Iterated Clasp Move and Upper Bounds for 2-Bridge Links
REU in Mathematics at CSUSB
Summer 2003

Timothy J. Nawojski

August 20, 2003

Abstract

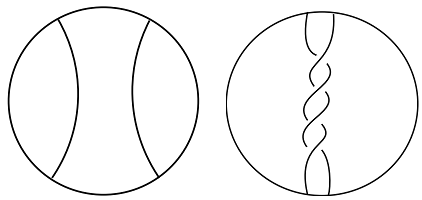
In this paper, we prove that performing a geometric technique called the iterated clasp will perform a T_{4n} move on a link. Along with that, the move also only adds 3 sticks for every time it is performed. Also applying this technique, we prove that the upper bound for 2-bridge links is $s(k) \leq m + \frac{3}{4}c(k)$.

1 Introduction

Lowering the upper bound of stick number will help construct more accurate numbers of sticks for higher numbered crossing knots. McCabe's theorem says that the upper bound on stick number for any 2 bridge link is $s(k) \leq c(k) + 3$ [1]. This is better than Negami's general result of $2c(k)$[2]. A we apply a new technique called the iterated clasps move[3] to reduce McCabe's theorem significantly for higher crossing number knots. In fact it comes asympototically close to $\frac{3}{4}c(k)$ Using the iterated clasp move when applicable can create knots that add 4 crossings but only 3 sticks. This technique is also repeatable. Making it result in lowering the stick number of knots, even less than McCabe's theorem of $s(k) \leq c(k) + 3$

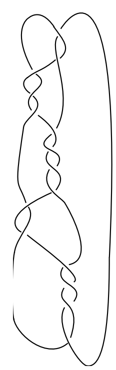
2 2-bridge Construction

To understand how to construct 2 bridge links, integer tangles are an importiant part of it. The construction of integer tangles consist of constructing a long stick, then having the tangle wrap around it. This create integer tangles. An integer tangle is part of a knot that alternates and crosses itself, each time alternating in crossings. The 4 tangle below represents how the tangle may look in the knot projection. You can alternate the crossings with it starting over, then under and alternating, or starting with a strand and making it an undercrossing. Then making it over and alternate so forth. Remember to alternate crossing throught the diagram. Because if we connect a 3 tangle with a 4 tangle but the crossing does not alternate, we do not have minimal crossing, and the knot can be untangled into another simpler knot.

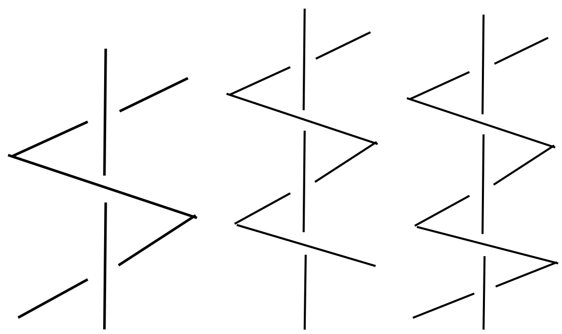


No Tangle   4 *Tangle*

We continue to do this until all the tangles are represented in the knot. Then we connect the top left to the next string that is to its right. The next one on the outer most right on top will connect to the outer most string on the right bottom. Then the last 2 will connect. Remember we are only tangling 2 string and then connecting them. So we will only have 6 ends to connect. So for example we will look at the [2,3,5,2,4] 2 bridge link. Constructing all the tangles aproprataly so that they alternate, and connecting the the strings at the ends this will look like this:

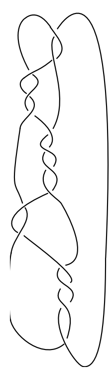


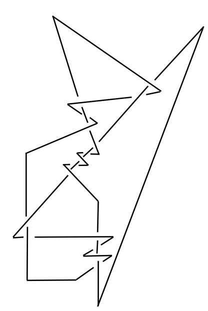
Through McCabe's stick construction of 2 bridge links, her integer tangle set up is the first step. Her setup of integer tangles constists of a long vertical axis on the integer tangles, and then wrapping the other stick around that axis to create the tangle. The 3,4, and 5 tangles below show how to wrap the edges around the axis. Constructing these integer tangle will add $| n | +1$ sticks, because we have 1 axis stick, and then we have n sticks wrapped around it.



$3-Tangle$ $4-Tangle$ $5-Tangle$

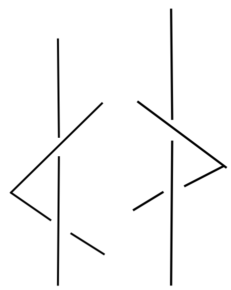
Then by seting up the tangles and connecting the bottom left, to the upper right of each tangle, we join the tangles together. An example of this tanglining is of the [2,3,5,2,4]. It starts with a integer tangle of 2, then it connect to the 3 integer tangle, connecting its bottom left, to the upper right corner. Then the next corner is the right bottom, connected to the upper left of the last tangle. So the 3 tangle will connect its bottom right strand, to its upper left strand. Alternating back and forth will yeald us the [2,3,5,2,4].





$[2,3,5,2,4]$ *knot* $[2,3,5,2,4]$ *stick representation*

We connect the last tangles bottom left to the left most tangle. Since it is odd, there will be another tangle to the left of it. The right bottom tangle is connected back to the top, and the last edges are joined up. Notice that we must have an odd number of tangles, our 2 bridge link will undo part of it. If there is a tangle such at [2,2], this can be reduced down to just a [2] 2 bridge link. Otherwise known as the Hopf link. So we must have an odd number of tangles or the bridge link can be reduced into a simpler knot. Another thing to notice is that if we start by twisting around the edge from the left, or the right, it will not matter. Both construct a 2 tangle if it wraps around the left, or the right. This may be used for edges that are of even . We will introduce notation to help define 2-bridge links. Let $[a_1, a_2, ..., a_m]$ stand for the knot we are creating. a_1 is just the first tangle, of the knot. In the example [2,3,5,2,4] starts with a 2 tangle of strands 2 and 3, then after that is a 3 tangle of strands 2 and 1, then a 5 tangle of strands 2 and 3, and so on.



both are 2-tangles

McCabe's construction is the simplest way to construct stick tangles of any number less than 5. However through a technique called the iterated clasp, we can construct 4 tangles simpler than this technique, when the correct setup is available. 2 bridge links have setups that respect this decompostion of this construction. That is when they are constructed out of integer tangles.

3 The Iterated Clasp Move

The iterated clasp move is a geometric construction used to create stick knots from other stick knots. It lowers stick numbers because Through this technique, new stick representations of knots are built from others when a specific arangment of sticks are aligned. It changes the knot into a new one with 4 more crossings, but only 3 sticks. Also, this construction can be repeated after it is done once. We say a knot respects the iterated clasp move when it has the correct alignment to perform the technique. You must have 3 edges, 2 that are connected at a vertex. Edges E_2 and E_3 have the correct setup, because E_2 and E_3 form an angle that contains E_1. Then we construct an epsilon neighborhood around the triangle formed by E_1 and V_1. (Figure 1.2), for an iterated clasp to be performed, there must be no part of the link on, or in the epsilon neighborhood of the triangle. If there are parts of the link on or in the triangle, we chose not to perform an iterated clasp move.

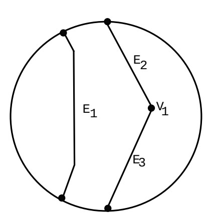


*Figure*1.1

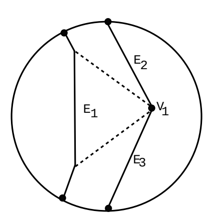
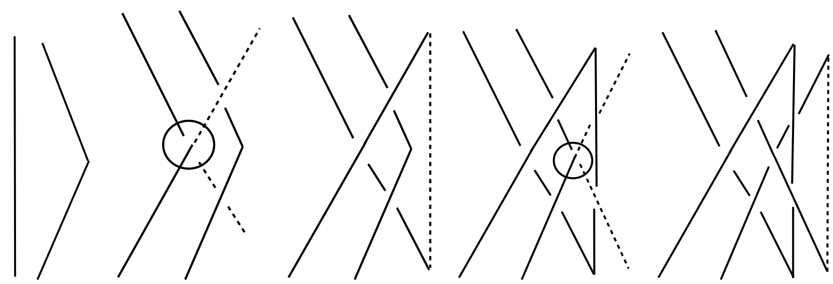


*Figure*1.2

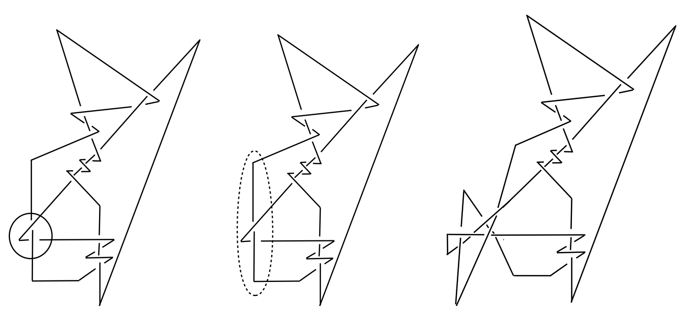
3.1 Definition of the Iterated Clasp

When the proper setup is achieved, then an iterated clasp can be performed. It begins with the breaking E_1 into 2 sticks. The once one, but now 2 sticks meet at a vertex V_2 which is immedialty behind V_1. When this happens we must resolve the crossing that was created by extending the sticks. This crossing needs to be a postitivley oriented. Step 2 shows how the crossing should look when oriented properly. If oriented the other way, we will lose a crossing. Step 3 shows the connecting of a third stick to connect the link again. Step 4 again is to extend and create another clasp starting at V_2. We must resolve this crossing that was created by postivily orienting it again. Step 5 is to close the link that we broke by extending the 2 sticks previously.



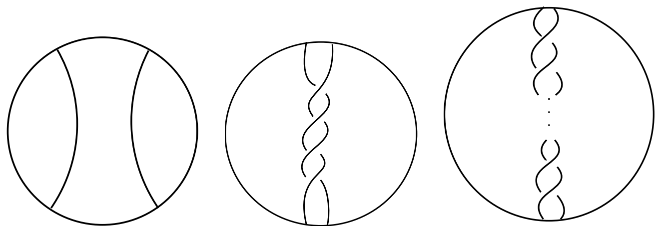
Step 1 *Step* 2 *Step* 3 *Step* 4 *Step* 5

Using the example earlier of the [2,3,5,2,4] we can perfrom the iterated clasp on the second 2 tangle. In this case there is a long vertical edge that extends through crossings that are above and below it. The big circle highlights E_1 of our base case for the clasp. However the triangle that is constructed from E_1 to the vertex joining E_2 and E_3 does not have any parts the knot passing through this plane. We then therefore can perform an iteratted clasp to the knot without interfering with the rest of the knot. Doing so has changed the tangle. In doing so it has added 4 more crossings to that tangle, making it a 6.



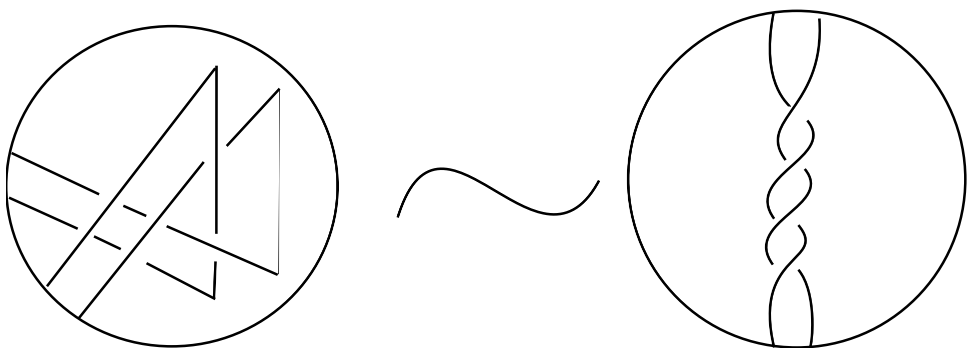
3.2 The Results of the Iterated Clasp

Once completed, it changes our original knot that respects the iterated clasp to a new representation. From the picture with no tangles in it, it creates 4 crossings. When looked at topologically, it constructs an integer tangle. An integer tangle is a knot that has n crossings that alternate such a tangle is refered to as T_n. The iterated clasp move constructs from the original knot with no tangle in it to a link with a 4 tangle. This is because it creates 4 crossings. This is called a T_4 move.



No Tangle $\quad$ T_4 Tangle $\quad$ T_n Tangle

Theorem 1 *Performing n-iterated clasp moves on a link with appropriate choice of double point resolution, is topologically equivalent to performing a T_{4n} move.*



Corollary 1 *Performing n-iterated clasp moves on a link will construct a new knot with 3n more sticks.*

3.3 Proof of Iterated Clasps

This can be proven with induction. Step One. Our base case

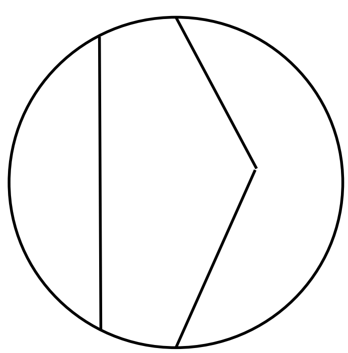


Figure 1.3

Any stick arangment in an epsilion radius sphere that contains one stick and has 2 other sticks to the side of it that are connected at a vertex (Figure 1.1). Constructing the triangle from the vertex of the 2 meeting sticks with the corner edges of the other stick that intersects the sphere (Figure 1.2) . As long as there are no intersecting strands, a clasp move can be applied to it. This will change the tangle into a T_4 move, or an iterated clasp as we have defined. 4 Crossings were added and 3 sticks (Figure 1.3) were added to our original link that respected an iterated clasp move

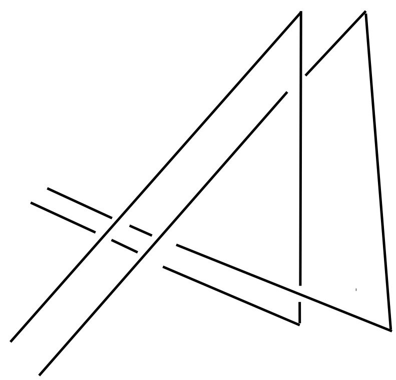


Figure 1.4

Step 2. The n-case
Assuming that the we have an iterated clasp, we can perform n-iterated clasps to it. Peforming the clasp move will construct another set of edges like our base case, this being the tangle that looks like Figure (1.3).

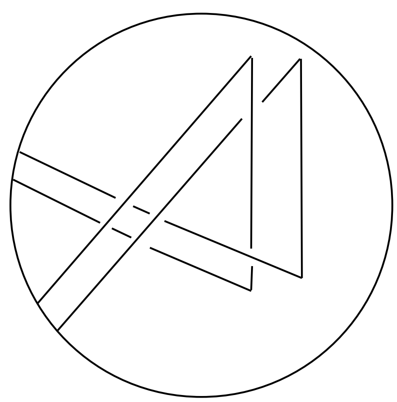


Figure 1.5

Figure 1.5 is the tangle that was formed by the n^{th} iterated clasp. The corner at the end can be used again to perform another iterated clasp move

Step 3. The n+1 case
Assuming that n cases of the iterated clasp can be constructed, we must show that the n+1 case can also be constructed. Taking an n-iterated claps move, we need have a corner and another edge with nothing intersecting it to introduce another clasp. An n-iterated clasp move will provide for an edge and a corner that will be used to create a clasp move. The first step is to change the stick that is not broken, that looks like the setup from our base case

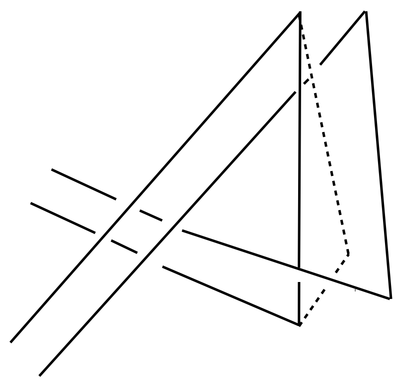


Figure 1.6

In Figure 1.6 we took the one stick and broke it into 2 pieces. This way there are now edges that can be clasped around another. Next step is to extend the sticks that we already have. This crossing may be chosen to go either way, over or under.

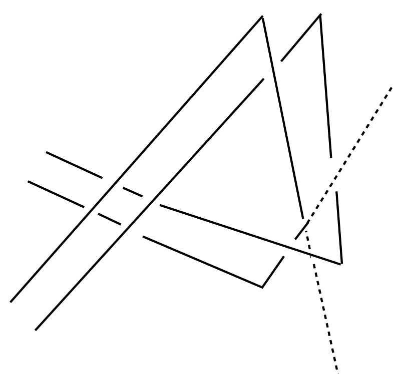


Figure 1.7

After that, we construct a stick that will connect the 2 extended pieces.(Figure 1.7)

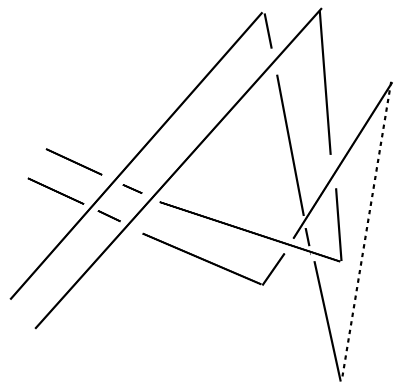


Figure 1.8

Connecting the 2 pieces we have created one of the clasps (Figure 1.8). However this can be undone with Riediemister moves. The additional clasp must be added to hold the first clasp in place. Now we extend the 2 sticks that we're needed in our base case. Up until now they have not been altered, but now we are going to extend those stick, and they will cross each other. This crossing is again arbitrary, it will change the knot you see, but not the number of sticks or crossings in it.

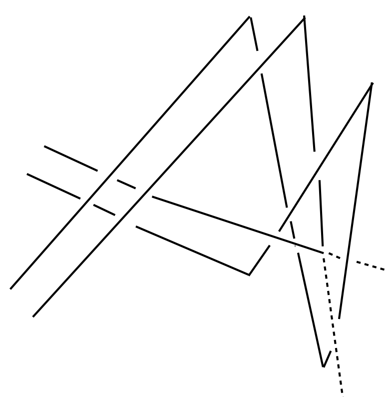


Figure 1.9

our final step is to complete the clasp move by constructing the final stick that connects the two extending sticks in such a way.

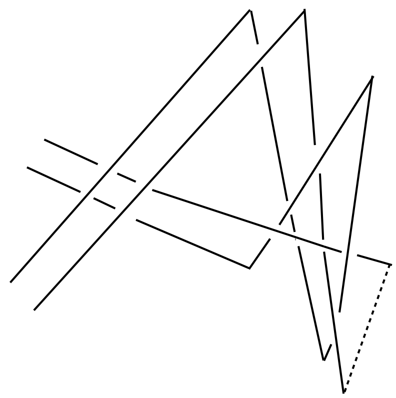
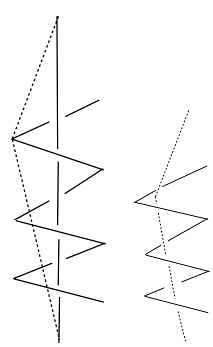


Figure 1.10
The iterated clasp move has now added an additional 3 sticks and 4 crossings (Figure 1.10). Also it has yeilded another straight edge and 2 edges bent inward that can again be used to construct another clasp move. This proving corollary 1 is true.
Each time we repeat the move, it adds the 4-tangle in between the second and third crossings in the tangle. this is done repeatedly depending on the number of times that you perform the clasp move. Thus giving you a tangle of T_{4n}

Lemma 1 *$a_1 .... a_n$, such that $a_i \geq 2$*
Integer tangles in 2 bridge links respects the natural tangle compostion that can have an iterated clasp performed upon it.

Proof
Each tangle constructed will have one longer horizontal stick that is crossed multiple times. This stick can be used as the long edge E_1 in the set up of the iterated clasp move. Breaking this stick does not change the crossings on it. Geometricly you can still construct the stick knot. Then you can chose 2 edges that form one of the tangles. These 2 edge will always fit the setup of the iterated clasp, and can be performed always. This holds true for any tangle on the knot 2-bridge link.



step 1   *step* 2

4 Upper Bound for 2-Bridge links

Now that we have an explanation of notation for the 2 bridge link we will use the iterated clasp to reduce the upper bound for the stick number required to create an alternating 2-bridge link in the form $[a_1, a_2, ..., a_m]$. This is just notation for our integer tangles. Given a knot in this form, we can reduce this knot into it's 4 base cases where it comes from, such that they are if a = 2,3,4 or 5. We will ignore the base case of a knot that has a 1 tangle in it because their are no edges for the clasp move to form around. Also we can ignore if the edges are reversed, as if the 2 tangle stick representation, was over, under, as to opposed to under than over [1]. This is accepable as long as the links are still alternating links.

4.1 Upper Bound

Theorem 2

$$\frac{1}{4}(\overline{a_1} + \overline{a_2} + ..\overline{a_n}) + \frac{3}{4}c(k) + 3 \ \geq s(k)$$

Using the iterated clasp, a newer upper bound for 2 bridge links can be found.

4.2 Proof

In a two bridge link the first situation that may occur for our base case is that of a 2 tangle. Such a tangle has now only 2 crossings. Here whenever their is a 2 tangle of sticks arranged like this, we may apply the clasp move right away according to lemma 1. Now any 2 tangle can insert 4 more crossings repeatedly.

Property 1 *Performing n-iterated clasp moves on a 4-tangle will produce a 2+4n tangle.*

However, property 1 states we only add 3n sticks to our stick knot. McCabe says it will take 4(n) more sticks to construct because this is the number of tangles we are adding. Clearly through iterated clasps, it can be done more effiecently.

Clearly we do not need a case of a=6, for we can creat a 6-tangle through one iterated clasp on the 2 tangle.Therefore we peform the iterated clasp to a 3,4 or 5 tangle, which respectivly has that many crossings. When constructing an iterated clasp on a tangle, there is more than 2 crossings that our tangle will contain, when constructing a cirlce around the edge and 2 other edges that form an angle which is less than 180 degrees. However, through lemma 1, the iterated clasp can be performed. No interference from any other parts of the knot disrupt its compostition, and allows for an iterated clasp

Property 2 *Performing n-iterated clasp moves on a 3-tangle will produce a 3+4n tangle.*

Property 3 *Performing n-iterated clasp moves on a 4-tangle will produce a 4+4n tangle.*

Property 4 *Performing n-iterated clasp moves on a 5-tangle will produce a 5+4n tangle.*

Now that we have our base cases set, we can construct any 2-bridge link of the form ${\sigma_2}^{-a_1}, \sigma_1^{a_2}, ..., {\sigma_2}^{-a_m}$.

Since we know how to construct our 2-bridge link and understand the effects of the itterated clasp, we may apply them to find an upper bound for any 2-bridge link. Let $\overline{a_1}, \overline{a_2}, \overline{a_3}...\overline{a_n}$ represent the number of sticks left over when we take the crossing number and divide it by 4. This is importiant for this is going to be the base tangle that will be used to create a situation for our iterated clasp. However if it has a remainder of 0, or 1 we cannot use these, for they represent a link if it were constructed from a 0-tangle, or a 1-tangle. However we did build them from a 4 or 5 tangle, then $\overline{a_1}, \overline{a_2}, \overline{a_3}...\overline{a_n} \in \{2, 3, 4, 5\}$. This will represent the base knot that we are contstructiong the 2-bridge link from. We must remember that our base knot is still build from the $c(k)+3$. So that must be taken into account for the stick number. The base is just however which is just $\overline{1}, \overline{a_2}, .., \overline{a_n}$. If we take the sum of $\overline{a_1}+\overline{a_2}+..\overline{a_n}+3$ we have the construction of our base knot. Then performing iterated clasp to the individual tangle and changing the tangles according to properties 2,3,4 and 5, we construct the knot back, however using the iterated clasp's corollary one, it says we only need 3 sticks for every 4 crossing now.
When we look at it algebraicly:

$$(\overline{a_1}+\overline{a_2}+...+\overline{a_n})+3+3\times(\frac{a_1-\overline{a_1}}{4}+\frac{a_2-\overline{a_2}}{4}+...+\frac{a_n-\overline{n}}{4}) \geq s(k)$$

The $\overline{a_1}+\overline{a_2}+...+\overline{a_n}+3$ is just the base case of sticks necessary to perform the itterated clasp from. Then the $3\times(\frac{a_1-\overline{a_1}}{4}+\frac{a_2-\overline{a_2}}{4}+...+\frac{a_n-\overline{a_n}}{4}$ is just the number of times we can do an iterated clasp, and each time we only add 3 sticks to it. Then through some algebra we can reduce the equation to:

$$\frac{\overline{a_1}+\overline{a_2}+..\overline{a_n}}{4}+\frac{3\times(a_1+a_2+...+a_n)}{4}+3$$

$(\mathrm{a}_1, a_2...a_n) = c(k)$ though. So therefore the lower bound is equivilant to

$$\frac{\overline{a_1}+\overline{a_2}+..\overline{a_n}}{4}+\frac{3\times(c(k)}{4}+3 \geq s(k)$$

This bound is more accurate than McCabe's $c(k)+3$, as long as $\frac{\overline{1}+\overline{2}+..\overline{n}}{4} \leq \frac{c(k)}{4}$. Now $\frac{\overline{a_1}+\overline{a_2}+..\overline{a_n}}{4}$ when $\overline{a_1}+\overline{a_2}+..\overline{a_n}$ is at its maximum, it will be when the remainders are all 5. Such that we will have $5n$.

$$\frac{5n}{4} \leq \frac{c(k)}{4}$$

$$5n \leq c(k)$$

$$n \leq \frac{c(k)}{5}$$

Although it will work better than McCabe's theorem for some cases than above, this bound will always show that this theorem will work better than McCabe's theorem as long as the fraction above is satisfied.

5 Acknowledgments

I would like to thank R. Trapp, and J. Chavez for all their help, guidance, and feedback on everything. This research was completed during the 2003 REU in Mathematics at California State University in San Bernardino California, jointly sponsered by CSUSB and NSF-REU Grant DMS-0139426.

References

[1] C. McCabe, *An Upper Bound on Edge Numbers of 2-Bridge Knots and Links*, J. Knot Theory Ramif. 7 (1998) 797-805

[2] S. Negami, *Ramsey theorems for knots, links, and spatial graphs*, Trans. Amer. Math. Soc 324 (1991) 527-541

[3] A.O'Connor, B. Podlesny, N. Soriano, R. Trapp, and D. Wall, *Clasp Moves and Stick Number*, Preprint

[4] C. Adams, *The Knot Book*, W.H.Freeman Co., New York, (1994)

[5] G. Burde and H. Zieschang, *Knots*, Walter de Gruyter Co. (1985)

[6] L. Kauffman,*State models and the Jones polynomial*, Topology 26 (1987) 395-407

[7] K. Murasugi, *Jones Polynomials and classical conjectures in knot theory*, Topology 26 (1987) 187-194

[8] M. Thistlethwaite, *A spanning tree expansion of the Jones polynomial*, Topology 26 (1987) 297-309