

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

JOÃO VÍCTOR RAMOS

**DIFERENTES ABORDAGENS EVOLUTIVAS APLICADAS NO
PROCESSO DE TRANSCRIÇÃO AUTOMÁTICA DE PARTITURAS
MUSICAIS EM TABLATURAS**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO

2016

JOÃO VÍCTOR RAMOS

**DIFERENTES ABORDAGENS EVOLUTIVAS APLICADAS NO
PROCESSO DE TRANSCRIÇÃO AUTOMÁTICA DE PARTITURAS
MUSICAIS EM TABLATURAS**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR como requisito parcial para a obtenção do título de “Mestre Profissional em Informática”.

Orientador: Prof. Dr. Danilo Sipoli Sanches

Coorientador: Prof. Dr. Carlos N. Silla Jr.

CORNÉLIO PROCÓPIO

2016

Dados Internacionais de Catalogação na Publicação

- R175 Ramos, João Víctor
Diferentes abordagens evolutivas aplicadas no processo de transcrição automática de partituras musicais em tablaturas / João Víctor Ramos. – 2015.
100 p. : il. color. ; 30 cm
- Orientador: Danilo Sipoli Sanches.
Coorientador: Carlos Nascimento Silla Junior.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. Cornélio Procópio, 2015.
Bibliografia: p. 85-89.
1. Algoritmos computacionais. 2. Música por computador. 3. Tablatura (Notação musical). 4. Informática – Dissertações. I. Sanches, Danilo Sipoli, orient. II. Silla Junior, Carlos Nascimento, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. IV. Título.

CDD (22. ed.) 004



Título da Dissertação Nº 13:

“Diferentes Abordagens Evolutivas Aplicadas no Processo de Transcrição Automática de Partituras Musicais em Tablaturas”.

por

João Vítor Ramos

Orientador: **Prof. Dr. Danilo Sipoli Sanches**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 13h e 30min do dia 11 de dezembro de 2015. O trabalho foi _____ pela Banca Examinadora, composta pelos professores:

Prof. Dr. Danilo Sipoli Sanches
(Presidente – UTFPR-CP)

Prof. Dr. Carlos Nascimento Silla Junior
(Coorientador – UTFPR-CP)

Prof. Dr. André Yoshiaki Kashiwabara
(UTFPR-CP)

Prof. Dr. Wesley Romão
(UEM-PR)

Visto da coordenação:

Carlos Nascimento Silla Junior
Coordenador do Programa de Pós-Graduação em Informática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico este trabalho à minha esposa e aos nossos filhos que virão.

AGRADECIMENTOS

Primeiramente agradeço a Deus, que é o princípio e o fim de todas as coisas. Aquele que me sustentou e me levantou nos momentos mais difíceis deste trabalho. Agradeço também à minha família, meus pais, minhas irmãs e em especial à minha esposa Luciana a qual pacientemente me compreendeu e foi meu refúgio nos momentos em que quis desistir. Também ao meu orientador Prof. Dr. Danilo Sipoli Sanches e ao meu coorientador Prof. Dr. Carlos N. Silla Jr. por terem me guiado durante o decorrer de todo o trabalho já executado, fazendo-me acreditar que tudo funcionava e era possível de ser realizado. Aos meus colegas de sala, à Secretaria do curso, à Coordenação e a todos os que, de algum modo, contribuíram para a realização desta pesquisa.

“Aprendi o segredo de me sentir contente em todo lugar e em qualquer situação, quer esteja farto ou com fome, quer tenha abundância e a padecer necessidade. Pois tudo posso naquele que me conforta.”
(Filipenses 4,12b-13)

RESUMO

RAMOS, João Víctor. DIFERENTES ABORDAGENS EVOLUTIVAS APLICADAS NO PROCESSO DE TRANSCRIÇÃO AUTOMÁTICA DE PARTITURAS MUSICAIS EM TABLATURAS. 104 f. Dissertação de Mestrado – Programa de Pós-graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

A tablatura é um formato de notação musical alternativo para instrumentos de corda. Ela contém a posição exata (corda e casa) que necessita ser tocada para produzir um som desejado. A conversão da notação musical padrão (partitura) para uma tablatura de violão, também conhecida como transcrição, não é um processo simplificado. A razão é que cada nota da notação musical padrão pode ser tocada em diferentes posições no braço do violão. Desta forma este problema pode ser classificada como um problema de otimização combinatório. Neste trabalho, foi empregado um estudo comparativo de diferentes algoritmos: A*, algoritmos genéticos (AG), algoritmos genéticos com uso de subpopulações (AG-SP), otimização por colônia de formigas (ACO), e evolução diferencial (ED). Heurísticas baseadas nas buscas locais 2-opt e 3-opt para as abordagens AG, AG-SP e ED também foram inclusas. Os resultados experimentais com um conjunto de dados de 87 músicas monofônicas demonstraram que os algoritmos ACO, AG-SP 2-opt e AG 2-opt apresentaram os melhores resultados. Por fim, cada abordagem foi estatisticamente comparada utilizando o teste ANOVA em conjunto com o pós-teste de Tukey (Tukey *post hoc*), onde as análises demonstraram que há diferenças significativas entre os algoritmos avaliados.

Palavras-chave: Algoritmos Evolutivos, Computação Musical, Transcrição de Tablatura

ABSTRACT

RAMOS, João Vítor. DIFFERENT EVOLUTIONARY APPROACHES APPLIED IN THE PROCESS OF AUTOMATIC TRANSCRIPTION OF MUSIC SCORES INTO TABLATRES. 104 f. Dissertação de Mestrado – Programa de Pós-graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2016.

A tablature is an alternative music notation format for string instruments. It contains the exact position (string and fret) which needs to be played to produce the desired sound. The conversion of the musical staff notation (sheet music) to a guitar tablature, also known as transcription, is not a straightforward process. The reason is that each note on the music notation format can be played in different positions of the guitar fretboard. So this problem can be classified as a combinatorial optimization problem. In this work, we have employed a comparative study of different algorithms: A*, genetic algorithms (AG), genetic algorithms with use of subpopulations (AG-SP), ant colony optimization (ACO) and differential evolution (ED). I was also included heuristics based on local search 2-opt and 3-opt in the approaches AG, AG-SP and ED. The experimental results with a dataset of 87 monophonic musics indicated that the approaches ACO, AG-SP 2-opt and AG 2-opt obtained the best results. Finally, each approach were statistically compared using the ANOVA with the Tukey *post hoc*, where the analysis indicated that there are significant differences between the evaluated algorithms.

Keywords: Evolutionary Algorithms, Computer Music, Guitar Tablature Transcription

LISTA DE FIGURAS

FIGURA 1	– Tábua Hurrita	13
FIGURA 2	– Hino a São João Batista	14
FIGURA 3	– Notação musical moderna	15
FIGURA 4	– Representação dos acidentes musicais	15
FIGURA 5	– O sistema de notação através de tablaturas	16
FIGURA 6	– Nota E4 sendo representada em um braço de um violão de 19 casas	17
FIGURA 7	– Possíveis posições para execução de 3 notas	18
FIGURA 8	– Grafo gerado para o fragmento de 3 notas	22
FIGURA 9	– Graduação das dificuldades	23
FIGURA 10	– Busca gulosa	24
FIGURA 11	– Busca com A*	25
FIGURA 12	– Pseudocódigo do algoritmo A*	26
FIGURA 13	– Movimento 2-opt	27
FIGURA 14	– Função com muitos máximos locais	29
FIGURA 15	– Exemplo de cruzamento uniforme	31
FIGURA 16	– Exemplo de cruzamento n-pontos	32
FIGURA 17	– Operador de mutação	32
FIGURA 18	– Visão geral do processo evolucionário de um algoritmo genético	34
FIGURA 19	– Aplicação dos operadores genéticos	35
FIGURA 20	– Caminhos percorridos pelas formigas no ACO	38
FIGURA 21	– Cruzamento na ED	42
FIGURA 22	– Fluxo do algoritmo ED clássico	42
FIGURA 23	– Exemplo do formato MusicXML	46
FIGURA 24	– Representação das notas no braço do violão	47
FIGURA 25	– Representação da distância entre duas posições no plano cartesiano	49
FIGURA 26	– Exemplo do cálculo usando a distância euclidiana	49
FIGURA 27	– Níveis das notas de uma música	50
FIGURA 28	– Expansão do nó raiz	51
FIGURA 29	– Retorno do algoritmo para um nó já expandido	52
FIGURA 30	– Pseudocódigo do algoritmo genético tradicional	53
FIGURA 31	– <i>Crossover</i> Uniforme	54
FIGURA 32	– Processo de mutação	55
FIGURA 33	– Divisão em subpopulações	56
FIGURA 34	– Pseudocódigo do AG-SP	56
FIGURA 35	– Subpopulações ordenadas por <i>fitness</i> da porção	57
FIGURA 36	– Geração da população ponderada	58
FIGURA 37	– Torneio entre os indivíduos das subpopulações	59
FIGURA 38	– Operador de <i>crossover</i> aplicado nas notas da música	60
FIGURA 39	– Análise da primeira subpopulação na estratégia de migração	61
FIGURA 40	– Migração dos indivíduos finalizada	61
FIGURA 41	– Estratégia de recomposição da população ponderada	62
FIGURA 42	– Possíveis caminhos percorridos pelas formigas.	63

FIGURA 43	– Pseudocódigo proposto para o ED com lista de movimentos	65
FIGURA 44	– Geração do conjunto de movimentos para abordagem proposta	66
FIGURA 45	– Exemplo de aplicação da escalar F	66
FIGURA 46	– Exemplo de geração da lista de movimentos para a abordagem proposta	67
FIGURA 47	– Exemplo da aplicação da lista de movimentos	67
FIGURA 48	– Solução encontrada para uma música de 5 notas	68
FIGURA 49	– Troca realizada pelo 2-opt	69
FIGURA 50	– Pseudocódigo da busca local 2-opt e 3-opt	70
FIGURA 51	– Convergência da música Katie's Lilt (grupo 1)	74
FIGURA 52	– Convergência da música Smiling Bride, The (grupo 2)	74
FIGURA 53	– Convergência da música Fifty Five (grupo 3)	75
FIGURA 54	– Convergência da música Suleman's Kopanitsa (grupo 4)	75
FIGURA 55	– Trecho de uma transcrição realizada pelo MuseScore	78
FIGURA 56	– Alternativas para posições melhor transcritas	79
FIGURA 57	– Comparativo entre as melhores abordagens e o MuseScore	80
FIGURA 58	– Comparação da tessitura entre o violão e o piano	83
FIGURA 59	– Regiões de concentração das notas no braço do violão	84

LISTA DE TABELAS

TABELA 1	– Exemplo da aplicação de <i>ranking</i>	30
TABELA 2	– Parâmetros propostos por Tuohy e Potter (2005)	35
TABELA 3	– Significado das marcações no MusicXML	47
TABELA 4	– Divisão das músicas em grupo de notas	71
TABELA 5	– Parâmetros utilizados na análise combinatória	72
TABELA 6	– Parâmetros utilizados nos experimentos com os algoritmos genéticos ..	72
TABELA 7	– Parâmetros utilizados nos experimentos com o ACO	73
TABELA 8	– Parâmetros utilizados nos experimentos com a evolução diferencial	73
TABELA 9	– Hardware e software utilizados	73
TABELA 10	– Ranqueamento dos resultados baseado no <i>fitness</i> dos algoritmos	76
TABELA 11	– Ranqueamento dos melhores métodos	76
TABELA 12	– Diferença significativa entre os algoritmos	77
TABELA 13	– Músicas utilizadas para exploração do problema	90
TABELA 14	– Resultados de <i>fitness</i> dos experimentos do grupo 1	94
TABELA 15	– Resultados do tempo (ms) dos experimentos do grupo 1	94
TABELA 16	– Resultados de <i>fitness</i> dos experimentos do grupo 2	95
TABELA 17	– Resultados do tempo (ms) dos experimentos do grupo 2	96
TABELA 18	– Resultados de <i>fitness</i> dos experimentos do grupo 3	97
TABELA 19	– Resultados do tempo (ms) dos experimentos do grupo 3	98
TABELA 20	– Resultados de <i>fitness</i> dos experimentos do grupo 4	99
TABELA 21	– Resultados do tempo (ms) dos experimentos do grupo 4	99

LISTA DE SIGLAS

2-opt	2-ótimo (do original <i>2-Optimal</i>)
3-opt	3-ótimo (do original <i>3-Optimal</i>)
A*	(Lê-se: A-estrela) Algoritmo de busca em grafos
a.C	Antes de Cristo
ACO	Otimização por colônia de formigas (do original <i>Ant colony optimization</i>)
AG	Algoritmo genético
AG-SP	Algoritmo genético baseado em subpopulações
ED	Evolução diferencial (do original <i>Differential Evolution (DE)</i>)
ms	Milisegundos
MusicXML	Linguagem de marcação extensível para música (do original <i>Music Extensible Markup Language</i>)
NIFF	Formato de arquivo de notação de intercâmbio (do original <i>Notation Interchange File Format</i>)
NP-C	Tempo polinomial não determinístico completo (do original <i>Nondeterministic polynomial time - complete</i>)
OBX	Cruzamento baseado em ordem (do original <i>Order-Based Crossover</i>)
PBX	Cruzamento baseado em posição (do original <i>Position-Based Crossover</i>)
PCV	Problema do caixeiro viajante
SPEA-2	Algoritmo evolucionário de força de pareto 2 (do original <i>Strength Pareto Evolutionary Algorithm 2</i>)
TSD	Diferença significativa de Tukey (do original <i>Tukey Significant Difference</i>)
XML	Linguagem de marcação extensível (do original <i>Extensible Markup Language</i>)

SUMÁRIO

1 INTRODUÇÃO	13
1.1 A EVOLUÇÃO DA NOTAÇÃO MUSICAL	13
1.2 A NOTAÇÃO MUSICAL MODERNA	14
1.3 A TABLATURA	16
1.4 O PROBLEMA DA TRANSCRIÇÃO MUSICAL	17
1.5 OBJETIVOS	19
1.5.1 Objetivo Geral	19
1.5.2 Objetivos Específicos	19
1.6 ESTRUTURA DO TRABALHO	20
2 REVISÃO DA LITERATURA	21
2.1 O USO DA BUSCA EM GRAFOS	21
2.1.1 Busca Heurística	23
2.1.2 Busca Local 2-opt e 3-opt	27
2.2 ALGORITMOS GENÉTICOS	28
2.2.1 Algoritmos Genéticos Aplicados ao Processo de Transcrição	34
2.2.2 Algoritmos Genéticos Baseados em Subpopulações	37
2.3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS	38
2.4 EVOLUÇÃO DIFERENCIAL	40
2.4.1 Evolução Diferencial com Lista de Movimentos	43
2.5 ANÁLISES ESTATÍSTICAS	44
3 ABORDAGEM PROPOSTA	45
3.1 REPRESENTAÇÃO DO PROBLEMA	45
3.2 AVALIAÇÃO DA TRANSCRIÇÃO	48
3.3 ABORDAGEM UTILIZANDO O ALGORITMO A*	50
3.4 ABORDAGEM UTILIZANDO O ALGORITMOS GENÉTICOS	52
3.4.1 Abordagem Utilizando o AG-SP	55
3.5 ABORDAGEM UTILIZANDO O ACO	62
3.6 ABORDAGEM UTILIZANDO A ED COM LISTA DE MOVIMENTOS	63
3.7 ABORDAGEM UTILIZANDO A BUSCA LOCAL 2-OPT E 3-OPT	68
4 RESULTADOS E DISCUSSÕES	71
4.1 BASE DE DADOS E CONFIGURAÇÕES GERAIS	71
4.2 RESULTADOS EXPERIMENTAIS	73
4.3 RANKING POR DOMINÂNCIA	75
4.4 COMPARAÇÃO COM O PROGRAMA MUESCORE	78
5 CONCLUSÕES E TRABALHOS FUTUROS	81
5.1 CONCLUSÕES	81
5.2 PERSPECTIVAS FUTURAS	82
REFERÊNCIAS	85
Apêndice A – RELAÇÃO DE MÚSICAS UTILIZADAS	90
Apêndice B – TABELAS COM O RESULTADO DOS EXPERIMENTOS	93
Apêndice C – TRANSCRIÇÃO COMPLETA DA MÚSICA THE MEDIATOR	100

1 INTRODUÇÃO

1.1 A EVOLUÇÃO DA NOTAÇÃO MUSICAL

A música tem um importante papel para as pessoas. Segundo Ribeiro (2001), o filósofo e matemático grego Platão em uma de suas obras relata que a música é capaz de atingir mais profundamente a alma de um cidadão podendo até moldar a pessoa tanto para o bem quanto para o mal. A educação musical, se utilizada de forma correta, iria afastar os maus vícios, acalmar os irascíveis, bem como atrair a coragem, trazer ordem à alma e até mesmo justiça.

Manter músicas registradas em um formato em que seja possível sua posterior reprodução não é uma preocupação recente. Muitos sistemas de notação musical diferentes foram propostos ao longo dos séculos. Logo após o período da invenção da escrita, na Antiguidade, esta preocupação já atingia os antigos babilônios. As mais antigas formas de notação musical foram descobertas por Kilmer e Civil (1986) em tábuas de escrita cuneiforme em cidades da antiga Mesopotâmia. Uma destas tábuas datadas do século XIII a.C., e que pode ser observada na Figura 1, é considerada o mais velho trabalho de notação musical substancialmente completo do mundo encontrado até hoje.

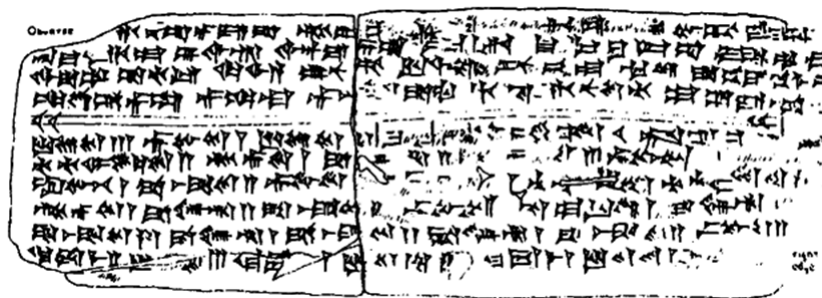


Figura 1: Tábua Hurrita.

Fonte: Adaptado de Duchesne-Guillemin (1984)

Conforme Otten (1910), os sistemas de notação musical foram evoluindo com o passar do tempo, impulsionados principalmente pelos mosteiros espalhados pela Europa no início da Idade Média. O monge beneditino italiano Guido d'Arezzo (991-1033) foi o fundador do que

hoje é considerado o padrão da música. Ele ensinou, através de um hino a São João Batista, a arte de cantar notas de uma escala através de sílabas: o solfejo. Hino que deu origem ao conjunto de notas musicais: Dó (*C*), Ré (*D*), Mi (*E*), Fá (*F*), Sol (*G*), Lá (*A*) e Si (*B*) e que pode ser observado na Figura 2. Este sistema de notação representa a entonação de uma melodia vocal que abrange uma única sílaba de canto.

Ut queant laxis
Resonare fibris
Mira gestorum
Famuli tuorum
Solve polluti
Labi reatum
Sancte Iohannes

Figura 2: Hino a São João Batista¹.

Fonte: Autoria própria.

1.2 A NOTAÇÃO MUSICAL MODERNA

A partir do século XVI, as notações musicais eventualmente evoluíram para a notação musical ocidental moderna utilizada hoje. A escrita musical atual é feita na partitura. Esta é formada por pentagramas. O pentagrama é um conjunto de 5 linhas horizontais com 4 espaços entre elas. Para o auxílio nas representações podem ser utilizadas linhas suplementares acima e abaixo do pentagrama. A Figura 3 representa este tipo de notação onde os itens em destaque representam: 1) as linhas e espaços da pauta, 2) as claves de Sol e de Fá, 3) uma nota musical, 4) um compasso e 5) uma das figuras da pausa.

As notas musicais são representadas através de símbolos em formato oval que, dependendo da posição em que estiverem escritas no pentagrama, indicarão os sons mais graves ou mais agudos. Quanto mais acima no pentagrama, mais agudas elas serão. O som de uma nota musical vibra a uma determinada frequência. Quando temos esta mesma nota musical sendo reproduzida com o dobro ou com a metade de sua frequência, dizemos que esta nota pertence a uma oitava diferente.

O nome de cada nota no pentagrama é definido pela clave musical, que é um símbolo colocado no início da pauta e que dá o seu nome à nota escrita em sua linha.

¹Tradução: Para que os servos possam, com suas vozes soltas, ressoar as maravilhas de vossos atos, limpa a culpa do lábio manchado, ó São João!



Figura 3: Notação musical moderna.

Fonte: Autoria própria.

Da mesma forma que encontramos as notas musicais, também encontramos figuras para representar as pausas. As pausas servem para representar o silêncio que deve haver em determinado período de tempo em que o instrumento não deverá ser tocado.

Cada nota ou pausa pode possuir um tempo de duração diferente, o qual é definido através de símbolos diferentes. Os tempos de duração de uma nota ou pausa variam de 1 a 1/64 de tempo e ficam organizados dentro de um compasso. O compasso é uma forma de medir e dividir em grupos os sons as notas de uma composição musical.

Em determinadas partituras pode-se encontrar símbolos utilizados na notação musical que quando indicados junto às notas musicais, fazem com que a mesma tenha sua altura modificada. Estes símbolos são chamados de acidentes musicais e podem ser visualizados na Figura 4.

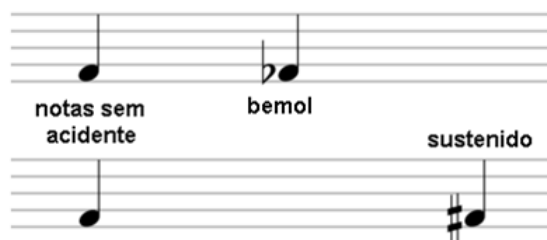


Figura 4: Representação dos acidentes musicais.

Fonte: Autoria própria.

Os acidentes musicais mais utilizados são o bêmol e o sustenido. O bêmol é responsável por abaixar a altura da nota que o segue em um semitom em relação à sua altura original, já o sustenido aumenta a altura da nota em um semitom em relação à sua altura original.

Bennett (1998) e Med (1996) apresentam outros símbolos e representações musicais como também elucidam de maneira mais aprofundada a teoria musical.

1.3 A TABLATURA

A tablatura é um sistema de notação musical utilizado em instrumentos de corda. O formato da tablatura se parece com o instrumento. Nela as linhas da pauta representam as cordas. Em cada linha são indicados os números que representam qual casa deve ser pressionada.

A Figura 5 apresenta uma comparação entre uma notação musical em partitura em contraste com a notação através do uso de uma tablatura para um instrumento de 6 cordas. É possível observar que as notas são representadas através de posições que devem ser pressionadas e tocadas para que seja reproduzida a nota. Além disso, percebemos que nem todos os elementos da música são representados pela tablatura. Informações sobre o tempo das notas, o ritmo, entre outros elementos, não são expressos na tablatura.

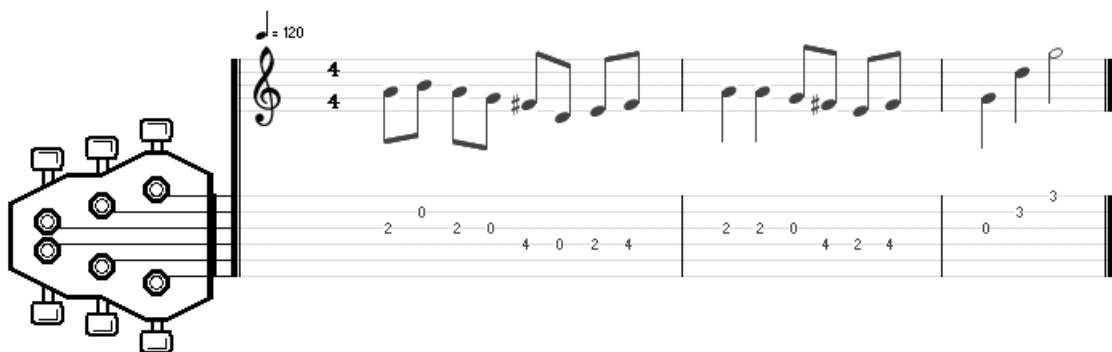


Figura 5: O sistema de notação através de tablaturas.

Fonte: Autoria própria.

De acordo com Radicioni (2006) o uso da tablatura vem sendo muito utilizado principalmente por músicos principiantes, nos casos em que se faz necessária uma forma simplificada de se representar a música a ser tocada.

A notação através de tablaturas, apesar de ser amplamente utilizada, não pretende substituir os sistemas de notação da música padronizados. Segundo Burlet (2013), estas formas de notação buscam passar por uma representação alternativa que é leve e facilmente compreensível.

Desta forma, o uso da tablatura em conjunto com a partitura permite que o músico tenha uma melhor visão da música a ser tocada, ora pela fácil cognição dada pela tablatura, ora pelos elementos adicionais dados pela partitura.

1.4 O PROBLEMA DA TRANSCRIÇÃO MUSICAL

Um violão clássico possui 6 cordas. O seu braço possui tamanhos variados, dependendo do número de casas que normalmente varia de 19 a 24. Há também violões com tamanhos personalizados.

O processo de se tocar uma nota em um instrumento de corda, mais especificamente num violão clássico, consiste basicamente em utilizar os dedos de uma das mãos para movimentar uma corda enquanto os dedos da outra mão pressionam a mesma corda em uma das casas do instrumento. É possível ainda reproduzir uma nota musical simplesmente movimentando as cordas, sem a necessidade de pressionar qualquer casa do instrumento. Caso se deseje construir um acorde, deve-se pressionar um conjunto de casas enquanto se movimenta um conjunto de cordas.

Apesar de a partitura ser o padrão mais utilizado para a escrita de músicas, ela não é tão intuitiva quanto uma tablatura para um músico que toca instrumentos de corda. O motivo se dá pelo fato de a tablatura conter a exata informação de qual casa e corda deve ser pressionada e tocada. Por outro lado o uso de partituras exige que o músico realize a transcrição, ou seja, decida em qual posição cada nota deverá ser tocada.

O processo de transcrição não é um processo tão simples. Uma nota representada em uma partitura pode ser tocada em média em quatro diferentes posições no braço do violão. No exemplo apresentado na Figura 6 podemos observar a nota Mi sendo reproduzida em cinco diferentes posições do violão.

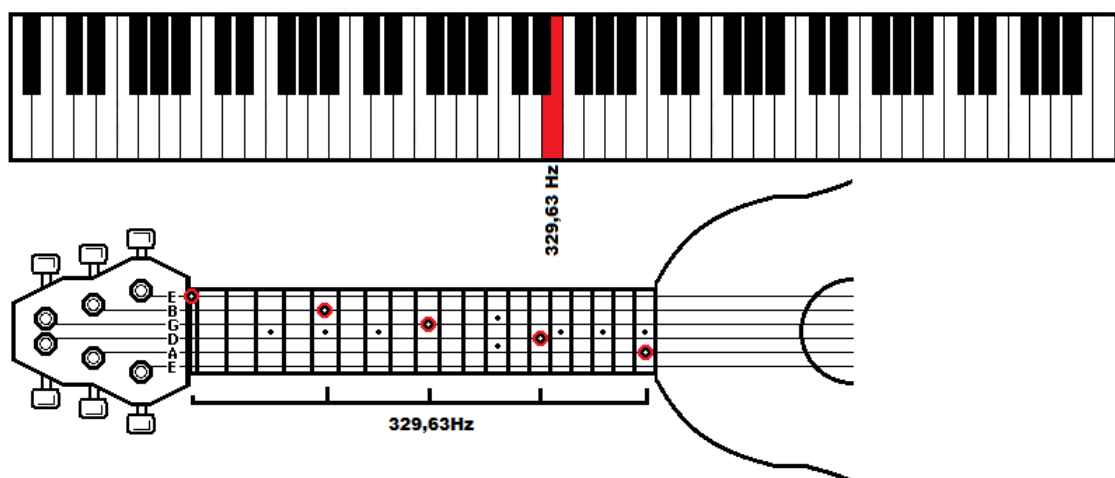


Figura 6: Nota E4 sendo representada em um braço de um violão de 19 casas.

Fonte: Autoria própria.

Esta transcrição pode ser classificada como um problema de otimização combinatório. Podemos representar uma nota a ser tocada através de uma tupla $\langle \text{corda}, \text{casa} \rangle$, onde os valores para a corda variam de 1 a 6 e os valores para casa de 0 a 19 sendo 0 a posição que indica que nenhuma casa deve ser pressionada. Desta forma se observarmos a Figura 6 podemos constatar que a nota E4 pode ser tocada em diferentes posições: $\langle 1,0 \rangle$, $\langle 2,5 \rangle$, $\langle 3,9 \rangle$, $\langle 4,14 \rangle$, $\langle 5,19 \rangle$.

Cada música é composta por uma sequência de notas, sendo que a transição de uma nota para outra pode ter múltiplos caminhos. Consideramos um pequeno trecho de 3 notas de uma música: Lá na quarta oitava, Fá na quarta oitava e Sol na quarta oitava (A4, F4 e G4). Neste caso, conforme a Figura 7, nos deparamos com 80 diferentes tablaturas. Este exemplo mostra que o número de tablaturas possíveis cresce exponencialmente quando o número de notas de uma música aumenta.

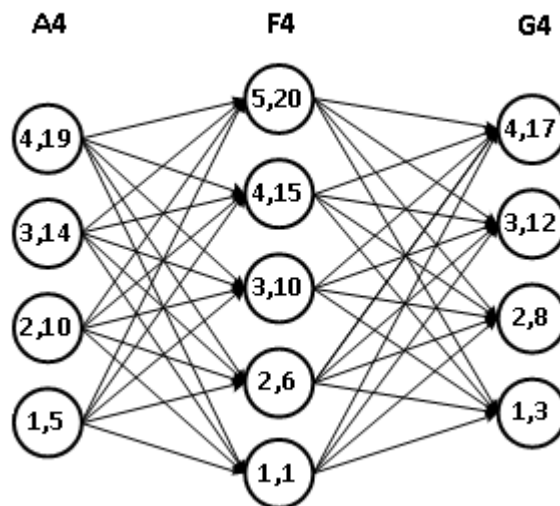


Figura 7: Possíveis posições para execução de 3 notas.

Fonte: Autoria própria.

Radicioni et al. (2005) também relata a complexidade desta transcrição. Ele descreve que uma mesma nota pode ser, em média, mapeada em até quatro posições diferentes no braço de um violão. Sendo assim um número n de notas pode gerar até 4^n diferentes posições. As possibilidades crescem ainda mais se for considerado que se pode utilizar quatro dedos diferentes da mão: indicador, médio, anelar e mínimo para reproduzir uma nota. A técnica de transcrição é um processo cognitivo complexo. Segundo Radicioni et al. (2004), além da análise filológica, ou seja, a compreensão e interpretação das notas de acordo com suas características estruturais, existem também restrições físicas apresentadas pelo instrumento musical e restrições bio-

mecânicas impostas pelas partes do corpo do músico que interage com o instrumento.

O presente estudo tem como foco a análise de diferentes abordagens para o processo de transcrição de músicas monofônicas. Atualmente a literatura carece de investigações que utilize meta-heurísticas mais recentes, tais como a otimização por colônia de formigas, também conhecido por *Ant Colony Optimization*, ou simplesmente ACO, o algoritmo genético com uso de subpopulações (AG-SP), a evolução diferencial (ED) além da inclusão de técnicas de busca local 2-opt e 3-opt.

Por causa disto, é proposto neste trabalho uma análise detalhada entre diversas abordagens evolutivas com diferentes aspectos e mecanismos de exploração das soluções no espaço de busca. Com o intuito de verificar se há ou não diferenças significativas entre as soluções obtidas, pretende-se analisar, através de testes estatísticos, estas novas abordagens em relação às abordagens clássicas que são utilizadas comumente na literatura para o tratamento do problema de transcrição automática de partituras em tablaturas.

Devido a fatores limitantes, tais como o tempo necessário para compor uma base de dados polifônica com uma quantidade de músicas grande o suficiente para a análise dos algoritmos e a dificuldade de se encontrar base de dados polifônicas completas e confiáveis em formato digital, este trabalho focará apenas processo de transcrição de músicas monofônicas. Desta forma, não será considerado outras características como o uso dos dedos, acordes, efeitos e sonoridade. A inclusão destas características será objeto de pesquisa futura, usando como base os algoritmos já desenvolvidos.

1.5 OBJETIVOS

1.5.1 OBJETIVO GERAL

Investigar diferentes características entre diversas abordagens evolutivas para o tratamento do problema de transcrição automática de partitura musical para uma tablatura.

1.5.2 OBJETIVOS ESPECÍFICOS

- Obter a transcrição de uma partitura musical para uma tablatura minimizando o deslocamento da mão e dos dedos;
- Utilizar a busca heurística como base para o processo de transcrição musical de partituras;
- Aplicar abordagens meta-heurísticas mais recentes para a transcrição de partituras em

tablaturas;

- Adaptar técnicas de busca local nos algoritmos evolutivos desenvolvidos.

1.6 ESTRUTURA DO TRABALHO

O trabalho está estruturado da seguinte forma: no capítulo 2 é apresentada a fundamentação teórica onde se faz presente a revisão da literatura; no capítulo 3, as abordagens propostas; no capítulo 4, os resultados experimentais obtidos; no capítulo 5, as conclusões e perspectivas futuras e por fim as referências bibliográficas e os apêndices.

2 REVISÃO DA LITERATURA

Neste capítulo será realizado uma visão geral sobre algumas das técnicas heurísticas e evolutivas utilizadas na literatura que serão objetos para a solução do problema de transcrição. Também serão abordados os estudos já realizados no processo de transcrição de uma partitura musical em uma tablatura.

2.1 O USO DA BUSCA EM GRAFOS

O processo de transcrição de uma partitura para uma tablatura de violão tem sido explorado por alguns pesquisadores. Radicioni et al. (2004) realizando uma extensão do trabalho de Sayegh (1989) que utiliza o paradigma de melhor caminho, sugere que o problema seja mapeado em um grafo onde cada vértice representa uma posição mapeada. Estas posições contém informações sobre a corda, a casa e o dedo utilizados de forma que cada lado do grafo conecta duas posições mapeadas adjacentes.

Heijink e Meulenbroek (2002) avaliaram os fatores biomecânicos que influenciam na complexidade e qualidade da reprodução musical, tais como o posicionamento das mãos no braço do violão, o distanciamento dos dedos e o reposicionamento das mãos no decorrer das notas que são tocadas. Em seus estudos, demonstraram que é preferível evitar o reposicionamento das mãos e o distanciamento dos dedos, bem como que é mais desejável o uso das casas mais próximas a cabeça do violão.

A transição de uma posição mapeada para a próxima posição mapeada possui um peso. Este peso, estima a complexidade da transição entre as notas gerando informações que irão auxiliar no processo de busca pelo grafo.

A estrutura em níveis deste grafo propõe que cada nível representa uma nota da partitura. Desta forma cada nível está conectada a outra por diferentes caminhos. Sendo assim, o processo é modelado como um problema de busca no grafo onde se deve encontrar um caminho adequado para ser trilhado entre estas níveis através de um algoritmo de busca.

A figura 8 apresenta um exemplo de grafo proposto por Radicioni et al. (2004), onde cada vértice possui uma posição mapeada contendo o número da corda, da casa e do dedo utilizado para pressioná-la. O grafo apresenta as possíveis transições entre as notas Fá da segunda, quarta e terceira oitavas.

Diferente de Sayegh (1989), ao invés de encontrar um dedilhado ótimo considerando a música inteira, Radicioni et al. (2004) foca em intenções musicais relacionadas ao processo de dedilhar. Ele acredita que os músicos provavelmente tendem a quebrar a música em frases, temas ou motivos. Isto se dá porque o processo de aprendizado se torna mais simplificado, e o processo de dedilhado mais otimizado.

Com base nisto, o modelo computacional para o processo de conversão realizado em seu trabalho divide a música em vários segmentos e com isto gera os grafos para os segmentos das músicas e encontra os menores caminhos em cada grafo.

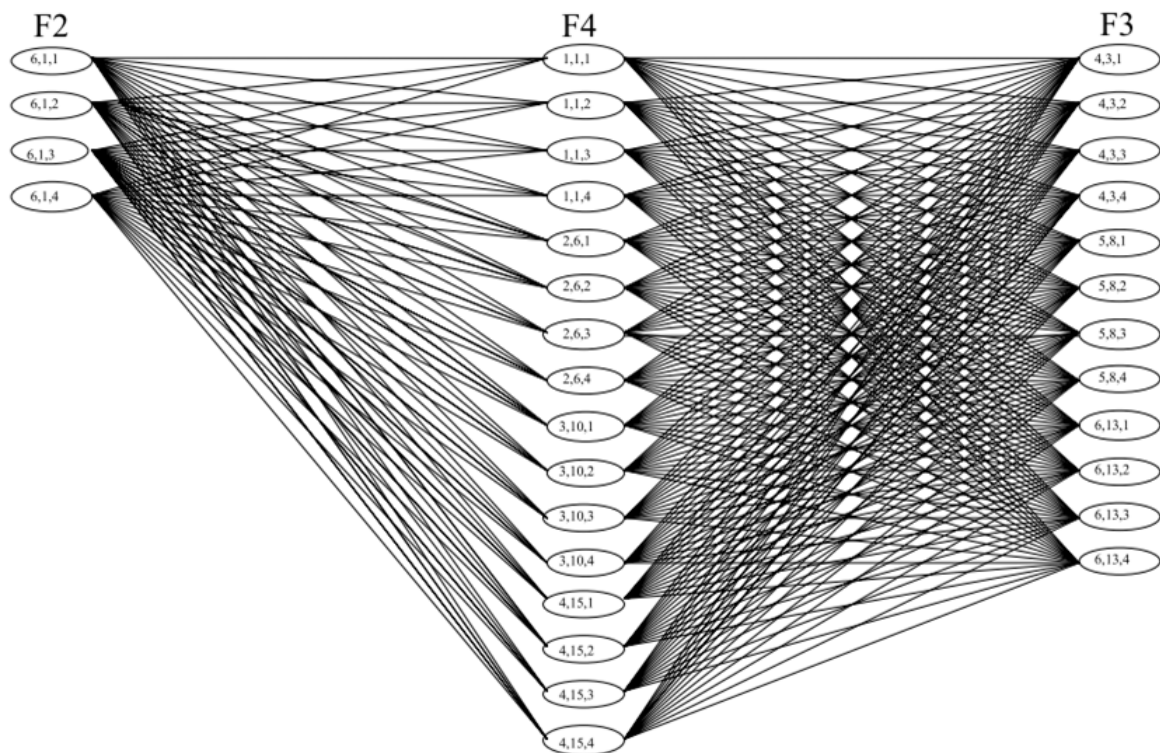


Figura 8: Grafo gerado para o fragmento de 3 notas.

Fonte: Radicioni et al. (2004)

Em outros trabalhos, Radicioni et al. (2005), sustentados pelo trabalho de Heijink e Meulenbroek (2002), também propõe que as dificuldades são estimadas através da movimentação das mãos pelo braço do violão durante a execução das notas da música. Ele assume a existência de diferentes tipos de dificuldades onde o movimento horizontal e vertical possui complexida-

des distintas.

Os movimentos horizontais são mais complexos e exigem o reposicionamento das mãos, já os verticais são menos complexos e exigem apenas o deslocamento dos dedos. A estimativa do peso é calculada levando-se em consideração estes dois tipos de movimentos. Desta forma o peso entre duas posições diferentes é atribuído através da soma linear entre estes dois tipos de dificuldades.

Outros fatores que influenciam as dificuldades horizontais e verticais são a distância entre os dedos e a localização da casa no braço do violão. Uma distância confortável entre os dedos exige o mínimo de esforço possível do músico. Esta distância é encorajada através da atribuição de um peso menor. Por outro lado, caso a transição de uma casa a outra tender a uma casa mais longe da cabeça do violão, esta é desencorajada uma vez que, conforme nos distanciarmos da pestana, as cordas vão ficando progressivamente mais elevadas em relação às trastes. Isto atua no aumento da dificuldade decorrente de pressionar estas posições (RADICIONI et al., 2005).

A partir destes cálculos, é possível encontrar os caminhos que exigem esforços maiores e menores entre as camadas do grafo. A figura 9 expressa a distância máxima, a distância mínima e a distância confortável. Caso este limite seja ultrapassado, o reposicionamento das mãos se faz necessário.

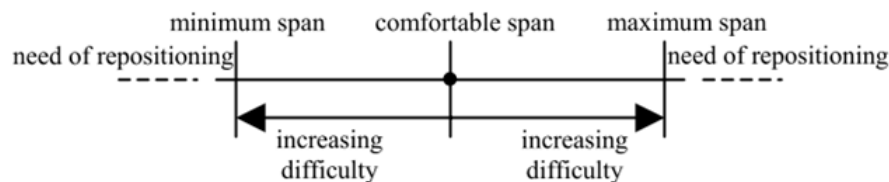


Figura 9: Graduação das dificuldades.

Fonte: Radicioni et al. (2004)

2.1.1 BUSCA HEURÍSTICA

Em muitos casos as estratégias de busca sem informações podem ser ineficientes devido ao fato de não se possuir um conhecimento específico do problema tratado. A busca com informação, ou busca heurística, utiliza um conhecimento específico sobre o problema para encontrar soluções de forma mais eficiente do que os métodos de busca sem informação (RUSSELL et al., 2004).

Na busca heurística uma função de avaliação é utilizada para estimar o quanto determi-

nado estado é desejável. Uma das estratégias de busca heurística é a busca pela melhor escolha, conhecida também como busca gulosa (RUSSELL et al., 2004).

A busca gulosa é uma técnica que, em uma árvore de pesquisa, tenta expandir o nó mais próximo da meta, pois acredita que isso levará à solução de maneira mais rápida. Deste modo, a busca gulosa utiliza apenas a função heurística como forma de avaliação. Sua função é apresentada na equação 1, onde $f(x)$ representa a função de avaliação e $h(x)$ representa o valor da heurística para chegar até o objetivo.

$$f(x) = h(x) \quad (1)$$

Um caso clássico que explica esta busca é determinar o menor caminho entre as cidades de forma que possamos traçar uma rota da cidade de Arad até Bucareste o que pode ser visualizado na Figura 10. Para este exemplo, supomos como função heurística a distância em linha reta até a cidade de destino Bucareste.

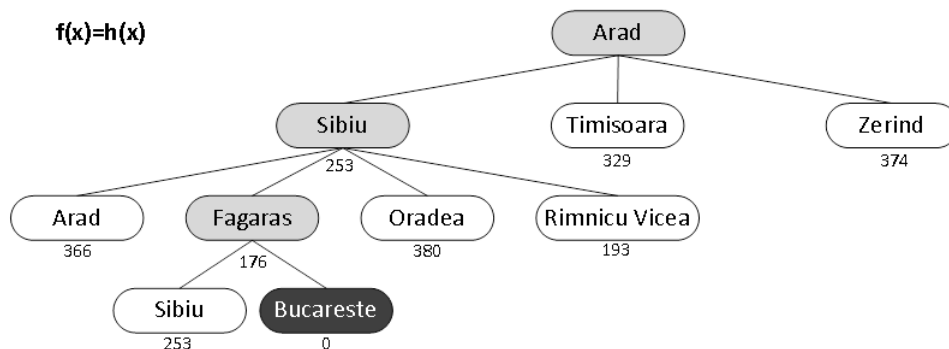


Figura 10: Busca gulosa.

Fonte: Adaptado de Russell et al. (2004)

Ao expandir os nós pela árvore de busca e traçar a rota para o caminho até Bucareste, percebe-se que a busca gulosa inicialmente opta pela cidade de Sibiu, pois possui 253 como a menor função de avaliação para o primeiro nível de expansão dos nós. Em seguida a escolha é pela cidade de Fagaras escolhida através da função com valor 176 e por fim Bucareste que possui valor 0 para sua função, uma vez que este é o objetivo final.

Este método de busca não é ótimo e é incompleto, pois segue o melhor caminho ponderando apenas o estado atual sem levar em consideração a possibilidade de caminhos melhores até o objetivo, apesar de em alguns pontos serem piores na árvore de busca. Além disso, pode levar a falsos indícios e cair em um caminho infinito e nunca testar novas possibilidades (RUSSELL et al., 2004).

A forma mais conhecida de busca por melhor escolha é o algoritmo A* (HART et al., 1968). Diferente da busca gulosa, ele utiliza como função de avaliação a combinação do cálculo do valor para alcançar cada nó, representado por $g(x)$, com o valor da heurística para ir do nó até o objetivo, representado por $h(x)$ (RUSSELL et al., 2004). A equação 2 apresenta a composição desta função de avaliação.

$$f(x) = g(x) + h(x) \quad (2)$$

Isto torna a estratégia ao mesmo tempo ótima e completa, desde que a função heurística seja admissível satisfazendo as condições de busca e nunca superestimando o custo para alcançar o objetivo. O algoritmo A* é um ramo dos algoritmos de busca com informação e exploração que utilizam estimativa da escolha do caminho não só o custo para chegar a um nó, mas também uma estimativa da distância ao destino (RUSSELL et al., 2004).

A Figura 11 representa o exemplo clássico da literatura proposto por Russell et al. (2004) que tem como objetivo encontrar o caminho mais curto entre a cidade de Arad e Bucareste.

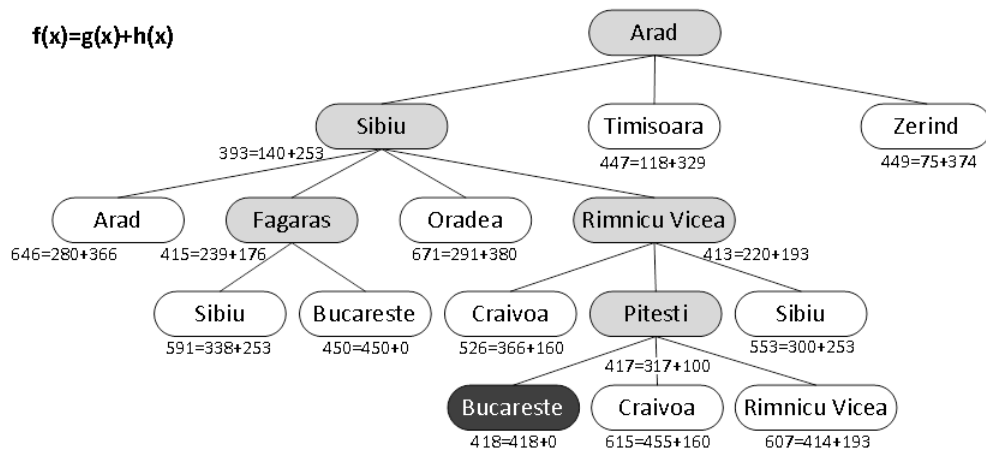


Figura 11: Busca com A*.

Fonte: Adaptado de Russell et al. (2004)

O algoritmo A* expande nó a nó em busca do melhor caminho levando em consideração não só a heurística para se dirigir de uma cidade a outra, mas também o custo da distância em linha reta de uma cidade até Bucareste.

A implementação do algoritmo A* para a conversão consiste na criação de um estado objetivo, ou seja, o estado ao qual se deseja chegar ao término da busca. A partir deste estado será gerado um grafo que parte do estado inicial até alcançar o seu estado objetivo. Neste

sentido, exploram-se os melhores nós que ainda não foram explorados. Após isso, deve-se guardar a informação de quais nós já foram expandidos para que não seja necessário expandi-los novamente. Uma das formas de se realizar este processo é com o uso de duas listas: uma para os nós já expandidos, ou seja, fechados e outra para os nós ainda não expandidos, ou abertos. A Figura 12 contém um pseudocódigo proposto para a implementação deste algoritmo.

```

1  Crie um nó contendo o estado objetivo;
2  Crie um nó contendo o estado inicial;
3  Coloque o nó do estado objetivo em uma lista de abertos;
4  enquanto a lista de abertos não estiver vazia faça
5      Retire o nó que tiver menor f da lista de abertos e chame-o de nó atual;
6      se se o nó atual é o mesmo estado do nó objetivo então
7          encontramos a solução;
8          break;
9      senão
10         Gere cada estado (nó sucessor) que vier após o nó atual;
11         para cada nó sucessor do nó atual faça
12             Defina o custo do nó sucessor para que seja o custo do nó atual mais o
13             custo para chegar até o nó sucessor a partir do nó atual;
14             Encontre o nó sucessor na lista de abertos;
15             se nó sucessor estiver na lista de abertos, mas o existente for melhor
16             então
17                 descarte este nó sucessor;
18                 continue;
19             fim
20             se nó sucessor estiver na lista de fechados, mas o existente for melhor
21             então
22                 descarte este nó sucessor;
23                 continue;
24             fim
25             Remova as ocorrências do nó sucessor da lista de abertos e fechados;
26             Defina o nó atual como pai do nó sucessor;
27             Defina o h para ser a distância estimada ao estado objetivo (utilizando
28             a função heurística);
29             Adicione o nó sucessor na lista de abertos;
30         fim
31     Adicione o nó atual na lista de fechados;
32 fim

```

Figura 12: Pseudocódigo do algoritmo A*.

Fonte: Adaptado de Heyes-Jones (2013).

Aplicados ao processo de transcrição de partituras em tablaturas de violão, Burllet e Fujinaga (2013) estendem os estudos de Sayegh (1989) implementando a composição de grafos

em conjunto com o algoritmo A*, proposto por Hart et al. (1968), a fim de buscar um melhor arranjo de uma sequência de notas e acordes. O algoritmo A* é uma das formas mais conhecidas de busca pela melhor escolha. Ela avalia os nós, combinando o custo do vértice inicial até o vértice final, além de combinar uma heurística que calcula o custo de ir até o nó objetivo (RUSSELL et al., 2004).

Para estimar o custo de transição de cada lado, Burlet e Fujinaga (2013) utiliza a diferença absoluta entre a casa de origem e a casa de destino. Já para casos onde a próxima casa é um acorde, o cálculo é realizado através do valor absoluto da média entre as distâncias das casas do acorde subtraída da casa de origem.

2.1.2 BUSCA LOCAL 2-OPT E 3-OPT

As técnicas de busca local 2-opt e 3-opt surgiram para solucionar o problema do caixeiro viajante (PCV) (CROES, 1958) (LIN, 1965). O PCV trata de um problema em que, dado um conjunto de cidades, cada cidade deverá ser visitada apenas uma vez e ao término retornar à cidade inicial. O objetivo é que seja possível encontrar o menor percurso (RUSSELL et al., 2004).

O método de busca local 2-opt proposto por Croes (1958) a partir de uma sugestão de Flood (1956), baseia-se na quebra de duas arestas de uma solução a fim de que estas arestas possam se reconectar em outros possíveis vértices. Desta forma, dado um percurso encontrado, inverte-se duas arestas do circuito para formar um novo percurso factível e melhor. O processo continua até que não haja mais melhorias. A Figura 13 ilustra esta troca denominada movimento 2-opt (HELSCAUN, 2000).

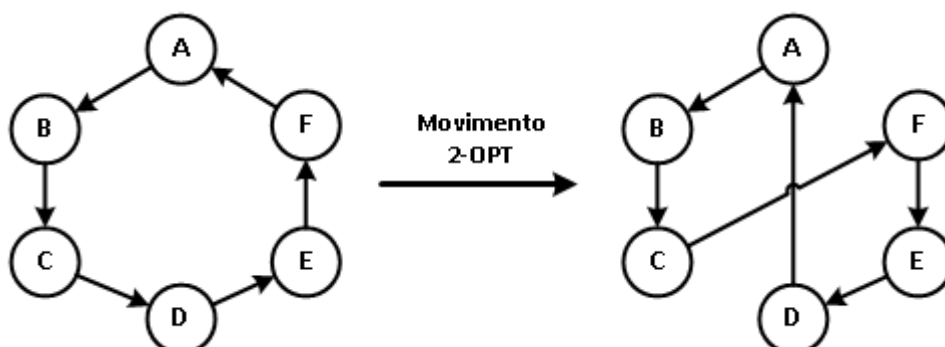


Figura 13: Movimento 2-opt.

Fonte: Autoria própria.

Este método foi aperfeiçoado por Lin (1965) propondo a metodologia 3-opt onde o

número de arestas quebradas e reconectadas é de três. Posteriormente desenvolve um algoritmo altamente efetivo para solução dos PCV simétricos onde o número de arestas a serem trocadas é variável a cada iteração do algoritmo (LIN; KERNIGHAN, 1973) (HELSGAUN, 2000).

2.2 ALGORITMOS GENÉTICOS

Darwin (1859) conclui que, quanto melhor um indivíduo se adapta ao seu meio, maior são suas chances de sobreviver, reproduzir e gerar descendentes. Este conceito é base da construção de um algoritmo genético que é um ramo dos algoritmos evolucionários. Usando operadores de seleção, permutação e mutação, e analisando o desempenho de cada indivíduo, Holland (1975) dá início a estes métodos de otimização que buscam encontrar soluções cada vez melhores a partir da evolução de cada indivíduo.

Devido ao fato de estes algoritmos serem muito dependentes de fatores probabilísticos (estocásticos), isto faz com que haja dificuldades em se repetir seus resultados com perfeição. São heurísticas que não têm garantia sobre a qualidade de seus resultados, apesar de habitualmente tenderem a encontrar uma solução ótima ou até ficarem muito próximos dela (LINDEN, 2008).

Técnicas de busca e otimização utilizando algoritmos genéticos apresentam um espaço de busca. Neste espaço são encontradas todas as possíveis soluções para o problema. As técnicas de busca e otimização então são aplicadas ao espaço de busca a fim de se encontrar uma solução ótima ou então uma solução satisfatória para o problema a ser otimizado.

Para que seja possível verificar se uma solução é boa, utiliza-se uma função de aptidão, também conhecida como função de *fitness*. Esta função é responsável por dar uma nota à solução encontrada quantificando-a e qualificando-a. A partir desta nota pode-se verificar se uma solução é melhor que outra.

A busca por uma melhor solução nem sempre é encontrada de maneira fácil, uma vez que o espaço de busca poderá apresentar vários pontos que maximizam o valor de uma função, porém não apresentam de fato o maior valor que a função pode atingir. Estes pontos são denominados máximos locais e em muitos casos acabam se tornando armadilhas em busca do maior valor que a função poderia atingir: o máximo global.

A Figura 14 apresenta um exemplo onde o espaço de busca possui vários máximos locais e apenas um único máximo global.

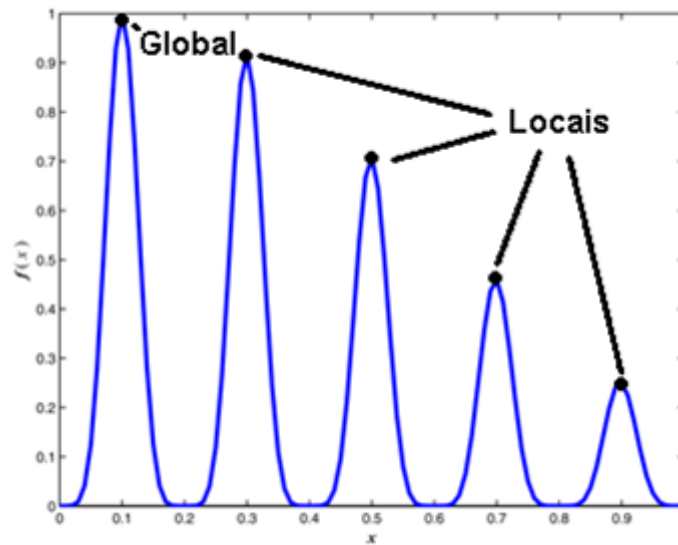


Figura 14: Função com muitos máximos locais.

Fonte: Adaptado de Yu e Gen (2010)

Antes de iniciar as etapas de um algoritmo genético, devemos encontrar uma maneira de demonstrar as informações de nosso problema de forma que possa ser computacionalmente tratada. A representação dos dados é feita através da codificação dos dados em estruturas denominadas genes. Um conjunto de genes formará um cromossomo. O cromossomo possui um conjunto de valores que representa uma possível solução para o problema a ser otimizado (LACERDA; CARVALHO, 1999).

Esta representação, sempre que possível, deve ser simples, implícitas de condições impostas e livres de soluções proibidas para o problema. A representação normalmente é feita através de uma cadeia de bits, mas também podem assumir cadeias de inteiros, reais e outros tipos de valores (LINDEN, 2008).

A execução de um algoritmo genético inicia-se com a criação de um conjunto de vários cromossomos, o qual é chamado de população. Todos os cromossomos desta população inicial são avaliados através da função de aptidão. Depois de avaliados estes cromossomos passam a ser conhecidos como indivíduos.

Os indivíduos ficam sujeitos ao processo de evolução, ou seja, caso não se encontre nenhuma solução esperada, ou caso não haja nenhum outro critério pré-estipulado para que o algoritmo seja interrompido, a população inicial passa por um processo de evolução.

Este processo tem o intuito de fazer com que os indivíduos gerem descendentes mais aptos, ou seja, gerem novos cromossomos que possuam soluções melhores para o problema.

A primeira etapa do processo de evolução é a seleção dos indivíduos para que gerem seus descendentes. A seleção busca privilegiar os melhores indivíduos, porém sem descartar os piores, uma vez que estes podem possuir características importantes – não encontráveis em nenhum outro indivíduo – para a geração de um indivíduo ótimo.

Dentre as formas de seleção conhecidas estão: a seleção por torneio, a roda da roleta e o ranqueamento.

Na seleção por torneio, um número de indivíduos da população inicial é selecionado. Estes indivíduos entram em uma competição pelo direito de ser um dos pais que irão gerar seus descendentes. A decisão pelo torneio se dá pelo indivíduo que possuir o melhor valor da função de aptidão.

Na seleção por roda da roleta uma roleta virtual é criada, na qual cada indivíduo recebe uma fatia proporcional a sua função de aptidão. Indivíduos com melhor aptidão recebem uma fatia maior, indivíduos piores, recebem uma fatia menor. A roleta então é rodada. O valor correspondente àquele em que a roleta parar designará o indivíduo a ser selecionado.

A seleção por ranqueamento, também conhecida por *ranking*, busca evitar a dominância de um superindivíduo, ou seja, um indivíduo com aptidão muito boa e que sempre acaba sendo selecionado. A seleção por *ranking* parte do princípio em que todos os indivíduos são ordenados de acordo com sua função de avaliação.

Uma das formas de se estabelecer um *ranking* para os indivíduos ordenados consiste em inserir um valor arbitrário mínimo para o pior indivíduo e um valor arbitrário máximo para o melhor indivíduo. Os valores para os indivíduos restantes são obtidos através da interpolação dos dois extremos por uma reta.

A Tabela 1 exemplifica a aplicação da técnica de *ranking* para os indivíduos de uma população onde a coluna ordem indica a ordenação dos indivíduos com base na coluna aptidão, a coluna cromossomo expressa as características de cada indivíduo, e a coluna valor *ranking* indica os novos valores atribuídos aos indivíduos para que possam ser selecionados.

Tabela 1: Exemplo da aplicação de *ranking*

Ordem	Cromossomo	Aptidão	Valor <i>Ranking</i>
1	11010100	424	2,0
2	11010010	420	1,5
3	10010100	296	1,0
4	10001110	284	0,5
5	10001010	276	0,0

Fonte: Adaptado de Yu e Gen (2010)

Uma vez selecionados, os indivíduos passam a fazer parte de uma nova população intermediária, a qual sofrerá os operadores de cruzamento e mutação. Estes operadores são os principais mecanismos que permitem a exploração de regiões desconhecidas no espaço de busca.

O primeiro dos operadores a ser aplicado é o cruzamento. No cruzamento, um par de indivíduos é selecionado a fim de que suas características sejam trocadas. Para que isto seja possível, um ponto de corte é definido em uma região do cromossomo de forma que possam gerar segmentos a serem trocados entre os indivíduos selecionados.

Os tipos de cruzamento mais conhecidos são o cruzamento uniforme e o cruzamento de n-pontos. No cruzamento uniforme, um ponto de corte é selecionado de forma aleatória no par de cromossomos de forma a produzir uma cabeça e uma cauda. Em seguida as caudas são trocadas de forma que dois novos cromossomos possam ser gerados. A Figura 15 exemplifica a aplicação deste tipo de cruzamento.

	Cabeça	Cauda
Pai 1	1 1 0 1 0	0 1 0
Pai 2	1 0 0 1 0	1 0 0
Filho 1	1 1 0 1 0	1 0 0
Filho 2	1 0 0 1 0	0 1 0

Figura 15: Exemplo de cruzamento uniforme.

Fonte: Autoria própria.

Por outro lado no cruzamento de n-pontos, mais de um ponto de corte é escolhido aleatoriamente no par de cromossomos. Em seguida os segmentos gerados a partir destes pontos de corte são trocados gerando os novos cromossomos, como podemos observar na Figura 16.

Além destas duas, outras técnicas de cruzamento também são abordadas pela literatura. O cruzamento baseado em ordem, *order-based crossover* (OBX), baseia-se na ordenação dos elementos selecionados de um pai. Por outro lado o cruzamento baseado em posição, *position-based crossover* (PBX), ao invés de impor a ordem, impõe a posição. Estes e outros operadores são explicados de forma detalhada por Lacerda e Carvalho (1999).

	Ponto de Corte							
Pai 1	1	1	0	1	0	0	1	0
Pai 2	1	0	0	1	0	1	0	0
Filho 1	1	0	0	1	0	0	0	0
Filho 2	1	1	0	1	0	1	1	0

Figura 16: Exemplo de cruzamento n-pontos.

Fonte: Autoria própria.

O cruzamento é responsável por gerar novos filhos que substituirão os pais na população intermediária. O cruzamento nem sempre é feito com toda a população, ele geralmente é aplicado com uma determinada taxa de cruzamento. Isto permite que alguns tipos de soluções sejam preservados em indivíduos nos quais não são aplicados o cruzamento.

Em seguida a população intermediária esta apta a receber o operador de mutação. A mutação consiste em trocar o valor de um gene do cromossomo por outro valor admissível. A Figura 17 exprime um exemplo de mutação onde o valor do gene é substituído por outro admissível, ou seja, se o bit for 0 o mesmo é mudado para o bit 1, se o bit for 1 o mesmo é mudado para bit 0.

	Indivíduo Original							
Filho 1	1	0	0	1	0	0	0	0
Filho 2	1	1	0	1	0	1	1	0
	Indivíduo Mutado							
Filho 1	1	1	0	1	0	1	0	0
Filho 2	1	1	1	1	0	0	1	1

Figura 17: Operador de mutação.

Fonte: Autoria própria.

O operador de mutação tende a melhorar a diversidade dos cromossomos da população, mas pode ser capaz de destruir informações importantes. Neste caso uma taxa de mutação é

utilizada para controlar estas alterações. Realizado o operador de mutação nos indivíduos da população, eles passam novamente pelo processo de avaliação, em que os valores de aptidão são atualizados. Neste ponto verifica-se novamente se há critérios de parada que o impeçam de evoluir e, caso contrário, reinicia-se o processo de evolução de uma nova geração.

Após sucessivas rodadas de seleções e avaliações, o algoritmo deve chegar ao fim quando satisfizer certo critério de parada. Caso a solução seja de conhecimento prévio, este irá parar ao encontrar a solução.

Por outro lado, caso não tenhamos o conhecimento da solução do problema, o critério de parada pode ser definido dependendo do problema. Este poderá, por exemplo, finalizar após um determinado número de gerações ou após encontrarmos perda de diversidade dos indivíduos ou quando nas últimas n gerações não houve melhoria na aptidão ou qualquer outro motivo que venha a ser definido pelo usuário do algoritmo (LINDEN, 2008).

A cada iteração do algoritmo genético, a função de aptidão separa os cromossomos mais aptos e encontra novas soluções dentro deste espaço de busca (LINDEN, 2008). É importante observar que no processo de evolução, o melhor indivíduo pode ser perdido entre uma geração e outra devido aos operadores aplicados a ele. Uma das estratégias utilizadas para evitar a perda do melhor indivíduo é o elitismo.

O uso de técnicas de elitismo auxilia de forma que o desempenho do algoritmo sempre cresça com o decorrer das gerações. No elitismo, o melhor ou um conjunto de melhores indivíduos de cada geração não são descartados. Estes indivíduos passam para as próximas gerações de modo que preservem suas características. Isto é motivado pelo fato de se ter mantido dentro da população características responsáveis por boas avaliações das melhores soluções (LINDEN, 2008).

Um dos grandes problemas na utilização de algoritmos genéticos ocorre quando a convergência do algoritmo se dá de forma prematura. Esta ocorre quando surgem diversos bons cromossomos, mas que não possuem soluções ótimas (LINDEN, 2008).

Estes indivíduos, chamados de superindivíduos, acabam por gerar um número muito grande de descendentes com suas características que dominam a população enquanto os genes dos demais indivíduos desaparecem e impedem o surgimento do indivíduo ótimo. Desta forma o algoritmo converge para um ótimo local (LACERDA; CARVALHO, 1999).

Buscar manter a diversidade da população através de técnicas como o *ranking*, o controle da taxa de mutação e evitar a inserção de filhos duplicados na população são uma das formas encontradas de se combater esta convergência prematura.

O fluxograma apresentado na Figura 18 resume as etapas relacionadas ao processo de execução de um algoritmo genético.

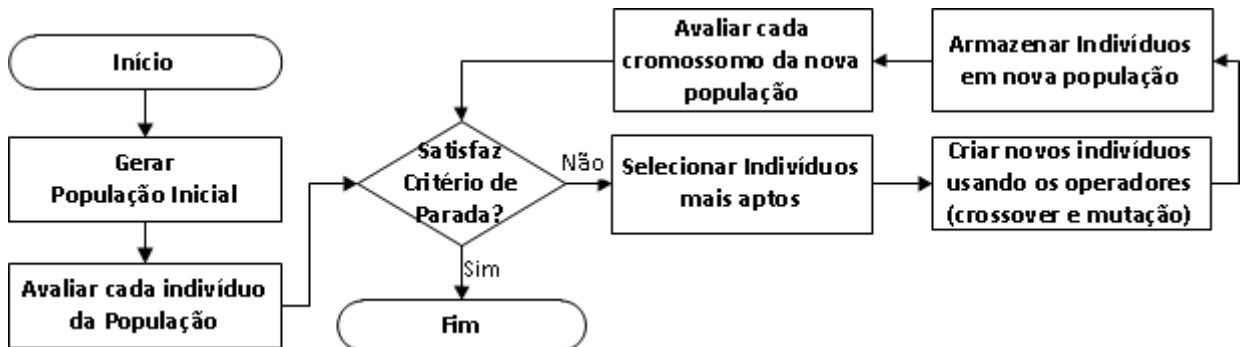


Figura 18: Visão geral do processo evolucionário de um algoritmo genético.

Fonte: Autoria própria.

É importante lembrar que o uso de algoritmos genéticos é mais adequado e recomendado quando o número de soluções possíveis, ou seja, o tamanho do espaço de busca é tão grande que o uso de qualquer outra técnica de busca exaustiva para que consiga encontrar boas soluções se torna computacionalmente impraticável. Estes tipos de problemas são classificados como NP-C em que atualmente todos os algoritmos conhecidos para problemas NP-C exigem um tempo superpolinomial (CORMEN et al., 2009).

2.2.1 ALGORITMOS GENÉTICOS APLICADOS AO PROCESSO DE TRANSCRIÇÃO

Tuohy e Potter (2005) inicia os estudos de algoritmos genéticos (AG) com foco na geração automática de tablaturas para violão. Propõe encontrar um método que resultasse em boas tablaturas uma vez que na época tanto os softwares comerciais quanto as pesquisas acadêmicas buscavam atingir este mesmo objetivo porém, com limitações.

As limitações identificadas por Tuohy e Potter (2005) descrevem que os softwares realizam a conversão de duas formas, ou nota a nota ou no máximo através de um conjunto de notas do compasso. Desta forma, as posições geradas pelos softwares comerciais testados, em muitos casos, tornam a tablatura improvável de ser tocada. As pesquisas acadêmicas da época também não contemplavam todos os tipos de músicas, e ficavam limitadas apenas a músicas que não possuíam acordes.

Diferente dos softwares comerciais, Tuohy e Potter (2005) busca analisar a música como um todo. A metodologia utilizada por eles aplica o uso dos algoritmos genéticos diversas vezes durante 100 gerações, até que fosse possível criar uma população inicial com os melhores

indivíduos evoluídos. Após a geração desta população inicial, o algoritmo genético é aplicado novamente em busca de uma melhor solução.

A figura 19 demonstra a aplicação dos operadores de mutação e permutação, também conhecido como *crossover*, utilizados no algoritmo genético proposto por Tuohy e Potter (2005).

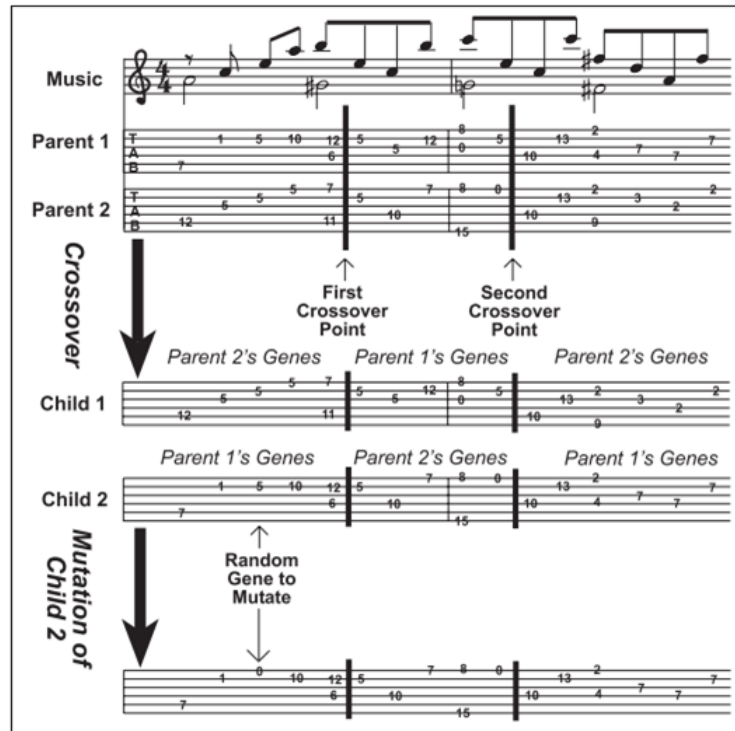


Figura 19: Aplicação dos operadores genéticos.

Fonte: Tuohy e Potter (2005)

Os parâmetros utilizados podem ser observados na Tabela 2. De acordo com Tuohy e Potter (2005), a metodologia de escolha destes parâmetros se dá através da convergência empírica dos mais aptos.

Tabela 2: Parâmetros propostos por Tuohy e Potter (2005)

Parâmetro	Valor
População:	300 indivíduos
Tipo de Seleção:	Torneio Binário
Número de Pontos Crossover:	2 pontos
Taxa de Crossover:	60%
Taxa de Mutação:	7%

Fonte: Autoria própria.

Para Tuohy e Potter (2005), evoluir uma peça inteira é geralmente indesejável, uma vez que criaria provavelmente um espaço de busca muito grande, com um consumo razoável de

tempo. Desta forma a quebra das peças em frases a fim de manter o espaço de busca gerenciável é realizada por Tuohy e Potter (2006).

Apesar da existência de uma segmentação automatizada proposta por Radicioni et al. (2004), Tuohy e Potter (2006) executa a segmentação por conta própria. Então o algoritmo genético é executado individualmente em cada frase a fim de propor um bom arranjo para a música. Uma vez que a primeira frase tenha sido arranjada, cada frase seguinte inclui as últimas três posições da frase anterior.

Quando a partitura original for composta para um instrumento diferente daquele que se pretende tocar, é necessário adaptar a partitura para que possa ser tocada por este outro instrumento. Este processo é denominado arranjo. O arranjo pode exigir certas alterações na partitura original de forma que possa ser utilizada para novos instrumentos.

Nestes casos Tuohy e Potter (2006) destaca outros aspectos que também devem ser observados para que se possa obter um bom arranjo. Inicialmente é necessário adaptar a música existente para que seja tocada em um violão. Deve-se também realizar alterações de tal forma que a intenção original do compositor seja preservada sempre que possível. Salienta também que, em casos onde se fizer necessário realizar reduções, não se deve realizar a introdução de novas harmonias e melodias, mas simplesmente o máximo possível do que o compositor forneceu.

Tuohy e Potter (2006), continua seu trabalho baseado nas pesquisas de Whitley e Starkweather (1990) na qual o método escolhido se justifica pela confiabilidade e pela velocidade em que o algoritmo chega a conversão.

Um dos paradigmas do algoritmo é ser estável, a população recebe os indivíduos de forma individual. Outro é o fato de dividir a população em populações menores a fim de serem evoluídas de maneira independente, todavia com a troca dos melhores indivíduos periodicamente entre elas (TUOHY; POTTER, 2006).

Já os fatores de avaliação utilizados são diferentes dos trabalhados por Heijink e Meulenbroek (2002). Tuohy e Potter (2006) considera importante outros fatores a serem inseridos na avaliação e busca utilizar sua experiência para criar novas funções-objetivo. Quanto aos fatores biomecânicos, aponta 12 tipos que envolvem diferentes dificuldades de movimentação das mãos.

Os fatores cognitivos são as notas altas que devem ser tocadas em casas altas. Outro fator cognitivo apontado é a preferência em utilizar as casas mais baixas, ou seja, mais próximas da cabeça do violão.

Em outros estudos, redes neurais artificiais são abordadas onde uma vez criada a tablatura, é aplicada uma rede neural para atribuir os dedos para cada nota. Passos que os autores consideram maximizar a consistência do algoritmo (TUOHY et al., 2006) (TUOHY; POTTER, 2006).

2.2.2 ALGORITMOS GENÉTICOS BASEADOS EM SUBPOPULAÇÕES

Há situações em problemas de otimização em que encontrar uma solução ótima dentre várias disponíveis envolve o uso de técnicas diferentes das dos algoritmos genéticos tradicionais. Diante disto, o uso de algoritmos evolutivos baseados em subpopulações tem sido utilizado para resolver problemas de natureza combinatorial.

Osaba et al. (2015) apresenta um novo modelo de comunicação entre as subpopulações, também chamado de migração, para problemas de roteamento, Osaba et al. (2015) obtém alta capacidade de exploração, altos fatores de convergência e bons resultados finais.

Li et al. (2013) propõe o uso de subpopulações para solução do PCV no qual o processo de migração é realizado com base em um valor limiar de diversidade de cada subpopulação. Li et al. (2013) obtém melhores resultados com confiança de 95% quando comparado a estratégias diferentes do esquema proposto.

Zhao et al. (2012) propõe um novo modelo de algoritmo genético híbrido baseado nos conceitos de imunidade e subpopulações, a fim de superar os problemas de convergência lenta e também da convergência prematura. Zhao et al. (2012) obtém uma redução de mais de 20% do tempo para chegar ao mesmo resultado quando comparado aos algoritmos genéticos paralelos.

Santos et al. (2010) aplica uma modelagem de representação nó-profundidade a problemas de reestabelecimento de energia em sistemas de distribuição. Santos et al. (2010) utiliza algoritmos evolutivos multiobjetivos baseados em tabelas de subpopulações a fim de permitir uma melhor exploração do espaço de busca e obtém soluções factíveis com baixo índice de computação. Por outro lado, Sanches et al. (2013) combina a técnica desenvolvida por Santos et al. (2010) com a abordagem multiobjetiva do algoritmo evolucionário de força de pareto 2 (SPEA-2), porposta por Zitzler et al. (2001), a fim de explorar não só o espaço de busca quanto o espaço de objetivos.

A técnica de subpopulações é bastante similar ao algoritmo genético tradicional, porém inclui funcionalidades que permitem a criação e manipulação de indivíduos em mais de uma população. O método consiste em dividir a população inicial em diversas subpopulações e cada subpopulação trabalha em uma determinada tarefa a fim de otimizar o objetivo como um todo.

2.3 OTIMIZAÇÃO POR COLÔNIA DE FORMIGAS

Um outro conjunto de algoritmos inspirados por fenômenos naturais é a otimização por colônia de formigas, também conhecido por *Ant Colony Optimization*, ou simplesmente ACO (DORIGO et al., 2006).

Foi estudado o comportamento de formigas enquanto procuravam alimentos para trazer de volta a sua colônia recém criada. A partir deste estudo, foi elaborada uma hipótese sobre o motivo da eficiência destas formigas em encontrar, com certa precisão, um dos caminhos mais curtos existentes para suas fontes de comida (DENEUBOURG et al., 1990).

A hipótese proposta baseia-se em uma substância chamada feromônio que é depositada por cada formiga pelo caminho percorrido e que evapora com o passar do tempo. Observou-se que as formigas escolhem um caminho a percorrer com uma probabilidade proporcional a esta quantidade de feromônio existente em cada caminho.

Ao sair em busca de comida, inicialmente todas as formigas escolhem direções aleatórias com a mesma probabilidade para todos caminhos. Isto se dá pelo fato de ainda não existir feromônio em nenhum destes caminhos. Desta forma, as formigas podem em algumas situações, seguir tanto por trechos diferentes como podem percorrer trechos iguais. Isto pode ser melhor visualizado na Figura 20.

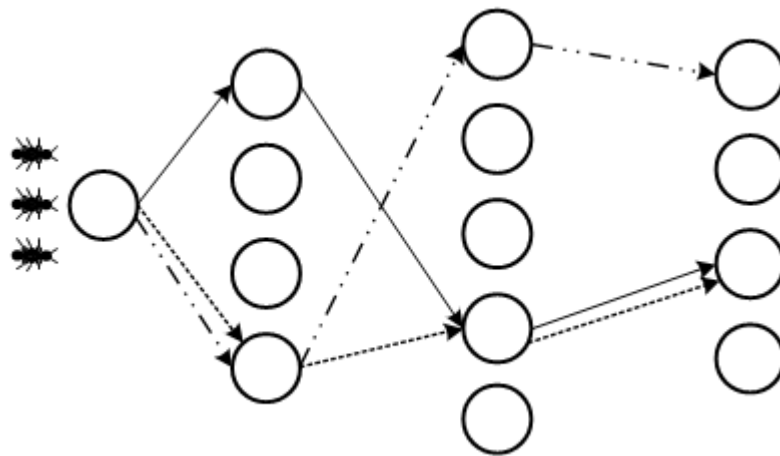


Figura 20: Caminhos percorridos pelas formigas no ACO.

Fonte: Autoria própria.

As formigas caminham depositando a cada instante de tempo t uma quantidade x de feromônio que evapora-se parcialmente com o passar do tempo. Uma vez que chegam ao seu destino, para fins de simplicidade, assume-se que elas retornam para a colônia utilizando o

mesmo caminho já percorrido.

A partir destes fatores, chegou-se às seguintes conclusões: as formigas que encontram o menor caminho para uma fonte de comida vão voltar mais rapidamente que as outras; este caminho possuirá maior quantidade de feromônio, visto que o tempo que se decorreu para que a sua evaporação ocorresse antes que a formiga voltasse depositando mais feromônio, é menor (BLUM, 2005).

Nas próximas repetições deste ciclo as formigas terão uma probabilidade maior de escolher o caminho com maior quantidade de feromônio. O resultado deste ciclo é a concentração das formigas em um caminho que não necessariamente é o ideal mas um dos menores caminhos possíveis.

O algoritmo ACO é definido pela repetição do ciclo apresentado a seguir.

Uma ou mais formigas são dispostas na posição inicial e escolhem os caminhos a percorrer com a probabilidade definida pela equação 3

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{I \in N_i^k} (\tau_{ij})^\alpha (\eta_{ij})^\beta} \quad (3)$$

Onde N_i^k são os vértices para onde a formiga pode caminhar a partir de sua posição atual, η_{ij} é o valor heurístico de cada caminho sendo este inversamente proporcional a sua distância, τ_{ij} é o *trail* que se encontra em uma devida aresta, α a importância relativa da pegada e β a importância relativa da visibilidade.

As formigas percorrem o grafo até que não seja mais possível ocorrer movimentação.

Após todas as formigas chegarem a uma solução, é calculado o valor de *trail* a ser depositado em cada aresta percorrida por uma formiga de acordo com a equação 4

$$\Delta \tau_{ij}^{(k)} = \frac{Q}{L_k} \quad (4)$$

onde Q é a constante de qualidade da pegada e L_k é o tamanho da aresta. Para as arestas que não foram percorridas, não há adição de *trail*.

A constante ρ controla a evaporação deste *trail*. Desta forma, a atualização do *trail* existente em cada aresta é calculada a partir da equação 5

$$\tau_{ij} = (1 - \rho) \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^{(k)} \quad (5)$$

Após a atualização dos valores de *trail* no grafo, uma nova iteração se inicia e o processo se repete até que o número de iterações desejado seja alcançado. Ao final do processo as formigas convergem para um único caminho, que será a melhor solução encontrada.

Este tipo de algoritmo obteve um crescente interesse na área de otimização, sendo aplicado com sucesso na resolução de diversos problemas como do caixeiro viajante (DORIGO; GAMBARDILLA, 1997), otimização de rotas para veículos de transporte (BELL; MCMULLEN, 2004) e de otimização de *design* no planejamento de distribuição de água (MAIER et al., 2003), tendo, neste último caso, resultados favoráveis quando analisados em comparação com resultados obtidos através outros de algoritmos clássicos.

De maior interesse para o desenvolvimento deste trabalho é a escassez de trabalhos que fazem a utilização de algoritmos baseados em colônias de formigas aplicados a problemas de domínio musical, mais especificamente para a transcrição de partituras em tablaturas.

Beneficiando-se do aspecto social e do aproveitamento do *feedback* das formigas, foram realizados trabalhos por Geis e Middendorf (2007) e por Guéret et al. (2004) com o objetivo de encontrar bons caminhos para realizar a composição de músicas. Por outro lado, com o intuito de realizar a transcrição de partituras para tablaturas foram propostos os estudos de RAMOS et al. (2015a) e também RAMOS et al. (2015b).

2.4 EVOLUÇÃO DIFERENCIAL

O algoritmo de evolução diferencial (ED), proposto por Storn e Price (1995) baseia-se em um conjunto de indivíduos criados aleatoriamente a fim de gerar uma população inicial. Por conseguinte, a população inicial é evoluída através de operadores de modo que gerem soluções ótimas (PRADO, 2013).

Um dos operadores utilizados para que a população seja evoluída é o operador de mutação, que no algoritmo ED é denominado de mutação diferencial. Ela é embasada em gerar, para cada indivíduo da população $X_{i,g}$, um indivíduo mutante através da diferença vetorial entre dois indivíduos distintos da população, ponderada por uma constante escalar F . O resultado da diferença é somado com um vetor base, ou seja, outro indivíduo distinto da população (PRICE et al., 2006). O cálculo desta mutação pode ser observado através da equação 6

$$V_{i,g} = X_{r0,g} + F(X_{r1,g} - X_{r2,g}) \quad (6)$$

desde que satisfaça a condição em 7

$$r_0, r_1, r_2 \in [0, NP - 1]; \quad r_0 \neq r_1 \neq r_2 \neq i; \quad F > 0 \quad (7)$$

onde $V_{i,g}$ é o indivíduo mutado, também chamado de vetor modificado, $X_{r_0,g}$ o vetor base, $X_{r_1,g}$ e $X_{r_2,g}$ os indivíduos envolvidos na diferença vetorial e F a constante escalar. Os subscritos g e i são respectivamente a geração do indivíduo e o indivíduo dentro da população. Por fim $NP-1$ indica o último indivíduo do vetor (STORN; PRICE, 1995) (PRADO, 2013).

Após a mutação, os indivíduos são cruzados com os indivíduos da geração atual a fim de gerar indivíduos denominados indivíduos experimentais. Estes indivíduos experimentais irão ser comparados com os indivíduos da população atual, então os mais aptos irão formar a próxima geração da população (PRICE et al., 2006) (PRADO, 2013).

O cruzamento ocorre de maneira uniforme e tem como objetivo aumentar a diversidade (PRICE et al., 2006). A partir do cruzamento de cada vetor alvo $X_{i,g}$ da população e seu respectivo vetor mutante $V_{i,g}$, são gerados vetores experimentais de acordo com 8

$$U_{i,g} = U_{j,i,g} \begin{cases} V_{j,i,g} \text{ caso } (rand_j(0,1) \leq Cr \text{ ou } j = j_{rand}) \\ X_{j,i,g} \text{ caso contrário} \end{cases} \quad (8)$$

onde j é o parâmetro do vetor correspondente, Cr a probabilidade de cruzamento e j_{rand} o parâmetro do vetor mutante escolhido aleatoriamente.

Desta forma, caso um número gerado aleatoriamente seja menor ou igual a probabilidade de cruzamento Cr , o vetor experimental adquire o parâmetro do vetor mutado, caso contrário adquire do vetor alvo. A condição $j = j_{rand}$ é utilizada de forma garantir que o vetor experimental não seja idêntico ao vetor alvo (STORN; PRICE, 1997). A Figura 21 ilustra o processo de cruzamento.

Por fim, após a geração dos vetores experimentais, um método de seleção determina se o vetor experimental irá compor a geração da nova população. A decisão é feita com base na aptidão, ou seja, caso a aptidão do vetor experimental seja melhor, este toma lugar na nova população, caso contrário o vetor alvo é retido (STORN; PRICE, 1995). A Figura 22 expressa o fluxo do algoritmo ED, também conhecido por ED clássico.

Há outros tipos de variações do ED e por este motivo a notação ED/ $x/y/z$ é utilizada para representar estas diferentes variações. Neste caso, x especifica o vetor a ser mutado, y especifica o número de vetores de diferença utilizados e z o tipo de cruzamento utilizado (PRICE et al., 2006).

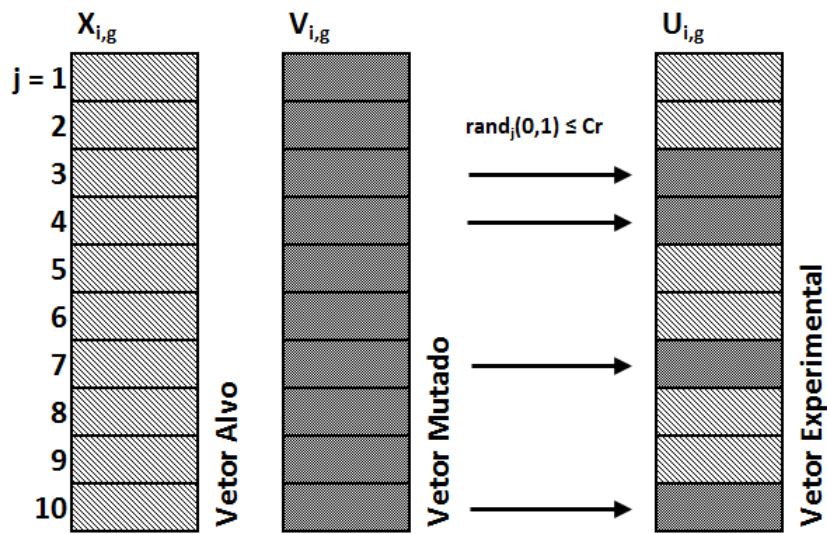


Figura 21: Cruzamento na ED.

Fonte: Adaptado de Storn e Price (1997)

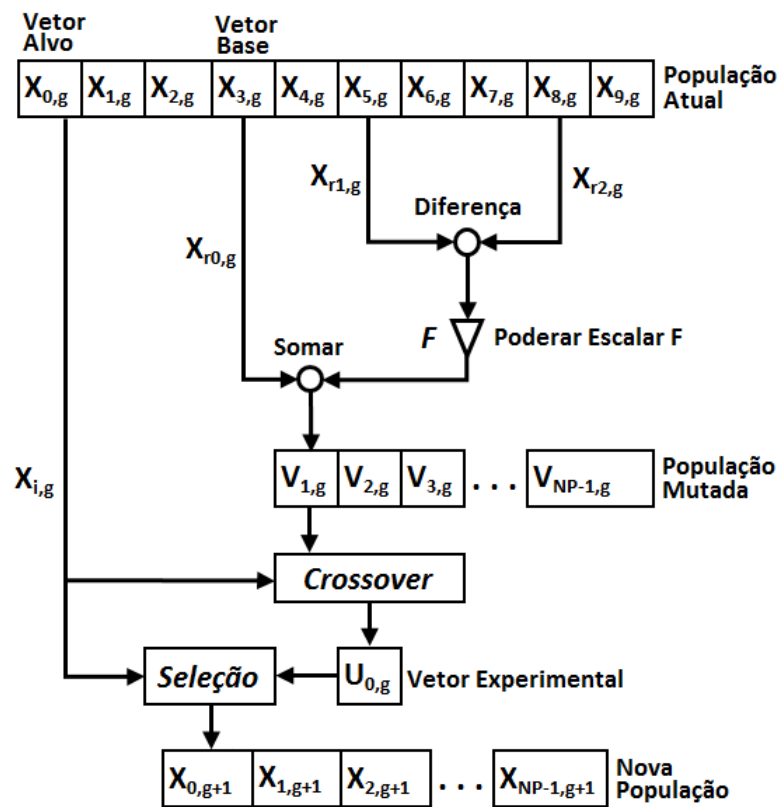


Figura 22: Fluxo do algoritmo ED clássico.

Fonte: Adaptado de Price et al. (2006)

Para o exemplo apresentado, o nome técnico utilizado é ED/rand/1/bin. Uma vez que o vetor a ser mutado foi determinado de forma aleatória (*rand*), o número de vetores de diferença é l e o cruzamento é do tipo binomial (*bin*).

É importante destacar algumas variações do ED. Uma delas, é ED/best/1/bin, em que o melhor indivíduo é sempre escolhido para ser o vetor base (STORN, 1996). Nesta variação, o algoritmo normalmente converge de forma mais rápida do que o ED/rand/1/bin.

Por outro lado, a variação ED/target-to-best/1/bin foi proposta a fim de gerar novos vetores sem aumentar tanto a pressão seletiva (PRICE, 1997). Para isto, a aptidão do vetor base deve ser menor ou igual a do vetor alvo. Outras modificações ainda sugerem, diferente da versão clássica que utiliza 1 vetor de diferença, o uso de 2 ou até 3 vetores para a mutação diferencial.

2.4.1 EVOLUÇÃO DIFERENCIAL COM LISTA DE MOVIMENTOS

Prado et al. (2012) e Prado et al. (2010) inicialmente propôs uma abordagem discreta para a ED a fim de resolver problemas de natureza combinatória de forma que preserve as características do seu mecanismo de busca.

Esta nova abordagem inclui a criação de uma *Lista de Movimentos* que é formada pela diferença vetorial entre dois candidatos à equação da mutação diferencial. Desta forma, a operação de subtração gera esta lista de movimentos que pode ser observada através da equação 9

$$M_{r1r2} = X_{r1,g} \ominus X_{r2,g} \quad (9)$$

onde, para dois candidatos, $X_{r1,g}$ e $X_{r2,g}$, a realização da diferença vetorial através do operador de diferença \ominus resulta em M_{r1r2} como lista de movimentos.

Esta lista representa a *distância* entre os dois candidatos $X_{r1,g}$ e $X_{r2,g}$ de tal forma que a aplicação da lista de movimentos à solução $X_{r2,g}$ leva à solução $X_{r1,g}$. É importante observar que a diferença vetorial não calcula matematicamente esta diferença (PRADO et al., 2010).

A criação desta lista também está sujeita a multiplicação do escalar F . Esta multiplicação, apresentada pela equação 10

$$M'_{r1r2} = F \otimes M_{r1r2} \quad (10)$$

onde $M'_{r_1 r_2}$ é o resultado dos primeiros movimentos da lista $M_{r_1 r_2}$.

Por fim, a lista de movimentos é aplicada ao vetor base conforme a equação 11

$$V_{i,g} = X_{r0,g} \oplus M'_{r_1 r_2} \quad (11)$$

onde \oplus é um operador binário especial de adição no domínio das variáveis discretas e $V_{i,g}$ é o vetor mutante resultante da operação (PRADO, 2013) (PRADO et al., 2012). Com isso, pode-se escrever a equação 12, que é a versão discreta da equação 6.

$$\begin{aligned} V_{i,g} &= X_{r0,g} \oplus F \otimes (X_{r1,g} \ominus X_{r2,g}) \\ V_{i,g} &= X_{r0,g} \oplus F \otimes M_{r_1, r_2} \\ V_{i,g} &= X_{r0,g} \oplus M'_{r_1, r_2} \end{aligned} \quad (12)$$

Esta abordagem utilizando lista de movimentos tem sido estudada e utilizada em soluções de problemas na restauração de energia elétrica (PRADO et al., 2014) (SANCHES et al., 2014) (SANCHES, 2013). Por outro lado, até o momento, a aplicação desta abordagem não foi encontrada para solução de problemas na área de computação musical, mais especificamente para transcrições de partituras.

2.5 ANÁLISES ESTATÍSTICAS

Com a finalidade de analisar estatisticamente os resultados obtidos, análises através de ranqueamento (*ranking*) por dominância são propostos por Kuncheva e Rodríguez (2007) e Gois et al. (2013). Para esse ranqueamento, cada método é comparado com todos os outros, par-a-par onde os ganhos e perdas significativas são contabilizados para determinação da dominância.

A dominância de um método é o resultado da diferença entre o número de ganhos e o número de perdas estatisticamente significativas em todas as músicas utilizadas. Assim, quanto maior a dominância de um método, melhor será seu posicionamento no ranqueamento em relação aos demais.

3 ABORDAGEM PROPOSTA

A abordagem proposta é baseada na utilização de algoritmos de busca heurística e algoritmos evolutivos para a conversão de partituras musicais em tablaturas.

Para o uso das técnicas de busca heurística será abordado o algoritmo A* e para o uso das técnicas evolucionárias serão abordados o ACO, o AG, o AG-SP e a ED. Técnicas de busca local 2-opt e 3-opt também foram aplicadas aos algoritmos AG, AG-SP e ED.

3.1 REPRESENTAÇÃO DO PROBLEMA

Para conseguir realizar a conversão da partitura em tablatura, é preciso encontrar uma forma de inserir uma partitura em um ambiente computacional a fim de que esta partitura possa ser tratada.

O processo de leitura e conversão de arquivos de áudio e digitalização de partituras e conversão dos dados das notas usando reconhecimento de imagem foi descartado uma vez que sairia do escopo do problema tratado. Neste sentido, optou-se pela importação de arquivos de partituras digitais.

Atualmente pode-se encontrar diversas notações musicais disponíveis em formato digitalizado. Entre elas existem os formatos utilizados pelos softwares Encore¹, Sibelius², Finale³ e MuseScore⁴.

Há também formatos simplificados como o *ABC Notation*⁵ que é um sistema de notação baseado em texto que foi desenvolvido para músicas *folk* e tradicionais sendo atualmente muito utilizado para estes estilos (WALSHAW, 2014). Já o formato de arquivo de notação de intercâmbio (NIFF⁶) é um formato aberto que foi muito utilizado por programas de digitalização

¹<http://www.passportmusic.com/products/encore>

²<http://www.avid.com/US/products/sibelius>

³<http://www.finalemusic.com/products/finale/>

⁴<https://musescore.org>

⁵<http://abcnotation.com/about>

⁶<http://neume.sourceforge.net/niff/>

de reconhecimento ótico de partituras musicais.

Dentre tantos formatos encontrados destacamos a Linguagem de Marcação Extensível para Música (MusicXML) que foi desenvolvido para o compartilhamento de músicas entre softwares, bem como para ser utilizado como forma de arquivamento para uso posterior (GOOD, 2001).

A decisão pelo uso do formato MusicXML foi motivada por ele ser baseado em uma linguagem de marcação, o que o torna flexível e garante maior interoperabilidade com outros sistemas. Um exemplo de um trecho no formato MusicXML contendo as marcações utilizadas no processo de importação da partitura pode ser melhor visualizado na Figura 23.

Mesmo que não sejam utilizados todos os recursos do MusicXML, é possível empregá-lo limitando o número de notas, descartando as marcações não importantes para o processo de conversão.

```

<score-partwise>
  <part id="p1">
    <measure number="1">
      <attributes>...</attributes>
      <note>
        <pitch>
          <step>...</step>
          <alter>...</alter>
          <octave>...</octave>
        <pitch>
      </note>
      <note>...</note>
    </measure>
    <measure number="2">...
  </measure>
</part>
  <part id="P2">...</part>
</score-partwise>

```

Figura 23: Exemplo do formato MusicXML.

Fonte: Autoria própria.

Entretanto há marcações chaves que são indispensáveis no processo de conversão. Estas podem ser visualizadas na Tabela 3.

Como o braço de um violão clássico dispõe de 6 cordas, e normalmente de 19 a 24 casas isto permite com que seja possível criar uma matriz correspondente a cada nota em cada posição diferente do braço do violão. Entretanto, se estas notas, apesar de serem as mesmas, possuírem frequências diferentes, elas representarão notas musicais em oitavas diferentes umas

das outras.

Tabela 3: Significado das marcações no MusicXML.

Marcação	Descrição
<score-partwise>	Raiz da estrutura. É composta de partituras.
<part>	Representa a partitura musical. É composta de compassos.
<measure>	Representa o compasso musical.
<attributes>	Contém dados musicais representados no começo da partitura ou no início de cada compasso (transposições, claves etc).
<note>	Representa a nota musical.
<pitch>	Informações referentes a altura do nota musical.
<step>	Tom da nota musical.
<octave>	Oitava referente à nota musical
<alter>	Acidente musical (bemol, sustenido etc.)

Fonte: Autoria própria.

Desta forma, é criado um mapeamento baseado no trabalho de Radicioni et al. (2005), porém, com algumas modificações. Este mapeamento numérico permitirá que seja identificada uma nota e a sua posição a fim de que esta representação possa ser utilizada como fonte do algoritmo de conversão. Este mapeamento pode ser observado na Figura 24 onde as cores indicam as oitavas das notas e os números, o valor da frequência harmônica para cada uma das notas. Os valores da frequência estão expressos em Hertz e arredondados para números inteiros.

Casa	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	329	349	370	392	415	440	466	494	523	554	587	622	659	698	740	748	831	880	932	988	1047	1109	1175	1245	1319
2	247	262	277	294	311	329	349	370	392	415	440	466	494	523	554	587	622	659	698	740	748	831	880	932	988
3	196	208	220	233	247	262	277	294	311	329	349	370	392	415	440	466	494	523	554	587	622	659	698	740	748
4	147	156	165	175	185	196	208	220	233	247	262	277	294	311	329	349	370	392	415	440	466	494	523	554	587
5	110	117	123	131	139	147	156	165	175	185	196	208	220	233	247	262	277	294	311	329	349	370	392	415	440
6	82	87	92	98	104	110	117	123	131	139	147	156	165	175	185	196	208	220	233	247	262	277	294	311	329

Figura 24: Representação das notas no braço do violão.

Fonte: Baseado em Radicioni et al. (2005)

A fonte de dados utilizada para a exploração do problema foi retirada de uma organização sem fins lucrativos através do sítio *The Session*⁷. Esta base de dados é formada por músicas irlandesas disponíveis no formato *ABC Notation* e que foram convertidas para o formato MusicXML.

⁷<http://www.thesession.org>

3.2 AVALIAÇÃO DA TRANSCRIÇÃO

Para que se possa verificar se a transcrição realizada é boa, necessita-se de uma forma de avaliá-la.

Heijink e Meulenbroek (2002) assume que há diferentes tipos de dificuldades biomecânicas que são geradas a partir do movimento horizontal das mãos, do movimento vertical das mãos e da distância entre os dedos.

Apesar de a metodologia proposta por Heijink e Meulenbroek (2002) ser uma estimativa mais elegante e precisa, ela é muito difícil de ser implementada, sendo considerado um esquema impraticável (TUOHY; POTTER, 2005). Desta forma, muitos trabalhos quando optam por seguir a proposta de Heijink e Meulenbroek (2002) realizam de forma parcial ou abordam outras formas de avaliação (RADICIONI et al., 2004) (TUOHY; POTTER, 2005) (BURLET; FUJINAGA, 2013) (RUTHERFORD, 2009).

Não há uma forma padronizada de computar estas dificuldades (BURLET, 2013). Por esta razão, a fim de simplificar este cálculo, neste trabalho o mapeamento das notas no braço de um violão é representado através de um espaço bidimensional. Sendo assim, opta-se pelo uso da distância euclidiana para o cálculo entre uma posição mapeada e a próxima posição a ser executada.

Tomando-se duas sequências de notas a serem executadas e inserindo-as no plano cartesiano, é possível observar que suas coordenadas são ortogonais, desta forma o cálculo da distância é dado através da equação 13.

$$d(x,y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (13)$$

A equação realiza o cálculo da distância entre dois pontos. Onde x_1 representa a posição da corda atual e x_2 a próxima corda, assim como y_1 representa a posição da casa atual e y_2 representa a próxima casa. O uso deste cálculo faz com que seja encontrada a distância entre a posição mapeada que representa a nota atual e a próxima nota a ser tocada.

Na Figura 25 podemos observar o deslocamento da nota Ré sustenido (D#) mapeada no ponto (2,4) para a nota Fá (F) mapeada no ponto (5,8), projetadas no plano cartesiano. A distância entre estes pontos é passível de ser calculada através da distância euclidiana.

As casas do braço de um violão não possuem mesma distância entre si. Não obstante isto, a fim de simplificar o cálculo, é considerada uma distância simétrica entre todas elas.

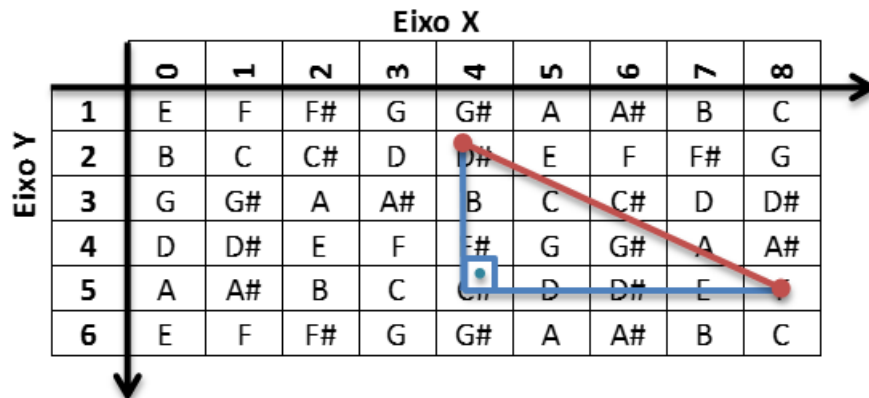


Figura 25: Representação da distância entre duas posições no plano cartesiano.

Fonte: Autoria própria.

Deve-se levar em consideração que a casa 0 representa uma música tocada utilizando apenas a corda do violão. Isto significa que não há deslocamento das mãos para outra casa. Nestes casos, para o cálculo da distância euclidiana considera-se apenas o deslocamento entre as cordas.

Onde houver a possibilidade de decisão entre mais de uma posição para representar a mesma nota a ser tocada, a distância entre a posição atual e todas as demais possíveis posições são calculadas para a próxima nota. Em seguida opta-se pela posição que representa a menor distância. Em caso de empate nestas situações, é priorizada a posição que possuir casa 0. Se não houver, a decisão é feita de forma arbitrária.

A Figura 26 mostra um exemplo para cálculo usando a distância euclidiana partindo do ponto (5,8) até os pontos (3,0), (4,5), (5,10) e (6,15).

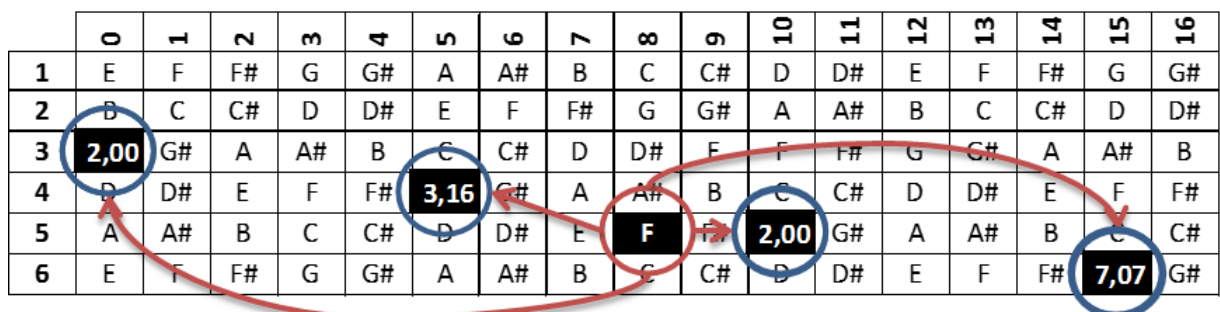


Figura 26: Exemplo do cálculo usando a distância euclidiana.

Fonte: Autoria própria.

3.3 ABORDAGEM UTILIZANDO O ALGORITMO A*

Uma das formas de conversão proposta é o uso de um algoritmo de busca heurística A*. Uma vez preparada a base de dados para a leitura das informações relevantes da partitura digital, é possível disponibilizá-la ao algoritmo de conversão.

Após a importação de todas as informações, o algoritmo inicia o processo de transcrição em busca de posições para cada uma das notas das músicas.

Uma música é composta de várias notas musicais e uma nota é capaz de ser representada em diferentes posições no braço do violão. No algoritmo de busca heurística, a música é mapeada de forma que cada nota musical seja representada em um nível de uma árvore, onde o nó raiz é representado pela primeira nota a ser tocada.

A Figura 27 exemplifica a representação destas notas em níveis, onde cada nota pertence a um nível e possui diferentes posições que a representam.

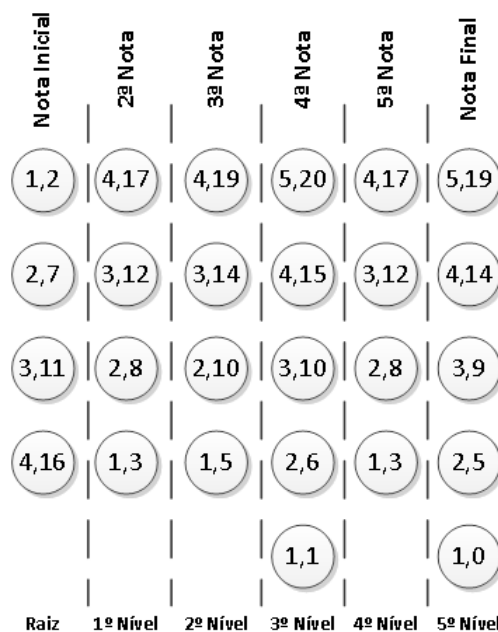


Figura 27: Níveis das notas de uma música.

Fonte: Autoria própria.

Na primeira etapa o algoritmo escolhe uma posição referente à nota inicial e define esta posição como o nó raiz. Neste momento, para facilitar a explicação do funcionamento, esta posição será denominada nó atual, e cada uma das próximas posições referentes à próxima nota a ser tocada, ou seja, a 2ª nota da música passa a ser chamada de próximo nó.

O algoritmo expande então o nó atual para um possível próximo nó e durante este pro-

cesso ele verifica o custo para se transitar entre um vértice atual e o próximo vértice expandido. Esta expansão e cálculo do custo são realizados em todos os possíveis próximos nós.

O cálculo do custo de transição se dá através da equação 2 apresentada na seção 2.1.1 deste trabalho onde $h(x)$ é calculado através da distância euclidiana apresentada na seção 3.2 e $g(x)$ representa o custo da distância já percorrida, ou seja, o acúmulo dos custos entre as arestas dos vértices.

Uma vez expandido todos os nós e calculado todos os custos dos vértices para os possíveis próximos nós, o algoritmo verifica qual o caminho de menor custo, ou seja, qual aresta possui o menor valor calculado pela heurística e passa a chamar esta próxima aresta de nó atual, mantendo guardadas as informações dos demais nós já expandidos.

A Figura 28 mostra a expansão do nó raiz para os próximos nós e o cálculo da heurística entre as arestas. Os valores apresentados como custo são simbólicos apenas para fins de exemplificação. Podemos observar que a escolha do próximo nó, devido ao menor custo, se deu para a posição $\langle 1,3 \rangle$. O nó de posição $\langle 1,3 \rangle$ passa a ser o nó atual e o processo de expansão continua passando a escolher o nó $\langle 1,5 \rangle$.

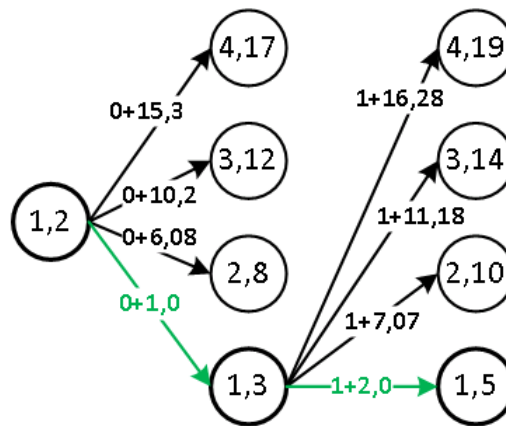


Figura 28: Expansão do nó raiz.

Fonte: Autoria própria.

Escolhido o próximo nó, o algoritmo continua realizando o processo de expansão e cálculo dos custos entre as arestas. Entretanto, poderá ocorrer que ao calcular o custo entre a transição das arestas o algoritmo se depare com uma situação onde um caminho já expandido anteriormente possua um custo menor. Como o algoritmo não descarta os nós já expandidos, este retorna o nó atual ao nó de menor custo e passa a expandi-lo.

Na Figura 29 podemos observar esta situação onde ao expandir para o 4º nível da

árvore, o menor custo de transição tem valor 6,41 ($4,41+2,0$). Este custo é maior do que um dos vértices já expandidos anteriormente. Sendo assim o algoritmo volta para o vértice de posição $\langle 2,8 \rangle$ do primeiro nível e passa a expandi-lo.

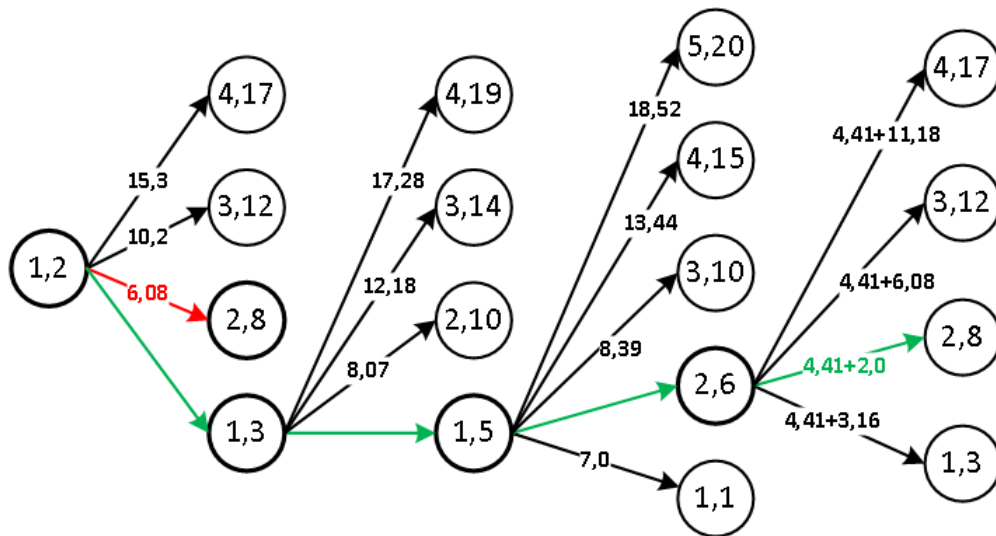


Figura 29: Retorno do algoritmo para um nó já expandido.

Fonte: Autoria própria.

O processo de expansão e cálculo do melhor caminho continua a ser realizado até que o algoritmo se depare com o seu destino. Neste caso, com uma posição referente à última nota a ser tocada.

Uma vez encontrado o destino, o algoritmo traça o caminho informando todos os nós percorridos que resultaram em um caminho de menor custo. Depreende-se este caminho como sendo todas as posições que precisam ser tocadas para que a música seja executada.

Ao término da execução, o algoritmo também considera as demais possíveis posições para a nota inicial. Ele torna cada possível posição inicial uma raiz da árvore de busca e realiza todo o processo de expansão e cálculo com cada uma delas.

3.4 ABORDAGEM UTILIZANDO O ALGORITMOS GENÉTICOS

Outra forma de conversão proposta é o uso de um algoritmo genético tradicional.

A ciência de alguns conceitos é fundamental para a compreensão do funcionamento do algoritmo. O espaço de busca no processo de conversão representa todas as possíveis soluções convertidas. Em outras palavras, o espaço de busca compreende todas as possibilidades de se combinar as sequências das notas a fim de que sejam tocadas para reproduzir a música no

violão.

Nem todas as sequências de notas produzidas são boas, algumas são impossíveis de serem tocadas. Isto é avaliado através de uma função de aptidão, também conhecida como *fitness*.

Um gene pode ser descrito como uma nota representada através de uma posição a ser tocada. Isto nada mais é do que parte da solução para o problema. Já um cromossomo é uma estrutura de dados que codifica uma solução completa para o problema, ou seja, um conjunto de sequências de posições que irão compor a música a ser tocada. Apesar de formar uma música por completo, o cromossomo representa apenas um simples ponto no espaço de busca.

Um indivíduo é formado pela combinação do cromossomo e sua aptidão. E a população é formada por um conjunto de indivíduos que representam as possíveis soluções para o problema.

A execução do algoritmo genético tradicional segue o pseudocódigo apresentado na Figura 30.

- 1 Realizar a importação da música para um cromossomo;
- 2 Inicializar os parâmetros do algoritmo genético;
- 3 Gerar uma população inicial randômica;
- 4 Avaliar os indivíduos da população;
- 5 **enquanto** *critério de parada não for satisfeito* **faça**
- 6 Selecionar indivíduos mais aptos;
- 7 Realizar *crossover*;
- 8 Realizar mutação;
- 9 Avaliar os indivíduos;
- 10 **fim**

Figura 30: Pseudocódigo do algoritmo genético tradicional.

Fonte: A autoria própria.

Uma vez importada a música a ser convertida, dá-se início a execução do algoritmo genético através da inicialização dos parâmetros a serem utilizados no processo de conversão.

Inicializado os parâmetros do algoritmo, a próxima etapa consiste em criar a população inicial com base na música importada e representada através da classe cromossomo. Para cada gene do cromossomo, utilizando os atributos de tom e oitava, é possível selecionar uma possível combinação de casa e corda dentre as possíveis posições suportadas pela nota. Esta decisão pela posição a ser utilizada é feita de forma aleatória a fim de garantir uma boa diversidade de indivíduos.

Gerada a população inicial a mesma é avaliada e verificada se atende aos critérios de parada preestabelecido.

Caso os critérios não forem atendidos a população inicial é evoluída através dos operadores de *crossover* e mutação.

Para que sejam aplicados estes operadores, os indivíduos mais aptos da população inicial são selecionados através da técnica de torneio. O que possuir a melhor aptidão é inserido em uma população intermediária, que por sua vez será evoluída formando as próximas gerações.

A partir da população intermediária inicia-se a aplicação do operador de *crossover*. O *crossover* busca dividir 2 cromossomos na quantidade de pontos de corte definidas previamente pelos parâmetros do algoritmo genético e em seguida trocar as partes entre estes 2 cromossomos.

Duas metodologias foram utilizadas: o *crossover* randômico e o *crossover* uniforme. O *crossover* randômico, seleciona no mínimo um segmento e no máximo metade dos segmentos do cromossomo que será permutado. A definição de qual segmento será selecionada, como próprio nome da técnica já diz, é feito de forma randômica.

Já no *crossover* uniforme, apresentado na Figura 31, os segmentos são selecionados alternadamente, um sim outro não, a fim de que estes possam ser permutados e gerarem dois novos filhos.

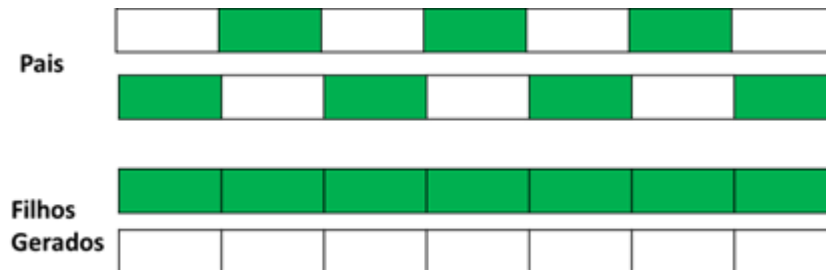


Figura 31: Crossover Uniforme.

Fonte: Autoria própria.

O operador de mutação consiste em trocar a posição (corda e casa) da nota a ser mudada por outra possível combinação capaz de representar a mesma nota musical.

É importante observar que a mutação acontecer com sucesso, a nota musical não deve ser alterada. Se a nota musical fosse alterada, estaríamos alterando a sequência de notas da música e criando uma música diferente da proposta originalmente. Desta forma, a mutação realiza a troca da posição no braço do instrumento para uma outra posição que representa a

mesma nota em questão.

A mutação proposta, baseada nas técnicas abordadas por Radicioni et al. (2004) e Tuohy e Potter (2006) para procedimentos de segmentação, utiliza o princípio de buscar a troca da posição da nota para a melhor combinação possível. Isto é feito através da análise da nota anterior e da nota atual a ser mudada.

A transição entre a nota atual e as possíveis próximas notas é calculada através da função de aptidão e posteriormente é decidido qual é a melhor nota para que a mutação aconteça. A Figura 32 representa o funcionamento desta mutação. Neste exemplo, a posição atual $\langle 5,12 \rangle$ deve ser mudada. Para isto, o peso da transição da nota anterior para todas as possíveis posições é calculado e é decidido mudar para posição $\langle 4,7 \rangle$.

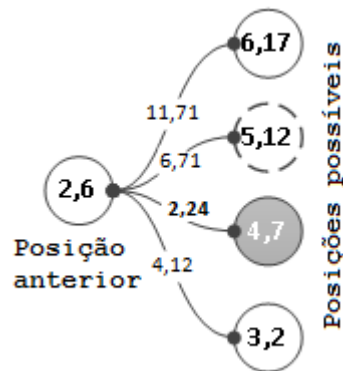


Figura 32: Processo de mutação.

Fonte: Autoria própria.

Terminados os operadores de *crossover* e mutação, uma nova população é gerada. Esta primeira geração de indivíduos deve ser novamente avaliada em busca dos critérios de parada pré-estabelecidos. O laço é repetido até que os critérios de parada sejam atendidos.

3.4.1 ABORDAGEM UTILIZANDO O AG-SP

A técnica de subpopulações é bastante similar ao algoritmo genético tradicional, ela consiste em dividir a população inicial em diversas subpopulações e cada subpopulação trabalha em uma determinada tarefa a fim de otimizar o objetivo como um todo. No caso do problema de conversão, cada subpopulação é responsável por evoluir uma quantidade de notas da música.

A Figura 33 apresenta um exemplo da divisão da música em 5 subpopulações. Cada subpopulação fica responsável por uma porção de 3 notas.

Para que isto seja possível, os indivíduos das subpopulações deverão guardar não só o *fitness* relacionado à música como um todo, mas também outro adicional referente à porção de notas das quais é responsável por evoluir.

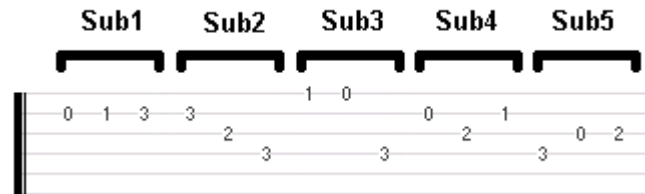


Figura 33: Divisão em subpopulações.

Fonte: Autoria própria.

A execução do AG-SP pode ser melhor compreendido através do pseudocódigo apresentado na Figura 34.

- 1 Realizar a importação da música para um cromossomo;
- 2 Inicializar os parâmetros do algoritmo genético;
- 3 Gerar subpopulações iniciais de forma aleatória;
- 4 Gerar uma população ponderada;
- 5 Avaliar os indivíduos das populações iniciais;
- 6 **enquanto** *critério de parada não for satisfeito* **faça**
- 7 Realizar *crossover*;
- 8 Realizar mutação;
- 9 Avaliar os indivíduos;
- 10 Migrar os indivíduos para as subpopulações;
- 11 Recriar a população ponderada;
- 12 **fim**

Figura 34: Pseudocódigo do AG-SP.

Fonte: Autoria própria.

A primeira etapa do pseudocódigo, assim como no algoritmo genético tradicional, exige a importação dos dados para a classe Cromossomo. Em seguida os parâmetros do algoritmo genético são inicializados para dar início à criação das subpopulações.

Diferente do algoritmo genético tradicional, ao invés de se criar uma população com vários indivíduos, o algoritmo divide esta única população em subpopulações com uma quantidade menor de indivíduos que, se somados, dariam o mesmo tamanho da população inicial do genético tradicional.

Ao término da geração das subpopulações iniciais, é necessário calcular o *fitness* de

cada um dos indivíduos, bem como calcular também o *fitness* da porção pela qual cada uma das subpopulações é responsável. A definição da quantidade de notas pela qual cada subpopulação é responsável é encontrada através da divisão da quantidade de genes do cromossomo pela quantidade de subpopulações do algoritmo. Caso a divisão for inexata, as notas restantes ficam sob responsabilidade da última subpopulação gerada.

Visto que o principal objetivo do *fitness* é avaliar quão bom é o indivíduo e visto que a otimização visa minimizar os esforços envolvidos na geração de tablaturas que sejam fáceis de serem tocadas, os indivíduos das subpopulações geradas são ordenados de forma crescente por *fitness* da porção a fim de que possam ser utilizada para a geração de uma população denominada população ponderada.

A Figura 35 apresenta a formação das subpopulações geradas e ordenadas por *fitness* da porção (P.Fitness). Note que o *fitness* geral (G.Fitness) nem sempre ficará ordenado de forma crescente, uma vez que nem sempre a melhor porção indica ser o melhor indivíduo.

	Subpop1		Subpop2		Subpop3		Subpop4		Subpop5	
	P.Fitness	G.Fitness	P.Fitness	G.Fitness	P.Fitness	G.Fitness	P.Fitness	G.Fitness	P.Fitness	G.Fitness
Indiv. 1	450	2750	270	2880	420	1480	360	1550	240	2070
Indiv. 2	480	1620	300	2900	450	2750	420	1590	300	2300
Indiv. 3	540	2750	330	2940	480	1490	450	1700	330	2640
Indiv. 4	600	1620	390	2990	480	1620	450	2750	390	2720
Indiv. 5	600	3000	420	2080	480	1930	480	1620	420	3200
Indiv. 6	660	3300	450	2750	510	2640	480	2000	450	3300

Figura 35: Subpopulações ordenadas por *fitness* da porção.

Fonte: Autoria própria.

Dando continuidade no algoritmo, a criação de uma população ponderada será necessária antes que sejam aplicados os operadores de *crossover* e mutação.

Para geração de um indivíduo da população ponderada é utilizado um indivíduo de cada subpopulação inicial. O indivíduo ponderado gerado é resultante da concatenação das porções de cada subpopulação.

Por exemplo, para gerar o primeiro indivíduo ponderado é selecionado o melhor indivíduo da primeira subpopulação, o melhor indivíduo da segunda subpopulação e assim por diante até que sejam selecionados todos os indivíduos de todas as subpopulações iniciais.

Uma vez selecionados, concatena-se as porções por quais estes indivíduos são res-

ponsáveis a fim de que seja gerado um novo indivíduo ponderado. Estes passos são repetidos com os segundos melhores indivíduos, posteriormente com os terceiros melhores indivíduos e assim sucessivamente até que toda a população ponderada seja criada.

A Figura 36 exemplifica a criação da população ponderada. Note que apenas nas subpopulações os indivíduos são avaliados com base nas porções.

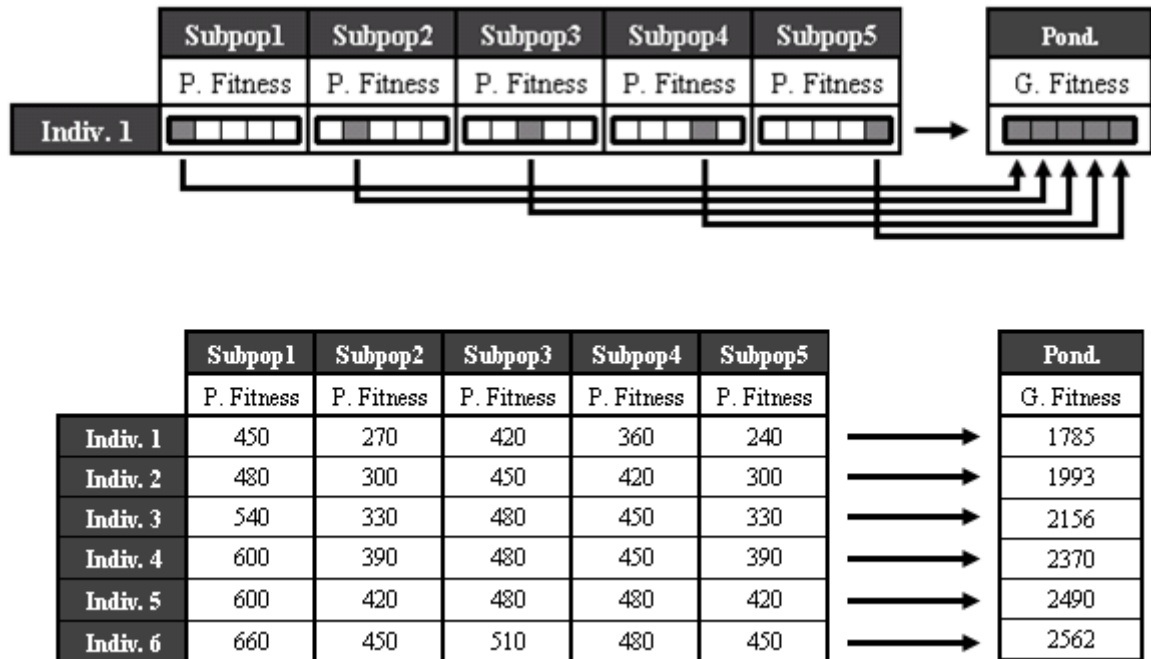


Figura 36: Geração da população ponderada.

Fonte: Autoria própria.

Na ponderada os indivíduos são avaliados com base no *fitness* geral, ou seja, da música com um todo. O *fitness* do novo indivíduo gerado na ponderada não implica na soma dos *fitness* das porções e sim em um novo recálculo de distância entre todas as notas. Desta forma o *fitness* geral da ponderada poderá ser maior que a soma dos *fitness* parciais uma vez que entre a última nota da porção anterior e a primeira nota de uma porção subsequente há uma distância que deverá ser considerada.

É pouco provável que ao término da geração das subpopulações iniciais e da população ponderada os critérios sejam atendidos, e, caso não forem, o laço de repetição do algoritmo se inicia a fim de que os indivíduos sejam evoluídos.

Um método de seleção por torneio é executado para que seja possível realizar a seleção dos indivíduos mais aptos. Neste método, um indivíduo de cada subpopulação, além de um indivíduo da população ponderada são escolhidos para fazerem parte do torneio. Uma vez que

o algoritmo aplica o uso de elitismo, os melhores indivíduos das subpopulações são excluídos do torneio. Dentre os indivíduos selecionados, vence o que tiver melhor *fitness* relacionado à música como um todo.

A Figura 37 demonstra os indivíduos selecionados de cada subpopulação e da população ponderada onde o indivíduo da subpopulação 4 é o vencedor pois contém o menor *fitness*.

	Subpop1	Subpop2	Subpop3	Subpop4	Subpop5	Pond.
	G.Fitness	G.Fitness	G.Fitness	G.Fitness	G.Fitness	G. Fitness
Indiv. 1	2750	2880	1480	1550	2070	1785
Indiv. 2	1620	2900	2750	1590	2300	1993
Indiv. 3	2750	2940	1490	1700	2640	2156
Indiv. 4	1620	2990	1620	2750	2720	2370
Indiv. 5	3000	2080	1930	1620	3200	2490
Indiv. 6	3300	2750	2640	2000	3300	2562

Figura 37: Torneio entre os indivíduos das subpopulações.

Fonte: Autoria própria.

O processo de torneio segue separando os indivíduos vencedores em uma população intermediária até que a mesma esteja completa. A partir de então a população intermediária passa a ser evoluída através dos operadores de cruzamento e mutação.

Os operadores de *crossover* e de mutação são executados de forma idêntica ao genético tradicional. Sendo que inicialmente são selecionados, com base na taxa de *crossover*, os indivíduos para que sejam permutados. A Figura 38 permite a visualização de um *crossover* de 2 pontos sendo realizado em uma música contendo 21 notas. O segmento do meio é selecionado para ser trocado e dois novos indivíduos são gerados.

Em seguida também é selecionada uma quantidade pré-estabelecida de indivíduos para que seja realizada a mutação. Esta quantidade é determinada através de uma taxa de mutação responsável por indicar a quantidade de indivíduos que serão mudados. Uma segunda taxa de mutação fica responsável por determinar a quantidade de notas que sofrerão a mutação. Terminados os operadores de cruzamento e mutação, a primeira geração de indivíduos é avaliada em busca dos critérios de parada preestabelecidos.

É importante ressaltar que nesta etapa os indivíduos são avaliados com base na música como um todo através de um *fitness* geral. O único local onde os indivíduos são avaliados com base em suas porções é na formação das subpopulações. Caso algum critério de parada seja atendido, o algoritmo finaliza, caso contrário ele dá continuidade a etapa de migração.

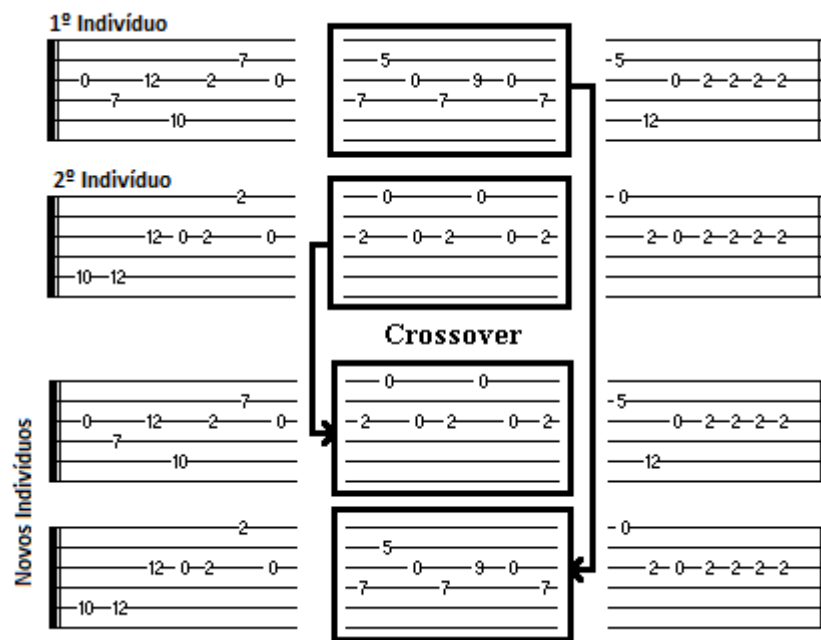


Figura 38: Operador de *crossover* aplicado nas notas da música.

Fonte: Autoria própria.

A etapa de migração ocorre a fim de retornar os indivíduos evoluídos para as subpopulações. O processo de migração faz com que os melhores indivíduos da população intermediária passem a ocupar o lugar dos piores indivíduos das subpopulações existentes. Isto é realizado tendo como base o *fitness* da porção correspondente.

Para que o processo ocorra, cada indivíduo da população intermediária é selecionado. Após a seleção, verifica-se o *fitness* referente à porção da primeira subpopulação. Caso o *fitness* deste indivíduo seja maior que do pior indivíduo da primeira subpopulação, o pior indivíduo é descartado e substituído pelo indivíduo selecionado.

Passa-se então a analisar o indivíduo selecionado na segunda subpopulação, averiguando, porém, a porção referente à segunda subpopulação.

O processo se repete até que o indivíduo selecionado seja analisado em todas as subpopulações. Isto significa que um indivíduo poderá ocupar o lugar de um indivíduo pior em mais de uma subpopulação diferente.

A Figura 39 exemplifica este processo de migração. Nela os valores expressos para P1, P2, P3, P4 e P5 representam o *fitness* para as 5 diferentes partes da música. É possível verificar que o *fitness* dos indivíduos da população intermediária são melhores que os piores indivíduos da primeira subpopulação. Sendo assim os piores indivíduos da primeira subpopulação são

descartados dando lugar aos melhores indivíduos da intermediária.

	Subpop1	Subpop2	Subpop3	Subpop4	Subpop5	Intermediária				
	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P1	P2	P3	P4	P5
Indiv. 1	450	270	420	360		215	290	256	156	249
Indiv. 2	480	300	450	420		372	343	370	370	356
Indiv. 3	540	330	480	450		429	356	490	490	378
Indiv. 4	600					566	470	562	562	569
Indiv. 5	600					857	490	785	785	685
Indiv. 6	660					913	562	893	993	930

Figura 39: Análise da primeira subpopulação na estratégia de migração.

Fonte: Autoria própria.

Caso não haja indivíduos melhores que se encaixem na subpopulação analisada em questão, dá-se continuidade na análise da próxima subpopulação até que todas as subpopulações sejam analisadas e migradas. A Figura 40 demonstra o processo de migração finalizado.

	Subpop1	Subpop2	Subpop3	Subpop4	Subpop5	Intermediária				
	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P1	P2	P3	P4	P5
Indiv. 1	450	270	420	360	240	215	290	256	156	249
Indiv. 2	480	300	450	420	300	372	343	370	370	356
Indiv. 3	540	330	480	450	330	429	356	490	490	378
Indiv. 4	215	290	480	450	249	566	470	562	562	569
Indiv. 5	372	343	256	156	356	857	490	785	785	685
Indiv. 6	429	356	370	370	378	913	562	893	993	930

Figura 40: Migração dos indivíduos finalizada.

Fonte: Autoria própria.

A população ponderada não entra nesta etapa de migração uma vez que existe uma etapa no algoritmo específica por recriar a ponderada.

A estratégia de migração privilegia os melhores indivíduos atuais da subpopulação que está sendo verificada e os melhores indivíduos da população intermediária. Finalizado o processo de migração, a última etapa do laço consiste em recriar a população ponderada.

A estratégia adotada para a recomposição da população ponderada é parecida com a estratégia de criação da população ponderada inicial. Nela os indivíduos da população ponderada são recriados a partir da concatenação da subpopulações que receberam os indivíduos migrados.

Uma vez concatenados, compara-se, através do *fitness* geral, se estes novos indivíduos gerados são melhores que os indivíduos da atual população ponderada. Se os indivíduos forem melhores, eles irão substituir os piores indivíduos da atual população ponderada. Desta forma permanecem na nova população ponderada apenas os indivíduos que apresentarem melhor *fitness* geral. Um exemplo para a estratégia de recomposição da população ponderada pode ser visualizado através da Figura 41.

	Subpop1	Subpop2	Subpop3	Subpop4	Subpop5		Pond.	Pond. Antiga	Nova Pond.
	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P. Fitness		G. Fitness	G. Fitness	G. Fitness
Indiv. 1	215	270	256	156	240	→	1150	1785	1150
Indiv. 2	372	290	370	360	249	→	1712	1993	1712
Indiv. 3	429	300	420	370	300	→	1899	2156	1785
Indiv. 4	450	330	450	420	330	→	1980	2370	1899
Indiv. 5	480	343	480	450	356	→	2149	2490	1980
Indiv. 6	540	356	480	450	378	→	2274	2562	1993

Figura 41: Estratégia de recomposição da população ponderada.

Fonte: Autoria própria.

3.5 ABORDAGEM UTILIZANDO O ACO

Por meio de um grafo direcional é possível representar o problema proposto utilizando a abordagem ACO. A extração da sequência de notas do arquivo MusicXML irá constituir um grafo onde cada vértice representa uma possível posição para a nota e as arestas, um caminho entre uma posição e as possíveis posições da próxima nota.

A distância do caminho entre dois vértices é calculada através da distância euclidiana, proposta na seção 3.2, entre suas referentes posições.

Após o grafo ter sido construído, uma formiga é posicionada em cada uma das possíveis posições para a nota inicial da música. Sendo assim, as formigas percorrem o grafo até o vértice referente à nota final da música a ser transcrita. A Figura 42 demonstra uma simulação dos caminhos feitos pelas formigas.

Cada caminho completo de uma formiga indica uma solução dentre as soluções disponíveis no espaço de busca para o problema apresentado.

Foram definidos valores para operadores que descrevem as variáveis envolvidas no ACO de forma que sejam compatíveis com o problema a ser resolvido (DORIGO et al., 2006).

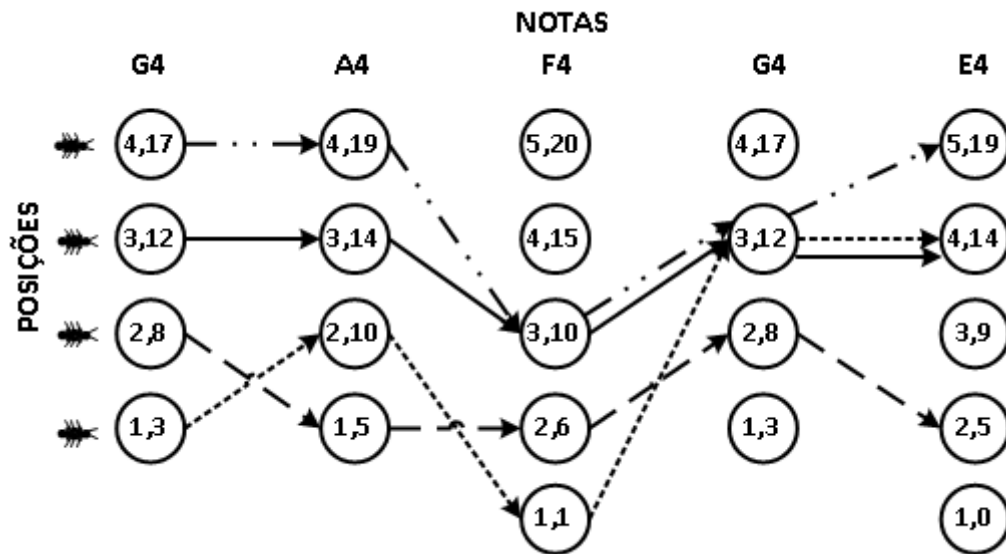


Figura 42: Possíveis caminhos percorridos pelas formigas.

Fonte: Autoria própria.

As variáveis envolvidas são: a constante de evaporação ρ , a importância relativa da pegada α , a importância relativa da visibilidade β e a constante de qualidade da pegada Q .

É importante destacar que um alto valor de α em relação a β faz com que a informação heurística tenha sua importância reduzida, levando a um pior rendimento do algoritmo. Uma constante mediana ρ possibilita a convergência sem que esta ocorra prematuramente para um ótimo local.

O operador β é modificado dinamicamente para incentivar as formigas a procurar outros caminhos menores e confirmar a confiabilidade da convergência ocorrida. Esta modificação do operador é realizada com base no *fitness* de cada iteração. Desta forma, caso o *fitness* se mantenha estagnado por 100 iterações, o valor de β é dobrado de maneira a aumentar a influência da qualidade heurística do caminho em relação a quantidade de feromônio das arestas. Caso após 400 iterações o *fitness* continuar estagnado o valor de β é dobrado novamente. Ao chegar a 500 iterações estagnadas o algoritmo é finalizado, pois entende-se que esta não é uma estagnação prematura devido a um alto acúmulo de feromônio.

3.6 ABORDAGEM UTILIZANDO A ED COM LISTA DE MOVIMENTOS

O uso da abordagem utilizando a evolução diferencial com lista de movimentos foi aplicada a este trabalho como alternativa para criar uma evolução diferencial com representação

discreta.

Para a realização da evolução diferencial, foi utilizada uma adaptação do esquema proposto por Storn e Price (1995) para realização da diferença vetorial em conjunto com a lista de movimentos por Prado et al. (2012) e Prado et al. (2010). Nesta adaptação, o vetor modificado é alcançado através da equação 14

$$V_{i,g} = X_{i,g} + F(X_{best,g} - X_{i,g}) + F(X_{r1,g} - X_{r2,g}) \quad (14)$$

onde $V_{i,g}$ representa o vetor modificado, $X_{i,g}$ o vetor alvo, $X_{best,g}$ o vetor referente ao melhor indivíduo da geração e $X_{r1,g}$, $X_{r2,g}$ como vetores aleatórios usados na diferença vetorial. O valor da escalar F é o mesmo para ambas as operações.

Após a importação dos dados encontrados na base de dados MusicXML para o ambiente computacional, o algoritmo ED se inicia. Em sua primeira etapa, uma população inicial é criada de forma aleatória a fim de garantir uma boa diversidade de indivíduos. Logo após é avaliada e inicia-se o processo de evolução dessa população até que atenda os critérios de parada estabelecidos. A Figura 43 demonstra o pseudocódigo utilizado para a abordagem proposta do algoritmo.

Após entrar no laço de repetição, caso o critério de parada não tenha sido satisfeito, o algoritmo proposto realiza a primeira etapa para que possa evoluir sua população.

Esta primeira etapa consiste em definir o vetor alvo, o vetor referente ao melhor indivíduo da geração e os vetores utilizados no processo da diferença vetorial. O vetor alvo, a cada iteração, é um indivíduo diferente da população que está sendo evoluída. Em seguida os vetores utilizado na diferença vetorial são escolhidos aleatoriamente bem como o melhor indivíduo da população.

Uma vez definidos os vetores, a próxima etapa do algoritmo realiza a geração da lista de movimentos.

Para o problema apresentado, a lista de movimentos consiste em encontrar quais posições são diferentes entre os vetores. Estas posições diferentes são denominadas movimentos. Uma parte dos movimentos se refere às posições diferentes entre o melhor indivíduo e o vetor alvo, a outra parte entre dois indivíduos aleatórios.

A Figura 44 demonstra um exemplo da geração destes movimentos para uma música composta de 10 notas. É possível observar os movimentos sendo apresentados através da tupla $\langle corda, casa, nota \rangle$ onde *nota* se refere ao número da nota na música.

```

1 Realizar a importação da música;
2 Gerar uma população inicial randômica;
3 Avaliar os indivíduos da população;
4 enquanto critério de parada não for satisfeito faça
5     para cada individuo da população faça
6         Defina os vetores  $X_{i,g}$ ,  $X_{best,g}$ ,  $X_{r1,g}$  e  $X_{r2,g}$ ;
7         Gere os movimentos  $M_{best} = (X_{best,g} - X_{i,g})$ ;
8         Gere os movimentos  $M_{rand} = (X_{r1,g} - X_{r2,g})$ ;
9         Gere a lista de movimentos  $M_{best,rand}$ ;
10        Aplique o escalar F à lista de movimentos  $M'_{best,rand} = F \times M_{best,rand}$ ;
11        Gere o vetor mutante  $V_{i,g} = X_{i,g} + M'_{best,rand}$ ;
12        para cada gene do Vetor Alvo  $X_{j,i,g}$  faça
13            se  $rand(0,1) \leq Cr$  então
14                Vetor experimental recebe gene do vetor modificado  $U_{j,i,g} = V_{j,i,g}$ ;
15                ;
16            senão
17                Vetor experimental recebe gene do vetor alvo  $U_{j,i,g} = X_{j,i,g}$ ;
18            fim
19        fim
20        Avaliar vetor experimental;
21        se  $aptidaoVetorExperimental \leq aptidaoVetorAlvo$  então
22            Nova população recebe vetor experimental;
23        senão
24            Nova população recebe vetor alvo;
25        fim
26 fim

```

Figura 43: Pseudocódigo proposto para o ED com lista de movimentos.

Fonte: Autoria própria.

Na primeira parte dos movimentos do exemplo, uma vez que a primeira, quarta, oitava e nona nota são diferentes entre os dois vetores envolvidos, foi gerado o conjunto de movimentos M_{best} possuindo as posições $\langle 1,3,1 \rangle$, $\langle 2,8,4 \rangle$, $\langle 1,1,8 \rangle$ e $\langle 1,5,9 \rangle$. Na segunda parte dos movimentos foi gerado o conjunto de movimentos M_{rand} possuindo as posições $\langle 1,1,3 \rangle$, $\langle 2,10,8 \rangle$.

Após a geração dos conjuntos de movimentos, estes são submetidos à aplicação do escalar F a fim de que sejam ponderados e posteriormente combinados para gerar a lista de movimentos. A Figura 45 demonstra a aplicação deste escalar para $F = 0,5$ onde os movimentos $\langle 1,1,8 \rangle$ e $\langle 1,5,9 \rangle$ de M_{best} e o movimento $\langle 2,10,8 \rangle$ de M_{rand} são eliminados do conjunto.

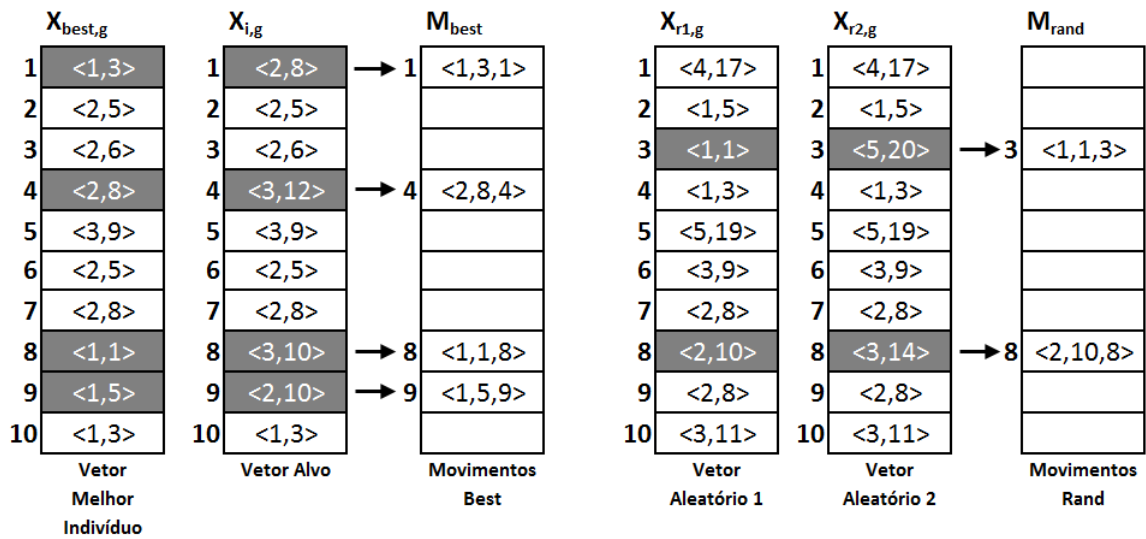


Figura 44: Geração do conjunto de movimentos para abordagem proposta.

Fonte: Autoria própria.

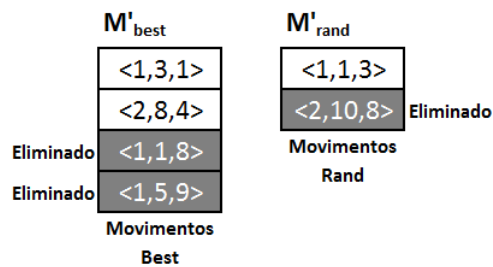


Figura 45: Exemplo de aplicação da escalar F .

Fonte: Autoria própria.

A combinação dos movimentos para a geração da lista de movimentos é realizada da seguinte forma: o primeiro movimento de um dos vetores é selecionado para compor a lista, o segundo movimento é selecionado do outro vetor. Esta alternância entre as seleções segue até que não haja mais movimentos a serem selecionados para formar a lista de movimentos.

Caso um conjunto de movimentos seja maior que o outro, quando não houver mais movimentos de um conjunto para serem selecionados e o outro conjunto ainda possuir, todos os movimentos do conjunto restante são selecionados para compor a lista de movimentos. A Figura 46 demonstra a geração de uma lista de movimentos a partir dos conjuntos de movimentos M'_{best} e M'_{rand} .

Por fim a lista de movimentos é aplicada ao vetor alvo a fim de gerar o vetor mudado. A aplicação da lista de movimentos consiste em substituir as posições do vetor alvo pelos movimentos contidos na lista de movimentos. A Figura 47 exemplifica a aplicação da lista.

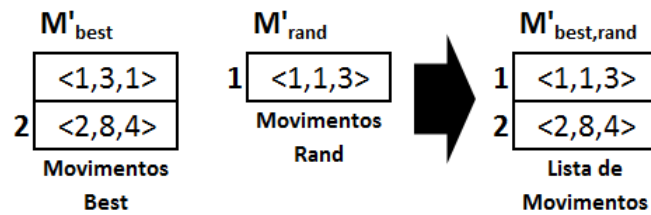


Figura 46: Exemplo de geração da lista de movimentos para a abordagem proposta.

Fonte: Autoria própria.

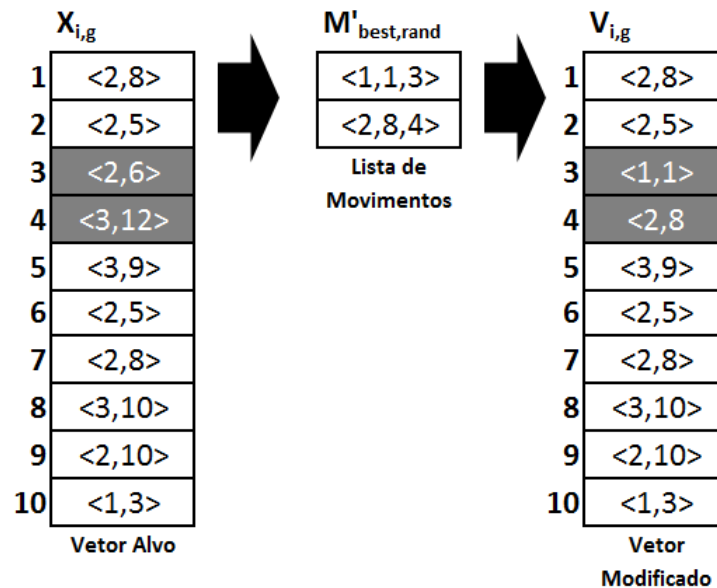


Figura 47: Exemplo da aplicação da lista de movimentos.

Fonte: Autoria própria.

A próxima etapa do algoritmo consiste em criar o vetor experimental a partir do vetor alvo e do vetor modificado. A criação do vetor experimental está sujeita a uma taxa de *crossover* Cr . Para um número aleatório gerado entre 0 e 1, se este número for menor ou igual à taxa então a posição do vetor experimental receberá a posição do vetor modificado, caso contrário receberá do vetor alvo. Este processo se repete até que todo o vetor experimental receba os genes.

Ao término da geração do vetor experimental, este é avaliado e caso sua aptidão seja melhor que o vetor alvo, o vetor experimental é selecionado para formar a população da próxima geração, caso contrário o vetor alvo é selecionado.

3.7 ABORDAGEM UTILIZANDO A BUSCA LOCAL 2-OPT E 3-OPT

Conforme descrito na seção 2.1.2, utilizando a metodologia 2-opt, duas arestas são quebradas e reconectadas com outros dois vértices. Por outro lado, a metodologia 3-opt utiliza três arestas. Estas abordagens foram utilizadas em conjunto com as demais técnicas exploradas com a finalidade de auxiliar a busca de melhores soluções.

Para o problema proposto, não é possível realizar a troca de uma aresta para um vértice referente a outra nota que não seja a nota seguinte. Se isso ocorresse, estaríamos alterando a sequência de notas da música e conseqüentemente criando uma música diferente da original.

É importante ressaltar que para a abordagem 2-opt e 3-opt utilizada, a troca é realizada entre a posição referente à nota atual (vértice atual) e as possíveis posições da próxima nota a ser tocada (possíveis vértices seguintes).

A Figura 48 representa uma solução para uma música composta de 5 notas musicais onde $\langle 2,8 \rangle$, $\langle 1,5 \rangle$, $\langle 2,6 \rangle$, $\langle 2,8 \rangle$ e $\langle 2,5 \rangle$ são as posições respectivas a estas notas. É possível observar também que as arestas $A = \{ \{ \langle 2,8 \rangle, \langle 1,5 \rangle \}, \{ \langle 2,8 \rangle, \langle 2,5 \rangle \} \}$ foram selecionadas para serem quebradas e reconectadas.

Já na Figura 49 podemos observar as arestas reconectadas usando o movimento 2-opt, dando assim origem a um novo caminho, porém sem alterar a estrutura da sequência de notas referente à música.

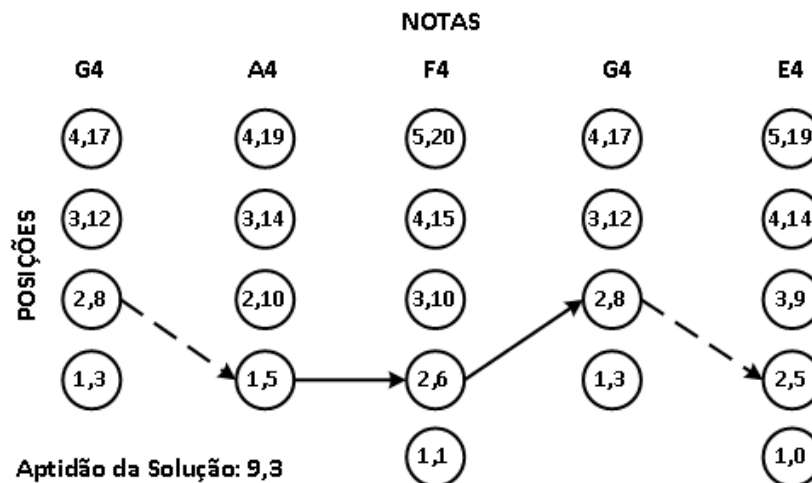


Figura 48: Solução encontrada para uma música de 5 notas.

Fonte: Autoria própria.

Para a aplicação destes algoritmos de busca, inicialmente foi definida uma taxa que

indica a quantidade de indivíduos que sofrem a busca local. Esta taxa tem como finalidade diminuir a ocorrência da busca em toda a população de maneira a permitir que o algoritmo não seja tão penalizado em seu tempo de execução.

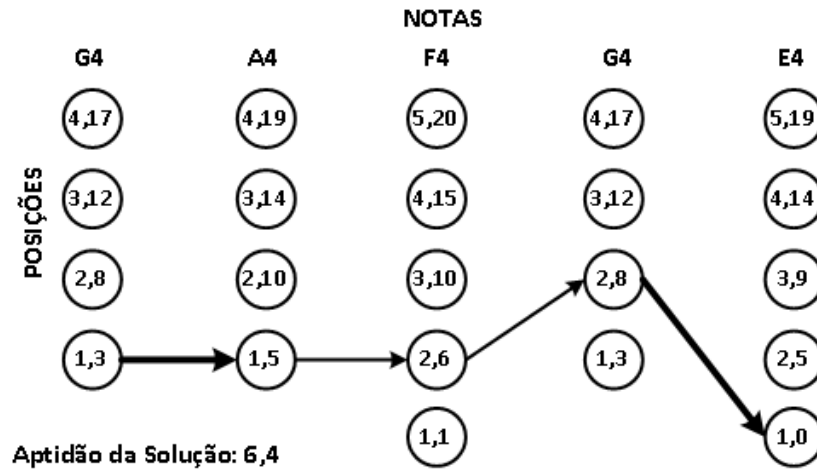


Figura 49: Troca realizada pelo 2-opt.

Fonte: Autoria própria.

Também foi definido que o algoritmo iterará até que seja encontrada uma solução melhor que a atual. A quantidade de tentativas é o resultado absoluto da equação 15

$$qt = \text{abs}(nn/t) \quad (15)$$

onde nn é o número de notas que a música possui e t o tipo de busca: 2 para 2-opt e 3 para 3-opt. Por exemplo: uma música de 150 notas utilizando o algoritmo de busca 2-opt resulta em 75 tentativas de troca. Já uma música de 250 notas utilizando o algoritmo 3-opt resulta em 83 tentativas.

Uma vez definidos estes parâmetros, o algoritmo inicia o processo de busca local para os indivíduos da população. O pseudocódigo da aplicação deste algoritmo pode ser visualizado na Figura 50.

A estratégia de busca local foi combinada com os algoritmos evolutivos com o objetivo de melhorar os resultados obtidos.

Nos algoritmos genéticos, tanto o método tradicional quanto com uso de subpopulações, a etapa de busca local foi inserida logo após a aplicação dos operadores de mutação e *crossover*.

Já no algoritmo de evolução diferencial, a técnica foi aplicada após a seleção dos indivíduos para formar a população da próxima geração, ou seja, como última etapa do laço da geração.

```

1 Define a quantidade de indivíduos a serem aplicados a busca local;
2 Define a quantidade de trocas a serem efetuadas;
3 para cada indivíduo da população a se aplicado a busca faça
4   para cada quantidade de trocas disponíveis faça
5     para cada quantidade de arestas a serem trocadas faça
6       Selecione uma nota aleatória;
7       Selecione uma nova posição (casa e corda) para a nota aleatória;
8     fim
9     Avalie a nova solução com base nas novas posições;
10    se NovaAptidão < AptidãoOriginal então
11      Notas do individuo atual permanecem com as novas posições;
12      Break;
13    senão
14      Notas do indivíduo atual retornam às suas posições originais;
15    fim
16  fim
17 fim

```

Figura 50: Pseudocódigo da busca local 2-opt e 3-opt.

Fonte: Autoria própria.

4 RESULTADOS E DISCUSSÕES

4.1 BASE DE DADOS E CONFIGURAÇÕES GERAIS

Com a finalidade de realizar uma comparação entre os diferentes algoritmos evolucionários apresentados, foram utilizados 87 diferentes instâncias, cada qual com uma quantidade diferente de notas, sendo que todas as músicas utilizadas são monofônicas, ou seja, não incluem o uso de acordes. A fim de avaliar o desempenho de cada algoritmo em relação ao número de notas existentes em cada música, as músicas foram separadas em 4 grupos de acordo com o número de notas, conforme descrito na Tabela 4.

A base de dados das músicas utilizadas contendo os grupos, a quantidade de notas e os valores de ótimo para o *fitness*, pode ser visualizada¹ na Tabela 13. Os valores de ótimo foram encontrados através do algoritmo proposto por Dijkstra (1959).

Tabela 4: Divisão das músicas em grupo de notas.

Grupo	Número de Notas	Quantidade de Instâncias
1	até 100	15
2	101-200	25
3	201-300	33
4	acima de 301	14

Fonte: Autoria própria.

Para realizar os experimentos com os algoritmos genéticos, os parâmetros utilizados foram escolhidos de forma empírica. Diante disto, foram realizados diversos experimentos com diferentes combinações de valores a fim de se buscar o melhor conjunto de parâmetros a ser utilizado. Os detalhes dos melhores parâmetros utilizados podem ser visualizados na Tabela 5.

Os melhores valores encontrados para cada parâmetro estão destacados na Tabela 6. O método seleção por torneio utilizado na abordagem AG-SP é a partir da seleção aleatória entre os indivíduos de cada uma das subpopulações. Para o AG tradicional, foi utilizada apenas uma população com 150 indivíduos e a seleção por torneio entre 3 indivíduos. O critério de

¹Disponível no apêndice A deste trabalho.

parada utilizado para os algoritmos AG e AG-SP foi por estagnação do melhor *fitness* após 500 iterações. Caso contrário, os algoritmos são finalizados após 20 mil iterações.

Tabela 5: Parâmetros utilizados na análise combinatória.

Parâmetros Combinados	Valores	Acréscimos
Tamanho da população	50 a 400	50
Taxa de mutação (Notas)	40% a 70%	5%
Taxa de mutação (indivíduos)	5 a 20	5
Taxa de <i>crossover</i>	50 a 90%	5%
Números de pontos de corte	1 a 5	1

Fonte: Autoria própria.

Tabela 6: Parâmetros utilizados nos experimentos com os algoritmos genéticos.

Parâmetro	Valor AG	Valor AG-SP
Quantidade de populações	1	15
Quantidade de indivíduos por população	150	10
Taxa de mutação (notas)	50%	50%
Taxa de mutação (indivíduos)	60%	60%
Tipo de <i>crossover</i>	Uniforme	Uniforme
Taxa de <i>crossover</i>	55%	55%
Pontos de corte	3	3
Número máximo de gerações	20.000	20.000
Estagnação	500	500
Elitismo	Sim	Sim
Tipo de seleção	Torneio 3 indivíduos	Torneio subpopulações
Tipo de busca local	2-opt	2-opt

Fonte: Autoria própria.

Para realizar os experimentos com a otimização de colônia de formigas (ACO), os parâmetros utilizados também foram encontrados de forma empírica levando em consideração o fato do *trail* ser cumulativo e também o valor de α ser menor que β de tal forma que não seja insignificante mas permita uma convergência. O melhor conjunto de parâmetros resultantes está apresentado na Tabela 7. Por fim, para os experimentos com a evolução diferencial, os parâmetros empiricamente escolhidos são apresentados na Tabela 8.

A execução dos experimentos foi realizada em um mesmo equipamento de testes. As configurações do equipamento utilizado são expressas na Tabela 9.

A importação dos dados encontrados na base de dados MusicXML para o ambiente computacional é realizada através de uma interface de análise XML encontrado na própria na linguagem de programação utilizada. A interface é utilizada para buscar e coletar as marcações referentes ao compasso, tom, oitava e armação das notas musicais onde estas informações são

utilizadas pelos algoritmos para a execução da transcrição.

Tabela 7: Parâmetros utilizados nos experimentos com o ACO.

Parâmetro	Valor
α	0,15
β	1
ρ	0,5
Q	200
Estagnação	500

Fonte: Autoria própria.

Tabela 8: Parâmetros utilizados nos experimentos com a evolução diferencial.

Parâmetro	Valor
Quantidade de indivíduos por população	80
Taxa de <i>crossover</i>	55%
Número máximo de gerações	20.000
Estagnação	500
Tipo de busca local	3-opt

Fonte: Autoria própria.

Tabela 9: Hardware e software utilizados.

Tipo	Modelo/Versão
Processador	Intel® Xeon® CPU X5675 3.07GHz
Memória RAM	16Gb
Sistema Operacional	Windows Server 2008 R2
Linguagem de programação	Java 1.8.0_25-b18

Fonte: Autoria própria.

4.2 RESULTADOS EXPERIMENTAIS

Os resultados com as abordagens evolutivas apresentadas foram obtidos a partir da execução de 30 experimentos para cada música. A quantidade de experimentos com o algoritmo A* está relacionada de acordo com a quantidade de possíveis posições iniciais para cada música. Com isso, foram executadas todas as combinações alterando-se o nó raiz inicial de acordo com todas as possíveis posições iniciais.

Para cada execução foram obtidos o *fitness* e o tempo de execução. Desta forma, foram calculadas a média e o desvio padrão de cada instância. Os resultados obtidos foram inseridos em tabelas²separadas por grupo. As tabelas 14 e 15 mostram os resultados de *fitness* e tempo

para o grupo 1. As tabelas 16 e 17 para o grupo 2, as tabelas 18 e 19 para o grupo 3 e as tabelas 20 e 21 para o grupo 4.

Com o objetivo de avaliar o comportamento de cada algoritmo durante o processo de busca pelas melhores soluções, as figuras 51, 52, 53 e 54 ilustram os gráficos de convergência entre as diferentes abordagens evolutivas desenvolvidas. Para a geração destes gráficos, foram consideradas músicas de grupos distintos.

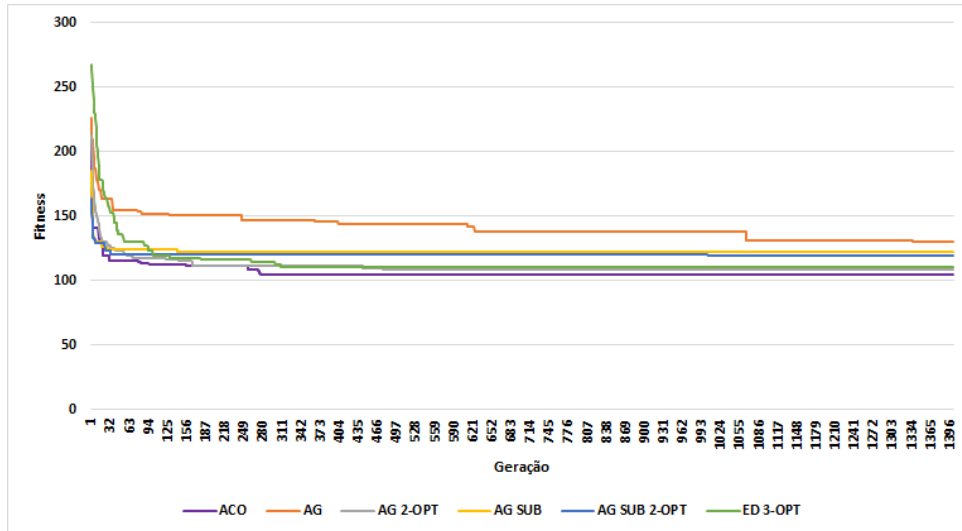


Figura 51: Convergência da música Katie's Lilt (grupo 1).

Fonte: Autoria própria.

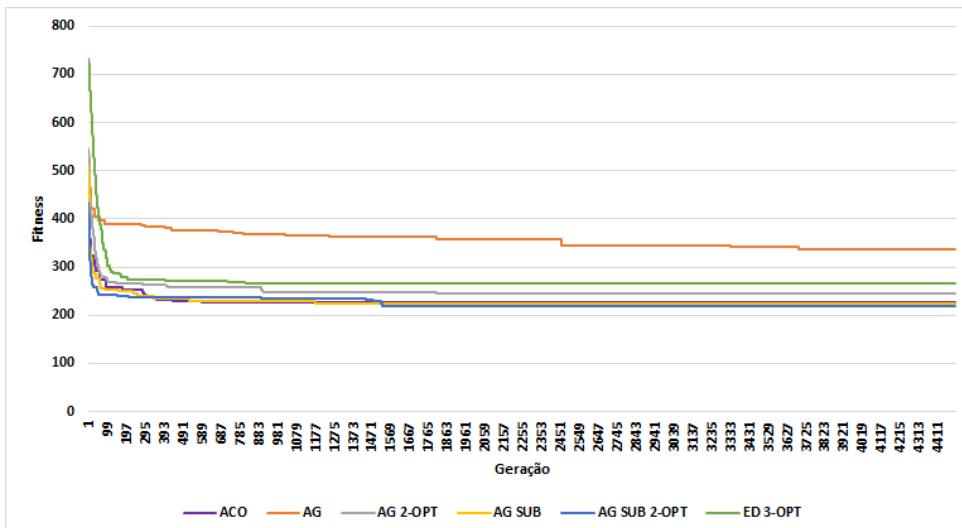


Figura 52: Convergência da música Smiling Bride, The (grupo 2).

Fonte: Autoria própria.

¹ As tabelas com resultados dos grupos estão disponíveis no apêndice B deste trabalho.

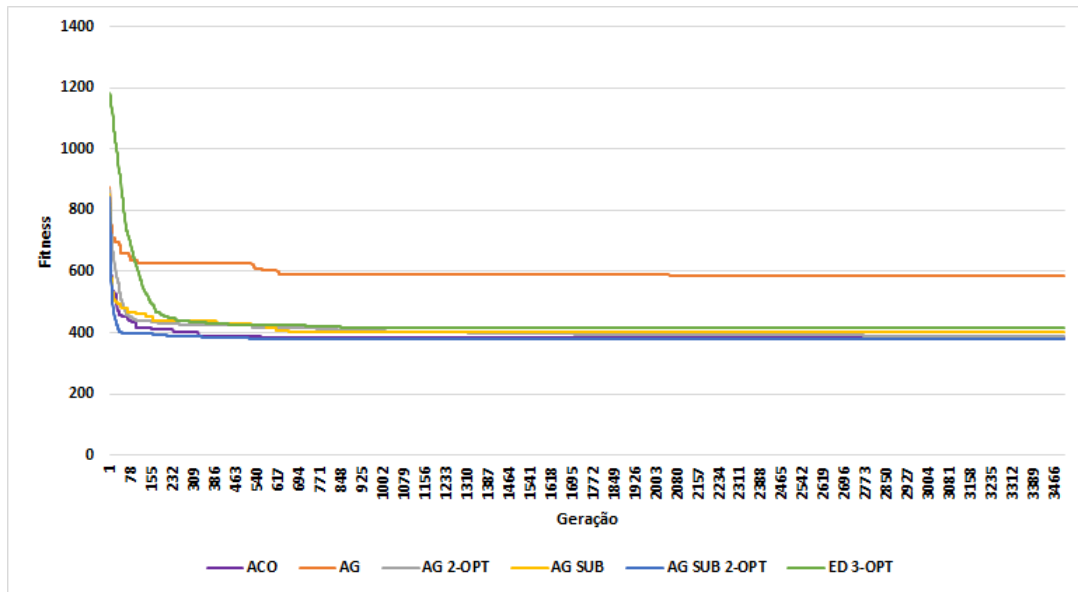


Figura 53: Convergência da música Fifty Five (grupo 3).

Fonte: Autoria própria.

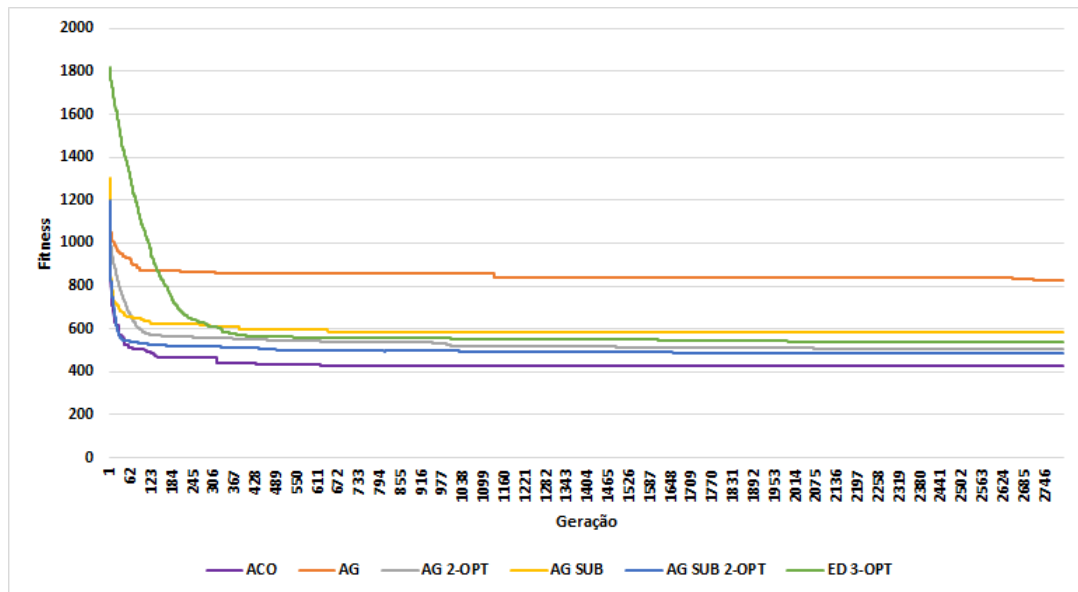


Figura 54: Convergência da música Suleman's Kopanitsa (grupo 4).

Fonte: Autoria própria.

4.3 RANKING POR DOMINÂNCIA

Neste trabalho é proposto uma análise de ranqueamento (*ranking*) por dominância dos algoritmos. Diante disto, a Tabela 10 apresenta os resultados de *ranking* por dominância aplicados nos resultados apresentados nas tabelas 14, 16, 18 e 20. Por outro lado, a Tabela 11

apresenta a posição (classificação) de cada algoritmo baseado no *ranking* da Tabela 10.

De acordo com os resultados das tabelas 10 e 11 é possível observar que os algoritmos que apresentaram melhores desempenho foram os algoritmos ACO, AG-SP 2-opt e AG 2-opt, com um *ranking* total de dominância de 370, 354 e 104, respectivamente. É importante destacar o bom desempenho do algoritmo ACO que obteve a primeira posição na ordenação por *ranking* na maioria dos grupos.

Tabela 10: Ranqueamento dos resultados baseado no *fitness* dos algoritmos.

Grupo	ACO	AG-SP	AG-SP 2-opt	AG	AG 2-opt	A*	ED 3-opt
1	56 / 90	-34 / 90	16 / 90	-82 / 90	48 / 90	-48 / 90	44 / 90
2	82 / 150	46 / 150	134 / 150	-150 / 150	-2 / 150	-78 / 150	-32 / 150
3	156 / 198	-16 / 198	152 / 198	-198 / 198	34 / 198	-82 / 198	-46 / 198
4	76 / 84	-42 / 84	52 / 84	-84 / 84	24 / 84	-14 / 84	-12 / 84
Todos	370 / 522	-46 / 522	354 / 522	-514 / 522	104 / 522	-222 / 522	-46 / 522

Fonte: Autoria própria.

Tabela 11: Ranqueamento dos melhores métodos.

Grupo	Ranqueamento						
	1 ^o	2 ^o	3 ^o	4 ^o	5 ^o	6 ^o	7 ^o
Grupo 1	ACO	AG 2-opt	ED 3-opt	AG-SP 2-opt	AG-SP	A*	AG
Grupo 2	AG-SP 2-opt	ACO	AG-SP	AG 2-opt	ED 3-opt	A*	AG
Grupo 3	ACO	AG-SP 2-opt	AG 2-opt	AG-SP	ED 3-opt	A*	AG
Grupo 4	ACO	AG-SP 2-opt	AG 2-opt	ED 3-opt	A*	AG-SP	AG
Todos	ACO	AG-SP 2-opt	AG 2-opt	AG-SP	ED 3-opt	A*	AG

Fonte: Autoria própria.

Com o intuito de avaliar se os valores de *ranking* obtidos (Tabela 10) apresentam diferenças significativas entre os algoritmos, foi aplicado neste trabalho um teste de hipótese estatístico. Primeiramente foi aplicado o teste de normalidade em relação aos dados obtidos. Como os dados apresentaram uma distribuição normal, o teste paramétrico de análise de variância, também conhecido por *Analysis of Variance* ou simplesmente ANOVA, proposto por Fisher (1925) foi utilizado. Após a aplicação deste teste foi possível observar um *p*-valor abaixo de 0.05, indicando que ao menos um par de algoritmos é estatisticamente diferente. Com isso, foi aplicado a abordagem pós-teste proposta por Tukey (1949), conhecida por *post hoc* de Tukey (TSD), para verificar quais os algoritmos apresentam diferenças com nível de confiança de 95% entre os *rankings*. Na Tabela 12 são apresentados os resultados obtidos com o teste *post hoc* de Tukey.

Como pôde ser observado na Tabela 12, novamente o algoritmo ACO apresentou um melhor desempenho, com diferenças com nível de confiança de 95% em relação aos algoritmos AG, A*, ED e AG-SP. Por outro lado, os outros dois melhores algoritmos AG-SP 2-opt e AG 2-opt não apresentaram as mesmas diferenças obtidas pelo algoritmo ACO. Neste caso, o AG-SP 2-opt apresentou diferenças apenas em relação aos algoritmos AG e A* enquanto que o algoritmo AG 2-opt apresentou diferenças apenas em relação ao algoritmo AG.

Tabela 12: Diferença significativa entre os algoritmos.

Algoritmo 1	Algoritmo 2	<i>p</i> -valor	Diferença significativa
AG	ACO	0,0000115	SIM
AG 2-opt	ACO	0,3816802	NÃO
AG-SP	ACO	0,0448905	SIM
AG-SP 2-opt	ACO	0,9999995	NÃO
A*	ACO	0,0020004	SIM
ED 3-opt	ACO	0,0448905	SIM
AG 2-opt	AG	0,0012469	SIM
AG-SP	AG	0,0184905	SIM
AG-SP 2-opt	AG	0,0000151	SIM
A*	AG	0,2812986	NÃO
ED 3-opt	AG	0,0184905	SIM
AG-SP	AG 2-opt	0,8899213	NÃO
AG-SP 2-opt	AG 2-opt	0,4520291	NÃO
A*	AG 2-opt	0,1796278	NÃO
ED 3-opt	AG 2-opt	0,8899213	NÃO
AG-SP 2-opt	AG-SP	0,0583831	NÃO
A*	AG-SP	0,7972559	NÃO
ED 3-opt	AG-SP	1,0000000	NÃO
A*	AG-SP 2-opt	0,0026753	SIM
ED 3-opt	AG-SP 2-opt	0,0583831	NÃO
ED 3-opt	A*	0,7972559	NÃO

Fonte: Autoria própria.

Com base nesta análise, é possível observar que o algoritmo ACO apresentou um melhor desempenho em relação aos demais algoritmos desenvolvidos, encontrando com maior frequência regiões promissoras do espaço de busca. Outro importante aspecto a ser destacado é em relação a inclusão da busca local nos algoritmos, que apresentou melhora no mecanismo de busca, permitindo melhores resultados na maioria dos experimentos realizados. Finalmente, é possível observar o fraco desempenho de uma busca heurística simples como o A*, que na maioria dos experimentos não alcançou as regiões mais promissoras do espaço de busca.

4.4 COMPARAÇÃO COM O PROGRAMA MUSESCORE

Nesta seção é apresentada uma pequena comparação entre as posições geradas pelos algoritmos ACO, AG-SP 2-opt e AG 2-opt, definidos neste trabalho como as abordagens com melhor de desempenho e as posições fornecidas pelo programa MuseScore 2.

Foi possível observar que o programa MuseScore adota alguns padrões para a realização da transcrição. Entre os padrões encontrados, foi possível verificar que ele prioriza a inserção das notas na primeira corda do instrumento e opta por outra corda apenas nos casos em que não é possível reproduzir a nota através da corda 1.

O MuseScore não levou em consideração também o deslocamento da mão e dos dedos ao realizar a transcrição. Isto é possível observar através da Figura 55 que representa um trecho de uma transcrição realizada pelo MuseScore para a música de instância 15 (*Mediator, The*).



Figura 55: Trecho de uma transcrição³ realizada pelo MuseScore.

Fonte: Autoria própria.

O fato de o programa MuseScore não levar em consideração o deslocamento dos dedos pode impedir uma boa transcrição.

Se observarmos novamente a Figura 55, a mudança entre a posição $\langle 1, 5 \rangle$ para a posição $\langle 1, 10 \rangle$ no 4º compasso e também da posição $\langle 1, 7 \rangle$ para $\langle 1, 15 \rangle$ no 5º compasso exige um deslocamento muito grande da mão do músico. Com isso a música se torna muito mais difícil de ser tocada podendo até chegar ao ponto de ser impossível de ser tocada caso não se consiga executar as notas dentro do tempo correto na música.

A Figura 56 exhibe alternativas que poderiam ter sido melhor escolhidas para a transição entre estas posições. Para a 3ª posição referente ao 4º compasso ($\langle 1, 5 \rangle$), uma alternativa seria a escolha da posição $\langle 2, 10 \rangle$. Por outro lado o MuseScore poderia ter levado em consideração

³Veja a transcrição completa no apêndice C deste trabalho.

as posições anteriores a fim de evitar que apenas a mudança da posição $\langle 1, 5 \rangle$ para $\langle 2, 10 \rangle$, apesar de diminuir a distância com a próxima nota ($\langle 1, 10 \rangle$), fizesse com que a nova posição $\langle 2, 10 \rangle$ se tornasse muito distante das posições anteriores.

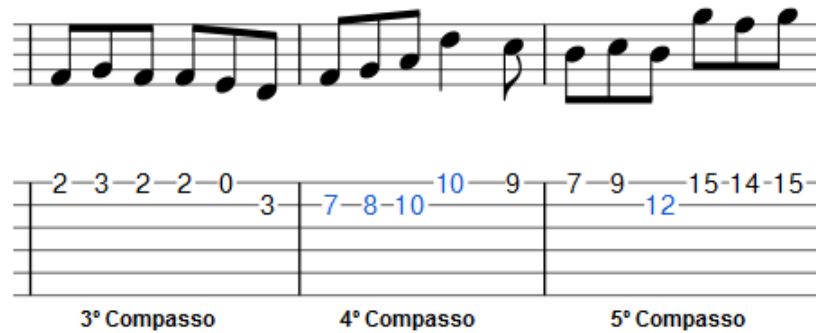


Figura 56: Alternativas para posições melhor transcritas.

Fonte: Autoria própria.

Já para a 3ª posição do 5º compasso ($\langle 1, 7 \rangle$) um simples ajuste seria suficiente para melhorar a transcrição. Transcrevendo a posição desta nota para $\langle 2, 12 \rangle$ seria suficiente para diminuir consideravelmente a distância entre suas posições vizinhas.

Se analisar a diferença entre os padrões de transcrição do MuseScore com as abordagens de melhor desempenho definidas neste trabalho pode-se observar através da Figura 57 que os algoritmos ACO, AG-SP 2-opt e AG 2-opt realizam a transcrição das posições de forma distribuída entre as cordas. Isto permite que as distâncias entre as posições sejam menores, uma vez que as notas no braço do violão tendem a se concentrar em regiões específicas do braço.

Outras padrões observáveis na música se referem ao fato do AG-SP 2-opt ter se concentrado nas regiões mais altas do braço do violão, ou seja, da casa 11 à casa 24.

As primeiras posições do AG 2-opt são encontradas no início do braço do violão, mas especificamente nas casas 2 e 3, e posteriormente as posições concentraram-se no final do braço, entre as casas 14 à 24. Já o ACO explorou o braço do violão como um todo escolhendo posições no início, meio e final do braço.

Tanto o AG 2-opt quanto o ACO usaram da posição $\langle 1, 0 \rangle$ como uma oportunidade para se deslocarem para outras regiões do braço do violão.

Estas mudanças podem ser benéficas, pois permitem a exploração de todo o braço do violão sem a necessidade de uma evolução gradativa através das casas. Por outro lado, se ocorrerem com muita frequência podem tornar a música difícil de se executada. O deslocamento de uma posição de casa 0 para outra posição é favorável uma vez que permite que o músico tenha

tempo de deslocar sua mão pelo braço do violão enquanto toca a corda solta vinculada a esta casa 0.

The figure displays a musical score for guitar, comparing a standard MuseScore notation with four alternative guitar-specific approaches: ACO, AG-SP 2-opt, and AG 2-opt. The score is organized into four measures, labeled '7° Compasso' through '10° Compasso'. The MuseScore staff at the top shows a melody with two first endings, marked '1' and '2'. The tablature systems below provide fret numbers for each measure, with some measures containing multiple lines of fret numbers to represent different fingerings or techniques. The ACO system uses higher fret numbers (up to 21) and includes a '17' in the 8th measure. The AG-SP 2-opt system uses fret numbers up to 17. The AG 2-opt system uses fret numbers up to 20. The 10th measure in all systems concludes with a double bar line and a final fret number (14-15 or 19-20).

Figura 57: Comparativo entre as melhores abordagens e o MuseScore.

Fonte: Autoria própria.

5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 CONCLUSÕES

O uso de tablaturas facilita o entendimento e a execução de peças musicais, principalmente se a execução for realizada por algum músico iniciante e que está em processo de aprendizado ou até mesmo por algum profissional que já está habituado com determinada música. Diante disto, o grande desafio no processo de transcrição de partituras em tablaturas é o de encontrar a combinação correta para se utilizar em cada nota musical, tornando um problema de explosão combinatorial.

Neste caso, uma alternativa consiste na utilização de abordagens heurísticas e meta-heurísticas que permitam uma exploração adequada do espaço de busca, verificando as melhores combinações de posições para se tocar a música.

Neste trabalho foi proposta a investigação de diversas abordagens evolutivas com o objetivo de verificar as características de cada abordagem que contribuem de forma significativa para a resolução do problema de transcrição musical.

Diante disto, foi possível observar que o algoritmo ACO apresentou melhores resultados para a maioria das músicas utilizadas, reforçando o bom desempenho do mecanismo de busca probabilística deste algoritmo, que alcançou regiões mais promissoras no espaço de busca quando comparado com as outras abordagens desenvolvidas. Por outro lado, a inclusão da busca local permitiu melhoria nas características *exploitation* e *exploration* dos algoritmos genéticos desenvolvidos. Já a abordagem evolução diferencial, apresentou resultados promissores para o primeiro grupo avaliado. Entretanto, para os demais grupos compostos por músicas com maior quantidade de notas, o algoritmo apresentou dificuldades no processo de busca com falta de diversidade entre as soluções.

As abordagens clássicas A* e algoritmo genético tradicional que foram utilizadas por outros pesquisadores em outros trabalhos com foco no tratamento do problema de transcrição automática de partitura em tablatura, não alcançaram resultados promissores, apresentando um

mecanismo fraco no processo de busca e, conseqüentemente, obtiveram um pior desempenho em relação aos demais algoritmos desenvolvidos.

Finalmente, este trabalho apresentou um estudo detalhado entre diversas abordagens evolutivas compostas por diferentes mecanismos de busca no espaço de soluções. A partir do estudo desenvolvido neste trabalho, foi possível observar que abordagens evolutivas baseadas em colônias de formigas e abordagens evolutivas usando técnicas de busca local são alternativas promissoras em relação às abordagens clássicas que são comumente utilizadas para o tratamento do problema de transcrição musical.

Podemos observar também que o *software* MuseScore não leva em consideração o deslocamento necessário para reprodução da música, bem como prioriza a transcrição na primeira corda do instrumento. Diferente destas características, os métodos evolutivos que apresentaram melhores resultados, quando comparados com o *software* MuseScore, exploram melhor o braço do violão em busca de posições melhores.

5.2 PERSPECTIVAS FUTURAS

Nos artigos estudados, o processo de cálculo dos pesos se dá utilizando a distância vertical e a distância horizontal em uma mesma função objetivo. Uma vez separados estes dois objetivos acredita-se que o processo de conversão poderá apresentar melhores resultados. Isto indica o uso de algoritmos biobjetivos ou até mesmo multiobjetivos caso seja incluso outros tipos de problemas para serem otimizados.

Face ao exposto, uma possível continuidade para este trabalho é o caráter multiobjetivo do problema tratado, adaptando a abordagem AG-SP de forma que seja possível a inclusão de outras características de uma partitura, como por exemplo: acordes, tempo de silêncio entre as notas, efeitos especiais encontrados no violão, sonoridade entre outros recursos.

Com isso, uma adaptação na funções de *fitness* também será necessária, uma vez que, o problema não será mais concentrado na minimização das distâncias das notas utilizadas em uma determinada música.

Outro aspecto a ser considerado é em relação à tessitura do instrumento, ou seja, a extensão de notas em que um instrumento pode ser tocado. A tessitura do piano é diferente de um violão. Esta tessitura é identificada através de sua nota mais grave e sua nota mais aguda, ambas acompanhadas de suas respectivas oitavas. Um piano de 88 teclas possui uma extensão que vai do A0 (Lá na oitava 0) ao C8 (Dó na oitava 8).

Na Figura 58 podemos visualizar o espaço de notas entre 27,5Hz e 4.186Hz comumente encontrado em um piano. O violão por outro lado não possui a mesma extensão de um piano, mas pode abranger até 5 oitavas diferentes. Esta restrição pode ser um problema a ser explorado.

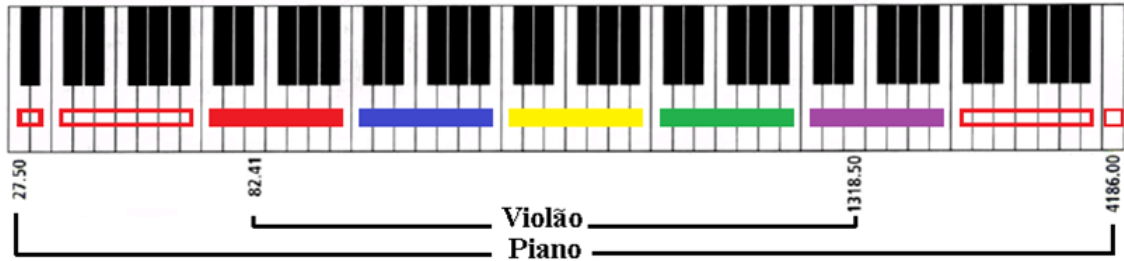


Figura 58: Comparação da tessitura entre o violão e o piano.

Fonte: Adaptado de Sengpiel (2014)

Além disso, outro aspecto que poderá ser explorado é o fato de que as notas que não se repetem concentram-se em 5 regiões distintas do braço do violão. Isto pode ser melhor observado na Figura 59. Uma alternativa, seria incluir o uso deste conhecimento para tratar as notas as serem escolhidas no processo de mutação ou até mesmo relacionar a função heurística de acordo com as notas encontradas na música a ser convertida.

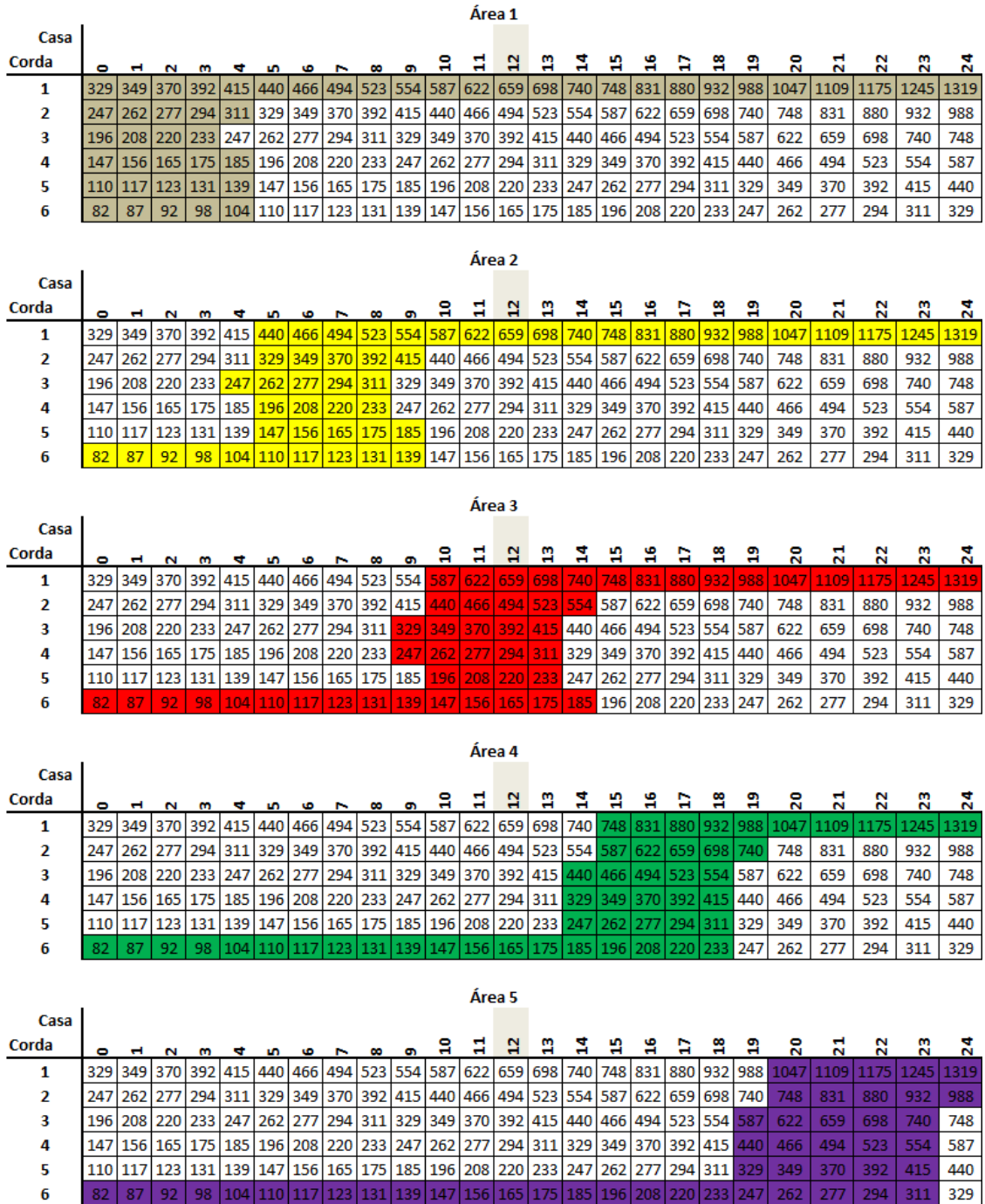


Figura 59: Regiões de concentração das notas no braço do violão.

Fonte: Autoria própria.

REFERÊNCIAS

- BELL, J. E.; MCMULLEN, P. R. Ant colony optimization techniques for the vehicle routing problem. **Advanced Engineering Informatics**, Elsevier, v. 18, n. 1, p. 41–48, 2004.
- BENNETT, R. Elementos básicos da música. **Rio de Janeiro: Jorge Zahar**, 1998.
- BLUM, C. Ant colony optimization: Introduction and recent trends. **Physics of Life reviews**, Elsevier, v. 2, n. 4, p. 353–373, 2005.
- BURLET, G. **Automatic guitar tablature transcription online**. Master's thesis — McGill University, Montréal QC, 2013.
- BURLET, G.; FUJINAGA, I. Robotaba guitar tablature transcription framework. In: **ISMIR**. [S.l.: s.n.], 2013. p. 517–522.
- CORMEN, T. H. et al. **Introduction to algorithms**. 3. ed. [S.l.]: The MIT Press, 2009.
- CROES, G. A. A method for solving traveling-salesman problems. **Operations research, INFORMS**, v. 6, n. 6, p. 791–812, 1958.
- DARWIN, C. On the origins of species by means of natural selection. **London: Murray**, p. 247, 1859.
- DENEUBOURG, J.-L. et al. The self-organizing exploratory pattern of the argentine ant. **Journal of insect behavior**, Springer, v. 3, n. 2, p. 159–168, 1990.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, v. 1, n. 1, p. 269–271, 1959. ISSN 0945-3245.
- DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. **IEEE Computational Intelligence Magazine**, IEEE, v. 1, n. 4, p. 28–39, 2006.
- DORIGO, M.; GAMBARDILLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. **Evolutionary Computation, IEEE Transactions on**, IEEE, v. 1, n. 1, p. 53–66, 1997.
- DUCHESNE-GUILLEMIN, M. A hurrian musical score from ugarit: The discovery of mesopotamian music, sane2/2. **Malibu: Undena I**, v. 984, 1984.
- FISHER, R. A. **Statistical methods for research workers**. [S.l.]: Genesis Publishing Pvt Ltd, 1925.
- FLOOD, M. M. The traveling-salesman problem. **Operations Research, INFORMS**, v. 4, n. 1, p. 61–75, 1956.
- GEIS, M.; MIDDENDORF, M. An ant colony optimizer for melody creation with baroque harmony. In: IEEE. **IEEE Congress on Evolutionary Computation**. [S.l.], 2007. p. 461–468.

- GOIS, M. M. et al. Evolutionary multi-criterion optimization: 7th international conference, emo 2013, sheffield, uk, march 19-22, 2013. proceedings. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. cap. Multi-Objective Evolutionary Algorithm with Node-Depth Encoding and Strength Pareto for Service Restoration in Large-Scale Distribution Systems, p. 771–786. ISBN 978-3-642-37140-0.
- GOOD, M. Musicxml for notation and analysis. **The virtual score: representation, retrieval, restoration**, MIT Press, Cambridge, MA, v. 12, p. 113–124, 2001.
- GUÉRET, C.; MONMARCHÉ, N.; SLIMANE, M. Ants can play music. In: **Ant Colony Optimization and Swarm Intelligence**. [S.l.]: Springer, 2004. p. 310–317.
- HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **Systems Science and Cybernetics, IEEE Transactions on**, IEEE, v. 4, n. 2, p. 100–107, 1968.
- HEIJINK, H.; MEULENBROEK, R. G. On the complexity of classical guitar playing: functional adaptations to task constraints. **Journal of motor behavior**, Taylor & Francis, v. 34, n. 4, p. 339–351, 2002.
- HELGAUN, K. An effective implementation of the lin–kernighan traveling salesman heuristic. **European Journal of Operational Research**, Elsevier, v. 126, n. 1, p. 106–130, 2000.
- HEYES-JONES, J. **A* algorithm tutorial**. 2013. Disponível em: <<http://heyes-jones.com/pseudocode.php>>. Acesso em: 08/08/2014.
- HOLLAND, J. H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, 1975.
- KILMER, A. D.; CIVIL, M. Old babylonian musical instructions relating to hymnody. **Journal of Cuneiform Studies**, JSTOR, p. 94–98, 1986.
- KUNCHEVA, L. I.; RODRÍGUEZ, J. J. An experimental study on rotation forest ensembles. In: **Proceedings of the 7th International Conference on Multiple Classifier Systems**. Berlin, Heidelberg: Springer-Verlag, 2007. p. 459–468. ISBN 978-3-540-72481-0.
- LACERDA, E. G. de; CARVALHO, A. de. Introdução aos algoritmos genéticos. **Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais**, v. 1, p. 99–148, 1999.
- LI, C. et al. Subpopulation diversity based accepting immigrant in distributed evolutionary algorithms. In: IEEE. **International Conference on Parallel and Distributed Systems**. [S.l.], 2013. p. 422–423.
- LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal, The**, Alcatel-Lucent, v. 44, n. 10, p. 2245–2269, 1965.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations research**, INFORMS, v. 21, n. 2, p. 498–516, 1973.
- LINDEN, R. **Algoritmos genéticos**. 2. ed. [S.l.]: Brasport, 2008.
- MAIER, H. R. et al. Ant colony optimization for design of water distribution systems. **Journal of water resources planning and management**, American Society of Civil Engineers, v. 129, n. 3, p. 200–209, 2003.

MED, B. Teoria da música. 4ª edição revista e ampliada. **Brasília-DF, Musimed**, 1996.

OSABA, E. et al. A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems. **International Journal of Artificial Intelligence**, v. 13, n. 2, p. 46–56, 2015.

OTTEN, J. The catholic encyclopedia: An international work of reference on the constitution, doctrine, discipline and history of the catholic church. In: HERBERMANN, C. G. et al. (Ed.). **Guido of Arezzo**. [S.l.]: New York, Robert Appleton Company, 1910. VII, p. 65–66.

PRADO, R. S. **Restauração de Sistemas de Distribuição de Energia Elétrica Utilizando Evolução Diferencial com Arvore de Ancestralidade**. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2013.

PRADO, R. S. et al. Using differential evolution for combinatorial optimization: A general approach. In: IEEE. **International Conference on Systems Man and Cybernetics (SMC)**. [S.l.], 2010. p. 11–18.

PRADO, R. S. et al. Uma nova abordagem para a evolução diferencial em otimização discreta. **XVIII Congresso Brasileiro de Automática, Mato Grosso do Sul, Bonito**, 2010.

PRADO, R. S. et al. A new differential evolution based metaheuristic for discrete optimization. **Nature-Inspired Computing Design, Development, and Applications**, IGI Global, p. 104, 2012.

PRADO, R. S. et al. Differential evolution using ancestor tree for service restoration in power distribution systems. **Applied Soft Computing**, Elsevier, v. 23, p. 498–508, 2014.

PRICE, K.; STORN, R. M.; LAMPINEN, J. A. **Differential evolution: a practical approach to global optimization**. [S.l.]: Springer Science & Business Media, 2006.

PRICE, K. V. Differential evolution vs. the functions of the 2nd icoe. In: IEEE. **International Conference on Evolutionary Computation**. [S.l.], 1997. p. 153–157.

RADICIONI, D.; ANSELMA, L.; LOMBARDO, V. A segmentation-based prototype to compute string instruments fingering. In: CITESEER. **Proceedings of the Conference on Interdisciplinary Musicology**. [S.l.], 2004. v. 17, p. 97.

RADICIONI, D.; SCIENZA, C. D.; LOMBARDO, V. Guitar fingering for music performance. In: **International Computer Music Association**. [S.l.: s.n.], 2005. p. 527–530.

RADICIONI, D. P. **Computational Modeling of Fingering in Music Performance**. Tese (Doutorado) — Centro di Scienza Cognitiva, Università degli Studi di Torino, Torino, Italy, 2006.

RAMOS, J. V. et al. Avaliação entre diferentes abordagens evolutivas para o processo de transcrição automática de tablaturas. In: SBPO. **XLVII Simpósio Brasileiro em Pesquisa Operacional**. [S.l.], 2015.

RAMOS, J. V. et al. Comparative study of genetic algorithm and ant colony optimization algorithm performances for the task of guitar tablature transcription. In: BRACIS. **4th Brazilian Conference on Intelligent Systems**. [S.l.], 2015.

RIBEIRO, H. L. Papel da música na educação segundo platão. MUS 504 –Estudos Especiais. Julho 2001.

RUSSELL, S.; NORVIG, P.; SOUZA, V. D. de. **Inteligência artificial: tradução da segunda edição.** [S.l.]: Elsevier, 2004.

RUTHERFORD, N. T. Fingar, a genetic algorithm approach to producing playable guitar tablature with fingering instructions. **Undergraduate project dissertation, Dept. of Computer Sci., Univ. of Sheffield**, v. 15, 2009.

SANCHES, D. S. **Algoritmos Evolutivos Multi-Objetivo para Reconfiguração de Redes em Sistemas de Distribuição de Energia Elétrica.** Tese (Doutorado) — Universidade de Sao Paulo, 2013.

SANCHES, D. S. et al. Integrating several subpopulation tables with node-depth encoding and strength pareto for service restoration in large-scale distribution systems. In: IEEE. **Power and Energy Society General Meeting (PES)**. [S.l.], 2013. p. 1–5.

SANCHES, D. S. et al. Multiobjective evolutionary algorithm with a discrete differential mutation operator developed for service restoration in distribution systems. **International Journal of Electrical Power & Energy Systems**, Elsevier, v. 62, p. 700–711, 2014.

SANTOS, A. et al. Node-depth encoding and multiobjective evolutionary algorithm applied to large-scale distribution system reconfiguration. **IEEE Transactions on Power Systems**, IEEE, v. 25, n. 3, p. 1254–1265, 2010.

SAYEGH, S. I. Fingering for string instruments with the optimum path paradigm. **Computer Music Journal**, The MIT Press, v. 13, n. 3, p. 76–84, 1989. ISSN 01489267, 15315169.

SENGPIEL, E. **Tontechnik-Rechner.** 2014. Disponível em: <<http://www.sengpielaudio.com/calculator-notenames.htm>>. Acesso em: 29/08/2014.

STORN, R. On the usage of differential evolution for function optimization. In: NAFIPS. **Biennial Conference of the North American**. [S.l.], 1996. p. 519–523.

STORN, R.; PRICE, K. **Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces.** [S.l.]: ICSI Berkeley, 1995.

STORN, R.; PRICE, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, v. 11, n. 4, p. 341–359, 1997.

TUKEY, J. W. Comparing individual means in the analysis of variance. **Biometrics**, JSTOR, p. 99–114, 1949.

TUOHY, D.; POTTER, W. A genetic algorithm for the automatic generation of playable guitar tablature. In: SN. **Proceedings of the International Computer Music Conference.** [S.l.], 2005. p. 499–502.

TUOHY, D. R.; POTTER, W. Guitar tablature creation with neural networks and distributed genetic search. In: **Proceedings of the 19th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.** [S.l.: s.n.], 2006. v. 6, p. 27–30.

TUOHY, D. R.; POTTER, W.; CENTER, A. I. An evolved neural network/hc hybrid for tablature creation in ga-based guitar arranging. In: **Proceedings of the International Computer Music Conference**. [S.l.: s.n.], 2006.

TUOHY, D. R.; POTTER, W. D. Ga-based music arranging for guitar. In: IEEE. **IEEE Congress on Evolutionary Computation**. [S.l.], 2006. p. 1065–1070.

WALSHAW, C. A Statistical Analysis of the ABC Music Notation Corpus: Exploring Duplication. In: HOLZAPFEL, A. (Ed.). **Proc. 4th Intl Workshop on Folk Music Analysis**. [S.l.]: Boğaziçi University, 2014. p. 2–9. DOI 10.13140/2.1.4340.0961.

WHITLEY, D.; STARKWEATHER, T. Genitor ii: A distributed genetic algorithm. **Journal of Experimental & Theoretical Artificial Intelligence**, Taylor & Francis, v. 2, n. 3, p. 189–214, 1990.

YU, X.; GEN, M. **Introduction to evolutionary algorithms**. [S.l.]: Springer Science & Business Media, 2010.

ZHAO, F. et al. A novel genetic algorithm based on immunity and its application. In: IEEE. **Proc. of the 12th International Conference on Hybrid Intelligent System**. [S.l.], 2012. p. 566–571.

ZITZLER, E. et al. **SPEA2: Improving the strength Pareto evolutionary algorithm**. [S.l.]: Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) Zürich, Switzerland, 2001.

APÊNDICE A – RELAÇÃO DE MÚSICAS UTILIZADAS

Tabela 13: Músicas utilizadas para exploração do problema.

Instância	Música	Notas	Grupo	Ótimo
1	What You Will	45	1	66
2	La Duenna	59	1	73
3	Mary Ann Pierson's	63	1	90
4	Water, The	69	1	100
5	Middle Of May, The	73	1	105
6	Katie's Lilt	79	1	103
7	Behind The Bush In The Garden	85	1	122
8	New Found Out, The	89	1	131
9	Lynch's	95	1	122
10	St. Kilda Wedding	95	1	135
11	Farewell To Whalley Range	96	1	153
12	Shifting Peat And Feet	96	1	113
13	When Wars Alarmes	96	1	115
14	Bonnie Scotland	97	1	146
15	Mediator, The	98	1	141
16	La Petite Suisse	153	2	184
17	Jimmy Duffy's	161	2	205
18	Lily's March	161	2	162
19	Gallow's Hill	165	2	217
20	Stride, The	165	2	151
21	Wooden Shutters, The	165	2	247
22	Beardless Boy, The	166	2	146
23	Da Norna	166	2	233
24	Man In The Bog	167	2	229
25	Light And Airy	168	2	230
26	Empty Pod, The	174	2	223
27	Lumsden's Rant	174	2	241
28	Over The Drumlins	175	2	250
29	Jackson's	178	2	228
30	Smiling Bride, The	178	2	214
31	Aoife's	181	2	262
32	Loon And His Quine, The	181	2	246
33	House On The Hill, The	182	2	211
34	Tubber Fair	182	2	261
35	Muireann's	185	2	281

Tabela 13: Músicas utilizadas para exploração do problema. (continuação)

Instância	Música	Notas	Grupo	Ótimo
36	Father Kelly's	186	2	237
37	New Road, The	187	2	265
38	Coyle's Piano 1	191	2	253
39	Nancy	194	2	283
40	Maya's	196	2	275
41	Robbery Of Chester Jail, The	203	3	247
42	Ownie Davis'	204	3	299
43	Powers Of Punch, The	206	3	274
44	Tommy Gunn's	206	3	267
45	Ash City	212	3	192
46	Mademoiselle Kennedy	212	3	287
47	Redeside, The	216	3	297
48	Requiem Pour Le Pendu	216	3	268
49	Billy's	235	3	320
50	Floating Candles	235	3	319
51	Troy's Wedding	240	3	279
52	Earthsongs	241	3	331
53	Lemascally Mill	245	3	362
54	Gan Ainm	248	3	235
55	Man From Skye, The	248	3	318
56	Fifty Five	279	3	351
57	Lady Bermingham	281	3	368
58	Seamus Connolly's	281	3	430
59	Old Gray Goose, The	282	3	346
60	Maltman Comes On Monday, The	283	3	364
61	Mrs MacPherson Of Inveran	283	3	310
62	High Road To Lyon, The	284	3	415
63	Bucks Of Oranmore, The	288	3	406
64	Iain MacDonald's	291	3	338
65	Heartland Part 2	293	3	234
66	Panda, The	293	3	425
67	Ballintore, The	297	3	401
68	Groves, The	298	3	415
69	Waltz Of The White Lilies	298	3	301
70	Strayaway Child, The	299	3	334
71	Manali Beetle, The	300	3	418
72	Mason's Apron, The	300	3	402
73	Pressed For Time	300	3	320
74	A Visit To Ireland	304	4	391
75	Piper In The Cave, The	304	4	431
76	Liz's	307	4	418
77	Cape Breton Fiddlers' Society's, The	311	4	392
78	Mouseskin Shoe, The	313	4	435
79	Wissahickon Drive	314	4	497
80	Long Note, The	315	4	398
81	Billy Nicholson	317	4	461

Tabela 13: Músicas utilizadas para exploração do problema. (continuação)

Instância	Música	Notas	Grupo	Ótimo
82	Flick It Up And Catch It	330	4	503
83	Tara Lynn's March To God	336	4	398
84	Jewel Of A Hundred Years	338	4	453
85	Suleman's Kopanitsa	379	4	413
86	Andy Renwick's Ferret	402	4	478
87	Bridge Attack	475	4	487

Fonte: Autoria própria.

APÊNDICE B – TABELAS COM O RESULTADO DOS EXPERIMENTOS

Tabela 14: Resultados de *fitness* dos experimentos do grupo 1.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
What You Will	45	72,84 ± 2,46	66,9 ± 0,96	71,8 ± 2,44	67,43 ± 0,9	79,4 ± 3,85	78,37 ± 3,89	67 ± 1,2
La Duenna	59	81,35 ± 1,36	73,4 ± 0,5	87,13 ± 4,71	75,37 ± 1,61	93,37 ± 5,13	86,7 ± 4,67	78,87 ± 1,48
Mary Ann Pierson's	63	102,1 ± 2,22	90,83 ± 1,12	102,33 ± 2,4	91,17 ± 0,65	98,33 ± 2,45	94,1 ± 1,99	90,93 ± 1,31
Water, The	69	115,8 ± 1,99	100,67 ± 0,76	118,37 ± 3,77	102,67 ± 2,29	114,57 ± 4,69	107,43 ± 4,39	100,97 ± 1,56
Middle Of May, The	73	121,43 ± 1,71	108,57 ± 1,28	126,2 ± 4,43	107,7 ± 1,15	121,77 ± 4,25	113,73 ± 2,77	107,63 ± 1,27
Katie's Lilt	79	122,21 ± 3,5	104,77 ± 0,77	135,37 ± 4,21	108,07 ± 2,02	123,63 ± 4,82	119,77 ± 4,25	109,23 ± 2,1
Behind The Bush In The Garden	85	136,64 ± 1,45	129,13 ± 0,68	151,37 ± 3,91	128,93 ± 3,16	128,57 ± 2,79	126,97 ± 3,08	132,6 ± 2,11
New Found Out, The	89	155,82 ± 2,06	134,7 ± 1,26	162,83 ± 4,84	135,63 ± 2,31	140,7 ± 3,78	136,37 ± 3,36	135,13 ± 1,5
Lynch's	95	139,6 ± 1,23	126,83 ± 0,65	164,57 ± 4,33	127,87 ± 2,26	130,37 ± 3,46	129,33 ± 2,9	131,1 ± 1,86
St. Kilda Wedding	95	161,4 ± 4,3	143,67 ± 1,52	172,37 ± 3,48	140,37 ± 2,5	143,77 ± 2,4	140,97 ± 2,46	138,73 ± 1,46
Farewell To Whalley Range	96	176,04 ± 4,76	162,4 ± 2,53	187 ± 4,73	156,83 ± 2,46	159,07 ± 3,05	155,73 ± 1,95	156,07 ± 1,57
Shifting Peat And Feet	96	130,7 ± 0,71	126,77 ± 1,91	160,67 ± 5,39	124,87 ± 4,7	126,97 ± 6,13	122,93 ± 3,68	126,4 ± 2,54
When Wars Alarmes	96	140,35 ± 1,2	123,33 ± 1,75	161,23 ± 6,01	126,87 ± 4,06	138,8 ± 6,86	133,23 ± 6,06	130,2 ± 3,25
Bonnie Scotland	97	161,95 ± 2,54	153,1 ± 1,84	185,37 ± 4,06	152,13 ± 2,47	156,7 ± 4	153,53 ± 2,76	151,3 ± 1,47
Mediator, The	98	154,03 ± 3,16	146,03 ± 1,03	184,37 ± 5,46	150,1 ± 3,1	158,2 ± 3,73	151 ± 3,99	150,23 ± 1,55

Fonte: Autoria própria.

Tabela 15: Resultados do tempo (ms) dos experimentos do grupo 1.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
What You Will	45	0 ± 0	1779,97 ± 212,8	3691,5 ± 906,64	11767,67 ± 3262	284,43 ± 47,31	998,4 ± 475,02	17692,07 ± 4550,08
La Duenna	59	0 ± 0	1675,43 ± 242,42	5991,5 ± 2802,81	18983,77 ± 6180,7	382,73 ± 70,79	1288,57 ± 452,32	24037,67 ± 5394,94
Mary Ann Pierson's	63	0 ± 0	1108,13 ± 153,08	6161 ± 1805,14	16111,3 ± 4864,81	385,33 ± 76,19	1228,23 ± 471,84	26950,23 ± 6815,38
Water, The	69	0 ± 0	2112,77 ± 254,09	6758,5 ± 2501,11	15864,27 ± 2750,68	499,2 ± 143,49	1762,3 ± 1106,16	32679,6 ± 10874,94
Middle Of May, The	73	0 ± 0	1796,13 ± 341,53	6359,13 ± 2066,83	18833,5 ± 4392,9	511,67 ± 157,61	1821,03 ± 1078,47	34565,67 ± 10951,22
Katie's Lilt	79	0 ± 0	2633,8 ± 418,48	8835,4 ± 2794,66	23555,63 ± 7805,17	516,37 ± 144,34	1701,97 ± 892,39	42285,1 ± 12654,54
Behind The Bush In The Garden	85	0 ± 0	1961,97 ± 195,03	9420,53 ± 4065,29	25151,53 ± 8118,08	605,8 ± 209,77	1940,67 ± 992,28	39165,07 ± 14266,11
New Found Out, The	89	0 ± 0	2648,9 ± 324,17	9192,63 ± 2930,07	25144,77 ± 7296,69	653,63 ± 215,87	2404,5 ± 1631,78	49111,73 ± 14771,99
Lynch's	95	0 ± 0	1438,33 ± 255,07	10635,63 ± 2774,43	34345,2 ± 12501,7	592,3 ± 134,54	2256,27 ± 1449,23	44999 ± 13857,94
St. Kilda Wedding	95	0 ± 0	2292,17 ± 440,99	10607,53 ± 2977,85	28083,6 ± 9283,9	594,37 ± 160,73	2140,33 ± 1123,71	42691,87 ± 15316,17
Farewell To Whalley Range	96	0 ± 0	2870,93 ± 444,97	10243,53 ± 3829,69	29974,53 ± 9921,1	587,63 ± 95,29	1856,43 ± 406,07	51832,4 ± 17223,2
Shifting Peat And Feet	96	0 ± 0	1006,73 ± 128,83	9819,23 ± 2895,86	29598,63 ± 10633,85	609,97 ± 166,05	2073,77 ± 1091,25	52825,07 ± 21760,96
When Wars Alarmes	96	4 ± 8	1792,43 ± 337,84	11083,37 ± 4221,5	33438,8 ± 11071,2	586,53 ± 107,25	2316,6 ± 1496,99	53612,83 ± 16787,14
Bonnie Scotland	97	0 ± 0	2705,6 ± 451,84	10644,47 ± 4041,97	27873,23 ± 10128,89	632,87 ± 204,5	1924,03 ± 585,72	42684,47 ± 12548,99
Mediator, The	98	3 ± 6,71	2258,37 ± 270,77	10942,93 ± 4320,05	72876,37 ± 209229,49	625,03 ± 186,76	2071,7 ± 668,25	48465,07 ± 16219,93

Fonte: Autoria própria.

Tabela 16: Resultados de *fitness* dos experimentos do grupo 2.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
La Petite Suisse	153	203,07 ± 2,84	186,83 ± 1,68	292,17 ± 7,82	207,43 ± 5,06	201,23 ± 6,19	199,23 ± 4,98	210,77 ± 4,12
Jimmy Duffy's	161	241,52 ± 2,21	211,17 ± 2,21	321,17 ± 7,63	230,67 ± 5,44	216,23 ± 3,9	214,3 ± 3,94	240,2 ± 4,34
Lily's March	161	191,34 ± 2,31	171,1 ± 1,16	286 ± 8,2	189,63 ± 8,17	177,23 ± 5,46	172,4 ± 4,3	195,3 ± 4,73
Gallow's Hill	165	263,46 ± 2	229,13 ± 1,38	333,93 ± 7,87	237,23 ± 5,43	230,17 ± 5,46	224,6 ± 4,11	238,5 ± 3,68
Stride, The	165	170,11 ± 1,58	152,07 ± 0,52	253,53 ± 6,31	172,8 ± 8,13	156,07 ± 3,36	154,17 ± 4,93	221,37 ± 8,99
Wooden Shutters, The	165	281 ± 1,53	257,33 ± 2,02	349,13 ± 7,75	270,33 ± 5,65	260,73 ± 5,74	256,13 ± 3,5	267,93 ± 3,4
Beardless Boy, The	166	162,16 ± 1,71	156,8 ± 1,13	282,5 ± 11,12	177,17 ± 7,84	175,43 ± 7,07	169,33 ± 6,43	191,37 ± 6,48
Da Norna	166	288,98 ± 1,71	261,8 ± 2,43	354,3 ± 10,56	263,43 ± 7,13	251,1 ± 6,15	246,93 ± 5,46	269,2 ± 4,37
Man In The Bog	167	294,58 ± 2,31	260,5 ± 2,61	329,47 ± 7,88	247,6 ± 5,82	250 ± 7,63	241,17 ± 5,65	256,67 ± 3,62
Light And Airy	168	274,59 ± 7,11	239,93 ± 2,02	359,67 ± 5,89	252,23 ± 8,03	243,33 ± 3,39	238,33 ± 3,19	255,6 ± 3,51
Empty Pod, The	174	279,55 ± 1,29	257,47 ± 2,15	352,2 ± 8,1	243,03 ± 5,47	243,27 ± 6,17	235,73 ± 4,27	247,67 ± 3,42
Lumsden's Rant	174	282,91 ± 3,5	248,2 ± 2,61	357,2 ± 9,76	251,13 ± 3,89	254,5 ± 3,91	248,13 ± 3,05	250,2 ± 2,71
Over The Drumlins	175	307,15 ± 3,56	273,3 ± 2,15	373,63 ± 9,08	286,47 ± 8,32	280,6 ± 8,85	273,07 ± 9,41	292,77 ± 5,08
Jackson's	178	258,53 ± 1,94	240,6 ± 2,19	352,87 ± 10,01	247,1 ± 6,1	249 ± 6,19	240,1 ± 4,4	242,5 ± 3,75
Smiling Bride, The	178	248,35 ± 2,71	226,83 ± 1,42	357,17 ± 9,66	251,5 ± 7,4	232,77 ± 8,22	221,93 ± 6,24	263,57 ± 4,97
Aoife's	181	341,93 ± 2,26	316,4 ± 3,2	395,03 ± 7,07	289,1 ± 6,43	286,73 ± 7,62	277,17 ± 5,81	296,3 ± 4,26
Loon And His Quine, The	181	270,8 ± 1,33	251,47 ± 1,59	365,6 ± 8,28	265 ± 4,64	262,93 ± 4,23	257,1 ± 3,43	264,2 ± 3,58
House On The Hill, The	182	258 ± 1,67	232,87 ± 1,83	351,6 ± 10,96	237,4 ± 7,81	229,17 ± 6,1	224,67 ± 5,77	244,43 ± 5,72
Tubber Fair	182	301,67 ± 1,49	277,1 ± 2,25	380,1 ± 9,91	277,7 ± 4,55	276,9 ± 5,39	273,1 ± 3,21	284,13 ± 5,12
Muireann's	185	319,3 ± 1,2	290,87 ± 2,34	397,43 ± 7,17	303,03 ± 5,57	296,43 ± 6,93	290,17 ± 5,53	299,97 ± 2,74
Father Kelly's	186	293,41 ± 3,35	250,47 ± 1,78	390,37 ± 8,22	274,33 ± 7,37	254,1 ± 6,1	249,73 ± 4,56	274 ± 5,21
New Road, The	187	316,51 ± 2,17	287,93 ± 1,44	401,13 ± 9,27	287,93 ± 4,67	283,27 ± 6,85	278,13 ± 4,66	289,87 ± 3,85
Coyle's Piano 1	191	284,93 ± 3,78	262,87 ± 2,85	401,17 ± 8,43	276,57 ± 6,7	277,3 ± 7,5	266,67 ± 4,38	284,63 ± 6,07
Nancy	194	335,42 ± 1,07	304,1 ± 1,84	428,57 ± 7,98	306,93 ± 5,84	308,57 ± 6,65	298,9 ± 4,32	306,77 ± 4,06
Maya's	196	337,71 ± 3,64	322,27 ± 4,13	429,03 ± 10,29	303,3 ± 5,12	301,73 ± 7,93	292,4 ± 6,04	305,93 ± 5,25

Fonte: Autoria própria.

Tabela 17: Resultados do tempo (ms) dos experimentos do grupo 2.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
La Petite Suisse	153	12,6 ± 7,06	3617,13 ± 354,64	19463,43 ± 6926,65	52440,77 ± 15821,86	1312 ± 444,79	4514,67 ± 2047,42	112464,83 ± 47227,61
Jimmy Duffy's	161	11,75 ± 14,89	2882,93 ± 388,39	22618,6 ± 10377,52	71114,9 ± 26871,89	1354,57 ± 314,19	5153,77 ± 2484,93	101257,13 ± 33720,55
Lily's March	161	11,5 ± 7,68	2622,37 ± 304,99	20514,67 ± 7647,93	60088,03 ± 19137,55	1244,9 ± 259,61	4201,6 ± 1237,34	122181,53 ± 50259,68
Gallow's Hill	165	9,4 ± 8,59	3767,43 ± 425,2	20606,7 ± 9464,42	70898,8 ± 22929,24	1602,13 ± 510,98	4820,97 ± 1799,34	106996,93 ± 36258,06
Stride, The	165	9,4 ± 8,59	4339,43 ± 479,24	21464,7 ± 8275,5	129611,93 ± 206568,58	1698,83 ± 412,97	5285,83 ± 1827,06	176540,1 ± 70401,57
Wooden Shutters, The	165	3 ± 6,71	4032,63 ± 628,44	20762,67 ± 9520,56	69012,77 ± 30441,81	1684,3 ± 568,22	5566,63 ± 3209,63	128634,27 ± 44294,93
Beardless Boy, The	166	9,4 ± 8,59	3613,5 ± 532,81	22085,07 ± 5906,08	82782,73 ± 21392,57	1656,73 ± 452,68	5098,1 ± 1884,22	124294,7 ± 45541,94
Da Norna	166	0 ± 0	3393,57 ± 515,7	22405,9 ± 9327,82	71708,47 ± 25666,68	1756,57 ± 498,06	5557,3 ± 2376,49	122548,13 ± 39566,1
Man In The Bog	167	9,2 ± 8,41	3650,43 ± 584,24	24460,43 ± 7580,93	78279,73 ± 21404,47	1648,93 ± 554,9	5291,53 ± 2616,03	125118,57 ± 50405,29
Light And Airy	168	3,2 ± 7,16	4333,7 ± 498,08	20147,03 ± 6995,5	76158,63 ± 23065,87	1645,8 ± 521,99	5297,3 ± 1628,23	111499,13 ± 44610,9
Empty Pod, The	174	4 ± 8	3272,37 ± 713,23	19458 ± 6986,98	73821,23 ± 19442,99	1637 ± 492,22	5695,07 ± 4246,88	125131 ± 42588,56
Lumsden's Rant	174	5 ± 8,66	2548,53 ± 513,43	22795,37 ± 8937,3	65548 ± 16750,84	1789,83 ± 624,79	5343,57 ± 1732,77	124129,47 ± 46404,6
Over The Drumlins	175	9,4 ± 8,59	4437,2 ± 572,77	27316,3 ± 9935,62	125061,87 ± 171564,52	2045,17 ± 654,71	6793,33 ± 3854,67	112323,83 ± 56036,28
Jackson's	178	15,67 ± 27,14	2278,13 ± 509,64	23916 ± 7053,86	64164,27 ± 20289,41	1738,37 ± 460,88	5819,9 ± 2411,13	116728,8 ± 42515,43
Smiling Bride, The	178	7,83 ± 8,59	5692,47 ± 879,94	26716,73 ± 10645,32	75864,87 ± 18773	2030,1 ± 504,56	7006 ± 1643,29	125334,7 ± 46406,26
Aoife's	181	6,2 ± 8,5	4140,77 ± 731,37	27892,97 ± 9605,41	77554,33 ± 28066,92	1811,17 ± 761,17	7848,93 ± 3057,38	128136,27 ± 45930,42
Loon And His Quine, The	181	3,75 ± 7,5	3584,4 ± 491,48	24066,8 ± 8092,21	77529,9 ± 31505,96	1595,9 ± 477,32	5161,03 ± 2077,75	126420,77 ± 39481,49
House On The Hill, The	182	6,4 ± 8,76	3699,83 ± 521,04	25887,33 ± 10734,43	86550,9 ± 26195,5	2198,6 ± 692,52	7050,73 ± 3648,31	142781,43 ± 56918,67
Tubber Fair	182	7,75 ± 8,96	3510 ± 496,1	27467,07 ± 10458,65	121700,97 ± 207045,9	2001,47 ± 803,65	5499,53 ± 1773,97	167495,83 ± 74204,53
Muireann's	185	3,75 ± 7,5	3497,03 ± 458,13	25803,1 ± 11680,29	66000,4 ± 19191,36	2104,97 ± 733,56	5774,63 ± 4061,32	118743,3 ± 40868,56
Father Kelly's	186	6,4 ± 8,76	4483,47 ± 495,4	24502,57 ± 8455,44	85396,2 ± 28177,39	2153,87 ± 555,91	7870,77 ± 4531,75	144474,9 ± 53890,95
New Road, The	187	3 ± 6,71	4830,3 ± 677,56	24788,5 ± 10401,14	82709,67 ± 20565	1870,47 ± 701,57	6251,47 ± 2419,82	156510,1 ± 45772,48
Coyle's Piano 1	191	6,2 ± 8,5	4770 ± 787,22	25986,33 ± 11477,02	87507,2 ± 24698,34	2034,77 ± 915,68	6669,57 ± 2269,82	170910,53 ± 88247,59
Nancy	194	6,4 ± 8,76	3915,1 ± 430,42	26544,4 ± 8232,98	133894,43 ± 163141,55	2208,47 ± 680,11	6638,37 ± 2959,12	143669,63 ± 53640,5
Maya's	196	6,2 ± 8,5	3504,83 ± 640,87	25557,13 ± 10883,35	83185,97 ± 25405,39	2339,5 ± 924,01	7591 ± 3064,4	130846,37 ± 51948,63

Fonte: Autoria própria.

Tabela 18: Resultados de *fitness* dos experimentos do grupo 3.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
Robbery Of Chester Jail, The	203	282,83 ± 1,89	264,5 ± 2,45	419,37 ± 10,7	289,17 ± 7,75	274,8 ± 7,47	270,3 ± 7,98	294,93 ± 5,08
Owne Davis'	204	342,19 ± 1,49	322,63 ± 1,59	452,23 ± 8,94	324,83 ± 6,51	336,6 ± 6,23	320,23 ± 6,69	326,23 ± 5,83
Powers Of Punch, The	206	322,73 ± 1,69	303 ± 2,29	441,27 ± 10	306,23 ± 6,91	314,27 ± 9,24	299,47 ± 5,28	317,77 ± 4,31
Tommy Gunn's	206	335,96 ± 1,58	295,03 ± 2,41	418,83 ± 7,97	299,33 ± 9,43	289,73 ± 6,28	281,23 ± 4,41	310,67 ± 6,56
Ash City	212	240,69 ± 5,52	203,7 ± 3,26	392,53 ± 13,72	233,2 ± 10,82	221,43 ± 7,74	213,1 ± 7,43	234,3 ± 6,2
Mademoiselle Kennedy	212	349,18 ± 1,98	309,67 ± 2,37	447,23 ± 9,61	323,5 ± 9,18	310,5 ± 6,55	304,47 ± 7,62	343,87 ± 6,24
Redeside, The	216	392,17 ± 0,82	347,47 ± 3,25	477,3 ± 9,85	339,13 ± 8,43	334,7 ± 10,3	323,63 ± 7,6	344,93 ± 5,01
Requiem Pour Le Pendu	216	330,63 ± 4,16	295,53 ± 3,48	441,67 ± 7,29	305,5 ± 9,02	304,03 ± 7,73	293,7 ± 6,41	320,4 ± 5,56
Billy's	235	363,1 ± 0,82	351,77 ± 2,16	508,03 ± 11,21	345,8 ± 5,63	356,33 ± 5,95	338 ± 5,45	350,1 ± 5,72
Floating Candles	235	344,95 ± 0,84	330,6 ± 3,21	503,5 ± 11,36	347,3 ± 7,48	340,47 ± 8,37	332,13 ± 4,71	347,2 ± 3,7
Troy's Wedding	240	330,91 ± 6	285,9 ± 1,09	510,13 ± 8,83	315,47 ± 8,46	318,07 ± 7,24	301,27 ± 5,94	317,17 ± 5,59
Earthsongs	241	391,04 ± 1,44	363,47 ± 2,24	529,17 ± 10,02	364 ± 7,39	367,03 ± 7,81	356,57 ± 7,08	369,03 ± 5,18
Lemascallly Mill	245	419,81 ± 0,79	381,97 ± 2,75	561,43 ± 9,51	399,43 ± 6,93	398,1 ± 8,57	383,73 ± 7,5	395,93 ± 4,84
Gan Ainm	248	296,09 ± 1,58	251,3 ± 2,48	453 ± 10,81	277,93 ± 12,22	278,2 ± 11,65	260,9 ± 8,62	325,1 ± 10,91
Man From Skye, The	248	363,71 ± 2,12	328,37 ± 2,88	538,17 ± 10,23	351,97 ± 6,39	356,5 ± 9,75	341,13 ± 7,01	357,8 ± 5,12
Fifty Five	279	405,85 ± 6,98	382,43 ± 2,24	592,7 ± 10,11	400,2 ± 10,58	406,83 ± 12,68	380,97 ± 8,08	410,37 ± 8,9
Lady Bermingham	281	425,75 ± 4,16	409,33 ± 3,21	617,47 ± 15,18	414,3 ± 6,11	427,63 ± 9,22	402,07 ± 8,07	432,1 ± 7,97
Seamus Connolly's	281	515,16 ± 1,67	492,2 ± 1,92	688,43 ± 11,83	475,77 ± 8,74	483 ± 10,01	465,77 ± 7,22	488,27 ± 6,34
Old Gray Goose, The	282	408,06 ± 2,38	375,97 ± 3,1	606,93 ± 13,34	398,83 ± 9,11	406,43 ± 9,49	386,2 ± 12,31	405,77 ± 7,2
Maltman Comes On Monday, The	283	451,36 ± 1,8	406,23 ± 3,06	651 ± 11,89	421,43 ± 10,07	430,67 ± 10,52	411,33 ± 9,47	439,7 ± 7,96
Mrs MacPherson Of Inveran	283	371,29 ± 2,1	322,17 ± 3,41	593,87 ± 14,36	359,57 ± 9,33	375,7 ± 11,97	342,27 ± 7,74	362,33 ± 7,13
High Road To Lyon, The	284	499,42 ± 1,23	438,17 ± 2,88	653,07 ± 9,1	448,37 ± 9,22	468,1 ± 9,34	444,67 ± 7,79	454,2 ± 5,07
Bucks Of Oranmore, The	288	479,23 ± 2,54	430,7 ± 1,62	661,37 ± 12,01	449,93 ± 6,43	459,23 ± 10,22	441,37 ± 8,37	457,13 ± 5,26
Iain MacDonald's	291	414,31 ± 2,89	374 ± 1,89	609,07 ± 9,94	390,07 ± 10,86	399,17 ± 9,72	378,63 ± 12,24	416,87 ± 7,78
Heartland Part 2	293	270,2 ± 1,29	242,97 ± 1,97	542,4 ± 14,23	316,6 ± 12,9	311,17 ± 12,47	289,37 ± 12,81	369,17 ± 10,57
Panda, The	293	497,71 ± 0,29	441,23 ± 3,83	686,23 ± 11,64	462,3 ± 8,38	483,7 ± 9,8	453,67 ± 7,53	462,1 ± 6,83
Ballintore, The	297	475,69 ± 4,53	420,47 ± 3,3	679,2 ± 12,64	445,07 ± 8,05	463,27 ± 13,17	440,33 ± 12,36	463,67 ± 5,42
Groves, The	298	491,69 ± 1,77	445,03 ± 2,93	693,07 ± 12,45	461,8 ± 10,17	483,43 ± 9,82	455,07 ± 9,23	470,23 ± 5,34
Waltz Of The White Lilies	298	331,87 ± 1,58	320,6 ± 1,13	582,67 ± 11,27	355,37 ± 14,55	374,03 ± 14,72	335,67 ± 9,73	398,33 ± 13,94
Strayaway Child, The	299	381,43 ± 1,67	370,27 ± 2,64	604,3 ± 13,83	384,33 ± 11,29	401 ± 10,54	378,2 ± 11,89	421,13 ± 8,39
Manali Beetle, The	300	479,53 ± 2,44	427,93 ± 1,48	691,43 ± 8,92	454,67 ± 9,87	474,17 ± 8,7	444,8 ± 6,94	455,43 ± 6,3
Mason's Apron, The	300	495,36 ± 2,2	447,07 ± 5,55	677,3 ± 13,78	451,97 ± 7,97	473,1 ± 12,15	441,57 ± 9,71	448,8 ± 6,22
Pressed For Time	300	365,31 ± 3,32	324,57 ± 2,65	637,93 ± 12,13	373,83 ± 11,95	398,67 ± 12,23	356,57 ± 11,15	384,03 ± 10,47

Fonte: Autoria própria.

Tabela 19: Resultados do tempo (ms) dos experimentos do grupo 3.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
Robbery Of Chester Jail, The	203	7,75 ± 8,96	3420,57 ± 482,67	26542 ± 11261,91	93198,6 ± 30409,14	2603,13 ± 823,76	7121,97 ± 2814,27	152356,33 ± 57676,24
Owne Davis'	204	6,2 ± 8,5	4915,07 ± 858,63	26251,87 ± 11368,08	88019,43 ± 27806,99	2080,53 ± 721,65	8559,77 ± 4509,27	156343,17 ± 76178,19
Powers Of Punch, The	206	9,4 ± 8,59	4350,37 ± 559,61	29244,97 ± 9859,83	100431,33 ± 26370,12	2548,53 ± 815,06	7769,9 ± 3058,58	169032,1 ± 64616,77
Tommy Gunn's	206	9,4 ± 8,59	4642,6 ± 615,26	31992,7 ± 12533,84	149690,5 ± 225178,49	2692,57 ± 1085,14	9340,8 ± 3723,04	236138,17 ± 98545,58
Ash City	212	10,33 ± 8,96	2757,57 ± 393,68	27928,3 ± 8417,33	96092,47 ± 30321,25	2753,93 ± 803,19	8880,1 ± 4402,06	168028,17 ± 54274,6
Mademoiselle Kennedy	212	12,6 ± 7,06	5219,27 ± 692,13	32809,63 ± 10285,18	115038,23 ± 29674,26	2579,73 ± 648,58	10382,4 ± 5707,64	173211,03 ± 71416,77
Redeside, The	216	15,5 ± 0,58	3832,93 ± 624,7	29473,77 ± 10648,67	115051,27 ± 35998,79	3237,57 ± 1428,04	8360,1 ± 3142,74	154992,73 ± 62933,17
Requiem Pour Le Pendu	216	15,5 ± 12,66	4062,27 ± 702,93	33730,57 ± 14707,95	114938 ± 41015,19	2729,5 ± 1072,56	9459,9 ± 4450,02	185810,7 ± 73659,12
Billy's	235	11,75 ± 7,85	4348,27 ± 860,97	35078,37 ± 11264,66	113162,17 ± 28619,83	3213,1 ± 1134,96	9556,63 ± 3201,26	218564,87 ± 103248,99
Floating Candles	235	10,33 ± 8,96	3225,57 ± 487,57	39618 ± 12973,49	115118,43 ± 27366,66	3273,4 ± 1089,17	9600,83 ± 2973,5	189417,67 ± 80979,49
Troy's Wedding	240	12,6 ± 7,06	5329,5 ± 748,55	33770,57 ± 10660,4	116806,8 ± 29371,59	3223,5 ± 1014,94	10670,97 ± 3882,97	229390,1 ± 75142,19
Earthsongs	241	12,4 ± 6,95	5084,6 ± 630,32	40428,2 ± 19385,09	128009,2 ± 38516,42	3191,8 ± 790,29	10385 ± 4237,62	209904,53 ± 84656,4
Lemasically Mill	245	10,33 ± 8,96	2937,5 ± 405,59	31841,87 ± 11465,9	113752,3 ± 31838,8	3113,77 ± 1081,45	10698,53 ± 5272,87	184148,73 ± 74852,74
Gan Ainm	248	18,8 ± 6,83	5330,53 ± 649,03	42455,67 ± 14226,27	143764,8 ± 33442,25	3702,43 ± 1354,1	13046,87 ± 6070,89	310859,9 ± 134457,18
Man From Skye, The	248	19,5 ± 7,68	4971,2 ± 710,08	37202,6 ± 10666,96	119696,93 ± 24812,82	4038,33 ± 1519,25	10158,27 ± 3851,54	209753,23 ± 104886,59
Fifty Five	279	11,75 ± 7,85	4500,63 ± 404,2	45332,87 ± 16899,32	176536,47 ± 49404,11	4328,5 ± 1687,93	13325,1 ± 4362,12	277725,47 ± 129978,17
Lady Bermingham	281	15,6 ± 0,55	5675,87 ± 830,49	48282,8 ± 21614,27	171388,43 ± 47725,82	4620,23 ± 1496,88	18364,43 ± 5830,77	284796,93 ± 128706,14
Seamus Connolly's	281	12,4 ± 6,95	5398,67 ± 778,37	40226,97 ± 11297,89	165101,6 ± 47573,26	4389,33 ± 1364,6	16003,63 ± 5611,29	262411,37 ± 126746,59
Old Gray Goose, The	282	13 ± 6,39	7030,97 ± 1117,57	49699,8 ± 17116,47	160023,2 ± 46542,05	5125,17 ± 2007,49	16148,7 ± 5608,43	271103,77 ± 101735,87
Maltman Comes On Monday, The	283	15,6 ± 0,55	5982,63 ± 727,55	47081,1 ± 19342,86	186378,43 ± 44168,36	5118,9 ± 1536,66	16048,87 ± 6317,73	312166,97 ± 124894,33
Mrs MacPherson Of Inveran	283	10,33 ± 8,96	2997,83 ± 505,14	43089,57 ± 15476,65	147943 ± 35286,82	4061,73 ± 1163,43	14507,03 ± 4844,71	290641,33 ± 108288,24
High Road To Lyon, The	284	11,75 ± 7,85	5509,43 ± 733,51	44817,53 ± 16508,24	174461 ± 36631,7	4315,53 ± 1906,64	15893,4 ± 9058,05	253900,47 ± 91817,61
Bucks Of Oranmore, The	288	9,4 ± 8,59	6736,13 ± 738,75	45471,7 ± 17353,29	177755,87 ± 45398,92	5058,57 ± 2026,02	14109,23 ± 4936,57	248523,13 ± 123700,85
Iain MacDonald's	291	15,67 ± 0,52	8070,97 ± 963,92	47173,67 ± 16848,69	173398,23 ± 52421,08	5257,77 ± 1626,32	17475,23 ± 9103,88	309926,8 ± 121940,88
Heartland Part 2	293	19,5 ± 7,68	6512,53 ± 954,09	46927,8 ± 18225,95	221914,07 ± 55561,19	5522,97 ± 2212,48	20116,83 ± 7923,2	404265,17 ± 154869,14
Panda, The	293	10,33 ± 8,96	3868,83 ± 547,75	45051,87 ± 21650,18	151648,57 ± 36321,89	4478,27 ± 1518,26	17180,9 ± 8587,59	277503,27 ± 115444,81
Ballintore, The	297	15,6 ± 0,55	5922,83 ± 710,75	46292,8 ± 16279,32	176520,33 ± 49974,72	4640,5 ± 1673,27	16379,57 ± 7549,06	228802,5 ± 89207,42
Groves, The	298	12,4 ± 6,95	5258,8 ± 700,81	43436,9 ± 14641,7	184926,73 ± 57950,95	4989,43 ± 1767,07	14890,3 ± 4643,34	297630,7 ± 96625,24
Waltz Of The White Lilies	298	15,6 ± 0,55	6200 ± 537,46	51498,53 ± 17832,45	186798,17 ± 42995,76	4388,8 ± 1214,26	18509 ± 6773,58	394123,53 ± 196799,02
Strayaway Child, The	299	21,8 ± 8,41	6887,43 ± 742,9	50596,33 ± 17387,95	238547,67 ± 63302,45	5404,93 ± 2151,6	18393,57 ± 6688,94	326530,9 ± 168059,13
Manali Beetle, The	300	9,4 ± 8,59	7251,43 ± 851,75	50215,67 ± 15389,39	158374,67 ± 47654,48	4413,27 ± 1156,98	15130,53 ± 4081,74	288368,93 ± 145624,14
Mason's Apron, The	300	10,33 ± 8,96	2929,17 ± 423,9	51131,4 ± 17760,89	143075,8 ± 41479,13	4027,97 ± 1366,1	12871,63 ± 5855,61	313021,7 ± 141375,28
Pressed For Time	300	15,6 ± 0,55	7105,83 ± 1038,81	44653,23 ± 15454,56	162144,83 ± 52520,47	4684,2 ± 1412,44	16919,33 ± 6714,71	351720,4 ± 190096,63

Fonte: Autoria própria.

Tabela 20: Resultados de *fitness* dos experimentos do grupo 4.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
A Visit To Ireland	304	450,09 ± 2,84	402,27 ± 3,35	693,83 ± 14,01	448,5 ± 11,49	459,7 ± 9,08	429,83 ± 10,06	458,27 ± 8,64
Piper In The Cave, The	304	483,87 ± 2,16	446,7 ± 3,39	690,9 ± 11,83	465,37 ± 7,65	487,37 ± 9,6	456,87 ± 7,92	469,3 ± 5,86
Liz's	307	494,65 ± 0,75	434,43 ± 2,91	705,63 ± 9,82	456,17 ± 7,94	482,27 ± 8,82	455,6 ± 6,8	462 ± 5,19
Cape Breton Fiddlers' Society's, The	311	453,64 ± 2,3	410,73 ± 1,28	681,27 ± 11,39	437,83 ± 8,47	464,03 ± 10,99	426,83 ± 6,38	453 ± 8,63
Mouseskin Shoe, The	313	531,07 ± 3,61	502,9 ± 3,07	735,57 ± 14,37	487,5 ± 11,87	522,03 ± 12,12	478,5 ± 9,61	500,3 ± 5,72
Wissahickon Drive	314	576,65 ± 1,38	527,33 ± 2,9	757 ± 13,18	534,37 ± 7,1	564,57 ± 9,62	529,57 ± 6,89	536,73 ± 6,61
Long Note, The	315	465,28 ± 1,65	444,5 ± 2,22	701,43 ± 14,26	456,93 ± 13,74	473,63 ± 11,55	448,8 ± 13,85	479,27 ± 9,12
Billy Nicholson	317	545,47 ± 2,73	504,83 ± 2,31	746,43 ± 15,42	505,83 ± 7,88	529,4 ± 12,21	497,73 ± 8,35	511,07 ± 7,49
Flick It Up And Catch It	330	552,1 ± 1,97	515,7 ± 2,63	783,07 ± 15,04	547,87 ± 7	560,8 ± 14,77	539,07 ± 10,25	551,7 ± 6,25
Tara Lynn's March To God	336	428,91 ± 1,94	410,6 ± 1,28	710,17 ± 13,45	450,67 ± 9,55	479,8 ± 10,38	441 ± 14,96	484,9 ± 13,3
Jewel Of A Hundred Years	338	529,53 ± 2,46	484,77 ± 2,43	790,9 ± 15,56	506,13 ± 8,76	541,47 ± 9,88	495,73 ± 8,13	504,77 ± 7,04
Suleman's Kopanitsa	379	450,05 ± 3,25	427,27 ± 2,49	824,27 ± 16,76	505,07 ± 12,81	546,8 ± 21,73	499,07 ± 16,89	533 ± 9,17
Andy Renwick's Ferret	402	550,98 ± 1,28	499,67 ± 3,5	926,57 ± 15,53	540,73 ± 12,79	624,73 ± 14,37	537,17 ± 12,08	545,2 ± 12,58
Bridge Attack	475	521,88 ± 7,07	503,6 ± 2,81	1015,73 ± 25,36	624,63 ± 20,26	694,8 ± 21,44	638,63 ± 27,76	714,8 ± 13,37

Fonte: Autoria própria.

Tabela 21: Resultados do tempo (ms) dos experimentos do grupo 4.

Música	Nº Notas	A*	ACO	AG	AG 2-opt	AG-SP	AG-SP 2-opt	ED 3-opt
A Visit To Ireland	304	11,75 ± 7,85	5229,67 ± 598,15	45675 ± 14690,36	176133,97 ± 53154,09	5393,43 ± 2063,34	17728,5 ± 8126,84	330859,37 ± 122945,94
Piper In The Cave, The	304	10,33 ± 8,96	3563,8 ± 442,78	50606,23 ± 24115,72	183848,73 ± 53889,98	4677,43 ± 1967,34	14264,73 ± 4886,72	284125,13 ± 120061,93
Liz's	307	11,75 ± 7,85	5253,57 ± 931,36	42134,3 ± 13945,88	180955,63 ± 53176,36	4436,67 ± 1411,2	15405,1 ± 5039,36	257404,27 ± 99812,64
Cape Breton Fiddlers' Society's, The	311	13 ± 6,39	8198,47 ± 1104,5	54070,47 ± 22470,26	182581,47 ± 47377,67	4889,57 ± 1931,86	17585,47 ± 5853,24	370707,8 ± 182156,23
Mouseskin Shoe, The	313	15,6 ± 0,55	7190,17 ± 822,76	53786,03 ± 18264,38	232334,37 ± 66318,19	4971,77 ± 1444,82	19099,2 ± 7398,88	288108,2 ± 104017,09
Wissahickon Drive	314	10,5 ± 8,14	8323,23 ± 1134,03	48180,9 ± 16729,87	199283,5 ± 68791,37	4644,17 ± 1702,78	17356,67 ± 5662,12	329142,87 ± 159094,8
Long Note, The	315	15,6 ± 0,55	7166,33 ± 1028,51	50144,47 ± 14311,27	215569,97 ± 53538,13	4877,1 ± 1455,21	17357,17 ± 6788,26	346245,27 ± 150025,52
Billy Nicholson	317	15,6 ± 0,55	6512,57 ± 1078,73	53483,9 ± 21566,58	185968,13 ± 39715,44	4939,53 ± 1628,26	16458,13 ± 4470,24	378955,63 ± 164191,47
Flick It Up And Catch It	330	15,6 ± 0,55	7071,1 ± 979,03	52776,17 ± 18578,04	176853,13 ± 64332,92	5078,87 ± 1644,8	16588,6 ± 6876,17	286873,4 ± 144686,11
Tara Lynn's March To God	336	19,5 ± 8,35	5993,07 ± 528,47	47195,5 ± 11269,54	210300,77 ± 51522,21	5530,23 ± 1659,09	18278,67 ± 6395,5	473807,3 ± 236198,55
Jewel Of A Hundred Years	338	23,5 ± 8,66	6136,03 ± 653,86	50564,47 ± 17564,85	193818,77 ± 50860,59	5661,8 ± 1482,03	19349,83 ± 6971,33	414583,6 ± 138250,56
Suleman's Kopanitsa	379	28 ± 7,28	8566,53 ± 1326,57	73301,23 ± 32709,51	305901 ± 96149,81	7292 ± 3046,84	25288,43 ± 8912,25	437262 ± 220654,31
Andy Renwick's Ferret	402	19,5 ± 7,68	7334,13 ± 883	61900,7 ± 26982,47	346929,87 ± 235384,56	7576,47 ± 2246,96	27517,53 ± 9337,29	526115,73 ± 232778,92
Bridge Attack	475	54,5 ± 8,66	10618,47 ± 1467,85	96234,93 ± 37150,26	483465,26 ± 124538,3	11057,87 ± 3752,6	43227,9 ± 19487,08	747623,2 ± 499058,12

Fonte: Autoria própria.

APÊNDICE C – TRANSCRIÇÃO COMPLETA DA MÚSICA THE MEDIATOR

Mediator, The

1

MuseScore

ACO

AG-SP 2-opt

AG 2-opt

2-3-2-2-0 3 2-3-5-10-5 2-3-2-2-0 3 2-3-5-10-9 7-9-7-15-14-15

2-3-2-2-0 7 7-8-10 10 11-12-11-11 3 2-3-5-10-9 12-14 15-14-15 16

16-17-16-16-14-12 11-12-14 15 14 16-17-16-16-14-12 11-12-14 15-14 16-18 20-19-20 21

2-3-2-2-0 17 16-17-19 19 21-22-21-21-19 21-22-24 24-23 21 15-14-15 18-16

22

6

MuseScore
T
A
B

ACO
T
A
B

AG-SP 2-opt
T
A
B

AG 2-opt
T
A
B

11

MuseScore
T
A
B

ACO
T
A
B

AG-SP 2-opt
T
A
B

AG 2-opt
T
A
B

16

MuseScore
T
A
B

ACO
T
A
B

AG-SP 2-opt
T
A
B

AG 2-opt
T
A
B