ROYAL INSTITUTE
OF TECHNOLOGY

Optimal piano fingering for simple melodies

Authored by:

Eric Nellåker
0737038090
nellaker@kth.se
VALTHORNSVÄGEN 29
75650 UPPSALA

Xingye Lu
0720325305
xingye@kth.se
BJÖRNKULLARING 3B
141 51 HUDDINGE

Supervisor:
Anders Askenfelt

School of Computer Science and Communications
Royal Institute of Technology

Degree Project in Computer Science, First Cycle, DD143X

Abstract

This study proposes an ergonomic model-based algorithm to automatically decide an optimal piano fingering for a given simple melody that can be played by the right hand. The ergonomic model is represented by 13 rules based on physical constrains related to piano playing which can score the difficulty of a fingering. Optimal fingering thus is generated by finding the fingering with minimum difficulty in the tree of possible fingerings. The proposed algorithm was tested through generating optimal fingerings for three pieces and making a comparison between the generated fingerings and the fingerings provided by two experienced pianists. The result indicated that the automatically generated fingerings were close enough to the proposed fingerings from the pianists.

Sammanfattning

Denna studie undersöker möjligheten att generera fingersättningar för enstämmiga pianostycken med hjälp av en algoritm baserad på ergonomiska regler. De 13 regler algoritmen utnyttjar representerar fysiologiska begränsningar av sträckningar och förflyttningar av högerhanden vilka poängsätts utifrån det intervall som spelas. Algoritmen testades på tre pianostycken och resultaten jämfördes med fingersättningar från två erfarna pianister. Jämförelsen visade att algoritmen lyckades med att generera fingersättningar som var rimligt lika vad en erfaren pianist hade valt att spela.

Contents

1 Introduction

When playing a piece of music on the piano the fingers chosen for each note determine at what ease the player can perform the music. In some rare cases the piece is so simple that the fingering is obvious to the pianist. But for most pieces, even for relatively simple melodies, there will be several different fingerings to consider. Several factors contribute to this complexity. Pianists have different sizes of their hands and the maximum span of keys they can reach with the fingers differs. Further, pianists have developed different playing techniques and are at different levels in developing their playing skills. All together this makes "the optimal piano fingering" a relative definition rather than an absolute one. This fact points to the difficulty of deciding on an appropriate fingering for a piece, and especially beginners may need guidance to find a viable fingering. This project proposes to use algorithms to find the optimal fingering for piano pieces in order to alleviate this problem.

1.1 Problem Statement

The main goal of the project was to create an implementation that automatically generates optimal piano fingerings from the musical score that are close enough to what a professional pianist would choose. The pieces are limited to monophonic melodies, meaning that only one key is played at a time.

2 Background

2.1 Literature Study

Many approaches can be made to finding the optimal fingering for a piece of piano music. Parncutt(1997)[1] discussed using 12 rules based on the physical constraints of the hand to value the difficulty of each node in the fingering tree that generated from a given piece. This tree can then be used to search for the most optimal fingerings. The 12 rules use a 'reach table' which contains the movement constraints (maximum and minimum finger spans in several situations) for each possible finger pair . By tweaking parameters of the reach table ((Table 2.2)Parncutt et al.,1997) the algorithm can be accommodated to people with different playing habits and size of hands.

Yuichiro (2010)[2] used a Hidden Markov Model (HMM)-based algorithm for automatic decision of optimal piano fingering. They represent the forms of hands and fingers as the hidden states in HMM and model the sequence of notes from given piece as outputs associated with HMM transition. The parameters of the HMM like transition probabilities and emission probabilities are all estimated by physical constraints of piano playing. Optimal fingering decision then can be found by searching the sequence of hidden state transition with maximum probability using Viterbi algorithm.

2.2 Ergonomic model

The ergonomic model based on physical constraints using a set of rules to score fingerings in difficulty (Parncutt et al.,1997).

Physical constraints in piano playing mean limited spans of finger pairs and a desired limited movement of the hand. So rules are needed to constrain the fingering, e.g. in case that the maximum usable span of finger pairs used to play two consecutive notes is less than physical distance on the keyboard between the two notes.

[1] Parncutt R., Sloboda J.(1997) "An Ergonomic Model of Keyboard Fingering for Melodic Fragments",*Music Perception*, Vol.14
1

[2] Yuichiro Y., Hirokazu K. and Shigeki S.(2010) "Automatic Decision of Piano Fingering Based on Hidden Markov Models"
2

A *reach table* (see Table 2.2) is used to score the fingering by the constraint rules. The values in the table indicate different spans of semitone steps for different finger pairs. The largest and smallest span which can be used in practice (*MaxPrac* & *MinPrac*) are used to gauge how difficult a certain interval (number of semitones) is to play. They represent the maximum and minimum physical reach of finger pairs.

The largest and smallest comfortable span (*MaxComf* & *MinComf*) describe the maximum and minimum comfortable interval to play by finger pairs.

The relaxed spans (*MaxRel* & *MinRel*) describes the maximum and minimum reach of finger pairs in a relaxed state. A negative value means 'thumb-passing' playing, meaning that the thumb passes under a couple of the other fingers.

The numbering of the fingers used in piano playing is shown in Fig. 2.2. The 12 tones in an octave used in Western music is shown in Fig. 2.1 The interval (pitch ratio) between each two consecutive tones is one semitone. Accidentals (indicated by #) are played on shorter, black keys which extend above the level of the white keys. This design of the keyboard has important implications for the ergonomic constraints in piano fingering.

The rules use the reach tables to measure the physical difficulty (ergonomic cost) associated with a certain fingering, weighting certain rules heavier. Spans outside the maximal and minimum practical distances causes a very high cost as this effectively means that the whole hand has to move.

Finger Pairs	MinPrac	MinComf	MinRel	MaxRel	MaxComf	MaxPrac
1-2	-5	-3	1	5	8	10
1-3	-4	-2	3	7	10	12
1-4	-3	-1	5	9	12	14
1-5	-1	1	7	10	13	15
2-3	1	1	1	2	3	5
2-4	1	1	3	4	5	7
2-5	2	2	5	6	8	10
3-4	1	1	1	2	2	4

3-5	1	1	3	4	5	7
4-5	1	1	1	2	3	5

Table 2.2 Reach table for right hand(see text for explanation)

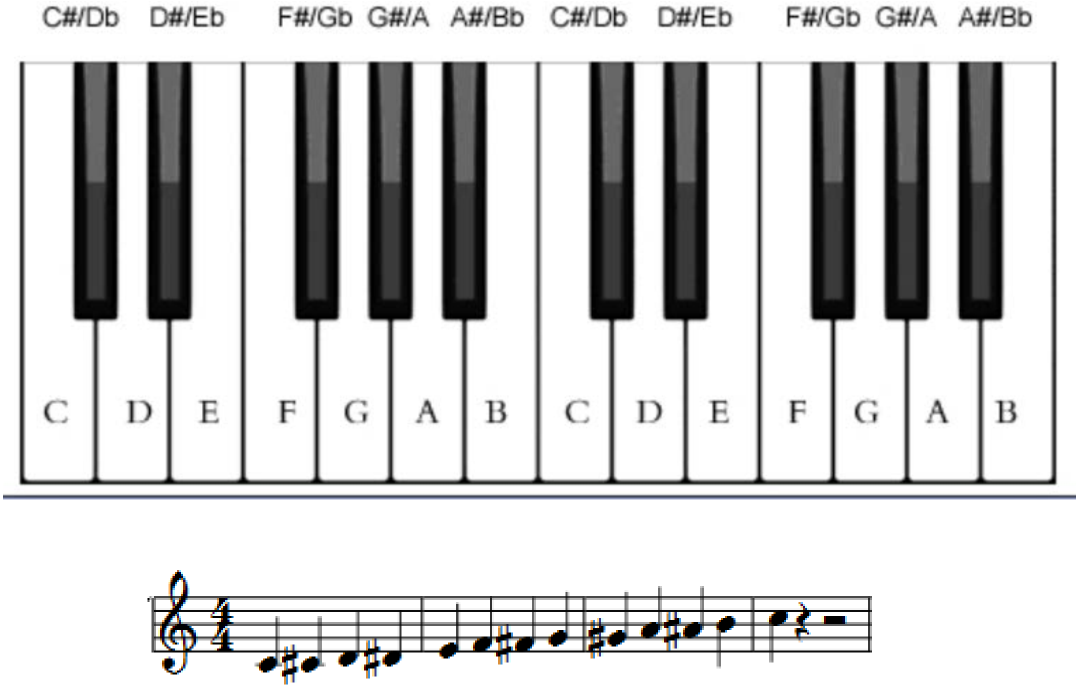


Fig. 2.1. The 12 tones in an octave used in Western music

Both papers mentioned above used physical constraints in piano playing to estimate and determine parameters used in the process of generating optimal fingering. In Parncutt(1997), physical constraints are represented as the form of a set of rules to score the difficulty of fingering. In Yuichiro (2010), physical constraints are showed in the form of transition probability and emission probability. Both Parncutt et al(1997) and Yonebayashi et al (2010) treated input melodies without pauses (consecutive sequence of notes), which might have influence on the optimal fingering generated by algorithms.

The method proposed in this study will be based on the ergonomic model from Parncutt et al.(1997) with the added feature of taking pauses into account for finding the optimal fingering. Pauses allow movements of the hand without disturbing the musical phrasing, which may have a strong influence on the choice of fingering. A special rule for pauses was implemented.

Since the emission probability in Hidden Markov Model is not given by Yonebayashi et al (2010) and is really hard to be estimated with limited resources and time, HMM approaches to the problem will not be taken into consideration.

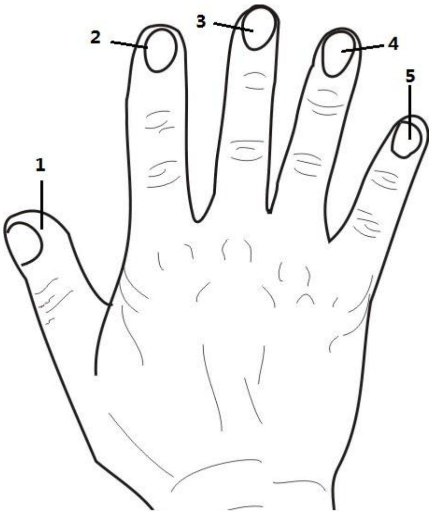


Fig. 2.2 .Numbering of fingers in piano playing

3 Method

3.1 Algorithm

The algorithm must be able to:

1. Input a piece in midi format, composed of a monophonic of notes (no chords), that can be played with the right hand;
2. Generate several possible fingerings for the piece;
3. Measure the difficulty score of each fingering;
4. Output the optimal fingering with the lowest difficulty score for the piece in the format of a series of numbers from 1-5 where each number represents a finger (FIG1.1).

The algorithm described in Parncutt et al.(1997) was implemented in Java. Instead of treating input melodies as uninterrupted, consecutive sequences of notes, pauses were into consideration and an extra rule for pauses was added.

According to the original 12 rules, the change of hand position is not allowed or adds a large cost to the estimated difficulty of fingering. However, in the case of playing pieces including pauses, pianists prefer to change the position of the hands during the pause, if the interval between the notes preceding and following the pause is really large or small (exceeding MaxComf or smaller than the MinComf).

3.2 Rules

There are 12 rules used to evaluate a sequence of fingering according to Parncutt (1997) and a new rule was designed to handle pauses.

3.2.1 Stretch rule

Assign 2 points for each semitone that an interval exceeds Maxcomf or is less than MinComf. (Parncutt et al.,1997,P349)

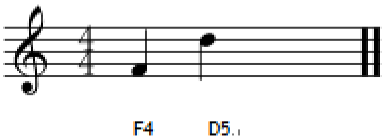


Fig. 3.2.1 Example of Strech Rule and Large Span Rule

The interval between the two notes F4-D4 in Fig 3.2.1 is 9 semitones. If finger

pair 1-2 is chosen to play the two notes, according to Table 2.2 the MaxComf of finger pair 1-2 is 8 semitones. F4-D4 is 1 semitone larger than the MaxComf, so the strech rule assigns 2×1=2 points.

3.2.2 Small-Span Rule

For finger pairs including the thumb, assign 1 point for each semitone that an interval is less than MinRel.(Parncutt et al.,1997,P350)

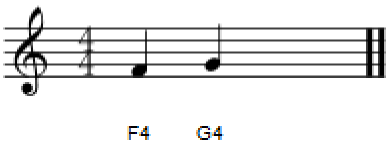


Fig. 3.2.2 Example of Small-span Rule and Weak Finger Rule

If notes F4-G4 in Fig 3.2.2 are played by finger pair 1-4, since the interval for F4-G4 is 2 semitones and the MinRel for 1-4 is 5 semitones. According to Small-Span Rule, there will be 5-2=3 points added to the estimated difficulty of this fingering.

3.2.3 Large-Span Rule

For finger pairs including the thumb, assign 1 point for each semitone that an interval exceeds MaxRel. For finger pairs not including the thumb, assign 2 points per semitone. (Parncutt et al.,1997,P351)

If the interval F4-D4 in Fig 3.2.1 is played by finger pair 1-2, the MaxRel for 1-2 is 5 semitones and thumb (finger 1) is included in the finger pair. Large-Span Rule will assign 2×(9-5)=8 additional difficulty points.

3.2.4 Position-Change-Count Rule

Assign 2 points for every full change of hand position and 1 point for every half change. A change of hand position occurs whenever the first and third notes in a consecutive group of three span an interval that is greater than MaxComf or less than MinComf for the corresponding fingers. In a full change, three conditions are satisfied simultaneously: The finger on the second of the three notes is the thumb; the second pitch lies between the first and third pitches; and the interval between the first and third pitches is greater than MaxPrac or less than MinPrac. All other changes are half changes.(Parncutt et al.,1997,P353)

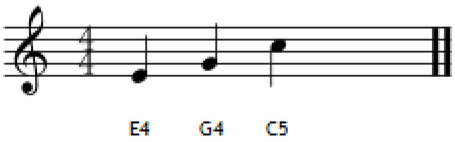


Fig. 3.2.4 Example of Position-Change-Count Rule

If E4-G4-C5 in Fig 3.2.4 is played by fingers 2-1-4, thumb is used to play the second of three notes, G4 lies between E4 and C5 and the interval for E4-C5 is greater than the MaxPrac of 2-4. So it is a full change of hand position and 2 points need to be assigned.

3.2.5 Position-Change-Size Rule

If the interval spanned by the first and third notes in a group of three is less than MinComf, assign the difference between the interval and MinComf (expressed in semitones). Conversely, if the interval is greater than MaxComf, assign the difference between the interval and MaxComf.(Parncutt et al. ,1997,P354)

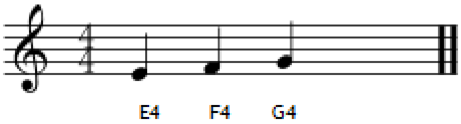


Fig 3.2.5 Example of Position-Change-Size Rule

The fingering for E4-F4-G4 in Fig 3.2.5 is 4-1-2. The MinComf for 4-2 is -1 semitone and the interval for E4-G4 is 3 semitones. So the rule contributes 3-(-1)=4 difficulty points.

3.2.6 Weak-Finger Rule

Assign 1 point every time finger 4 or finger 5 is used.(Parncutt et al.,1997,P 357)

If finger pair 4-5 is chosen to play F4-G4 in Fig 3.2.2, there will be 2 additional difficulty points added according to Weak-Finger Rule.

3.2.7 Three-Four-Five Rule

Assign 1 point every time fingers 3,4, and 5 occur consecutively in any order, even when groups overlap.(Parncutt et al.,1997,P357)

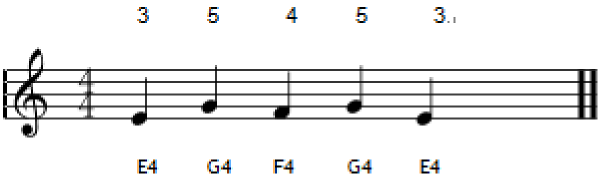


Fig 3.2.7 Example of Three-Four-Five Rule

In Fig 3.2.7, fingering 3-5-4-5-3 is used to play the piece. There are consecutive sequences 3-5-4 and 4-5-3 in the fingering, so the Three-Four-Five Rule will assign 2 points.

3.2.8 Three-to-Four Rule

Assign 1 point each time finger 3 is immediately followed by finger 4.(Parncutt et al.,1997,P 358)

3.2.9 Four-on-Black Rule

Assign 1 point each time fingers 3 and 4 occur consecutively in any order with 3 on white and 4 on black.(Parncutt et al.,1997,P358)

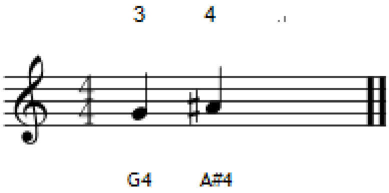


Fig 3.2.9 Example of Four-on-Black Rule

In Fig 3.2.9, notes G4-A#4 are played by 3-4. Since finger 3 is on white key and finger 4 occurs consecutively on black key, 1 difficulty point will be assigned according to the rule.

3.2.10 Thumb-on-Black Rule

Assign 1 point whenever the thumb plays a black key. If the immediately preceding note is white, assign a further 2 points. If the immediately following note is white, assign a further 2 points.(Parncutt et al.,1997,P 358)

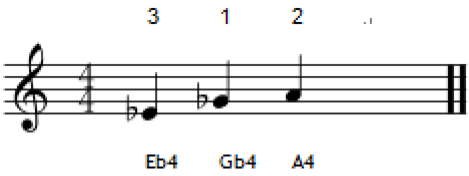


Fig 3.2.10 Example of Thumb-on-Black Rule

Thumb is used to play the black key Gb4 in Fig 3.2.10. The immediately preceding note of Gb4, Eb4, is black, so 0 point is assigned. The immediately following note A4 is white, so the Thumb-on Black Rule will contribute 2 points to the estimated difficulty.

3.2.11 Five-on-Black Rule

If the fifth finger plays a black key and the immediately preceding and following notes are also black, assign 0 points. If the immediately preceding note is white, assign 2 points. If the immediately following key is white, assign 2 further points.(Parncutt et al.,1997,P358)

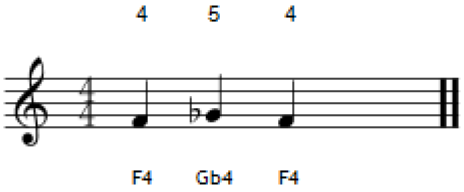


Fig 3.2.11 Example of Five-on-Black Rule

In Fig 3.2.11, Gb4 (black) is played by finger 5 and the immediately preceding and following key (F4 & F4) are both white. So there will be 2+2=4 points assigned.

3.2.12 Thumb-Passing Rule

Assign 1 point for each thumb- or finger-pass on the same level (from white to white or black to black). Assign 3 points if the lower note is white, played by a finger other than the thumb, and the upper is black, played by the thumb.(Parncutt et al.,1997,P360)

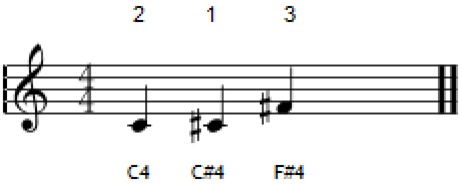


Fig 3.2.12 Example of Thumb-Passing Rule

Given notes C4-C#4-F#4 is played by 2-1-3, the Thumb-Passing Rule assigns 3 points for passing from finger 2 (nonthumb) on white (C4) to finger 1 (thumb) on black(C#4) and 1 points for passing from finger 1 (C#4) on black to finger 3 on black()F#4.

3.2.13 Extra Rule for Pause

Assign one point for each pause where the interval between the note prior and the note after the pause either is less than MinRel or larger than MaxRel. This means that as long as the interval spanned over the pause can be reached in a relaxed positioning of the hand, the hand will remain in the same position. Otherwise the hand will have to be moved. The low penalty for a hand move over a pause is to accomodate for pieces where a move of the hand during a pause may yield an easier fingering. Moving the hand during a pause will not break the phrasing (keeping notes tight together). A pause indicates a natural end of a previous phrase and the start of a new.

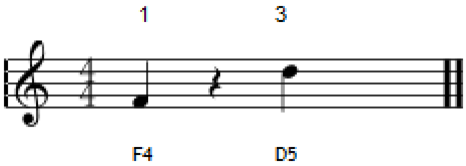


Fig 3.2.13 Example of Extra Rule for Pause

According to Fig 3.2.13, the interval for the immediately preceding and following note of the pause (F4 & D5)) is 9 semitones which is 2 semitones larger than the MaxRel for finger pair 1-3. So, there will be 1 point assigned.\

3.3 Finger Tree

By creating a tree of all possible fingerings for a given piece of music and evaluating the difficulty of each node by assigning a rule-based difficulty score, a path can be found with a minimal total score. This path corresponds to the optimal fingering. When generating the tree each new node is pushed to a sorted queue where the lowest score is at the front. For each iteration of node generation the node with the lowest current score is popped from the queue. Its children are then sorted into the queue, and the process is repeated until a generation has found a full fingering for the piece where the final node has a smaller score than the next node in the queue. The path found is then the optimal fingering without the need of generating a full tree, massively decreasing run time.

Figure 3.3.1 illustrates an example of how the finger tree is generated. The values in the figure are not based on a real generation though, as even the simplest song quickly generates a large tree that is dificult to illustrate. The generation starts with each finger (F1-F5) pushed to a sorted list. Then, while a full path has not been found or a node still exists in the list that has a lower or equal score to the found path, a node is popped from the sorted list and its children are generated.

The score for each node is a sum of the nodes owns score and the score of its parent. As such, each node shows the total score of the fingering thus far in the generation.

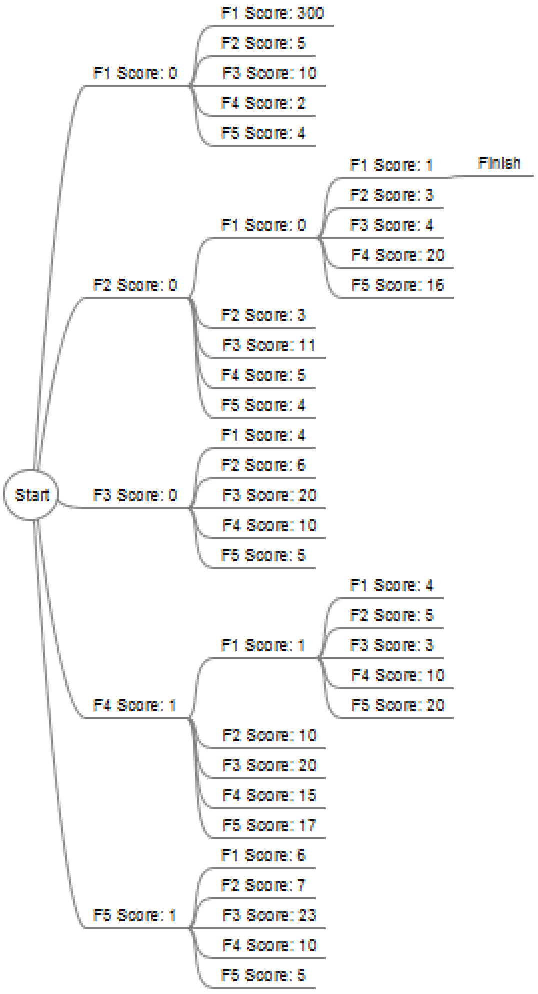


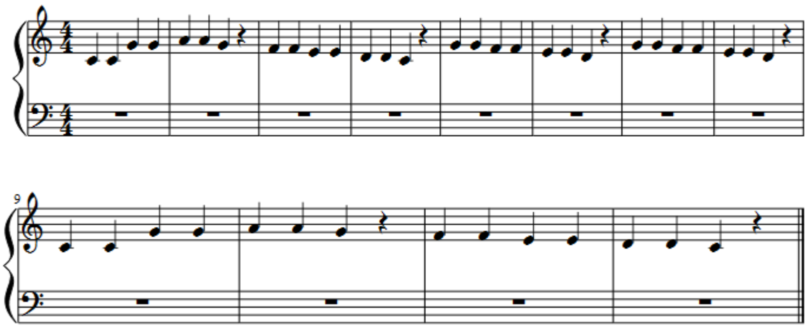
Fig. 3.3.1 Example of Fingering Tree

4 Result

Tests were run on three different pieces: *Twinkle Twinkle little star (*Traditional children song*)*, *Divenire (*Ludovico Einaudi, 2006*)* and *Eine kleine Nacht music (*Wolfgang Amadeus Mozart, 1787*)* . The pieces were given to two pianists who wrote their suggested fingerings for each piece. The fingerings were then analyzed by the algorithm to give a score based on the algorithm's rule set to for comparison with the pianists' fingerings (see Table 4.1, Table 4.2 and Table 4.3). Differences in the chosen fingering between the pianists are highlighted.

4.1 Twinkle Twinkle

The musical score for *Twinkle twinkle* and the fingerings suggested by the algorithm and the two pianists, respectively, are given in Table 4.1.



Fingering from Algorithm	Score for each note	Rules	Fingering from Pianist 1	Score for each note	Rules	Fingering From Pianist 2	Score for each note	Rules	Pitch
1	0	[]	1	0	[]	1	0	[]	C4
1	0	[]	1	0	[]	1	0	[]	C4
2	2	[3]	4	1	[6]	4	1	[6]	G4
2	0	[]	4	1	[6]	4	1	[6]	G4
3	0	[]	5	1	[6]	5	1	[6]	A4
3	0	[]	5	1	[6]	5	1	[6]	A4
2	0	[]	4	1	[6]	4	1	[6]	G4
-1	1	[13]	-1	1	[13]	-1	1	[13]	Pause
3	0	[]	4	1	[6]	4	1	[6]	F4
3	0	[]	4	1	[6]	4	1	[6]	F4
2	0	[]	3	0	[]	3	0	[]	E4
2	0	[]	3	0	[]	3	0	[]	E4

1	0	[]	2	0	[]	2	0	[]	D4
2	1	[2]	2	0	[]	2	0	[]	D4
1	0	[]	1	0	[]	1	0	[]	C4
-1	0	[]	-1	0	[]	-1	0	[]	Paus
3	0	[]	5	1	[6]	4	1	[6]	G4
3	0	[]	5	1	[6]	4	1	[6]	G4
2	0	[]	4	1	[6]	3	0	[]	F4
2	0	[]	4	1		3	0	[]	F4
1	0	[]	3	0	[]	2	0	[]	E4
2	1	[2]	3	0	[]	2	0	[]	E4
1	0	[]	2	0	[]	1	0	[]	D4
-1	0	[]	-1	0	[]	-1	0	[]	Pause
3	0	[]	5	1	[6]	4	1	[6]	G4
3	0	[]	5	1	[6]	4	1	[]	G4
2	0	[]	4	1	[6]	3	0	[]	F4
2	0	[]	4	1	[6]	3	0	[]	F4
1	0	[]	3	0	[]	2	0	[]	E4
2	1	[2]	3	0	[]	2	0	[]	E4
1	0	[]	2	0	[]	1	0	[]	D4
-1	1	[13]	-1	0	[]	-1	1	[13]	Pause
1	0	[]	1	0	[]	1	0	[]	C4
1	0	[]	1	0	[]	1	0	[]	C4
2	2	[3]	4	1	[6]	4	1	[6]	G4
2	0	[]	4	1	[6]	4	1	[6]	G4
3	0	[]	5	1	[6]	5	1	[6]	A4
3	0	[]	5	1	[6]	5	1	[6]	A4
2	0	[]	4	1	[6]	4	1	[6]	G4
-1	1	[13]	-1	1	[13]	-1	1	[13]	Pause
3	0	[]	4	1	[6]	4	1	[6]	F4
3	0	[]	4	1	[6]	4	1	[6]	F4
2	0	[]	3	0	[]	3	0	[]	E4
2	0	[]	3	0	[]	3	0	[]	E4
1	0	[]	2	0	[]	2	0	[]	D4
2	1	[2]	2	0	[]	2	0	[]	D4
1	0	[]	1	0	[]	1	0	[]	C4
Final Score	11		Final Score	24		Final Score	21		
Accuracy				28 %			36 %		

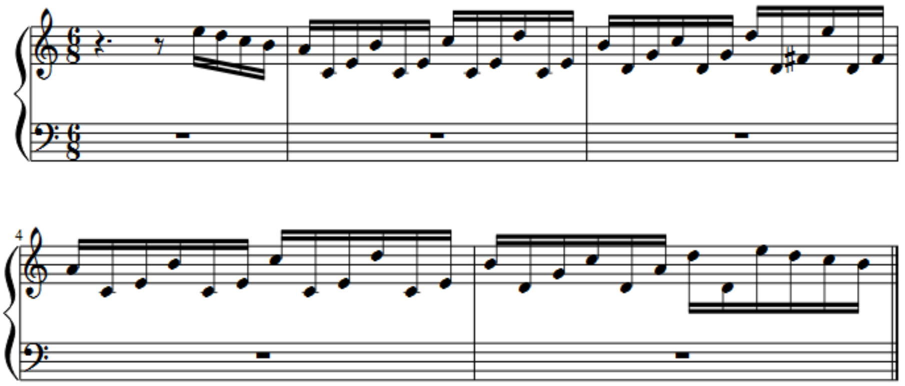
Table 4.1 Comparison between fingerings of *Twinkle twinkle* by the algorithm and the two pianists. The rules applied and the scores for each note are indicated.

Rule 6 *Weak finger* causes a lot of differences between the fingerings by the algorithm and the pianists. This is due to passages where several notes are repeated which causes the use of fingers 4 and 5 to add several points to the score, even though a pianist can see that it is a good trade off to use fingers 4 or 5 instead of having to switch fingers later in the passage.

The pause rule (rule 13) is applied in this piece successfully, allowing a shift of hand to accommodate for an easier fingering, as not shifting the hand would have resulted in the need for crossing fingers to be able to play the following parts.

4.2 Divenire

The musical score for *Diveniere* and the fingerings suggested by the algorithm and the two pianists, respectively, are given in Table 4.2.



Fingering from Algorithm	Score for each note	Rules	Fingering from Pianist 1	Score for each note	Rules	Fingering From Pianist 2	Score for each note	Rules	Pitch
4	1	[6]	4	1	[6]	4	1	[6]	E5
3	0	[]	3	0	[]	3	0	[]	D5
2	0	[]	2	0	[]	2	0	[]	C5
1	0	[]	1	6	[4 5]	1	6	[4 5]	B4
2	4	[2 12]	4	11	[1 2 6 12]	4	11	[1 2 6 12]	A4

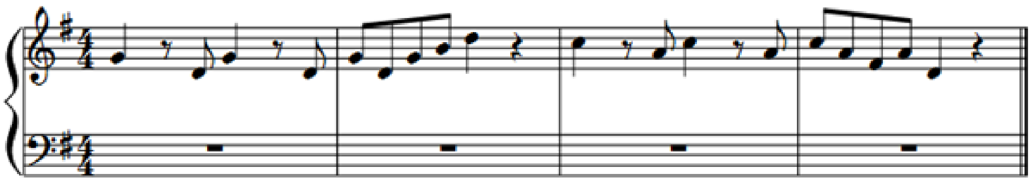
1	6	[1 3]	1	4	[4 5]	1	0	[]	C4
2	0	[]	3	0	[]	2	0	[]	E4
5	3	[3 6]	5	11	[1 3 6]	5	3	[3 6]	B4
1	1	[3]	1	4	[3 4 5]	1	1	[3]	C4
2	0	[]	3	0	[]	2	0	[]	E4
5	5	[3 6]	5	30000	[]	5	5	[3 6]	C5
1	2	[3]	1	6	[3 4 5]	1	2	[3]	C4
2	2	[4 5]	3	2	[4 5]	2	2	[4 5]	E4
5	13	[1 3 6]	5	30000	[]	5	13	[1 3 6]	D5
1	9	[1 3 4 5]	1	12	[1 3 4 5]	1	9	[1 3 4 5]	C4
2	0	[]	3	0	[]	2	0	[]	E4
5	3	[3 6]	4	30000	[]	5	3	[3 6]	B4
1	0	[]	1	0	[]	1	0	[]	D4
2	0	[]	2	0	[]	2	0	[]	G4
5	1	[6]	5	1	[6]	4	3	[3 6]	C5
1	0	[]	1	0	[]	1	1	[3]	D4
2	0	[]	2	0	[]	2	0	[]	G4
5	3	[3 6]	5	3	[3 6]	5	3	[3 6]	D5
1	2	[3]	1	2	[3]	1	2	[3]	D4
2	2	[4 5]	2	2	[4 5]	2	2	[4 5]	F#4
5	13	[1 3 6]	5	13	[1 3 6]	5	13	[1 3 6]	E5
1	9	[1 3 4 5]	1	9	[1 3 4 5]	1	9	[1 3 4 5]	D4
2	0	[]	2	0	[]	2	0	[]	F#4
4	1	[6]	4	1	[6]	4	1	[6]	A4
1	0	[]	1	0	[]	1	0	[]	C4
2	0	[]	2	0	[]	2	0	[]	E4
5	3	[3 6]	5	3	[3 6]	5	3	[3 6]	B4
1	1	[3]	1	1	[3]	1	1	[3]	C4
2	0	[]	2	0	[]	2	0	[]	E4
5	5	[3 6]	5	5	[3 6]	5	5	[3 6]	C5
1	2	[3]	1	2	[3]	1	2	[3]	C4
2	2	[4 5]	2	2	[4 5]	2	2	[4 5]	E4
5	13	[1 3 6]	5	13	[1 3 6]	5	13	[1 3 6]	D5
1	9	[1 3 4 5]	1	9	[1 3 4 5]	1	9	[1 3 4 5]	C4
2	0	[]	2	0	[]	2	0	[]	E4
5	3	[3 6]	4	11	[1 3 6]	5	3	[3 6]	B4
1	0	[]	1	0	[]	1	0	[]	D4
2	0	[]	2	0	[]	2	0	[]	G4
5	1	[6]	5	1	[6]	5	1	[6]	C5

1	0	[]	1	0	[]	1	0	[]	D4
3	0	[]	3	0	[]	3	0	[]	A4
5	3	[3 6]	5	3	[3 6]	5	3	[3 6]	D5
1	4	[3 4 5]	1	4	[3 4 5]	1	4	[3 4 5]	D4
5	7	[1 3 6]	5	7	[1 3 6]	5	7	[1 3 6]	E5
4	2	[6 7]	4	2	[6 7]	4	2	[6 7]	D5
3	0	[]	3	0	[]	3	0	[]	C5
2	0	[]	2	0	[]	2	0	[]	B4
Final Score	135		Final Score	90151		Final Score	145		
Accuracy				87 %			96 %		

Table 4.2 Comparison between fingerings of *Diveniere* by the algorithm and the two pianists. The rules applied and the scores for each note are indicated.

Divenire's output from the algorithm had the most accurate fingering in comparison to the pianists choice of fingering. The most obvious difference is the final scores, where pianist 1 has a score of 90151. This is due to how the algorithm 'cuts off' fingerings it deems outside the possible span a pianist can reach (minPrac and maxPrac). In such cases 30000 points is added to score of the node. Tweaking the reach tables to better reflect the size of the hand of pianist 1 hand would avoid this penalty. Subtracting the high cut off scores, pianist 1 score would reach around 150 points, still very close to the algorithm's output.

4.3 Eine Kleine Nacht Musik



Fingering from Algorithm	Score for each note	Rules	Fingering from Pianist 1	Score for each note	Rules	Fingering From Pianist 2	Score for each note	Rules	Pitch

2	0	[]	4	1	[6]	2	0	[]	G4
-1	0	[]	-1	0	[]	-1	0	[]	Pause
1	0	[]	1	0	[]	1	0	[]	D4
2	0	[]	4	1	[6]	2	0	[]	G4
-1	0	[]	-1	0	[]	-1	0	[]	Pause
1	0	[]	1	0	[]	1	0	[]	D4
2	0	[]	4	1	[6]	2	0	[]	G4
1	0	[]	1	2	[4 5]	1	0	[]	D4
2	0	[]	2	0	[]	2	0	[]	G4
4	1	[6]	3	6	[1 3]	3	6	[1 3]	B4
5	3	[3 6]	5	1	[6]	5	1	[6]	D5
-1	1	[13]	-1	0	[]	-1	1	[13]	Pause
2	0	[]	4	1	[6]	5	1	[6]	C5
-1	0	[]	-1	1	[13]	-1	0	[]	Pause
1	0	[]	1	0	[]	3	0	[]	A4
2	0	[]	4	3	[2 6]	5	1	[6]	C5
-1	0	[]	-1	1	[13]	-1	0	[]	Pause
1	0	[]	1	0	[]	3	0	[]	A4
2	0	[]	4	3	[2 6]	5	1	[6]	C5
1	0	[]	1	4	[2 4 5]	3	0	[]	A4
2	4	[2]	2	4	[2]	2	2	[3]	F#4
4	1	[6]	3	2	[3]	3	2	[3]	A4
1	0	[]	1	0	[]	1	0	[]	D4
Final Score	10		Final Score	31		Final Score	15		
Accuracy				65 %			65 %		

Table 4.2 Comparison between fingerings of *Eine kleine Nacht musik* by the algorithm and the two pianists. The rules applied and the scores for each note are indicated.

Rule 6 causes most of the differences in this piece, once again due to how the algorithm tries to avoid the use of fingers 4 and 5 as much as possible. Rule 13 allows the hand to remain stationary over pauses (the pauses in bars 1 and 3) while where applicable the hand is moved (the pause in bar 2).

5 Discussion

Each test piece showed differences between how pianists chose to play the pieces, and the suggested optimal fingering by the algorithm. However, the algorithm generally generated a good enough fingering. *Twinkle twinkle little star* had the strangest choices of fingerings, including a part where the same note was repeated but not played by the same finger. Both pianists chose to play the repeated notes with one finger. These deviations seem to mostly be caused by Rule 6 being weighted too heavily. Further, the algorithm was not tuned to the hand sizes of the participating pianists, causing the pianists to be able to span their fingerings over much larger intervals without issue, while the algorithm never even considered those fingerings as viable. We chose not to tune these parameters because of lack of time. Figure 5 shows the percentage of triggers for each rule during the generation for each test piece. The reason rule 6 is over represented is that for each child generation, fingers 4 and 5 will always be generated and as such rule 6 will be triggered twice every time a node is popped from the list.

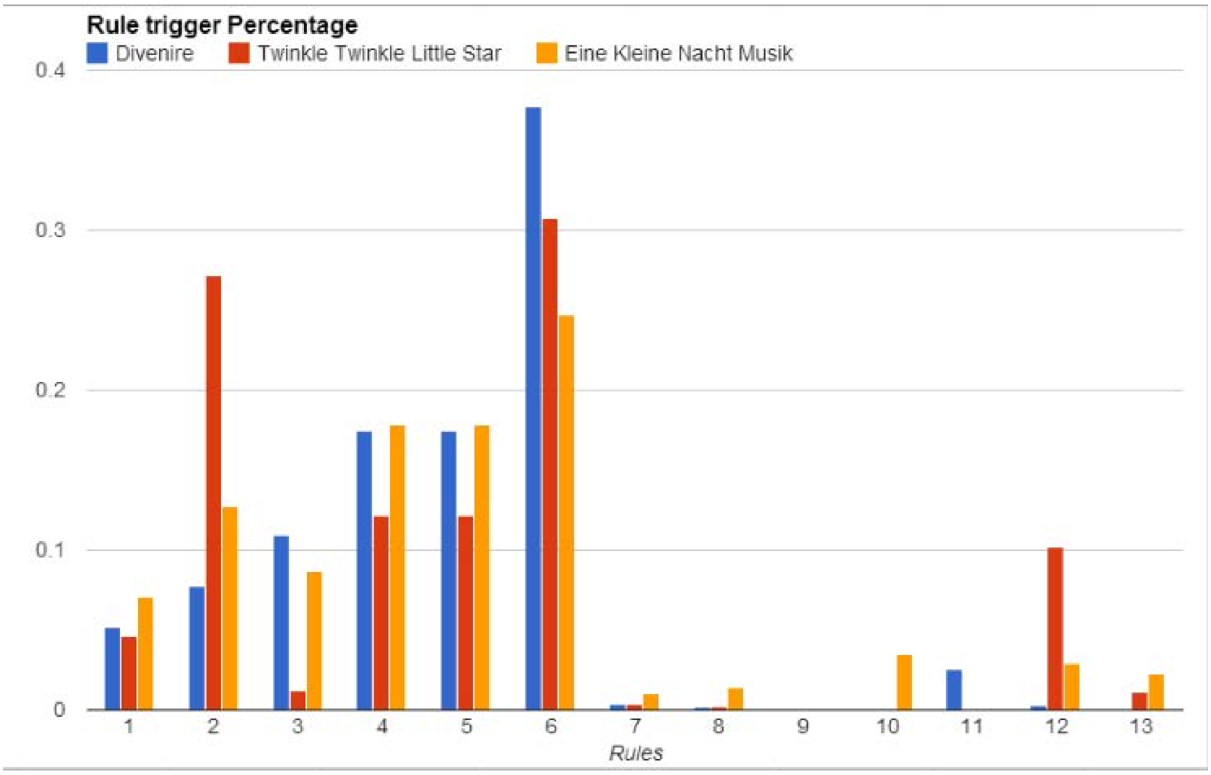


Fig. 5 Rule Trigger Percentage of Three Pieces

The advantages of the method include:

- The method is based on an ergonomic model which contains several important physical constraints in playing piano. Each of the 13 rules represents a specific ergonomic difficulty and can help making the optimal fingering generated by algorithm closer to the pianists' fingerings.

- Rule 13 was a new feature, which allowed the algorithm to accommodate for pieces including pauses. While it would be possible to just split these pieces into sub pieces, or outright ignore the pauses, this approach would cause many odd fingerings as the algorithm is mainly developed for legato playing (no pauses and as such finger movements must never be done in a way where the last note is let go before the next is played). By letting the penalty for a movement of the hand in the case of a pause to be low, only movements where the net win further down the piece is high will be considered. In *Twinkle twinkle little star* hand movements are natural during the pauses, and as such the single point penalized for moving the hand was heavily outweighed by the gains of having a fresh start on the next part of the piece. In *Eine kleine nacht musik* though, movement of the hand during pauses didn't give any gains, and consequently the hand remained stationary through the pauses. In these regards, our proposed rule can be regarded as a valuable expansion of the algorithm, greatly increasing the performance and usability.

The method can be improved in the following ways:

- All parameters, including the spans in the reach table and score points for each rule, are estimated and may not be suitable for everyone and at all levels of playing. In particular, many piano beginners are children with small hands. This problem can be solved by measuring the physical size of player' hand and adapting the penalty scores to the student's playing level.

- By creating a more efficient data structure for the node graphs and better estimation of dead routes through the fingering tree, performance could be increased to allow for longer pieces. The final version of the algorithm would often run out of memory for longer pieces than the ones tested. *Divenire*, the longest piece analyzed, took around 25 seconds to process on a 2.5 GHz quad core PC and needed almost 1 GB of memory to complete.

- The proposed algorithm can only handle monophonic melodies played by the right hand (only one finger pressing a key at a time). The method can be improved to deal with some interleaved chords, but polyphonic melodies which use both hands to play would present a big challenge.

6 Conclusion

The developed algorithm generated fingerings that in most cases were close enough to experienced pianists' choice of fingering. In the cases this was not true, adapting parameters of the reach table to the actual size of the players' hand and weighting the rules differently would give a more accurate output. The addition of the pause rule allowed the algorithm to generate fingerings for a larger set of pieces, while still giving an accurate result. The algorithm developed shows that fingerings can be generated by computers suitable for real world use.

7. References

Parncutt, R., Sloboda, J., Clarke, E., Raekallio, M. and Desain, P. (1997)“An ergonomic model of keyboard fingering for melodic fragments,” *Music Perception*, Vol.14, pp. 341 - 382

Yonebayashi, Y., Kameoka, H. and Sagayama,S.(2010)“Automatic Decision of Piano Fingering Based on Hidden Markov Models”