Journal of Knot Theory and Its Ramifications
Vol. 18, No. 8 (2009) 1159–1173

World Scientific
www.worldscientific.com

MINIMAL KNOTTING NUMBERS

CASEY MANN, JENNIFER MCLOUD-MANN*, RAMONA RANALLI
and NATHAN SMITH

University of Texas at Tyler,
3900 University Blvd. Tyler, TX 75799, USA
**jmcloud@uttyler.edu*

BENJAMIN MCCARTY

Louisiana State University,
303 Lockett Hall, Baton Rouge LA 70803, USA

Accepted 23 June 2008

ABSTRACT

This article concerns the minimal knotting number for several types of lattices, including the face-centered cubic lattice (fcc), two variations of the body-centered cubic lattice (bcc-14 and bcc-8), and simple-hexagonal lattices (sh). We find, through the use of a computer algorithm, that the minimal knotting number in sh is 20, in fcc is 15, in bcc-14 is 13, and bcc-8 is 18.

Keywords: Knots; minimal knotting numbers; lattices; knot reductions.

Mathematics Subject Classification 2000: 57M27, 03G10

1. Introduction

Knots in the simple cubic lattice (sc or $\mathbb{Z}^3$) have received some attention in past articles (see [2,13–16]). These lattice knots are simple closed polygonal curves made from unit segments whose endpoints lie in $\mathbb{Z}^3$. One of the most obvious questions to ask is how many unit edges it takes to form a nontrivial knot in $\mathbb{Z}^3$. This question was answered by Diao in [2]; the answer is 24. That is, the *minimal knotting number* in $\mathbb{Z}^3$ is 24. Such a minimal knot is shown in Fig. 1.

Natural lattice-specific invariants can be defined for knot types, such as the number of unit edges required to form a knot of the given type in $\mathbb{Z}^3$ (the *minimal step number* of a knot type). Janse van Rensburg used simulated annealing algorithms to estimate these invariants for several knot types of small crossing number in $\mathbb{Z}^3$ [13].

Our goal in this paper is to generalize the result of Diao concerning minimal knotting numbers to other three-dimensional point lattices. It was conjectured by Janse van Rensburg in [16] that the minimal knotting number in the face-centered cubic lattice is 16. We show this conjecture is false, and we will give the minimal

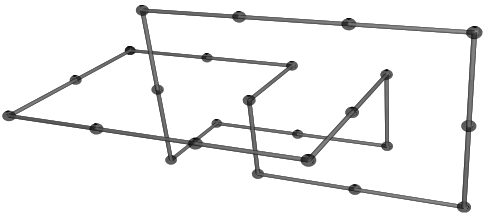

Fig. 1. A minimal $\mathbb{Z}^3$-lattice knot.

knotting numbers of other lattices as well. In order to do so, some definitions will be needed.

1.1. *Definitions*

A *point lattice* in $\mathbb{R}^3$ is a $\mathbb{Z}$-module $\mathcal{L} = \{a_1v_1 + a_2v_2 + a_3v_3 | a_i \in \mathbb{Z}, v_i \in \mathbb{R}^3\}$. The *simple cubic lattice*, denoted *sc* or $\mathbb{Z}^3$, is a point lattice with basis $\{(1,0,0),(0,1,0),(0,0,1)\}$. If we define the *neighbors* of $(0,0,0)$ to be $\{(\pm1,0,0),(0,\pm1,0),(0,0,\pm1)\}$, we can then obtain the neighbors of any other point in $\mathbb{Z}^3$ by translation as lattices are invariant up to translations along directions which are linear combinations of the basis vectors.

In this paper, we consider knots in four other point lattices, which we will define by giving a basis and neighbors of $(0,0,0)$. These lattices are the *simple hexagonal lattice (sh)*, the *face-centered cubic lattice (fcc)*, and two versions of the *body centered cubic lattice* which we call *bcc-8* and *bcc-14*. For *sh* the basis is

$$\{h_1,h_2,h_3\} = \left\{(1,0,0), \left(\frac{1}{2},\frac{\sqrt{3}}{2},0\right),(0,0,1)\right\}$$

and the eight neighbors of zero are

$$\left\{\pm(1,0,0), \pm\left(\frac{1}{2},\frac{\sqrt{3}}{2},0\right), \pm\left(-\frac{1}{2},\frac{\sqrt{3}}{2},0\right), \pm(0,0,1)\right\},$$

and for *fcc* the basis is

$$\{f_1,f_2,f_3\} = \left\{(1,0,0),(0,1,0),\left(\frac{1}{2},\frac{1}{2},\frac{1}{\sqrt{2}}\right)\right\}$$

and the twelve neighbors of zero are

$$\pm(1,0,0), \quad \pm(0,1,0), \quad \pm\left(\frac{1}{2},\frac{1}{2},\frac{1}{\sqrt{2}}\right),$$
$$\pm\left(-\frac{1}{2},\frac{1}{2},\frac{1}{\sqrt{2}}\right), \quad \pm\left(\frac{1}{2},-\frac{1}{2},\frac{1}{\sqrt{2}}\right), \quad \pm\left(-\frac{1}{2},-\frac{1}{2},\frac{1}{\sqrt{2}}\right).$$

Each of these lattices is well-known, and for more details we point the reader to [8,17].

For the *body centered cubic lattice* we have basis

$$\{b_1, b_2, b_3\} = \{(1,0,0), (0,1,0), (1,1,1)\}$$

and consider two different ways of defining the neighbors of $(0,0,0)$. The first of these, which we will call *bcc-8*, is the one typically used by chemists, and results in a total of eight neighbors of (0,0,0):

$$\{\pm(1,1,1), \pm(-1,1,1), \pm(1,-1,1), \pm(-1,-1,1)\},$$

again we refer the reader to [17]. While this notion of neighbors works well for describing crystallographic structures of atoms, we were motivated by the connection between lattices and their corresponding Voronoi tessellations [18], which is what led us to think about different versions of *bcc*. Briefly, there are only five combinatorially distinct convex polyhedra which tessellate 3-space by translation, and the centroids of these polyhedra in a tessellation determine a point lattice. It seems natural to define neighbors in a lattice in such a way that two points in the lattice are neighbors if their polyhedra share a face. The tessellation of 3-space by cubes gives the *sc* lattice, the tessellation of 3-space by hexagonal prisms gives the *sh* lattice, and the tessellation of 3-space by rhombic dodecahedra gives the *fcc* lattice. We point out that a tiling of space by elongated dodecahedra corresponds to a point lattice which, for our purposes, is combinatorially equivalent to fcc, so we did not consider it separately. Finally, the tessellation by truncated octahedra gives the points of *bcc*, but the neighbors determined by polyhedra sharing a face results not in the eight neighbors of $(0,0,0)$ given above, but rather in the following fourteen neighbors:

$$\{\pm(1,0,0), \pm(0,1,0), \pm(1,1,1), \pm(-1,1,1), \pm(1,-1,1), \pm(-1,-1,1), \pm(2,0,0)\},$$

giving bcc-14.

Given two neighboring lattice points, the straight line segment joining them is called a *step*. We point out that in bcc-14 there are two distinct step lengths: the ratio of the distance between neighbors corresponding to hexagonal faces of the truncated octahedron to neighbors corresponding the square faces is $1 : \sqrt{3}$. The minimum number of steps required to form a knot in a given lattice is the *minimal knotting number* of the lattice.

2. Main Result

Theorem 2.1. *The minimal knotting numbers are summarized below*:

(1) *The minimal knotting number in sh is* 20.
(2) *The minimal knotting number in fcc is* 15.
(3) *The minimal knotting number in bcc*-14 *is* 13.
(4) *The minimal knotting number in bcc*-8 *is* 18.

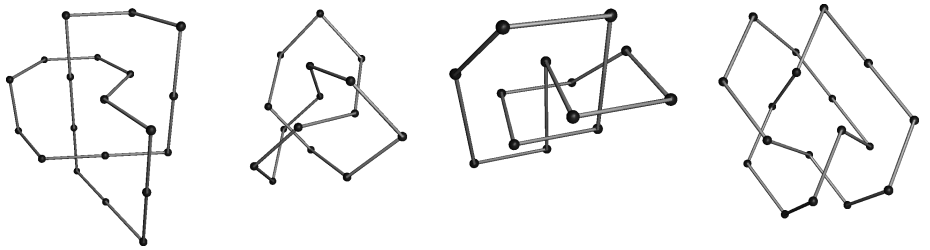

Fig. 2. Minimal knots (left to right) in sh, fcc, bcc-14, and bcc-8.

The knots in Fig. 2 certainly establish upper bounds on the minimal knotting numbers for sh, fcc, bcc-14, and bcc-8. To verify that knots (other than the unknot) formed from fewer steps in each of these lattices do not exist, our approach was to use a computer program to form every simple closed curve in each lattice of length less than or equal to the minimal knotting number for that lattice. We then used a loop reduction algorithm to verify that (1) all closed loops of length less than the claimed minimal knotting number can be shrunk via ambient isotropy to the point where they are trivial and (2) nontrivial knots of the claimed minimal length exist. This resulted in a listing of nontrivial knots of minimal length. Many of the knots in this list were congruent, so the list of knots could be reduced somewhat by factoring out obvious congruences. Upon visual inspection of these remaining knots, all appear to be trefoil knots. The algorithms used are described in the next section.

3. Details of the Computations

3.1. *Metrics in sh, fcc, and bcc*

In order for our algorithm to operate efficiently, we need metrics for each of the lattices; that is, for each of the lattices, we need a function that can count the minimum number of steps required to connect two lattice points. Such a function for fcc was demonstrated in [12], but to the best of the authors' knowledge, no such functions have been previously demonstrated for sh, bcc-14, or bcc-8. In a separate paper, we derive the metrics for sh, bcc-14, and bcc-8 and prove that they do in fact count the minimum number of steps required to connect two lattice points [11]. One way to express these metrics involves using coordinates relative to the bases given above. For example, for *sh* we would write $xh_1 + yh_2 + zh_3$ as simply (x, y, z). Also, for simplicity our metrics measure the distance from the lattice point (x, y, z) to the origin. Some of the metrics are given in the following theorem:

Theorem 3.1 (Lattice Metrics #1). *Define*

(1) [sh] $d_{sh}(x, y, z) = \max\{|x|, |y|, |x + y|\} + |z|$.
(2) [fcc] $d_{fcc}(x, y, z) = \max\{|x + y + z|, |x - y|, |z|, \frac{1}{2}(|x + y + z| + |x - y| + |z|)\}$.

Then, d_{sh} and d_{fcc} are metrics that measure the minimum number of steps required to connect the point (x, y, z) expressed in terms of basis of sh and fcc (respectively) to the origin.

Part 2 of Theorem 3.1 is given in [12]. Using the Conway–Sloan characterization of the fcc (the subset of points (x, y, z) in $\mathbb{Z}^3$ for which $x + y + z$ is even), a second metric for fcc was found as well. In [11], we give a simplified version of that metric,

$$\bar{d}_{\text{fcc}}(x, y, z) = \max\left\{|x|, |y|, |z|, \frac{1}{2}(|x| + |y| + |z|)\right\},$$

and go on to give metrics for both bcc-14 and bcc-8. Since fcc, bcc-14, and bcc-8 are geometrically and combinatorially similar, we can model bcc-14 and bcc-8 as fcc with added or removed adjacencies. Thus, using the Conway–Sloan characterization of fcc to model bcc-14 and bcc-8, we get the following theorem:

Theorem 3.2 (Lattice Metrics #2). *Define*

(1) [bcc-14] $d_{bcc\text{-}14}(x, y, z) = \max\{|x|, |y|, \frac{1}{2}(|x| + |y| + |z|)\}$.
(2) [bcc-8] $d_{bcc\text{-}8}(x, y, z) = \max\{|x|, |y|, |z|\}$.

Then $d_{\text{bcc-14}}$ and $d_{\text{bcc-8}}$ are metrics that measure the minimum number of steps required to connect the point (x, y, z) expressed in terms of the Conway–Sloan description of bcc-14 and bcc-8 (respectively) to the origin.

3.2. *The knot reduction algorithm*

In each of the lattices studied, surrounding each lattice point is a set of neighbors, and corresponding to each of these neighboring points is a direction vector. So, each lattice $\mathcal{L}$ has associated with it a set of neighbor vectors, $\mathcal{A}_{\mathcal{L}}$. A polygonal curve formed from edges connecting neighboring lattice points can then be represented by a list

$$(x_1, x_2, x_3, \ldots, x_n),$$

where $x_i \in \mathcal{A}_{\mathcal{L}}$. This is, essentially, the data structure for polygonal lattice curves that we used in our program. We note that the neighbor vectors of the lattice can be expressed in terms of a basis for the lattice so that all arithmetic needed to be performed in the algorithm can be done with integers.

If $(x_1, x_2, x_3, \ldots, x_n)$ is a closed lattice curve, to determine whether or not this curve is knotted, we developed an algorithm that attempts to reduce the curve via a sequence of transformations that do not change the knot type of the curve (i.e. the transformations never introduce self intersection and never allow one strand to pass through another). Thus, the original loop and the reduced loop will be isotropic. If this sequence of transformations reduces the given loop to a length of four or less, then we are assured that the original loop is isotropic to the unknot. If the

original loop does not reduce to length four or less, we record the original loop as a suspected knot, which is later checked for knottedness by other means.

For sh, fcc, and bcc-14, there are only two kinds of transformations needed. We call these *parallel moves* and *triangular moves*. bcc-8 requires parallel moves along with two more kinds of transformations we call *square moves* and *flip moves*. We will illustrate parallel and triangle moves for sh only. They are similar in fcc and bcc-14. In sh, the neighbor vectors are $a_1 = (1,0,0)$, $a_2 = (-1,0,0)$, $a_3 = \left(\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$, $a_4 = \left(-\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0\right)$, $a_5 = \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0\right)$, $a_6 = \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0\right)$, $a_7 = (0,0,1)$, $a_8 = (0,0,-1)$. Let $\mathcal{A}_{sh} = \{a_1, a_2, \ldots, a_8\}$. We note that $a_2 = -a_1$, $a_4 = -a_3$, $a_6 = -a_5$, and $a_8 = -a_7$. Let $K = (x_1, x_2, \ldots, x_n)$ with $x_i \in \mathcal{A}_{sh}$ be a closed loop in sh.

3.2.1. *Parallel moves*

A parallel move on K in the direction of a_i is any transformation of K of the form

$$\begin{gathered}(x_1, \ldots, x_{j-1}, \underline{a_i}, x_{j+1}, \ldots, x_{k-1}, \underline{-a_i}, x_{k+1}, \ldots, x_n) \\ \downarrow \\ (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n).\end{gathered}$$

Thus, a parallel move on K in the direction of a_i simply deletes an instance of a_i and an instance of $-a_i$ in K. This move reduces the number of steps in the curve by 2.

For example, the knot at left in Fig. 3 is then given by

$$(a_7, \underline{a_7, a_1, a_6, a_8}, a_8, a_8, a_2, a_2, a_5, a_7, a_3, a_7, a_6, a_4, a_6, a_8, a_8, a_5, a_5, a_7, a_7).$$

Upon deleting the first instance of the pair (a_7, a_8), we get the knot at left, given by

$$(a_7, \underline{a_1, a_6}, a_8, a_8, a_2, a_2, a_5, a_7, a_3, a_7, a_6, a_4, a_6, a_8, a_8, a_5, a_5, a_7, a_7).$$

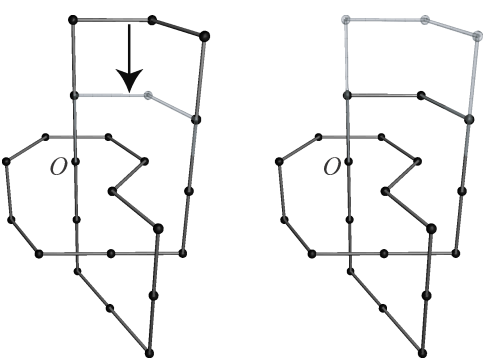

Fig. 3. A parallel move.

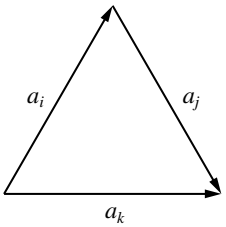

Fig. 4. $a_i + a_j = a_k$.

It is certainly recognized that such a move may result in self-intersection. In our reduction algorithm, each time such a move is performed, the resulting curve is checked to see if it is simple.

3.2.2. *Triangular moves*

Let a_i, a_j, and a_k be three neighbor vectors for which $a_i + a_j = a_k$, as in Fig. 4. A triangular move is any move of one of the following two forms (the *front triangular move* and the *back triangular move* corresponding to the relation $a_i + a_j = a_k$):

$$(x_1, \ldots, x_{j-1}, \underline{a_i}, x_{j+1}, \ldots, x_{k-1}, \underline{a_j}, x_{k+1}, \ldots, x_n)$$
$$\downarrow$$
$$(x_1, \ldots, x_{j-1}, \underline{a_k}, x_{j+1}, \ldots, x_{k-1}, x_{k+1}, \ldots, x_n)$$

or

$$(x_1, \ldots, x_{j-1}, \underline{a_i}, x_{j+1}, \ldots, x_{k-1}, \underline{a_j}, x_{k+1}, \ldots, x_n)$$
$$\downarrow$$
$$(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{k-1}, \underline{a_k}, x_{k+1}, \ldots, x_n).$$

We note that a triangular move reduces the length of the curve by one step. As an example, in Fig. 5, we see the following triangular move being performed

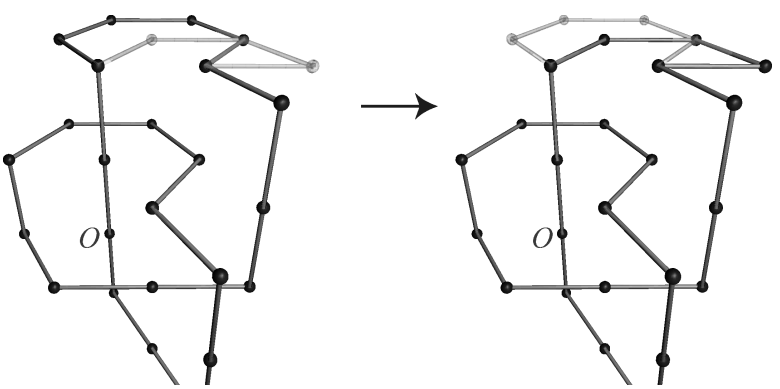

Fig. 5. A triangular move.

(using the relationship $a_4 + a_5 = a_2$):

$$(a_7, a_7, \underline{a_5}, a_3, a_1, a_6, \underline{a_4}, a_6, a_8, a_8, a_2, a_2, a_5, a_7, a_3, a_1, a_6, a_4, a_6, a_8, a_8, a_5, a_5, a_7)$$
$$\downarrow$$
$$(a_7, a_7, a_3, a_1, a_6, \underline{a_2}, a_6, a_8, a_8, a_2, a_2, a_5, a_7, a_3, a_1, a_6, a_4, a_6, a_8, a_8, a_5, a_5, a_7).$$

3.2.3. *Square moves (bcc-8 only)*

In bcc-8, the neighbor vectors are $a_1 = (1, 1, 1)$, $a_2 = (-1, -1, -1)$, $a_3 = (-1, 1, 1)$, $a_4 = (1, -1, -1)$, $a_5 = (-1, -1, 1)$, $a_6 = (1, 1, -1)$, $a_7 = (1, -1, 1)$, $a_8 = (-1, 1, -1)$. Exactly as before, let $\mathcal{A}_{\text{bcc-8}} = \{a_1, a_2, \ldots, a_8\}$ so that a curve K in bcc-8 can be expressed as $K = \{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathcal{A}_{\text{bcc-8}}$. Suppose that x_i, x_j, and x_k are consecutive edges of K for which the initial point of x_i and the terminal point of x_k are neighbors. Then a *square move* is a move of the form

$$(x_1, x_2, \ldots, \underline{x_i, x_j, x_k}, \ldots, x_n)$$
$$\downarrow$$
$$(x_1, x_2, \ldots, \underline{x_m}, \ldots, x_n)$$

where x_m is the edge connecting the initial point of x_i to the terminal point of x_k. We illustrate a square move in Fig. 6.

3.2.4. *Flip moves (bcc-8 only)*

A *flip move* is a transformation of a given closed loop in bcc-8 which does not change the length of the closed loop. In bcc-8, there exist closed curves which are clearly trivial knots but to which none of the previous moves apply. For example, consider the curve $\{a_1, a_1, a_8, a_8, a_5, a_5, a_3, a_3\}$ depicted at left in Fig. 7. The idea of a flip move is to change the closed loop without changing its knot type to a different closed loop to which one of the previous moves *does* apply. All flip moves replace a pair of consecutive edges between which the angle is $\theta = \arccos(1/3)$ by rotating them about the axis through the initial point of the first edge and the terminal point of the second edge by an angle of either $\pi/2$ or π. In Fig. 7, we illustrate such a transformation.

3.2.5. *The algorithm*

For a given lattice, there are only a handful of possible types of moves. Thus, the basic idea for our reduction algorithm is to apply all of these possible moves in

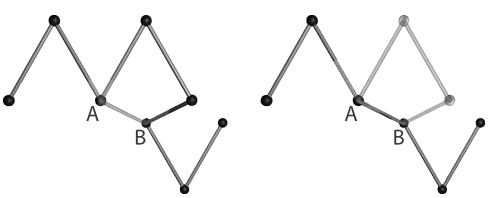

Fig. 6. A square move in bcc-8. The points A and B are neighbors.

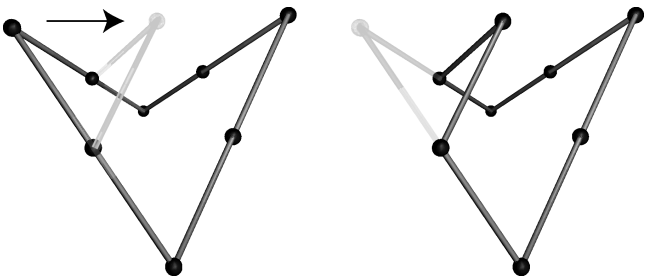

Fig. 7. A flip move in bcc-8.

some order until one of the moves successfully changes the closed curve, and then repeat this until either the closed curve reduces all the way or it cannot be reduced any further. For the purpose of illustrating our ideas, we will focus on how this is done in sh, with the other lattices being done similarly.

Let the parallel move in the direction of a_i be denoted by $[a_i, -a_i]$. The triangular moves corresponding to the relation $a_i + a_j = a_k$ will be denoted by $[a_k, a_i, a_j]$ or $[a_i, a_j, a_k]$ (for the front and back triangular moves, respectively). In sh we have a total of 16 possible moves (see Table 1).

The algorithms then goes as follows:

(1) Start with a simple closed lattice curve $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
(2) Starting with m_1 and going in order to m_{16}, apply the moves to $\mathbf{x}$ until either one of the following happens:
   (a) $\mathbf{x}$ is successfully reduced by m_i (i.e. reduced without introducing self intersection or allowing strand passage). Let $\mathbf{x} = m_i(\mathbf{x})$
      (i) If $\mathbf{x}$ has been reduced completely, then $\mathbf{x}$ is the unknot.
      (ii) If $\mathbf{x}$ has not been completely reduced, return to the beginning of step 2.
   (b) $\mathbf{x}$ is not reduced by any of the moves. If this happens, we store x as a probable knot and analyze it later.

Note that in our enumeration of moves we listed the parallel moves first. This makes the algorithm more efficient because parallel moves reduce the number of steps in the curve by 2, while triangle moves reduce the number of steps in the curve

Table 1. sh moves.

Parallel	Front triangular	Back triangular
$m_1 = [a_1, a_2]$	$m_5 = [a_3, a_1, a_5]$	$m_{11} = [a_1, a_5, a_3]$
$m_2 = [a_3, a_4]$	$m_6 = [a_6, a_1, a_4]$	$m_{12} = [a_1, a_4, a_6]$
$m_3 = [a_5, a_6]$	$m_7 = [a_5, a_2, a_3]$	$m_{13} = [a_2, a_3, a_5]$
$m_4 = [a_7, a_8]$	$m_8 = [a_4, a_2, a_6]$	$m_{14} = [a_2, a_6, a_4]$
	$m_9 = [a_1, a_3, a_6]$	$m_{15} = [a_3, a_6, a_1]$
	$m_{10} = [a_2, a_4, a_5]$	$m_{16} = [a_4, a_5, a_2]$

by 1. We also point out that this reduction algorithm is not capable of reducing *all* unknots, regardless of their length or complexity (see [7, pp. 13–14] for an example of such a long tangled unknot). However, since we are using the algorithm to find *minimal* knotting numbers, we consequently do not attempt to reduce knots of great length, and hence our algorithm never encounters such unknots which it cannot reduce.

3.3. *The exhaustive knot finding algorithm*

The program utilized to determine the minimal knotting number for each of these lattices is fairly straight forward. The gist of it is the following: Given a positive integer $N \geq 4$, the program forms all possible closed polygonal curves (up to some rigid motions) of length less than or equal to N in the lattice (using a so-called "worm" algorithm) and attempts to reduce each of these closed curves to the unknot using the previously discussed reduction algorithm. Any closed curve that cannot be reduced is saved to disk and analyzed later. So, for example, when we wanted to show that the minimal knotting number is 13 in bcc, we ran the program with $N = 13$, which generated a list of irreducible curves of length at most 13. One then checks that (1) at least one of the length 13 irreducible curves is a knot, and (2) this list does not contain length 12 or less knots. Each of the knots this list can then be visually checked for knot type as a final step.

As a stand alone program running on a single CPU, the outline of the program is as follows:

(1) Start with a polygonal curve of length 1 consisting of the first neighbor vector from $\mathcal{A}_{\mathcal{L}}$.
(2) Add a vector onto the end of the currently formed polygonal curve, starting with the first neighbor vector in $\mathcal{A}_{\mathcal{L}}$.
    (a) If the last neighbor vector added causes the curve to self-intersect, then remove this vector and replace it the next vector in $\mathcal{A}_{\mathcal{L}}$.
        (i) If no more unused vectors are available in $\mathcal{A}_{\mathcal{L}}$, then the last vector (in the curve) is deleted and replaced with the direction in $\mathcal{A}_{\mathcal{L}}$ whose index follows the index of the vector just deleted. This process may cascade for several steps, and if it cascades until all of the vectors are deleted, then the program has completed its search.
    (b) If the distance (measured in steps) from the endpoint of the newly expanded curve to the origin is greater than the difference in the current length of the curve and the maximum length allowed, this means that the curve cannot close itself off without using more than N vectors. So remove the newly added vector and replace it with the next vector in $\mathcal{A}_{\mathcal{L}}$.
        (i) If no more unused vectors are available in $\mathcal{A}_{\mathcal{L}}$, then the last vector is deleted and replaced with next direction in $\mathcal{A}_{\mathcal{L}}$ that comes after the

vector just deleted. This process may cascade for several steps, and if it cascades until all of the vectors are deleted, then the program has completed its search.

(3) If a vector was successfully added in step 2, then check to see if the resulting polygonal curve is closed.

   (a) If the curve is closed, the knot reduction algorithm is applied to that closed curve to detect whether or not it is the unknot. If the algorithm fails to reduce the curve to the unknot, the curve is recorded to disk for later inspection.

      (i) After checking for knottedness, the last vector of the curve is incremented to the next vector in $\mathcal{A}_{\mathcal{L}}$ and tested as in step 2.

   (b) If the curve is not closed, then return to step 2.

This algorithm is called a "worm" algorithm because one can visualize a polygonal worm of initial length 1 that can grow to a maximal length of N. With its tail attached at the origin, it stretches along the lattice, its head searching out all possible ways through the lattice to make it back to its tail without biting itself in two.

Obviously, the number of steps required for this program to exhaustively find and check each closed polygonal curve is exponential in the number of primary directions in the lattice and the maximum length. For this reason, we found that a single desktop computer could not complete the job required to verify the minimum knotting numbers in Theorem 2.1 in a reasonable amount of time. We instead used a computing cluster along with the program *PVM* (*Parallel Virtual Machine*) [6] to break up the big computing job into lots of smaller jobs and distributed those smaller jobs to many computing nodes. To demonstrate our strategy, consider fcc with minimal knotting number 15. We pre-computed the complete list of all closed curves in fcc of length less than or equal to 8. This is a fairly long list, but not astronomically large. Because the program essentially orders the curves, we used this pre-computed list of curves as starting and stopping points for the algorithm. Using *PVM*, a master program which has access to the pre-computed list of length 8 curves starts a certain number of slave programs. The master program then reads the first two lines of the list and sends them to slave program #1, the second and third lines to slave program #2, and so on, until all of the slave programs have a job. Each slave program then processes all closed curves of length less than or equal to 15 "between" the the start and stop values it received. When a slave program has completed its search, it sends a message back to the master that it has completed its job, and the master sends it another job. This goes on until the master has sent out all lines of the list of length 8 curves and has received messages that all jobs have been completed. A cluster of 45 dual processor computing nodes (90 CPUs) reduced jobs that would have taken months or years on a single computer to a week or so.

A web page where the computer code for this program along with the output can be seen at

http://www.math.uttyler.edu/cmann/knotprograms/.

4. Program Results and Postprocessing

For each lattice, the program produced a list of irreducible simple closed lattice curves. Many of these curves were congruent to one another. As a postprocessing step, we checked for some congruences (e.g. cyclic permutation of the vectors defining the knot, or a cyclic permutation of a lattice symmetry transformation of the knot) to reduce the number of knots. In the *sh* lattice we found 496 irreducible knotted curves, which reduced to 124 distinct knotted lattice curves after removing lattice symmetric duplications. In the *fcc* lattice we found 160 irreducible knotted lattice curves, 32 of which were unique after removing lattice symmetric duplications. In the *bcc-8* lattice we found 7128 irreducible knotted curves, which reduced to 132 distinct knotted lattice curves after removing lattice symmetric duplications. Finally, in the *bcc-14* lattice we found 80 irreducible knotted lattice curves, of which 52 remained after removing lattice symmetric duplications. All irreducible curves were knotted and appear to be trefoil knots.

These lists of knots are given in Tables 2–5 (and at the web site given above).

5. Related Open Questions

(1) We define the $\mathcal{L}$-*minimum step number* of a knot type K to be the minimum number of steps required to form a knot of type K in $\mathcal{L}$. Very little is known about minimal step numbers of knot types in lattices other than $\mathbb{Z}^3$. In $\mathbb{Z}^3$, Janse van Rensburg [13] estimated, using simulated annealing, the minimum step number of several knot types with low crossing number. His estimates show that the minimal step number increases less than linearly in the number of crossings. (Preliminary investigations of our own point toward a similar trend in fcc.) Diao and Ernst [3] provide a particular family of torus knots in which the minimal step numbers grow on the order of $n^{3/4}$ where n is the minimal crossing number of a knot. Diao, Ernst and Thistlethwaite [4] establish that the aforementioned growth does not always happen by giving a family of knots in which the minimal step numbers grow linearly with respect to the minimal crossing numbers.

(2) In [10] a relationship between the minimal step number of a knot in $\mathbb{Z}^3$ and the minimal number of cubes required to form a knot of the same type is given: The number of cubes required is twice the number of steps required. Are there similar relationships between knots formed in sh, fcc and bcc-14 and knots formed from hexagonal prisms, rhombic dodecahedra, or truncated octahedra (respectively)?

Table 2. List of irreducible bcc-8 knots using the convention of $0 = (1,1,1)$, $1 = (-1,-1,-1)$, $2 = (-1,1,1)$, $3 = (1,-1,-1)$, $4 = (-1,-1,1)$, $5 = (1,1,-1)$, $6 = (1,-1,1)$, and $7 = (-1,1,-1)$.

002713460057144135	002713460357244135	002713463057244135	002713463507244135
002713463552244135	002713463570244135	002713466557244135	002713566427553144
002713566427571166	002713566427571346	002713566427571364	002713566427571436
002713566427571463	002714135004271363	003314127506631722	003314172506631722
003314175004631722	003314175026631722	003314175206631722	003314177006631722
003314217506631722	003314227536624175	003314227536631722	003314227536641275
003314227536641725	003314227536641752	003314227536641770	003314227536642175
003314227536644775	006315722463557144	006316427550641177	020631127506421135
020631127506441375	020631127506441735	020631127506443175	020631127506447135
020631127506461175	027511360247753646	027511460502711636	027511460551724636
027511460557124636	027511460557214636	027511460557241636	056351724660571142
056357124660571142	056357214660571142	056357744660571142	056361427550641172
056364127550641172	056364217550641172	056364271550641172	056364477550641172
064477133002241355	064477135602241355	064477136502241355	064477153602241355
066117350244635772	066117420533642775	066117500241331720	042775364122053136
042775364420753136	042775364422553136	042775364427053136	042775364472053136
053112200631771460	053112226635771460	053112240635771460	053112246035771460
053112246350771460	053112246655771460	063177146005317420	063177146005371420
063177146005731420	063177204635311220	063177204635314225	063177204635314270
063177204635317420	063177204635371420	063177204635511420	063177204635731420
063177226635311220	063177226635314225	063177226635314270	063177226635317420
063177226635371420	063177226635511420	063177240635311220	063177240635314225
063177240635314270	063177240635317420	063177240635371420	063177240635511420
063177240635731420	063177246035511420	063177246035731420	063377224635077146
063377224635527146	065315724660571142	065317224635527146	072414360577246335
072414603577246335	072414630577246335	072414665577246335	004411350220631175
004411350221463575	004411350224163575	004411350224631575	036424137500641775
036424173500641775	036424713500641775	042471350064177536	057172046335724436
057172406335724436	057174206335724436	006341720053144175	006341720753644175
022175364122053136	022175364400571136	022175364402753136	022175364472053136
022175364602571136	022175364605721136	042753714600271336	050631127206351142
050634177206351142	005371460027133142	053314200271336427	053314200571346427
055724163550241146	055724163605271146	055724613560271146	063642715506413772

Table 3. List of irreducible bcc-14 knots using the convention that $0 = (1,0,0)$, $1 = (-1,0,0)$, $2 = (0,1,0)$, $3 = (0,-1,0)$, $4 = (1,1,1)$, $5 = (-1,-1,-1)$, $6 = (-1,1,1)$, $7 = (1,-1,-1)$, $8 = (-1,-1,1)$, $9 = (1,1,-1)$, $a = (-1,1,1)$, $b = (1,-1,-1)$, $c = (0,0,1)$, and $d = (0,0,-1)$.

4c1d3c42d31d0	40313d24c3d12	4031d1c40d312	42bd3cc0db13a	4c1d3d24c31d0
4c31d042133d2	40311d04c13d2	40311d04c31d2	40311d04c35b2	40311d04c3d12
40311d04c85d2	46137d2cc1d70	4a351200c15d2	4c3d3124031d2	4a31bd0cc3db2
4c1d3042131d0	4c1d3042133d2	40751c42d3312	4c3dd1c4031d2	4c3dd1c403d12
4c3dd1c407512	4c3dd1c40d312	4c3d1c40d13d2	4c3d1d04c13d2	40d58c42d3312
42133d24c13d0	42133d24c1570	42133d24c85d0	4c3d1304213d2	40d13c42d3312
421d3c40d1130	421d3c42d3130	461d73c221370	4213d1c40d130	4c1dd3c4213d0
4c1dd3c421570	4c1dd3c421d30	4c1dd3c42d130	42d31c40d1130	4213d04c1d130
4031d04c1d312	40d31c42d3312	407d1cc2d7316	40d53c2203516	4213d3c42d130
42d58c40d1130	46153022c35d0	42d13c40d1130	42b53c40d1130	403d1c40d1312
403d1c42d3312	4031d3c42d312			

Table 4. List of irreducible sh knots using the convention that $0 = (0,0,1)$, $1 = (0,0,-1)$, $2 = (1,0,0)$, $3 = (-1,0,0)$, $4 = (1/2,\sqrt{3}/2,0)$, $5 = (-1/2,-\sqrt{3}/2,0)$, $6 = (1/2,-\sqrt{3}/2,0)$, and $7 = (-1/2,\sqrt{3}/2,0)$.

02611550047142653374	02611550074142653374	02611560077142653374	02621133000241155374
02621133000421155374	02651135004412653774	02653771426500331124	02653771426500351144
02653771426500531144	02477356212400331156	02477356214200331156	02477356214400351156
02477356214400531156	02653117700621653744	02653177426503311240	02653177426503311420
02653177426503511440	02653774126500331124	02653774126500331142	02653774126500351144
02653774126500531144	02655374212600331174	02655374216200331174	02655374216600371174
02655374216600731174	02473556124700331126	02473556124700331162	02473556124700731166
02421133000261177356	02421133000621177356	02611770005611447356	02611770006511447356
02611770035612447356	02611770035621447356	02611770356242113300	02611770356244113500
02611770356244115300	02611770356244173560	02265137440056111370	02265137440056111730
02265137440065111730	02265137440066111370	02265137470066111370	02265137470066111730
02426153770026111330	02426153770062111330	02611147005516247330	02611147005561247330
02611147005562147330	02611147005562417330	02611147005562471330	02611153004412653770
02611153004421653770	02611153004426513770	02611153004426531770	02411330056214773562
02426531770026111330	02426531770062111330	02653311240005511474	02653311240005611774
02653311420005511474	02653311420005611774	02653311420006511774	02411740055162473356
02611330074216553742	02611773056242113300	02611773056244113500	02611773056244115300
02216537440056111370	02216537440265311370	02216537442605311370	02216537470066111370
02611133002412653770	02611133002421653770	02611133002426513770	02611135004426513770
02611135004426531770	02247315660047111350	02247315660047111530	02247315660074111530
02265317440056111730	02265317440056111730	02411550004711665374	02411550007411665374
02411550037412665374	02411133002612473550	02411133002624173550	02411133002624713550
02411133002624731550	02214735650044111350	02214735650044111530	02214735660047111350
02214735660047111530	02214735660074111530	02214735662407311350	02473711660004711556
02211137006621473530	02211137006624173530	02211173006621473530	02211173006624173530
02426513770062111330	02411770056162473356	02411770065162473356	02421653770062111330
02211350374266117300	02624173550042111330	02473551142000331126	02611653074426531770
02611653704426531770	02624731550042111330	02211370005611447356	02211370006511447356
02211370035612447356	02211370035621447356	02473311260004711556	02211350004711665374
02211350037412665374	02411137006624173550	02411137006624731550	02655114700062113374
02655117400026113374	02211530007411665374	02211530037412665374	02624713550042111330

Table 5. List of irreducible fcc knots using the convention that $0 = (1,0,0)$, $1 = (-1,0,0)$, $2 = (0,1,0)$, $3 = (0,-1,0)$, $4 = (1/2,1/2,1/\sqrt{2})$, $5 = (-1/2,-1/2,-1/\sqrt{2})$, $6 = (-1/2,1/2,1/\sqrt{2})$, $7 = (1/2,-1/2,-1/\sqrt{2})$, $8 = (-1/2,-1/2,1/\sqrt{2})$, $9 = (1/2,1/2,-1/\sqrt{2})$, $a = (1/2,-1/2,1/\sqrt{2})$, $b = (-1/2,1/2,-1/\sqrt{2})$.

407b168a792b13a	407b168a792b58a	407b1683092b13a	407b1683092b58a	4095168a792b13a
4095168a792b58a	42153702681590a	4618a79b2483579	4618a79b26a3579	4a83592640351b9
4a8359264a751b9	4a85b2903861590	4a85b2903861b70	42153794681590a	4a38129704815b9
4a3812970a615b9	42613790481b530	42613790a61b530	4815b94a386b970	4a751b94a837b26
4a837b2640351b9	4a07b1862073512	4a0951862073512	42685790481b530	42685790a61b530
48375b640a85b92	483751240a31b92	483751240a85b92	40a31b926a37512	429b58a04215738
4815b20a386b970	421573a629b58a0			

(3) Changing the definition of neighbors so that two points are neighbors if their Voronöi cells intersect (rather than just sharing a facet as before) results in some interesting open questions. For example, in the cubic lattice there would be 26 neighbors around each lattice point (think of a Rubik's cube). What

would the minimal knotting number be in this lattice? What are the distance functions in these lattices? Open questions 1 and 2 would apply as well.

Acknowledgments

The authors would like to thank the referees for their helpful comments. The authors would like to thank UT Austin for access to their computing cluster which allowed us to process the computational results of this paper; this cluster was supported by NSF grant #0322962, PI Jack Xin.

References

[1] J. H. Conway and N. J. A. Sloane, *Sphere Packing, Lattices, and Groups* (Springer Verlag, New York, NY, 1988).
[2] Y. Diao, Minimal knotted polygons on the cubic lattice, *J. Knot Theory Ramifications* **2** (1993) 413–425.
[3] Y. Diao and C. Ernst, The complexity of lattice knots, *Topol. Appl.* **90**(1–3) (1998) 1–9.
[4] Y. Diao, C. Ernst and M. Thistlethwaite, The linear growth in the lengths of a family of thick knots, *J. Knot Theory Ramifications* **12** (2003) 709–715.
[5] E. S. Fedorov, *Nachala Ucheniya o Figurakh* (Notices of the Imperial St. Petersburg Mineralogical Society, Russia, 1885).
[6] A. Geist *et al.*, *PVM-Parallel Virtual Machine: A Users' Guide and Tutorial for Networked Parallel Computing* (MIT Press, Cambridge, MA, 1994).
[7] R. B. Kusner and J. M. Sullivan, Möbius-invariant knot energies, in *Ideal Knots*, eds. A. Stasiak, V. Katrich and L. H. Kauffman (World Scientific, Singapore, 1998), pp. 315–352.
[8] J. Lagarias, Point lattices, in *Handbook of Combinatorics, Volume 1*, eds. R. L. Graham, M. Grotschel and L. Lovasz (Elsevier Science B. V., Amsterdam, 1995), pp. 919–966.
[9] E. Luczak and A. Rosenfeld, Distance on a hexagonal grid, *IEEE Trans. Computers* **25** (1976) 532–533.
[10] C. Mann and J. McLoud-Mann, On the relationship between minimal lattice knots and minimal cube knots, *J. Knot Theory Ramifications* **14**(7) (2005) 841–851.
[11] C. Mann, B. McCarty, J. McLoud-Mann, R. Ranalli and N. Smith, Metrics in three-dimensional lattices, *J. Geom. Graphics* **12**(2) (2008) 133–140.
[12] A. McAndrew and C. Osborn, Metrics on the face-centered cubic grid, in *SPIE*, in *Proceedings of Vision Geometry*, eds. R. A. Melter and A. Y. Wu (1993), pp. 49–60.
[13] E. J. Janse van Rensburg, Lattice invariants for knots, *Inst. Math. Appl.* **82** (1996) 11–20.
[14] E. J. Janse van Rensburg, Minimal lattice knots, in *Ideal Knots*, eds. A. Stasiak, V. Katrich and L. H. Kauffman (World Scientific, Singapore, 1998), pp. 88–106.
[15] E. J. Janse van Rensburg, Minimal knots in the cubic lattice, *J. Knot Theory Ramifications* **4** (1995) 115–130.
[16] E. J. Janse van Rensburg and S. G. Whittington, The knot probability in lattice polygons, *J. Phys. A Math. Gen.* **23** (1990) 3573–3590.
[17] M. Jaswon, *An Introduction to Mathematical Crystallography* (American Elsevier Pub. Co., 1965).
[18] M. Senechal, *Quasicrystals and Geometry* (Cambridge University Press, New York, 1995).