calculation of complexity $\rho\alpha^{n-1}$. We shall say that this is a reduction of type $(n, 1)$. Suppose that another choice of reduction provides a complexity of $\rho\alpha^{n-2}$ and gives an auxiliary case of the same complexity, thus having complexity type (2.2). If, for example, $\alpha > 2$, the second choice would be preferable to the first since

$$\rho\alpha^0 + \rho\alpha^{n-1} = \rho(\alpha + \alpha^{2-n})\,\alpha^{n-2} > 2\,\rho\alpha^{n-2} = \rho\alpha^{n-2} + \rho\alpha^{n-2}$$

The systematic evaluation and choice of crossing changes based upon this strategy is what we mean by load balancing in this context.

We next outline the structure of the load balancing algorithm for the case of the oriented polynomial, assuming that the necessary initialization and loading of the oriented knot or lonk has been completed :

First, search for and immediately remove any Reidemeister type I and type II configurations until no further reductions of these types remain.

Next, search for an evaluate potential complexity type of triples : A triple is a configuration whose projection has the motif indicated in Figure 2.1. Null triples of types 1,2, and 3 are immediately reduced by means of a generalized Reidemeister II move and the reduction process is immediately restarted by seeking instances of Reidemeister type I and II moves, above. When none of the remaining possibilities can be reduced in this elementary fashion, their complexity types are evaluated for their potential benefit.

Next search for and evaluate potential complexity type of gammas : A gamma is a configuration whose basic pattern is shown in Figure 2.2. Note that in cases 1 and 3, a crossing can be removed by a simple twist.

This gives a bigon configuration shown in the same figure. These twists are performed immediately and the algorithm is restarted with the search fo Reidemeister moves of types I and II shown in Figure 1.5.

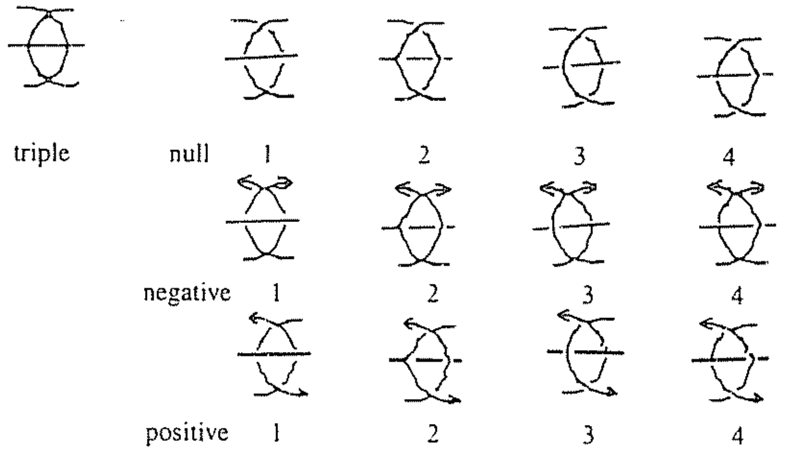
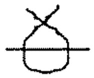


**Figure 2.1**

Next search for and identify the potential complexity type of circuits :

Here one is searching for large portions of the knot or link which can be simplified

by changing relatively fe
provide crossings change
A circuit is either a
in Figure 2.3, or a genera



This generalizatio
its ends, which may cros
of this is also shown in l
crossing and continues, l
Either it returns to the ini
case one does not have a
segment having been ide

since

$^{-2} + \rho\alpha^{n-2}$.

n this strategy is what we

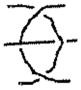
or the case of the oriented
the oriented knot or lonk

ype I and type II config-

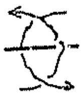
es : A triple is a configu-
riples of types 1,2, and 3
move and the reduction
ster type I and II moves,
this elementary fashion,

ammas : A gamma is a
cases 1 and 3, a crossing

ese twists are performed
meister moves of types I


4

4


4

circuits :
which can be simplified

by changing relatively few crossings. Furthermore, circuits will always occur and thereby provide crossings changes of " last resort".

A circuit is either an entire component of the link which never crosses itself, as indicated in Figure 2.3, or a generalization of the gamma configuration described in Figure 2.2.

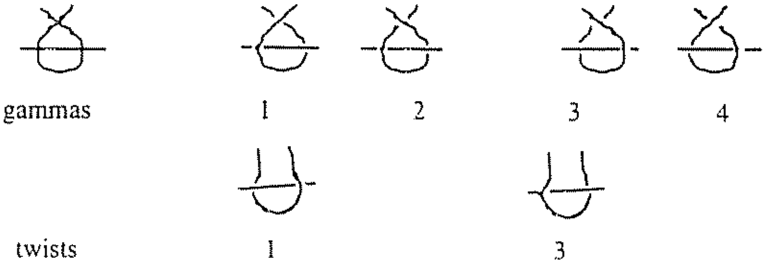


**Figure 2.2**

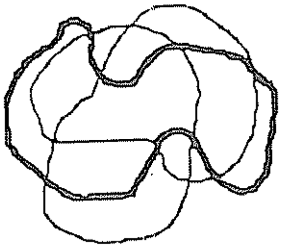
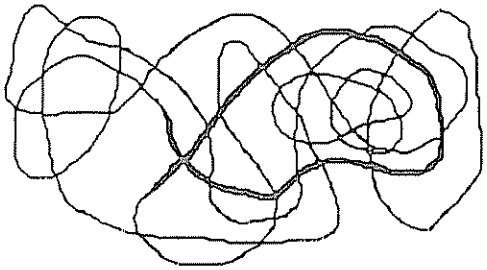


**Figure 2.3**

This generalization is defined as a segment which crosses itself precisely once and near its ends, which may cross the exterior of the segment in any arbitrary fashion. An example of this is also shown in Figure 2.4. The process begins by following a strand as it leaves a crossing and continues, listing the crossings encountered and their type, i.e. over or under. Either it returns to the initial crossing or encounters an earlier crossing on the list. In the later case one does not have a circuit beginning at the initial crossing in that direction (a smaller segment having been identified as a circuit)



type II circuit
**Figure 2.4**

so that the analysis goes back to the initial crossing and considers another direction, or the next crossing if none is available. Each of the circuits is evaluated to determine its potential contribution to the reduction processes. Of special importance are cases where half or more of crossings of the circuit occur with the same type, i.e. under or over, in a sequence as one travels around the circuit. Such occurrences allow us to do a global movement of the presentation and create a skinny circuit in the interior of which there are no further crossings. A potential skinny circuit is assigned a complexity based upon the number of potential immediate type II reductions following its creation. This situation provides for particularly effective crossing changes and associated reductions. An example of this type of modification is shown in Figure 2.5. In the first situation one has a sequence of undercrossings numbering half or more of the total while in the second a global transformation, another "generalized Reidemeister type II move", has been performed to achieve a skinny circuit configuration. These transformations have several essential properties. First, they never transform a skinny circuit that may already exist in the presentation into one which is no longer skinny and, second, although these transformations do not always reduce the number of crossings they do create opportunities for advantageous reduction. If all the crossings were of the same type, i.e. under or over, this allows for the removal of the entire circuit via a "generalized Reidemeister type II move". Any time a skinny circuit is created we immediately restart the analysis by searching again for instances of Reidemeister type I and II moves.

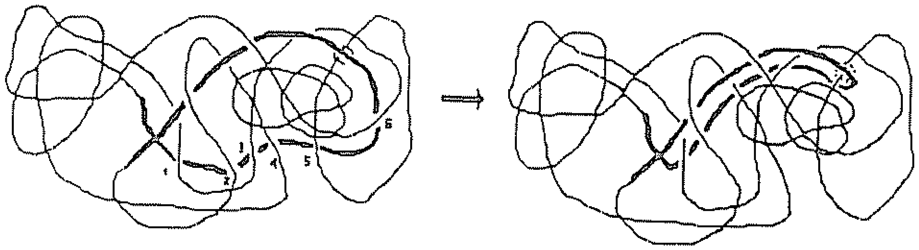


**Figure 2.5**

Next search for and evaluate the potential complexity type of bigons : A bigon is a configuration whose projection has the pattern indicated in Figure 2.6. Bigons occur frequently in knot and link presentations and are created as a byproduct of the creation of skinny circuits.

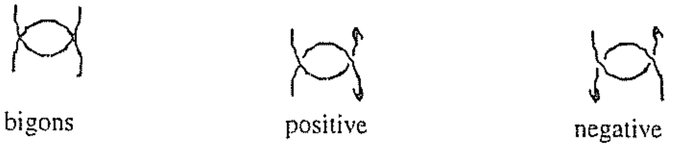


**Figure 2.6**

Next, if the knot or link presentation has five or fewer crossings the associated polynomial is read in a previously prepared table and is stored in another table which accumulates to the final polynomial. If the knot or link has six or more crossings, the crossing change having

the greatest potential adv
giving a modified presen
according to the number (
with the last presentation

At this point the c
to the user.

The final algorithı
developed in the master c
rithm for Computing the
Mellon University in June
growing region within th
jection is to be one of a s
a certain number, $k$, of 'i

This determines a
these polynomials can be
inhabitant of the region
are no interior compone
and outputs can be numl
strands cross, the strand
using Reidemeister move
configuration then it is al
one can achieve a simple
separated from the stranc
factor, i.e. $-(\ell^{-1} + \ell)$ı
configurations associatec
$k$ as small as possible in

The Jenkins algo
inhabitant is simple or s
necessary to change tha
crossing is changed the
must also be calculated b
the associated exterior c
expressions associated t
that one can thereby red
of the calculation is sub
region has engulfed all
strand. This terminal sit

The complexity c
discuss later, is estimat
the planar separator theo
planar graph on $n$ vertic
a vertex of $A$ meets a v
$C$ has no more that $2\sqrt{}$
that $2\sqrt{2}$ was reduced

the greatest potential advantage for reducing the level of complexity is performed thereby giving a modified presentation and one auxiliary presentation These are added to the list according to the number of crossings in the presentation The whole process is then restarted with the last presentation and continued until there are no knots or links remaining in the list.

At this point the calculation of the polynomial is completed and the result is reported to the user.

The final algorithm which we shall discuss in this paper, the Jenkins algorithm, was developed in the master of science dissertation, 'Knot Theory, Simple Weaves, and an Algorithm for Computing the Homfly Polynomial', submitted by Robert J. Jenkins Jr. to Carnegie Mellon University in June, 1989 The basic principle of the method is the creation of a steadily growing region within the presentation of the knot of link within which the form of the projection is to be one of a specific set of 'solved' possibilities The region will be a disk having a certain number, $k$, of 'inputs' (and hence $k$ 'outputs').

This determines a region for which the multilinearity of the Conway skein theory of these polynomials can be applied. For any configuration of boundary inputs and outputs, an inhabitant of the region or 'weave' in the Jenkins vocabulary is said to be simple if there are no interior components, no strand within the region has a self crossing, and the inputs and outputs can be numbered counterclockwise from 0 through $2k - 1$ so that, when two strands cross, the strand with the lowest numbered input is the overpass. In addition, if by using Reidemeister moves interior to the region, the configuration can be moved to a simple configuration then it is also called simple. By changing some set of crossings within a region, one can achieve a simple configuration Indeed, if there are interior components, they can be separated from the strands and absorbed into the coefficient by multiplying by the appropriate factor, i.e. $-(\ell^{-1} + \ell)m^{-1}$ in the case of the oriented polynomial. There are $k!$ simple configurations associated to the region. As a consequence, one endeavors to keep the size of $k$ as small as possible in the course of the calculation of the polynomial.

The Jenkins algorithm proceeds by creating an ever growing region within which the inhabitant is simple or solved Each time a new crossing is added to the region it may be necessary to change that crossing or others to insure that the result is simple. Each time a crossing is changed the recursion formula creates another configuration whose polynomial must also be calculated by the same method. By keeping track of the simple configurations and the associated exterior configurations one can identify duplications By adding the algebraic expressions associated to duplicate configurations, only one copy is retained To the extent that one can thereby reduce the number of independent cases to be calculated the complexity of the calculation is substantially diminished The calculation terminates when the solved region has engulfed all the crossings and the exterior region has been reduced to a single strand This terminal situation is, therefore, a projection of a trivial knot or link.

The complexity of an analog of the Jenkins algorithm, whose implementation we shall discuss later, is estimated by first estimating the size of the number of inputs by means of the planar separator theorem of Lipton and Tarjan, [LT], which states that the vertices of any planar graph on $n$ vertices can be partitioned into three sets A,B,C such that no edge meeting a vertex of $A$ meets a vertex of $B$, neither $A$ nor $B$ contains more than $2n/3$ vertices and $C$ has no more that $2\sqrt{2}\sqrt{n}$ vertices. $C$ is called a separator of the graph. Jenkins reports that $2\sqrt{2}$ was reduced to $\sqrt{6}$ by Djidjev Jenkins states that he recursively partitions the regions of the link diagram into separators and regions until every crossing is in a separator at some level of the recursion or is in a single vertex region. By the Lipton-Tarjan theorem, this requires $\ell n(n)/\ell n(3/2)$ as an upper bound on the depth of the separation tree, where $n$ is the number of crossings. One can achieve the computation by 'solving' the polynomials associated to the irreducible regions and proceeding to the next higher level by 'solving' the polynomials associated to the regions gotten by adding the pairs of adjacent regions and their separators. This one strategy to obtain a solved region engulfing all the crossings in a knot diagram. By 'solving' the polynomial, Jenkins means determining the expression of the tangle in the region in terms of a choice of skein generators associated to the region. At level $t \geq 1$ in the binary tree of separations, there are at most $\sqrt{6}\,(2/3)^{(t-1)/2}\sqrt{n}$ vertices in a separator between regions at that level. The number of edges connecting a pair of regions to their common separator at level $t$ is bounded by $2\sqrt{6}(2/3)^{(t-1)/2}\sqrt{n}$. From this one can deduce that the number of inputs in a region is bounded above by $2\sqrt{6}(1/(1-\sqrt{2/3}))(\sqrt{n})$ which is $O(\sqrt{n})$. A more precise bound is given by

$$2\sqrt{6}\sqrt{n}[1-[\sqrt{(2/3)}]^{\ell n(n)/\ell n(3/2)}]/[1-\sqrt{(2/3)}].$$

During the engulfing process one will join a strand at the boundary between two regions. This could require $2^{\sqrt{n}-2}$ operations in order to express the resulting configuration in terms of simple generating configurations, configurations of the region. A case by case analysis shows that each individual case is one of a small number of simple cases. Thus it is possible to estimate the theoretical complexity as $O((\sqrt{n})!\ell n(n)2^{\sqrt{n}-2})$. Jenkins employs a slightly different method to estimate the theoretical complexity and gets a slightly larger estimate : $O((\sqrt{n})!n2^{\sqrt{n}-2})$.

In the actual implementation, provided to us by Jenkins, a different algorithm is used to choose the crossings which occur in the engulfing. Jenkins 'greedy' method is to begin with a randomly selected crossing and, at each stage, to select a crossing contiguous to the solved region which has the most connections with it and, in case of a tie, that crossing which connects with largest number of crossings not in the solved region. For this approach, Jenkins was unable to give a theoretical estimate of the calculation complexity. In his manuscript, Jenkins states that tests seem to indicate that his 'greedy' algorithm runs in $O(n((\sqrt{n})!)2^{\sqrt{n}-1})$ time. If correct, this estimate would compare poorly with, for example, a modified standard ascender algorithm, cf. Ewing-Millet [EM]. In our paper we studied the relationship between implementations of the load balanced algorithm, a modifield standard ascender algorithm and, a standard a scender algorithm, and found that the load balanced algorithm was significantly faster than the others. In the next section we consider the results of similar tests with the implementation provided us by Jenkins. These tests seem to suggest that, in some range, his implementation is much faster than his assertion would indicate.

## 3 Evaluation and comparison of the Jenkins and load balanced algorithms

In this section we shall present data for the Jenkins algorithm and the load balanced algorithms applied to a variaty of test cases : all knots with up to 13 crossings, in the Thistlethwaite enumeration, as well as certain 25, 48 and 56 crossing test knots. The goal is to discover

the extent and origin of th
ins algorithms for the ori
and links that are produce
In addition we wish to cc
Jenkins algorithms.

The following tabl
algorithm requires 6.29 t
calculation of the average

avera
avera

Tabl

The following tabl
algorithm requires 5.39 t
calculation of the average

avera
avera

Tabl

some slight evide
to the load balanced impl

The real purpose
interesting examples. Ir
oriented polynomial of a

We next consider
knot. The advantage of J

The following dat
$13_6714$ in the Thistleth
is significantly less in th
under consideration.

ry crossing is in a separator
the Lipton-Tarjan theorem,
the separation tree, where $n$
·y 'solving' the polynomials
higher level by 'solving' the
of adjacent regions and their
g all the crossings in a knot
g the expression of the tangle
to the region. At level $t \geq 1$
$/^2\sqrt{n}$ vertices in a separator
ıg a pair of regions to their
ι. From this one can deduce
$1/(1-\sqrt{2/3}))(\sqrt{n})$ which

$\sqrt{(2/3)}]$.

undary between two regions.
ılting configuration in terms
ın. A case by case analysis
ɔle cases. Thus it is possible
. Jenkins employs a slightly
s a slightly larger estimate :

different algorithm is used to
:dy' method is to begin with
ing contiguous to the solved
that crossing which connects
this approach, Jenkins was
y. In his manuscript, Jenkins
uns in $O(n((\sqrt{n})!)2^{\sqrt{n}-1})$
xample, a modified standard
lied the relationship between
dard ascender algorithm and,
l algorithm was significantly
ılts of similar tests with the
gest that, in some range, his

## ; and load balanced

rithm and the load balanced
3 crossings, in the Thistleth-
:nots. The goal is to discover

the extent and origin of the reduction in complexity achieved by the load balanced and Jenkins algorithms for the oriented polynomial. We compare the number of recursions or knots and links that are produced by the two algorithms, as a function of the number of crossings. In addition we wish to compare speed, measured in average time, of the load balanced and Jenkins algorithms.

The following table gives the average data for the 2176 12 crossing knots. The Jenkins algorithm requires 6.29 times as many cases and takes 1.49 times as long to complete the calculation of the average 12 crossing knot.

| | load balanced | Jenkins | ratio |
|---|---|---|---|
| average recursions | 10.4163 | 65.4982 | 6.29 |
| average time | 0.03097 $s$ | 0.04617 | 1.49 |

**Table 3.1.** Oriented polynomial : 12 crossing knots

The following table gives the average data for the 9988 13 crossing knots. The Jenkins algorithm requires 5.39 times as many cases and takes 1.44 times as long to complete the calculation of the average 13 crossing knot. Thus, we see some slight evidence that the Jenkins implementation could improve in speed relative to the load balanced implementation with increasing number of crossings.

| | load balanced | Jenkins | ratio |
|---|---|---|---|
| average recursions | 14.2815 | 76.9757 | 5.39 |
| average time | 0.03242 $s$ | 0.04672 | 1.44 |

**Table 3.2.** Oriented polynomial : 12 crossing knots

The real purpose of our effort is the calculation of significantly larger topologically interesting examples. In this direction, the following data concerns the calculation of the oriented polynomial of a certain 25 crossing test knot.

| | load balanced | Jenkins | ratio |
|---|---|---|---|
| recursions | 951 | 1644 | 1.73 |
| time | 1.58 $s$ | 1.40 | 0.89 |

**Table 3.3.** A 25 crossing example

We next consider the calculation of the oriented polynomial of a certain 48 crossing knot. The advantage of Jenkins suggested by smaller knots is supported by this case.

| | load balanced | Jenkins | ratio |
|---|---|---|---|
| recursions | 7247038 | 84765 | 0.011 |
| time | 11965.56 $s$ | 127.63 $s$ | 0.012 |

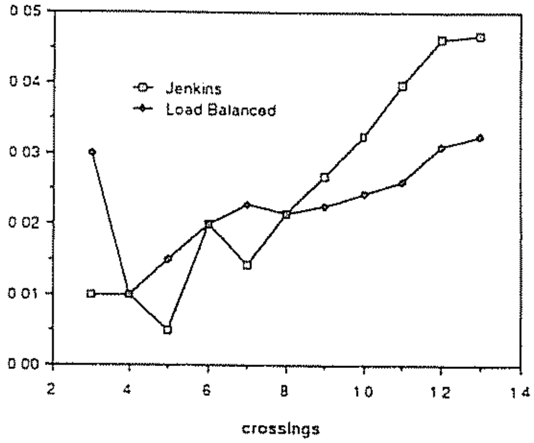
**Table 3.4.** A 48 crossing example

The following data concerns a certain 56 crossing knot, the untwisted double of the knot $13_6714$ in the Thistlethwaite enumeration. The relative advantage of the Jenkins algorithm is significantly less in this case and, therefore, is very much dependent upon the specific knot under consideration

| | load balanced | Jenkins | ratio |
|---|---|---|---|
| recursions | 66738 | 30137 | 0.452 |
| time | 185.13 *s* | 33.83 *s* | 0.183 |

**Table 3.5.** A 56 crossing example : The untwisted double of $13_6714$.

One can get some impression of the relationship between the algorithms and the number of crossings in the knot presentation by consideration of the graph of the data reflecting the average time of calculation for knots of less than 14 crossings.



**Graph. 3.6.** Average time for knots of less than 14 crossings

The 25 crossing knot was designed as a challenging test case for the load balanced algorithm. The fact that the Jenkins algorithm requires slightly less time for this knot and that its symmetric configuration would seem to imply that it would be a relatively typical 'worst case' for the Jenkins algorithm leads us to conclude that for knots of roughly 25 or more crossings, the Jenkins implementation would be faster. Unfortunately there is not yet an enumeration of knots of 14 or more crossings to fill out the trends indicated in the graph and, thereby, provide more substantial evidence of the conjecture. Some evidence is provided by the examples of the 48 and 56 crossing knots. This question is the object of ongoing research.

Another indication of the relative efficiency of the Jenkins approach is found in the consideration of the case where the load balanced algorithm is least efficient, the $(2, k)$-torus knot. For the (2,25)-torus knot the Jenkins implementation required 76 recursions and took only 0.05 seconds while the load balanced implementation required 17710 recursions and took 16.88 seconds. There is, however, a significant problem which arises in the Jenkins implementation that is avoided in the load balanced implementation. The problem is the growth of the space required to accomplish the calculation. As an easy test of this, and to determine relative speed, we attempted to calculate the oriented polynomial of the untwisted double of the untwisted double of the trefoil knot. This a knot with 90 crossings. The load balanced implementation required 101165604 recursions and 235075.00 seconds, i.e. roughly 2.7 cpu days, to calculate its polynomial. But, because of the space use constraints, such calculations are not presently possible with the Jenkins implementation.

## 4 Conclusion

The load balanced
lations of the oriented pol
The load balancing algori
and link presentation con
gorithm appears to effect
identification of relations
and links, in terms of the
algorithm requires signifi
size of the knot increases.
and links, it seems likely
increases in speed thereby

## References

[A] J W ALEXANDER, To
275-306

[BLM] R D BRANDT, W
ented knots and links, Inven

[BZ] G BURDE, H ZIESC

[Con] J H CONWAY, An e
putational problems in abstr
329-358

[CrF] R H CROWELL and
Springer 1977

[DTh] C H DOWKER and
its Applications 16 (1983),

[EM] B EWING and K C
knot and link invariants, " t
Singapore, 1990

[F] R H FOX, A quick trip
Hall (1962) 120-167

[FYHLMO] P. FREYD, D
A new polynomial invarian
Society, Vol 12, No 2, April

[Ho] C F Ho, A new polyn
abstract 821-57-16

[Hos] J HOSTE, A polyno

[Ja1] F JAEGER, Tutte pc
647-654

[Ja2] F JAEGER, D L VE
and Tutte polynomials, Ma

[Je] R J JENKINS. Knot th

2
3

le of $13_6714$.
lgorithms and the number
of the data reflecting the

crossings
ıse for the load balanced
ss time for this knot and
ld be a relatively typical
ır knots of roughly 25 or
unately there is not yet an
ıdicated in the graph and,
ɜ evidence is provided by
bject of ongoing research.
approach is found in the
efficient, the $(2,k)$-torus
ɜd 76 recursions and took
ɜed 17710 recursions and
ıich arises in the Jenkins
ion The problem is the
n easy test of this, and to
lynomial of the untwisted
with 90 crossings The
d 235075 00 seconds, i.e.
the space use constraints,
mentation.

## 4 Conclusion

The load balanced algorithm and the Jenkins algorithm, which provide recursive calculations of the oriented polynomial associated to classical knots and links, have been described. The load balancing algorithm provides a substantial extension of the effective range of knot and link presentation complexity where computations are feasible. The Jenkins greedy algorithm appears to effectively exploit the reduction in complexity associated to successful identification of relations between skein generators, especially for a middle range of knots and links, in terms of the number of crossings in their presentation. Unfortunately the Jenkins algorithm requires significantly more space, compared to the load balanced algorithm, as the size of the knot increases. Although each implementation is quite fast for middle range knots and links, it seems likely that a merging of the two methodologies could yield substantial increases in speed thereby significantly increasing the effective calculational range.

## References

[A] J W ALEXANDER, Topological invariants of knots and links, Trans Amer Math Soc 30 (1928), 275-306

[BLM] R D BRANDT, W B R LICKORISH, and K C MILLETT, A polynomial invariant for unoriented knots and links, Invent. Math 84 (1986), 563-573

[BZ] G. BURDE, H ZIESCHANG, Knots, de Gruyter (1986).

[Con] J H CONWAY, An enumeration of knots and links and some of their algebraic properties, Computational problems in abstract algebra, (John Leech, ed ), Pergamon Press, Oxford and New York 1969, 329-358

[CrF] R H CROWELL and R H FOX, Introduction to knot theory, Graduate Texts in Mathematics 57, Springer 1977

[DTh] C H DOWKER and M B THISTLETHWAITE, Classification of knot projections, Topology and its Applications 16 (1983), 19-31.

[EM] B EWING and K C MILLETT, A load balanced algorithm for the calculation of the polynomial knot and link invariants, " the Mathematical Heritage of C F Gauss", World Scientific Publishing Co , Singapore, 1990

[F] R H FOX, A quick trip through knot theory, Topology of 3-manifolds (ed M K Fort Jr ) Prentice Hall (1962) 120-167.

[FYHLMO] P FREYD, D YETTER, J HOSTE, W B R LICKORISH, K MILLETT, A OCNEANU, A new polynomial invariant for knots and links, Bulletin (New Series) of the American Mathematical Society, Vol 12, No 2, April 1985, 239-246

[Ho] C F Ho, A new polynomial for knots and links - preliminary report, Abstracts AMS 6, 4 (1985), abstract 821-57-16.

[Hos] J HOSTE, A polynomial invariant of knots and links, Pacific J Math 124 (1986), 295-320.

[Ja1] F JAEGER, Tutte polynomials and link polynomials, Proc Amer Math Soc 103 (1988), 647-654.

[Ja2] F JAEGER, D L VERTIGAN and D J A WELSH, On the computational complexity of the Jones and Tutte polynomials, Math Proc Camb Phil Soc 108 (1990), 35-53

[Je] R J JENKINS, Knot theory, simple weaves, and algorithm for computing the HOMFLY polynomial, Master of Science in Mathematics Thesis. Carnegie Mellon University, 1989.

[Jon1] V.F.R. JONES, A polynomial invariant for knots via von Neumann algebras, Bull. Am. Math. Soc. 12 (1985), 103-111.

[Jon2] V.F.R. JONES, Hecke algebra representations of braid groups and link polynomials, Ann. of Math. 126 (1987), 335-388.

[Jon3] V.F.R. JONES, On knot invariants related to some statistical mechanical models, Pacific J. Math. 137 (1989) 2. 311-334.

[Kau1] L. KAUFFMAN, State models and the Jones polynomial, Topology, 26 (1987), 395-407

[Kau2] L. KAUFFMAN, On knots, Ann. of Math. Studies 115, Princeton University Press (1987).

[Kau3] L. KAUFFMAN, An invariant of regular isotopy, Trans. AMS, 318 (2), 417-471.

[Kau4] L. KAUFFMAN, New invariants in the theory of knots, Amer. Math. Monthly 95 (1988), 195-242.

[KM] R. KIRBY & P. MELVIN, On the 3-manifold invariants of Witten and Reshetikhin-Turaev for sl(2C), preprint 1990.

[L1] W.B.R. LICKORISH, Polynomials for links. Bull. London Math. Soc. 20 (1988), 558-588.

[L2] W.B.R. LICKORISH, Three-manifold invariants from the combinatorics of the Jones polynomial, Math. Ann. 290 (1991) (4) 657-670

[L3] W.B.R. LICKORISH, Three-manifolds and the Temperley-Lieb algebra, preprint 1990.

[LM1] W.B.R. LICKORISH and K.C. MILLETT, A polynomial invariant of oriented links, Topology 26 (1987), 107-141.

[LM2] W.B.R. LICKORISH and K.C. MILLETT, The new polynomial invariants for knots and links, Mathematics Magazine 61 (1988), 3-23.

[LT] Richard J. LIPTON and Robert Endre TARJAN, A separator theorem for planar graphs, SIAM Journal on Applied Mathematics, 36 (1979), 177-189.

[RT] N.Y. RESHETIKHIN and V.G. TURAEV, Invariants of 3-manifolds via link polynomials and quantum groups, Inventiones Math., 103 (1991) (3) 547-597

[Rol] D. ROLFSEN, Knots and Links, Publish or Perish Press (1976).

[Th1] M. THISTLETHWAITE, Knot tabulations and related topics, Aspects of Topology, Ed. I. M. James and E.H. Kronheimer, L.M.S. Lecture Notes 93 (1985). 1-76.

[Th2] M. THISTLETHWAITE, On the Kauffman polynomial of an adequate link, Inventiones Math, 93 (1988), 285-296.

[Tu1] V.G. TURAEV, The Yang-Baxter equation and invariants of links, Invent. Math. 92 (1988), 527-553.

[Ve] D. VERTIGAN, The computational complexity of the HOMFLY and Kauffman invariants for links, preprint 1990.

[W1] E. WITTEN, Address to the Centennial Meeting of the American Mathematical Society, Providence, R.I., Agust 1988

[W2] E. WITTEN, Quantum field theory and the Jones polynomial, Comm. Math. Phys. 121 (1989), 351-399.

Univ. of Cal. Santa Barbara.

# On som
# to the

## 1 Introduction

Spatial graphs (i
generalization of classi
theory have their counte
in [26] where the Unkn
are obtained in [5,14] b
spatial graph.

As for the recent
and its two-variable ge
way on link diagrams.
invariants for certain c
introduced an invarian
links, coincides with a
established a simple rel
the Jomes polynomial
we shall show how to u
vertex" spatial graphs,
point out that Yamada'
this graph. This object
Four-Color problem. W
and knot theory : new p
[18,28], and a state mo
the knot-theoretic appr

## 2 An invariant o

### 2.1 Flat vertex spa

All graphs will
rected uness otherwise
terminology. The num
denoted by $v(G), e(G$
graph with no vertex a
since the graph propert

The notions to f
A *spatial graph* is a g
loss of generality we