International Baccalaureate Diploma Program

Determining the Method of Playing a Chord Progression on the Guitar that Requires the Minimum Amount of Movement

Mathematics HL

Internal Assessment

Keith Choa

May 2018

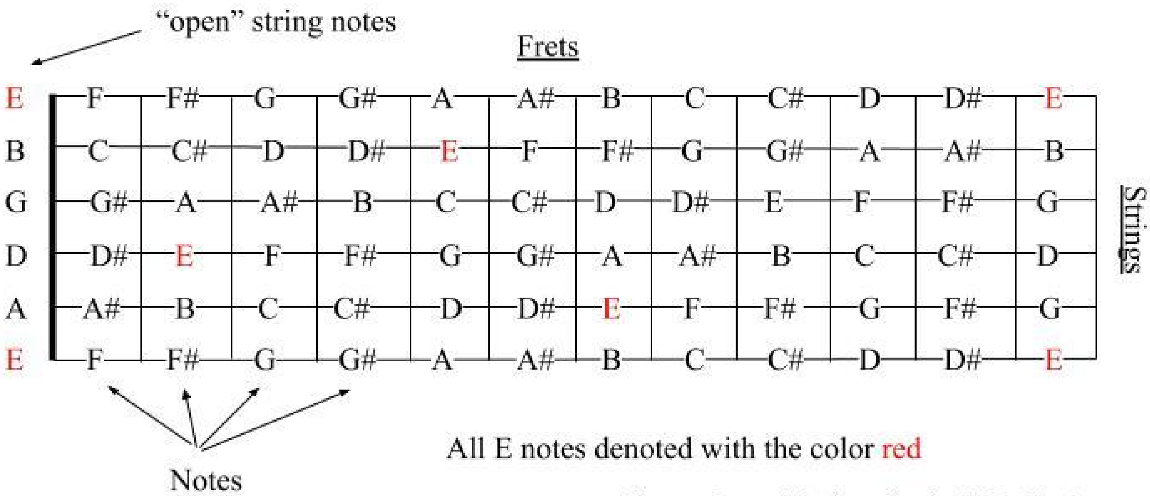
Abstract: The aim of this exploration is to determine the most efficient way to move between distances by minimizing the distance needed to travel. In particular, this will tackle distances traveled by fingers on a fretboard on a guitar. This will in turn lead to minimizing the distance needed to travel between chords. By minimizing that action for a particular song, that particular song may be played more efficiently. This interests me because as an amateur guitarist, I want to find the most efficient way to play a particular piece of music and, in that way, play it the best way I can.

1 Introduction

Learning to play the guitar is a potentially difficult and confusing process. Having played for over 5 years now, I understand that the main struggle for aspiring guitarists is usually in learning and playing chords. The difficulty in playing and learning chords arises not only from having to memorize them, but also from having to know exactly where to place your fingers, and how to quickly transition between the chords that have been memorized. This can all be very challenging for beginner guitarists, as the amount of practice needed to pull this off can be enough to discourage anyone from continuing on their musical journey.

1.1 Background of the Study

Figure 1.1: Notes on a guitar fretboard



Simply put, chords are a collection of three notes played in unison. Specific notes in combination make different chords (Konecky, Larry). When these notes are strummed on the guitar, they produce a sound corresponding to that specific chord. When specific chords are then strummed in sequence, it produces a song. However, when looking at a guitar fretboard (as shown in Figure 1.1), the same note can be found on multiple points throughout. Since a chord is simply made by an arrangement of specific notes, this means that the same chord can technically have multiple positions throughout the fretboard (as long as the same notes are being strummed).

This brings up the difficulty in playing chords. For each chord ‘position’ throughout the fretboard, there will be different finger positionings (where the fingers need to be to play the chord). Thus, when shifting between chords in a song, the player must be able to know exactly what shape to put their fingers in and how far to move their hands up and down the fretboard. All this must be done in under a second to make the transition seamless. An additional point of interest is that depending on the player, one may also find fret-based movements (finger movements that stay along the same strings, but switches frets) more difficult than string-based movements (finger movements that stay along the same frets, but switches strings), or vice versa. This preference can be due to a variety of factors, including the size of their hand, their experience, etc.

1.2 Statement of the Research Problem

Since the area of concern for most guitar players are the movement of the fingers in switching between chords, the most natural solution to this would be to minimize the distance needed for each finger to travel. This will make guitar playing much more efficient. To make this task more relevant to a beginning guitar player, it is important to apply this to a song the musician can relate to. In this way, it is important to note that the majority of pop songs in the recent decades have a common chord progression of E, B, C#m, A (Caudill, Heather). Minimizing the distance needed for fingers to travel between chords within that chord progression will then be very beneficial for beginning guitar players, as they will then be able to play a wide range of songs. Thus, this paper aims to answer the following question:

1. What combination of guitar chord positions will minimize the distance needed for each finger to travel between frets in the chord progression of E, B, C#m and A?

1.3 Significance of the Study

This study will not only tackle the difficulties faced by guitarists in learning the instrument, it will do so using one of the most popular chord progressions in modern culture. Meaning, the results of this exploration will be able to aid guitar players in being able to easily play many of the songs they already know of.

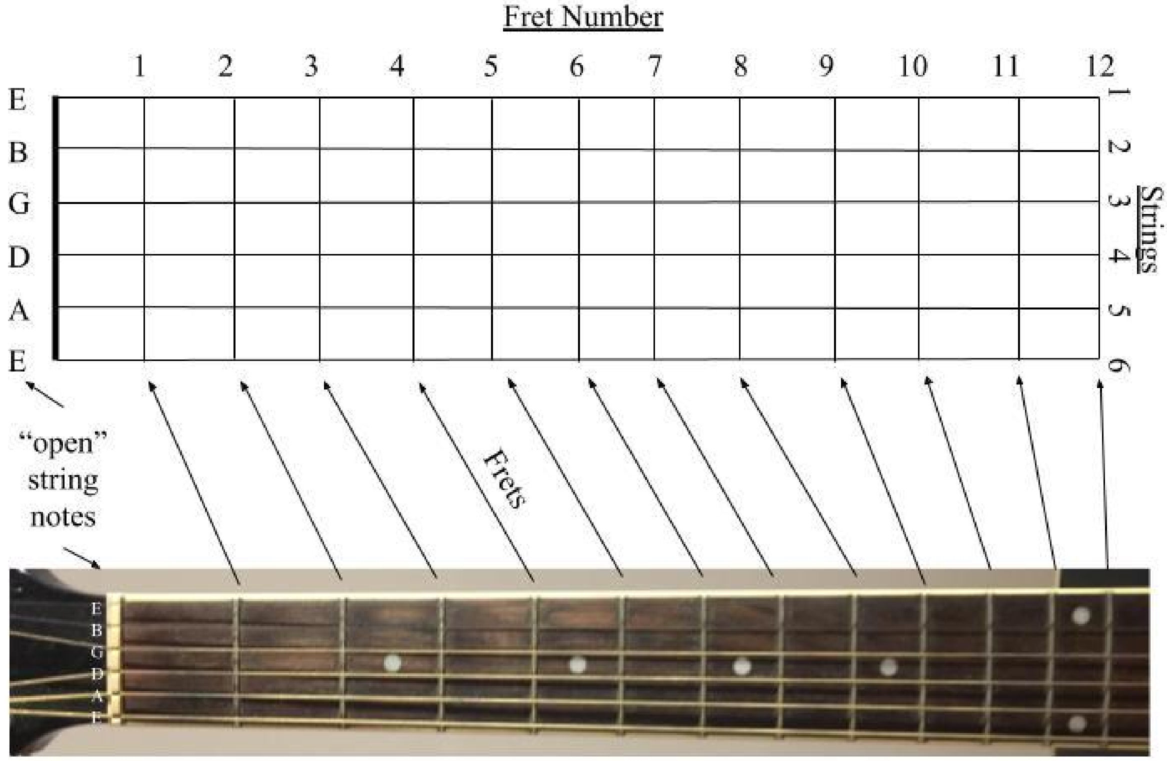
2 Review of Related Literature

To understand specifically how chords are structured, one needs to understand the structure of the guitar, how a chord is formed, how to read tablature, and fingers needed to play it.

The Fretboard

On a fretboard, the strings are numbered 1-6, starting from the topmost E string. The frets, on the other hand, are numbered 1-12 starting from the first fret. When no frets are being pressed, the string is called “open” (Newbold, Ben).

Figure 2.1: Diagram showing the mapping out of a fretboard



(diagram is an original creation by Keith Choa)

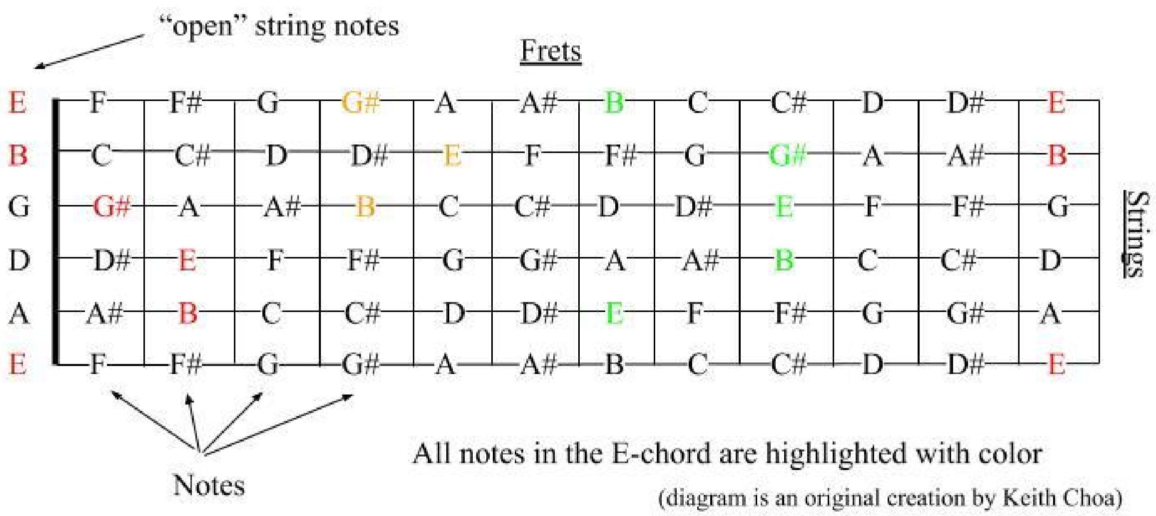
Chord Structure

As mentioned earlier, chords are comprised of a specific set of notes. Generally, these sets of notes are called triads (with the exception of C#m). The reasoning behind why these particular notes create a particular chord is not relevant to this exploration, and will be excluded. The notes for each chord is given as (Davies, Scott):

- E-Chord Notes: E, G#, B
- B-Chord Notes: B, D#, F#
- C#m-Chord Notes: C#, E, G#
- A-Chord Notes: A, C#, E

Again, as mentioned earlier, as long as all the notes in their respective chord are being played, then it will produce a sound corresponding to it’s chord. So, in Figure 2.2, all E, G# and B notes (all notes of the E chord) have been highlighted. For ease of viewing, notes that are in relatively close proximity to each other are colored in the same color (this will be relevant later). Thus, as long as any of the highlighted frets are being pressed down on while strumming, an E chord will be played.

Figure 2.2: Notes on a guitar fretboard



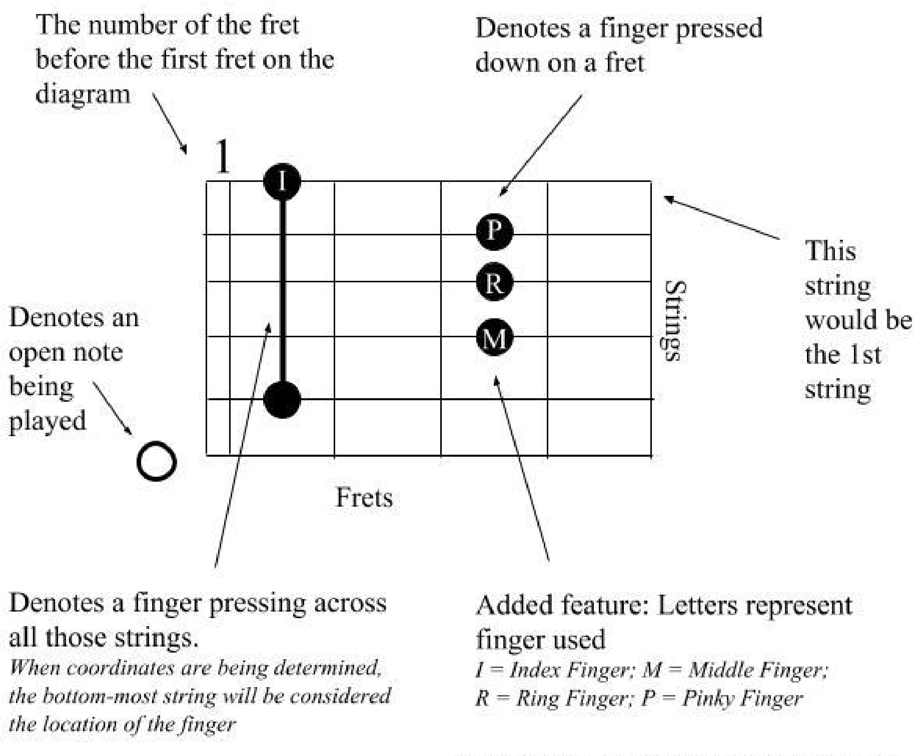
(diagram is an original creation by Keith Choa)

This process can be done for all chords, and will reveal all of the chord positions.

Tablature

The diagram shown in Figure 2.2 is quite large and would be impractical to use when demonstrating how to play multiple different chords. So, guitarists use an alternate method to notate how to play a specific chords. This is called *tablature* or *tab* (Newbold, Ben). Tablature is a method of noting down music for fretted instruments, such as the bass guitar, guitar, ukelele, etc. ("How to Read Guitar Tabs."). Figure 2.3 shows the components of a proper tablature. Often, a rotated version of the diagram in Figure 2.3 is used, however, the concepts stay the same.

Figure 2.3: Tablature form of the E-chord



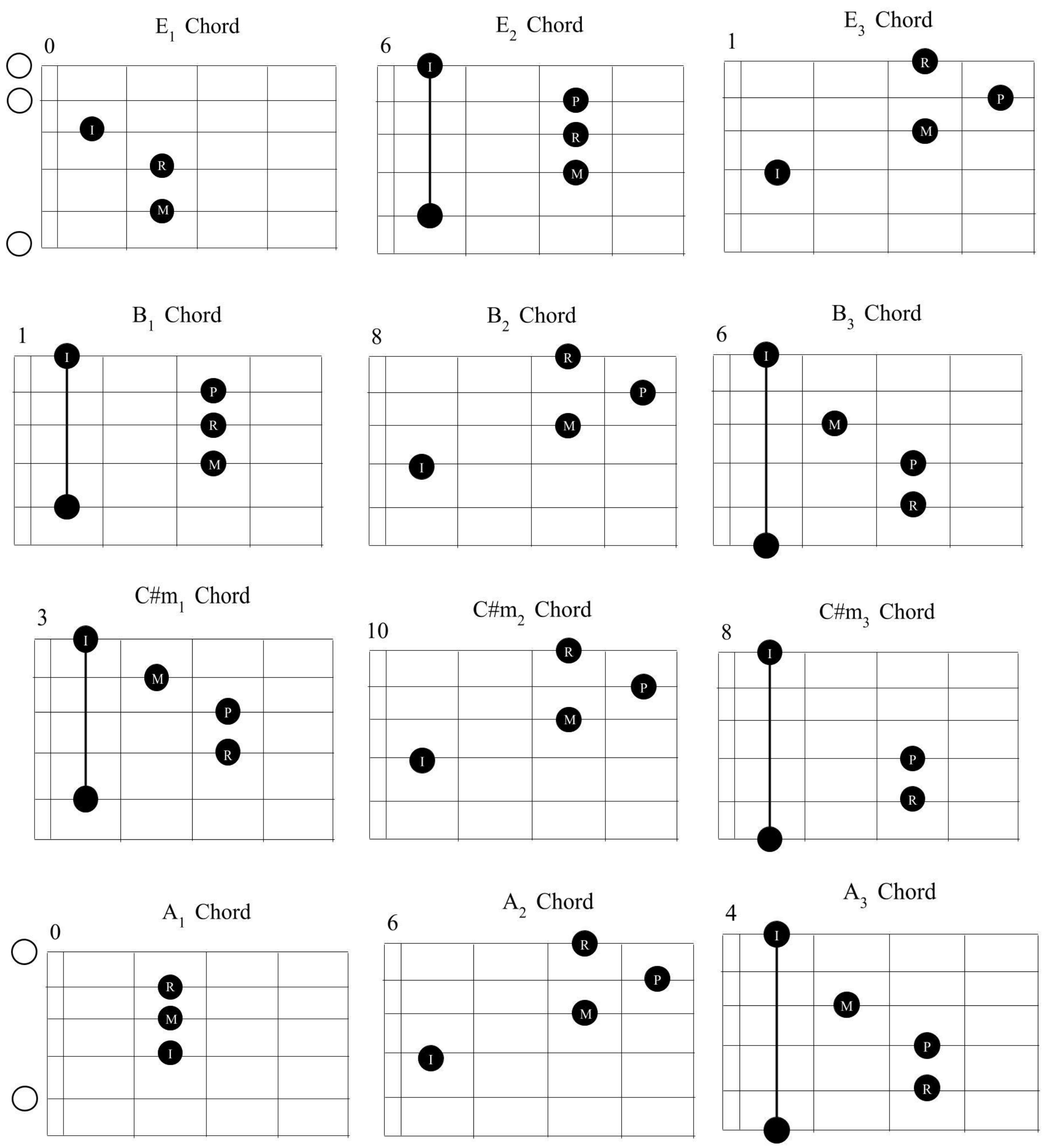
(diagram is an original creation by Keith Choa)

An added feature which is not typically shown in tablature, but will be used in this exploration, is the use of designated fingers for each 'pressed-down' position. This restricts certain chord positions such that there will only be one given way to play the particular chord, and only one will be taken into consideration. Each finger will be represented with a letter (I = Index Finger, M = Middle Finger, R = Ring Finger, and P = Pinky Finger) Later, this will be relevant why this has been restricted.

Playing Chords

While there are technically numerous possible positions for a specific chord, as described earlier, this exploration will limit itself to the three most common positions to play these chords. These are all shown in Figure 2.4. Each different position will be denoted with a subscript of either 1, 2, and 3.

Figure 2.4: All considered chord positions in the chord progression E, B, C#m, A



(diagrams are original creations by Keith Choa)

When these chords are played in progression (in this case, E, B, C#m, A), they are typically played cyclically. This means that after the A chord is played, the E chord is played again. This is repeated until the song ends.

3 Research Framework

Computing Distance

In order to determine the distance between finger positions within a chord, the fret number and string number need to be set as two separate variables. The fret number is considered an x-coordinate and the string number is considered a y coordinate. Therefore, the restrictions of the x and y coordinate should be:

$$\{x \in Z | 1 \leq x \leq 12\}$$
$$\{y \in Z | 1 \leq y \leq 6\}$$

All finger positions will be noted as coordinates with (x, y). Then, distances travelled between each finger will be determined with the formula (Osikiewicz, Beth-Allyn):

$$Distance = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2}$$

Where 'f' denotes the value of that variable for the next chord in sequence, and 'i' denotes the value of that variable for the current (or initial) chord. However, in order to take into consideration the preference between fret-based finger movements and string-based finger movements, a weighted system will be introduced. The modified formula will be:

$$Weighted\ Distance = \sqrt{F(x_f - x_i)^2 + S(y_f - y_i)^2}$$

Such that F is the fret preference multiplier, and S is the string preference multiplier. Increasing the value of S will make it such that string based movements will be considered larger than their actual value. By adjusting the relationship between F and S (their ratio), specific preferences can be considered. The first scenario considered will be a situation where F and S both have a weight of 1, and so are equally considered. On the other hand, the variable n will represent the ratio between the S weight and the F weight for the following unequal weight scenarios such that:

$$n = \frac{S}{F}$$

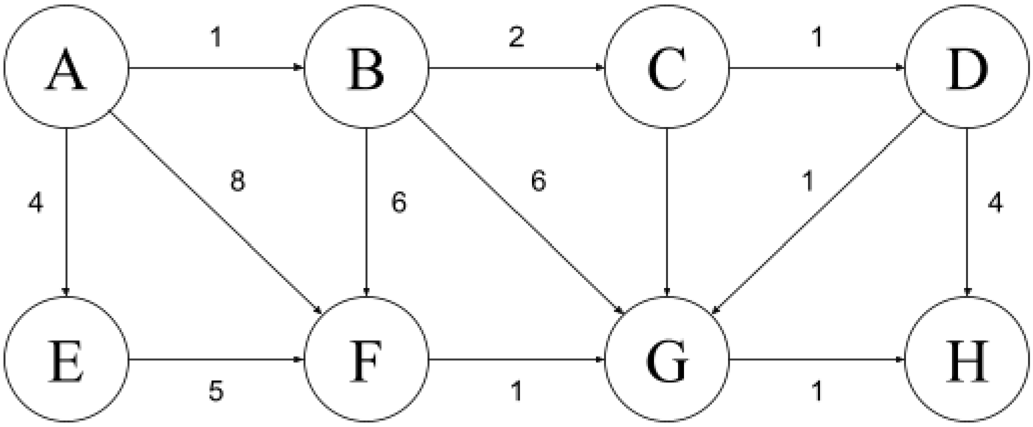
and the n values tested will be: 0.2, 0.25, 0.33, 0.5, 2, 3, 4, 5.

Once all finger movement distances have been calculated, the sum of all distances (Index (I), Middle (M), Ring (R), and Pinky (P)) will be used to represent the total distance travelled between chord positions. This value will be then used in Dijkstra's algorithm to determine the path requiring the least amount of movements.

Dijkstra's Algorithm

Dijkstra's algorithm, as the name suggests, is an algorithm developed by Edsger Wybe Dijkstra that aims to find the shortest distance from a single point to all points on a weighted graph (Yan, Melissa, and Thomas Rothvoß). A weighted graph has both vertices and edges. The vertices represent destinations, and the edges represent the distances travelled. A sample mapping of which can be found below.

Figure 3.1: Demonstration of a weighted map used for reference in Dijkstra's algorithm



(diagram is an original creation by Keith Choa)

Note that all edges are nonnegative. Dijkstra's algorithm starts at a point, and finds a neighboring vertex with the shortest distance from it (all vertices not directly connected are considered a distance of infinity). Then, from this point, the shortest distance is again determined until the final vertex or destination is reached.

More specifically, the process of Dijkstra's involve the following steps:

1. Create a map of nodes (vertices) and, when applicable, connect each node.
2. Pick a starting node.
3. Compute the distances from the selected node to adjacent nodes.
4. Select the node with minimal distance from the starting node.
5. Repeat adjacent node distance calculations to unvisited nodes.
6. Select again the adjacent node (unvisited) with minimal distance.
7. Repeat steps 3-6 until all nodes have been visited.

(obtained from (Abiy, Thaddeus))

This particular set of instructions demonstrates the iterative nature of Dijkstra's algorithm. This is something that is very important to note, as Dijkstra's algorithm finds the shortest distance to all points on the map. It does so by always choosing the shortest distance out of the entire set of distances calculated. Looking at Figure 3.1, this means that even though the path A → F may have been initially set at 8, the path of A → B would be set at 1, and then A → B → F would be set as 7. Since the path of A → F via B has a smaller distance, that is the value that would be recorded.

Contextualizing the Solution

An additional step to make the results more understandable, would be to take the optimal path, and use the diagrams shown back in Figure 2.4 to give a more understandable instruction on how to play the progression. In general, after obtaining the solution using Dijkstra's algorithm, the corresponding chord will be determined and played for a practical test.

4 Results and Analysis

4.1 Preliminaries

Breaking down the fret and string to create the coordinates of each finger per way of playing a chord, the following table would be used as reference to compute the corresponding distances from one chord to the other.

Table 4.1. Fret and string coordinates for each finger within each chord form

Chord	E			B			C#m			A		
Form	1	2	3	1	2	3	1	2	3	1	2	3
Index	(1, 3)	(7, 5)	(2, 4)	(2, 5)	(9, 4)	(7, 6)	(4, 5)	(11, 4)	(9, 6)	(2, 4)	(7, 4)	(5, 6)
Middle	(2, 5)	(9, 4)	(4, 3)	(4, 4)	(11, 3)	(8, 3)	(5, 2)	(12, 1)		(2, 3)	(9, 3)	(6, 3)
Ring	(2, 4)	(9, 3)	(4, 1)	(4, 3)	(11, 1)	(9, 5)	(6, 4)	(13, 3)	(11, 5)	(2, 2)	(9, 1)	(7, 5)
Pinky		(9, 2)	(5, 2)	(4, 2)	(12, 2)	(9, 4)	(6, 3)	(14, 2)	(11, 4)		(10, 2)	(7, 4)

Since the progression sought to be optimized follows

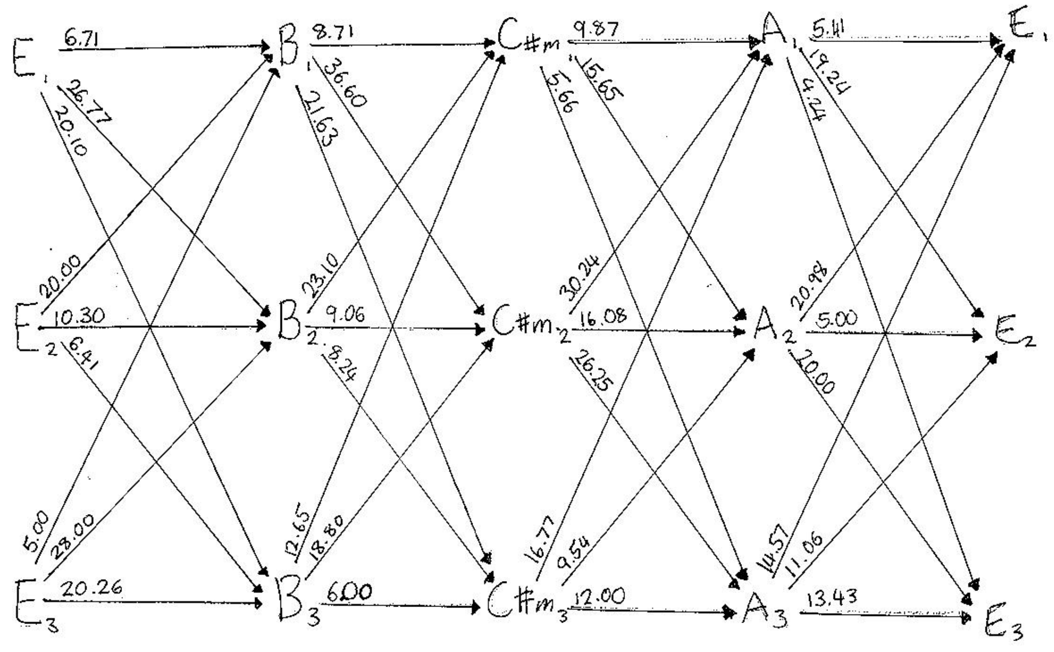
$$E \rightarrow B \rightarrow C\#_m \rightarrow A,$$

the distances from each way of playing chord E to each way of chord B would be computed. This would be done for each pair of consecutive chords until A. Note that the distances from non-consecutive chords such as E to C#m will not need to be computed since they are not part of the progression. Furthermore, the distance from A chord variations to E chord variation also needs to be computed, because of the aforementioned cyclical nature of chord progressions in songs.

4.2 Constructing Dijkstra's Mapping of the Progression

After the relevant distances are computed, the distances are then represented within a weighted graph. The weighted graph is created such that vertices represent variations of playing a particular chord, and the edges represent the distances. Arrows will also be placed on the edges, pointing from the origin chord to the destination chord. This will signify that within the chord progression, fingers can only travel from one particular chord to the next chord. This represents the fact that for a chord progression to make sense, the chords need to be played in that particular order. For instance, the B chord is meant to be played after the E chord. However, if the E chord were to be played after the B chord, that would not be a part of the chord progression.

Figure 4.2: A weighted map showing the distances between each position of a chord in the sequence of E, B, C#m, and A



(diagram is an original creation by Keith Choa)

Note that from the diagram, the E vertices are repeated after the A vertices, as per the cyclical nature of chord progressions mentioned earlier. After this, using the above mapping, Dijkstra's algorithm can now be applied to determine the "path" or way of playing the progression.

4.3 Applying Dijkstra's Algorithm to the Mapping

Dijkstra's algorithm is applied manually, using a table format. The aim in this is to find the shortest distance from a variation of the E chord (E_1, E_2, or E_3) to the same variation of the E chord on the other side of the diagram. This represents an iteration of the chord progression. For the first case (where fret preference multiplier and string preference multipliers are equal), the progression starts with E_1. The distance from E_1 to all vertices is at first considered infinity (∞). Looking at the map in Figure 4.2, there are three edges coming off of E_1. Since these numbers are smaller than infinity, these are noted down as the distances to those vertices (6.71 to B_1, 26.77 to B_2, and 20.10 to B_3). Out of all of these distances, the shortest distance is chosen as the next step in the progression. The distance to that vertice is also highlighted in yellow to show that the distance noted down is the shortest possible distance to that point from E_1.

The next 'step' is signified by a new row in the table. In this case, it would be to B_1. There are three more edges coming out of the B_1 vertice. These are C#m_1, C#m_2, and C#m_3. The distance from E_1 to these vertices is then noted as the sum of the distance between E_1 to B_1 and the distance between B_1 to each specified vertice. Thus, the distance from E_1 to C#m_1 via B_1 is 6.71+8.71 (15.42), the distance from E_1 to C#m_2 via B_1 is 6.71+36.60 (43.31), and the distance from E_1 to C#m_3 via B_1 is 6.71+21.63 (28.34). Out of all these values (including the values for distances between E_1 and B_1, B_2 and B_3), the smallest value is then chosen as the next step.

This would, again, be shown through the addition of another row. In this case, the shortest distance out of all distances is the distance from E_1 to $C\#m_1$. The $C\#m_1$ cells are also highlighted in yellow to, again, signify that the value noted there is the shortest possible distance to $C\#m_1$ from E_1. From $C\#m_1$, distances are added again to find the distance from E_1 to A_1, A_2 and A_3, via $C\#m_1$. Once these values have been added up, the smallest distance out of all those calculated (including from E_1 to B_1, B_2, B_3, $C\#m_1$, etc.) will be used in the next step. In this case, that would be B_3, with a distance value of 20.10. Thus, B_3 is highlighted in yellow and is added to a new row. Then, all distances to $C\#m_1$, $C\#m_2$ and $C\#m_3$ via B_3 are summed up. The key point here is that if the values are higher than those values already in the cell (currently, the distance to $C\#m_1$, $C\#m_2$, and $C\#m_3$ via B_1), then they are ignored. If they are smaller, then they are replaced by the shorter distance.

This process is repeated until the shortest distance to E_1 has been calculated (when the E_1 cell is highlighted yellow). An example of a full processed table is shown below:

Table 4.3: Application of Dijkstra's algorithm for the equal-weight case starting from chord E_1.

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_1
E_1	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	43.31 $B_1 \to C\#m_2$	28.34 $B_1 \to C\#m_3$	∞	∞	∞	∞
$C\#m_1$	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	43.31 $B_1 \to C\#m_2$	28.34 $B_1 \to C\#m_3$	25.29 $C\#m_1 \to A_1$	31.07 $C\#m_1 \to A_2$	21.08 $C\#m_1 \to A_3$	∞
B_3	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	38.90 $B_3 \to C\#m_2$	26.10 $B_3 \to C\#m_1$	25.29 $C\#m_1 \to A_1$	31.07 $C\#m_1 \to A_2$	21.08 $C\#m_1 \to A_3$	∞
A_3	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	38.90 $B_3 \to C\#m_2$	26.10 $B_3 \to C\#m_1$	25.29 $C\#m_1 \to A_1$	31.07 $C\#m_1 \to A_2$	21.08 $C\#m_1 \to A_3$	35.65 $A_3 \to E_1$
A_1	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	38.90 $B_3 \to C\#m_2$	26.10 $B_3 \to C\#m_1$	25.29 $C\#m_1 \to A_1$	31.07 $C\#m_1 \to A_2$	21.08 $C\#m_1 \to A_3$	30.70 $A_1 \to E_1$
$C\#m_3$	6.71 $E_1 \to B_1$	26.77 $E_1 \to B_2$	20.10 $E_1 \to B_3$	15.42 $B_1 \to C\#m_1$	38.90 $B_3 \to C\#m_2$	26.10 $B_3 \to C\#m_1$	25.29 $C\#m_1 \to A_1$	31.07 $C\#m_1 \to A_2$	21.08 $C\#m_1 \to A_3$	30.70 $A_1 \to E_1$

Note that this table represents the optimal progression originating from the E_1 position. Therefore, vertices E_2, and E_3 are not included, because the concept of switching between chord positions between iterations/cycles of a chord progression within a song is not a considered outcome (for example, there will not be a situation where the progression will be

$$E_1 \to B_1 \to C\#m_1 \to A_1 \to E_2,$$

because the progression would not be able to be played cyclically).

Furthermore, since numbers within the cells represent the added distance between vertices, it will be hereby referred to as the 'accumulated distance'. The same process will be applied to the different cases considered in this study, varying the value of n (the ratio of the string-based preference and fret-based preference) when computing distances.

4.4 Results

4.4.1 Optimal distance given no preference towards fret and string-based movements

Interpreting the result from above shows that (highlighted in yellow), the optimal route to E_1 comes from A_1, the optimal route to A_1 is from $C\#m_1$, the optimal route from $C\#m_1$ is from B_1 and the optimal route from B_1 is E_1. Thus, the optimal route, with an accumulated distance of 30.70, is

$$E_1 \rightarrow B_1 \rightarrow C\#m_1 \rightarrow A_1.$$

As stated earlier, the denotation of the number 1 in a specific chord position represents the position of that chord requiring the lowest fret numbers. This is otherwise coincidentally the most common method of playing that particular chord. This calculation shows, then, that the most optimal way of playing this particular chord progression, starting from E_1, involves using the most commonly used positions of the chords. However, this particular calculation only considers the optimum chord progression originating from the E_1 position.

If the chord progression were to originate from E_2, the sequence would be different (see Appendix 1). The chord progression in this case would be

$$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_2,$$

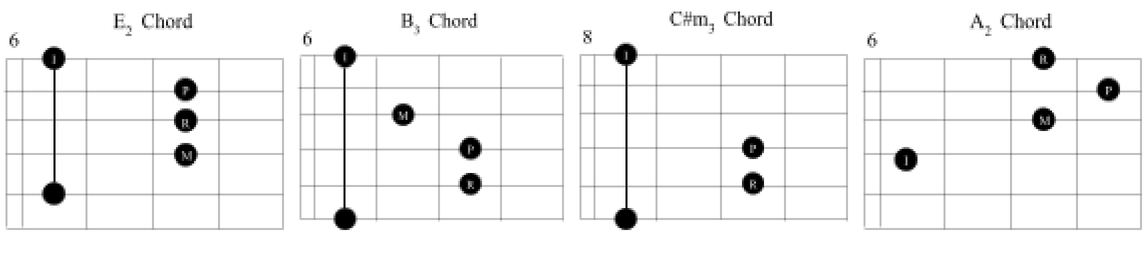
with an accumulated distance of 23.95. A possible complication in this may be the fact that the A_2 chord is considered more difficult to play. From the E_3 position, the chord progression would be

$$E_3 \rightarrow B_1 \rightarrow C\#m_1 \rightarrow A_1.$$

This result has an accumulated distance of 27.82 (see Appendix 2).

Thus, if the progression with the shortest accumulated distance is considered the most optimal progression, this would be the progression of $E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_2$, with an accumulated distance of 23.95. To place this in a more musician-friendly format, the tablature is shown as:

Figure 4.4.1: Tablature form of the sequence of chord positions requiring the least amount of distance traveled by fingers



(diagrams are original creations by Keith Choa)

4.4.2 Optimal distance with varied preferences towards fret and string-based movements

As mentioned in the research framework, various guitar players have different preferences towards string-based movements and fret-based movements, and this is captured by n. Moreover, a value of $n > 1$ would represent a preference towards fret-based movements, and a value of $n < 1$

would represent a preference towards string-based movements. Table 4.4.2.1 summarizes the results of Dijkstra's algorithm given different values of n (refer to Appendices 3-11 for the computations).

Table 4.4.2.1: Optimal Progression for varying values of n

Starting Chord	Weight: $n = \frac{S}{F}$	Optimal Progression
E_1	$0.2 \leq n < 1$	$E_1 \rightarrow B_1 \rightarrow C\#m_1 \rightarrow A_1$
	$1 \leq n < 4$	
	$4 \leq n \leq 5$	$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$
E_2	$0.2 \leq n < 1$	$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_2$
	$1 \leq n < 2$	
	$2 \leq n \leq 5$	$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$
E_3	$0.2 \leq n < 1$	$E_3 \rightarrow B_1 \rightarrow C\#m_1 \rightarrow A_1$
	$1 < n \leq 5$	

Movement Preference: String-based Fret-based

Varying the value of n for the starting chord E_1 shows no difference in the optimal progression unless fret-based movements are strongly preferred ($4 \leq n \leq 5$). On the other hand, for E_2 only requires a value of n = 2 in order for the optimal progression to change. However, for E_3, any change in the n value from 0.2 to 5 shows no change in the optimal progression. From this, it can be observed that the impact of the preference on string-based and fret-based movements depends on the chosen starting chord. The significance varies due to the positioning of the fingers to play different chords.

Coincidentally, the optimal progression for string-based preference movements follows the same optimal progression presented in the no preference case for all starting chords in the table above. Thus, for guitar players who have a preference for string-based movements, the "no-preference" optimal progression can be used instead. The optimal progression will only vary if there is a certain level of preference to fret-based movements.

4.4.3 Using the optimal chord progressions

To further test the significance of the obtained progressions in the study, guitarists with different experiences were asked to determine the most comfortable chord progression for them. The concept of the most comfortable chord progression lines up with the concept of the most efficient progression. The results of the survey (see Appendix 12) showed that there was a strong preference to the starting chord of E_1 and less so for E_2 and E_3. The most common preferred progression chosen was:

$$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$$

Furthermore, certain guitarists claiming a preference towards string-based movements chose the optimal progression obtained specifically for the fret-based movement preference after evaluating all of the choices. This shows that, for the given sample of guitarists, the progression of:

$$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$$

is, overall, the easiest progression to play, regardless of a preference towards string-based or fret-based movements.

5 Conclusion and Recommendations

5.1 Summary

This research paper aimed to aid beginner guitarists by minimizing the distances needed for fingers to travel in the chord progression E, B, C#m, A. Three possible sets of finger positions are then set for each chord, and are considered different chord positions. The finger position in each chord position is set as a fret and string coordinate pair. Then, the distances between finger positions in adjacent chords within the chord progression are calculated using a weighted distance formula. From this, all distances are summed up and placed in a weighted map as edge weights. Then, Dijkstra's algorithm is applied to find the shortest path connecting a variation of the E chord to itself on the opposite end of the map. This is done for all three variations of the E chord, and the shortest path of all of these (the one with the smallest value for distance), is considered the optimal path. This is then translated into tablature for a guitarist to read.

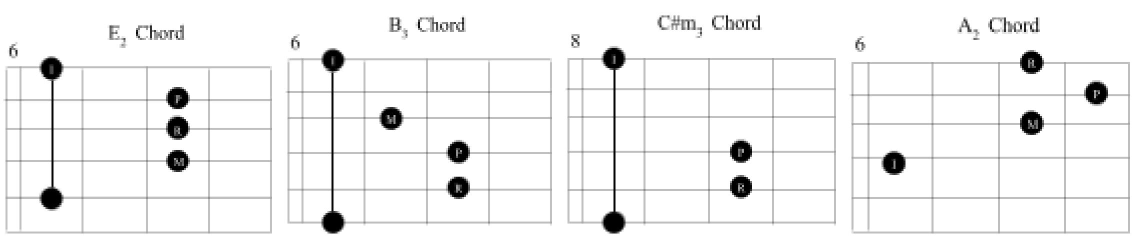
5.2 Conclusion

The aim of this research paper is to answer the question:

- What combination of guitar chord positions will minimize the distance needed for each finger to travel between frets in the chord progression of E, B, C#m and A?

To answer this, assuming that there is no preference towards fret-based movements and towards string-based movements, the progression of $E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_2$ minimizes the distance needed for fingers to travel. The tablature/tab representation of this is shown as:

Figure 5.2: Optimal Chord Progression in Tablature Form



(diagrams are original creations by Keith Choa)

On the other hand, for guitar players who have a preference for string-based movements, the "no-preference" optimal progression can be used. The optimal progression only varies if there is a certain level of preference to fret-based movements, as seen in the results of Dijkstra's, given different values of n. Though, the survey conducted to a sample set of guitarists suggests that despite having an optimal progression for specific preferences, the progression of:

$$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$$

is preferred. This is because, for them, E_1 is the most comfortable and familiar starting chord; and the movements to the other chords in this progression are simply translations along the fretboard. Despite there being a greater distance overall travelled by their fingers, the minimal change in form or "simplicity" already compensates for this.

It is very important to note that the study is directed towards beginners. Optimizing distances needed for fingers to travel will be more helpful for a beginner guitarist than it would be for a seasoned/professional. For a professional, the movement between chords has already been set in muscle memory. Thus, there would be no added benefit in knowing how to minimize finger movements. Furthermore, even if minimizing distances between finger positions decreased the amount of time professional guitarists spent switching between chords, that does not change the fact that music needs to be played following a certain meter or beat. At a certain point, switching between chords faster will be unnecessary, as one would still have to strum the next chord at a specific interval of time.

5.3 Recommendations

One key point of consideration that has not been included in this investigation is the difficulty of chords. While difficulty is definitely subjective to the musician, the chord positions E_3, B_2, $C\#m_2$, and A_2 are generally found to be more difficult to play. This is also because the distance between fingers within the chord shape are quite large relative to the other shapes. Thus, a suitable method of taking this into consideration would be to calculate the distance between fingers within the chord, and using that as a variable to multiply against the weight values to discourage paths towards that particular chord. This is, however, very subjective and completely depends on the guitarist. One guitarist may find the E_3 chord particularly difficult, for example, while another one may find the E_2 chord more difficult.

Appendices

Appendix 1

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_2

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_1
E_2	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	∞	∞	∞	∞	∞	∞	∞
B_3	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	25.21 $B_3 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	∞	∞	∞	∞
B_2	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	∞	∞	∞	∞
$C\#m_3$	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	29.18 $C\#m_3 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	∞
A_3	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	29.18 $C\#m_3 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	23.95 $A_2 \to E_2$
$C\#m_1$	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	28.93 $C\#m_1 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	23.95 $A_2 \to E_2$
$C\#m_2$	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	28.93 $C\#m_1 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	23.95 $A_2 \to E_2$
B_1	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	28.93 $C\#m_1 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	23.95 $A_2 \to E_2$
A_3	20.00 $E_2 \to B_1$	10.30 $E_2 \to B_2$	6.41 $E_2 \to B_3$	19.06 $B_3 \to C\#m_1$	19.36 $B_2 \to C\#m_2$	12.41 $B_3 \to C\#m_3$	28.93 $C\#m_1 \to A_1$	18.95 $C\#m_3 \to A_2$	24.40 $C\#m_3 \to A_3$	23.95 $A_2 \to E_2$

Appendix 2

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_3

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_1
E_3	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m_1$	41.60 $B_1 \rightarrow C\#m_2$	26.63 $B_1 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_1$	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m_1$	41.60 $B_1 \rightarrow C\#m_2$	26.63 $B_1 \rightarrow C\#m_3$	23.58 $C\#m_1 \rightarrow A_1$	29.36 $C\#m_1 \rightarrow A_2$	19.37 $C\#m_1 \rightarrow A_3$	∞
A_3	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m_1$	41.60 $B_1 \rightarrow C\#m_2$	26.63 $B_1 \rightarrow C\#m_3$	23.58 $C\#m_1 \rightarrow A_1$	29.36 $C\#m_1 \rightarrow A_2$	19.37 $C\#m_1 \rightarrow A_3$	32.80 $A_3 \rightarrow E_3$
B_3	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m$	39.06 $B_3 \rightarrow C\#m_2$	26.26 $B_3 \rightarrow C\#m_3$	23.58 $C\#m_1 \rightarrow A_1$	29.36 $C\#m_1 \rightarrow A_2$	19.37 $C\#m_1 \rightarrow A_3$	32.80 $A_3 \rightarrow E_3$
A_1	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m$	39.06 $B_3 \rightarrow C\#m_2$	26.26 $B_3 \rightarrow C\#m_3$	23.58 $C\#m_1 \rightarrow A_1$	29.36 $C\#m_1 \rightarrow A_2$	19.37 $C\#m_1 \rightarrow A_3$	27.82 $A_1 \rightarrow E_3$
$C\#m_3$	5.00 $E_3 \rightarrow B_1$	28.00 $E_3 \rightarrow B_2$	20.26 $E_3 \rightarrow B_3$	13.71 $B_1 \rightarrow C\#m$	39.06 $B_3 \rightarrow C\#m_2$	26.26 $B_3 \rightarrow C\#m_3$	23.58 $C\#m_1 \rightarrow A_1$	29.36 $C\#m_1 \rightarrow A_2$	19.37 $C\#m_1 \rightarrow A_3$	27.82 $A_1 \rightarrow E_3$

Appendix 3

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_1 with a ratio of string preference to fret preference of 0.2

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_1
E_1	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	60.96 $B_1 \rightarrow C\#m_1$	205.60 $B_1 \rightarrow C\#m_2$	130.61 $B_1 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_1$	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	60.96 $B_1 \rightarrow C\#m_1$	205.60 $B_1 \rightarrow C\#m_2$	130.61 $B_1 \rightarrow C\#m_3$	106.14 $C\#m_1 \rightarrow A_1$	131.34 $C\#m_1 \rightarrow A_2$	81.36 $C\#m_1 \rightarrow A_2$	∞
A_3	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	60.96 $B_1 \rightarrow C\#m_1$	205.60 $B_1 \rightarrow C\#m_2$	130.61 $B_1 \rightarrow C\#m_3$	106.14 $C\#m_1 \rightarrow A_1$	131.34 $C\#m_1 \rightarrow A_2$	81.36 $C\#m_1 \rightarrow A_2$	146.7 $A_3 \rightarrow E_1$
B_3	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	60.96 $B_1 \rightarrow C\#m_1$	180.61 $B_3 \rightarrow C\#m_2$	130.61 $B_1 \rightarrow C\#m_3$	106.14 $C\#m_1 \rightarrow A_1$	131.34 $C\#m_1 \rightarrow A_2$	81.36 $C\#m_1 \rightarrow A_2$	146.7 $A_3 \rightarrow E_1$
A_1	25.48 $E_1 \rightarrow B_1$	130.16 $E_1 \rightarrow B_2$	95.23 $E_1 \rightarrow B_3$	60.96 $B_1 \rightarrow C\#m_1$	180.61 $B_3 \rightarrow C\#m_2$	130.61 $B_1 \rightarrow C\#m_3$	106.14 $C\#m_1 \rightarrow A_1$	131.34 $C\#m_1 \rightarrow A_2$	81.36 $C\#m_1 \rightarrow A_2$	115.24 $A_1 \rightarrow E_1$

Appendix 4

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_1 with a ratio of string preference to fret preference of 4

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_1
$\underline{E_1}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
$\underline{B_1}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	60.28 $B_1 \rightarrow C\#m_2$	46.33 $B_1 \rightarrow C\#m_3$	∞	∞	∞	∞
$\underline{B_3}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	60.28 $B_1 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$\underline{B_2}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$\underline{C\#m_1}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	52.51 $C\#m_1 \rightarrow A_3$	∞
$\underline{C\#m_3}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	49.48 $C\#m_3 \rightarrow A_3$	∞
$\underline{A_3}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	49.48 $C\#m_3 \rightarrow A_3$	77.68 $A_3 \rightarrow E_3$
$\underline{A_1}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	49.48 $C\#m_3 \rightarrow A_3$	77.56 $A_1 \rightarrow E_3$
$\underline{C\#m_2}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	49.48 $C\#m_3 \rightarrow A_3$	77.56 $A_1 \rightarrow E_3$
$\underline{A_2}$	17.01 $E_1 \rightarrow B_1$	35.99 $E_1 \rightarrow B_2$	31.48 $E_1 \rightarrow B_3$	36.02 $B_1 \rightarrow C\#m_1$	56.30 $B_2 \rightarrow C\#m_2$	37.48 $B_3 \rightarrow C\#m_3$	54.44 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	49.48 $C\#m_3 \rightarrow A_3$	77.56 $A_1 \rightarrow E_3$

Appendix 5

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_1 with a ratio of string preference to fret preference of 5

Current Vertex	*B_1*	*B_2*	*B_3*	*$C\#m_1$*	*$C\#m_2$*	*$C\#m_3$*	*A_1*	*A_2*	*A_3*	*E_1*
E_1	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	67.12 $B_1 \rightarrow C\#m_2$	53.84 $B_1 \rightarrow C\#m_3$	∞	∞	∞	∞
B_3	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	67.12 $B_1 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
B_2	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_3$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	72.12 $C\#m_3 \rightarrow A_1$	82.77 $C\#m_3 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	∞
$C\#m_1$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	∞
A_3	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_1$
$C\#m_2$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_1$
A_1	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_1$
A_2	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	64.70 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_1$

Appendix 6

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_2 with a ratio of string preference to fret preference of 0.2

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_2
E_2	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_3	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	95.48 $B_3 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_3$	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	95.48 $B_3 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	∞
B_2	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	∞
A_2	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	166.45 $A_2 \rightarrow E_3$
$C\#m_1$	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	166.45 $A_2 \rightarrow E_3$
$C\#m_2$	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	166.45 $A_2 \rightarrow E_3$
B_1	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	166.45 $A_2 \rightarrow E_3$
A_3	100.00 $E_1 \rightarrow B_1$	45.30 $E_1 \rightarrow B_2$	10.10 $E_1 \rightarrow B_3$	70.23 $B_3 \rightarrow C\#m_1$	80.88 $B_2 \rightarrow C\#m_2$	40.1 $B_3 \rightarrow C\#m_3$	120.26 $C\#m_3 \rightarrow A_1$	66.45 $C\#m_3 \rightarrow A_2$	101.1 $C\#m_3 \rightarrow A_3$	165.44 $A_3 \rightarrow E_3$

Appendix 7

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_2 with a ratio of string preference to fret preference of 0.25

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_2
E_2	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_3	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	57.29 $B_3 \rightarrow C\#m_1$	77.59 $B_3 \rightarrow C\#m_2$	33.12 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_3$	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	57.29 $B_3 \rightarrow C\#m_1$	77.59 $B_3 \rightarrow C\#m_2$	33.12 $B_3 \rightarrow C\#m_3$	97.32 $C\#m_3 \rightarrow A_1$	54.78 $C\#m_3 \rightarrow A_2$	81.12 $C\#m_3 \rightarrow A_3$	∞
B_2	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	57.29 $B_3 \rightarrow C\#m_1$	65.09 $B_2 \rightarrow C\#m_2$	33.12 $B_3 \rightarrow C\#m_3$	97.32 $C\#m_3 \rightarrow A_1$	54.78 $C\#m_3 \rightarrow A_2$	81.12 $C\#m_3 \rightarrow A_3$	∞
$C\#m_1$	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	57.29 $B_3 \rightarrow C\#m_1$	65.09 $B_2 \rightarrow C\#m_2$	33.12 $B_3 \rightarrow C\#m_3$	97.32 $C\#m_3 \rightarrow A_1$	54.78 $C\#m_3 \rightarrow A_2$	73.78 $C\#m_1 \rightarrow A_3$	∞
A_2	80.00 $E_1 \rightarrow B_1$	36.37 $E_1 \rightarrow B_2$	9.12 $E_1 \rightarrow B_3$	57.29 $B_3 \rightarrow C\#m_1$	65.09 $B_2 \rightarrow C\#m_2$	33.12 $B_3 \rightarrow C\#m_3$	97.32 $C\#m_3 \rightarrow A_1$	54.78 $C\#m_3 \rightarrow A_2$	73.78 $C\#m_1 \rightarrow A_3$	62.28 $A_2 \rightarrow E_3$

Appendix 8

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_2 with a ratio of string preference to fret preference of 2

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_2
E_2	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_3	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	35.61 $B_3 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
B_2	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_3$	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	∞
B_1	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	∞
$C\#m_2$	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	∞
$C\#m_1$	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	∞
A_3	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	48.49 $A_3 \rightarrow E_3$
A_2	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	44.00 $A_2 \rightarrow E_3$
A_1	20.00 $E_1 \rightarrow B_1$	13.13 $E_1 \rightarrow B_2$	12.24 $E_1 \rightarrow B_3$	26.66 $B_3 \rightarrow C\#m_1$	25.73 $B_2 \rightarrow C\#m_2$	18.24 $B_3 \rightarrow C\#m_3$	37.12 $C\#m_3 \rightarrow A_1$	35.00 $C\#m_3 \rightarrow A_2$	30.24 $C\#m_3 \rightarrow A_3$	44.00 $A_2 \rightarrow E_3$

Appendix 9

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_2 with a ratio of string preference to fret preference of 5

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_2
E_2	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	67.12 $B_1 \rightarrow C\#m_2$	53.84 $B_1 \rightarrow C\#m_3$	∞	∞	∞	∞
B_3	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	67.12 $B_1 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
B_2	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	∞	∞	∞	∞
$C\#m_3$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	72.12 $C\#m_3 \rightarrow A_1$	82.77 $C\#m_3 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	∞
$C\#m_1$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	∞
A_3	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_3$
$C\#m_2$	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_3$
A_1	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_3$
A_2	20.82 $E_1 \rightarrow B_1$	40.38 $E_1 \rightarrow B_2$	36.42 $E_1 \rightarrow B_3$	43.64 $B_1 \rightarrow C\#m_1$	64.63 $B_2 \rightarrow C\#m_2$	42.42 $B_3 \rightarrow C\#m_3$	65.63 $C\#m_1 \rightarrow A_1$	77.57 $C\#m_1 \rightarrow A_2$	54.42 $C\#m_3 \rightarrow A_3$	87.79 $A_3 \rightarrow E_3$

Appendix 10

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_3 with a ratio of string preference to fret preference of 0.2

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_3
E_3	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	44.48 $B_1 \to C\#m_1$	189.12 $B_1 \to C\#m_2$	114.13 $B_1 \to C\#m_3$	∞	∞	∞	∞
$C\#m_1$	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	44.48 $B_1 \to C\#m_1$	189.12 $B_1 \to C\#m_2$	114.13 $B_1 \to C\#m_3$	89.66 $C\#m_1 \to A_1$	114.86 $C\#m_1 \to A_2$	64.88 $C\#m_1 \to A_3$	∞
A_3	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	44.48 $B_1 \to C\#m_1$	189.12 $B_1 \to C\#m_2$	114.13 $B_1 \to C\#m_3$	89.66 $C\#m_1 \to A_1$	114.86 $C\#m_1 \to A_2$	64.88 $C\#m_1 \to A_3$	115.73 $A_3 \to E_3$
A_1	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	44.48 $B_1 \to C\#m_1$	189.12 $B_1 \to C\#m_2$	114.13 $B_1 \to C\#m_3$	89.66 $C\#m_1 \to A_1$	114.86 $C\#m_1 \to A_2$	64.88 $C\#m_1 \to A_3$	109.71 $A_1 \to E_3$
B_3	9.00 $E_1 \to B_1$	140.00 $E_1 \to B_2$	90.50 $E_1 \to B_3$	44.48 $B_1 \to C\#m_1$	175.88 $B_3 \to C\#m_2$	114.13 $B_1 \to C\#m_3$	89.66 $C\#m_1 \to A_1$	114.86 $C\#m_1 \to A_2$	64.88 $C\#m_1 \to A_3$	109.71 $A_1 \to E_3$

Appendix 11

Manual application of Dijkstra's algorithm to find the most optimal route to minimize distance traveled in the chord progression E, B, C#m, A, coming from the chord position E_3 with a ratio of string preference to fret preference of 5

Current Vertex	B_1	B_2	B_3	$C\#m_1$	$C\#m_2$	$C\#m_3$	A_1	A_2	A_3	E_3
E_3	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	∞	∞	∞	∞	∞	∞	∞
B_1	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	67.30 $B_1 \to C\#m_2$	54.02 $B_1 \to C\#m_3$	∞	∞	∞	∞
B_2	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	54.02 $B_1 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	∞
C_1	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	54.02 $B_1 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	∞
B_3	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	52.57 $B_3 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	∞
$C\#m_2$	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	52.57 $B_3 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	∞
$C\#m_3$	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	52.57 $B_3 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	∞
A_3	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	52.57 $B_3 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	107.08 $A_3 \to E_3$
A_1	21.00 $E_1 \to B_1$	28.00 $E_1 \to B_2$	46.57 $E_1 \to B_3$	43.82 $B_1 \to C\#m_1$	52.25 $B_2 \to C\#m_2$	52.57 $B_3 \to C\#m_3$	65.81 $C\#m_1 \to A_1$	77.75 $C\#m_1 \to A_2$	64.22 $C\#m_1 \to A_3$	73.2 $A_1 \to E$

Appendix 12

Results of the Survey to Test the Optimal Progressions Obtained in the Study, from a sample of 4 guitarists

Name	Years of Playing	Preferred Movement (String/Fret-based)	Preferred Starting Chord	Preferred Progression	Notes
Sae Joon Cheon	< 1	String-based*	E_1	$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	Struggle with switching between positions in the other progressions
John Chua	8	Fret-based	E_1	$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	n/a
			E_2 (second favourite)	$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	The A_2 chord was difficult to play
Lexi St. Laurent	~5	String-Based*	E_1	$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	n/a
			E_2 (second favourite)	$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	Couldn't play the A_2 chord
William Hu	2	Fret- Based	E_1	$E_1 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	E_1 to B_1 in the other progression is difficult
			E_2 (switched from E_3)	$E_2 \rightarrow B_3 \rightarrow C\#m_3 \rightarrow A_3$	n/a

Works Cited

Abiy, Thaddeus, et al. “Dijkstra's Shortest Path Algorithm.” Brilliant, brilliant.org/wiki/dijkstras-short-path-finder/.

Caudill, Heather. “THE FOUR CHORDS OF POP AND THEIR PREVALENCE IN HIT SONGS.” The Pacific, 14 Apr. 2014, thepacific.web.unc.edu/2014/04/24/the-four-chords-of-pop-and-their-prevalence-in-hit-songs/.

Davies, Scott. “Musical Chords and Scales.” Scott Davies, scottdavies.net/chords_and_scales/music.html.

“How to Read Guitar Tabs.” BLGuitar.com, www.blguitar.com/lesson/beginner/reading_tab.html.

Konecky, Larry. “Triads.” Alcorn State University, musictheory.alcorn.edu/Version1/theory1/triad.htm.

Newbold, Ben. “Learning to Play the Guitar.” University of Utah School of Computing, www.cs.utah.edu/~bnewbold/guitarguide/play.html.

Osikiewicz, Beth-Allyn. “MIDPOINT AND DISTANCE FORMULAS.” Math 11011, www.personal.kent.edu/~bosikiew/Algebra-handouts/midpt-dist.pdf.