SISTEMA DE CONSULTA CANTAROLADA COM GERAÇÃO AUTOMÁTICA DE UM BANCO DE MÚSICAS ADAPTATIVO

Maurício do Vale Madeira da Costa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro
Setembro de 2015

SISTEMA DE CONSULTA CANTAROLADA COM GERAÇÃO AUTOMÁTICA DE UM BANCO DE MÚSICAS ADAPTATIVO

Maurício do Vale Madeira da Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Sergio Lima Netto, Ph.D.

Prof. Marcio Nogueira de Souza, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2015

Costa, Maurício do Vale Madeira da

Sistema de consulta cantarolada com geração automática de um Banco de Músicas Adaptativo/Maurício do Vale Madeira da Costa. – Rio de Janeiro: UFRJ/COPPE, 2015.

XVII, 124 p.: il.; 29, 7cm.

Orientador: Luiz Wagner Pereira Biscainho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 82 – 87.

1. Query by humming. 2. Processamento digital de sinais de áudio. I. Biscainho, Luiz Wagner Pereira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

"A natureza nunca nos engana;
somos sempre nós que nos
enganamos."
Jean-Jacques Rousseau

Agradecimentos

Agradeço primeiramente a Deus pela vida que tenho, sabendo que tudo que me acontece é para meu aprendizado. Agradeço a todos os meus amigos e familiares que me incentivaram e apoiaram durante o curso de mestrado, em especial à minha esposa Julia por suportar a distância durante o período em que estive fora do país em pesquisa. Agradeço também aos amigos uruguaios, principalmente a Martín Rocamora e Pablo Cancela, pelos valorosos momentos de trabalho e de convívio e pela dedicação que a mim prestaram. Presto também agradecimentos ao meu dedicado orientador e amigo Luiz Wagner, que me acompanha na jornada acadêmica desde os primeiros períodos de graduação e muito tem contribuído para meu crescimento profissional e pessoal. Agradeço a todos os amigos e colegas de laboratório, que muitas vezes também me prestaram auxílio e compartilharam comigo dos momentos de estudo e trabalho. Obrigado a todos.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMA DE CONSULTA CANTAROLADA COM GERAÇÃO AUTOMÁTICA DE UM BANCO DE MÚSICAS ADAPTATIVO

Maurício do Vale Madeira da Costa

Setembro/2015

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Sistemas de consulta cantarolada, em inglês *query by humming* (QBH), são ferramentas desenvolvidas com o propósito de permitir que usuários realizem consultas de músicas cantarolando suas melodias, em vez de realizarem uma busca textual. Esse tipo de sistema tornou-se mais acessível com a popularização de dispositivos portadores de microfones e com um razoável poder de processamento. O maior problema relacionado aos sistemas de QBH atualmente se encontra na dificuldade de criação do banco de músicas no qual são realizadas as consultas, que, em geral, é construído a partir de transcrições manuais das músicas ou adaptações de transcrições já existentes. Para superar este problema, é possível fazer uso de alguns sistemas que buscam transcrever automaticamente a melodia presente em um arquivo de áudio. Assim, os objetivos deste trabalho são o estudo de sistemas de QBH e o desenvolvimento de formas de permitir maior escalabilidade no uso de tais sistemas. Neste trabalho é apresentado um panorama geral da arquitetura de sistemas de QBH, detalhando, em seguida, o sistema pré-existente Tararira, que foi utilizado para realizar diversos experimentos com bases geradas automaticamente utilizando dois dos principais sistemas de transcrição automática de melodia encontrados na literatura. Como os resultados demonstram que não é possível transcrever todo o banco automaticamente de forma satisfatória, foi proposto um sistema de adaptação automática do banco de músicas, que faz uso de informações simples fornecidas pelos usuários para gradativamente melhorar tal banco, utilizando o sistema Tararira como ponto de partida.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

QUERY-BY-HUMMING SYSTEMS WITH AUTOMATIC GENERATION OF AN ADAPTIVE DATABASE

Maurício do Vale Madeira da Costa

September/2015

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

Query-by-humming (QBH) systems are tools developed with the purpose of allowing people to search for songs by humming their melodies instead of a textual approach. This kind of system has become very accessible as mobile devices with built-in microphone and reasonable computational power are ubiquitous nowadays. The main shortcoming of current QBH systems is related to the creation of the songs database over which the queries are to be performed, which usually consists of manual transcriptions or the adaptation of previously written music scores. In order to overcome this limitation, it is possible to make use of systems which can automatically transcribe the main melody present in an audio file. The objectives of this work are to study QBH systems and to develop a way to properly circumvent the lack of scalability. This work presents an overview of the theory of QBH systems, followed by a more detailed description of an available QBH system called Tararira, over which several experiments using automatically generated music databases extracted by means of two of the principal melody extraction systems found in the literature were performed. The attained results demonstrate the impossibility of transcribing the whole database with satisfactory accuracy, thus suggesting the development of a system (having Tararira as a starting point) capable of automatically increase its music database from using informations gathered from the users themselves.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Apresentação

A Seção 1.1 introduz o tema de consulta por sinal cantarolado e motiva seu estudo e desenvolvimento. Após a exposição dos objetivos na Seção 1.2 , a Seção 1.3 apresenta brevemente a organização do conteúdo abordado em cada capítulo.

1.1 Introdução

Atualmente, a popularização de dispositivos móveis com razoável capacidade de processamento tem mudado a forma como usuários interagem com *software*. Com a crescente quantidade de informação acessível aos usuários, o esforço em aumentar a facilidade de busca por conteúdo tem sido um dos grandes alvos dos fabricantes de *software* e aparelhos eletrônicos. Grande parte das formas de busca utiliza consulta por conteúdo textual, devido ao baixo custo relacionado a armazenamento e comparação de palavras. Porém, em muitos casos, essa categoria de busca não atende satisfatoriamente à forma como pessoas se recordam de ou tomam contato com determinados conteúdos que desejam buscar, como imagens ou sons. No contexto de áudio, sistemas de *query by humming* (QBH) têm sido estudados e aprimorados pela comunidade científica [1–6], por permitirem que usuários realizem a busca por conteúdo musical cantando ou cantarolando parte de sua melodia.[1]

O maior problema relacionado a sistemas de QBH é a geração da base de melodias em que serão realizadas as consultas. Na maioria dos sistemas propostos na literatura, essa base consiste em melodias armazenadas em notação simbólica, como MIDI.[2] Apesar de haver uma grande quantidade de arquivos MIDI ou partituras

[1] Há, na literatura, referências a sistemas com a nomenclatura *query by singing/humming* (QBSH), de forma a incluir as duas formas de busca. Aqui nos referenciaremos a QBH como sistemas que permitem as duas formas de entrada, de maneira mais geral, por ser mais usual.

[2] O padrão MIDI, ou em inglês *Musical Instrument Digital Interface*, é um padrão técnico que descreve um protocolo de comunicação desenvolvido na década de 1970 para enviar informações referentes a instrumentos, como as notas que são tocadas, os instantes em que começaram e deixaram de ser tocadas e informações auxiliares de expressão, entre outras.

disponíveis na *internet*, não há garantia da fidelidade da transcrição com relação à melodia original. Além disso, a maioria desses arquivos contém elementos indesejados, como instrumentos e vocais secundários. Assim sendo, o uso de tais arquivos impõe limitações, por necessitar da adaptação de algum arquivo existente retirando informações desnecessárias e, por vezes, corrigindo a melodia escrita. Sempre que uma nova música tiver de ser incluída no banco, isso demandará trabalho manual em algum nível.

Uma forma de lidar com esse problema é armazenar as consultas cantadas pelos usuários e formar o banco de músicas a partir delas [2]. Cada nova consulta realizada é, então, comparada com as já armazenadas e pode ser utilizada para agregar informação à base. Esse procedimento é utilizado em serviços de busca por músicas como o SoundHound[3], o Midomi[4] e o Tunebot.[5] Porém, o uso dessa abordagem traz diversas limitações, por depender das consultas, frequentemente inconsistentes, dos usuários.

Nesse contexto, desenvolver um processo de geração automática do banco de músicas pode ser uma boa solução para a questão da escalabilidade do sistema. Há, na literatura, algumas propostas nesse sentido [3–6]. O desempenho insatisfatório desses sistemas, quando comparados com sistemas que usam transcrições manuais, mostra que ainda há espaço para novas propostas. Daí surgiu a ideia de desenvolver um sistema híbrido que poderia partir de uma base independente das consultas, obtida, por exemplo, a partir de gravações comerciais quaisquer, e que lançasse mão das próprias consultas para adaptar continuamente a base de músicas gerada.

1.2 Objetivos

O objetivo principal a ser atingido neste trabalho é o desenvolvimento de uma nova abordagem em sistemas de QBH, automatizando a criação e manutenção do banco de músicas de forma eficaz. Como os sistemas de transcrição automática de melodia ainda não são capazes de gerar resultados satisfatórios para toda variedade de música, a proposta é criar um algoritmo de adaptação do banco que faça uso das consultas realizadas pelos usuários, corrigindo ou substituindo elementos defeituosos do banco. Para isso, o sistema [7] descrito na Seção 2.5 será utilizado para os testes e também como base para as adaptações necessárias à criação do novo sistema. Para a criação automática dos bancos de música serão utilizados dois métodos de transcrição automática encontrados na literatura [8, 9].

[3] `http://www.soundhound.com/`
[4] `http://www.midomi.com/`
[5] `http://tunebot.cs.northwestern.edu/`

1.3 Organização

No Capítulo 2, é feito um estudo sobre sistemas de QBH de forma geral, abordando as técnicas mais frequentes na literatura. É descrito, em seguida, o sistema Tararira [7, 10], utilizado neste trabalho.

O Capítulo 3 apresenta a teoria dos dois sistemas de transcrição automática de melodia utilizados da literatura, seguido do Capítulo 4, que contém a descrição das bases de músicas adotadas, da forma como foram gravadas as consultas, dos critérios de escolha das músicas e de como foram transcritas as diferentes versões das bases com os sistemas de transcrição automática.

A discussão dos resultados das transcrições obtidas está presente no Capítulo 5 em detalhes, motivando o desenvolvimento do sistema proposto de adaptação contínua da base de músicas. Então, o Capítulo 6 aborda o sistema proposto, que faz uso das consultas realizadas pelos usuários para adaptar o banco de músicas automaticamente. Em seguida, são descritos e comentados os resultados dos experimentos realizados.

Por fim, no Capítulo 7, são expostas as conclusões do trabalho, incluindo discussões acerca do método proposto e dos resultados obtidos. A proposição de trabalhos futuros se encontra no Capítulo 8.

Capítulo 2

Sistemas de QBH

Neste capítulo, é feito um estudo sobre sistemas de QBH e as principais técnicas de cada estágio de tais sistemas encontradas na literatura, com ênfase nas técnicas utilizadas no sistema Tararira [7, 10], o qual foi utilizado como base para o sistema adaptativo proposto. Em geral, as técnicas não utilizadas em [7, 10] serão brevemente descritas ou apenas citadas.

2.1 Introdução a Sistemas de QBH

Sistemas de QBH funcionam da seguinte forma: o usuário cantarola uma melodia e o sistema transcreve o sinal de entrada a fim de determinar a melodia cantada. De posse dessa melodia, o sistema precisa compará-la com as demais melodias armazenadas em sua base. Para que esse procedimento funcione bem para a maioria dos usuários, o sistema precisa ser robusto a inconsistências comumente presentes na melodia cantada. Assim, é adotada uma representação melódica e é realizada uma codificação que possibilite tornar a comparação de melodias robusta a tais problemas. Por fim, o sistema envia ao usuário uma lista de músicas contidas em seu banco interno ordenadas por similaridade com relação à consulta realizada. Na Figura 2.1 está ilustrado um diagrama básico de um sistema de QBH típico com diversas formas de se gerar o banco de músicas.

Uma primeira maneira de se gerar a base de músicas do sistema é simplesmente alimentá-la com as consultas realizadas pelos usuários. Esse procedimento é bastante utilizado e apresenta a vantagem de tornar o sistema versátil e adaptável às demandas dos usuários. Porém, o uso dessa abordagem traz também diversos problemas, como: a necessidade de haver uma grande quantidade de consultas realizadas para que o sistema comece a funcionar razoavelmente; as melodias cantadas pelos usuários conterem erros; a necessidade de o usuário cadastrar músicas que são buscadas pela primeira vez; e, por fim, haver interferências do ambiente em que os usuários se encontram.

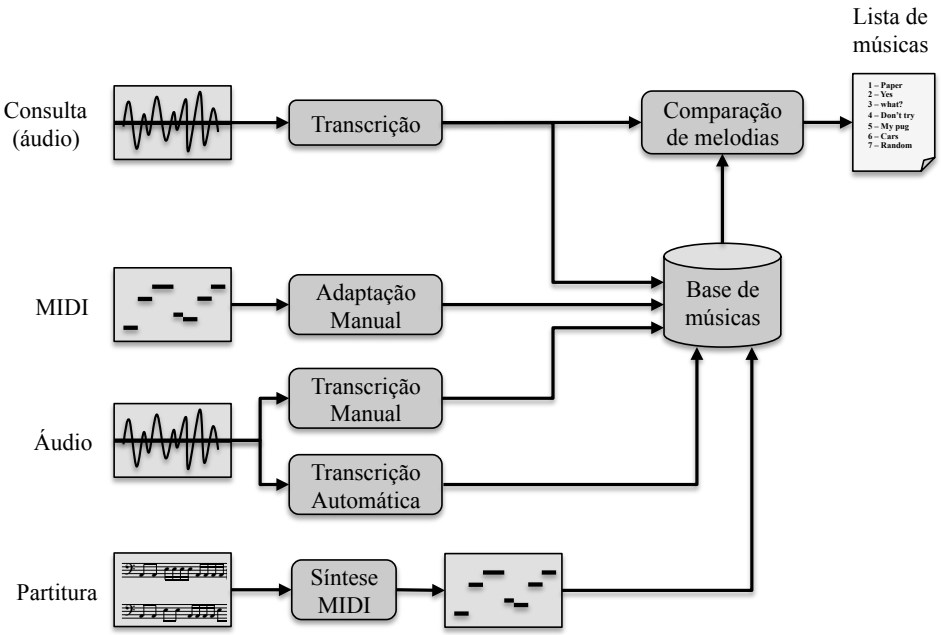


Figura 2.1: Diagrama de blocos ilustrando os princípios de um sistema de QBH.

Uma forma de gerar uma base estática é adaptar arquivos que já se encontrem em formato MIDI, removendo todo o conteúdo que não pertença à melodia principal e, em seguida, convertendo o arquivo para o formato adotado para a base, que, em geral, segue o padrão de representação do próprio MIDI, guardando as informações mais relevantes (quais notas foram tocadas, o instante em que começam e o instante em que terminam). Outra forma muito usual de gerar os arquivos da base de músicas seria realizar uma síntese MIDI da linha melódica presente em partituras das músicas e, em seguida, realizar o mesmo procedimento citado de conversão. Para que seja gerada a base a partir de sinais de áudio, é preciso que um músico transcreva suas melodias para MIDI — o que pode demandar muito tempo e esforço humanos — ou que isso seja feito automaticamente por sistemas especializados em tal tarefa, sendo esta opção menos custosa quanto aos aspectos citados, porém mais sujeita a introduzir erros. No Capítulo 4 será descrita a forma como foram construídos os bancos de músicas utilizados neste trabalho.

2.2 Representação Melódica

Para que as melodias de cada música sejam armazenadas na base do sistema em um formato que seja econômico e que também torne possível realizar uma comparação rápida entre melodias, adota-se uma representação simbólica de alto nível. Nessa representação, são registradas as principais informações da melodia: altura, *onset* (instante em que as notas se iniciam) e *offset* (instante em que as notas terminam)

de cada nota.

O sistema adotado na música ocidental que determina a frequência fundamental (que no presente contexto pode ser aproximadamente associado à altura) de cada nota é a chamada escala igualmente temperada. Nesta escala, o intervalo mínimo entre notas é de um semitom, cuja relação em termos frequenciais é dada por

$$f_{\text{nota}(i+1)} = f_{\text{nota}(i)} \sqrt[12]{2}, \tag{2.1}$$

onde a frequência $f_{\text{nota}(i)}$ da nota i serve de referência à frequência da nota seguinte $f_{\text{nota}(i+1)}$ da escala. Assim, basta multiplicar por $\left(\sqrt[12]{2}\right)^n$ a frequência de uma nota para se chegar a uma que está a um intervalo de n semitons, o que vale para intervalos positivos ou negativos. Pela convenção da notação científica proposta pela Sociedade de Acústica da América [11] a nota Lá central ($\text{A}_4 = 440$ Hz)[1] foi escolhida como referência para a escala igualmente temperada, a partir da qual se calcula a frequência fundamental de qualquer outra nota da escala.

Grande parte dos sistemas de pesquisa por consulta cantada utiliza o MIDI como representação de melodias, que é um padrão de comunicação digital consistindo em uma representação numérica das notas da escala igualmente temperada e de outras informações como os *onsets*, *offsets*, intensidade, variação de *pitch*[2] e sinais de controle. Nesse padrão, as frequências referentes às notas da escala são mapeadas em números MIDI,

$$n_{\text{MIDI}} = 69 + 12 \log_2 \left(\frac{f}{440} \right), \tag{2.2}$$

com $n_{\text{MIDI}} \in \{0, 1, 2, ..., 127\}$ representando notas de C_{-1} a G_9.

Portanto, toda melodia armazenada na base de músicas ou transcrita segue esse padrão. Para as melodias serem comparadas, realiza-se uma codificação dessas informações, que permita lidar com os problemas citados de inconsistências na melodia e diferenças de tonalidade, o que será abordado na Seção 2.4.

2.3 Transcrição do Sinal Cantado

Havendo uma base na qual se pode realizar as consultas, o sistema precisa obter a melodia cantada nas consultas. Para isso, são necessárias as seguintes etapas:

- Realizar um rastreamento da frequência fundamental F0 presente no sinal ao longo do tempo (ou função/série temporal de F0);

[1] A nomenclatura adotada na notação científica musical das notas de Dó a Si é: C, D, E, F, G, A, B; e o subscrito em cada nota é relativo à oitava, que é iniciada sempre nas notas C, e.g. após um B_0 vem um C_1.

[2] Ver definição na Seção 2.3.1.

- Detectar quando começa e termina cada nota cantada (segmentação);
- Realizar uma análise melódica das notas para ajustar suas alturas à escala de igual temperamento.

Algumas das formas mais usuais de realizar tais procedimentos serão abordadas ou citadas nesta seção. Como o objetivo deste trabalho não é o estudo e análise de tais técnicas, sugerimos a leitura da literatura indicada para maiores detalhes.

2.3.1 Estimação da Frequência Fundamental - *Pitch Tracking*

As notas musicais são associadas a um *pitch*, que é a altura percebida a uma determinada intensidade sonora. Por sua vez, o *pitch* é relacionado à frequência fundamental F0 de um sinal harmônico, mesmo que esta componente frequencial do sinal não esteja presente. Portanto, para estimar o *pitch* é necessário rastrear a F0 presente no sinal ao longo do tempo. Como sinais puramente harmônicos são periódicos, pode-se estimar a F0 pelo período T0 do sinal com a relação $F0 = 1/T0$. Mesmo que o sinal não seja periódico, havendo F0 é possível estimar um melhor candidato a T0.

Como as consultas serão realizadas por voz e grande parte da energia do sinal está contida na região de frequências até 4 kHz, é muito comum utilizar sinais amostrados com taxa de amostragem de 8 kHz. O uso de tal taxa traz a vantagem de diminuir a quantidade de informação processada, se comparado ao uso de taxas de alta qualidade, como 48 kHz, sem perder informação relevante para a estimação da função de F0.

Algumas das técnicas mais utilizadas para estimação de *pitch* serão apresentadas a seguir. Dentre elas, pode-se citar como as mais relevantes a *Autocorrelation Function* (ACF), a *Harmonic Product Spectrum* (HPS) [12], a Análise Cepstral (CEPS) [13] e a YIN [14]. A última será explorada com mais detalhes nesta seção por ser a técnica que apresenta melhores resultados e pelo fato de ter sido aplicada no sistema em que se baseia este trabalho.

YIN

O método conhecido por YIN [14] de estimação da função de F0 é composto pelos passos abordados nos próximos tópicos.

Função Diferença - FD

Consideremos um sinal periódico $x[n]$ cujo período é τ. Por ser periódico, uma réplica sua atrasada de τ é idêntica ao próprio $x[n]$, o que nos permite afirmar que

$$\sum_n x[n] - x[n-\tau] = 0. \tag{2.3}$$

O mesmo ocorre, portanto, para o quadrado dessa diferença, que avalia a energia da diferença dos sinais

$$\sum_{n}(x[n] - x[n-\tau])^2 = 0. \tag{2.4}$$

Como τ é o menor período do sinal $x[n]$, τ é o menor atraso que se pode aplicar ao sinal para que este se cancele quando subtraído dele mesmo. Porém, o mesmo cancelamento ocorre para múltiplos inteiros de τ. Sendo o sinal de voz quasi-periódico, quanto maior fosse o atraso aplicado, mais diferente seria o sinal de sua réplica atrasada. Como se deseja estimar a função de F0 ao longo do tempo, o sinal é dividido em *frames* e estes são janelados para suavização. Com isso, cada *frame* $x_w^l[k]$ é analisado separadamente, onde l é o índice do *frame* e k é o índice das amostras dentro dele. Considerando que há N amostras dentro de um *frame*, temos o cálculo da função diferença realizado, em cada *frame* l, como

$$d^l[\tau] = \sum_{k=1}^{N}(x_w^l[k] - x_w^l[k-\tau])^2. \tag{2.5}$$

Note que, mesmo que o sinal original $x[n]$ seja periódico, o que não é verdade para sinais de voz, os *frames* $x_w^l[k]$ não poderão mais ser periódicos, uma vez que têm comprimento finito e são janelados.

Função Diferença Normalizada - FDN

Assumimos que o sinal vozeado tem a característica de ser quasi-periódico. Assim, é necessário detectar quando o sinal é vozeado, a fim de, nesse caso, estimar-lhe a função de F0. Sinais não vozeados não têm forte periodicidade presente, uma vez que são gerados apenas pela passagem do ar pelo trato vocal, em contraposição aos sinais vozeados, cuja modulação do fluxo de ar realizada pelas pregas vocais gera um trem de pulsos de ar. Uma vez que não haja forte periodicidade, a FD desses sinais muito provavelmente não assumirá valores próximos de zero, embora tais sinais possam ter energia muito menor do que os sons vozeados.

Para determinar quando um bloco do sinal pode ser considerado como contendo sinal vozeado ou não vozeado, adota-se um limiar para a FD de forma que a presença de valores abaixo dele indique periodicidade suficiente para considerar o sinal como vozeado e, portanto, fará sentido atribuir-lhe uma F0.

Porém, surge a necessidade de se equiparar sinais de diferentes energias, de forma a comparar apenas a presença de periodicidade entre eles. Para isso, é realizada a normalização na FD, de forma a definir uma Função Diferença Normalizada. A

ideia central é normalizar os valores pela média acumulada da função até o atraso avaliado. Temos, então,

$$d'[\tau] = \begin{cases} 1 & \text{caso } \tau = 0 \\ \dfrac{\tau d[\tau]}{\sum_{i=1}^{\tau} d[i]} & \text{em caso contrário.} \end{cases} \tag{2.6}$$

Como o atraso nulo sempre irá zerar a saída da função e representa frequência fundamental infinita, não nos interessa calcular seu valor, motivo pelo qual se atribui à função o valor 1 em $\tau = 0$. Além disso, deseja-se estimar a F0 de uma voz humana, que não pode assumir qualquer valor. Por isso, é importante adotar limites inferiores e superiores na busca pela F0. Assim, há um τ mínimo, relacionado à maior F0 possível, e um τ máximo, relacionado à F0 mais baixa possível. A partir do τ mínimo, inicia-se uma busca pelo primeiro τ relativo a um mínimo local, pois haverá mínimos locais em T0 e nos múltiplos de T0.

Limiar de Decisão

Como foi dito, para saber se o bloco de sinal analisado é composto majoritariamente por sinal vozeado ou não vozeado, define-se um limiar de referência. Caso os mínimos da FDN estejam abaixo desse limiar, o sinal é suficientemente periódico para ser classificado como vozeado. Caso o contrário ocorra, o sinal é classificado como não vozeado. O limiar também é útil para encontrar o mínimo local mais provável de ser relativo a T0, pois os harmônicos do sinal causam periodicidade com intervalos menores que T0. Dependendo da resposta em frequência do sistema de gravação utilizado, a frequência fundamental do sinal pode estar atenuada em relação aos outros harmônicos, e isso pode gerar problemas para encontrar-se corretamente a F0.

Interpolação Parabólica

Uma vez que se esteja lidando com sinais discretos, não é possível estimar com precisão um período T0 que não seja múltiplo do período de amostragem do sinal. Por essa razão, após o sistema ter encontrado a melhor amostra candidata a T0, é realizado um processo de refinamento por interpolação da amostra escolhida e das outras em torno. A interpolação sendo realizada por uma aproximação de uma função parabólica já fornece resultados suficientemente acurados. Para isso, basta utilizar, além da amostra escolhida do mínimo local em $d'[\tau]$, as duas amostras que lhe estão mais próximas.

O algoritmo ainda possui um último passo que realiza uma busca entre as melhores estimativas locais que, por não fornecer melhora significativa, é relevado aqui

neste trabalho. O algoritmo implementado em [7] realiza alguns outros passos para melhorar a estimativa.

Como os mínimos presentes na FDN são, na maioria das vezes, relativos a sub-harmônicos, soma-se à FDN uma reta de inclinação positiva, de maneira a aumentar a probabilidade de o mínimo global ser relativo à frequência correta. E, ainda antes de se processar o sinal de voz, é realizada uma filtragem passa-faixa (50-500 Hz). A filtragem abaixo de 50 Hz é realizada com a intenção de diminuir os ruídos de baixa frequência gerados pela respiração do usuário próximo ao microfone; a filtragem acima de 500 Hz visa a reduzir os harmônicos e outros sons de médias/altas frequências do sinal de fala, tornando-o mais próximo de periódico e, com isso, melhorando a estimativa de F0 [15]. Na implementação, foram utilizados filtros de Butterworth de quarta ordem.

Pós-processamento

Após estimada a função de F0 ao longo dos blocos do sinal, ainda pode haver a presença de erros, que comumente não ocorrem por um grande número de blocos consecutivos. Na verdade, ocorrem na forma de picos (ruído impulsivo) na função de F0. Na implementação do sistema Tararira, estas etapas de pós-processamento não são realizadas, uma vez que, sem elas, já eram alcançados bons resultados; elas serão citadas aqui para a título de informação.

Uma forma de se corrigir tais erros é através do uso do filtro de mediana. O filtro de mediana realiza uma filtragem não linear ao longo de uma sequência da seguinte forma: reúne-se a cada amostra da entrada um número fixo de amostras contíguas de cada lado; ordena-se o conjunto de amostras resultante por seus valores; e entrega-se como saída a amostra central do conjunto ordenado. Em vez de realizar uma média de cada conjunto de amostras, o filtro de mediana escolhe a que representa o valor do meio dentre os valores existentes. Com isso, amostras que destoam muito das demais não irão alterar o resultado da filtragem, contanto que não aconteçam em muitas amostras.

Mesmo após a filtragem por mediana, é possível que o resultado ainda não esteja livre de erros. Se esse for o caso, pode-se utilizar ainda outros critérios de correção como o tempo de duração de um trecho cujos início e fim são caracterizados por transições abruptas. Caso o trecho seja curto demais (para isso é preciso determinar um limiar, e.g. 150 ms), o trecho é substituído pelo valor da amostra que o antecedeu.

Por fim, é desejável também limitar a faixa de valores possíveis para a F0, pelos limites naturais da voz humana. Limites comuns encontrados na literatura são 1000 Hz como máximo e 65 Hz como mínimo. Caso haja amostras com valores acima do máximo, dividem-se pelo menor múltiplo de 2 que gere um resultado dentro da faixa determinada de valores de F0; caso haja amostras com valores abaixo do

mínimo, considera-se que o sinal é não vozeado.

2.3.2 Detecção de *Onsets*

A próxima etapa a ser realizada para determinar a melodia cantada pelo usuário é encontrar os instantes em que as notas começam a ser emitidas (*onsets*). Como melodias são melhor determinadas pelos *onsets* das notas do que pelos instantes em que elas terminam (*offset*), as pessoas tendem a cantar com muito maior acerto o início das notas, não tendo boa precisão com relação à duração das mesmas. Por esse motivo, a determinação precisa do início das notas é muito mais relevante para sistemas de QBH do que o fim das mesmas.

Notas cantadas podem ter inícios muito bem definidos, com grande variação de energia, mas também podem ser "ligadas" com outras, dificultando a determinação de quando houve a transição. Para melhor determinar tais instantes, é possível combinar técnicas no domínio do tempo e no domínio da frequência. Em [16] há um estudo em que o autor compara diferentes técnicas de detecção de *onsets*, o qual referenciamos para maiores detalhes.

Dentre as técnicas de detecção de *onset* mais conhecidas, podemos citar *Phase Deviation* (PD) [17], *Weighted Phase Deviation* (WPD) [18], *Complex Domain* (CD) [17], *Spectral Flux* (PD) [16], e Derivada Relativa da Envoltória (DRE) [19]. O método escolhido pelo autor em [7, 10] para desempenhar a função de detecção de *onsets* é a DRE combinada com heurísticas que fazem uso da função de F0 estimada para detectar mudanças de notas que não apresentem variação significativa de envoltória.

Derivada Relativa da Envoltória (DRE)

A técnica conhecida por Derivada Relativa da Envoltória (DRE), apresentada em [19], é aplicada ao sinal no domínio do tempo, reunindo informação da envoltória do sinal em diferentes sub-bandas. O uso do processamento em sub-bandas pretende aproximar o sistema da forma como o ouvido humano processa os sinais acústicos. Há outras propostas na literatura a esse respeito, como [20].

O diagrama ilustrativo dos processos que envolvem o cálculo da DRE é mostrado na Figura 2.2. O sinal de entrada é dividido em bandas de frequência e, em seguida, a envoltória do sinal de cada sub-banda é extraída para que seja calculada a DRE. Após calculada a DRE, são aplicados critérios para a escolha dos picos candidatos a *onset* e, ao final, os candidatos de cada sub-banda são combinados para determinar os *onsets*.

Como já foi dito, a primeira etapa a ser realizada é a separação do sinal de entrada em sub-bandas por meio de um banco de filtros [19]. Em [7], o espectro

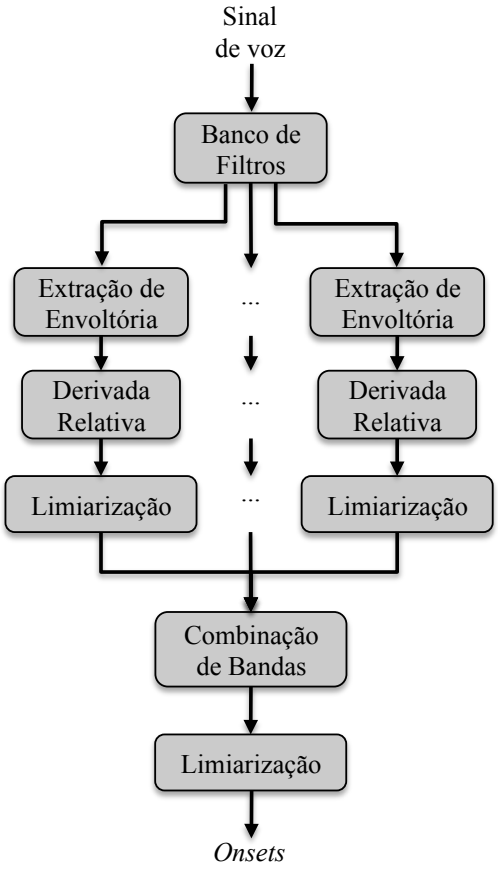


Figura 2.2: Diagrama de blocos do processamento da DRE.

é dividido em 6 faixas de oitava em oitava, utilizando filtros de Butterworth de ordem 6. A primeira sub-banda é obtida por um filtro passa-baixas com corte em 127 Hz; as quatro seguintes são obtidas por filtros passa-faixa em 127–254 Hz, 254–508 Hz, 508–1016 Hz e 1016–2032 Hz, respectivamente; e a última banda é obtida por um filtro passa-altas com corte em 2032 Hz. Essas especificações do processo de filtragem foram propostas em [20, 21].

Em seguida, para se obter a envoltória de cada faixa ao longo do tempo, é calculada a média do módulo do sinal ao longo de *frames* janelados do sinal

$$\mathrm{Env}(n) = \mathrm{AVG}_{k=0}^{L-1} \left(|x_w^b(n,k)| \right), \tag{2.7}$$

onde $x_w^b(n,k)$ é o sinal da sub-banda b contido no *frame* n janelado, k é o índice da amostra dentro do *frame*, L é o número de amostras de um *frame* e AVG é a função que calcula a média.

A implementação em [7, 10] se diferencia nessa etapa, por reduzir a taxa de amostragem dos sinais de cada sub-banda para 100 Hz e calcular a Env(n) por

convolução dos sinais filtrados com meia janela de Hann[3] de 100 ms. Isso proporciona uma avaliação da envoltória que considera instantes passados com uma espécie de fator de esquecimento ou decaimento, processo parecido com o que ocorre na audição humana.

De posse de Env(n), a Derivada Relativa da Envoltória é aproximada por

$$\mathrm{DRE}(n) = \frac{\mathrm{Env}(n) - \mathrm{Env}(n-1)}{\mathrm{Env}(n)}. \tag{2.8}$$

Dessa forma, o salto da envoltória entre *frames* estará sempre normalizado em relação ao instante n observado. Esse procedimento produz resultados mais acurados que a simples Derivada da Envoltória [19]. Com isso, obtêm-se picos candidatos a *onsets* em todas as bandas de frequência, cuja amplitude corresponde à variação de intensidade percebida. Os máximos locais são detectados e se tornam os picos candidatos. A próxima etapa é selecionar os melhores candidatos.

Para realizar essa seleção, são adotados três critérios. Primeiro, um limiar fixo é aplicado para preservar apenas os picos que se encontrem acima dele, atribuindo amplitude zero aos restantes. Note que, com este limiar, os finais das notas (*offsets*), que possuem derivada negativa da envoltória, são excluídos, restando apenas os *onsets*. Como existe baixa probabilidade de se cantar notas diferentes em um intervalo curto de tempo, o segundo critério consiste na exclusão dos picos que estiverem mais próximos que 50 ms de outro de maior intensidade. Além disso, o efeito de reverberação de ambientes cria réplicas atrasadas do sinal cantado, o que motiva o uso do último critério, que consiste em aplicar limiares dinâmicos. Para aplicar tais limiares, cria-se, a partir de cada pico, uma reta que decai até a metade do valor do pico em 200 ms. Os picos que estiverem abaixo de algum limiar criado por um pico são descartados. Na Figura 2.3 é possível observar um exemplo da aplicação desses limiares, onde há um pico (em vermelho) que foi excluído. Os picos estão ilustrados em linha cheia e os seus respectivos limiares dinâmicos, em linha tracejada.

Combinam-se os sinais gerados em cada sub-banda e, como dificilmente os picos em cada sub-banda estarão temporalmente coincidentes, os picos que se encontrarem a uma distância temporal menor que 50 ms entre si serão considerados como sendo relativos ao mesmo *onset*. Após agregados os picos, elege-se a mediana de cada conjunto de picos como o pico relativo ao *onset* correto. A amplitude atribuída ao pico selecionado é produto do valor do pico escolhido pelo número de picos reunidos pelo critério anterior de agregação.

Por fim, são utilizados limiares fixos e dinâmicos de forma similar ao que foi feito para cada sub-banda. Estipula-se um limiar de amplitude mínima, como antes, e outro para separar *onsets* fracos de *onsets* fortes, classificando-os dessa maneira.

[3]Comumente chamada de Hanning, devido à janela de Hamming.

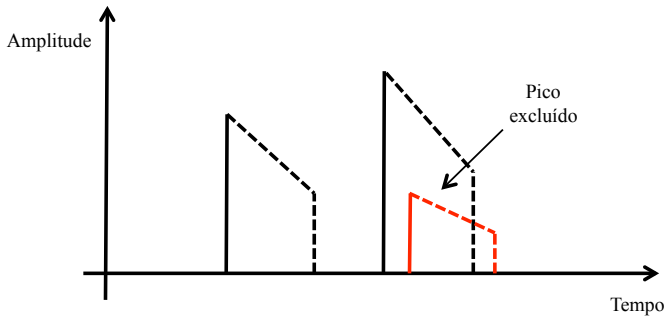


Figura 2.3: Exemplo de exclusão de um pico (em vermelho) pela aplicação do limiar dinâmico em sub-banda.

Quanto aos limiares dinâmicos, há a diferença de que o limiar dinâmico neste ponto, além de se estender para 200 ms depois, também se estende para 100 ms antes de cada pico, como ilustrado na Figura 2.4. Este critério segue também a motivação da impossibilidade ou inutilidade de se considerar como uma nota independente o que ocorreu tão próximo de uma nota genuína. Poderia ser até um efeito de modulação gerado pela palavra cantada, como alguma que comece com as letras "tr".

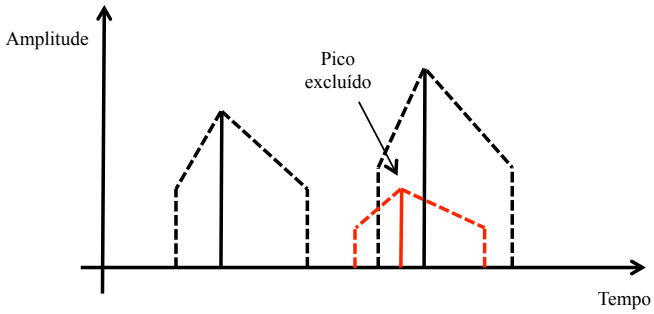


Figura 2.4: Exemplo de exclusão de um pico (em vermelho) pelo critério do limiar dinâmico aplicado após a combinação dos picos de cada sub-banda.

Ao fim desse processo, obtêm-se os picos relativos aos *onsets* fracos e fortes. Porém, há ainda transições suaves entre notas, que não apresentam variação significativa na envoltória. Por esse motivo, combina-se aos *onsets* encontrados uma análise da função de F0, a fim de analisar sua evolução ao longo do tempo e detectar variações que caracterizem mudança de nota.

Combinação Básica de Início de Notas e F0

Como os *onsets* fornecidos pelo algoritmo de segmentação não garantem que toda mudança de nota será considerada, é necessário fazer uso da informação frequencial que a função de F0 estimada provê, a fim de detectar as outras mudanças de notas.

Para evitar trechos da função temporal de F0 em que a intensidade do sinal é baixa e, por isso, a precisão da emissão da nota por parte do usuário é pior (e.g. início e fim de notas), são selecionados os trechos em que há amplitude significativa da envoltória do sinal de voz por meio de um limiar fixo. Com isso, definem-se janelas temporais somente dentro das quais a F0 poderá assumir valores diferentes de zero e cujos fins são considerados *offsets*. Ademais, sempre que uma nota inicia, seu *onset* é atribuído ao *offset* da anterior, caso este ainda não exista.

Em um primeiro estágio, consideram-se todos os *onsets* fortes como confiáveis e a eles se atribuem inícios de notas genuínas. Depois, é realizada uma busca, nos *onsets* fracos, por mudanças de *pitch*. Na maioria das vezes, os *onsets* fracos são relativos a modulações de amplitude de uma mesma nota; porém, há os casos em que se passa suavemente de uma nota para outra.

A validação desses *onsets* é feita por verificar se os mesmos são acompanhados por variações significativas de *pitch*. Como na escala igualmente temperada o menor intervalo significativo é de meio tom (cerca de 6% do valor em frequência), é utilizado um valor de mínima variação aceitável um pouco menor que 6%, uma vez que as variações entre notas podem ser lentas. Para determinar o intervalo, é calculada a mediana para algumas amostras da função de F0 antes e depois dos *onsets* fracos. Caso o intervalo ultrapasse o limiar estabelecido, o *onset* em questão é considerado como correto; se não, é descartado.

Por fim, é feita uma busca apenas na função de F0 por variações significativas de *pitch* que ainda não tenham sido consideradas. Essa tarefa não é simples, uma vez que deve-se separar efeitos de expressividade, como vibratos[4], de mudanças de nota genuínas. Todas as notas identificadas até o momento são processadas a fim de determinar-lhes as alturas correspondentes por meio do cálculo da mediana de seus valores de F0. O uso da mediana é extremamente favorável à determinação da altura na grande maioria dos casos de desvio, como vibrato, transições suaves, picos espúrios e instabilidade.

Obtidas as alturas das notas, para cada uma se verifica se há desvio da F0 maior que meio tom e, caso haja, por quanto tempo. Caso o desvio dure mais que um tempo determinado por um limiar, pode ser que a nota possa se dividir em outras. Se a nota não puder se dividir segundo estes critérios, avalia-se a nota seguinte; em caso contrário, são realizados e repetidos os seguintes passos:

- Identifica-se um trecho da função de F0 que se desvie mais de um semitom da altura determinada para o trecho e, se supera certa duração mínima (durMin), é considerado como um candidato a nova nota;
- É calculada, também pela mediana, a altura do candidato a nota e, caso a F0

[4]Expressão relacionada à variação periódica de altura em torno de uma nota.

se mantenha dentro de uma faixa de erro em relação à altura por um tempo maior que (durEstavel), o trecho é considerado uma nova nota.

Dessa forma, completa-se o processo de segmentação das notas cantadas.

2.3.3 Quantização de Alturas

Uma vez estimadas as alturas de cada trecho separadamente, é necessário adequá-las à escala igualmente temperada, de forma a corrigir imperfeições do que foi cantado e a permitir que se represente a melodia por notas, fazendo uso de uma representação simbólica de alto nível. Para isso, faz-se uma quantização pelo mapeamento de cada altura obtida nas notas de tal escala.

Como os usuários desse tipo de sistema em geral não são treinados para cantar precisamente em um tom de referência absoluto, é muito comum haver um desvio global em relação à escala igualmente temperada [22].

Dentre as diversas maneiras de resolver esse problema [22–25], foi implementada em [7, 10] a técnica presente em [22], em que é estimado o desvio mais frequente (em décimos de semitom) das notas cantadas para as notas da escala temperada e depois compensado tal desvio para quantizar as notas pelo número MIDI mais próximo.

Para estimar o desvio mais frequente, divide-se o semitom em 10 intervalos iguais e se arredondam as alturas estimadas para essa representação de décimo de semitom. Em seguida, calculam-se os desvios (para cima) das alturas para as notas absolutas da escala em décimos de semitom; contabiliza-se qual dos 10 intervalos de desvio é o mais frequente e selecionam-se as notas cantadas que obtiveram tais desvios. Com essas notas, calcula-se a média do desvio de suas alturas (não mais quantizadas) para as notas da escala igualmente temperada mais próximas por arredondamento para baixo.

Assim, nesse sistema, as melodias serão sempre representadas em alto nível por números MIDI inteiros. Em termos mais formais, as melodias são armazenadas como sequências de M notas $\mathbf{a} = (a_1, a_2, ..., a_M)^T,\ a_i \in \{0, 1, 2, ..., 127\}$. Como a função de F0 é calculada e representa de forma mais precisa o sinal cantado, esta também é utilizada em um estágio de comparação de melodias mais refinado, conforme será abordado nas Seções 2.4 e 2.5.

2.4 Comparação de Melodia

Sistemas de consulta por sinal cantado/cantarolado precisam lidar com as imprecisões presentes nas melodias cantadas pelos usuários, que têm grande chance de serem pessoas com pouca prática e técnica vocal. Portanto, quando da comparação

de melodias, é necessário buscar formas de tornar o sistema robusto às imprecisões no sinal cantado.

Há diversas formas de se realizar a comparação de melodias em sistemas de QBH. Podemos citar desde o uso de modelos ocultos de *Markov*, em inglês *Hidden Markov Models* (HMM) [26], Transformada Rápida de Fourier, em inglês *Fast Fourier Transform* (FFT) [27], Distância de Levenshtein [24, 28] e *Local Dinamic Time Warping* (LDTW) [29, 30]. Estes dois últimos métodos citados são mais usuais para se realizar a comparação de melodias e são os utilizados em [7, 10], motivo pelo qual serão detalhados aqui.

O método que calcula a Distância de Levenshtein é similar ao de Programação Dinâmica [31], amplamente utilizado para comparação de cadeias de caracteres (*strings*). Essa é a solução mais comum para comparação de melodias em sistemas de QBH, e proporciona resultados razoáveis, além de demandar baixo custo computacional. Porém, a transcrição do sinal cantado em notas introduz erros, por conta da discretização das alturas em notas da escala temperada. Esse método é abordado na Seção 2.4.2.

A outra técnica utilizada para se comparar melodias é o algoritmo conhecido por *Local Dinamic Time Warping* (LDTW), que faz uso da série temporal da F0 cantada. Alcança um nível de detalhes na comparação superior à técnica anterior, mas demanda grande poder computacional, além de dificultar a busca por subsequências de notas dentro de sequências maiores. Tal técnica é abordada na Seção 2.4.3.

2.4.1 Codificação das Notas

Considerando a sequência de notas composta por M notas $\mathbf{a} = (a_1, a_2, ..., a_M)^T$, $a_i \in \{0, 1, 2, ..., 127\}$, a codificação é realizada de maneira a gerar uma sequência invariante com a tonalidade com que a melodia foi cantada por caracterizar a melodia com relação aos saltos entre as notas, no lugar das próprias notas. Assim, define-se uma sequência pelas diferenças das notas $\bar{\mathbf{a}} = (a_2 - a_1, a_3 - a_2, ..., a_M - a_{M-1})^T$.

Com relação às informações temporais das notas, considera-se a duração de cada uma, independentemente do instante de *onset* na referência temporal absoluta. Uma vez que as notas estão ordenadas pelas posições nos vetores, basta armazenar as durações como $\mathbf{b} = (b_1, b_2, ..., b_M)^T$. Porém, como o andamento (quão rapidamente a melodia é cantada, tradicionalmente chamado de *Tempo*) pode variar da consulta para a referência e ainda assim serem ambas da mesma melodia, define-se uma sequência de durações invariante com o andamento por $\bar{\mathbf{b}} = (\bar{b}_1, \bar{b}_2, ..., \bar{b}_{M-1})^T, \bar{b}_i = b_{i+1}/b_i$ [32]. Dessa forma, a mesma melodia cantada em diferentes andamentos produzirá representações idênticas.

Porém, é muito comum, ao cantar, não se prestar muita atenção às durações

corretas de cada nota. Uma maneira de compensar esse problema é considerar apenas o intervalo entre *onsets*. Assim, o *onset* de uma nota é adotado como o *offset* da nota anterior.

Outra maneira de lidar com as inconsistências do ponto de vista temporal da execução da melodia pelo usuário é quantizar os intervalos por $\hat{b}_i = \text{round}(10 \log_{10} \bar{b}_i)$. Com isso, a sequência de durações temporais passa a ser $\hat{\mathbf{b}} = (\hat{b}_1, \hat{b}_2, \hat{b}_3, ..., \hat{b}_{M-1})^T$.

2.4.2 Distância de Levenshtein

Como já foi dito, o processo de comparação de sequências de notas é muito semelhante ao processo de comparação de *strings*, e o método de Programação Dinâmica é aplicado com sucesso para calcular a distância de edição entre sequências de notas e de durações [31]. A distância de edição pode ser entendida como o menor número de edições a serem realizadas em uma sequência para se chegar à outra que com esta é comparada.

O que é utilizado, porém, difere um pouco do cálculo da distância de edição de caracteres, sendo, na verdade, uma busca aproximada de padrões. São feitas modificações no algoritmo a fim de permitir que o padrão buscado possa começar e terminar em qualquer ponto da melodia de referência, uma vez que não são penalizadas omissões de notas da consulta antes do início ou depois do fim da ocorrência do padrão.

Considerando-se que a sequência das notas é um parâmetro muito mais discriminativo entre as melodias que suas durações, passa a ser mais relevante para computar a distância de edição. Isso pode ser observado na Equação 2.9, onde o algoritmo recursivo utilizado para computar a distância de edição $d_{i,j} \in \mathbb{R}$ é definido. Assim, calculam-se os elementos da matriz de edição $\mathbf{D} \in \mathbb{R}^{(M+1)\times(N+1)}$, onde M é o comprimento da sequência $\bar{\mathbf{a}}$ e N é o comprimento da sequência $\bar{\mathbf{a}}'$. O algoritmo recebe como condições iniciais $d_{i,0} = i$ e $d_{0,j} = 0$, com $0 \leq i \leq M$ e $0 \leq j \leq N$.

Define-se o elemento $d_{i,j}$ por

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + W_I, & & \text{(inserção)} \\ d_{i,j-1} + W_D, & & \text{(deleção)} \\ d_{i-1,j-1} + W_{SN}, & & \text{(substituição de nota)} \\ d_{i-1,j-1} + W_C, & |\bar{a}_i - \bar{a}'_j| < 2 \text{ e} & \\ & |\hat{b}_i - \hat{b}'_j| < 2 & \text{(coincidência)} \\ d_{i-1,j-1} + W_{SD} & |\bar{a}_i - \bar{a}'_j| < 2 & \text{(substituição de duração)}, \end{cases} \tag{2.9}$$

onde $\bar{a}_i$ e $\hat{b}_i$ são elementos relativos às sequências de notas e durações da consulta, respectivamente; os elementos $\bar{a}'_j$ e $\hat{b}'_j$ são similares aos primeiros, mas relativos à melodia comparada da base; e W_* é o peso atribuído a cada situação especificada

Tabela 2.1: Valores dos pesos W utilizados no algoritmo Distância de Levenshtein.

W_I	1
W_D	1
W_{SN}	1
W_C	-1
W_{SD}	0

pelo subscrito de W, cujos valores escolhidos estão dispostos na Tabela 2.1. O custo computacional relativo a esse procedimento pode ser avaliado como $O(MN)$, que é o custo associado ao cálculo da matriz $\mathbf{D}$.

Note que as últimas duas situações (coincidência e substituição de duração) são computadas submetidas a condições que levam em consideração informações das notas e de suas durações. A situação de coincidência é considerada quando os intervalos comparados são menores que um tom (dois semitons) e as durações relativas são mais próximas que 2. As durações não foram levadas em conta no último caso, onde notas consecutivas podem ter sido agregadas pelo próprio usuário ao cantar ou por uma falha no algoritmo de transcrição por não ter separado devidamente as notas.

Após o fim das recursões, atribui-se à distância de edição E o menor valor encontrado d_{ME} na última linha da matriz $\mathbf{D}$, onde o índice de E é onde o padrão comparado termina alinhado com o trecho de maior similitude. Para encontrar o melhor alinhamento entre eles, elemento por elemento, podendo pular ou agregar elementos, basta traçar o caminho de volta à segunda linha da matriz, a partir do elemento de índice (M, E) seguindo o menor custo. Note que a segunda linha é referente ao primeiro elemento do vetor de notas da consulta, já que as primeiras linhas e colunas são preenchidas na inicialização da matriz.

Após encontrada a distância de edição E, é calculada uma medida de similaridade S referente às sequências comparadas, normalizando a distância de edição entre 0 e 100, por

$$S = 100\,\frac{(M-1)-E}{2(M-1)}, \tag{2.10}$$

onde M denota o número de notas da consulta e E é a distância de edição entre as sequências. No caso de haver um padrão idêntico ao buscado, sempre haverá um caminho de custo -1, chegando à última linha com um elemento tendo acumulado o valor $E = -(M-1)$, que leva a $S = 100$. Caso não haja nenhuma similaridade entre as sequências, cada penalização somará 1 a cada iteração; com isso, tem-se $E = (M-1)$, que leva a $S = 0$.

Como resultado desse processo, fragmentos similares aos padrões das consultas são identificados e pontuados para todos as melodias presentes na base de dados.

2.4.3 Deformação Temporal Dinâmica Local (LDTW)

Outro algoritmo utilizado para comparar melodias é a Deformação Temporal Dinâmica Local, em inglês *Local Dinamic Time Warping* (LDTW), que realiza a comparação das séries temporais de F0. Esse procedimento é consideravelmente mais custoso que o anterior, uma vez que faz uso de um número muito maior de elementos para serem comparados. Mais especificamente no sistema utilizado [7, 10], para que o custo computacional seja reduzido, o LDTW é aplicado apenas nos trechos de maior similitude apontados pelo algoritmo descrito anteriormente, inclusive porque esse algoritmo pressupõe que sejam definidos o início e o fim dos trechos que serão comparados. Este é outro motivo para que o LDTW seja aplicado posteriormente à Distância de Levenshtein.

O método LDTW consiste em comparar a sequência de F0 da consulta com a que se obtém a partir das notas MIDI dos elementos armazenados na base. Antes de computar a comparação, o andamento do trecho da música utilizada da base é modificado a fim de igualá-lo com a consulta em duração. Em seguida, as sequências de F0 são normalizadas, resultando em média nula e desvio padrão unitário. Assim as sequências de F0 podem ser comparadas ponto a ponto, permitindo certa deformação temporal dentro das sequências. Para evitar comparações entre sequências muito diferentes em andamento, descartam-se os candidatos da base cujo trecho deveria ser modificado em andamento com fator maior que 2 ou menor que 1/2.

A comparação é realizada com algoritmo semelhante à Distância de Levenshtein. Considerando-se as sequências x e y, ambas com L elementos, o cálculo da k-ésima distância é realizado de maneira recursiva por

$$d_{i,j}^W = \begin{cases} |x_i - y_j|^2 + \min \begin{cases} d_{i-1,j-1}^W \\ d_{i,j-1}^W \\ d_{i-1,j}^W \end{cases} & |i-j| \leq k \\ \\ \infty & |i-j| > k, \end{cases} \tag{2.11}$$

e a matriz $\mathbf{D}^W \in \mathbb{R}^{L \times L}$ é formada por tais elementos, com $0 \leq i \leq L$ e $0 \leq j \leq L$. A matriz $\mathbf{D}^W$ deve ser inicializada com $d_{1,j}^W = |x_1 - y_j|^2$, onde $j \in [1, k]$ e $d_{i,1}^W = |x_i - y_1|^2$, onde $i \in [1, k]$.

A distância no LDTW é calculada por $d_{min}^W = \sqrt{\min\{d_{L,j}^W, d_{i,L}^W\}}$ com $i, j \in [L-k+1, L]$. Note que a máxima distorção temporal permitida é de k amostras. Se $k = 0$, esse método será análogo ao cálculo da distância euclidiana. Convém notar que a implementação desse método também é semelhante ao algoritmo de Programação Dinâmica, mas restrita a uma banda diagonal de largura $2k + 1$ da matriz $\mathbf{D}^W$.

2.5 Sistema Tararira

Nas seções anteriores foram detalhadas todas as etapas do processamento realizado no sistema Tararira [7, 10], utilizado nesse trabalho. Por isso, aqui será feito apenas um breve resumo do funcionamento global do mesmo.

Na Figura 2.1, há um diagrama de blocos que descreve o sistema. O primeiro conjunto de blocos é responsável pela transcrição da consulta feita pelo usuário. O contorno do F0 é extraído utilizando-se a técnica YIN e o sinal é segmentado em notas através da análise da envoltória do sinal em sub-bandas. Em uma próxima etapa, são estimadas as alturas e as notas são quantizadas, com o fim de se transcrever a consulta para o formato MIDI. Importa ressaltar que, pela forma como foi construída a etapa de transcrição da consulta, o sistema é capaz de lidar com sinais cantarolados, cantados, ou mesmo assobiados.

Em seguida, o sinal é enviado ao segundo conjunto de blocos, responsável pela consulta às músicas do banco. Ali, o sinal passa por uma codificação, mudando sua representação de notas para intervalos em altura[5] e de instantes de tempo para razões entre durações das notas. Note que as melodias guardadas na base do sistema devem, assim como as melodias das consultas, ser codificadas.

O sistema combina as duas técnicas de comparação de melodias citadas, Distância de Levenshtein e LDTW, fazendo uso da primeira para gerar um ranque global da base de dados e da segunda para reordenar os primeiros colocados da lista, melhorando os resultados para os candidatos mais relevantes. Esse procedimento poupa processamento, por destinar o uso do LDTW apenas para os trechos mais importantes dos candidatos mais importantes.

O sistema provê uma busca eficiente, mas ainda realiza a comparação de melodias por busca exaustiva, sendo obrigado a passar por todas as músicas da base, o que pode se tornar proibitivo para bases de larga escala. Esse problema pode ser mitigado com técnicas de *hashing*[6] [5, 33].

[5] A codificação utilizada é uma representação de intervalos quantizados em 3 níveis: se a altura variou mais de 1 tom para cima ou para baixo ou se ficou dentro desse limite.

[6] Técnicas de *hashing* possibilitam buscas rápidas, ao associarem chaves de pesquisa a valores numéricos armazenados em tabelas, em substituição a uma busca exaustiva.

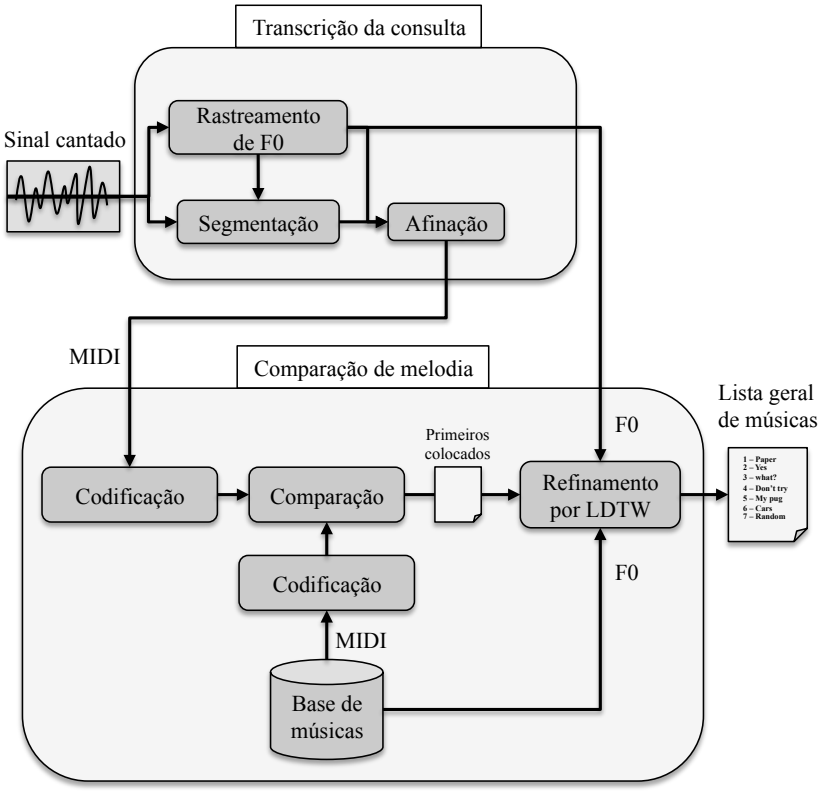


Figura 2.5: Diagrama de blocos ilustrando os princípios de funcionamento do sistema Tararira.

Capítulo 3

Transcrição Automática de Bancos de Músicas

Neste capítulo, serão abordados dois métodos de extração de melodia em sinais polifônicos da literatura, os quais foram adotados com a finalidade de testar a viabilidade do uso de tais sistemas para a geração automática de bases de músicas para sistemas de QBH.

3.1 Sistemas de Transcrição Automática de Bases de Músicas

3.1.1 GPA-UdelaR

No contexto de transcrição automática de melodias presentes em sinais polifônicos, o grupo de pesquisa GPA-UdelaR[1] desenvolveu um sistema [34–37] com essa finalidade, utilizando-o como ferramenta para geração automática de um banco de músicas em [8]. Tal sistema tem por objetivo extrair e transcrever somente o que é detectado por ele como voz principal, excluindo quaisquer instrumentos e vocais secundários. Como forma de aferição da eficácia desse procedimento, foram transcritas algumas músicas encontradas na base composta por músicas da banda *The Beatles*, conforme foi discutido na Seção 4.2, e comparou-se o desempenho [8] do sistema Tararira nas consultas relacionadas a tais músicas tomando como referência os MIDIs da base original e os MIDIs gerados pelo sistema de transcrição.

Assim, como os resultados reportados em [8] apontam para a possibilidade do uso de tal sistema para a geração automática de bases de músicas para uso em sistemas de QBH, decidiu-se aqui adotar tal sistema de transcrição para realizar um teste

[1] Grupo de Procesamiento de Audio: `http://iie.fing.edu.uy/investigacion/grupos/gpa/QBH/`

da mesma natureza, mas agora também com a base de músicas brasileiras, o que estressaria o sistema de maneira diferente, pelo fato de as características dos sinais cantados nessa base serem diferentes (inclusive por diferenças culturais). Como a base de músicas brasileiras contém apenas trechos de músicas, e menos músicas que a base *The Beatles* completa, foi mais fácil transcrever toda a base, propiciando um panorama geral mais completo da versatilidade do sistema.

A Figura 3.1 ilustra, com um diagrama de blocos, os princípios de funcionamento do sistema, que chamaremos de GPA-UdelaR, em que há três estágios principais: Separação, Classificação e Transcrição. Na entrada, o sistema recebe um sinal polifônico que contenha uma voz principal. Primeiramente, o sinal é processado pelo estágio de separação, em que os contornos de frequência são identificados e separados [34, 36]. No segundo estágio, os contornos são classificados para que sejam selecionados os que se caracterizarem como relativos a um sinal de voz [37]. Após a classificação, o sinal é reconstituído no domínio do tempo (apenas com os trechos classificados como voz) e transcrito com o mesmo procedimento utilizado para as consultas descrito na Seção 2.3. Os primeiros dois estágios serão descritos nas próximas seções.

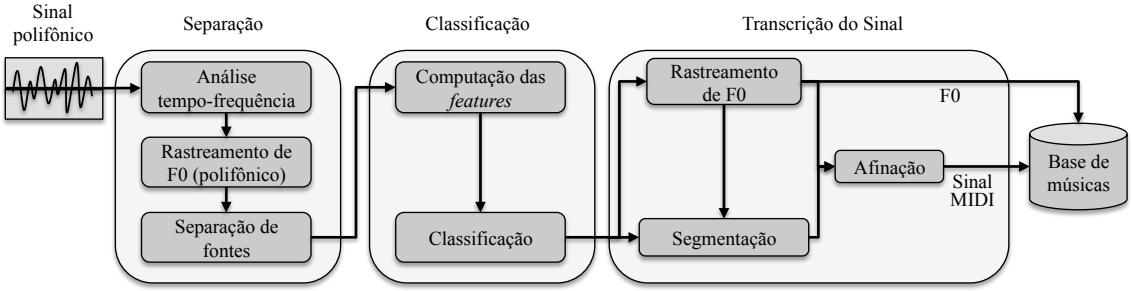


Figura 3.1: Diagrama de blocos ilustrando os princípios de funcionamento do sistema de transcrição de melodia GPA-UdelaR.

Separação de Sinais Harmônicos

A análise tempo-frequência empregada para a tarefa de separação de sons Harmônicos é baseada em [34], em que se aplica a *Fan Chirp Transform* (FChT) [38] para sinais musicais polifônicos. A FChT de um sinal $x(t)$ janelado por uma função $w(t)$ pode ser definida como

$$X_w(f,\alpha) = \int_{-\infty}^{\infty} x(t)w(\phi_\alpha(t))\phi'_\alpha(t)e^{-j2\pi f\phi_\alpha(t)}dt, \tag{3.1}$$

onde $\phi_\alpha(t) = t + \frac{1}{2}\alpha t^2$ é a função de deformação temporal e α é o parâmetro da taxa de variação da frequência instantânea do *chirp* de análise [34]. Note que, com a mudança de variável $\tau = \phi_\alpha(t)$, essa transformada é análoga à Transformada

de Fourier da versão temporalmente deformada de $x(t)$, o que permite realizar os cálculos de maneira otimizada por meio do algoritmo FFT. Com o sinal $X_w(f, \alpha)$, é calculada uma representação chamada F0grama [34], que consiste em um mapa da evolução temporal da proeminência dos contornos de altura dos sinais presentes na mistura $x(t)$ ao longo dos *frames*.

A função de saliência (ou proeminência) de uma frequência fundamental F0 pode ser obtida pela soma do logaritmo do espectro nas posições em que se encontram os harmônicos de F0:

$$\rho(\mathrm{F0}, \alpha) = \frac{1}{N_\mathrm{h}} \sum_{h=1}^{N_\mathrm{h}} \log |X(h \cdot \mathrm{F0}, \alpha)|, \tag{3.2}$$

onde h é o índice do harmônico e N_h é o número total de harmônicos considerados. Para encontrar os contornos de F0 do sinal polifônico, diferentemente do que é realizado na transcrição das consultas, são realizados procedimentos baseados em clusterização não supervisionada [36] de picos do F0grama. A ideia basicamente consiste em buscar os picos em cada *frame* que formam, quando clusterizados, linhas de F0 variando consistentemente ao longo do tempo.

De posse das diversas linhas de F0 encontradas, cada emissão é separada das outras por meio da aplicação de um conjunto de filtros passa-banda centrados nas frequências dos harmônicos, no domínio da FCh, desde de F0 até F0$\times N_\mathrm{h}$. Em seguida, os sinais separados são sintetizados novamente no domínio do tempo por meio da FChT inversa.

Classificação de Sinais Cantados

No estágio de classificação dos sinais, cada trecho obtido é classificado segundo os procedimentos propostos em [37]. Em linhas gerais, é feita uma classificação dos sinais por meio de atributos (*features*) extraídos destes.

Sinais cantados possuem, de maneira geral, características que os permitem ser distinguidos de instrumentos musicais. Alguns dos principais fatores que caracterizam a voz cantada são relacionados à modulação de altura em baixas frequências. O vibrato e o glissando[2], por exemplo, são expressões muito comuns em sinais cantados. Como outros instrumentos também fazem uso de tais recursos, é preciso combinar o atributo de modulação em frequência com outras informações.

Um dos melhores conjuntos de atributos para detecção de sinais de voz em sinais polifônicos são os *Mel-frequency Cepstral Coefficients* (MFCC) [39]. A implementação do cálculo do MFCC foi realizada baseada em [40], tal como disponibilizado na *internet*[3]. Para cada trecho são computados os coeficientes MFCC ao longo de *frames*. Em seguida, são computados a mediana e o desvio padrão dos coeficientes

[2] O glissando pode ser definido como uma transição contínua entre alturas.

[3] `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`

obtidos de cada *frame*, fazendo assim uma integração temporal das características observadas. É incluída também a derivada (diferença contígua) dos coeficientes, de forma a considerar a informação temporal presente (evolução dos coeficientes ao longo dos *frames*).

Para descrever as características da evolução temporal da F0, considera-se o contorno como um sinal no tempo discreto F0$[n]$. Com isso, é feita uma análise espectral de cada *frame* desse sinal por meio da Transformada Discreta de Cosseno (DCT), obtendo-se, para cada *frame*, uma sequência no domínio espectral $\tilde{c}[k] = \text{DCT}\{\text{F0}[n]\}$. Para combinar os coeficientes dos *frames* é feita, para cada coeficiente, a média entre o valor máximo e a mediana, considerando todos os *frames* do contorno, que teve seu valor médio zerado antes de ser analisado a fim de desconsiderar-se sua altura.

Dos coeficientes $\tilde{c}[k]$ obtidos dessa maneira, são extraídos dois atributos [35]. O primeiro é o LFC (*Low Frequency Content*), que é relacionado à informação em baixas frequências, obtido por

$$\text{LFC} = \sum_{k=1}^{k_\text{l}} |\tilde{c}[k]|, \tag{3.3}$$

onde k_l é o índice do coeficiente relativo a 20 Hz. Esse atributo contém as informações das modulações de baixa amplitude citadas.

O segundo atributo expressa a razão entre a soma dos coeficientes relativos às baixas frequências e a soma dos coeficientes relativos às frequências altas:

$$\text{BCR} = \frac{\sum_{k=1}^{k_l} |\tilde{c}[k]|}{\sum_{k=k_l+1}^{K} |\tilde{c}[k]|}, \tag{3.4}$$

onde N é o número total de amostras do espectro e BCR é relativo a *Band Content Ratio*. Esse atributo é computado pelo fato de sinais de F0 de voz cantada não apresentarem componentes de alta frequência proeminentes.

São extraídos outros dois atributos diretamente do sinal F0$[n]$. Um deles é relativo à variação máxima de frequência:

$$\Delta\text{F0} = \max\{\text{F0}[n]\} - \min\{\text{F0}[n]\}. \tag{3.5}$$

O outro computa a média da saliência do contorno de F0 em questão:

$$\Gamma\text{F0} = \text{AVG}\{\rho(\text{F0}[n])\}. \tag{3.6}$$

Esse atributo não indica somente o quão proeminente está a fonte, mas também privilegia sinais que possuem muitos harmônicos, como sinais de voz [34]. Adicio-

nalmente é feita uma ponderação que privilegia a faixa de frequência na qual é mais provável haver F0 relacionada a voz, proposta em [34].

A base de sinais utilizada para o treinamento do sistema contém mais de 2000 arquivos de áudio e compreende sinais de voz cantada e de instrumentos musicais comumente encontrados em música popular. Cada arquivo passa pelos procedimentos explicitados acima (de separação dos sinais) e todos os atributos são computados para os contornos de F0 extraídos. Assim, obtém-se um conjunto de treinamento com 13598 sinais (vários sinais gerados por arquivo, por serem estes separados em trechos) com os seus respectivos atributos, sendo que metade dos sinais são relativos a voz cantada e a outra metade são relativos a instrumentos [8]. Todos os sinais foram classificados como "voz" ou "não voz" com um classificador SVM com um *kernel* RBF gaussiano [41] foi treinado por meio do *software* Weka[4]. Uma vez treinado, o sistema passa a classificar quaisquer outros sinais, sendo esse o último procedimento do estágio de classificação.

3.1.2 Melodia

Melodia [9] é um sistema[5] desenvolvido para automaticamente transcrever a melodia principal de um sinal polifônico em uma série temporal de sua frequência fundamental (F0). Diferentemente do sistema GPA-UdelaR, esse não faz distinção de voz ou de instrumento, procurando apenas identificar qual melodia está em destaque na música.

O sistema está ilustrado de forma simplificada em um diagrama de blocos na Figura 3.2. Note que o sistema foi adaptado para gerar arquivos do formato utilizado no sistema de QBH, isto é, arquivos MIDI. Para isso, o sistema foi acoplado a uma versão simplificada do mesmo transcritor utilizado para a transcrição das consultas, que utiliza o F0 fornecido para reconhecer e separar as notas musicais presentes, mas sem a análise de envoltória do sinal de áudio.

Há quatro estágios principais pelos quais passam os sinais polifônicos analisados. Primeiramente, há um estágio de extração senoidal, em que é realizada uma análise tempo-frequencial para, em seguida, ser extraída a função de saliência do sinal. Com isso, são identificados os contornos de frequência e, por fim, o sistema seleciona, dentre os contornos obtidos, os que mais se destacam e parecem formar a melodia principal da mistura.

[4] `http://www.cs.waikato.ac.nz/ml/weka/`

[5] A versão plug-in está disponível gratuitamente para ser baixada na *internet* em `http://mtg.upf.edu/technologies/melodia`

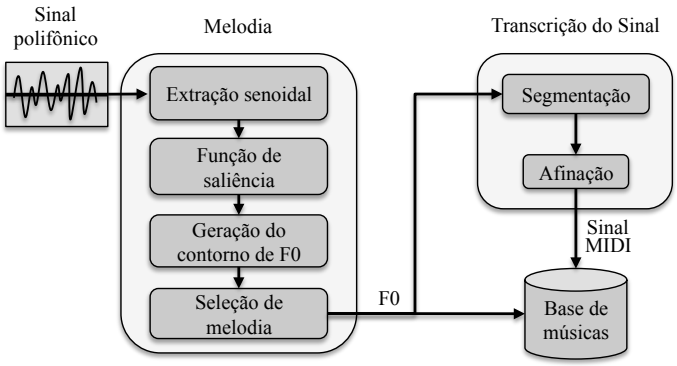


Figura 3.2: Diagrama de blocos ilustrando os princípios de funcionamento do sistema de extração de melodias Melodia.

Extração Senoidal

O estágio de extração senoidal consiste em três etapas: filtragem, transformação tempo-frequencial e correção de amplitude e frequência.

A primeira etapa consiste em aplicar uma compensação em frequência com a finalidade de emular a sensibilidade da audição humana, atenuando frequências altas e baixas. O filtro de igual audibilidade cumpre esse papel, fazendo com que o sinal analisado seja modificado de maneira a se assemelhar ao sinal percebido. Com isso, a faixa de frequência na qual é mais provável haver melodias principais é privilegiada. A implementação desse filtro é realizada por uma aproximação do inverso de uma medida de audibilidade [42] média por um filtro IIR de ordem 10 em cascata com um filtro de Butterworth passa-altas de segunda ordem.

Após a filtragem, o sinal é transformado para o domínio da frequência por meio da Transformada de Fourier de Tempo Curto, em inglês *Short-Time Fourier Transform* (STFT), dada por

$$X_l[k] = \sum_{n=0}^{M-1} w[n]x[n + lH]e^{-j\frac{2\pi}{N}kn}, \tag{3.7}$$

onde $x[n]$ é o sinal polifônico analisado, $w[n]$ é a janela de suavização aplicada ao *frame* (no caso, a janela de Hann), l é o índice do *frame*, $k = \{0, 1, ..., N-1\}$ é o índice da amostra em frequência, N é o número de amostras da DFT, M é o comprimento do *frame* e H é o salto em amostras dado entre *frames* consecutivos [9]. Para cada *frame* $X_l[k]$, são identificados seus respectivos picos p_i com índice i.

Como a resolução frequencial é limitada à resolução das amostras da STFT, o que pode resultar em erros relativamente grandes para baixas frequências se não forem considerados valores de frequência entre amostras, é aplicada uma correção de frequência e amplitude, com a finalidade de desfazer tal quantização. Isso é atingido por meio da fase do espectro $\phi_l[k]$, que auxilia na computação das frequências instantâneas do sinal relacionadas aos picos e de suas respectivas amplitudes [43].

A frequência instantânea $\hat{f}_i$ de um pico p_i relativo a uma amostra k_i do espectro é computada a partir da diferença de fase $\Delta(k)$ entre diagramas de fase de dois *frames* consecutivos utilizando o método *phase vocoder* [44]:

$$\hat{f}_i = (k_i + \kappa[k_i])\frac{f_s}{N}, \tag{3.8}$$

onde o desvio $\kappa[k_i]$ da amostra k_i é calculado por

$$\kappa[k_i] = \frac{N}{2\pi H}\Psi\left(\phi_l[k_i] - \phi_{l-1}[k_i] - \frac{2\pi H}{N}k_i\right), \tag{3.9}$$

e Ψ é uma função que mapeia a fase para valores entre $-\pi$ e $+\pi$.

A magnitude instantânea $\hat{a}_i$ é calculada a partir da magnitude espectral do pico $|X_l[k]|$ e do desvio da amostra $\kappa[k_i]$ por

$$\hat{a}_i = \frac{1}{2}\frac{|X_l[k_i]|}{W(\frac{M}{N}\kappa[k_i])}, \tag{3.10}$$

onde W é o *kernel* da janela de Hann.

Função de Saliência

No próximo estágio se calcula a função de saliência utilizando os picos do espectro corrigidos no estágio anterior. Os picos dessa função formarão os contornos de F0 candidatos à melodia principal que se deseja calcular.

A saliência de uma dada frequência é calculada pela soma das energias ponderadas de seus harmônicos. Esse procedimento é similar ao adotado em [45]. Porém, apenas os picos p_i do espectro são usados na soma, a fim de retirar informação frequencial indesejada e utilizar a correção de frequência citada. Com isso, a função de saliência tende a ganhar em acurácia.

A função de saliência empregada cobre uma faixa de frequências de 55 Hz a 1,76 kHz, que é quantizada em *bins* de *pitch* $b = \{1, 2, ..., 600\}$ equivalentes a 10 centésimos de semitom cada; assim, não é feita uma divisão linear do espectro, mas sim geométrica. Para uma dada frequência $\hat{f}$ em Hz, o *bin* correspondente é dado por

$$B(\hat{f}) = \left\lfloor \frac{1200\log_2(\frac{\hat{f}}{55})}{10} + 1 \right\rfloor. \tag{3.11}$$

Para cada *frame*, é calculada a função de saliência $S(b)$ por meio dos I picos p_i (com frequência $\hat{f}_i$ e magnitude $\hat{a}_i$) encontrados na etapa de extração senoidal da seguinte forma [9]:

$$S(b) = \sum_{h=1}^{N_\mathrm{h}}\sum_{i=1}^{I} e(\hat{a}_i)g(b, h, \hat{f}_i)(\hat{a}_i)^\beta, \tag{3.12}$$

onde β é um parâmetro de compressão da magnitude, N_h é o número total de harmônicos considerados, h é o índice do harmônico, $g(b, h, \hat{f}_i)$ é a função que define os pesos escolhidos e $e(\hat{a}_i)$ é a função que aplica um limiar de magnitude dada por

$$e(\hat{a}_i) = \begin{cases} 1 & \text{caso } 20\log_{10}(\hat{a}_M/\hat{a}_i) < \gamma, \\ 0 & \text{em caso contrário,} \end{cases} \tag{3.13}$$

onde $\hat{a}_M$ é a magnitude do maior pico espectral do *frame* em análise e γ é a maior diferença (em dB) entre $\hat{a}_i$ e $\hat{a}_M$.

A função de pesos $g(b, h, \hat{f}_i)$ define os pesos atribuídos aos picos p_i quando são considerados como o h-ésimo harmônico de um *bin* b,

$$g(b, h, \hat{f}_i) = \begin{cases} \cos^2(\delta\frac{\pi}{2})\alpha^{h-1} & \text{caso } |\delta| \leq 1, \\ 0 & \text{caso } |\delta| > 1 \end{cases}, \tag{3.14}$$

onde $\delta = |B(\hat{f}_i/h) - b|/10$ é a distância em semitons entre a frequência do harmônico $\hat{f}_i/h$ e a frequência central do *bin* b, e α é o parâmetro de peso do harmônico. Note que a primeira condição significa que cada pico contribui, não somente para um *bin* da função de saliência, mas também para os *bins* em torno deste, com uma distribuição cossenoidal ($\cos^2$). Com isso, evitam-se os erros causados por inarmonicidade[6] e pela quantização realizada [9]. Após um processo de otimização de tais parâmetros [46], chegou-se aos valores ótimos de: $\alpha = 0{,}8$, $\beta = 1$, $\gamma = 40$ e $N_\mathrm{h} = 20$.

Geração de Contornos de F0

Uma vez obtida a função de saliência, os picos da sequência de *frames* são agrupados usando-se um conjunto de características baseadas no sistema auditivo para criar os contornos de F0 coerentes ao longo dos *frames*. Para isso, a primeira etapa consiste em realizar dois procedimentos de seleção dos picos presentes, removendo os picos menos proeminentes.

Na primeira etapa é realizada uma filtragem em todos os *frames* retirando os picos que estejam abaixo de um limiar $\hat{a}_M \cdot \tau_+$, onde $\tau_+ < 1$ é uma fração da maior magnitude encontrada no *frame* avaliado. Em seguida, são calculados a média μ_s e o desvio padrão σ_s da saliência de todos os picos em todos os *frames*. São removidos, então, todos os picos cuja saliência esteja abaixo do limiar $\mu_s - \tau_\sigma \cdot \sigma_s$, onde τ_σ representa um grau de tolerância de desvio para baixo da média. Por otimização utilizando sinais de referência [46], $\tau_+ = \tau_\sigma = 0{,}9$.

[6] Quando as frequências dos harmônicos que compõem um determinado som não são precisamente múltiplos de sua frequência fundamental.

Todos os picos que passaram pela seleção são classificados como elementos do conjunto $\mathcal{S}^+$, enquanto os que foram descartados se tornam elementos de $\mathcal{S}^-$. Com isso, o primeiro contorno de F0 (todo contorno será formado por um conjunto de picos, assim como o são $\mathcal{S}^+$ e $\mathcal{S}^-$) é iniciado por atribuir a ele o elemento de maior valor de saliência de $\mathcal{S}^+$, retirando tal pico de $\mathcal{S}^+$.

Assim, inicia-se um processo recursivo em que, para cada pico, busca-se um sucessor (do *frame* seguinte) em $\mathcal{S}^+$ cuja diferença em frequência seja menor que 80 centésimos de semitom, ou 8 *bins*. Caso algum pico seja escolhido, este último também é removido do conjunto $\mathcal{S}^+$ e atribuído ao contorno em questão. Porém, podem haver *frames* em que a saliência de algum pico não tenha sido suficientemente alta para fazê-lo ser alocado em $\mathcal{S}^+$, o que pode gerar intervalos em que picos da melodia principal se percam, gerando separações de um contorno. Por isso, o algoritmo permite que, uma vez não havendo mais picos para serem alocados de $\mathcal{S}^+$ para o contorno em questão, sejam buscados picos em $\mathcal{S}^-$ por alguns *frames* (100 ms). Quando não houver mais nenhum pico candidato, o mesmo processo é realizado para trás no tempo (*frames* anteriores), partindo do primeiro pico alocado no contorno. Ao fim desse processo, o contorno é salvo e o mesmo procedimento é reiniciado com os picos que restaram em $\mathcal{S}^+$, até que se esvazie $\mathcal{S}^+$.

Uma vez criados os contornos, a última etapa desse estágio consiste em caracterizá-los para, depois, classificá-los e determinar quais são os que pertencem à melodia principal presente. Para isso, é utilizado o seguinte conjunto de atributos baseados na altura, no comprimento e na saliência dos contornos:

- Altura média $C_{\bar{p}}$ do contorno;
- Desvio padrão de altura C_{σ_p} do contorno;
- Saliência média $C_{\bar{s}}$ dos picos do contorno;
- Saliência total $C_{\sum s}$ dos picos do contorno;
- Desvio padrão da saliência C_{σ_p} dos picos do contorno;
- Comprimento C_l do contorno;
- Presença de vibrato C_v (variável booleana), aferida por identificar picos proeminentes em baixas frequências (5-8 Hz) no espectro do contorno, após extraída sua média [47];

Seleção de Melodias

No estágio de seleção de melodias, os procedimentos realizados se baseiam em um problema de filtragem, ou remoção, de contornos, de maneira a descartar os que não

se caracterizarem como melodias. Para isso, são realizadas 3 etapas: detecção de *voicing*, minimização de erro de oitava e seleção da melodia final.

A etapa de detecção de *voicing* reúne informações dos atributos para determinar quais os contornos que não se caracterizam como melodias. O primeiro atributo utilizado é a saliência média $C_{\bar{s}}$. Realiza-se a média desse atributo para todos os contornos, obtendo-se $\bar{C}_{\bar{s}}$. Por observação, concluiu-se que configurar um limiar ligeiramente abaixo de $\bar{C}_{\bar{s}}$ para selecionar os contornos faz com que seja excluída a maioria dos contornos não melódicos, havendo pouca perda de contornos de melodia. Define-se tal limiar τ_v por

$$\tau_v = \bar{C}_{\bar{s}} - \nu \cdot \sigma_{C_{\bar{s}}}, \tag{3.15}$$

onde ν é um parâmetro que define a margem proporcional a $\sigma_{C_{\bar{s}}}$ de tolerância para aceitar-se determinado contorno como melodia; quanto maior for o valor de ν, mais contornos (possivelmente não melódicos) serão selecionados.

Unidos ao critério do limiar τ_v, há outros dois critérios que fazem uso de outros atributos para impedir que contornos de melodia com baixa saliência sejam excluídos, caso pelo menos um dos dois seja satisfeito. Os critérios baseiam-se na observação de que a grande maioria dos contornos que possuem grande modulação em baixas frequências são contornos de melodia. Com isso, caso o contorno apresente vibrato (C_v = verdade) ou um desvio de *pitch* $C_{\delta_p} > 40$ centésimos (equivalente a 4 *bins*), este contorno não será excluído.

A segunda etapa da seleção de melodias consiste em buscar corrigir os erros de oitavas, que ocorrem quando se seleciona uma frequência múltipla de F0 como frequência fundamental. Para identificar se há oitavas duplicadas, os contornos que apresentam sobreposição temporal são comparados *frame* por *frame*. Caso a distância média entre os contornos esteja dentro da faixa de 120 ± 5 *bins* (o equivalente a uma oitava mais ou menos metade de um semitom), os contornos são considerados como duplicação de oitava e será preciso escolher o que melhor representa a melodia buscada.

O sistema Melodia utiliza dois princípios básicos para a escolha dos contornos, sendo o primeiro a comparação das trajetórias, como já foi citado, e o segundo uma comparação com os contornos adjacentes no tempo. Esse segundo princípio parte de duas suposições: (i) o contorno correto tende a ter maior valor de saliência, uma vez que os pesos utilizados no cálculo da saliência foram otimizados para esse propósito; e (ii) melodias tendem a ter uma trajetória contínua em altura, de maneira a evitar grandes saltos [48].

Pode-se resumir o procedimento adotado pelo sistema em oito passos a serem realizados iterativamente [46]:

1. Calcular o *pitch* médio $\bar{p}(t)$ por *frame*, ponderado pela saliência total $C_{\sum s}$ do

contorno a que pertence a amostra;

2. Calcular a média temporal do *pitch* médio $\bar{P}(t)$ dos últimos 5 s (média móvel) por *frame* — isso faz com que a evolução temporal de $\bar{P}(t)$ seja mais suave;

3. Detectar erros de oitava dos contornos, removendo-se os contornos que estiverem mais distantes de $\bar{P}(t)$;

4. Recalcular $\bar{P}(t)$ usando os contornos restantes, pelos os passos 1 e 2;

5. Remover qualquer contorno cuja distância média para $\bar{P}(t)$ seja maior que uma oitava;

6. Repetir os passos 3 a 6 mais duas vezes (já é o bastante para obter boas estimativas da melodia correta), removendo os contornos e atualizando $\bar{P}(t)$.

Após as iterações, os contornos restantes passam pelo último estágio, que consiste em selecionar os contornos que formam a melodia. Esta última etapa é realizada simplesmente por escolher, quando há mais de um contorno sobrepostos no tempo, o que tiver maior soma de saliência $C_{\sum s}$. Assim, o sistema entrega, como saída, uma série temporal de F0.

Uma vez que precisamos da sequência de notas em formato MIDI para incluirmos as músicas na base, foi necessário incluir um estágio de detecção de notas, o mesmo utilizado na transcrição do sinal da consulta. Como não há informação da envoltória de sinal para auxiliar na detecção dos *onsets*, esse procedimento foi realizado apenas pelas etapas descritas na Seção 2.3.2, em que são utilizados critérios de análise da evolução temporal da F0.

Capítulo 4

Bases de Músicas

Neste capítulo, serão descritas as bases de músicas e de consultas utilizadas nos experimentos realizados. Foram adotados dois pares de bases com características diferentes, sendo cada um mais representativo em seu contexto. O primeiro é formado apenas por músicas brasileiras e compreende uma base de músicas pequena, porém com uma base de consultas mais completa e controlada. O segundo é formado por músicas da banda *The Beatles* e possui um conjunto de consultas mais limitado em quantidade, porém mais representativo do uso do sistema na prática, com interferências acústicas ambientais e sistema de gravação de baixa qualidade. Ambos foram utilizados para testar sistemas de extração automática de melodia, mas apenas o primeiro par foi utilizado para testar o sistema proposto com banco de músicas dinâmico (abordado no Capítulo 6), justamente por possuir maior quantidade de consultas.

4.1 Conjunto de Bases de Teste I — Brasil

Para que um sistema de QBH seja capaz de realizar pesquisas por músicas para um determinado grupo de pessoas, é preciso responder adequadamente a buscas relacionadas à cultura local, que pode abranger tanto produções musicais locais quanto externas.

Há trabalhos na área de QBH que utilizam músicas de países específicos, como Estados Unidos, Alemanha, China e Irlanda, assim como grupos musicais, como a banda inglesa *The Beatles*. Com a finalidade de trabalhar com sistemas de QBH no contexto brasileiro, foi adotada uma base construída em [49] composta, exclusivamente, por músicas brasileiras.

Essa base contém músicas de diferentes épocas e autores, abrangendo diversos estilos. A referência utilizada para selecionar as músicas da base foi uma lista publicada na revista Rolling Stone [50] intitulada "As 100 Maiores Músicas Brasileiras". Para complementar a lista, foram utilizadas outras 10 músicas populares, como can-

ções infantis, músicas de festa e hinos. Dessas 110 músicas, 52 foram elencadas para constituírem a base de músicas por terem maior popularidade e apenas seus trechos mais conhecidos foram colocados no banco (a maioria com duração variando entre 15 s e 40 s). Com isso, a chance de as pessoas as conhecerem para gravar as consultas seria maior.

Para construir a base de melodias das músicas, arquivos MIDI que contivessem suas melodias principais foram obtidos na *internet* em diversas fontes. Em seguida, foi realizada a edição de tais arquivos com o auxílio de um profissional. Quando necessário, foram corrigidos erros da melodia e foi retirado tudo que não fizesse parte dela, como instrumentos e vocais secundários; e, como as músicas não foram postas inteiras na base, foram excluídos os trechos melódicos que não faziam parte dos trechos escolhidos para compor a base. Buscou-se para cada trecho, ainda, uma versão extraída de gravação comercial.

Para construir o banco de consultas dos usuários, 43 pessoas se candidataram e gravaram, no total, 1123 consultas entre agosto e dezembro do ano de 2011. Em [49] foram implementados e testados diferentes métodos de comparação de melodias e parte deles precisava que o trecho cantado fosse o mesmo do elemento correspondente da base (em vez de ser apenas uma parte). Por isso, para composição da base, pediu-se aos usuários que, uma vez tendo escolhido um conjunto de músicas que conhecessem, cantarolassem exatamente o trecho de cada música escolhida (utilizando para cada nota o fonema "lá") em três diferentes situações, cada uma delas caracterizando um tipo:

Tipo 1 – enquanto se ouvia a referência MIDI do elemento da base e se lia a letra correspondente;

Tipo 2 – enquanto se ouvia a gravação comercial da música escolhida e também se lia a letra correspondente;

Tipo 3 – enquanto se lia a letra, cantando de memória.

Convém ressaltar que, de certa forma, essas gravações correspondem a um caso menos geral do uso do sistema, uma vez que, na prática, pessoas poderiam cantar qualquer trecho de uma música para consultá-la.

Todas as consultas foram gravadas em uma sala projetada para gravação do laboratório SMT (Sinais Multimídia e Telecomunicações). A sala é acusticamente tratada, com baixo nível de ruído e de reverberação. Nesta sala, os voluntários usaram um par de fones de ouvido (modelo Sennheiser[1] HD 265) para monitoração e havia um microfone (modelo Shure[2] SM-81) de alta qualidade para a captura do sinal cantado.

[1] `http://en-br.sennheiser.com/`
[2] `http://www.shure.com/`

De dentro da sala de gravação, os voluntários tinham acesso ao programa de gravação de consultas (desenvolvido em [49]) por meio de um *mouse wireless* e um monitor posicionado na sala ao lado, porém em frente ao vidro que separa as duas salas. Dessa maneira, o ruído do computador e de outros equipamentos não influenciou na gravação, devido ao isolamento acústico entre as salas, e os usuários podiam ver na tela a letra da consulta que realizavam.

Nessa segunda sala havia um *notebook* ligado ao monitor e uma interface de áudio, também conectada ao *notebook*, utilizados para realizar a gravação do sinal capturado pelo microfone e para a reprodução concomitante dos sinais de referência. Todos os sinais foram transmitidos através de cabos de áudio que ligam os equipamentos de uma sala a outra.

As gravações foram realizadas com resolução de 16 bits e taxa de amostragem de 48 kHz. Convém ressaltar que as gravações, sendo realizadas em ambiente de estúdio, não representam fielmente o uso dos sistemas de QBH na prática, quando poderá haver ruído ambiente (e.g. pessoas falando ou som de automóveis) e reverberação, fatores que podem influenciar significativamente na transcrição do sinal cantado. Porém, dessa maneira é possível aferir o desempenho dos sistemas desconsiderando as interferências do meio em que se encontra o usuário. Quando for necessário considerar tais interferências, será preciso gravar uma nova base ou simular tais situações pela adição dos sinais de interferência de maneira controlada.

Na Tabela 4.1 estão relacionadas as músicas escolhidas, assim como o número de gravações de consultas discriminado por tipo. A lista está ordenada pelo número total de gravações realizadas de cada música, estando este posicionado na última coluna da tabela. Note que a música "Rosa", apesar de não ter recebido nenhuma consulta, ainda faz parte da base de músicas, podendo ser elencada no ranque como falso positivo.

Tabela 4.1: Tabela de músicas do Conjunto I e número (total e discriminado por tipo) de consultas gravadas.

Título	**Tipo 1**	**Tipo 2**	**Tipo 3**	**Total**
Hino Nacional	13	15	28	56
Parabéns a Você	9	17	26	52
Ana Júlia	9	15	23	47
Asa branca	14	9	23	46
Garota de Ipanema	11	12	22	45
Ciranda cirandinha	12	10	22	44
Eu sei que vou te amar	12	9	21	42
Você não soube me amar	10	11	20	41

Continua na próxima página

Tabela 4.1 – *Continuação da página anterior.*

Título	**Tipo 1**	**Tipo 2**	**Tipo 3**	**Total**
País tropical	15	5	20	40
Que país é este?	6	14	18	38
Escravos de Jó	10	9	19	38
Quero que vá tudo pro inferno	9	9	18	36
Mamãe eu quero	11	7	18	36
Trem das onze	7	10	17	34
Ilariê	9	8	17	34
Preta pretinha	12	4	15	31
A banda	7	8	15	30
Gita	7	8	15	30
Águas de março	6	8	14	28
Fim de Ano	7	7	14	28
Carinhoso	8	5	13	26
Alegria, alegria	4	8	12	24
Detalhes	6	4	10	20
Luar do sertão	4	5	10	19
Chega de saudade	4	5	9	18
Pra não dizer que não falei das flores	5	4	9	18
Aquarela do Brasil	3	5	8	16
Ideologia	2	6	8	16
Meu mundo e nada mais	5	3	8	16
Alagados	4	3	7	14
O barquinho	4	3	7	14
Felicidade	1	6	7	14
Conversa de botequim	2	4	6	12
Comida	3	3	6	12
Hino da bandeira	4	2	6	12
Travessia	2	3	5	10
Sentado à beira do caminho	3	2	5	10
Inútil	1	3	4	8
Me chama	3	1	4	8
A flor e o espinho	2	2	4	8
Disritmia	2	2	4	8
Eu quero é botar meu bloco na rua	0	3	3	6
Ponteio	2	1	3	6

Continua na próxima página

Tabela 4.1 – *Continuação da página anterior.*

Título	**Tipo 1**	**Tipo 2**	**Tipo 3**	**Total**
As curvas da estrada de Santos	2	1	3	6
Coelhinho de olhos vermelhos	1	2	3	6
Ronda	0	2	2	4
Casa no campo	2	0	2	4
A noite de meu bem	1	1	2	4
Chegou a hora da fogueira	0	2	2	4
Manhã de carnaval	1	0	1	2
Foi um rio que passou em minha vida	0	1	1	2
Rosa	0	0	0	0

Todas as músicas escolhidas foram submetidas aos processos de transcrição dos sistemas. Porém, não foi possível obter sucesso na transcrição de todas as músicas dessa base utilizando o sistema GPA-UdelaR. Algumas músicas, como hinos e músicas infantis não possuem um vocal único proeminente, e com várias vozes cantando em uníssono não há boa definição das linhas de frequência da melodia no espectrograma; as linhas de F0, por conta de pequenas imperfeições naturais na execução, se cruzam frequentemente, dificultando a tarefa de se obter uma frequência fundamental consistente.[3] Como resultado, nesses casos os elementos de voz são removidos, por não terem sido caracterizados como voz humana pelo sistema. Com isso, as transcrições de "Você não soube me amar", "Ilariê", "Hino da Bandeira" e "Hino Nacional" acabaram por ter nenhuma nota e, por essa razão, foram removidas desse banco.

Realizando uma aferição auditiva das transcrições obtidas das músicas restantes, em muitas delas foi possível notar problema semelhante; grande número de notas não foram consideradas como sinal cantado, resultando em um pequeno número de notas transcritas. Como uma alternativa com relação a esse problema, foi criada uma outra base, removendo do sistema a etapa de classificação de voz. Isso aproxima o sistema GPA-UdelaR do Melodia, que não tem por objetivo separar somente a voz, mas sim a melodia principal presente. Talvez, uma possível explicação para o fenômeno do mal desempenho da etapa de classificação seja a falta de representatividade das amostras utilizadas para o treinamento do classificador, consequentemente causando falta de generalidade. As formas de cantar de diferentes culturas costumam ser distintas (formas de empostação da voz, de interligar as notas etc), e isso vale para

[3]Esses são casos típicos que impõem grande dificuldade a algoritmos de extração de melodia, que precisam de linhas espectrais nítidas e que concentrem grande saliência para que seja identificada corretamente a melodia.

os brasileiros. Considerando que a base de treinamento não contou com amostras dessa categoria, essa explicação nos parece razoável.

Como já foi dito, o sistema Melodia busca transcrever qualquer melodia que considere principal na música. Com isso, observamos que, em algumas músicas, havia fraseados e melodias secundárias que eram transcritas, principalmente notas de instrumentos que apareciam entre partes cantadas.

Como forma de realizar uma correção manual grosseira a respeito dos momentos da música em que há voz, foram realizadas marcações manuais de todas as músicas. Os instantes das músicas foram classificados segundo três categorias: voz, não voz e coral (que não tenha melodia principal). Essa análise não fornece meios de se averiguar se as notas obtidas estão corretas, mas apenas se há notas transcritas em momentos em que não há voz ou se faltam notas em momentos em que há voz. Os arquivos que guardam essas informações das marcações são arquivos texto com extensão ".lab" contendo uma lista de instantes iniciais e finais de cada região e a classificação correspondente. Com essas marcações, foi criada uma base com os arquivos fornecidos pelo sistema Melodia, eliminando quaisquer notas que estivessem fora dos instantes de voz, como maneira de aferição do grau de interferência dessas notas extras no desempenho do sistema. O motivo para isso não ter sido realizado com as bases transcritas pelo sistema GPA-UdelaR é que a origem do problema principal do transcritor resulta em falta de notas transcritas, e não no excesso delas.

Com isso, temos todas as bases utilizadas descritas abaixo:

MIDI — Composta pelos arquivos MIDI de referência gerados por um músico profissional a partir de partituras;

GPA — Composta pelos arquivos transcritos pelo sistema GPA-UdelaR (apenas 48 das 52 músicas puderam ser transcritas);

GPA 2 — Composta pelos arquivos transcritos pelo sistema GPA-UdelaR sem a etapa de classificação;

Melodia — Composta pelos arquivos transcritos pelo sistema Melodia;

Melodia (Lab.) — Composta pelos arquivos transcritos pelo sistema Melodia, mas usando as marcações para remover as partes da música que não possuem voz principal.

4.2 Conjunto de Bases de Teste II — *The Beatles*

A outra base de músicas adotada foi construída em [10] com todas as músicas da banda britânica *The Beatles* que contêm voz principal. A base de músicas é composta

por 208 arquivos MIDI, também obtidos na *Internet*[4] e manualmente editados para retirar as informações que não fossem relativas ao vocal principal.

As músicas foram utilizadas inteiras, possibilitando que o trecho consultado aparecesse mais de uma vez (quando repetido na música), o que pode ter contribuído para melhorar o resultado das buscas. Por outro lado, há maior possibilidade de se encontrar falsos positivos, por haver maior quantidade de trechos melódicos.

Para gravar a base de consultas, um computador equipado de um microfone foi colocado em uma sala onde voluntários podiam livremente entrar e cantar algum trecho de uma música dos *Beatles* que lhes viesse à mente. Os equipamentos utilizados para as gravações eram muito simples (como usuários comuns teriam à disposição) e a sala não era acusticamente tratada, o que fez com que as gravações fossem contaminadas por ruído ambiente e reverberação. Os voluntários podiam cantar qualquer trecho das músicas de que lembrassem, com a letra ou apenas cantarolando. Todas as gravações foram feitas em resolução de 16 bits e taxa de amostragem de 8 kHz. Com isso, a base representa um conjunto de testes menos controlado, porém mais realista quanto ao uso do sistema na prática.

Em [8], foi utilizada parte da base de consultas e a respectiva base de músicas para realizar experimentos com o sistema de extração automática de melodia abordado na Seção 3.1.1. O experimento consistiu em escolher algumas músicas da base para transcrever automaticamente e testar o desempenho do sistema Tararira utilizando as consultas relativas às músicas transcritas. Como o algoritmo de extração de melodias testado demanda muito tempo de processamento, o teste foi apenas realizado a título de prova de conceito, selecionando-se apenas as 12 músicas com o maior número de consultas para serem transcritas. Essas músicas estão relacionadas na Tabela 4.2, assim como o número de consultas realizadas para cada uma delas.

Tabela 4.2: Lista de músicas com mais consultas do Conjunto 2 e número de consultas gravadas.

Título	**Total**
Blackbird	12
Do you want to know a secret	7
For no one	8
Girl	10
Hey Jude	7
I call your name	4
I've just seen a face	8

Continua na próxima página

[4]Em *sites* como "The Beatles MIDI and video heaven", `http://beatles.zde.cz/`.

Tabela 4.2 – *Continuação da página anterior.*

Título	**Total**
Michelle	10
Rocky raccoon	8
The fool on the hill	11
When I'm sixty four	8
Yesterday	13

Com a mesma finalidade da criação das bases descritas anteriormente, as músicas escolhidas da base *The Beatles* foram transcritas utilizando ambos os sistemas. Diferentemente da experiência anterior, não houve problemas na classificação dos sinais de voz pelo sistema GPA-UdelaR, o que reforça a hipótese de a origem do problema estar na falta de generalização do sistema por consequência da escolha das amostras de voz utilizadas no treinamento não ter contemplado músicas brasileiras.

As bases foram criadas de maneira semelhante às anteriores e estão resumidas abaixo:

MIDI — Composta pelos arquivos MIDI encontrados na internet e editados de forma a manter apenas o vocal principal;

GPA — Contendo as 12 músicas listadas na Tabela 4.2 transcritas pelo sistema GPA-UdelaR e as músicas restantes da base MIDI;

Melodia — Criada da mesma forma que a base GPA, mas utilizando o sistema Melodia para a transcrição das 12 músicas selecionadas;

Melodia (Lab.) — Composta pelos mesmos arquivos da base Melodia, mas usando as marcações para remover as partes que não possuem voz principal nas músicas transcritas.

Capítulo 5

Experimentos com Transcrição Automática de Bases de Músicas

Neste capítulo, serão descritos experimentos que objetivaram a avaliação do uso, em sistemas de QBH, das bases de músicas transcritas automaticamente pelos sistemas de transcrição de melodia GPA-UdelaR e Melodia em comparação com o uso das bases MIDI. Também serão descritos e discutidos os resultados obtidos em cada experimento realizado.

5.1 Definições Preliminares

Os experimentos consistem em submeter ao sistema Tararira as consultas existentes nas bases para que sejam feitas as respectivas buscas nas bases de música, aferindo assim a viabilidade do uso das bases transcritas em comparação com a base MIDI de referência.

Como forma de medida do desempenho do sistema, utilizam-se duas métricas: taxa de acerto dentro das Top-X e *Mean Reciprocal Rank* (MRR). A taxa de acerto nas Top-X é simplesmente a porcentagem de consultas cuja música correta foi elencada entre as primeiras X posições no ranque ($r_i < X$). O MRR é definido como

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_i}, \tag{5.1}$$

onde N é o número de consultas e r_i é o ranque obtido pela música correta quando feita a i-ésima consulta. Note que os resultados são mais sensíveis a ranques próximos às primeiras posições, que são os mais relevantes para os usuários.

5.2 Experimentos — *The Beatles*

Utilizando o conjunto de bases de teste *The Beatles*, foram realizados experimentos a fim dar continuidade ao trabalho [8], que foi a primeira publicação com o uso do sistema de transcrição GPA-UdelaR com sistemas de QBH. A principal característica desse conjunto de bases, que motiva seu uso além das bases com músicas brasileiras, é a maior proximidade com o uso real do sistema, em que pessoas cantam qualquer trecho das músicas e as consultas são gravadas em baixa qualidade. Além disso, esse conjunto de músicas apresenta diferentes características de instrumentação e arranjo vocal, quando comparado com as músicas brasileiras, fornecendo maior variedade para o teste do desempenho dos sistemas de transcrição de melodia.

Como já dito, todas as consultas são relativas às 12 músicas escolhidas (ver Seção 4.2), enquanto as demais músicas das bases são utilizadas como elementos candidatos a falsos positivos. Os ranques cumulativos estão ilustrados na Figura 5.1, com limite de 20 posições no ranque, facilitando a visualização das primeiras posições, enquanto os resultados obtidos em função das métricas MRR e acerto nas Top-X estão resumidos na Tabela 5.1. Os resultados obtidos com a base MIDI podem ser considerados como referência de limite superior de desempenho. Mesmo este não sendo perfeito, vale lembrar que não há como os usuários cantarolarem/cantarem as consultas bem o bastante para que, em média, se tenha 100% de acerto.

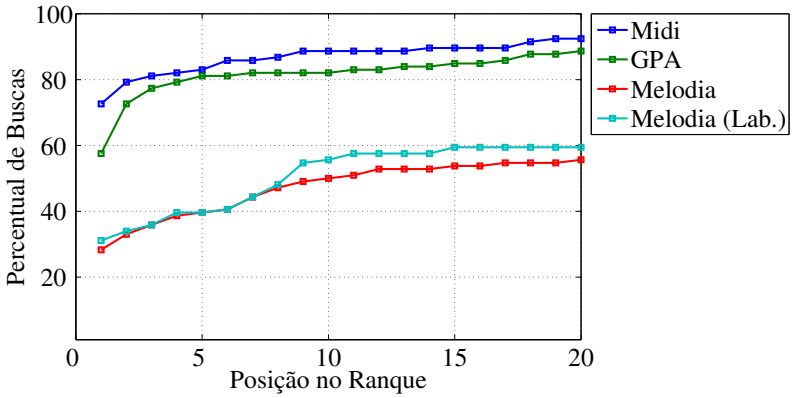


Figura 5.1: Ranques cumulativos para as bases do conjunto *The Beatles*.

Tabela 5.1: Resultados globais para o conjunto de bases *The Beatles*.

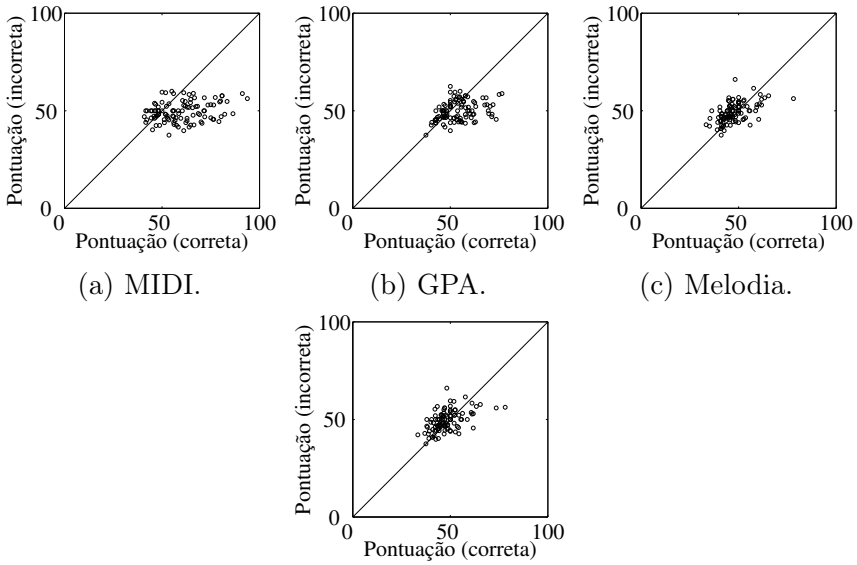
Base	MRR	Top-1	Top-5	Top-10
MIDI	0,78	72,64	83,02	88,68
GPA	0,68	57,55	81,13	82,08
Melodia	0,35	28,30	39,62	50,00
Melodia (Lab.)	0,37	31,13	39,62	55,66

Nesses resultados nota-se uma evidente superioridade do desempenho do sistema Tararira utilizando a base GPA em relação ao desempenho com as bases Melodia e Melodia (Lab.). O procedimento de remover as notas fora das regiões de voz praticamente não surtiu efeito para os resultados com posições do ranque menores que 9. Porém, convém ressaltar que o conjunto de músicas é demasiado reduzido e seletivo com relação ao estilo de músicas e sonoridade das gravações para ter relevância estatística, reduzindo tais resultados ao pequeno conjunto de teste utilizado e não nos permitindo assegurar de maneira patente a diferença de desempenho dos métodos de transcrição.

Os resultados obtidos para as bases MIDI e GPA apresentam pequena variação com relação aos resultados reportados em [8], possivelmente devido a mudanças de configuração de valores adotados nos algoritmos utilizados. Para a base MIDI, por exemplo, foi obtido um acerto nas Top-10 de 88,68%, enquanto foi reportado um acerto de 91,51% em [8]. Embora não tenhamos encontrado exatamente os mesmos resultados, consideramos que os resultados são equivalentes para as análises feitas e conclusões a que chegamos; além disso, as mesmas configurações do sistema foram utilizadas para todos os testes reportados neste trabalho, o que é suficiente para comparar os resultados dos experimentos com as bases utilizadas aqui.

Uma outra forma de analisar o desempenho do sistema é uma métrica que utiliza os resultados da comparação de melodias feita pela sequência de notas MIDI antes do refinamento por meio do LDTW; compara-se a pontuação encontrada para o elemento correto da base com a maior pontuação dos outros elementos, dando-nos uma ideia do quão distinguível está o elemento correto dos demais quando se utiliza o primeiro método. Uma análise dessa natureza foi proposta em [8], motivo pelo qual a reproduzimos também aqui.

Na Figura 5.2 podem-se observar os resultados obtidos utilizando essa métrica. Note que há uma linha na diagonal das figuras que representa o limiar entre os resultados; caso haja maior pontuação para a música correta da base, o ponto na figura que representa a consulta aparecerá abaixo da linha; em caso contrário, estará acima. É possível observar que, na base MIDI, a nuvem de pontos se espalha pelo meio do gráfico na vertical, e na horizontal tende a estar do meio para a direita, configurando bons resultados já no primeiro estágio de comparação de melodias. No caso da base GPA, também nota-se o mesmo comportamento, embora não tão acentuado como no caso anterior. Para a base Melodia e Melodia (Lab.) já não é possível observar o mesmo padrão, havendo uma distribuição dos pontos muito centrada, na figura.



(a) MIDI. (b) GPA. (c) Melodia.

(d) Melodia (Lab.)

Figura 5.2: Pontuação, para cada consulta, da música correta contra a maior pontuação das outras músicas.

Embora todos os resultados observados aqui sejam concordantes com os que são reportados utilizando as outras métricas, convém ponderar que, mesmo que a maior pontuação das outras músicas supere a pontuação da música correta (ponto acima da diagonal no gráfico), esta ainda poderá estar ranqueada em uma posição próxima do topo. Portanto, essa métrica não é apropriada para uma análise geral do desempenho do primeiro método de comparação de melodias. Além disso, para o caso das bases brasileiras, em que há uma quantidade muito grande de consultas, observar-se a nuvem de pontos dessa forma não é prático, devido à grande sobreposição dos pontos nos gráficos.

5.3 Experimentos — Brasil

Ao submeter o sistema ao uso do conjunto de bases Brasil obtivemos os resultados presentes na Figura 5.3, em que estão ilustrados os ranques cumulativos para todos os resultados obtidos até a posição 20.

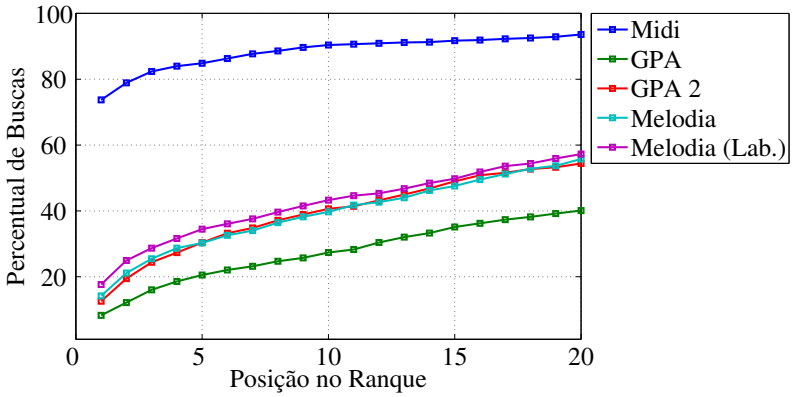


Figura 5.3: Ranques cumulativos para as bases do conjunto Brasil.

O primeiro fato a ser observado é que o desempenho do sistema utilizando a base MIDI é notavelmente superior, quando comparado ao desempenho obtido com o uso das bases transcritas automaticamente. Ademais, o desempenho obtido com a base GPA se destaca como o pior, enquanto os outros têm um comportamento semelhante entre si. Isso mostra quão mal sucedido foi o estágio de classificação do sistema GPA-UdelaR ao ser usado com as músicas brasileiras, pois retirando-o da cadeia de processamento (base GPA 2) os resultados melhoraram sensivelmente.

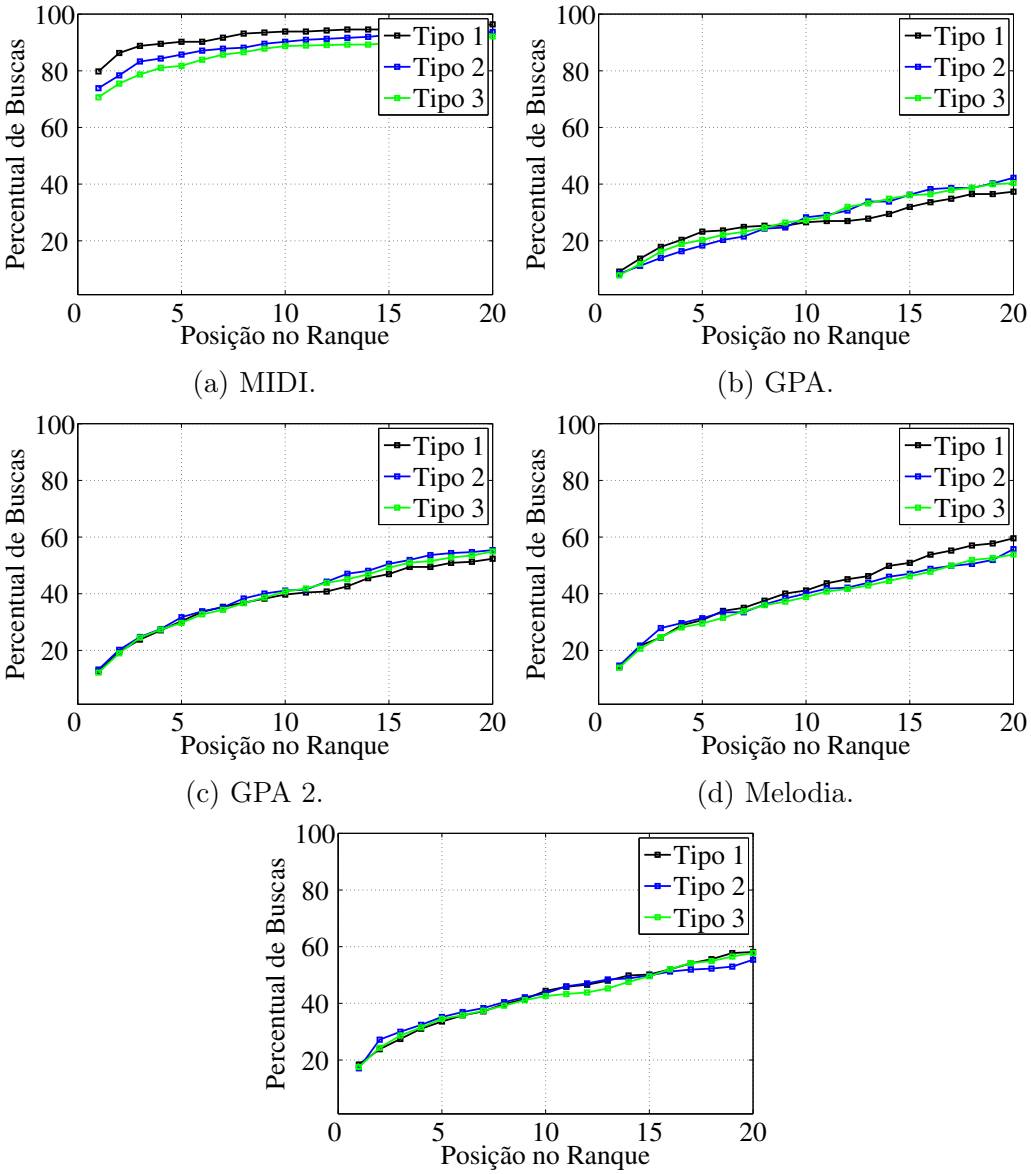
Quanto ao uso das marcações manuais (voz e não voz), nota-se que a melhora não foi significativa com a base Melodia (Lab.), pois o problema não consistia em haver notas fora dos instantes de atividade de voz. Convém salientar também que a base de músicas brasileiras não dá margem a grandes erros desse gênero, uma vez que foram escolhidos trechos das músicas que, em geral, eram curtos e cujo vocal principal estava presente em grande parte do tempo.

Na Tabela 5.3 estão resumidos os resultados para todos os métodos, utilizando as métricas MRR e acerto nas Top-X. Os resultados obtidos com a base MIDI novamente podem ser considerados como referência de limite superior de desempenho. Unindo esses resultados aos observados na Figura 5.3, nota-se que ambos os métodos de transcrição fornecem acertos nas Top-10 em torno de 40 %. Com isso, conclui-se que, nesse contexto de músicas brasileiras, ambos não são capazes de resolver o problema de transcrição completamente automática de bases de música para sistemas de QBH, fornecendo resultados equivalentes.

Tabela 5.2: Resultados globais para o conjunto de bases Brasil.

Base	MRR	Top-1	Top-5	Top-10
MIDI	0,79	73,73	84,86	90,38
GPA	0,16	8,27	20,51	27,35
GPA 2	0,23	12,56	30,37	40,61
Melodia	0,24	14,16	30,28	39,72
Melodia (Lab.)	0,27	17,63	34,46	43,28

As Tabelas 5.3, 5.4 e 5.5 resumem os resultados obtidos para cada tipo de consulta feita, enquanto os ranques cumulativos, também discriminados por tipo de consulta, estão ilustrados na Figura 5.4. Os resultados obtidos com a base de músicas MIDI demonstram que houve alguma vantagem de se cantar seguindo a referência de melodia utilizada na base. Isso pode ser constatado pelos resultados das consultas do tipo 1 (cantadas enquanto se ouvia a melodia). Já os resultados obtidos para as bases de música transcritas não apresentam diferença relevante de desempenho entre os tipos de consulta.



(a) MIDI.

(b) GPA.

(c) GPA 2.

(d) Melodia.

(e) Melodia (Lab.).

Figura 5.4: Ranques cumulativos discriminados por tipo de consulta.

Tabela 5.3: Resultados para as consultas do tipo 1.

Base	MRR	Top-1	Top-5	Top-10
MIDI	0,85	79,78	90,25	93,86
GPA	0,17	9,13	23,24	26,56
GPA 2	0,22	12,64	30,32	39,71
Melodia	0,24	14,08	30,69	41,16
Melodia (Lab.)	0,27	18,41	33,57	44,40

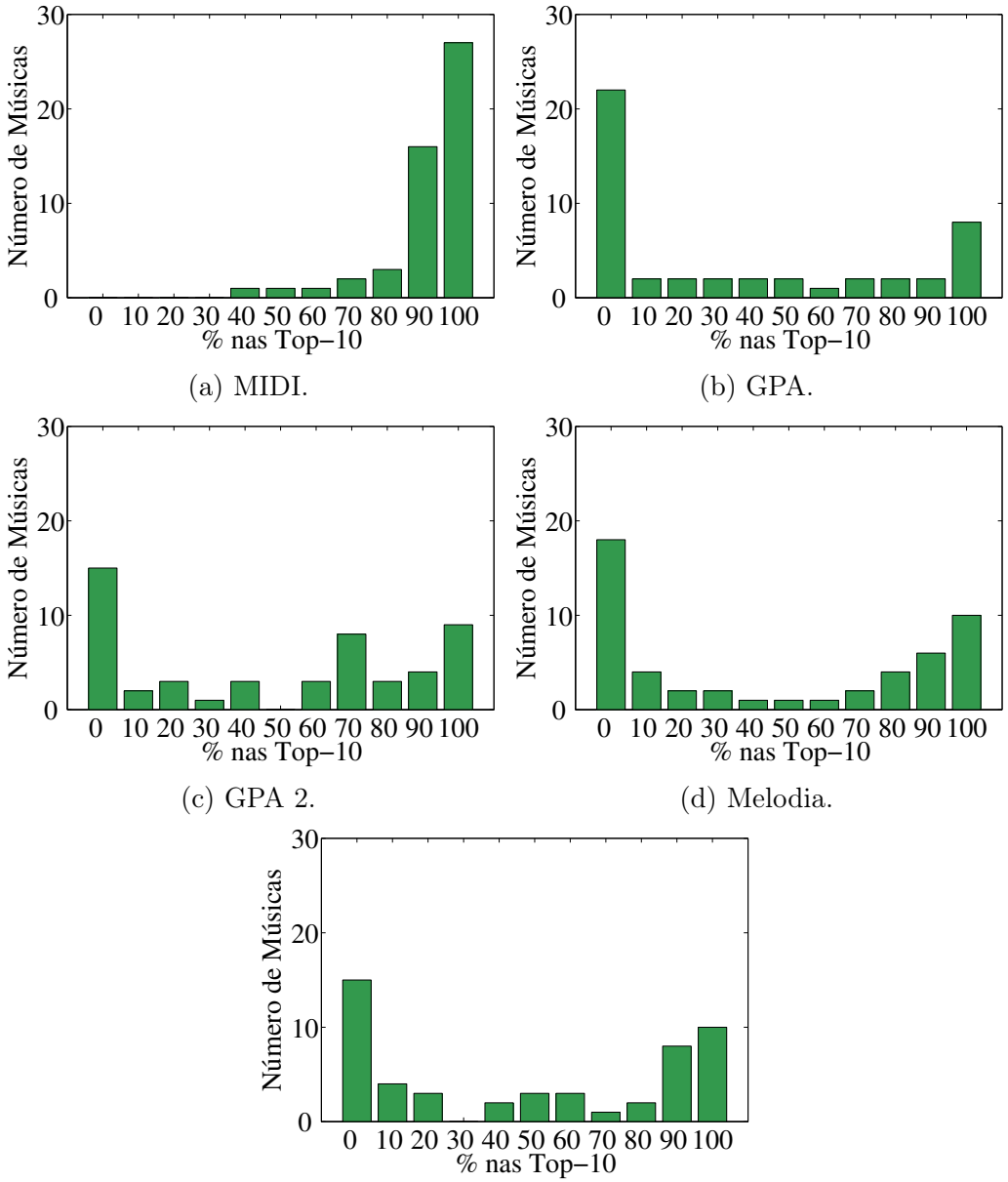
Tabela 5.4: Resultados para as consultas do tipo 2.

Base	MRR	Top-1	Top-5	Top-10
MIDI	0,79	73,87	85,71	90,24
GPA	0,16	8,37	18,33	28,29
GPA 2	0,23	13,24	31,71	41,11
Melodia	0,24	14,63	31,36	40,07
Melodia (Lab.)	0,27	17,07	35,19	43,55

Tabela 5.5: Resultados para as consultas do tipo 3.

Base	MRR	Top-1	Top-5	Top-10
MIDI	0,76	70,66	81,75	88,73
GPA	0,16	7,79	20,29	27,25
GPA 2	0,22	12,16	29,70	40,79
Melodia	0,23	13,95	29,52	38,82
Melodia (Lab.)	0,27	17,53	34,53	42,58

Para observar de outra forma os resultados obtidos, na Figura 5.5 estão ilustrados histogramas de percentual nas Top-10 para cada banco de músicas. Cada música de cada diferente banco contém um percentual de consultas a elas relacionadas cujo resultado obtido ficou entre as Top-10. Com isso, o histograma contabiliza, em cada *bin*, o número de músicas cujo percentual de aparecimento nas Top-10 está naquela faixa (e.g. o primeiro *bin* contabiliza quantas músicas tiveram menos de 5 % das suas consultas com o resultado correto entre as Top-10). Quanto mais músicas forem mal transcritas, mais o histograma será concentrado em baixos valores de % de Top-10.



(a) MIDI.

(b) GPA.

(c) GPA 2.

(d) Melodia.

(e) Melodia (Lab.).

Figura 5.5: Histogramas — Distribuição de percentual nas Top-10 das músicas por base.

As imagens na Figura 5.5 mostram que a maioria das músicas se concentra em *bins* que representam ou valores altos de percentual nas Top-10, ou valores muito baixos. Isso nos permite concluir que, de maneira geral, há músicas que foram bem transcritas, fornecendo bons resultados consistentemente, enquanto outras foram muito mal transcritas. Isso se reflete no padrão bimodal observado, com a exceção da base MIDI; por ser essa a base de referência, o sistema tende a obter com ela bons resultados consistentemente.

5.4 Conclusões Gerais

Como primeira constatação, comparando os resultados obtidos com os dois conjuntos de bases, nota-se que as transcrições realizadas pelo algoritmo GPA-UdelaR obtiveram um desempenho drasticamente superior no primeiro experimento, o que contribui para a hipótese de haver um problema no treinamento do sistema de classificação, conforme já discutido anteriormente; tendo sido removido o classificador, os resultados para o conjunto Brasil melhoraram e se equipararam aos das bases Melodia e Melodia (Lab.).

No experimento com as bases do conjunto *The Beatles*, é possível notar que os resultados com a base GPA são muito superiores aos obtidos com o sistema de transcrição Melodia, o que também pode ser explicado pelo fato de o Melodia apenas fornecer como resultado a série temporal de F0, limitando as possibilidades de encontrar corretamente os *onsets* das notas. Porém, embora os da base Melodia sejam inferiores aos da base GPA nesse conjunto, esses ainda são sensivelmente superiores aos obtidos com a base Melodia do conjunto Brasil, com 50% de resultados corretos nas Top-10, contra 39% para a base brasileira. Tudo isso indica que talvez a base de músicas brasileiras possa oferecer genericamente maiores desafios à transcrição automática de melodia.

Uma hipótese para explicar tal fenômeno recai sobre a maneira de os brasileiros cantarem música popular, de maneira geral. Culturalmente, os cantores brasileiros tendem a emitirem as notas com naturalidade [51], de maneira que a voz não se destaca tanto dos outros instrumentos na música, o que também pode variar com a época e o estilo de cada música. Com a emissão sendo realizada com maior intensidade, há maior presença de harmônicos e, com isso, há maior destaque das notas da voz (maior saliência). Outro ponto importante a se destacar é a presença de outras vozes, instrumentos melódicos e reverberação em grande parte das músicas escolhidas para compor a base brasileira, que também dificultam o processo de transcrição de melodia.

Os resultados obtidos pelas bases transcritas pelo sistema Melodia, parecem coerentes com resultados reportados [9] pelo autor. Um resultado global reportado aponta taxa de acerto das melodias principais de cerca de 70% com as figuras de mérito descritas em [52], que consideram diversos métricas (e.g. a taxa de acerto de *frames* como vozeados ou não vozeados, taxa de acerto de F0, etc). Se com as bases MIDI, que podemos considerar como tendo 100% de acerto da melodia presente, obteve-se cerca de 90% de acerto nas Top-10 para os experimentos, uma taxa de acerto de 70% na transcrição (talvez menos, pelas dificuldades impostas pelas bases do conjunto Brasil) justifica os 50% e 39% de consultas nas Top-10 dos experimentos com as bases dos conjuntos *The Beatles* e Brasil, respectivamente.

Por aferições auditivas das transcrições, foi possível constatar que houve, com certa frequência, saltos de oitava, intromissão da melodia de outros instrumentos na melodia da voz, perda de notas cantadas com baixa intensidade e, para o sistema Melodia, perda de *onsets*, o que já era previsto. Todos esses desvios na melodia, mesmo se não muito frequentes, podem causar um grande impacto no desempenho do sistema. Note que as melodias são codificadas usando uma quantização do salto entre as notas; portanto, se uma só nota for perdida, perdem-se os intervalos da nota anterior à que foi perdida e o intervalo desta para a próxima, substituindo ambos pelo intervalo entre as notas que restaram, que perde o sentido. Erros dessa natureza ocorrendo com frequência geram resultados muito ruins da comparação das sequências de notas, principalmente se o trecho consultado não tiver muitas notas.

Como conclusão dessa análise, os resultados obtidos demonstram que esses sistemas de transcrição automática ainda não são capazes de atingir, para o uso em sistemas de QBH, resultados consistentes e satisfatórios para todos os tipos de música. Com isso, não é possível obter uma base de músicas completamente composta por músicas transcritas automaticamente, havendo a necessidade de substituição ou correção de parte da base. Esses resultados motivaram o desenvolvimento do sistema proposto, descrito no Capítulo 6, que faz uso das consultas e de respostas dos usuários para melhorar progressivamente a base de músicas. Embora as consultas possuam erros e possam ser incompletas, elas podem ser boas opções para substituir músicas mal transcritas da base: mesmo não sendo ideais, podem compensar o trabalho da transcrição manual dos elementos cujos sistemas de transcrição automática não foram capazes de transcrever.

Capítulo 6

Sistema de Adaptação Automática da Base de Músicas

Neste capítulo, serão exploradas as possibilidades de uso das próprias consultas realizadas pelos usuários para a composição da base de músicas; será descrito o funcionamento do sistema proposto que realiza progressivamente a adaptação da base de músicas; e serão expostos e discutidos os resultados das simulações realizadas com o sistema quando submetido a diversas situações.

6.1 Uso de Consultas como Elementos de uma Base de Músicas

Conforme já foi discutido, o desempenho do sistema de QBH utilizando as bases de música transcritas pelos sistemas Melodia e GPA-UdelaR aponta para a inviabilidade do uso de tais bases na prática. Porém, é possível utilizar a parte bem transcrita da base de músicas fornecida pelos sistemas de transcrição automática e corrigir ou substituir as músicas mal transcritas.

A proposta deste trabalho consiste em fazer uso das consultas realizadas pelos usuários para tentar identificar as músicas da base que estejam mal transcritas e corrigi-las. Como os usuários, ao realizarem a consulta, fornecem ao sistema uma melodia, parece razoável a hipótese de que tal melodia, ainda que contenha erros, possa servir de subsídio para a tarefa de correção da música buscada, caso esta esteja mal transcrita.

Como a base de consultas do conjunto de músicas brasileiras é mais completa em termos de número de consultas, seu respectivo conjunto de bases foi adotado para os experimentos realizados para a avaliação de desempenho do sistema proposto.

Como uma primeira análise, que servirá de fundamento para o sistema proposto, realizaram-se testes utilizando tão somente as consultas feitas pelos usuários. O

objetivo desses primeiros testes é aferir o quão similares são as consultas relativas a uma mesma música e se é possível distingui-las das demais. Para isso, foi realizada a comparação entre as melodias de cada consulta e todas as outras através do algoritmo Distância de Levenshtein, obtendo assim uma matriz de similaridade $\mathbf{S} \in \mathbb{R}^{C \times C}$, tal que $C = 1023$, cujos elementos são as pontuações $S(c_i, c_j)$ entre 0 e 100 (ver Seção 2.4.2) relativas à similaridade entre as consultas c_i e c_j, que por sua vez foram realizadas para as músicas de índice k e l, respectivamente. Na Tabela 6.1 se encontra a estatística geral, com a similaridade média $\bar{S}$ e respectivo desvio padrão σ_S, de todos os pares (c_i, c_j) de consultas relativos à mesma música ($\bar{S}_{k=l}$) e de todos os pares de consultas relativos a músicas diferentes ($\bar{S}_{k \neq l}$).

Tabela 6.1: Médias e desvios padrão das pontuações obtidas nas comparações entre melodias das consultas relativas às mesmas músicas e a músicas diferentes.

	$\bar{S}$	σ_S
Mesma música ($k = l$)	46,82	10,03
Músicas diferentes ($k \neq l$)	34,55	6,99

Como uma primeira constatação, nota-se que $\bar{S}_{k=l}$ tende, de fato, a ser maior que $\bar{S}_{k \neq l}$. A diferença das médias obtidas é de mais de 10 pontos, embora haja uma região de interseção indicada pelos desvios padrão. Isso indica que, em média, as consultas de uma mesma música são mais parecidas entre si do que com consultas relativas a outras músicas, o que já era esperado.

Na Tabela 6.2 estão dispostos, para cada música k, a quantidade de comparações realizadas entre consultas, a média $\bar{S}_k$ e o desvio padrão σ_{S_k} da similaridade obtida entre as consultas da mesma música k. As músicas estão ordenadas por quantidade de comparações realizadas.

Tabela 6.2: Tabela de músicas do Conjunto I com número total de comparações entre consultas, média $\bar{S}_k$ e desvio padrão σ_{S_k} das similaridades S_k obtidas entre consultas relativas à k-ésima música.

Título	**Qtd. de Comp.**	$\bar{S}_k$	σ_{S_k}
Hino Nacional	1540	44,54	8,13
Parabéns a Você	1326	49,66	10,30
Ana Júlia	1081	48,23	12,29
Asa branca	1035	50,15	8,36
Garota de Ipanema	990	45,86	8,41

Continua na próxima página

Tabela 6.2 – *Continuação da página anterior.*

Título	**Qtd. de Comp.**	$\bar{S}_k$	σ_{S_k}
Ciranda cirandinha	946	47,14	8,81
Eu sei que vou te amar	861	46,03	8,15
Você não soube me amar	820	39,46	8,92
País tropical	780	39,76	8,85
Escravos de Jó	703	50,37	10,17
Que país é este?	703	47,06	11,33
Mamãe eu quero	630	40,07	7,57
Quero que vá tudo pro inferno	630	49,81	9,07
Ilariê	561	46,38	7,11
Trem das onze	561	45,76	6,91
Preta pretinha	465	57,17	12,54
Gita	435	51,39	10,76
A banda	435	50,61	7,85
Fim de Ano	378	48,45	8,37
Águas de março	378	43,17	6,54
Carinhoso	325	52,50	8,93
Alegria, alegria	276	49,56	8,75
Detalhes	190	45,65	8,71
Luar do sertão	171	42,62	15,41
Pra não dizer que não falei das flores	153	47,62	8,09
Chega de saudade	153	45,14	9,90
Meu mundo e nada mais	120	44,94	8,99
Ideologia	120	40,19	9,00
Aquarela do Brasil	120	49,15	6,22
Felicidade	91	52,48	8,06
O barquinho	91	47,28	4,90
Alagados	91	40,30	6,43
Hino da bandeira	66	42,00	6,38
Comida	66	59,40	5,75
Conversa de botequim	66	40,61	4,90
Sentado à beira do caminho	45	45,07	10,21
Travessia	45	43,99	8,07
Disritmia	28	39,31	7,77
A flor e o espinho	28	45,56	5,62
Me chama	28	46,69	6,80

Continua na próxima página

Tabela 6.2 – *Continuação da página anterior.*

Título	**Qtd. de Comp.**	$\bar{S}_k$	σ_{S_k}
Inútil	28	48,70	7,29
Coelhinho de olhos vermelhos	15	52,28	8,73
As curvas da estrada de Santos	15	41,51	12,69
Ponteio	15	41,81	6,51
Eu quero é botar meu bloco na rua	15	53,70	16,16
Chegou a hora da fogueira	6	47,45	7,32
A noite de meu bem	6	54,49	6,02
Casa no campo	6	55,36	2,23
Ronda	6	42,79	3,62
Foi um rio que passou em minha vida	1	67,57	0,00
Manhã de carnaval	1	50,00	0,00

É possível notar que há grande flutuação dos resultados obtidos por música; algumas músicas possuem pontuação média sensivelmente maior que as demais, o que talvez mostre que suas melodias sejam mais distinguíveis das outras. A música que obteve menor pontuação média entre suas consultas foi a intitulada "Mamãe eu quero", com pontuação média de 40,07±7,57 obtida em 630 comparações, enquanto a música "Comida" obteve pontuação média de 59,40±5,75 para 66 comparações.

É possível determinar uma faixa de valores em que haja grande probabilidade de as consultas serem relativas à mesma música. Pelos resultados obtidos entre consultas de diferentes músicas, temos a similaridade média $\bar{S}_{k\neq l} = 34{,}55$ e desvio padrão $\sigma_{S_{k\neq l}} \approx 7$; poderíamos considerar que, adotando-se um limiar $\bar{S}_{k\neq l} + \sigma_{S_{k\neq l}} \approx 41{,}5$ para selecionar os pares de consultas, grande parte das consultas em que $k \neq l$ seria eliminada; ou ainda, adotando-se um limiar mais rigoroso, digamos $\bar{S}_{k\neq l} + 2\sigma_{S_{k\neq l}} \approx 48{,}5$, essa chance seria ainda mais reduzida.

Porém, músicas como "Comida" provavelmente teriam suas consultas avaliadas como relativas a músicas diferentes; seria preciso encontrar um valor cujos resultados para $k = l$ também fossem aceitos na seleção. Em vez de estipular um limiar absoluto, o sistema proposto faz uso de diversas situações e associações criadas para aumentar a chance de acerto, determinando um limiar de seleção para cada situação, conforme será abordado na Seção 6.2.

Embora haja uma variação relativamente grande nas pontuações obtidas, conclui-se que, em média, seria possível utilizar as consultas como elementos da base de músicas, uma vez que a similaridade tende a ser maior quando a consulta é comparada com outra relativa à mesma música.

6.2 Sistema de Adaptação Automática da Base de Músicas

O problema principal que o sistema de adaptação automática da base de músicas precisa resolver consiste em determinar quais músicas da base estão mal transcritas (ou simplesmente o sistema não está conseguindo encontrar consistentemente com as consultas realizadas) e corrigi-las, precisando determinar para isso quais as consultas que seriam relativas a tais músicas e quais, dentre elas, seriam representativas o bastante para servir a esse propósito.

Daí surgem alguns problemas para os quais as soluções adotadas fazem parte do algoritmo proposto. Primeiramente, não é possível (pelo menos com as ferramentas que conhecemos) avaliar a qualidade de melodias sozinhas, sendo possível apenas compará-las com outras; ao compará-las, tem-se apenas uma estimativa do quão semelhantes ou diferentes estas são entre si, e não quão boas ou representativas elas são sozinhas. Suponhamos que se sabe que a consulta c_i e o elemento e_m da base de músicas $\mathcal{M}$ comparados sejam relativos à mesma música m e a pontuação obtida de similaridade seja baixa; como saber se é e_m que está mal transcrito ou se a consulta c_i está mal cantada? Consideraremos que quando uma música m é consultada e frequentemente não é bem colocada no ranque geral, o elemento e_m é um candidato a ser modificado. Note que não importa se a música está bem ou mal transcrita, o que importa é que os usuários possam consultá-la e o sistema identificá-la corretamente; se a maioria dos usuários canta determinado trecho de uma música errado, mas a maioria o canta de tal maneira especial, importa que o sistema aprenda com isso, melhorando assim seu desempenho.

Para que as consultas sejam utilizadas para corrigir as músicas da base, elas precisam ser armazenadas e identificadas. Os usuários recebem como resposta do sistema uma lista de músicas limitada em torno de 10 elementos; caso a música esteja mal colocada (fora dessa lista) o usuário será convidado a informar ao sistema qual é a música que foi consultada. Isso nos leva ao problema de os usuários poderem não saber informar qual o nome da música consultada; além disso, mesmo que o informem, não há garantia de que essa informação está correta.

O sistema proposto se baseia em duas hipóteses básicas:

1. É **provável** que **consultas boas** com o **mesmo trecho melódico** sejam **similares**

2. É **improvável** que duas **consultas boas** sejam, ao mesmo tempo, **similares** e relativas a **músicas diferentes**

E importa ressaltar dois outros pontos relacionados a esses:

1. Duas consultas podem ser **consultas boas**, pertencerem a uma **mesma música** e **não serem similares**, bastando para isso serem relativas a **trechos diferentes** da música

2. Duas **consultas ruins** podem ser **similares** e serem relativas a **músicas diferentes**

Diferenciamos aqui consultas boas de consultas ruins. Há usuários que não são capazes de cantar/cantarolar uma melodia mantendo a afinação e o ritmo razoavelmente corretos. Nesses casos, a consulta realizada possui demasiada quantidade de erros, configurando-se como o que chamamos de uma **consulta ruim**. Caso o usuário consiga realizar uma consulta dentro de certos padrões de afinação e ritmo, pelo menos dentro dos intervalos impostos pelas quantizações realizadas, sua consulta é considerada uma **consulta boa**.

Conforme discutido na seção anterior, não é muito bem definida uma pontuação de similaridade para que duas consultas sejam consideradas similares o bastante para se caracterizarem como relativas à mesma música. Para resolver esse problema, o sistema proposto agrupa as consultas em dois níveis: o primeiro nível consiste em grupos $\mathcal{C}_m$, onde m é o índice da música (ou índice X, para o caso das consultas de músicas não identificadas pelos usuários), que agregam as consultas por músicas; no segundo nível, cada grupo de consultas m poderá ter subgrupos $\mathcal{C}_{m,k} \subset \mathcal{C}_m$ que agregam consultas do mesmo grupo entre si por similaridade. Como os usuários podem cantar trechos diferentes de uma mesma música, em um grupo $\mathcal{C}_m$ pode haver vários subgrupos e consultas ainda não agrupadas neles. Para que uma consulta c_i (relativa à música m) seja alocada em um subgrupo, é necessário que esta obtenha uma pontuação de similaridade que ultrapasse determinado limiar $\gamma \in \{0, 100\}$. Este pode assumir valores relativos a três diferentes casos, circunstanciados pela consulta c_j (relativa à música n) com a qual se compara a consulta c_i, segundo a Equação (6.1):

$$\hat{s}_j = \begin{cases} S(c_i, c_j) - \gamma_1, & m = n \text{ (mesma música)} \\ S(c_i, c_j) - \gamma_2, & n = \text{X ou } m = \text{X (uma das músicas não é identificada)} \\ S(c_i, c_j) - \gamma_3, & n \neq m \neq \text{X (músicas diferentes identificadas)}, \end{cases} \tag{6.1}$$

onde $\hat{s}_j$ é o que chamaremos de similaridade relativa.

Os limiares γ (que também podem ser interpretados como penalidades) estão ordenados de forma crescente, de maneira a diminuir a tolerância conforme as circunstâncias apresentadas. No primeiro caso, as consultas já estão ambas alocadas no mesmo grupo de índice m; assim, a probabilidade de ambas serem realmente da mesma música é grande, motivo pelo qual a pontuação relativa a essa comparação é

penalizada com o menor limiar γ_1. Já no segundo caso, não há identificação da música de uma das consultas, o que reduz a probabilidade de serem ambas da mesma música e, portanto, é razoável exigir uma similaridade maior, penalizada por γ_2, para que sejam alocadas no mesmo subgrupo. No último caso, as consultas estão alocadas em grupos de músicas diferentes, o que pode ter acontecido por engano, mas faz com que a probabilidade de tais consultas serem relativas à mesma música seja ainda menor, motivando a escolha da maior penalização γ_3. O motivo para chamar γ de limiar é que, caso a $\hat{s}_j \leq 0$, ou seja, $S(c_i, c_j) \leq \gamma$, a consulta c_j não pode ser candidata a ser agrupada com c_i.

Como todo sistema que utiliza limiares, e este não é o primeiro apresentado aqui com essa característica, o sistema proposto se baseia em dados experimentais para determinar os valores de tais limiares, o que lhe confere certa fragilidade. Embora não se possa afirmar que as configurações utilizadas sejam universais, estas foram baseadas em uma grande variedade de consultas, e a própria diversidade encontrada nelas motivou a arquitetura do sistema, tornando-o mais imune a erros relativos à escolha desses parâmetros. Além disso, o sistema pode, no futuro, vir a adotar mecanismos de adaptação de tais limiares, utilizando os valores encontrados como ponto de partida. Modificar os limiares γ significa tornar o sistema mais permissivo ou mais intolerante com relação às consultas armazenadas, modificando-se o tempo (em termos de uso) que levará para alterar as músicas armazenadas em sua base e com qual tolerância de similaridade esses procedimentos serão feitos. Exigir valores muito altos de similaridade (γ's muito grandes) significa exigir que haja consultas muito boas de músicas ruins (para o uso do sistema), o que é pouco provável, mas previne o sistema contra erros de associação. Por outro lado, adotar valores demasiado baixos para os limiares significa passar a aceitar como similares consultas distintas, o que não fará com que o sistema modifique as músicas de sua base corretamente. É necessário comparar as consultas e agregá-las coerentemente de maneira que se tenha subgrupos dentro dos grupos das músicas com consultas que sejam boas candidatas a substituírem a música da base.

Para definir de maneira mais precisa o funcionamento do sistema, há uma descrição do mesmo no Algoritmo 1.

Algoritmo 1 Sistema de QBH com adaptação automática da base de músicas.

$c_i \leftarrow i$-ésima consulta (música m)
Ranque($\mathcal{M}$) $\leftarrow$ QBH(c_i, $\mathcal{M}$)
Defina os limiares $\{\gamma_1, \gamma_2, \gamma_3, \lambda_{\mathrm{g}}, \lambda_{\mathrm{X}}\} \in \mathbb{R}_+$
se Ranque(e_m) > 10 **então**
    Peça ao usuário que identifique a música m consultada
    se o usuário identificou a música (m^*) **então**
        Aloque c_i no grupo $\mathcal{C}_{m^*}$
        para todo $\{c_{j\neq i} | c_j \in \mathcal{C}_n\}$ **faça**
            Calcule $S(c_i, c_j)$ via Distância de Levenshtein
            Atualizar $\hat{s}_j$ segundo a Equação (6.1)
        fim para
        se $\exists\ \hat{s}_j\, |\hat{s}_j > 0$ **então**
            $j^* \leftarrow \arg\max_{j\in\mathbb{N}} \{\hat{s}_j | \hat{s}_j > 0\}$
            $s_i, s_{j^*} \leftarrow S(c_i, c_{j^*})$
            $v_i, v_{j^*} \leftarrow 0$
            se $n^* \neq$ X (conjunto identificado) **então**
                se c_{j^*} está alocada em algum subgrupo $\mathcal{C}_{n^*,k} \subset \mathcal{C}_{n^*}$ **então**
                    Aloque c_i no mesmo subgrupo $\mathcal{C}_{n^*,k}$
                senão
                    Crie o subgrupo $\mathcal{C}_{n^*,indNovo} \subset \mathcal{C}_{n^*}$
                    Aloque c_i e c_{j^*} no subgrupo $\mathcal{C}_{n^*,indNovo}$
                fim se
            senão
                Crie o subgrupo $\mathcal{C}_{m^*,indNovo} \subset \mathcal{C}_{m^*}$
                Mova c_i e c_{j^*} para $\mathcal{C}_{m^*,indNovo}$
            fim se
        senão
            para todo $\{v_j | c_j \in \mathcal{C}_n$ e $c_j \notin \mathcal{C}_{n,k}\}$ **faça**
                $v_j \leftarrow v_j + 1$
                se $v_j > \lambda_{\mathrm{g}}$ **então**
                    Exclua c_j
                fim se
            fim para
        fim se

senão (o usuário não identificou a música)
    Aloque c_i no grupo $\mathcal{C}_{\mathrm{X}}$
    para todo $\{c_{j\neq i}|c_j \in \mathcal{C}_{n\neq \mathrm{X}}\}$ **faça**
        Calcule $S(c_i, c_j)$ via Distância de Levenshtein
        Atualize $\hat{s}_j$ segundo a Equação (6.1)
    fim para
    se $\exists\ \hat{s}_j|\hat{s}_j > 0$ **então**
        $j^* \leftarrow \arg\max_{j\in\mathbb{N}}\{\hat{s}_j|\hat{s}_j > 0$ e c_j está alocado em grupo $\mathcal{C}_{n\neq \mathrm{X}}\}$
        $s_i \leftarrow S(c_i, c_{j^*})$
        $v_i \leftarrow 0$
        se c_{j^*} está alocada em algum subgrupo $\mathcal{C}_{n^*,k} \subset \mathcal{C}_{n^*}$ **então**
            Mova c_i para $\mathcal{C}_{n^*,k}$
        senão
            Crie o subgrupo $\mathcal{C}_{n^*,indNovo} \subset \mathcal{C}_{n^*}$
            Mova c_i e c_{j^*} para $\mathcal{C}_{n^*,indNovo}$
        fim se
    senão
        para todo $\{v_j|c_j \in \mathcal{C}_{\mathrm{X}}\}$ **faça**
            $v_j \leftarrow v_j + 1$
            se $v_j > \lambda_{\mathrm{X}}$ **então**
                Exclua c_j
            fim se
        fim para
    fim se
fim se
Chame a rotina de modificação da base de músicas

Algoritmo 2 Rotina de modificação da base de músicas.

Defina os limiares $\{\alpha_1, \alpha_2\} \in \mathbb{N}$
Defina o fator multiplicativo $\beta > 1 \in \mathbb{R}_+$
para todo $\mathcal{C}_{n \neq \mathrm{X}}$ **faça**
    se $\mathrm{num}(\mathcal{C}_n) > \alpha_1$ e $\exists\ \mathcal{C}_{n,k} | \mathrm{num}(\mathcal{C}_{n,k}) > \alpha_2$ **então**
        $c_{\mathrm{conc}} \leftarrow \{\}$
        $N \leftarrow \mathrm{num}(\mathcal{C}_n)$
        para todo $\mathcal{C}_{n,k} | \mathrm{num}(\mathcal{C}_{n,k}) > \alpha_2$ **faça**
            $j' \leftarrow \arg\max_{j \in \mathbb{N}} \{s_j | c_j$ está alocado em subgrupo $\mathcal{C}_{n,k}\}$
            $c_{\mathrm{conc}} \leftarrow \mathrm{conc}\,(c_{\mathrm{conc}}, c_{j'})$
            Exclua $\mathcal{C}_{n,k}$
        fim para
        se $N > \beta b_n$ **então**
            $c_{\mathrm{nova}} \leftarrow e_n$
            Alocar c_{nova} em $\mathcal{C}_n$
            $e_n \leftarrow c_{\mathrm{conc}}$
            $b_n \leftarrow 0$
        senão
            $e_n \leftarrow \mathrm{conc}\,(e_n, c_{\mathrm{conc}})$
            $b_n \leftarrow 0$
        fim se
    fim se
fim para

Inicialmente, o usuário faz, à música m, uma consulta c_i, a qual é transcrita e submetida ao sistema de QBH,[1] que em nossa implementação é o sistema Tararira, embora isso seja irrelevante quanto ao funcionamento do sistema proposto. O sistema de QBH recebe como entradas a i-ésima consulta c_i e a base de músicas, representada pelo conjunto de melodias $\mathcal{M} = \{e_1, e_2, ..., e_m, ..., e_M\}$, em que e_m é o elemento que contém a melodia da música consultada. Em seguida o sistema retorna o Ranque($\mathcal{M}$). São definidos os limiares γ_1, γ_2, γ_3, λ_{g} e λ_{X} e o usuário responde ao sistema se a música consultada está entre as primeiras do ranque.

Caso a música consultada não esteja dentre as primeiras do ranque, o usuário é convidado a identificar a música consultada. Aqui convém ressaltar a extrema importância desse procedimento para o bom funcionamento do sistema. Caso não haja nenhum usuário que saiba indicar ao sistema a música que foi consultada nesses casos de resultados ruins, o sistema jamais será capaz de corrigir músicas cujos

[1] Na prática, o sistema de Tararira transcreve a consulta e a transcrição é utilizada pelo sistema proposto.

resultados sejam sempre ruins; no entanto, não é necessário que todos o informem, vantagem que é obtida devido aos procedimentos de agrupamento de consultas. Ademais, é com este procedimento que o sistema consegue verter resultados ruins das consultas em bons elementos para a base de músicas, o que aparentemente seria um contrassenso.

No caso de o usuário identificar a música (e essa identificação pode estar errada), a consulta c_i é alocada no grupo relativo à música m^* identificada. Em seguida, a consulta c_i é comparada via Distância de Levenshtein com todas as outras consultas guardadas c_j anteriormente pelo sistema, caso existam. Os resultados obtidos são armazenados em $\hat{s}_j$, segundo a Equação (6.1), que leva em conta os grupos em que se encontram as consultas. A formulação criada para a $\hat{s}_j$ tem por objetivo penalizar, através dos limiares γ, os valores de similaridade de maneira a selecionar posteriormente a que ultrapassou seu limiar imposto por mais pontos. Isso é feito escolhendo-se $\hat{s}_{j^*}$, que é o maior dentre os $\hat{s}_j > 0$. Assim, escolhe-se a consulta c_{j^*} melhor candidata a ser agrupada com a consulta atual, o valor de similaridade $S(c_i, c_{j^*})$ é guardado e o contador de validade da consulta recebe 0. Tal contador servirá para excluir a consulta caso fique sem ser agrupada por muito tempo, evitando o crescimento desnecessário da base de consultas armazenadas pelo sistema. Caso nenhum $\hat{s}_j$ ultrapasse o valor 0, quer dizer que nenhuma consulta foi similar o bastante para o agrupamento, considerando suas alocações atuais.

Após haver identificado a melhor consulta c_{j^*}, caso esta esteja alocada em algum grupo relativo à música n^* (conjunto identificado) e esta não seja a música m^* indicada pelo usuário, a consulta c_i é movida para o mesmo subgrupo $\mathcal{C}_{n^*,k} \subset \mathcal{C}_{n^*}$ da consulta c_{j^*}; caso este subgrupo não exista ainda, é criado $\mathcal{C}_{n^*,indNovo}$ com um índice que garanta sua unicidade no grupo e ambas as consultas c_i e c_{j^*} são alocadas nele. Para o caso de a consulta c_{j^*} estar no grupo $\mathcal{C}_{\mathrm{X}}$, um novo subgrupo $\mathcal{C}_{m^*,indNovo} \subset \mathcal{C}_{m^*}$ é criado e ambas as consultas c_i e c_{j^*} são alocadas nesse subgrupo. Caso nenhuma consulta seja escolhida para o agrupamento, a consulta c_i permanece no grupo $\mathcal{C}_{m^*}$ e os contadores de validade de todas consultas de $\mathcal{C}_{m^*}$ que não pertençam a algum subgrupo são iterados; caso algum v_j ultrapasse o limite de $\lambda_{\mathrm{g}} = 10$, a consulta c_j é excluída. Com isso, v conta as vezes em que chegaram consultas ao grupo e não houve nenhuma outra similar o bastante para agrupá-la; se isso acontece muito frequentemente, é provável que as consultas que ficaram esse tempo sem serem agrupadas sejas inúteis.

Considerando o caso em que o resultado da consulta seja ruim e o usuário não informe ao sistema qual música foi consultada, a consulta c_i é alocada no grupo $\mathcal{C}_{\mathrm{X}}$, destinado às consultas dessa natureza. Em seguida, a consulta c_i é comparada, assim como explicado anteriormente, com todas as demais consultas c_j, mas apenas as que estejam armazenadas em grupos $\mathcal{C}_{n\neq\mathrm{X}}$, evitando criar subgrupos em $\mathcal{C}_{\mathrm{X}}$; e

são computadas as similaridades relativas $\hat{s}_j$, sendo que sempre será aplicada a penalidade γ_2 pelo fato de $c_i \in \mathcal{C}_{\mathrm{X}}$. Depois, é selecionado o $\hat{s}_{j^*}$ com os mesmos critérios já expostos e é guardada a similaridade $S(c_i, c_{j^*})$ apenas para s_i, mantendo o s_{j^*} original inalterado, e o contador de validade v_i é iterado. O motivo para não modificar s_{j^*} consiste em c_{j^*} ser a consulta responsável por atrair c_i, que não é uma consulta tão confiável por estar alocada em $\mathcal{C}_{\mathrm{X}}$. As consultas são alocadas no mesmo subgrupo $\mathcal{C}_{n^*,k}$ ao qual pertence c_{j^*}, podendo gerar um novo $\mathcal{C}_{n^*,indNovo}$, no caso de ainda não existir $\mathcal{C}_{n^*,k}$. Quando não há c_{j^*} selecionada, são iterados todos os v_j relativos às consultas $c_j \in \mathcal{C}_{\mathrm{X}}$ e são excluídas as consultas c_j cuja validade expirou i.e., $v_j > \lambda_{\mathrm{X}} = 4{\times}\mathrm{num}(\mathcal{M})$.[2]

O último caso avaliado ocorre quando o resultado da consulta é bom (está entre as Top-10). Nesse caso, itera-se um contador de bons resultados b_m para a música consultada. Ao fim, sempre é chamada a rotina de modificação da base de músicas, cujos procedimentos estão resumidos no Algoritmo 2.

Primeiro, são definidos os limiares α_1 e α_2 e o fator β. Em seguida, é feita uma varredura por todos os grupos de consultas $\mathcal{C}_{n\neq\mathrm{X}}$ e, caso o número de elementos tenha ultrapassado um limite inferior $\alpha_1 = 6$ e haja algum subgrupo com mais de $\alpha_2 = 4$ consultas, então o sistema realiza os procedimentos para a correção de e_n. O sistema elege como melhor consulta c_j (de cada subgrupo com mais de α_2 consultas) a que tiver maior pontuação s_j, e essas diversas consultas elegidas são concatenadas, formando uma melodia única c_{conc}. Em seguida, todos os subgrupos utilizados para esse procedimento são excluídos com suas consultas, uma vez que cada subgrupo representa diferentes trechos da música e cada um já fora contemplado com sua consulta representante. Para a correção do elemento e_n da base de músicas, é verificada a relevância em manter-se e_n pelo número de boas consultas obtidas com ele. Caso o número de consultas ruins, representado pelo número de consultas no grupo $\mathcal{C}_n$ antes da exclusão das consultas, seja maior que o número de boas consultas b_n multiplicado por um fator $\beta = 3$, então o elemento e_n é armazenado em $c_{nova} \in \mathcal{C}_n$ e inteiramente substituído por c_{conc}. Caso contrário, e_n ainda pode ser útil, podendo haver trechos bem transcritos, e concatena-se e_n com c_{conc}. Em ambos os casos, a contagem de boas consultas b_n é reinicializada com 0. Note que, mesmo que a transcrição original e_n seja substituída, esta ainda é armazenada em seu grupo respectivo como uma consulta qualquer, e poderá ser útil no futuro formando novos subgrupos. Caso seja realmente inútil, seu contador de validade irá aumentar gradativamente até chegar ao ponto de expirar ($v_j > \lambda_g = 10$), e ela será excluída, poupando processamento do sistema.

[2] O limiar λ_{X} é consideravelmente maior que λ_{g}, pois há muito maior diversidade de músicas relativas às consultas em $\mathcal{C}_{\mathrm{X}}$, e essa diversidade tende a ser proporcional a num($\mathcal{M}$). O fator 4 utilizado é mera convenção, podendo ser utilizado qualquer outro, bastando definir o quanto se quer armazenar de consultas no sistema.

Com exceção dos fatores de penalização γ, todos os outros fatores e limiares não têm grande impacto no desempenho do sistema, motivo pelo qual foram fixados em valores que pareceram razoáveis. Nos experimentos foram testadas duas diferentes configurações de γ's, sendo uma com valores mais baixos e outra mais conservadora com valores mais altos. Os limiares α influenciam na rapidez com que o sistema realiza a correção do elemento da base de músicas, mas acima de certo valor praticamente não influencia na qualidade das correções feitas.

6.3 Experimentos: Procedimentos, Resultados e Discussões

Nesta seção, serão mostrados os resultados relativos aos experimentos realizados para três diferentes base de músicas do conjunto I (músicas brasileiras): MIDI, como referência de limite superior de qualidade de transcrição das músicas e para testar o desempenho do método proposto quando utiliza uma base com essa qualidade; e as bases GPA e Melodia, para testar o desempenho do sistema com bases que tenham diversas músicas mal transcritas.

Foram utilizadas duas diferentes configurações de limiares γ do sistema, sendo elas:

Configuração 1: $\{\gamma_1 = 42, \gamma_2 = 50, \gamma_3 = 60\}$;

Configuração 2: $\{\gamma_1 = 45, \gamma_2 = 55, \gamma_3 = 70\}$.

A primeira configuração representa um grau de exigência menor para o agrupamento das consultas, o que pode gerar erros, conforme discutido anteriormente; a segunda configuração é mais conservadora, reduzindo os erros de agrupamento de consultas.

Como o sistema depende fortemente da ordem com que aparecem as consultas, concluiu-se que seria necessário submetê-las ao sistema em ordenações aleatórias, de maneira a comparar o desempenho do sistema nessas diferentes situações. Assim, foram geradas e armazenadas três diferentes ordens de consultas, as quais foram utilizadas para cada diferente configuração das simulações realizadas.

Foram realizadas duas diferentes categorias de experimentos, visando a estressar o sistema de diferentes perspectivas: (i) variando-se a probabilidade de os usuários informarem a música (sempre corretamente), com as três diferentes realizações da ordenação das consultas; e (ii) com os usuários sempre informando a música, quando os resultados fossem ruins, mas tendo uma chance de 70% de acerto ao informá-la. A primeira categoria das simulações objetiva aferir o quanto influencia a quantidade de informação, sempre correta, que o sistema recebe sobre as músicas ruins da base, enquanto a segunda busca analisar o quão robusto a erros nessas informações o sistema está, embora haja mecanismos no algoritmo que buscam mitigar tais erros. Deseja-se saber, por exemplo, se o sistema seria capaz de manter uma base bem transcrita, como a MIDI, mesmo com os usuários podendo informar errado as músicas dos maus resultados, ou mesmo se a base poderia ser melhorada em tais situações; deseja-se também aferir o quanto, e quão rapidamente, o sistema consegue corrigir bases mal transcritas.

Uma vez que os experimentos realizados fornecem um conjunto demasiado extenso para ser reportado aqui, estes foram expostos por completo no Apêndice A e

aqui apenas a parte mais relevante será mostrada e discutida em termos gerais. Serão apresentados, principalmente, os resultados obtidos com a segunda configuração dos limiares γ, que mostrou-se mais adequada para todas as situações simuladas.

6.3.1 Experimento 1 — Diferentes Probabilidades de Informar-se a Música Consultada

Primeiramente, deseja-se aferir se o método proposto é capaz de corrigir uma base de músicas mal transcrita. Para isso, foram escolhidas as duas bases cujos piores resultados foram obtidos com o sistema Tararira, sendo elas as bases Melodia e GPA. Sabe-se que a base GPA-2 melhora significativamente os resultados obtidos com esse transcritor, mas a base GPA fornece um conjunto de teste mais apropriado para o objetivo que temos aqui.

As Tabelas 6.3 e 6.4 resumem os resultados obtidos com as bases GPA e Melodia, respectivamente, para o experimento citado com o sistema utilizando a configuração 2 de limiares γ, para diferentes probabilidades de os usuários identificarem para o sistema a música consultada.

Tabela 6.3: Resultados obtidos com a base GPA para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,29	22,24	34,29	41,73	1
	0,33	25,20	39,90	47,24	2
	0,27	18,47	32,65	39,80	3
60	0,39	30,92	45,41	52,96	1
	0,38	29,29	44,39	52,96	2
	0,41	33,47	48,06	55,61	3
100	0,49	40,82	55,41	63,37	1
	0,43	35,10	50,20	59,39	2
	0,45	37,14	52,35	60,92	3
Tararira	0,16	8,27	20,51	27,35	—

Tabela 6.4: Resultados obtidos com a base Melodia para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,42	32,77	52,00	61,09	1
	0,36	26,27	43,63	53,25	2
	0,37	27,43	44,61	54,23	3
60	0,41	31,61	50,76	60,55	1
	0,41	31,26	50,58	60,55	2
	0,37	26,71	47,37	56,19	3
100	0,50	40,25	59,66	69,01	1
	0,43	32,50	52,00	62,33	2
	0,44	34,28	54,14	63,67	3
Tararira	0,24	14,16	30,28	39,72	—

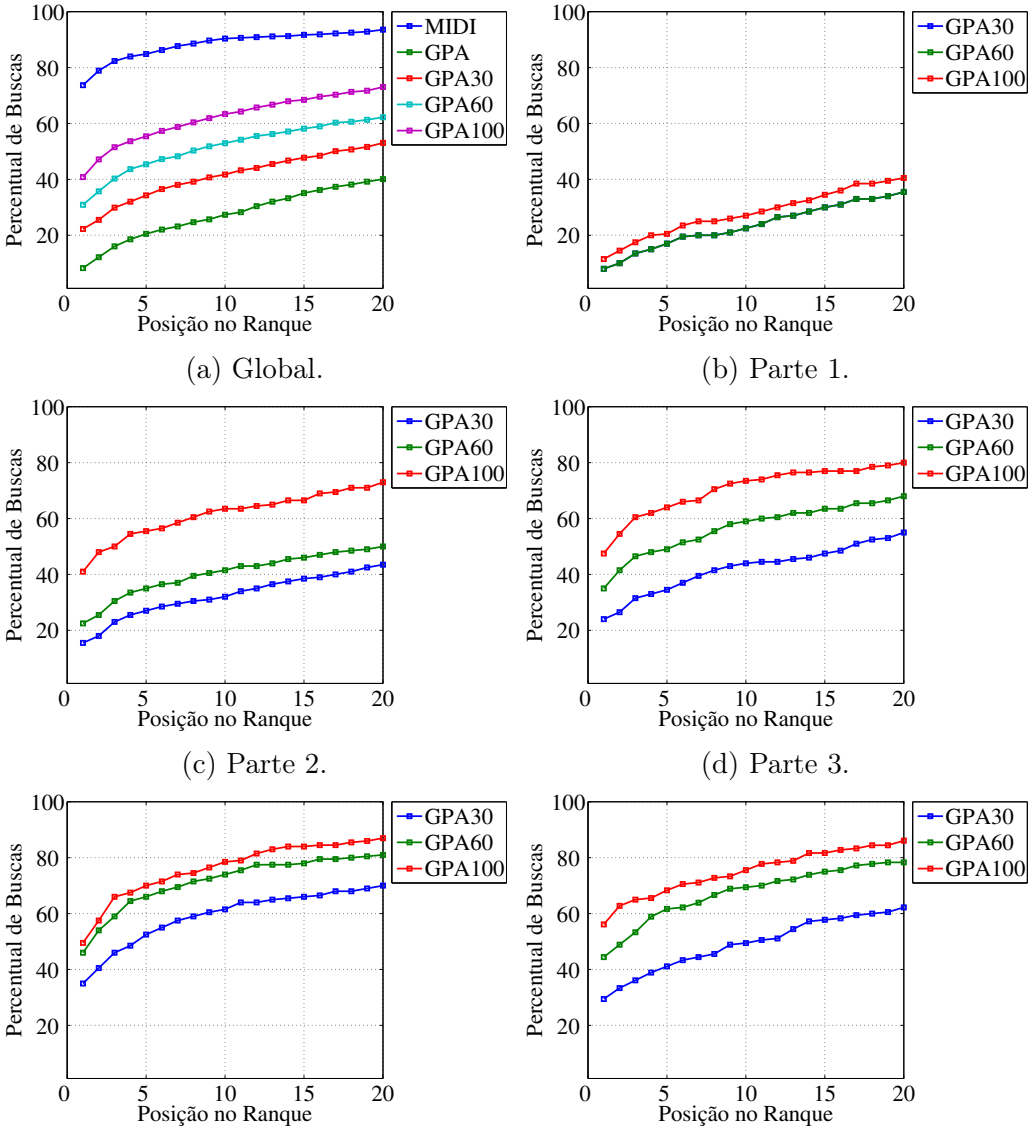
O primeiro fato a ser observado é que, realmente, quanto mais usuários informam ao sistema qual a música buscada, maior a chance de o sistema melhorar sua base de músicas. Uma outra constatação é a confirmação de que a ordem com que as consultas são feitas interfere na maneira como o sistema adapta sua base de músicas. Caso apareçam consultas ruins, mas boas o bastante para corrigirem a música correspondente, pode ser que o sistema melhore seu desempenho, mas não atinja resultados tão bons, que poderiam aparecer caso uma consulta melhor fosse feita primeiro. Das três realizações de ordem, nota-se que a primeira é a que fornece os melhores resultados globais. Embora a ordem seja salva em um arquivo e todos os experimentos as utilizem, importa ressaltar que a chance de o usuário responder ao sistema é sempre sorteada com a probabilidade indicada, podendo sempre ser diferente para cada realização. Porém, se fosse essa a causa da variação dos resultados, não haveria motivo para que os resultados variassem com a ordem quando 100% dos usuários identificam as músicas quando solicitados.

Os resultados mostrados abrangem todas as consultas realizadas e mostram que o sistema, no caso da base GPA, conseguiu modificar as músicas da base a ponto de melhorar os resultados globais de acerto nas Top-10 de cerca de 27% para, em média, 61% quando todos os usuários informaram ao sistema. Quando 60% dos usuários informou, o acerto nas Top-10 passou de 27% para aproximadamente 54%, que é o dobro. Para a base Melodia, a melhora foi de cerca de 40% para 65%, com todos os usuários informando, e 59% quando 60% dos usuários informou.

Convém notar que o sistema precisa que as consultas sejam realizadas para que,

após algumas consultas serem armazenadas, combinadas e utilizadas para corrigir as músicas da base, os resultados das consultas melhorem. Até que isso ocorra, para cada música que precisa ser corrigida haverá resultados ruins. Essa fase inicial de convergência do método ocorre inevitavelmente e contabilizá-la nos resultados globais significa não ter uma noção mais precisa de quão boas podem ficar as bases corrigidas. Para isso, é preciso realizar uma análise por blocos de consultas ao longo do tempo, acompanhando a evolução do desempenho do sistema.

Assim, buscou-se realizar tal verificação de maneira visual, colocando em ranques cumulativos blocos de 200 em 200 consultas. Nas Figuras 6.1 e 6.2 estão ilustrados tais resultados para as simulações realizadas com as bases GPA e Melodia, mostrando a evolução do sistema. Há também os resultados globais, incluindo os obtidos com a base MIDI, para referência de limite superior, e com a própria base sem o sistema adaptativo. Esses resultados são relativos à realização 1 da ordenação das consultas.

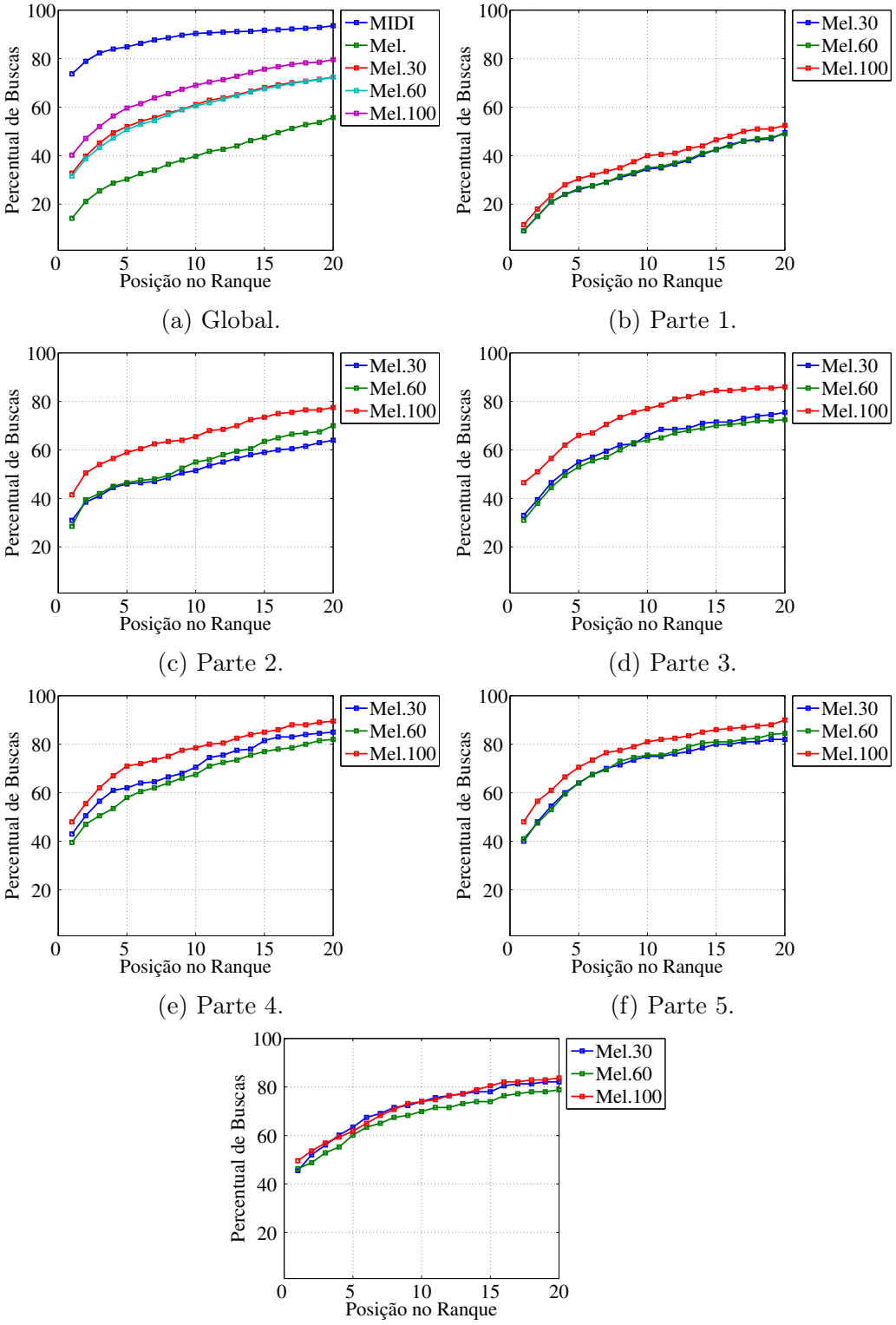


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura 6.1: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura 6.2: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.

Note que há 5 partes nas figuras relativas à base GPA, enquanto há 6, para a

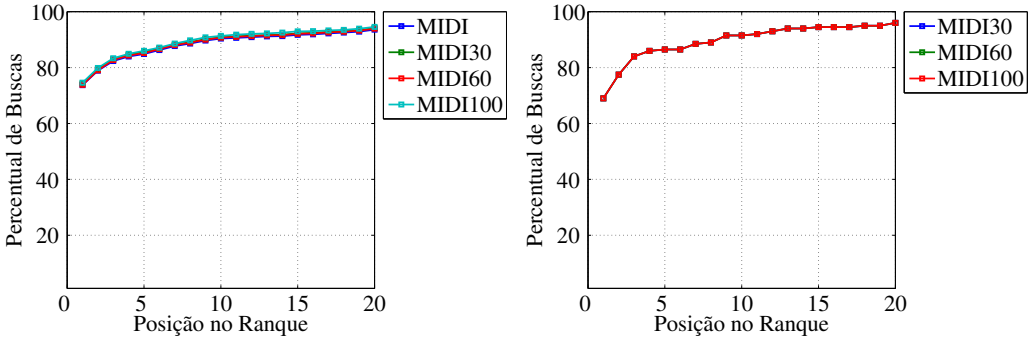
base Melodia, pois há músicas que não puderam ser transcritas na primeira, conforme explicado na Seção 4.1. De posse dessas figuras, nota-se que cada base configura uma situação distinta. De maneira geral, a base GPA originalmente estava em condições de uso piores que a base Melodia; note que, não apenas os resultados globais são piores, mas há clara discrepância entre as curvas com diferentes probabilidades de informação para a base GPA. Enquanto isso, as curvas de ranques cumulativos para a base Melodia tendem a convergir para valores próximos, talvez por ter menos elementos ruins na base de músicas.

É possível notar também que os resultados, embora melhorem significativamente, praticamente não passaram de cerca de 80% de acerto nas Top-10 em nenhum bloco de consultas, enquanto os resultados para a base MIDI alcançam 90% nas Top-10. Como as consultas que substituem as músicas da base não precisam ser perfeitas (e não seria nem mesmo possível o sistema saber quando são), é razoável que não se alcancem resultados tão próximos dos obtidos com a base MIDI, que podem ser considerados como limite superior.

Os resultados obtidos com as simulações utilizando a base MIDI estão dispostos na Tabela 6.5, e na Figura 6.3 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

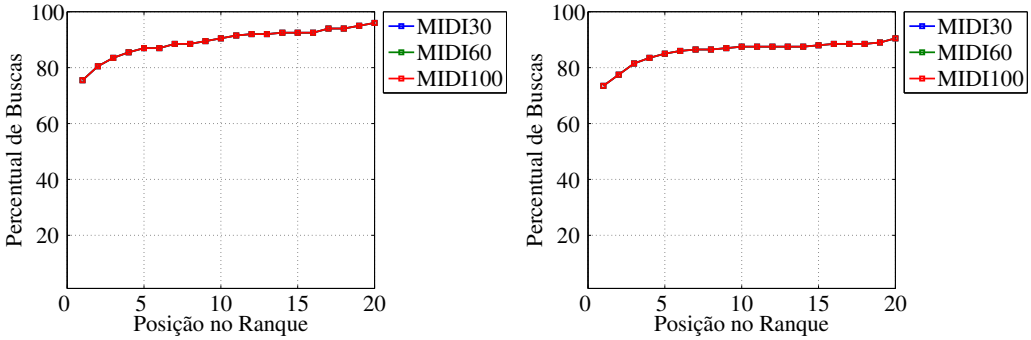
Tabela 6.5: Resultados obtidos com a base MIDI para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,80	74,27	85,57	90,83	1
	0,80	74,18	85,13	90,56	2
	0,80	74,09	85,22	90,65	3
60	0,79	73,82	85,22	90,74	1
	0,80	74,80	85,93	91,36	2
	0,80	74,27	85,40	90,83	3
100	0,80	74,62	86,02	91,36	1
	0,80	74,53	85,84	91,36	2
	0,81	75,24	86,02	91,45	3
Tararira	0,79	73,73	84,86	90,38	—



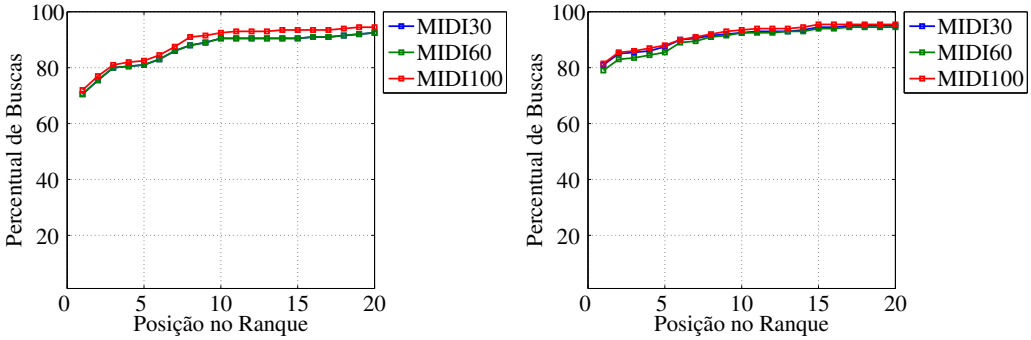
(a) Global.

(b) Parte 1.



(c) Parte 2.

(d) Parte 3.



(e) Parte 4.

(f) Parte 5.

Figura 6.3: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.

Os resultados demonstram que o sistema, além de não piorar o desempenho quando utilizou uma base de músicas já bem transcritas, ainda obteve resultados um pouco melhores em todas as simulações realizadas, o que demonstra que o sistema não deteriora a base original. Se essa base for equivalente a uma base ruim que foi melhorada, isso demonstra que o sistema não tende a piorar com o uso uma vez que já tenham obtidos bons resultados.

6.3.2 Experimento 2 — Usuários Podendo Errar ao Informarem a Música

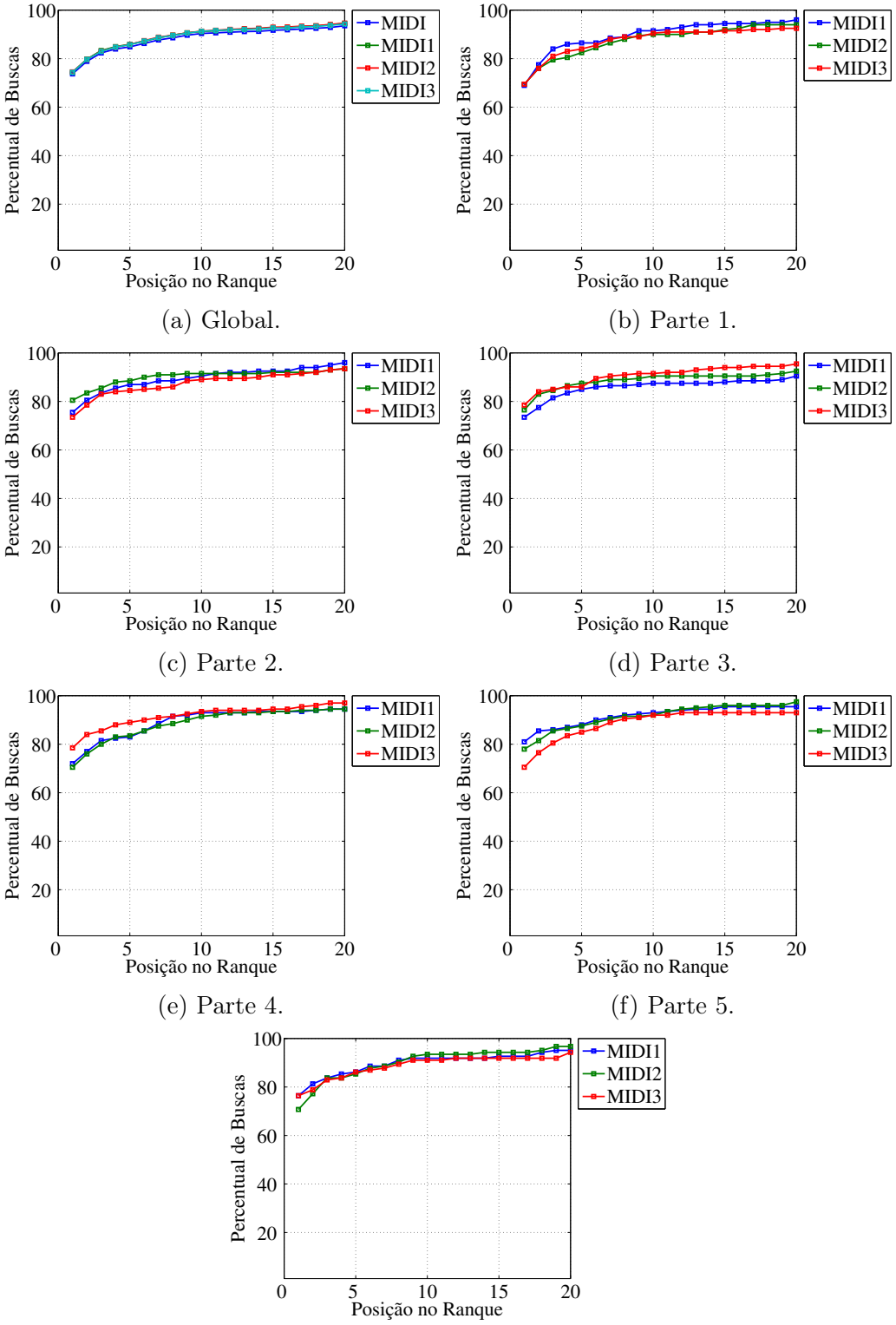
Para que se possa testar a robustez do sistema a erros de identificação das músicas por parte dos usuários, foram realizadas as simulações da categoria (ii), em que os usuários sempre informam ao sistema a música consultada, quando solicitados, mas podem errar ao informá-la. Assim, são marginalizadas as demais probabilidades consideradas no outro experimento e é contabilizado apenas o efeito do erro de identificação. Isso significa que 30% das consultas que o sistema armazenar poderão ser alocadas em grupos de músicas que não são as músicas consultadas.

O mecanismo que permite o sistema corrigir tal erro é possibilitar que consultas de grupos diferentes sejam agrupadas num mesmo subgrupo, fazendo com que a consulta mais recente seja movida de grupo. Caso a consulta tenha entrado no grupo errado por erro de identificação, é improvável que esta se agrupe com outras desse grupo e, mesmo que o faça, é ainda mais improvável que seja escolhida como a melhor consulta do subgrupo para formar a melodia da música a ser corrigida.

A Tabela 6.6 resume os resultados obtidos com a base MIDI, enquanto a Figura 6.4 ilustra os mesmos resultados, com a evolução do sistema a cada 200 consultas para as 3 diferentes realizações de ordem das consultas.

Tabela 6.6: Resultados obtidos com a base MIDI com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,80	74,44	85,93	91,18	1
	0,80	74,53	85,84	91,36	2
	0,80	74,35	85,75	91,27	3
Tararira	0,79	73,73	84,86	90,38	—



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura 6.4: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.

É possível observar que, mesmo com os usuários equivocadamente informando a música consultada em 30% das vezes, o sistema ainda obteve resultados melhores que os originais. Novamente, convém ressaltar que o sistema depende diretamente dessas informações fornecidas pelos usuários e que, se todos os usuários informarem a música errada, será impossível corrigir a base de músicas.

Para as bases GPA e Melodia, o sistema também obteve bons resultados. As Tabelas 6.7 e 6.8 resumem os resultados obtidos com essas bases para esse experimento, e as Figuras 6.5 e 6.6 ilustram os mesmos resultados com a evolução do sistema a cada 200 consultas.

Tabela 6.7: Resultados obtidos com a base GPA com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

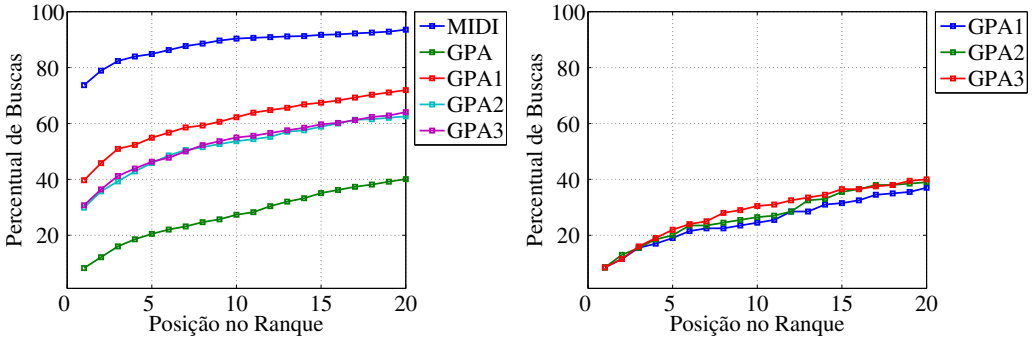
Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,48	39,69	54,90	62,24	1
	0,38	29,90	45,82	53,67	2
	0,39	30,71	46,33	55,00	3
Tararira	0,16	8,27	20,51	27,35	—

Tabela 6.8: Resultados obtidos com a base Melodia com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,45	33,66	53,16	62,69	1
	0,39	29,03	47,64	56,72	2
	0,37	26,71	45,86	54,59	3
Tararira	0,24	14,16	30,28	39,72	—

Importa destacar também que os resultados obtidos com a outra configuração de limiares, embora inferiores aos apresentados, também demonstram grande evolução do desempenho, demonstrando que o sistema ainda é capaz de melhorar a base de músicas mesmo em configurações subótimas.

Portanto, conclui-se que os experimentos apontam para a viabilidade do uso do sistema, demonstrando que, para as amostras disponíveis, foi possível melhorar consideravelmente o desempenho da base de músicas para os resultados globais, podendo atingir cerca de 80% de acerto nas Top-10 após um período inicial de coleta de consultas para a correção das músicas da base. Salienta-se ainda o fato de que muitas músicas contidas nas bases do conjunto de bases de teste I não contêm consultas o



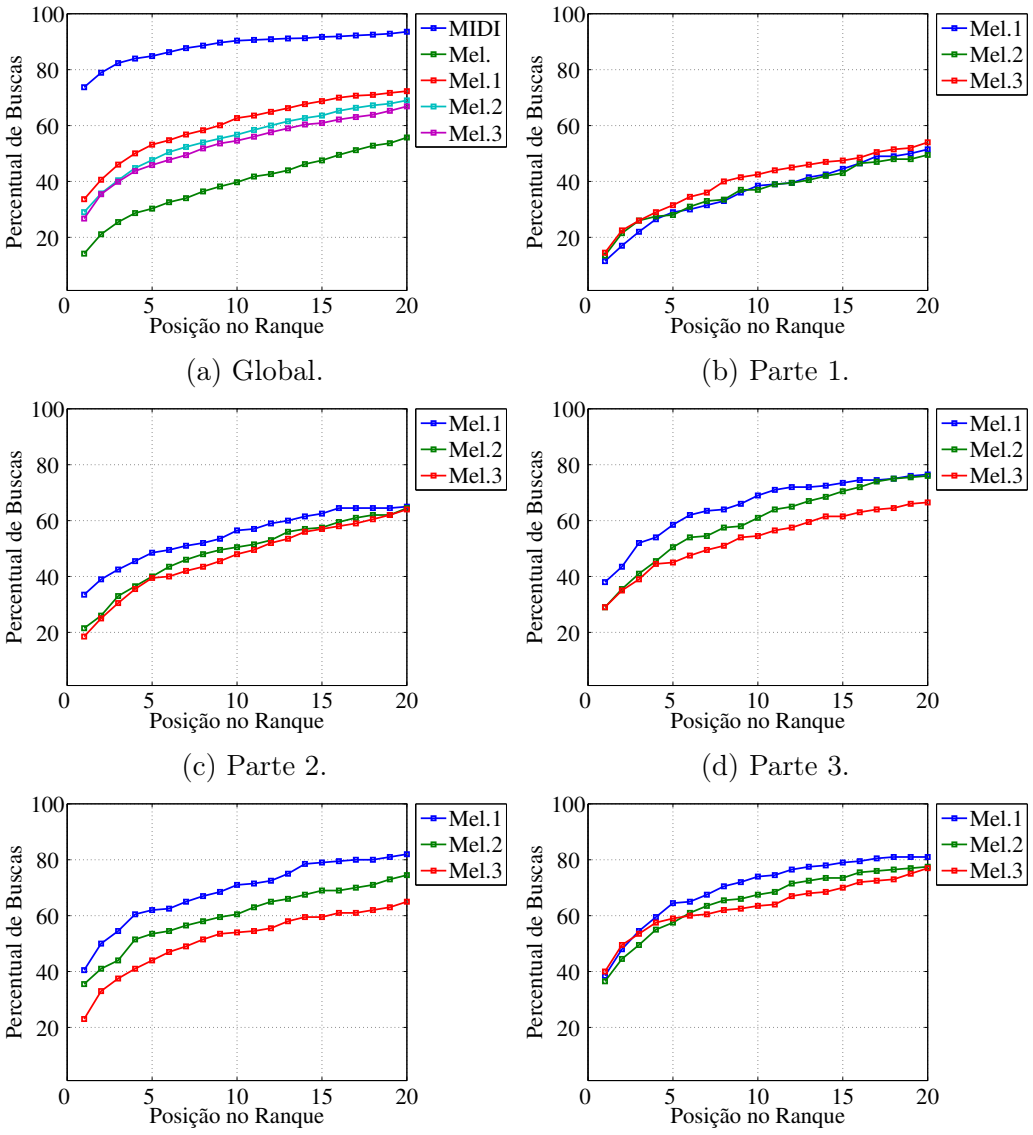
(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura 6.5: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.

bastante para que haja possibilidade de correção e bons resultados posteriores. Com isso, podemos inferir que os melhores resultados atingidos ainda poderiam ser ultrapassados caso houvesse uma base de consultas mais completa, uma vez que essas músicas mal transcritas e com poucas consultas sempre contribuem com resultados ruins.



(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura 6.6: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.

Capítulo 7

Conclusões

Neste trabalho foi feito um estudo sobre a teoria e a prática de sistemas de consulta cantarolada, em especial o sistema Tararira, que foi utilizado como plataforma para os experimentos realizados. Em seguida, foram expostos dois dos principais sistemas de transcrição automática de melodia em sinais polifônicos da literatura: Melodia e GPA-UdelaR. Estes dois sistemas foram utilizados para transcrever automaticamente bases de músicas a partir de suas gravações comerciais, e foram realizados testes de desempenho do sistema Tararira com tais bases. Como os resultados demonstraram que os sistemas de transcrição automática ainda não são capazes de fornecer resultados satisfatórios para todos os tipos de música (cerca de 40% da base estavam aproveitáveis), optou-se por desenvolver um sistema que adaptasse a base de músicas gradativamente conforme os usuários realizam as consultas. O sistema proposto foi detalhado e foram discutidos os conceitos que o embasam, de maneira a buscar motivar cada parte do algoritmo criado. Por fim, foram realizados diversos experimentos com o sistema proposto de adaptação automática da base de músicas. Os resultados confirmam que, ao menos para os conjuntos de amostras testados, o sistema foi capaz de melhorar consideravelmente o desempenho do sistema de QBH com a adaptação progressiva de sua base de músicas. Ao observar-se o desempenho ao longo das consultas, foi possível notar que, a base de música GPA, que oferecia um desempenho ao sistema de 27% de acerto nas Top-10, passou a acertar até 80%, após algumas modificações da base.

Como já foi discutido anteriormente, o maior problema relacionado aos sistemas de QBH atualmente se encontra na dificuldade de criação do banco de músicas no qual são realizadas as consultas. O sistema criado se propõe a mitigar tal problema, partindo de uma base extraída automaticamente de gravações comerciais das músicas, que pode conter diversos elementos mal transcritos, e adaptá-los conforme as consultas dos usuários. Salienta-se o fato de que o sistema proposto necessita de informações fornecidas pelos usuários para obter bons resultados, mas que, dentro de certos limites, consegue lidar mesmo com informações erradas recebidas. Assim,

pode-se concluir que o sistema criado atingiu o objetivo de lidar de maneira automática com a questão da escalabilidade de sistemas de QBH, tornando desnecessário qualquer esforço manual de criação da base, ao custo de, inicialmente, as músicas mal transcritas apresentarem maus resultados e alguns usuários precisarem indicar ao sistema a música que foi cantada na consulta.

Capítulo 8

Trabalhos Futuros

Como trabalhos que podem continuar a análise de desempenho do sistema, propõe-se, primeiramente, a criação de uma base de músicas e consultas mais representativa do uso do sistema na prática, em que haja ruído ambiente nas consultas, com os usuários podendo cantar qualquer trecho da música com ou sem letra e com as músicas sendo transcritas por completo para comporem a base do sistema. Com isso, os resultados obtidos tenderão a ser mais realistas enquanto simulação do uso do sistema.

Como desenvolvimento do sistema proposto, é possível incluir mecanismos de adaptação dos próprios limiares utilizados, podendo até atribuir valores específicos de γ por grupo, como complemento às três situações avaliadas, baseando-se no fato de que as diferentes músicas possuem diferentes valores de similaridade média entre suas consultas. Outra modificação possível é tornar o sistema capaz de analisar trechos de melodia comuns entre as consultas e formar uma melodia média representativa, ou apenas corrigir os trechos das músicas mal transcritos. Há também a ideia de, na comparação de melodias, buscar uma maneira de identificar os trechos das músicas que melhor as representem e as distinga das demais, dando ênfase a eles na pontuação de similaridade. Como última proposta, pretende-se experimentar novas ideias de algoritmos para comparação de melodias que façam uso da análise frequencial das funções de F0.

Referências Bibliográficas

[1] PARDO, B., SHIFRIN, J., BIRMINGHAM, W. "Name that tune: A pilot study in finding a melody from a sung query", *Journal of the American Society for Information Science and Technology*, v. 55, n. 4, pp. 283–300, Dezembro 2004.

[2] PARDO, B., LITTLE, D., JIANG, R., et al. "The VocalSearch music search engine". In: *Proceedings of the 8th ACM/IEEE Joint Conference on Digital Libraries (JCDL 2008)*, pp. 430–430, Nova Iorque, EUA, Junho 2008.

[3] SONG, J., BAE, S. Y., YOON, K. "Mid-Level music melody representation of polyphonic audio for query-by-humming system". In: *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 133–139, Paris, França, Outubro 2002.

[4] DUDA, A., NÜRNBERGER, A., STOBER, S. "Towards query by singing/humming on audio databases". In: *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pp. 331–334, Viena, Áustria, Setembro 2007.

[5] RYYNANEN, M., KLAPURI, A. "Query by humming of MIDI and audio using locality sensitive hashing". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'08)*, pp. 2249–2252, Las Vegas, EUA, Março 2008.

[6] SALAMON, J., SERRÀ, J., GÓMEZ, E. "Tonal representations for music retrieval: from version identification to query-by-humming", *International Journal of Multimedia Information Retrieval*, v. 2, n. 1, pp. 45–58, Março 2013.

[7] LÓPEZ, E., ROCAMORA, M. "Tararira: Query by singing system". In: *The 2nd Annual Music Information Retrieval Evaluation eXchange (MIREX 2006), Abstract Collection*, pp. 80–83, Champaign, EUA, Outubro 2006. Extended abstract.

[8] ROCAMORA, M., CANCELA, P., PARDO, A. "Query by humming: automatically building the database from music recordings", *Pattern Recognition Letters*, v. 36, n. 1, pp. 272–280, Janeiro 2014.

[9] SALAMON, J., GÓMEZ, E. "Melody extraction from polyphonic music signals using pitch contour characteristics", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 20, n. 6, pp. 1759–1770, Agosto 2012.

[10] LÓPEZ, E., ROCAMORA, M., SOSA, G. *Búsqueda de música por tarareo*. Projeto de Graduação, Instituto de Ingeniería Eléctrica/Faultad de Ingenieria/Universidad de la República Oriental del Uruguay, Montevidéu, Uruguai, 2004.

[11] YOUNG, R. W. "Terminology for logarithmic frequency units", *The Journal of the Acoustical Society of America*, v. 11, n. 1, pp. 134–139, Maio 1939.

[12] SCHROEDER, M. R. "Period histogram and product spectrum: new methods for fundamental frequency measurement", *Journal of the Acoustical Society of America*, v. 43, n. 4, pp. 829–834, Abril 1968.

[13] NOLL, A. M. "Cepstrum pitch determination", *Journal of the Acoustical Society of America*, v. 2, n. 41, pp. 293–309, Fevereiro 1967.

[14] DE CHEVEIGNÉ, A., KAWAHARA, H. "YIN, a fundamental frequency estimator for speech and music", *Journal of the Acoustical Society of America*, v. 111, n. 4, pp. 1917–1930, Abril 2002.

[15] UPPGARD, S. *Implementation and analysis of pitch tracking algorithms*. Tese de Mestrado, Departament of Signals, Sensors and Systems, Kungliga Tekniska Högskolangskolan, Estocolmo, Suécia, 2001.

[16] ROSÃO, C., RIBEIRO, R. "Trends in onset detection". In: *Proceedings of the Workshop on Open Source and Design of Communication (OSDOC'11)*, pp. 75–81, Lisboa, Portugal, Julho 2011.

[17] BELLO, J. P., DAUDET, L., ABDALLAH, S. "A tutorial on onset detection in music signals", *IEEE Transactions on Speech and Audio Processing*, v. 13, n. 5, pp. 1035–1047, Setembro 2005.

[18] DIXON, S. "Onset detection revisited". In: *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-06)*, pp. 133–137, Montreal, Canadá, Setembro 2006.

[19] KLAPURI, A. P. "Sound onset detection by applying psychoacoustic knowledge". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'99)*, v. 6, pp. 3089–3092, Phoenix, EUA, Março 1999.

[20] SCHEIRER, E. D. "Tempo and beat analysis of acoustic musical signals", *Journal of the Acoustical Society of America*, v. 103, n. 1, pp. 588–601, Janeiro 1998.

[21] KLAPURI, A. P. *Automatic transcription of music.* Tese de Mestrado, Institute of Signal Processing/Tampere Univerity of Technology, Tampere, Finlândia, 1997.

[22] POLLASTRI, E. *Processing singing voice for music retrieval.* Tese de Doutorado, Facoltà di Scienze Matematiche, Fisiche e Naturali, Università degli Studi di Milano, Milan, Italy, 2003.

[23] VIITANIEMI, T., KLAPURI, A., ERONEN, A. "A probabilistic model for the transcription of single-voice melodies". In: *Proceedings of the Finnish Signal Processing Symposium*, pp. 59–63, Tampere, Finlândia, Maio 2003.

[24] MCNAB, R. J., SMITH, L. A., WITTEN, I. H., et al. "Towards the digital music library: tune retrieval from acoustic input". In: *Proceedings of the 1st ACM International Conference on Digital Libraries (DL'96)*, pp. 11–18, Bethesda, EUA, Março 1996.

[25] PHIWMA, N., SANGUANSAT, P. "An improved melody contour feature extraction for query by humming", *International Journal of Computer Theory and Engineering*, v. 2, n. 4, pp. 1793–8201, Agosto 2010.

[26] SHIFRIN, J., BIRMINGHAM, W. P. "Effectiveness of HMM-based retrieval on large databases". In: *Proceedings of the Fourth International Conference on Music Information Retrieval*, Baltimore (Maryland), EUA, Março 2003.

[27] TSAI, W.-H., TU, Y.-M., MA, C.-H. "An FFT-based fast melody comparison method for query-by-singing/humming systems", *Pattern Recognition Letters*, v. 33, n. 16, pp. 2285–2291, Dezembro 2012.

[28] GHIAS, A., LOGAN, J., CHAMBERLIN, D., et al. "Query by humming: musical information retrieval in an audio database". In: *Proceedings of the 3rd ACM International Conference on Multimedia (MULTIMEDIA'95)*, pp. 231–236, Boston, EUA, Novembro 1995.

[29] HU, N., DANNENBERG, R. B. "A comparison of melodic database retrieval techniques using sung queries". In: *Proceedings of the 2nd ACM/IEEE Joint Conference on Digital Libraries (JCDL 2002)*, pp. 301–307, Portland, EUA, Junho 2002.

[30] ZHU, Y., SHASHA, D. "Warping indexes with envelope transforms for query by humming". In: *Proceedings of the 2003 ACM International Conference on Management of Data (SIGMOD 2003)*, pp. 181–192, San Diego, Califórnia, Junho 2003.

[31] LEMSTRÖM, K. *String matching techniques for music retrieval.* Tese de Doutorado, Department of Computer Science, University of Helsinki, Helsínquia, Finlândia, 2000.

[32] PARDO, B., BIRMINGHAM, W. P. "Encoding timing information for musical query matching". In: *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 267–268, Paris, França, Outubro 2002.

[33] SALAMON, J., ROHRMEIER, M. "A quantitative evaluation of a two stage retrieval approach for a melodic query by example system". In: *Proceedings of the 10th International Society for Music Information Retrieval Conference, (ISMIR 2009)*, pp. 255–260, Kobe, Japão, Outubro 2009.

[34] CANCELA, P., LÓPEZ, E., ROCAMORA, M. "Fan chirp transform for music representation". In: *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, pp. 330–337, Graz, Áustria, Setembro 2010.

[35] ROCAMORA, M. *Singing voice detection in polyphonic music.* Tese de Mestrado, Universidad de la República, Montevidéu, Uruguai, 2011.

[36] ROCAMORA, M., CANCELA, P. "Pitch tracking in polyphonic audio by clustering local fundamental frequency estimates". In: *Proceedings of the 9th Brazilian Audio Engineering Conference (AES Brasil 2011)*, pp. 80–87, São Paulo, Brasil, Maio 2011.

[37] ROCAMORA, M., PARDO, A. "Separation and classification of harmonic sounds for singing voice detection". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, v. 7441, *Lecture Notes in Computer Science*, Springer, pp. 707–714, 2012.

[38] WERUAGA, L., KÉPESI, M. "The fan-chirp transform for non-stationary harmonic signals", *Signal Processing*, v. 87, n. 6, pp. 1504–1522, Junho 2007.

[39] ROCAMORA, M., HERRERA, P. "Comparing audio descriptors for singing voice detection in music audio files". In: *11º Simpósio Brasileiro de Computação Musical (SBCM07)*, pp. 187–196, São Paulo, Brasil, Setembro 2007.

[40] ELLIS, D. P. W. "PLP and RASTA (and MFCC, and inversion) in Matlab". 2005. Disponível em: <`http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`>.

[41] WITTEN, I. H., FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. 2 ed. San Francisco, Morgan Kaufmann, 2005.

[42] ROBINSON, D. W., DADSON, R. S. "A re-determination of the equal-loudness relations for pure tones", *British Journal of Applied Physics*, v. 7, n. 5, pp. 166–181, 1956.

[43] DRESSLER, K. "Sinusoidal extraction using an efficient implementation of a multi-resolution FFT". In: *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, pp. 247–252, Montreal, Canadá, Setembro 2006.

[44] FLANAGAN, J. L., GOLDEN, R. M. "Phase vocoder", *Bell Systems Technical Journal*, v. 45, n. 9, pp. 1493–1509, 1966.

[45] KLAPURI, A. P. "Multiple fundamental frequency estimation by summing harmonic amplitudes". In: *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pp. 216–221, Victoria, Canadá, Outubro 2006.

[46] SALAMON, J., GÓMEZ, E., BONADA, J. "Sinusoid extraction and salience function design for predominant melody estimation". In: *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pp. 73–80. Paris, França, Setembro 2011.

[47] HERRERA, P., BONADA, J. "Vibrato extraction and parameterization in the spectral modeling synthesis framework". In: *Proceedings Workshop on Digital Audio Effects (DAFx-98)*, pp. 107–110, Barcelona, Espanha, Novembro 1998.

[48] HURON, D. "Tone and voice: a derivation of the rules of voice-leading from perceptual principles", *Music Perception*, v. 19, n. 1, pp. 1–64, Junho 2001.

[49] CARANHA, A. L. *Sistema de Pesquisa de Músicas Através de Solfejo com Foco em Músicas Brasileiras*. Tese de Mestrado, COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, Setembro 2013.

[50] MIYAZAWA, P., TERRON, P., CAVALCANTI, P., et al. "As 100 maiores músicas brasileiras", *Rolling Stone*, v. 1, n. 37, pp. 1–5, Outubro 2009.

[51] MACHADO, R. *Da Intenção ao Gesto Interpretativo: Análise Semiótica do Canto Popular Brasileiro*. Tese de Doutorado, Faculdade de Filosofia, Letras e Ciências Humanas / USP, São Paulo, Brasil, 2012.

[52] POLINER, G. E., ELLIS, D. P. W., EHMANN, F., et al. "Melody transcription from music audio: Approaches and evaluation", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, n. 4, pp. 1247–1256, Maio 2007.

Apêndice A

Sistema de Adaptação Automática da Base de Músicas: Resultados Completos

Neste capítulo, serão apresentados de maneira mais completa os resultados dos experimentos realizados com o sistema proposto de adaptação automática da base de música descritos na Seção 6.3.

A.1 Resultados — MIDI

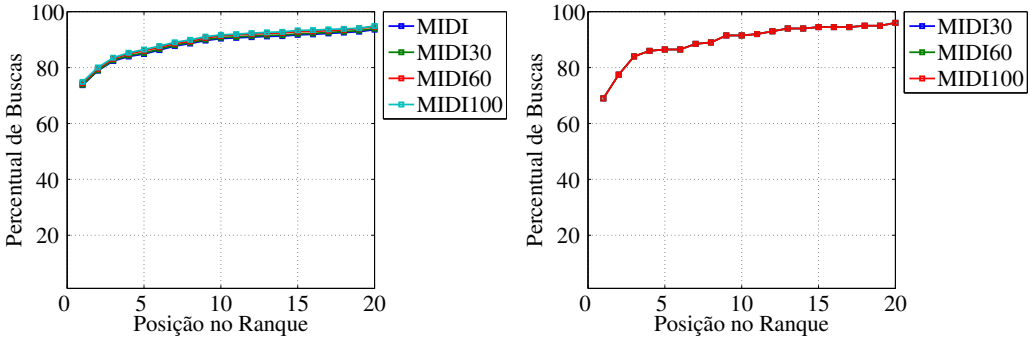
A.1.1 Experimento 1 — Diferentes Probabilidades de Informar-se a Música Consultada

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.1 resume os resultados obtidos com a base MIDI para o experimento da categoria 1 com o sistema utilizando a configuração 1 de limiares. Nas Figuras A.1, A.2 e A.3 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

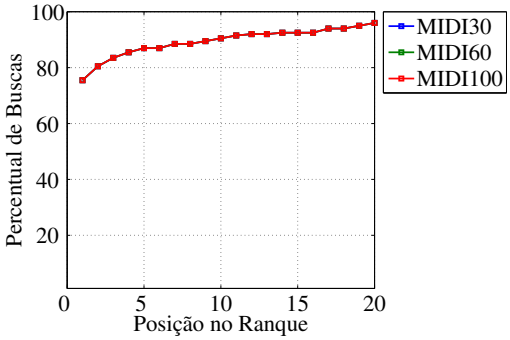
Tabela A.1: Resultados obtidos com a base MIDI para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,79	73,91	85,22	90,74	1
	0,80	74,09	85,40	90,65	2
	0,80	74,18	85,31	90,83	3
60	0,80	74,44	86,02	91,36	1
	0,80	74,53	85,66	91,01	2
	0,80	74,35	85,75	91,27	3
100	0,80	74,89	86,46	91,72	1
	0,80	74,62	86,02	91,45	2
	0,80	75,16	85,93	91,45	3
Tararira	0,79	73,73	84,86	90,38	—

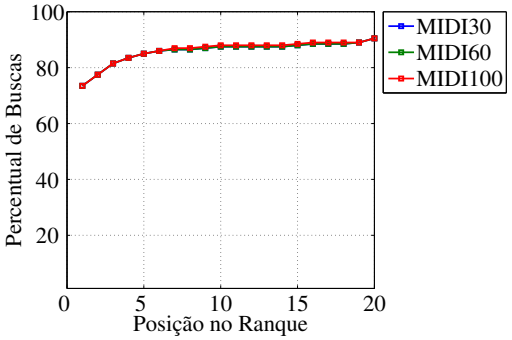


(a) Global.

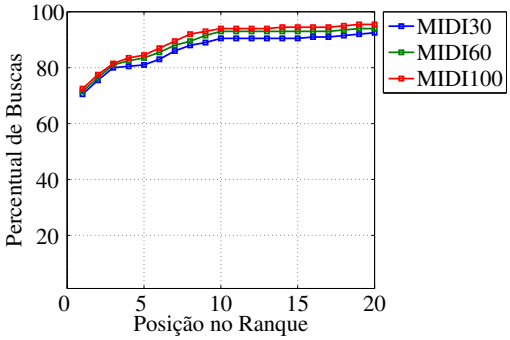
(b) Parte 1.



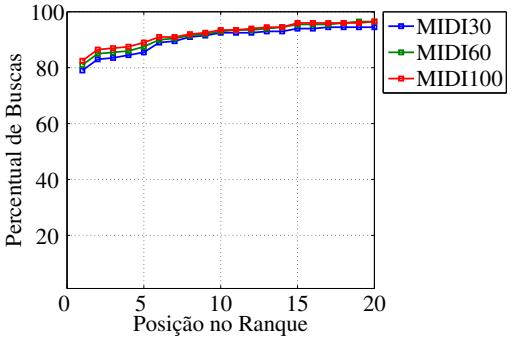
(c) Parte 2.



(d) Parte 3.

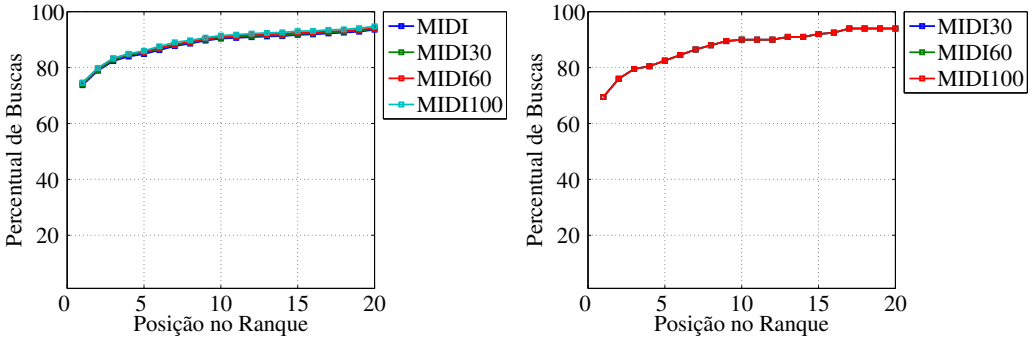


(e) Parte 4.



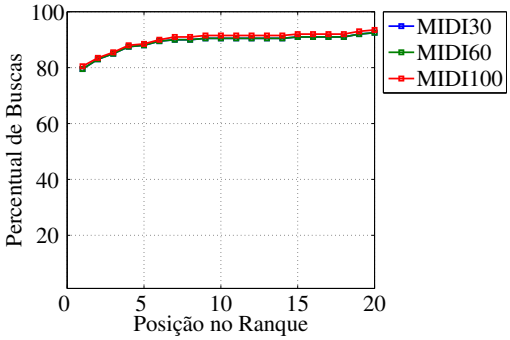
(f) Parte 5.

Figura A.1: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 1 de limiares.

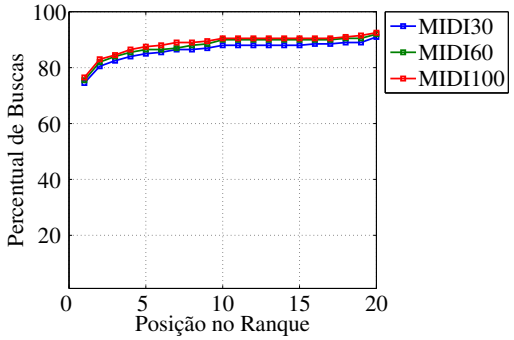


(a) Global.

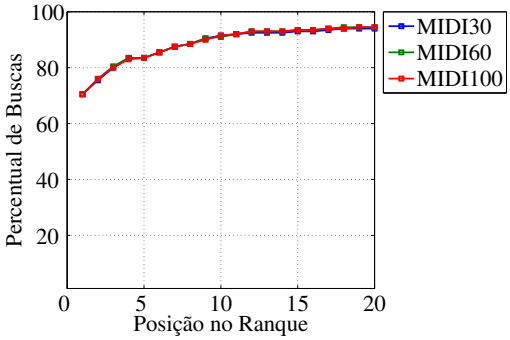
(b) Parte 1.



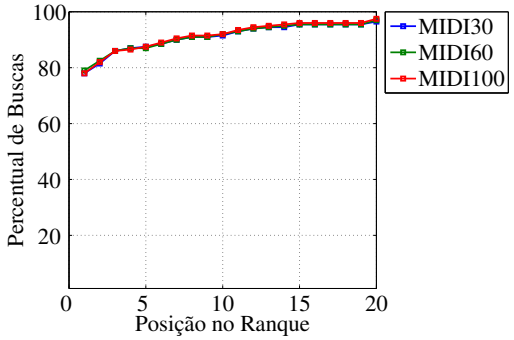
(c) Parte 2.



(d) Parte 3.

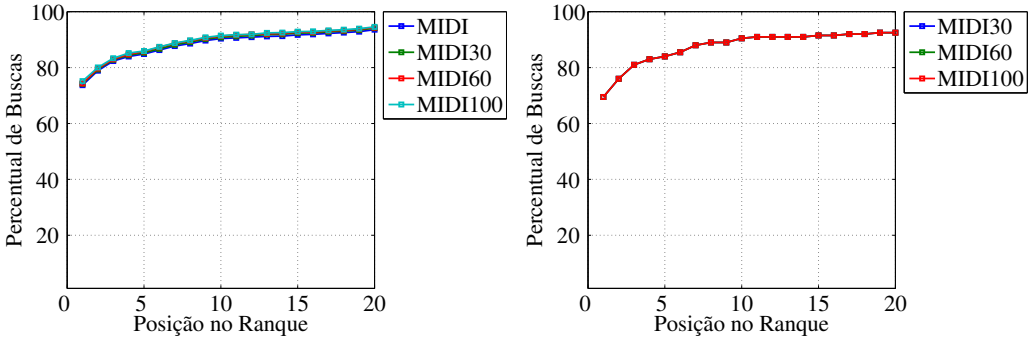


(e) Parte 4.



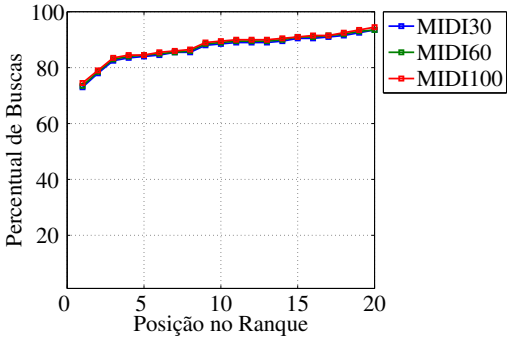
(f) Parte 5.

Figura A.2: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 1 de limiares.

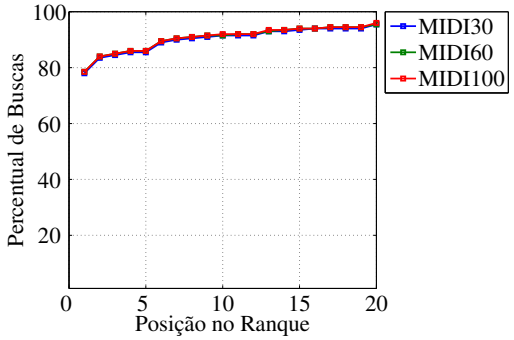


(a) Global.

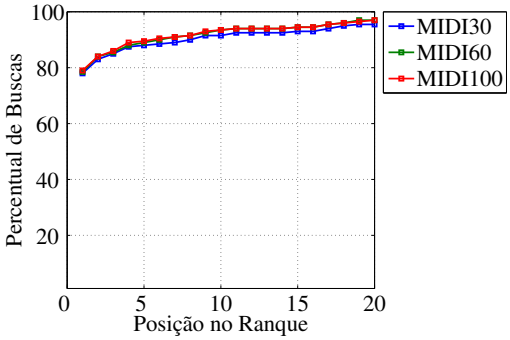
(b) Parte 1.



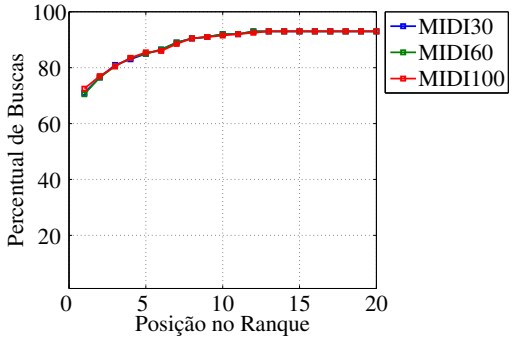
(c) Parte 2.



(d) Parte 3.



(e) Parte 4.



(f) Parte 5.

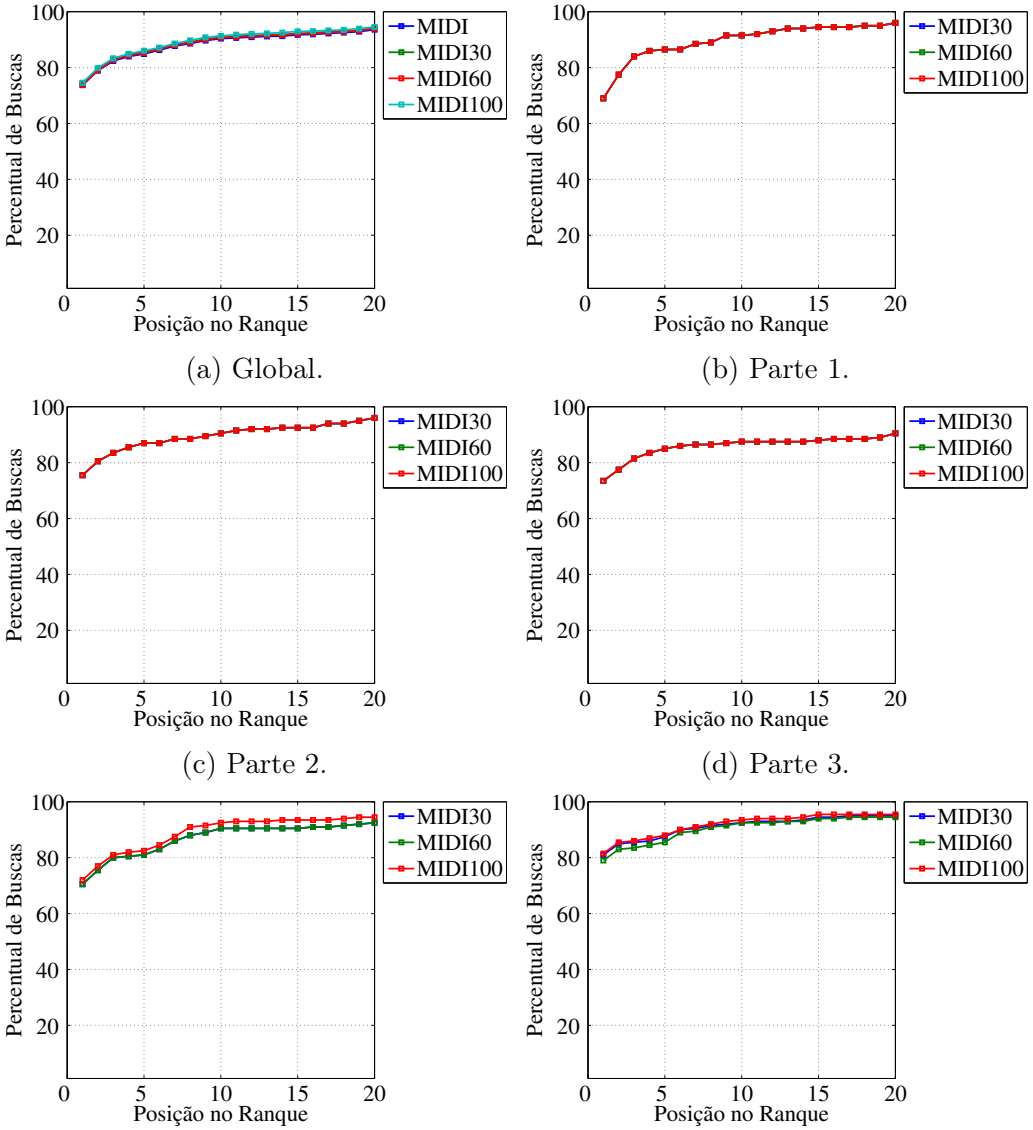
Figura A.3: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.2 resume os resultados obtidos com a base MIDI para o experimento da categoria 1 com o sistema utilizando a configuração 2 de limiares. Nas Figuras A.4, A.5 e A.6 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.2: Resultados obtidos com a base MIDI para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,80	74,27	85,57	90,83	1
	0,80	74,18	85,13	90,56	2
	0,80	74,09	85,22	90,65	3
60	0,79	73,82	85,22	90,74	1
	0,80	74,80	85,93	91,36	2
	0,80	74,27	85,40	90,83	3
100	0,80	74,62	86,02	91,36	1
	0,80	74,53	85,84	91,36	2
	0,81	75,24	86,02	91,45	3
Tararira	0,79	73,73	84,86	90,38	—

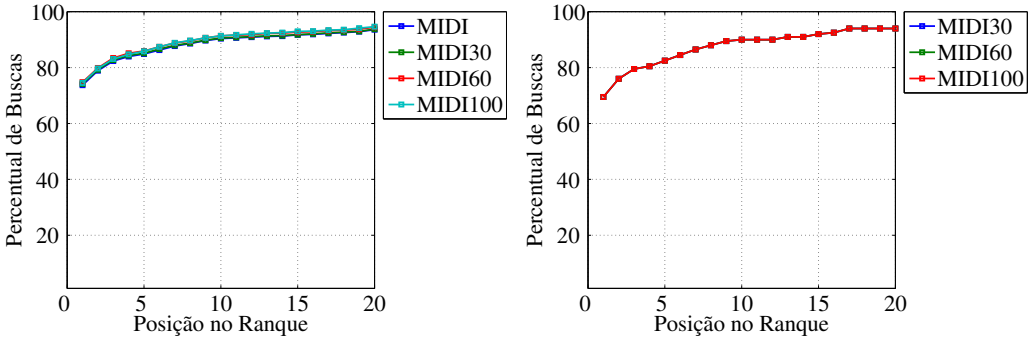


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

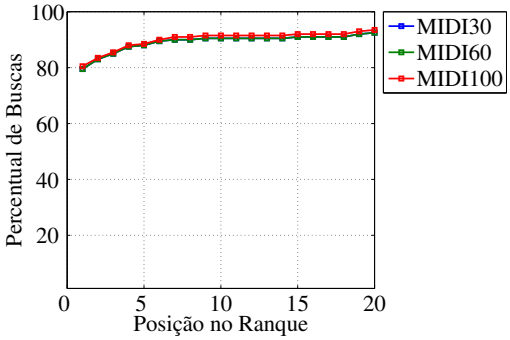
(e) Parte 4. (f) Parte 5.

Figura A.4: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.

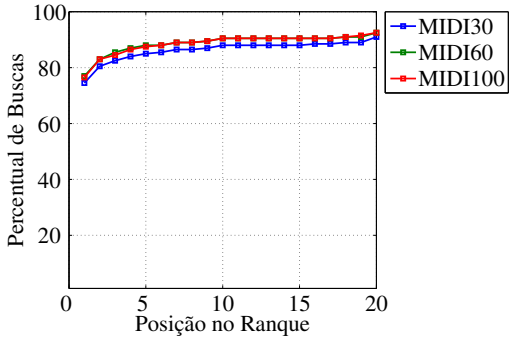


(a) Global.

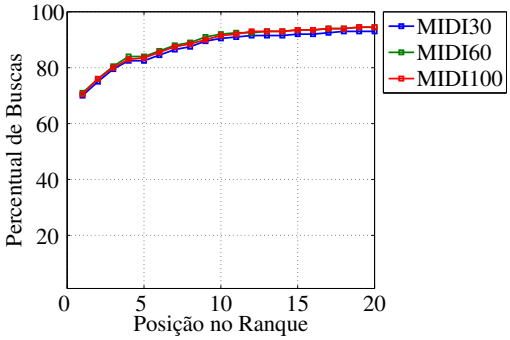
(b) Parte 1.



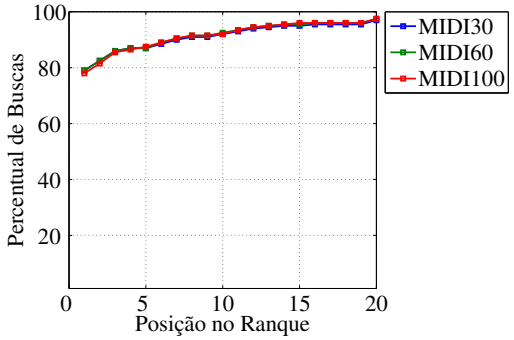
(c) Parte 2.



(d) Parte 3.

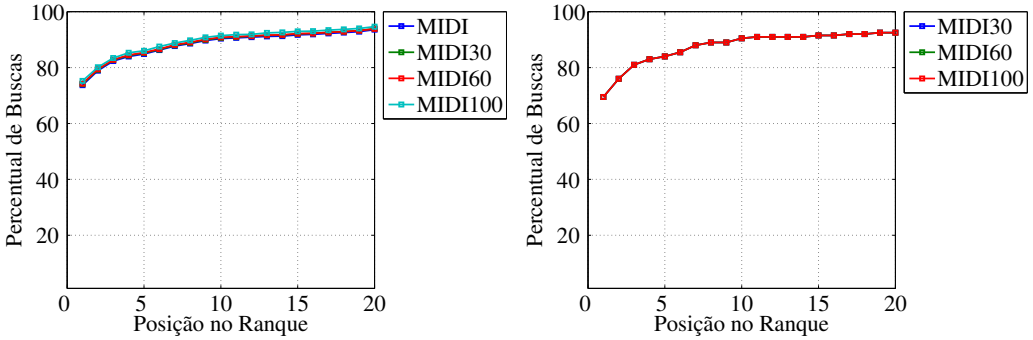


(e) Parte 4.



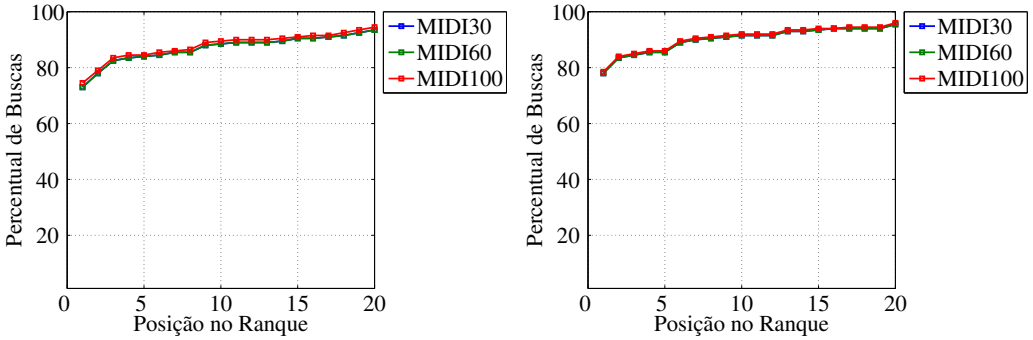
(f) Parte 5.

Figura A.5: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 2 de limiares.



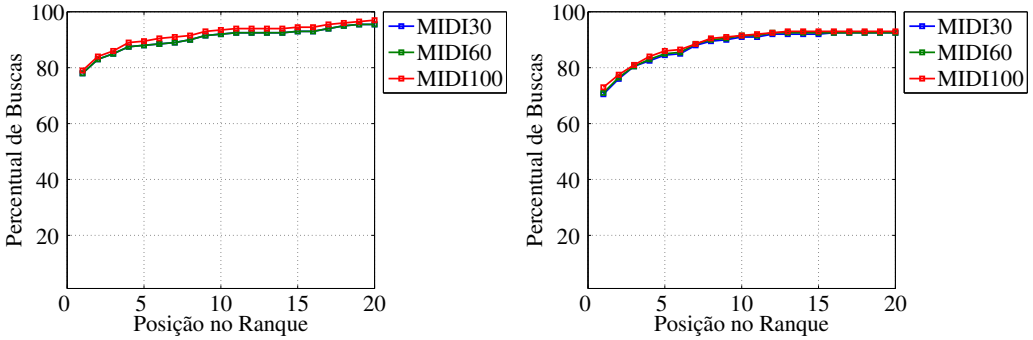
(a) Global.

(b) Parte 1.



(c) Parte 2.

(d) Parte 3.



(e) Parte 4.

(f) Parte 5.

Figura A.6: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 2 de limiares.

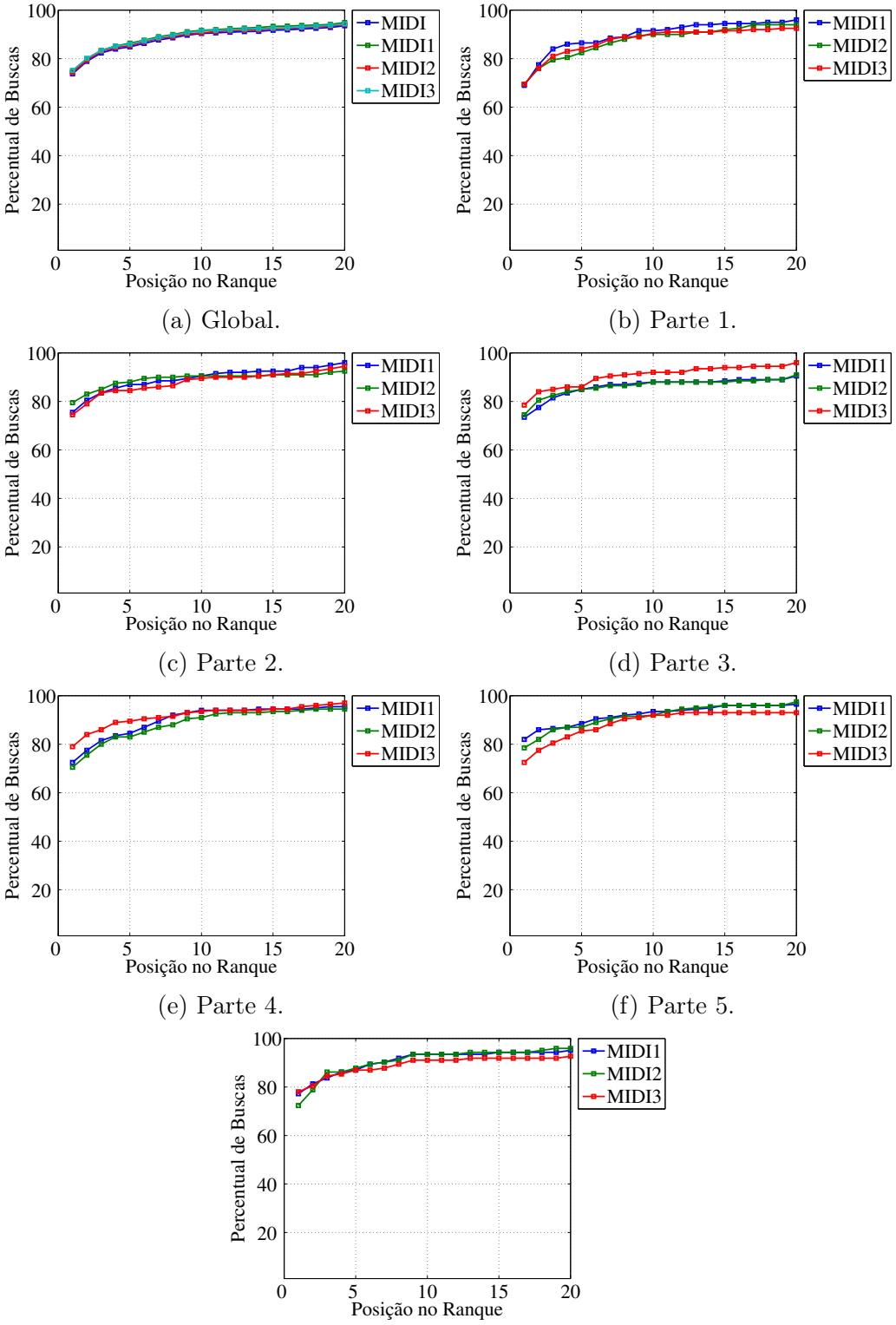
A.1.2 Experimento 2 — Usuários Podendo Errar ao Informarem a Música

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.3 resume os resultados obtidos com a base MIDI para o experimento da categoria descrita. Na Figura A.7 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.3: Resultados obtidos com a base MIDI com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 1 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,80	74,80	86,38	91,72	1
	0,80	74,27	85,40	90,65	2
	0,80	75,16	86,02	91,45	3
Tararira	0,79	73,73	84,86	90,38	—



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

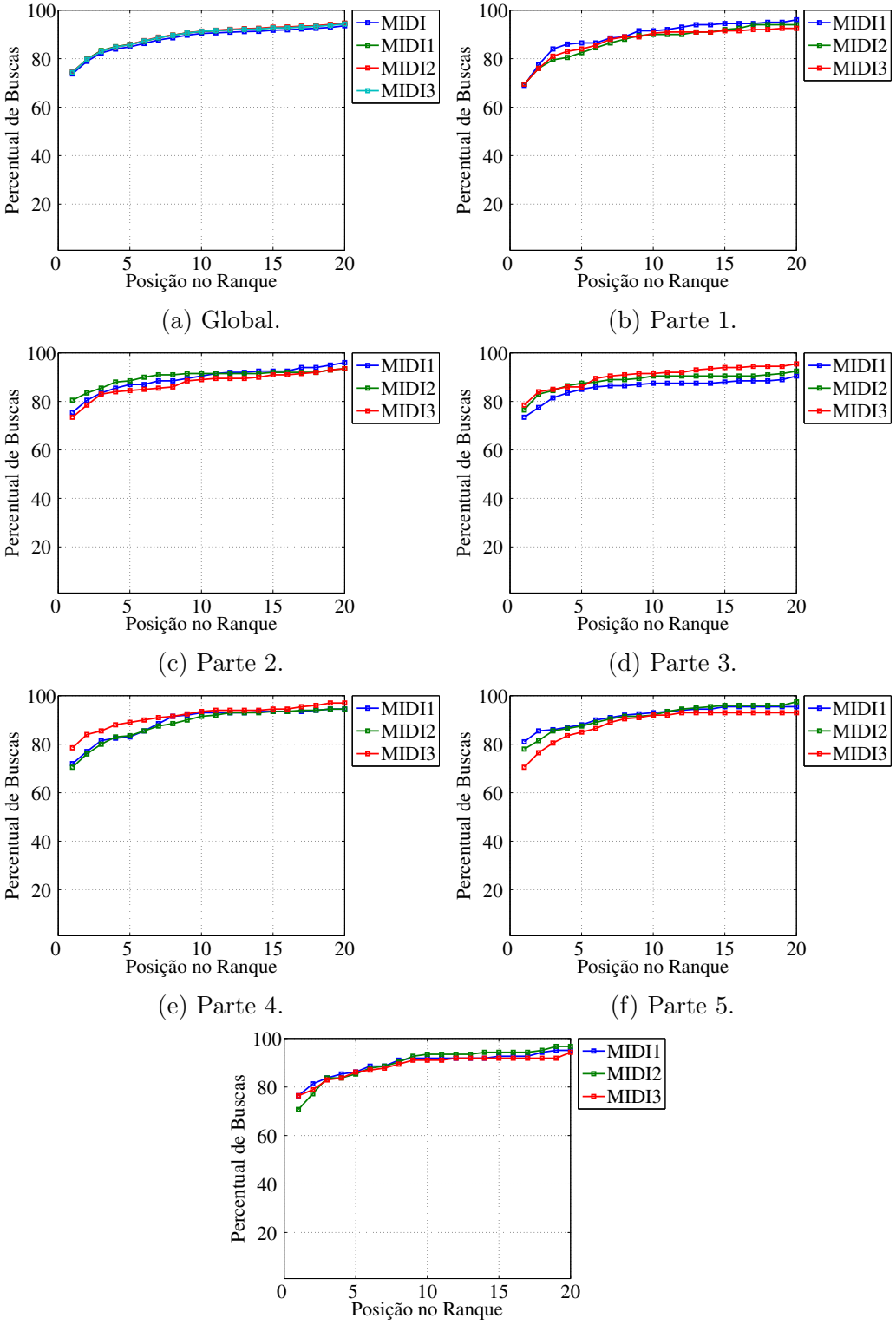
Figura A.7: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.4 resume os resultados obtidos com a base MIDI para o experimento da categoria 2 descrita. Na Figura A.8 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.4: Resultados obtidos com a base MIDI com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,80	74,44	85,93	91,18	1
	0,80	74,53	85,84	91,36	2
	0,80	74,35	85,75	91,27	3
Tararira	0,79	73,73	84,86	90,38	—



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.8: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base MIDI (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.

A.2 Resultados — GPA

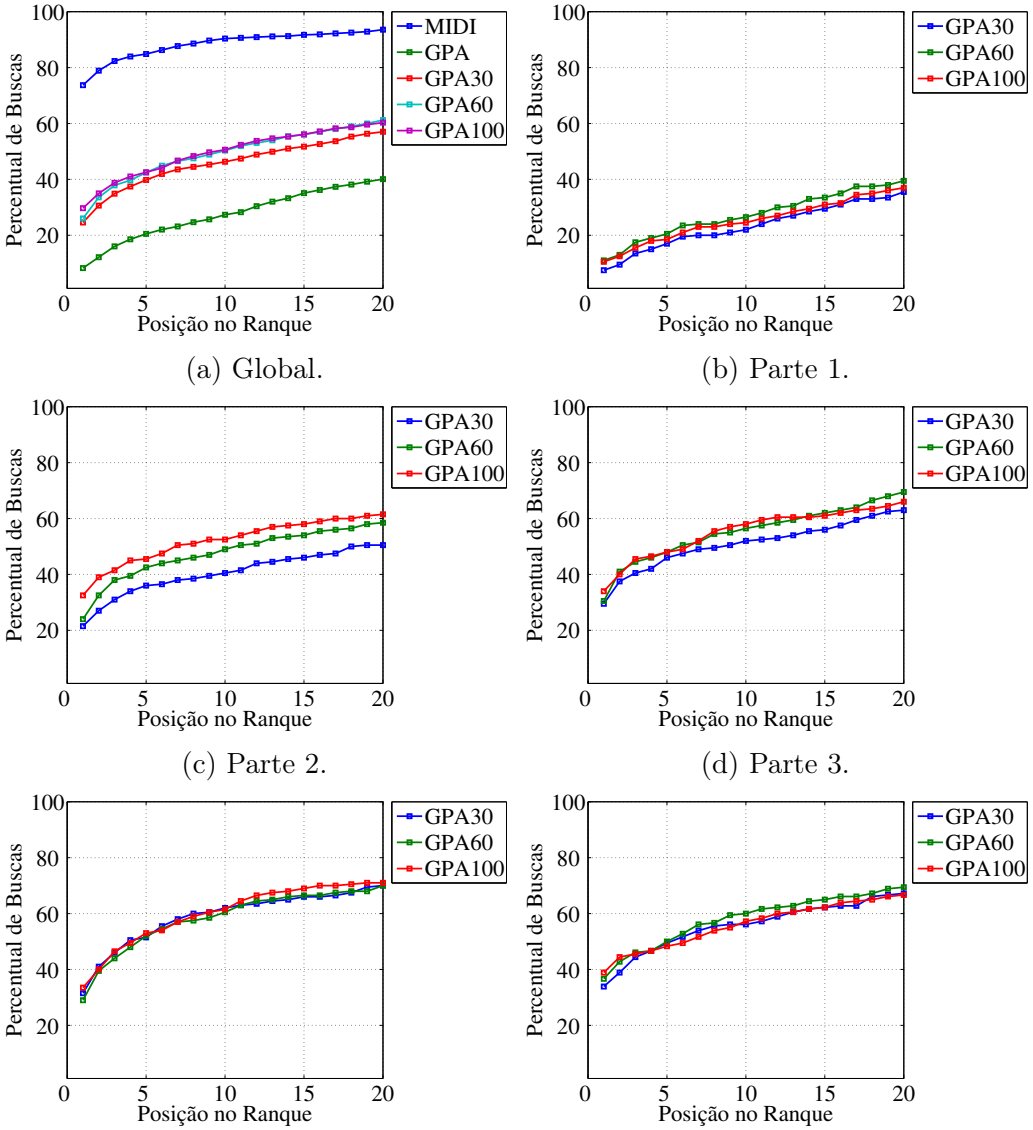
A.2.1 Experimento 1 — Diferentes Probabilidades de Informar-se a Música Consultada

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.5 resume os resultados obtidos com a base GPA para o experimento da categoria 1 com o sistema utilizando a configuração 1 de limiares. Nas Figuras A.9, A.10 e A.11 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.5: Resultados obtidos com a base GPA para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,33	24,59	39,80	46,33	1
	0,29	20,82	33,67	40,71	2
	0,30	21,94	37,04	43,98	3
60	0,35	26,02	42,45	50,31	1
	0,31	23,37	37,65	45,00	2
	0,41	33,06	47,65	55,51	3
100	0,37	29,69	42,55	50,61	1
	0,35	26,94	41,12	50,20	2
	0,35	26,73	41,53	49,08	3
Tararira	0,16	8,27	20,51	27,35	—

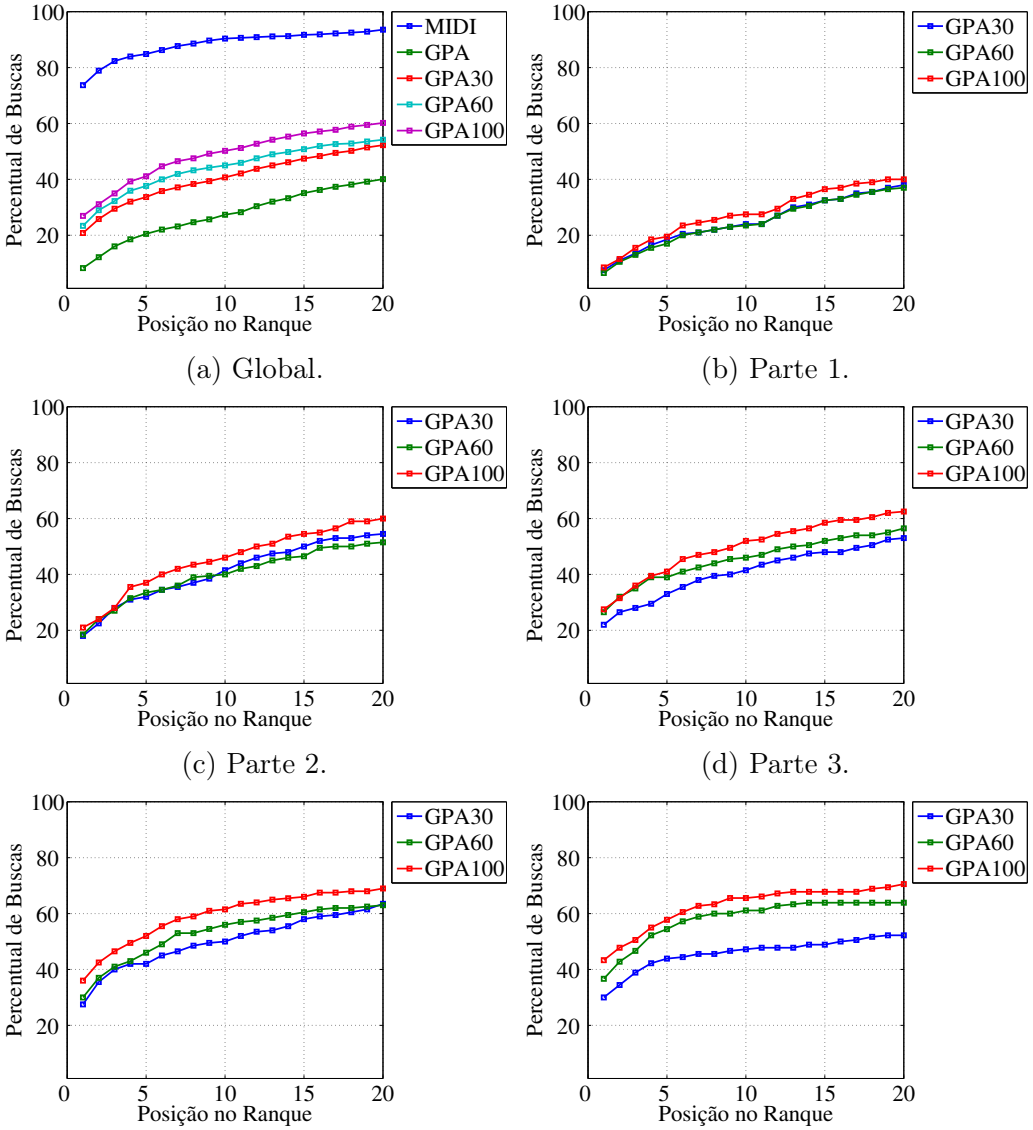


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura A.9: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 1 de limiares.

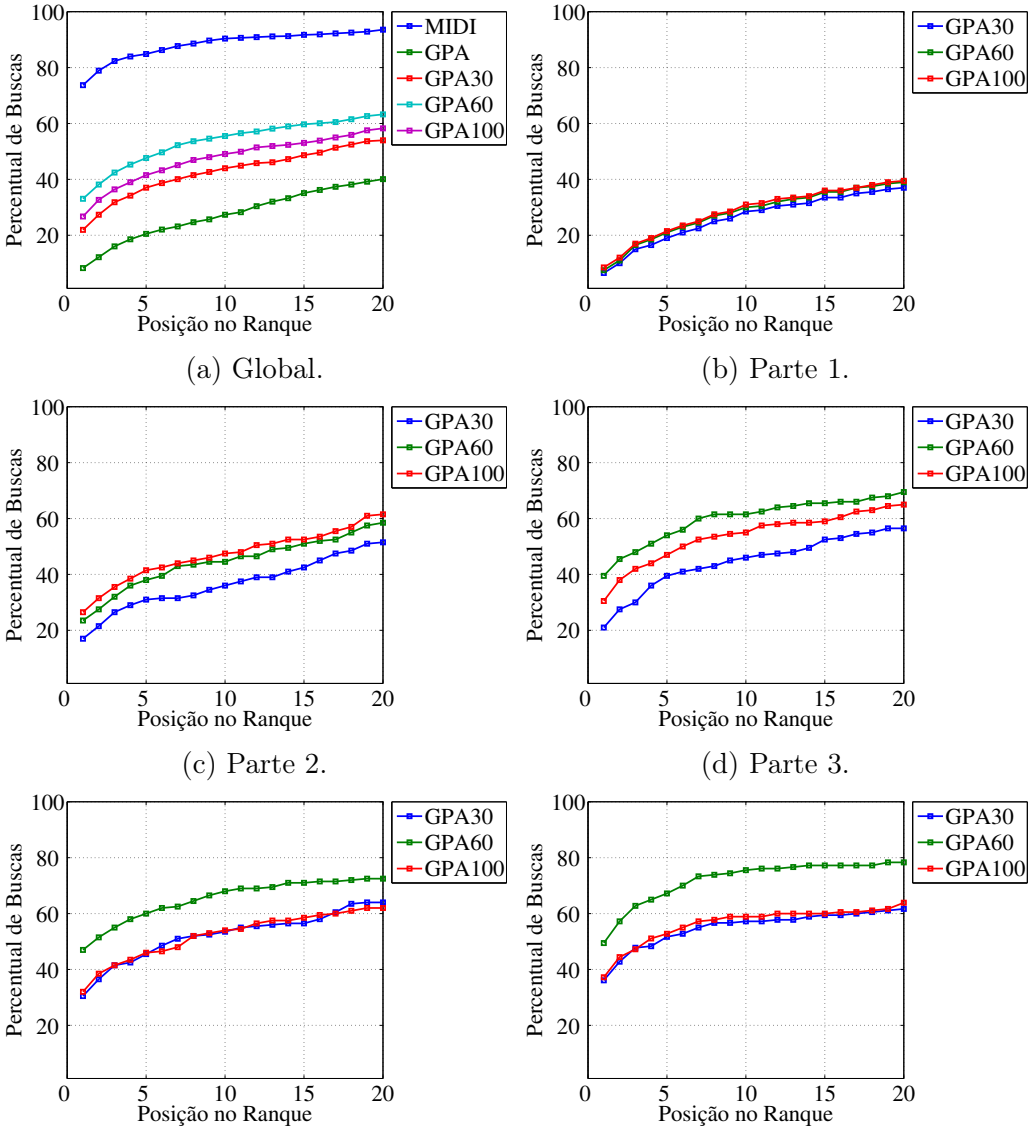


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura A.10: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 1 de limiares.



(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

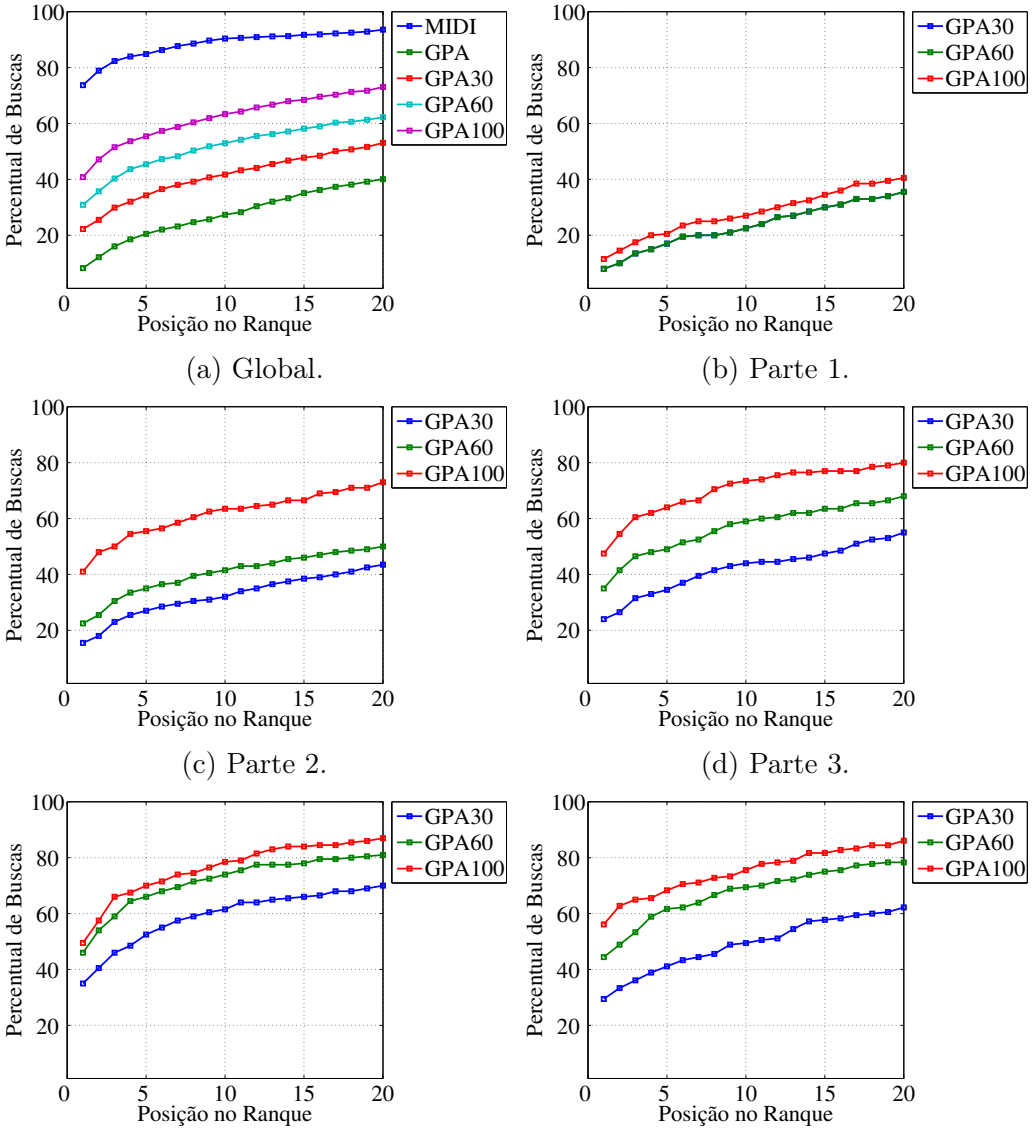
Figura A.11: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.6 resume os resultados obtidos com a base GPA para o experimento da categoria 1 com o sistema utilizando a configuração 2 de limiares. Nas Figuras A.12, A.13 e A.14 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.6: Resultados obtidos com a base GPA para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,29	22,24	34,29	41,73	1
	0,33	25,20	39,90	47,24	2
	0,27	18,47	32,65	39,80	3
60	0,39	30,92	45,41	52,96	1
	0,38	29,29	44,39	52,96	2
	0,41	33,47	48,06	55,61	3
100	0,49	40,82	55,41	63,37	1
	0,43	35,10	50,20	59,39	2
	0,45	37,14	52,35	60,92	3
Tararira	0,16	8,27	20,51	27,35	—

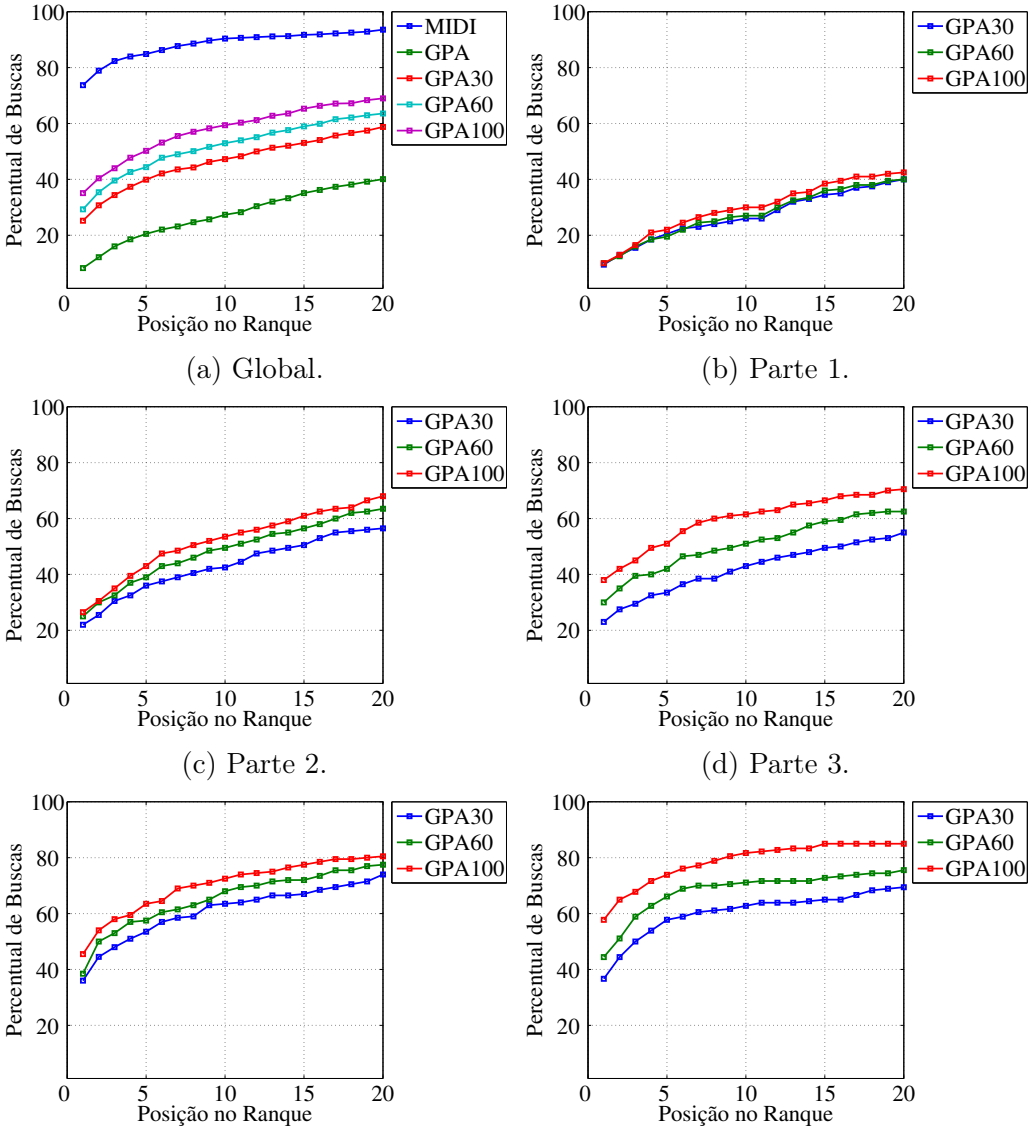


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura A.12: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.

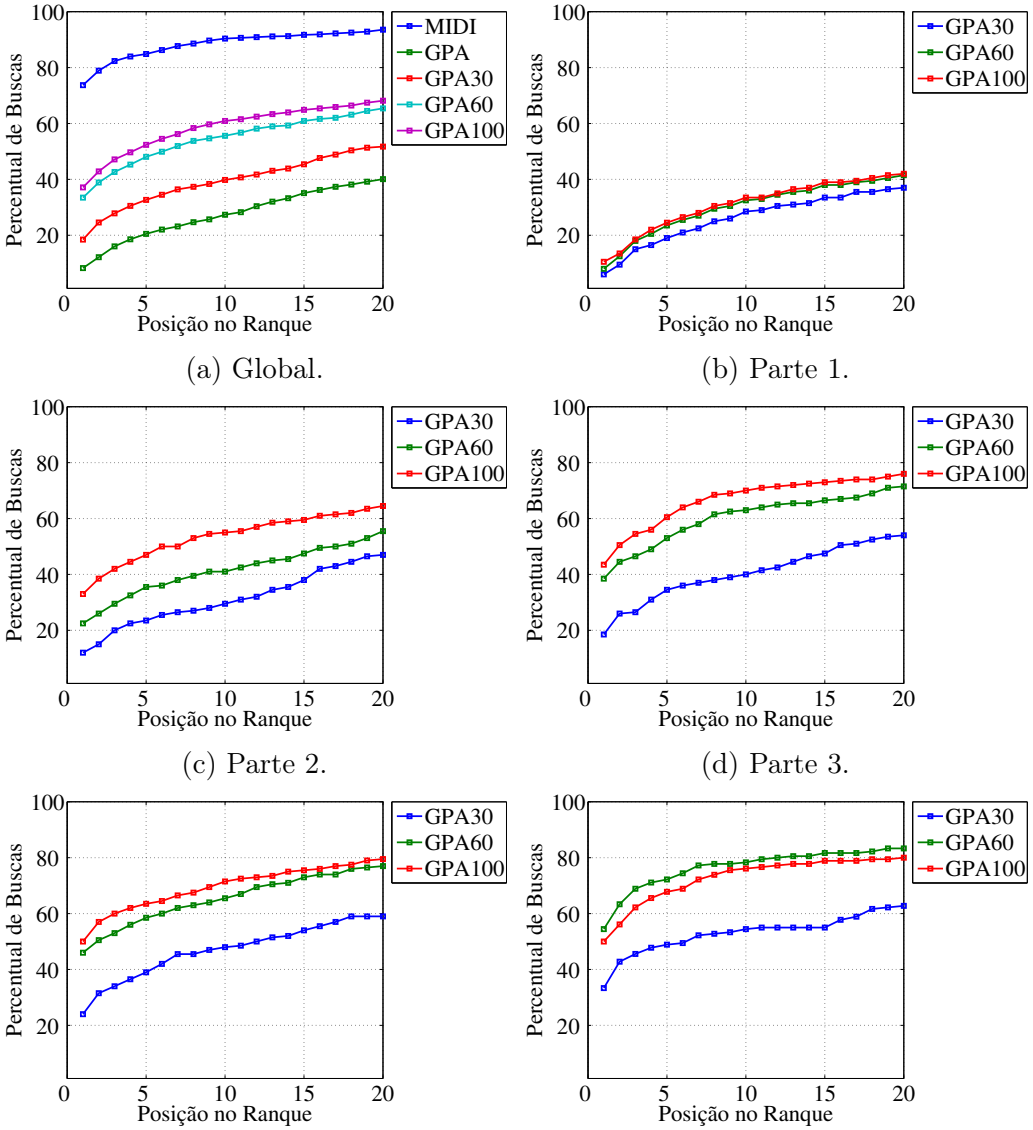


(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura A.13: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 2 de limiares.



(a) Global. (b) Parte 1.

(c) Parte 2. (d) Parte 3.

(e) Parte 4. (f) Parte 5.

Figura A.14: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 2 de limiares.

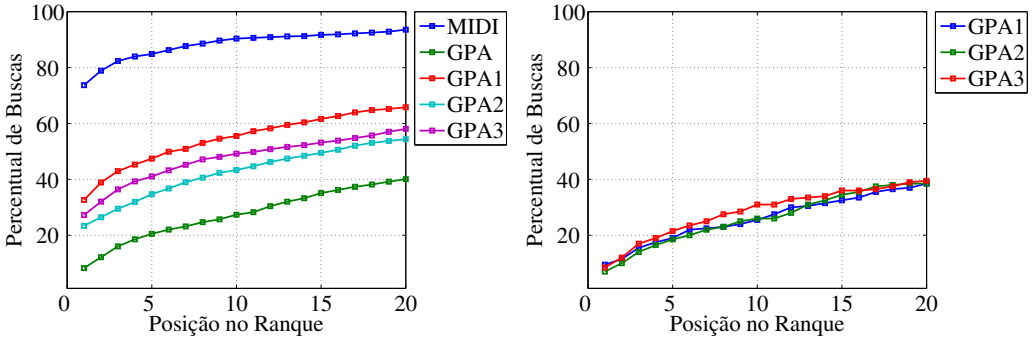
A.2.2 Experimento 2 — Usuários Podendo Errar ao Informarem a Música

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.7 resume os resultados obtidos com a base GPA para o experimento da categoria 2 descrita. Na Figura A.15 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

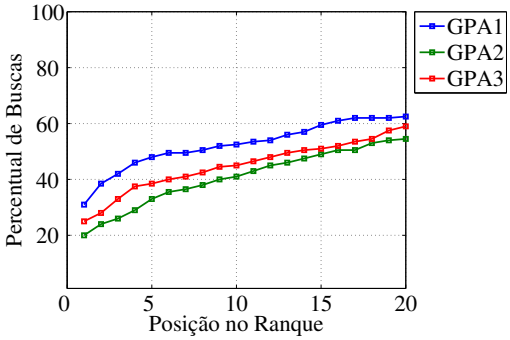
Tabela A.7: Resultados obtidos com a base GPA com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 1 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,41	32,65	47,45	55,51	1
	0,30	23,37	34,69	43,37	2
	0,35	27,24	41,02	49,18	3
Tararira	0,16	8,27	20,51	27,35	—

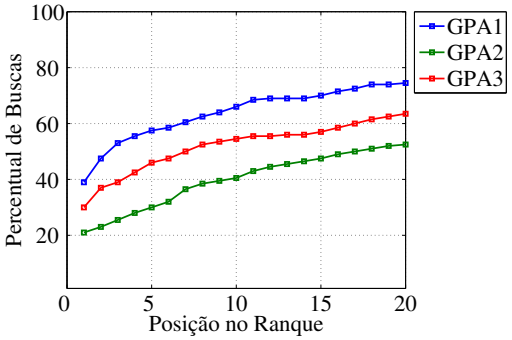


(a) Global.

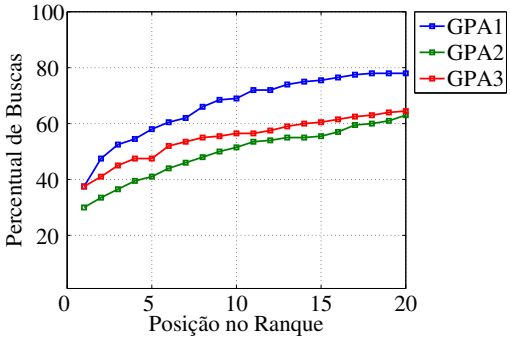
(b) Parte 1.



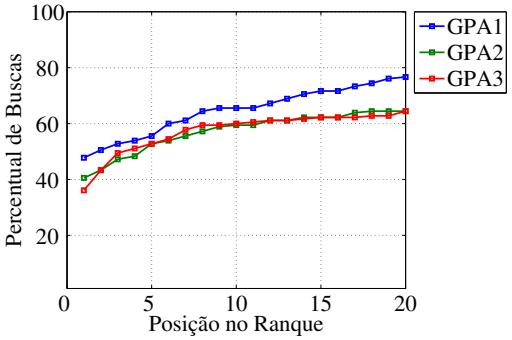
(c) Parte 2.



(d) Parte 3.



(e) Parte 4.



(f) Parte 5.

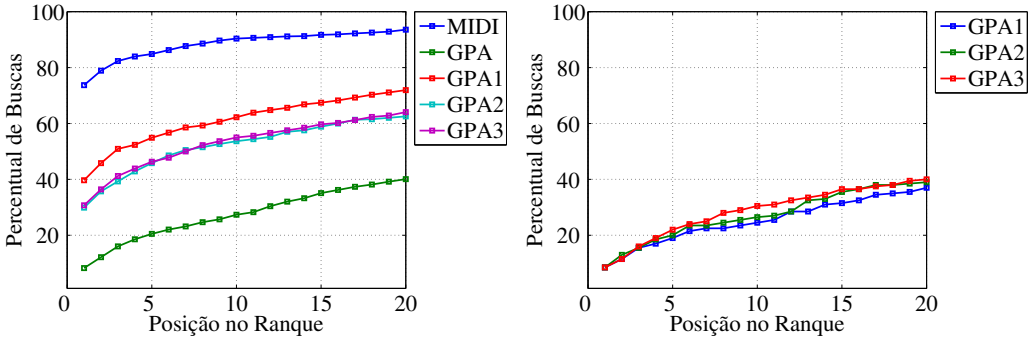
Figura A.15: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.8 resume os resultados obtidos com a base GPA para o experimento da categoria 2 descrita. Na Figura A.16 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

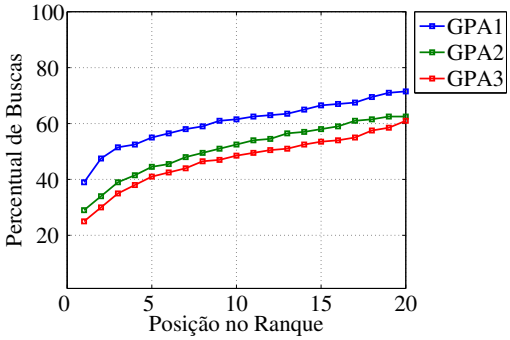
Tabela A.8: Resultados obtidos com a base GPA com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,48	39,69	54,90	62,24	1
	0,38	29,90	45,82	53,67	2
	0,39	30,71	46,33	55,00	3
Tararira	0,16	8,27	20,51	27,35	—

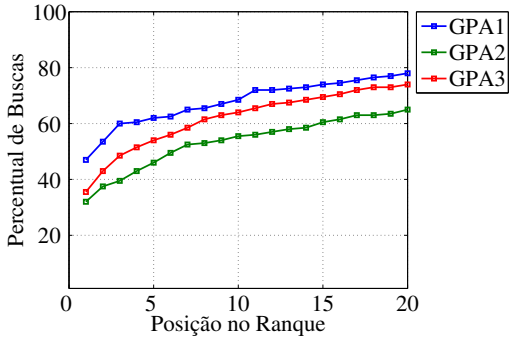


(a) Global.

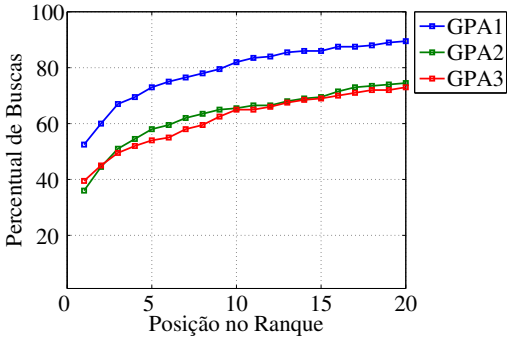
(b) Parte 1.



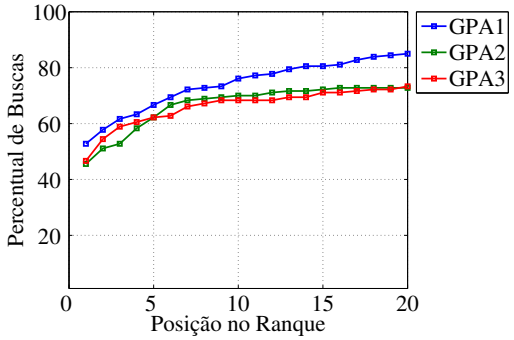
(c) Parte 2.



(d) Parte 3.



(e) Parte 4.



(f) Parte 5.

Figura A.16: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base GPA (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.

A.3 Resultados — Melodia

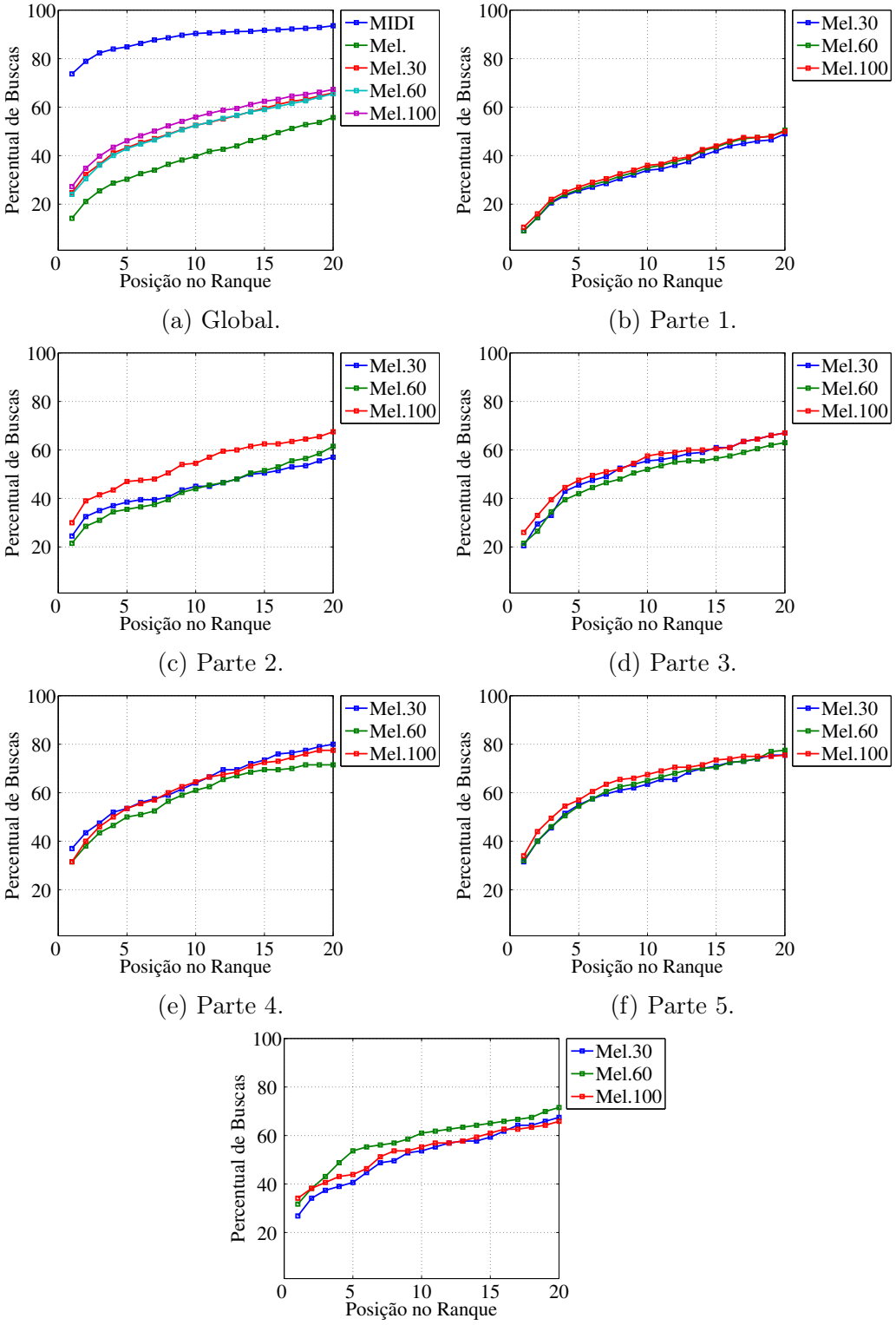
A.3.1 Experimento 1 — Diferentes Probabilidades de Informar-se a Música Consultada

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.9 resume os resultados obtidos com a base Melodia para o experimento da categoria 1 com o sistema utilizando a configuração 1 de limiares. Nas Figuras A.17, A.18 e A.19 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.9: Resultados obtidos com a base Melodia para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,35	24,76	43,28	52,54	1
	0,29	18,70	37,85	47,46	2
	0,29	19,23	35,98	45,86	3
60	0,34	24,04	42,92	52,45	1
	0,36	25,47	44,97	54,85	2
	0,39	28,32	48,17	57,35	3
100	0,37	27,25	46,13	55,92	1
	0,39	28,67	48,26	56,81	2
	0,41	30,90	49,24	58,33	3
Tararira	0,24	14,16	30,28	39,72	—



(a) Global.

(b) Parte 1.

(c) Parte 2.

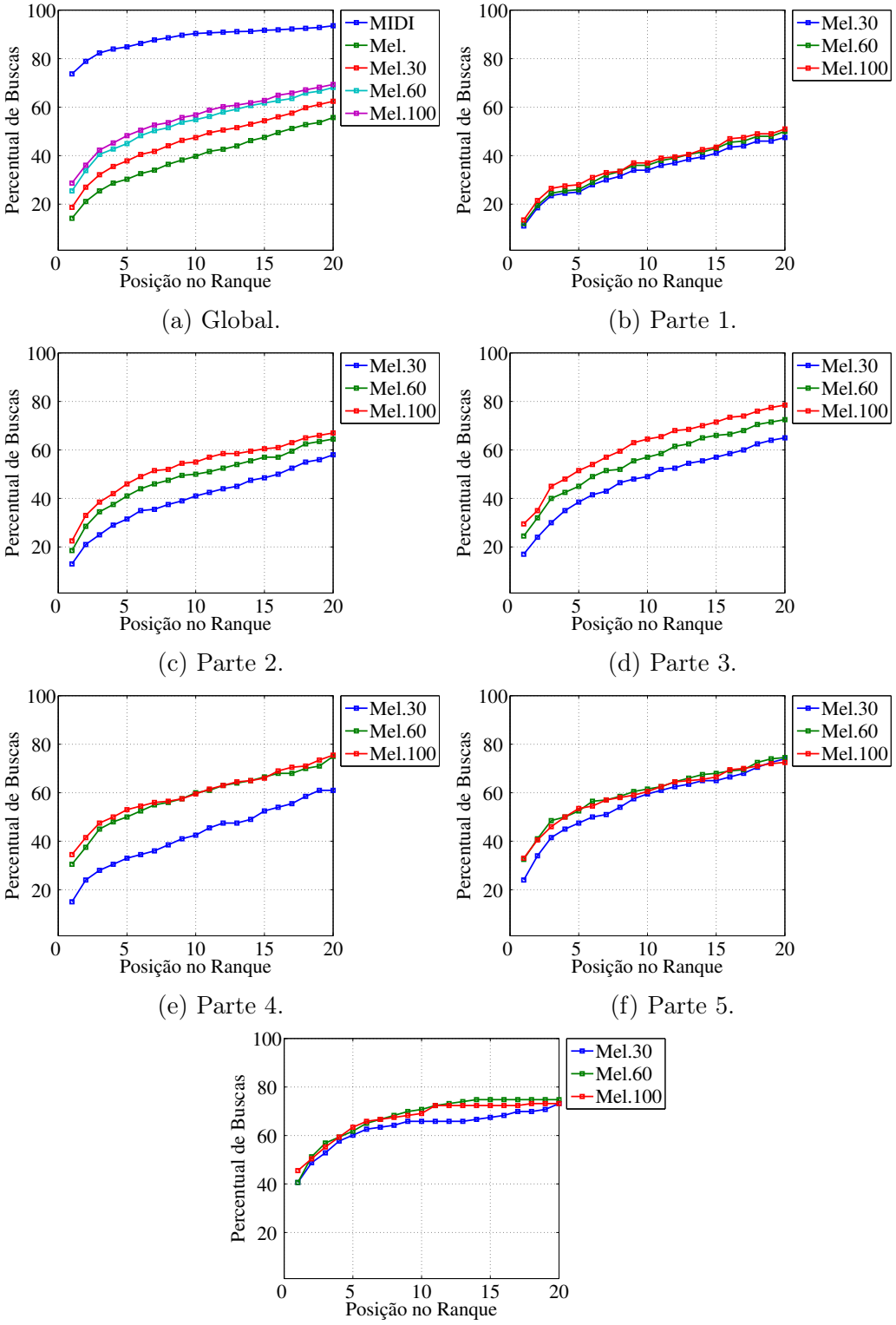
(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.17: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 1 de limiares.



(a) Global.

(b) Parte 1.

(c) Parte 2.

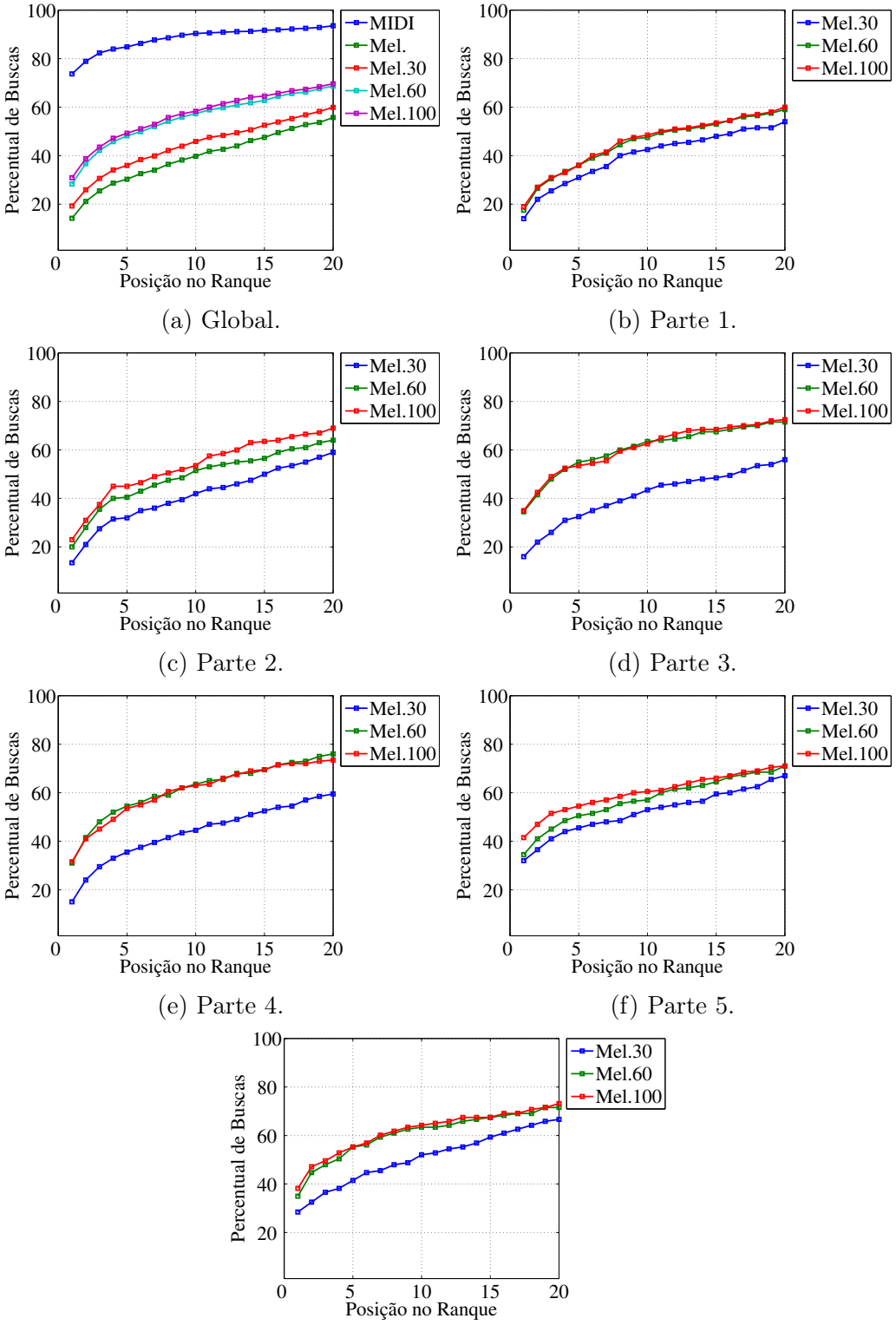
(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.18: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 1 de limiares.



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

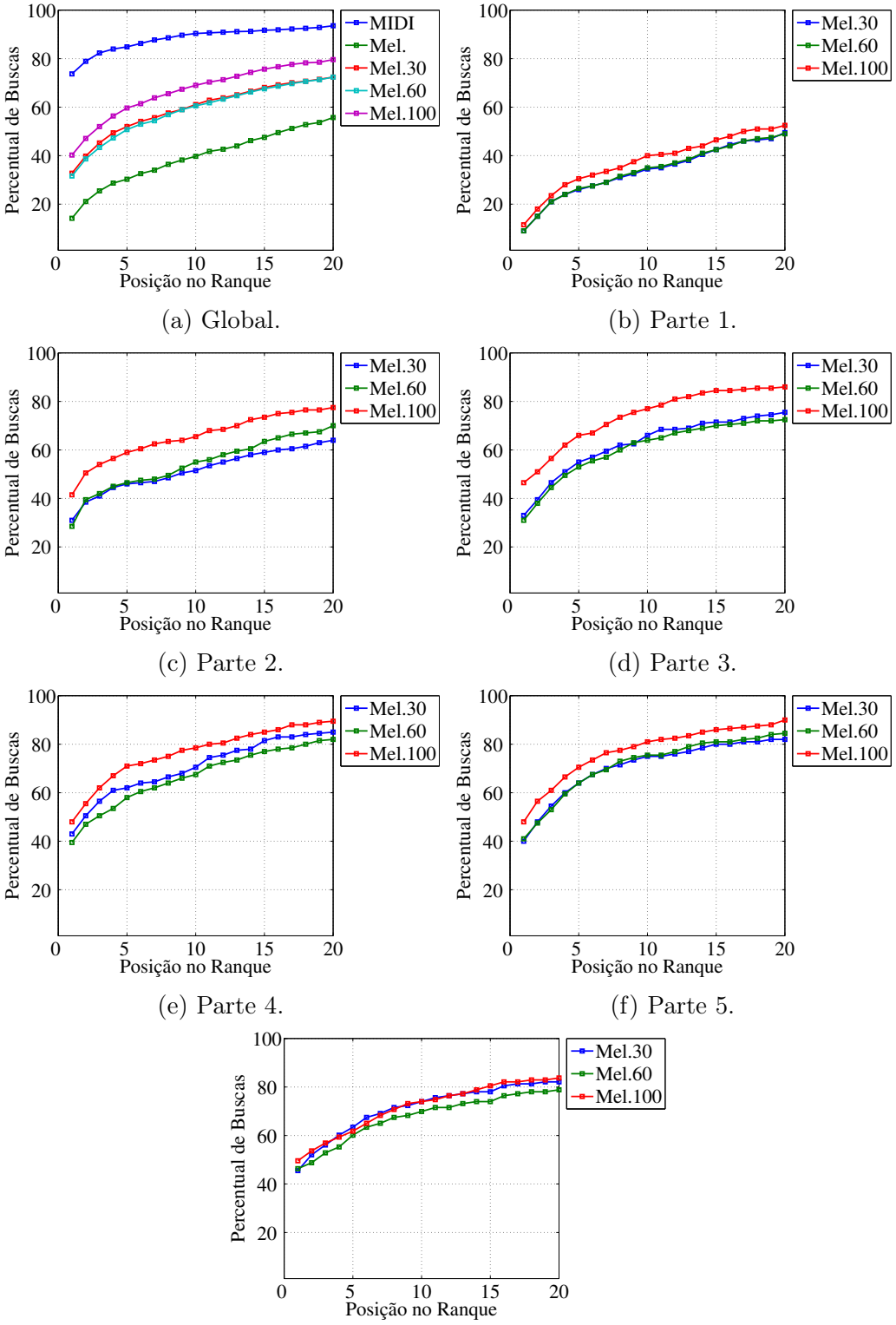
Figura A.19: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.10 resume os resultados obtidos com a base Melodia para o experimento da categoria 1 com o sistema utilizando a configuração 2 de limiares. Nas Figuras A.20, A.21 e A.22 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

Tabela A.10: Resultados obtidos com a base Melodia para as diferentes probabilidades de informação e realizações de ordem das consultas comparados com os resultados obtidos com o sistema Tararira.

Prob. de Informação (%)	MRR	Top-1	Top-5	Top-10	Ordem
30	0,42	32,77	52,00	61,09	1
	0,36	26,27	43,63	53,25	2
	0,37	27,43	44,61	54,23	3
60	0,41	31,61	50,76	60,55	1
	0,41	31,26	50,58	60,55	2
	0,37	26,71	47,37	56,19	3
100	0,50	40,25	59,66	69,01	1
	0,43	32,50	52,00	62,33	2
	0,44	34,28	54,14	63,67	3
Tararira	0,24	14,16	30,28	39,72	—



(a) Global.

(b) Parte 1.

(c) Parte 2.

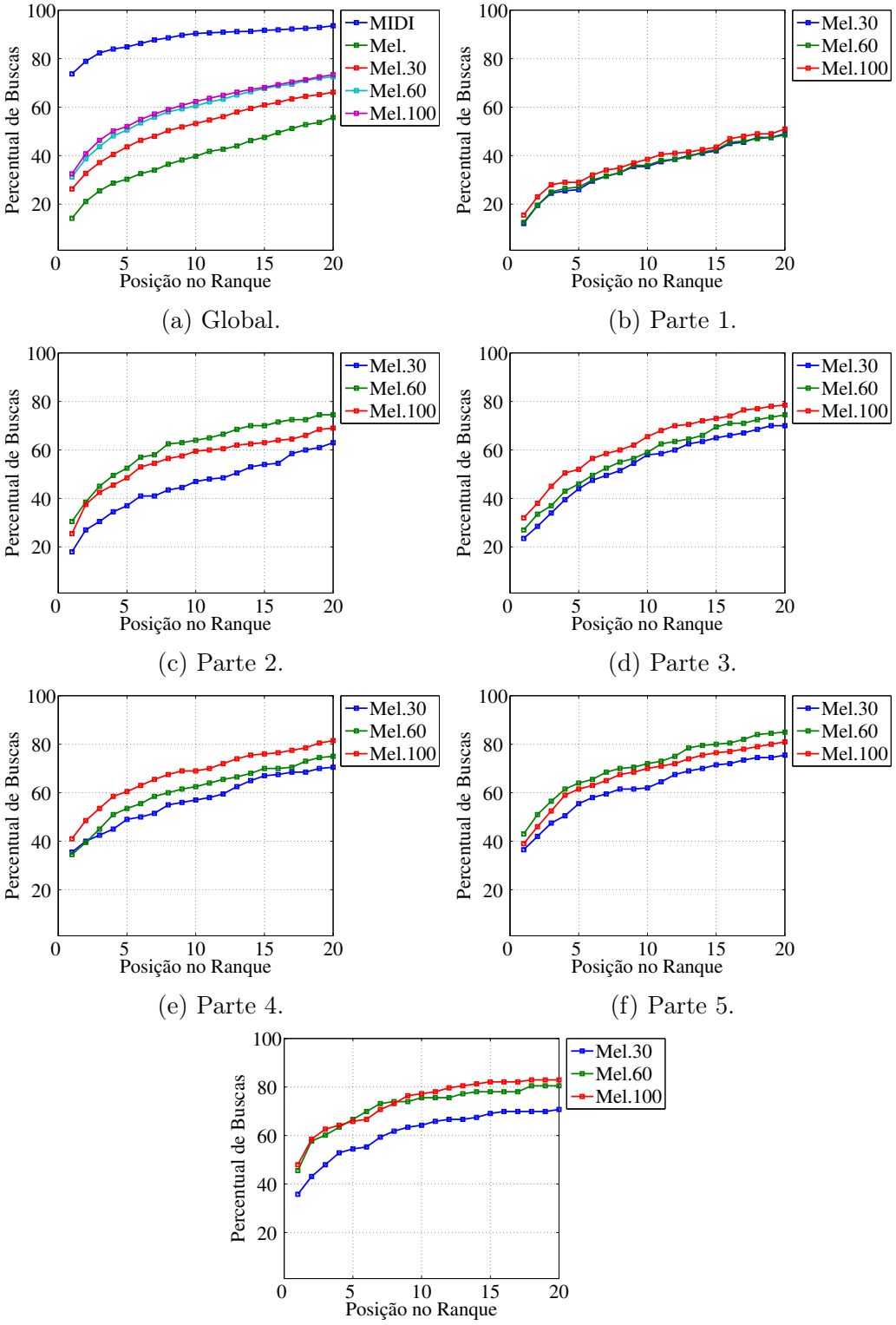
(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.20: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 1 da ordenação aleatória das consultas com configuração 2 de limiares.



(a) Global.

(b) Parte 1.

(c) Parte 2.

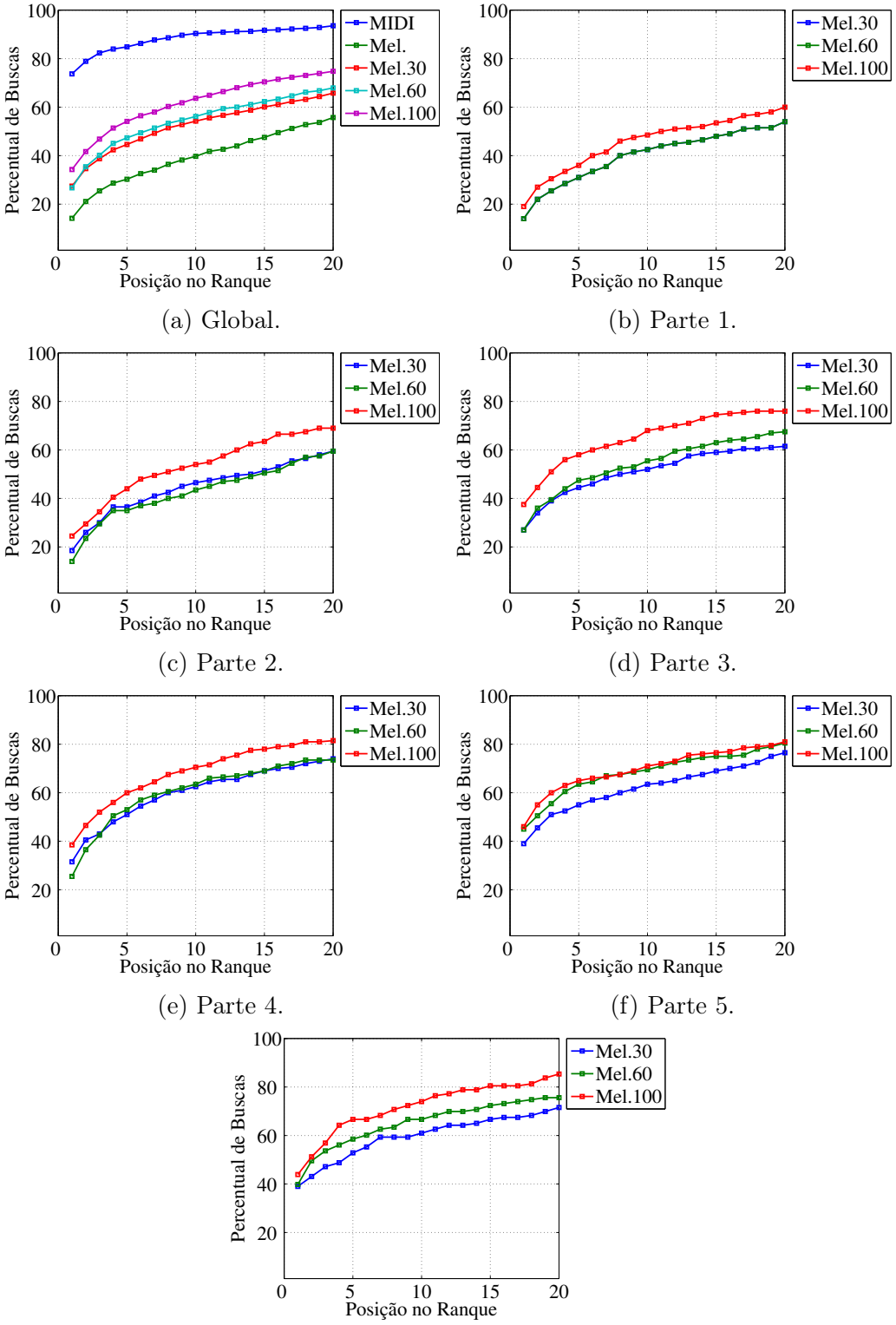
(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.21: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 2 da ordenação aleatória das consultas com configuração 2 de limiares.



(a) Global.

(b) Parte 1.

(c) Parte 2.

(d) Parte 3.

(e) Parte 4.

(f) Parte 5.

(g) Parte 6.

Figura A.22: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia com diferentes probabilidades de os usuários informarem a música — realização 3 da ordenação aleatória das consultas com configuração 2 de limiares.

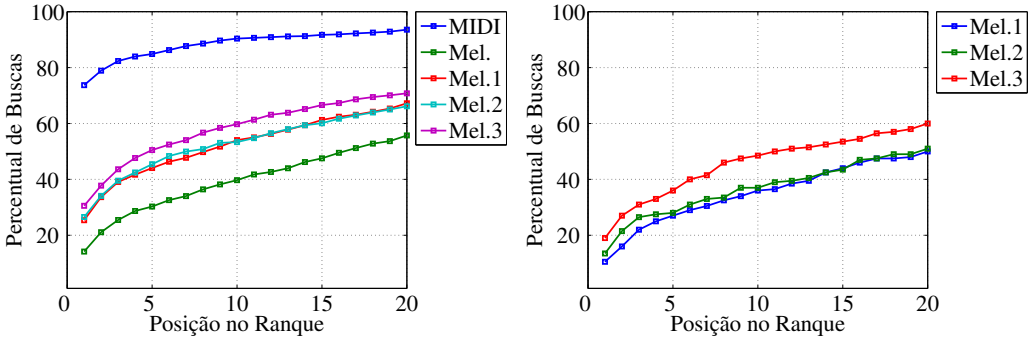
A.3.2 Experimento 2 — Usuários Podendo Errar ao Informarem a Música

Configuração de Limiares 1 com 3 Realizações de Ordem das Consultas.

A Tabela A.11 resume os resultados obtidos com a base Melodia para o experimento da categoria 2 descrita. Na Figura A.23 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

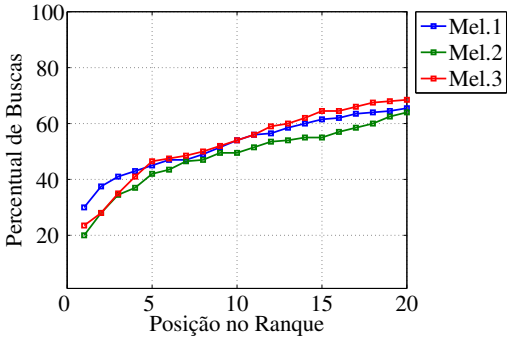
Tabela A.11: Resultados obtidos com a base Melodia com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 1 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0,36	25,38	44,08	54,05	1
	0,36	26,54	45,41	53,34	2
	0,41	30,54	50,49	59,75	3
Tararira	0,24	14,16	30,28	39,72	—

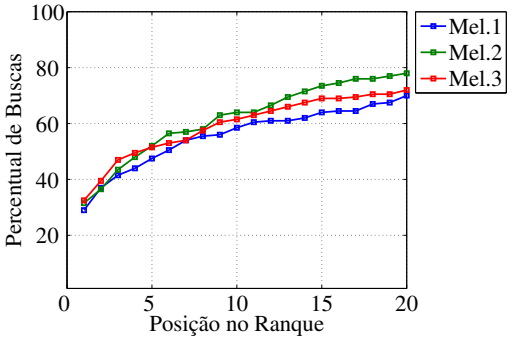


(a) Global.

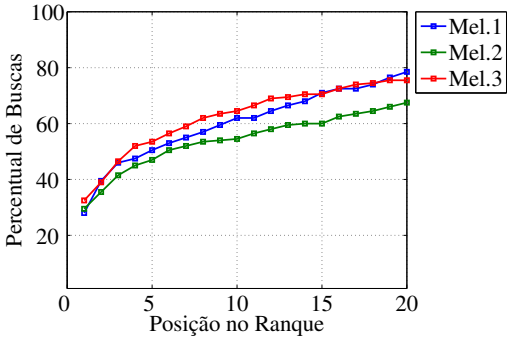
(b) Parte 1.



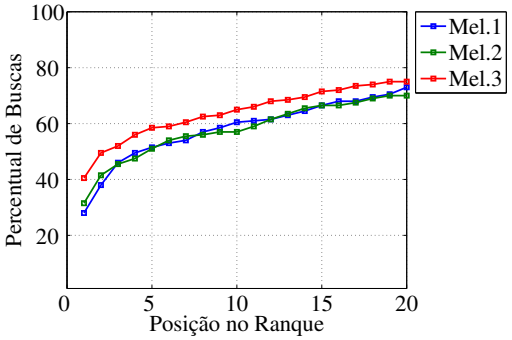
(c) Parte 2.



(d) Parte 3.



(e) Parte 4.



(f) Parte 5.

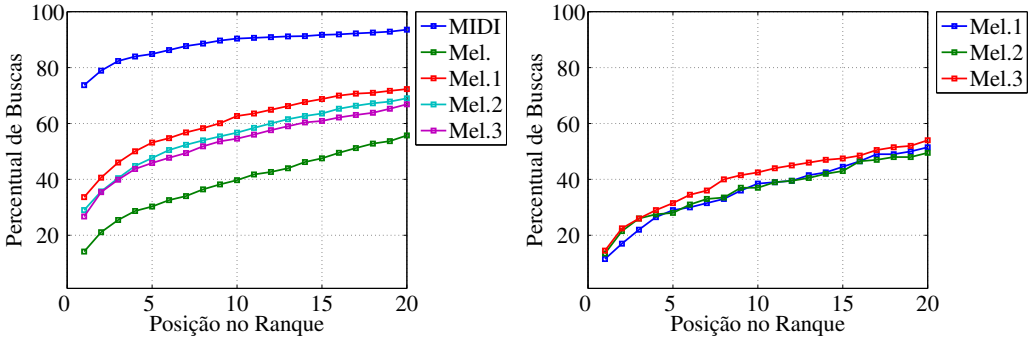
Figura A.23: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 1 de limiares.

Configuração de Limiares 2 com 3 Realizações de Ordem das Consultas.

A Tabela A.12 resume os resultados obtidos com a base Melodia para o experimento da categoria 2 descrita. Na Figura A.24 estão ilustrados os mesmos resultados, mas mostrando a evolução do sistema a cada 200 consultas.

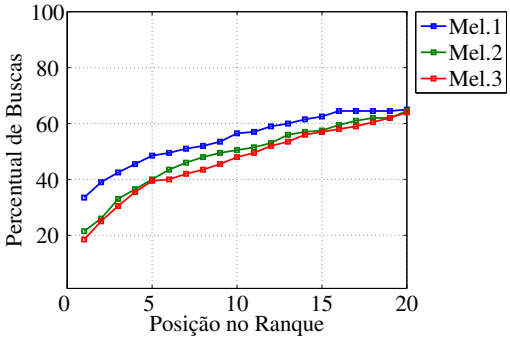
Tabela A.12: Resultados obtidos com a base Melodia com os usuários podendo informar errado, com chance de acerto de 70%, e com o sistema utilizando a configuração de limiares 2 comparados com os resultados obtidos com o sistema Tararira.

Sistema	MRR	Top-1	Top-5	Top-10	Ordem
Sist. Adaptativo	0.453	33.66	53.16	62.69	1
	0.39	29.03	47.64	56.72	2
	0.37	26.71	45.86	54.59	3
Tararira	0,24	14,16	30,28	39,72	—

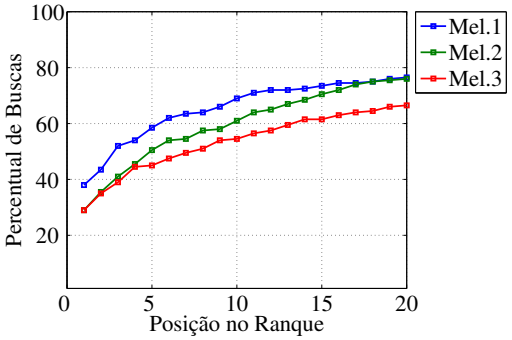


(a) Global.

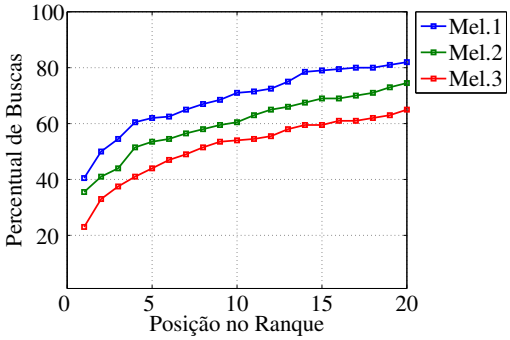
(b) Parte 1.



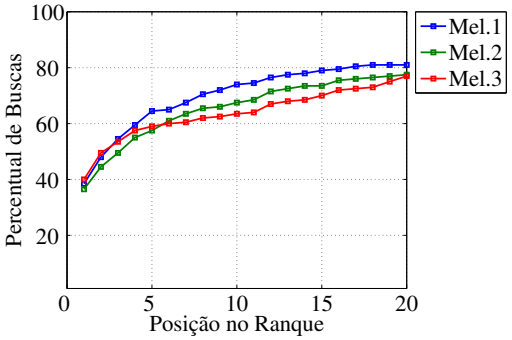
(c) Parte 2.



(d) Parte 3.



(e) Parte 4.



(f) Parte 5.

Figura A.24: Ranques cumulativos demonstrando a evolução (a cada 200 consultas) do desempenho do sistema adaptativo utilizando a base Melodia (selecionando as músicas que possuem mais consultas) com 70% de chance de os usuários informarem corretamente a música — 3 realizações da ordenação aleatória das consultas com configuração 2 de limiares.