AVALIAÇÃO ENTRE DIFERENTES ABORDAGENS EVOLUTIVAS PARA O PROCESSO DE TRANSCRIÇÃO AUTOMÁTICA DE TABLATURAS

João Víctor Ramos
Universidade Tecnológica Federal do Paraná
Avenida Alberto Carazzai, 1640 - Cornélio Procópio - PR
joao.v.ramos@outlook.com

André Stylianos Ramos
Universidade Tecnológica Federal do Paraná
Avenida Alberto Carazzai, 1640 - Cornélio Procópio - PR
andre.stylianos@gmail.com

Carlos N. Silla Jr
Universidade Tecnológica Federal do Paraná
Avenida Alberto Carazzai, 1640 - Cornélio Procópio - PR
carlosjunior@utfpr.edu.br

Danilo Sipoli Sanches
Universidade Tecnológica Federal do Paraná
Avenida Alberto Carazzai, 1640 - Cornélio Procópio - PR
danilosanches@utfpr.edu.br

RESUMO

O problema na transcrição de uma tablatura de violão é a conversão da música na notação musical padrão (partitura) para uma notação alternativa conhecida como tablatura. Considerando que cada nota pode ser tocada em diferentes posições do violão, esta conversão não é um processo simples. Neste trabalho, este tipo de problema foi categorizado como um problema de otimização. Por esta razão, foram aplicados três diferentes algoritmos evolucionários (algoritmos genéticos, algoritmos genéticos com subpopulações e colônia de formigas). Os resultados experimentais foram obtidos a partir de uma base de dados composta por 10 músicas. Com base nestes resultados, é possível observar que a otimização por colônia de formigas e os algoritmos genéticos com subpopulações produziram bons resultados para o problema tratado.

PALAVRAS CHAVE. Algoritmos evolutivos, otimização por colônia de formigas, transcrição de tablaturas.
Área principal: MH Metaheurísticas; OC - Otimização Combinatória; OA - Outras aplicações em PO

ABSTRACT

The problem of guitar tablature transcription is the conversion of a song in standard music notation (music sheet) to an alternative notation known as guitar tablature. Considering that each note can be played in different positions in the guitar, this conversion is not a straightforward process. In this paper we address the problem by categorizing it as an optimization problem. We have employed three different evolutionary algorithms (Genetic Algorithms, Genetic Algorithms with subpopulations and Ant Colony Optimization). Our experimental results with a dataset of 10 music songs shows that the Ant Colony Optimization and GA with Subpopulations produces good results for this task.

KEYWORDS. Evolutionary computition, ant colony optimization, tablature transcription.

Main Area: MH Metaheuristics; OC - Combinatorial Optimization; OA - Other applications in OR

1. Introdução

Diferentes formas de notação musical foram desenvolvidas e utilizadas através dos séculos. Algumas das antigas formas de notação musical podem ser encontradas em escritas cuneiforme em cidades da antiga Mesopotâmia (Kilmer 1986). Atualmente, a forma mais conhecida para transcrição de músicas é a notação musical moderna, conhecida, também, por partitura. Entretanto, a partitura não é tão intuitiva quanto uma tablatura para músico que irá tocar um violão ou uma guitarra. O motivo é devido ao fato de uma tablatura conter a informação exata de qual corda e qual casa (se houver) necessitam ser tocadas, enquanto na partitura o músico precisa decidir onde irá tocar cada nota em particular. A figura 1 mostra o exemplo de uma pequena sequência musical em uma partitura e sua equivalência em uma tablatura.

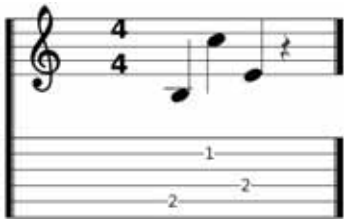


Figura 1: Exemplo de uma partitura e uma possível tablatura para o violão.

O processo de decidir onde tocar cada nota em particular no violão (também conhecido como transcrição) não é um procedimento simples, pois cada nota pode ser reproduzida, em média, em quatro diferentes posições do violão. Por exemplo, a nota "C5" pode ser tocada em cinco posições diferentes, como mostrada a figura 2.

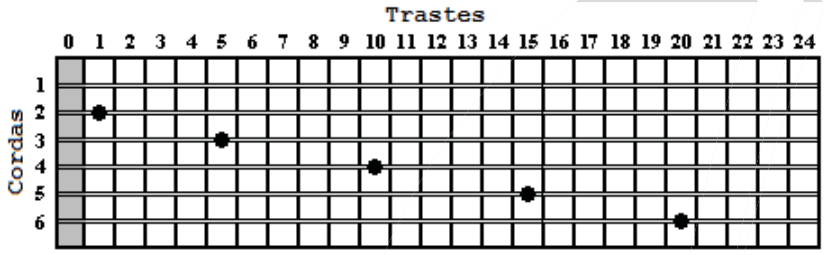


Figura 2: Diferentes posições que C5 pode ser tocada no violão.

O processo de transcrição tablaturas de violão pode ser enquadrado como um problema de otimização combinatória como se segue. Iremos definir que cada posição de uma determinada nota possa ser representada pela tupla $< s, f >$, onde s indica a corda (1-6) a ser tocada e f indica qual traste (casa) será pressionada (de 1 a 24). Se a corda for tocada para produzir uma nota sem pressionar nenhuma traste, f terá seu valor igual a zero.

O exemplo apresentado na figura 2, aborda que a nota "C5" pode ser tocada nas posições: $< 2, 1 >$, $< 3, 5 >$, $< 4, 10 >$, $< 5, 15 >$ e $< 6, 20 >$. Portanto, cada música pode ser vista como uma sequência de notas, em que a transição de uma nota para a próxima pode ter múltiplos caminhos. Por exemplo, vamos considerar uma sequência musical curta de três notas: $B3$, $C5$, $E4$. Neste caso, como ilustrado na figura 3, existem 30 diferentes tablaturas possíveis. Neste sentido, é possível observar que a complexidade computacional deste problema aumenta de acordo com o número de notas de uma música.(Radicioni 2004)(Heijink 2002)

A natureza da otimização combinatória deste problema tem motivado estudos anteriores de empregar algoritmos genéticos (AG) padrões para essa tarefa (Tuohy 2005)(Tuohy 2006c). A principal contribuição deste trabalho é a de avaliar três diferentes abordagens de computação evolucionária para o processo de transcrição automática de tablaturas de violão e mostrar que algumas

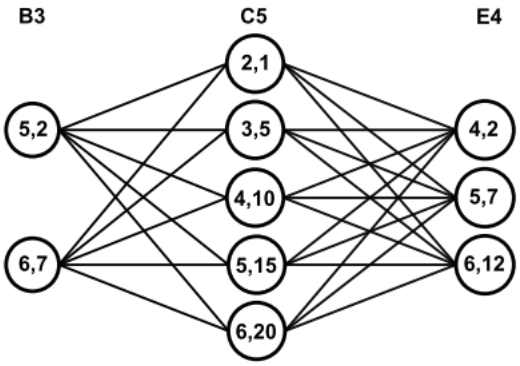


Figura 3: Diferentes posições que $B3$, $C5$ e $E4$ podem ser tocados em um violão.

abordagens são melhores do que o AG tradicional, o qual é atualmente, o estado da arte para este processo. As abordagens evolutivas utilizados neste trabalho são: o AG tradicional, o AG com subpopulações e o algoritmo de otimização por colônia de formigas (ACO) (*ant colony optimization*).

2. Metodologias

2.1. Avaliação da Transcrição

De acordo com (Heijink 2002), há diferentes tipos de dificuldades biomecânicas que determinam a complexidade de movimento das mãos. Algumas delas são: a posição da mão no braço da guitarra, a distância entre os dedos e o reposicionamento dos dedos através das notas que são tocadas. Devemos notar que até o presente momento, não há nenhuma maneira padronizada de como calcular estas dificuldades. Neste trabalho mapeamos o braço do violão em um espaço bidimensional. Isso nos permite calcular a distância entre duas posições a serem tocadas utilizando a distância euclidiana, apresentada na equação (1)

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{1}$$

onde x_1 representa a posição da corda atual; x_2 representa a posição da próxima corda; y_1 representa posição da traste atual; y_1 representa posição da próxima traste;

Considerando que, dadas duas notas, a transição entre a primeira nota e a segunda nota pode ter mais de uma possibilidade, é necessário para calcular a distância euclidiana para a posição atual em relação a todas as possíveis transições. No caso de notas que podem ser tocadas utilizando apenas a corda da guitarra, ou seja, não pressionando nenhuma posição do braço da guitarra, a distância é calculada considerando-se apenas a distância entre as cordas.

2.2. Algoritmo Genético Proposto

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

Inicialmente, foi proposto utilizar um AG tradicional, onde o espaço de busca representa todas as soluções possíveis. Em outras palavras, o espaço de busca considera todas as sequências de notas para tocá-las no violão. Neste sentido, um gene representa as notas para uma posição a ser tocada. Assim, 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 que está sendo tocada.

Há várias situações onde muitas sequências de notas não são consideradas viáveis para serem tocadas por causa da dificuldade de movimentação dos dedos sobre o braço do violão. A fim de avaliar a qualidade de uma solução, uma função de *fitness* é necessária. A função de *fitness* utilizada neste trabalho consiste na soma das distâncias de todas as notas no cromossomo (ver seção 2.1). Neste caso, a distância mais curta entre as notas tocadas é considerada a melhor solução para o problema.

O início do AG tradicional proposto consiste em gerar aleatoriamente uma população inicial de indivíduos. Um indivíduo é formado pelo cromossomo e o seu *fitness*. Esta população inicial garante uma boa diversidade de soluções porém raramente atende soluções viáveis. Desta forma a população inicial é evoluída a fim de gerar melhores indivíduos. Após a geração da população inicial, uma seleção por torneio é aplicada. No torneio três indivíduos são selecionados onde o que apresentar um melhor *fitness* irá compor uma população intermediária. O processo de torneio se repete até que a população seja completamente preenchida.

Em seguida o operador de *crossover* é utilizado. No *crossover*, um par de indivíduos é selecionado a fim de que suas características sejam trocadas, conforme ilustrado na figura 4. Para que isto seja possível, um ou mais pontos de corte são definidos em uma região do cromossomo de forma que possam gerar segmentos a serem trocados entre os indivíduos selecionados. (Lacerda 1999). O cruzamento é responsável por gerar novos indivíduos.

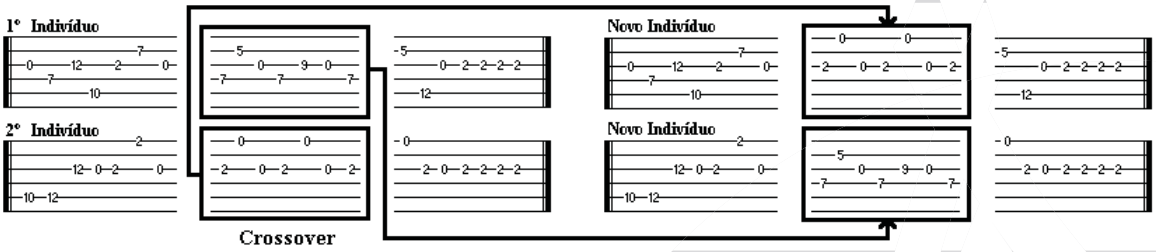


Figura 4: Operador de *crossover* aplicado ao indivíduo onde o segmento central é permutado.

O operador de mutação é responsável por trocar a posição (corda e traste) de uma nota específica para outra posição no braço do violão. É importante observar que durante o processo de mutação, a nota musical permanece a mesma, mudando apenas a sua posição. Este processo de mutação foi baseado nas abordagens propostas por (Radicioni 2004) e (Tuohy 2006c). Neste sentido, o operador de mutação proposto neste trabalho sempre dá prioridade à distância mais curta entre a nota predecessora e a posição atual da nota que será modificada.

Este procedimento é realizado utilizando o cálculo da distância (ver secção 2.1) entre as outras posições possíveis da nota em que será aplicada a mutação. A figura 5 ilustra o operador de mutação proposto. Neste exemplo, na posição $< 5, 12 >$ é aplicada a mutação, e as distâncias entre a nota anterior e as outras possíveis posições candidatas são calculadas. Então a posição $< 4, 7 >$ que possui a distância mais curta é selecionada.

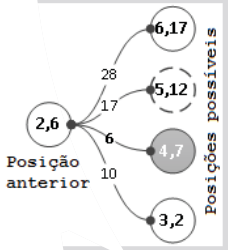


Figura 5: O operador de mutação.

2.2.1. Algoritmo Genético com Subpopulações

O AG baseado em subpopulações proposto é muito semelhante ao AG tradicional, no entanto, inclui as características que permitem a criação e manipulação de indivíduos em mais de uma população.

A abordagem consiste em dividir a população inicial em várias subpopulações, onde cada uma delas armazena um indivíduo completo, mas prioriza apenas um conjunto de notas deste indivíduo para a aplicação dos operadores de *crossover* e mutação.

Na figura 6, uma amostra de uma música com 15 notas é apresentada onde apenas um conjunto de três notas é tratada em cada uma das cinco subpopulações criadas.

Como cada subpopulação armazena uma solução completa, ou seja, toda a música, o *fitness* geral (G.Fitness) desta música é armazenado nesta subpopulação. Além do *fitness* geral, o *fitness* parcial (P.Fitness), que refere-se ao conjunto de notas tratadas em cada subpopulação específica, também é armazenado.

Este processo é mostrado na figura 7, onde durante várias iterações do algoritmo, cada subpopulação irá armazenar diferentes soluções do problema.

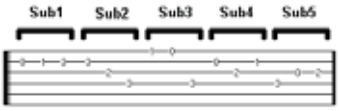


Figura 6: A música dividida em 5 subpopulações.

	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	15	275	9	288	14	148	12	155	8	207
Indiv. 2	16	162	10	290	15	275	14	159	10	230
Indiv. 3	18	275	11	294	16	149	15	170	11	264
Indiv. 4	20	162	13	299	16	162	15	275	13	272
Indiv. 5	20	300	14	208	16	193	16	162	14	320
Indiv. 6	22	330	15	275	17	264	16	200	15	330

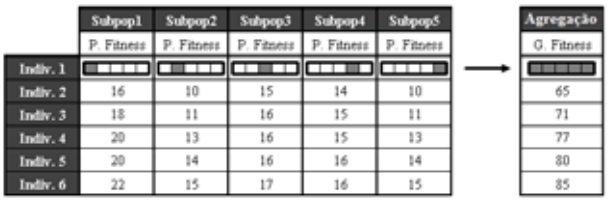
Figura 7: Exemplo dos valores *fitness* armazenados nas subpopulações.

Para melhorar a convergência do algoritmo, propõe-se a inclusão de uma subpopulação chamada agregação. Esta nova subpopulação é distinta pela concatenação do conjunto de notas que foram tratadas separadamente em cada subpopulação, gerando, assim, um novo indivíduo.

A figura 8 ilustra o processo de criação do primeiro indivíduo da tabela de agregação. Neste exemplo, o indivíduo da tabela de agregação é gerado a partir dos conjuntos de notas tratadas pelo primeiro indivíduo de cada subpopulação. O processo repete-se até que o último indivíduo é gerado na subpopulação de agregação (ver figura 9).

Figura 8: Construção do primeiro indivíduo da subpopulação agregação.

A seleção por torneio foi usado para escolher as melhores soluções entre todas as subpopulações. Para este propósito, um indivíduo de cada subpopulação, incluindo a subpopulação de agregação, é selecionado para o torneio e assim compor a população intermediária. Após a geração da população intermediária, os operadores de *crossover* e de mutação são aplicados da mesma forma como mostrado na seção 2.2.



	Subpop1	Subpop2	Subpop3	Subpop4	Subpop5	Agregação
	P. Fitness	P. Fitness	P. Fitness	P. Fitness	P. Fitness	G. Fitness
Indiv. 1						
Indiv. 2	16	10	15	14	10	65
Indiv. 3	18	11	16	15	11	71
Indiv. 4	20	13	16	15	13	77
Indiv. 5	20	14	16	16	14	80
Indiv. 6	22	15	17	16	15	85

Figura 9: Subpopulação agregação com todos os indivíduos gerados.

Após a aplicação dos operadores de cruzamento e mutação na população intermediária, os melhores indivíduos migram para as subpopulações que substituem os piores indivíduos. Este procedimento acontece através da comparação entre o *fitness* parcial dos novos indivíduos gerados e dos indivíduos pertencentes às subpopulações. Assim, os melhores indivíduos da população intermediária substituirão os piores indivíduos das subpopulações.

Neste caso, cada indivíduo da população intermediária é selecionado verificando se o *fitness* parcial deste indivíduo é menor do que o do pior indivíduo da primeira subpopulação.

Caso for, o pior indivíduo desta subpopulação é substituído pelo indivíduo selecionado. Caso contrário, a subpopulação permanece inalterada. Este processo é repetido para todas as subpopulações, com exceção da subpopulação de agregação.

O processo de migração na subpopulação de agregação é semelhante ao processo inicial. No entanto, após a geração dos indivíduos pelo método concatenação, os indivíduo que irão ficar na subpopulação de agregação são aqueles que se dispuserem de um *fitness* menor entre os indivíduos concatenados e os da população intermediária.

2.3. Otimização por Colônia de Formigas

Outra abordagem investigada neste trabalho é baseado na otimização por colônia de formigas, ou *ACO* (Blum 2005). Este algoritmo foi desenvolvido tendo como inspiração um fenômeno da natureza. Ao estudar-se o comportamento de formigas de uma colônia recém criada, enquanto estas procuram alimentos para trazer de volta para a colônia, foi elaborada uma hipótese sobre o motivo da eficiência destas formigas ao encontrar com certa precisão um dos caminhos mais curtos existentes para fontes de comida (Deneubourg 1990). A hipótese proposta se baseia em uma substância, chamada feromônio, depositada por cada formiga pelo caminho percorrido e que evapora com o passar do tempo. As formigas escolhem um caminho à percorrer com uma probabilidade proporcional à quantidade de feromônio existente em cada caminho possível. Com isso, o ACO se encaixa no objetivo deste estudo que visa encontrar o menor caminho de movimentação do dedo de um músico pelo braço do violão.

Neste contexto, para abordar o problema em questão, uma representação baseada em uma representação por grafos, os quais consistem de vértices e arestas que podem ser usados para simular caminhos pelos quais as formigas podem se movimentar, é necessária. Para se modelar o problema proposto como sendo o caminho de uma formiga, deve-se considerar um caminho completo como sendo a passagem da formiga por posições no braço do violão que possibilitam tocar sequencialmente cada nota que compõe uma música. Com isto, durante o processo de extração da informação do arquivo *MusicXML*, uma lista de adjacência é gerada com todas as notas da música. Assim, cada vértice no grafo é definida como uma posição possível para uma dada nota e as arestas representam um caminho entre uma posição e as possíveis posições de nota a seguir.

Um vértice tem duas partes de informação relacionadas a um plano bidimensional: a corda e o traste que devem ser tocados para soar uma certa nota. A distância de um caminho entre dois vértices é obtido através do cálculo da distância euclidiana (ver secção 2.1) entre as suas posições dadas. É importante notar que o nome dado a pegada é o valor que indica a quantidade de feromônio contida em uma aresta.

Após a construção do grafo, uma formiga é alocada para cada possível posição referente a nota inicial da música. Feito isso, as formigas devem atravessar o grafo até que eles cheguem a um vértice a partir do qual não há mais caminhos para outros vértices existentes, em outras palavras, um vértice que indica a posição da nota final da música que está sendo transcrita. Na figura 10, ao olhar para a linha tracejada, é possível observar o trajeto feito por uma formiga.

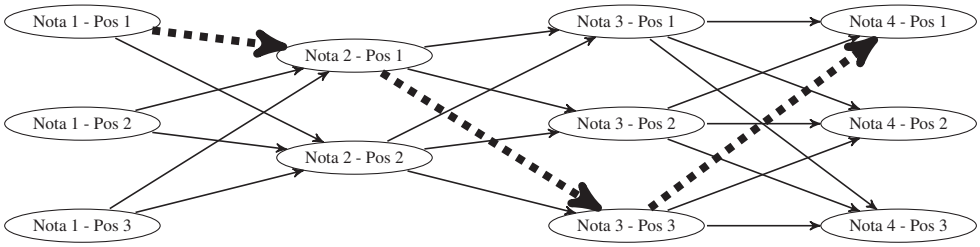


Figura 10: Possível caminho completo.

Na abordagem proposta, considera-se um caminho completo percorrido por uma formiga como uma das possíveis sequências de posições que devem ser pressionadas e tocadas no braço do violão para se reproduzir a música sendo transcrita. O caminho completo de uma formiga define portanto uma solução do problema em questão.

Para modelar computacionalmente o funcionamento de uma colônia de formigas, alguns valores devem ser definidos de forma a serem compatíveis com o problema a ser resolvido. Para alguns operadores (Dorigo 2006), que descrevem as diversas variáveis envolvidas neste fenômeno, a constante de evaporação ρ, a importância relativa da pegada α, a importância relativa da visibilidade β e a constante de qualidade para a pegada Q.

Os melhores valores obtidos para ρ, α, β e Q foram encontradas empiricamente. O algoritmo é definido pela seguinte repetição cíclica. Uma ou mais formigas estão posicionadas em possíveis posições da nota inicial e escolhem, conforme indicado na figura 11, os caminhos para viajar com uma probabilidade específica conforme a equação (2):

$$p_{ij}^{k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{I \in N_i^k}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}} \tag{2}$$

onde k é cada formiga, ij é a aresta entre os vértices i e j, p_{ij}^{k} é a probabilidade da formiga escolher caminhar por uma determinada aresta, N_i^k são os vértices aos quais a formiga pode ir a partir de sua posição atual, η_{ij} é o valor heurístico de cada caminho que é inversamente proporcional à sua distância e τ_{ij} é a pegada que é encontrada em uma determinada trilha. As formigas atravessam o grafo até que o movimento já não é possível.

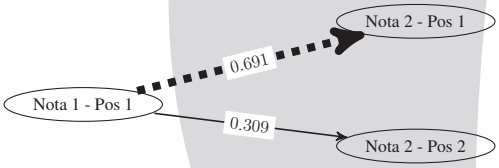


Figura 11: Representação de probabilidades de escolha entre dois caminhos para uma formiga.

Após todas as formigas construirem uma solução, o valor da pegada a ser depositada em cada borda percorrida é calculada, a qual é obtida pela equação (3):

$$\Delta\tau_{ij}^{(k)} = \frac{Q}{L_k} \tag{3}$$

onde L_k é o tamanho relativo da borda. Para as arestas que não tenham sido percorridas não há adição de pegada.

Além de adicionar o valor de pegada depositados por cada formiga, a pegada existente em cada aresta deve ser evaporada, de acordo com o valor da constante ρ, dada a equação (4):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{(k)} \quad (4)$$

No final da atualização dos valores das pegadas no grafo, uma nova iteração se inicia.

O processo se repete até que o número de iterações desejadas seja atingido. A fim de evitar convergências prematuras, o valor de β é dobrado quando uma ocorrência recorrente de estagnação é identificada. No final do processo as formigas tendem a convergir para um único percurso que será a melhor solução conhecida.

3. Experimentos

3.1. Base de Dados

Neste trabalho empregamos um conjunto de dados com uma quantidade variável de notas. Para obter uma análise mais confiável, foram retiradas desta base uma amostra de 10 partituras que contemplam as músicas com quantidades menores, quantidades intermediárias e as musicas com as maiores quantidades de notas. Os dados obtidos são provenientes do site *www.thesession.org*, que é um recurso importante para a comunidade de música tradicional irlandesa. Um aspecto importante das músicas deste site é que eles estão todas disponíveis gratuitamente por serem canções folclóricas. A tabela 1 apresenta as 10 partituras utilizadas neste trabalho contendo o nome da música e seu respectivo número de notas.

Tabela 1: Base de dados com 10 partituras musicais.

Instância	Nome da música	Número de notas
1	Ailiu Eanai	25
2	Hip Hop Berceuse	50
3	Griffenfeldt	100
4	Finchale	150
5	Sound Of Sleat, The	200
6	Alick C McGregor	250
7	Pressed For Time	300
8	School Reels	357
9	Andy Renwick's Ferret	402
10	Bridge Attack	475

3.2. Parâmetros Utilizados

A fim de comparar as metodologias ACO, AG com as subpopulações e um AG tradicional para o processo de transcrição de tabulaturas de violão, dez músicas foram usadas e foram mostrados na tabela 1. As metodologias ACO, AG com subpopulações e AG tradicional foram executados 10 vezes para cada música canção. Cada execução avaliou 20.000 soluções.

Os testes foram realizados utilizando um computador dotado de um processador 3.07 GHz Intel Xeon X5675, 16GB de memória RAM, com o sistema operacional Microsoft Windows Server 2008 R2 e a linguagem de programação Java versão 8.

Os parâmetros utilizados pelo ACO foram: α = 0,15, β = 1, ρ = 0,5 e Q = 200, enquanto na tabela 2 são mostrados os parâmetros utilizados pelo AG com subpopulações. É importante ressaltar que os parâmetros utilizados foram encontrados de forma empírica e que o AG tradicional usa os mesmos parâmetros do AG com subpopulações, com exceção de possuir uma única população com 150 indivíduos.

Tabela 2: Parâmetros utilizados pelo AG com subpopulações.

Parâmetros	Valores
Quantidade de subpopulações	15
Quantidade de indivíduos por subpopulação	10
Taxa de mutação	50%
Taxa de *crossover*	55%
Pontos de *crossover*	3
Seleção por torneio	3 indivíduos

3.3. Resultados Experimentais

A tabela 3 permite uma comparação entre o ACO, AG com subpopulações e o AG tradicional para o processo de transcrição.

Claramente, o ACO e o AG com subpopulações superam o AG tradicional com menor *fitness* (distância mínima necessária para tocar todas as notas da música). Em relação ao tempo de execução, é importante observar que o ACO encontrou um menor tempo quando comparado com AG com subpopulações e que o AG tradicional. Importa mencionar que maioria dos trabalhos existentes na literatura utilizam apenas um AG tradicional para o processo de transcrição automática. Neste sentido, este trabalho contribui com uma importante discussão entre abordagens alternativas para o problema tratado.

Tabela 3: Resultados obtidos para as 10 instâncias.

		ACO		AG Subpop		AG	
		FT[1]	TE[2]	FT	TE	FT	TE
1	MD[3]	19	20.6	19	64.3	19	26.8
	DP[4]	0	0.5	0	1.6	0	0.5
2	MD	47	16.9	47.2	69.6	50.8	52.9
	DP	0.6	0.4	0.4	1.1	1.5	0.3
3	MD	67.2	43.6	73.4	77.9	116.9	113.9
	DP	0.6	0.6	3.6	0.7	2.1	2.3
4	MD	114.8	51.1	102.0	86.1	171.7	185.8
	DP	1.2	0.9	0.0	6.5	5.2	4.2
5	MD	152.7	60.8	150.6	89.1	195.0	278.8
	DP	0.6	1.4	1.7	2.8	5.2	9.6
6	MD	189.9	71.4	190.1	105.8	397.0	373.9
	DP	1.9	1.6	2.2	2.6	5.1	17.1
7	MD	184.0	84.2	190.0	114.8	525.1	469.7
	DP	0.0	1.6	7.2	1.8	7.9	13.2
8	MD	312.3	69.1	294.3	126.5	549.6	603.4
	DP	1.4	1.5	0.8	2.5	6.4	27.5
9	MD	296.6	126.6	307.3	133.6	803.3	697.1
	DP	0.9	1.2	9.9	3.1	6.9	10.1
10	MD	405.1	57.9	394.3	147.1	847.6	845.7
	DP	5.1	1.7	4.4	1.4	12.1	12.5

[1]*Fitness*. [2]Tempo de execução (ms). [3]Média. [4]Desvio padrão.

Na figura 12 é possível verificar a comparação dos resultados de *fitness* obtidos nos 10

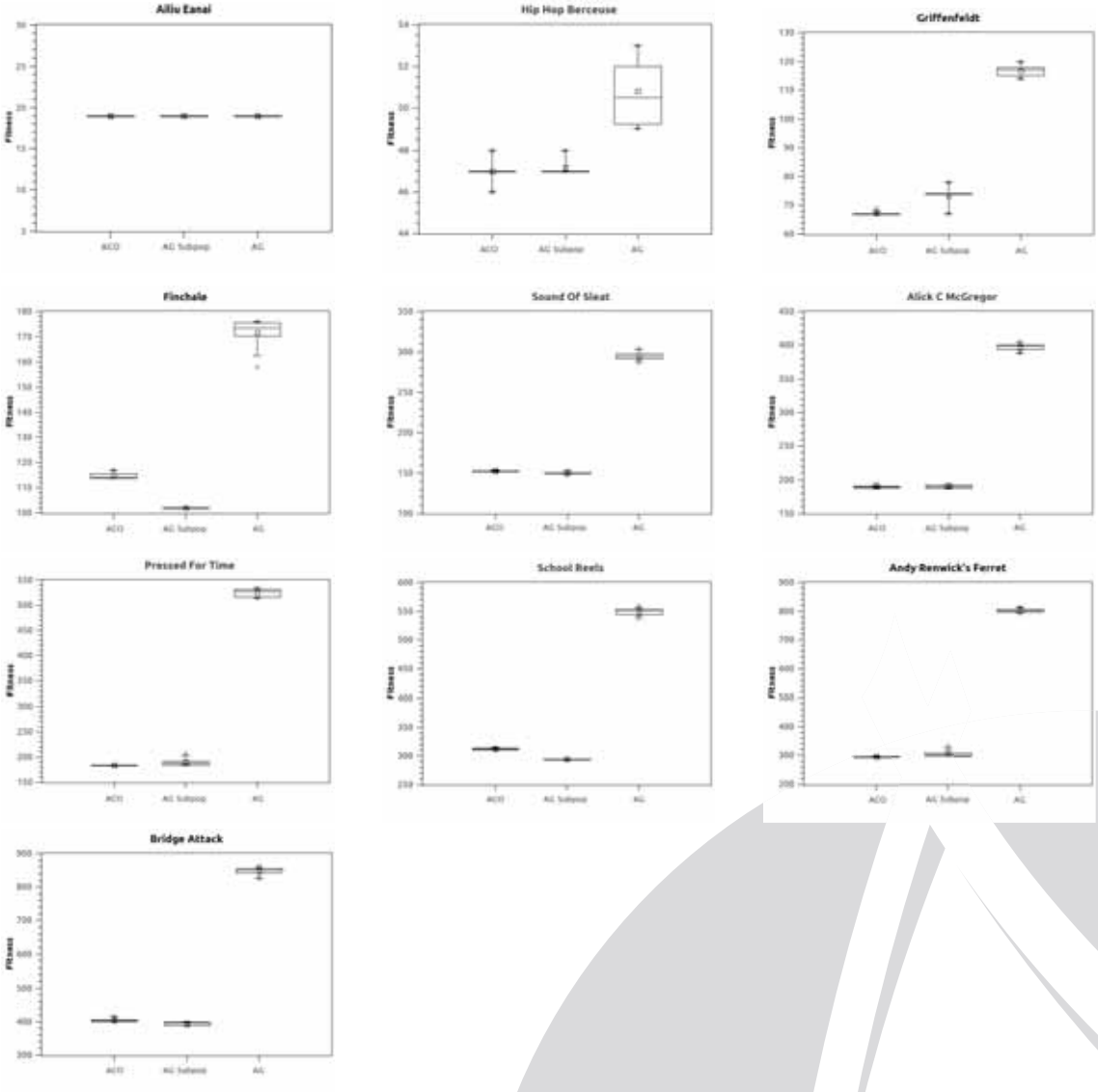


Figura 12: Representações gráficas obtidos a partir dos resultados dos 10 experimentos para cada música.

experimentos para as diferentes abordagens propostas neste trabalho. Outro fator a ser destacado é o processo de convergência entre os algoritmos. Com isso, a figura 13, ilustra este processo de convergência para a música *Bridge Attack*, que é a música composta pelo maior número de notas em relação a todas as outras músicas da base de dados utilizada. Neste sentido, é possível destacar o AG com subpopulações que obteve uma melhor convergência em relação aos demais algoritmos comparados.

Finalmente, a figura 14 ilustra as diferentes notas usadas pelo ACO, AG com subpopulações e o AG tradicional em um pequeno trecho da música *School Reels*. É possível perceber que o ACO e o AG com subpopulações utilizam a maioria das notas similares, enquanto o AG tradicional utiliza cinco diferentes posições para as notas.

4. Discussões e Trabalhos Relacionados

Não há muitos estudos que têm abordado o problema da transcrição automática tablaturas de violão. No trabalho de (Radicioni 2004) os autores abordam o problema como um problema de satisfação de restrições, onde a música é dividida em vários segmentos e cada segmento é representado em um grafo. No trabalho de (Tuohy 2005) um método para transcrever uma partitura em

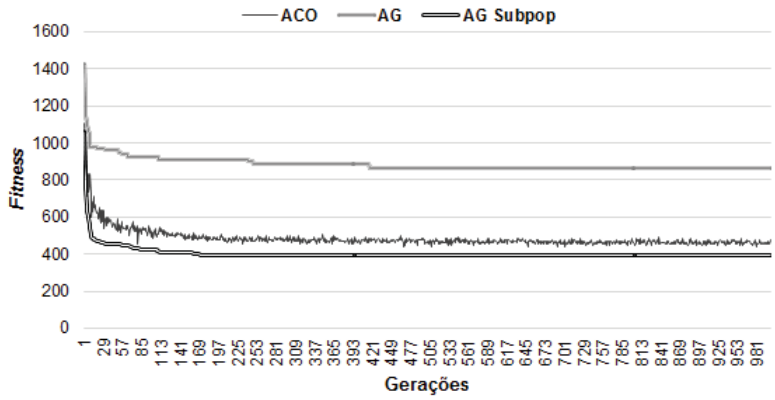


Figura 13: Gráfico de convergência obtido a partir do desempenho dos algoritmos para a música *Bridge Attack*.

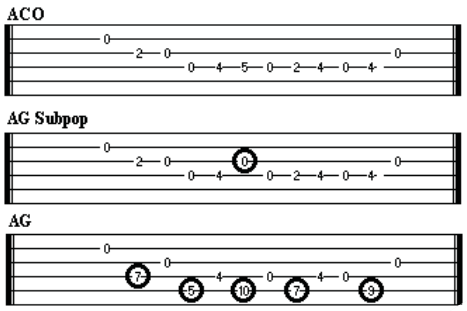


Figura 14: Uma parte da música *School Reels*.

uma tablatura de violão usando um algoritmo genético foi apresentado. De acordo com os autores de (Tuohy 2005) as tablaturas geradas são fáceis de tocar e alcançaram um elevado grau de consistência quando comparadas com a transcrição original da música e sua respectiva tablatura. Outras abordagens utilizadas para o processo pelo mesmo grupo foram as redes neurais artificiais (Tuohy 2006b) (Tuohy 2006a) e a combinação do algoritmo genético com a técnica *greedy hill-climber* (subida da encosta) (Tuohy 2006c).

A análise das abordagens existentes mostra que, para o melhor do conhecimento dos autores, apenas o algoritmo genético tradicional foi usado para resolver este problema. Os resultados computacionais obtidos neste trabalho, mostram que outras abordagens evolutivas (ACO ou AG com as subpopulações) podem conseguir melhores resultados do que o algoritmo genético tradicional.

5. Conclusões

Este artigo apresentou uma avaliação importante de diferentes abordagens evolucionárias para o processo de transcrição de tablaturas de violão. As abordagens utilizadas neste trabalho foram ACO, AG tradicional e AG com subpopulações. O AG com subpopulações proposto é responsável por compartilhar uma única música em vários segmentos distribuídos em várias subpopulações. Nos experimentos, as abordagens ACO, AG com as subpopulações e AG tradicional foram aplicados em um conjunto de dados de 10 músicas que variam entre 25 a 475 notas. Os resultados das simulações indicaram que ACO e o AG com subpopulações obtiveram melhores resultados quando comparados com AG tradicional. No entanto, em todos os conjuntos de dados o ACO obteve um tempo de operação reduzido quando comparado com as outras abordagens. A partir dos resultados obtidos, é possível observar que as abordagens evolucionárias como o ACO e o AG com subpopulações alcançaram resultados satisfatórios e melhores que o AG tradicional utilizado

na literatura. Finalmente, este trabalho proporcionou uma base interessante para avaliar aspectos promissores de diferentes abordagens de algoritmos evolucionários para o processo de transcrição de tablaturas.

6. Agradecimentos

Os autores gostariam de agradecer à CAPES, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Fundação Araucária pelo apoio financeiro dado a esta pesquisa.

Referências

Blum, C., Ant colony optimization: Introduction and recent trends. *Physics of Life reviews* 2.4, 353–373, 2005

Darwin, C., *On the origins of species by means of natural selection*. Murray, London, 247, 1958.

Deneubourg, J.-L., Aron, S., Goss, S., e Pasteels, J. M. (1990), The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior* 3, 2, 159–168.

Dorigo, M., Birattari, M., e Stutzle, T. (2006), Ant colony optimization. *Computational Intelligence Magazine*, IEEE 1, 4 , 28–39.

Heijink, H., e Meulenbroek, R. G. (2002), On the complexity of classical guitar playing: functional adaptations to task constraints. *Journal of motor behavior* 34, 4, 339–351.

Holland, John 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., e Civil, M. (1986), *O*ld babylonian musical instructions relating to hymnody. *Journal of Cuneiform Studies*, 94–98.

Lacerda, E. G., & Carvalho, A. C. P. L. F.. Introdução aos algoritmos genéticos. *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*, 1, 87-148, 1999

Radicioni, D., Anselma, L., e Lombardo, V. (2004), A segmentation-based prototype to compute string instruments fingering. *In Proceedings of the Conference on Interdisciplinary Musicology*, p. 97.

Tuohy, D., e Potter, W. (2005), A genetic algorithm for the automatic generation of playable guitar tablature. *In Proceedings of the International Computer Music Conference*, pp. 499–502.

Tuohy, D. R., e Potter, W. (2006a), Guitar tablature creation with neural networks and distributed genetic search. *In Proceedings of the 19th International Conference on Industrial and Engineering Applications of Articial Intelligence and Expert Systems*, pp. 27-30.

Tuohy, D. R., Potter, W., e Center, A. I. (2006b), An evolved neural network/hc hybrid for tablature creation in ga-based guitar arranging. *In Proceedings of the International Computer Music Conference (ICMC'06)*.

Tuohy, D. R., e Potter, W. D. (2006c), Ga-based music arranging for guitar. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 1065–1070.