

# 作曲支援システムにおけるコード進行及びキーの決定方法

東山 恵祐<sup>†</sup>

音楽的知識や経験の少ないユーザ向けに、2小節のメロディのみを入力として伴奏付の曲を生成する作曲支援システムの開発を行っている。作曲における最も困難な作業は、メロディ及びコード進行を有機的に連携づけて生成することであると考えられる。既存手法では、ユーザ自身がコード進行を生成しなくてはならない、予め決められたコード進行に合うようにメロディを生成しなくてはならない、あるいはコード進行を自動で生成するために一曲分すべてのメロディをユーザ自身が生成しなくてはならない、等といった条件を課しているものが多く、音楽的知識や経験の少ないユーザにとっての使い勝手がよいとは言えない。またキーの決定について、入力されたメロディの統計情報を用いて行う手法が提案されているが、入力メロディの長さが2小節程度と短い場合には、その推定精度が低くなる恐れがある。本報告では、予め用意した複数のコード進行パターン及びそれに様々なアレンジを施したものの中から、ユーザが入力した2小節のメロディと違和感なくマッチするものを、コード進行から導かれるスケールやメロディのリズム等を考慮して選択することにより、コード進行及びキーを同時に決定する方法について述べる。

## 1. はじめに

近年、一般消費者がコンテンツを自ら作成しソーシャルネットワークサービスや動画共有サービスなどにおいてシェアして楽しむ、いわゆるUGC (User Generated Contents) の流通が一般的になっている。音楽の世界においても、PCを用いて楽曲を制作するDTM (Desktop Music) とよばれる環境が充実してきたことに加え、ボーカロイドをはじめとした、歌詞とメロディを入力するだけで歌声を合成できるソフトウェアやアプリケーションの登場により、いわゆるプロの手によるものだけではなく、一般のユーザによる歌唱付楽曲の制作が可能となってきた。それにより、その楽曲が動画共有サイトなどに多くアップロードされる、あるいはCDやライブといった商業活動にも使用されて人気を博す、などといった現象が起こっている。しかし、楽曲の制作には音楽的知識や経験が不可欠であるため、大多数の一般ユーザにとっては自ら楽曲制作を行うことは困難であり、リスナーとしての受け身的な楽しみ方のみをしているのが現状である。そのようなユーザでも楽曲制作を楽しめるようにするためには、音楽的知識や経験がなくても高いクオリティの楽曲を簡単に作曲できるような仕組みを用意することが必要である。

深山らは、日本語の歌詞を入力することによりそれに合うメロディ及び伴奏を自動で生成するシステムを提案している[1]。歌詞付の楽曲を生成できる現段階では随一のシステムであるが、一方でメロディはシステムが自動で生成するためにユーザがメロディ生成に関わることができず、自分で作曲したという実感を得ることが難しい。

南高らは、楽曲の一部のメロディ及び一曲分のコード進行をユーザ自身が作成し、残りのメロディは自動で作成する手法を提案している[2]。この手法は、ユーザが入力したメロディをモチーフとして残りのメロディを生成するため、

自分で作曲をしたという実感を得ることができる。しかし、音楽的知識経験の少ないユーザにとって一曲分のコード進行を自分で作ることは非常に困難である。

Simon らは、ユーザが入力したメロディに合うようなコード進行及び伴奏を、予め学習された確率遷移モデルに基づいて自動で生成する手法を提案している[3]。しかし、そのためにはすべてのメロディを自身で生成することが求められるため、音楽的知識経験の少ないユーザにとって使いやすいシステムであるとは言い難い。また、各キーにおけるダイアトニックスケールとユーザが入力したメロディの各音の出現頻度を比較し、スケールと頻度がもっともマッチするキーを選択することにより、その曲のキーをメロディから決定する手法についても提案している。しかし、これはユーザが入力したメロディの音程の統計情報を利用して推定を行うため、例えば[2]のような入力メロディが2小節程度と少ないシステムに適用する場合には、キーを判断するのに十分な統計情報を得ることができず、その推定精度を確保することが困難である。また、スケールを満たしていることは音楽的にメロディらしいことの十分条件ではないため、結果として音楽的に違和感のあるメロディになってしまう場合もある。

作曲を支援する商用サービスとして、MAISTAMUSIC [4] が既に実用化されている。このサービスは、予め登録されている複数の伴奏から好みのものを1つ選択し、それに合わせて鼻歌でメロディを入力すると、その伴奏に合わせてメロディを自動で生成するというものである。しかし音楽的経験の少ないユーザにとっては、伴奏に合わせてメロディを歌うこと自体が簡単であるとは言い難い。

我々は、音楽的知識や経験の少ないユーザでもサビに使うメロディをハナウタで歌うことができる、との仮定のもとに、サビ先頭の2小節分のメロディのみをユーザが生成し、それ以外のメロディや伴奏をすべて自動で生成するシステムを開発している。本報告では、そのうち、生成する曲のコード進行及びキーを入力メロディのみから違和感なく決定する方法について述べる。

<sup>†</sup> ソニー株式会社  
Sony Corporation  
1-7-1 Konan, Shinagawa-ku, Tokyo, 108-0075 Japan  
keisuke.toyama@jp.sony.com

## 2. 技術詳細

本手法では、ユーザが生成するメロディは、

- ・ 2小節のフレーズ
- ・ 8分音符以上の長さの音により構成
- ・ 4分の4拍子

を満たすものとしている。その入力メロディに違和感なくマッチするコード進行及びキーを、予め用意した複数のコード進行パターンから選択することにより決定する。コード進行パターンについては、様々な基本パターンを予めテンプレートとして所持しておくだけでなく、さらに曲の多様性を確保するためにそのパターンに複数の手法によるアレンジを施すことにより、コード進行のパターン数を増やす。以下に、コード進行及びキーの決定方法の詳細について述べる。

### 2.1 メロディ入力

図1に入力メロディの一例を示す。システムに対するメロディの入力方法は、楽器や声（鼻歌）などでメロディを演奏したものを録音し、そのピッチ周波数を解析することにより音程やリズムを決定してもよいし、DTMソフトウェアへの打ち込みやMIDI（Musical Instrument Digital Interface）楽器の利用によるMIDIデータ入力でも構わないが、最終的にはメロディを構成する音それぞれの開始位置、長さ、及び音程情報を数値化して持っていればよい。我々のシステムでは、図2に示すようにMIDIフォーマットに準拠した形で所持するものとしている。なお、開始位置及び長さについては、4分音符あたりの長さを480とした場合の値で表現するものとしている。



図1: 入力メロディの例

開始位置	長さ	音程
240	240	67
480	240	72
720	240	74
960	240	76
1200	240	74
1440	240	72
1680	240	76
1920	480	74
2400	240	79
2640	720	79

図2: 図1の入力メロディを数値化したもの

### 2.2 コード進行テーブル

図3にコード進行パターンの例を示す。ここでは、コードのルート音がどのように変化するかを示しており、数字はキー音からの相対度数を表している。ここで示されているルート番号と実際のコード構成の関係は図4に示しており、調性がメジャーの場合とマイナーの場合で異なる。図3に示したコードルート番号進行に従って図4のコード構成音を並べていくことにより、コード進行が設定される。なお、この段階では相対度数で構成されたコード進行が設定されるだけであり、キーの設定はされない。このコード進行パターンは、パートごと（例えばイントロパート、Aパート、Bパート、サビパート、エンディングパートなど）に違うものを所持するようにしても良いし、1曲全体を通したものを所持するようにしても構わない。また、生成したい曲の調性に応じたコード進行パターンをそれぞれ所持するようにしてもよい。

ID	1		2		3...
	コード ルート 番号	長さ	コード ルート 番号	長さ	
コード 進行	1	1920	4	1920	
	1	1920	1	1920	
	2	960	5	1920	
	5	960	1	1920	
	1	1920	4	1920	
	4	1920	3	960	
	1	1920	6	960	
	2	1920	2	1920	
	5	1920	5	1920	
	:	:	:	:	

図3: コード進行パターンの例

コードルート番号	調性	
	メジャー	マイナー
1	I	
2	II <sub>m7</sub>	
3	III <sub>m7</sub>	III
4	IV	
5	V	
6	VI <sub>m7</sub>	VI <sub>m</sub>
7	VII <sub>m7</sub> <sup>-5</sup>	

図4: コードルート番号とコード構成の関係

レベル 0 (基本コード)

I	IV	V	I	IV	I	V	I
---	----	---	---	----	---	---	---

レベル 1 (ドミナントセブン, II<sub>m7</sub>→V<sub>7</sub>の導入)

I	IV	V <sub>7</sub>	I	IV	I	II <sub>m7</sub> V <sub>7</sub>	I
---	----	----------------	---	----	---	---------------------------------	---

レベル 2 (トニックコードへ代理コードの導入)

I	IV	V <sub>7</sub>	I	IV	III <sub>m7</sub> VI <sub>m7</sub>	II <sub>m7</sub> V <sub>7</sub>	I
---	----	----------------	---	----	------------------------------------	---------------------------------	---

レベル 3 (セカンダリドミナントの導入)

I	IV	V <sub>7</sub>	I	I <sub>7</sub>	IV	III <sub>m7</sub> VI <sub>7</sub>	II <sub>m7</sub> V <sub>7</sub>	I
---	----	----------------	---	----------------	----	-----------------------------------	---------------------------------	---

レベル 4 (サブドミナントマイナーの導入)

I	IV	V <sub>7</sub>	I	I <sub>7</sub>	IV	IV <sub>m</sub>	III <sub>m7</sub> VI <sub>7</sub>	II <sub>m7</sub> V <sub>7</sub>	I
---	----	----------------	---	----------------	----	-----------------	-----------------------------------	---------------------------------	---

レベル 5 (4 和音化)

I <sub>M7</sub>	IV <sub>6</sub>	V <sub>7</sub>	I <sub>M7</sub>	I <sub>7</sub>	IV <sub>6</sub>	IV <sub>m6</sub>	III <sub>m7</sub> VI <sub>7</sub>	II <sub>m7</sub> V <sub>7</sub>	I <sub>M7</sub>
-----------------	-----------------	----------------	-----------------	----------------	-----------------	------------------	-----------------------------------	---------------------------------	-----------------

レベル 6 (テンションの導入)

I <sub>M9</sub>	IV <sub>6</sub>	V <sub>9</sub> V <sub>9</sub>	I <sub>M9</sub> I <sub>7</sub> <sup>9</sup>	IV <sub>6</sub>	IV <sub>m6(add9)</sub>	III <sub>m7</sub>	III <sub>9</sub> <sup>b13</sup>	II <sub>m7</sub>	II <sub>9</sub> <sup>b13</sup>	I <sub>M9</sub>
-----------------	-----------------	-------------------------------	---	-----------------	------------------------	-------------------	---------------------------------	------------------	--------------------------------	-----------------

図 5: コードアレンジの例 (太枠: アレンジを行ったコード)

2.3 コード進行のアレンジ

生成される曲の多様性を実現するためには、コード進行のパターンを多数所持しておく必要があるが、様々なコード進行パターンを予めすべて用意するのは非現実的である。本手法では、用意されたコード進行パターンに対して複数のアレンジを施すことにより、コード進行パターンの多様性を確保している。アレンジは、例えば以下のようなレベル感を設定して行う。

[レベル 0] 基本コード

- ・ 図 4 のコード構成のみ

[レベル 1] ドミナントコード

- ・ V を V<sub>7</sub> に置き換え
- ・ ドミナントコード (V 及び V<sub>7</sub>) をツーファイブ (II<sub>m7</sub>→V<sub>7</sub>) に置き換え
- ・ sus4 の導入 (V<sub>sus4</sub>→V<sub>7</sub>)

[レベル 2] トニック代理コード

- ・ I を III<sub>m7</sub>, VI<sub>m7</sub> あるいは III<sub>m7</sub>→VI<sub>m7</sub> に置き換え

[レベル 3] セカンダリドミナント

- ・ VI<sub>m7</sub>→II<sub>m7</sub> を VI<sub>7</sub>→II<sub>m7</sub> に置き換え
- ・ I→IV を I→I<sub>7</sub>→IV に置き換え
- ・ II<sub>m7</sub>→V<sub>7</sub> を II<sub>7</sub>→V<sub>7</sub> に置き換え
- ・ VII<sub>m7</sub><sup>-5</sup>→III を VII<sub>7</sub>→III<sub>7</sub> に置き換え

[レベル 4] サブドミナントマイナー

- ・ IV→I 及び IV→III<sub>m7</sub> における IV を IV<sub>m</sub> または IV→IV<sub>m</sub> に置き換え

[レベル 5] 4 和音化

- ・ I を I<sub>add9</sub> または I<sub>M7</sub> に置き換え
- ・ IV を IV<sub>6</sub> または IV<sub>M7</sub> に置き換え
- ・ IV<sub>m</sub> を IV<sub>m6</sub> または IV<sub>m7</sub> に置き換え

[レベル 6] テンションコード

- ・ 9 度, 11 度, 13 度音の導入
- ・ テンションリゾルの導入

種類	基本コード	代理コード
トニック	I	III <sub>m7</sub> , VI <sub>m7</sub> など
サブドミナント	IV	II <sub>m7</sub> など
ドミナント	V	VI <sub>m7</sub> <sup>(5)</sup> , II <sub>7</sub> <sup>b</sup> など

図 6: 代理コードの例

ここに挙げた例以外にも、例えばサブドミナント及びドミナントコードの代理コードへの置き換えや、ディミニッシュコードの導入など様々なアレンジ手法があるが、本手法に関する本質ではないためここでは詳細を述べない。

上記のようにレベル分けを行うことにより、ユーザはレベルの違いによるコードの印象の変化、すなわち編曲の効果を感じることができるようになる。ただし、アレンジレベルが高い変換が常に音楽的に高度である、あるいは正しいということではなく、単にアレンジの違いを示しているだけであり、ユーザの好みに応じてアレンジレベルを変えることが出来るようにしてもよい。また、アレンジを施すコードについても、コード進行パターンに含まれるすべてのコードに対して自動的にアレンジを行うようにしても良いし、アレンジを行うコードをユーザが選択できるような手段を用意しても構わない。

## 2.4 コード進行とメロディのマッチング

2.3 節で生成されたすべてのコード進行パターンと入力メロディのマッチングスコアの計算を行い、もっともスコアの高いものを選択することにより、コード進行の決定を行う。あるいは、スコアの高いものから複数を選択することにより、ユーザがそのコード進行の違いを楽しむことができるようにしてもよい。この段階でのコード構成音は図4や図5などに示すように相対音で書かれているが、キーをCからBまで半音ずつ上げながら設定することにより絶対音化したうえでマッチングスコアを計算し、もっともスコアが高くなるものを選択することにより、コード進行と同時にキーを決定することができる。

コード構成	Key C (A <sub>m</sub> ) 時のコード名	Key C (A <sub>m</sub> ) 時のコード構成音
I	C	ド・ミ・ソ
II <sub>m7</sub>	D <sub>m7</sub>	レ・ファ・ラ・ド
III <sub>m7</sub>	E <sub>m7</sub>	ミ・ソ・シ・レ
III	E	ミ・ソ <sup>#</sup> ・シ
IV	F	ファ・ラ・ド
V	G	ソ・シ・レ
VI <sub>m7</sub>	A <sub>m7</sub>	ラ・ド・ミ・ソ
VI <sub>m</sub>	A <sub>m</sub>	ラ・ド・ミ
VII <sub>m7</sub> <sup>-5</sup>	B <sub>m7</sub> <sup>-5</sup>	シ・レ・ファ・ラ

図 7: コード構成と Key C (Key A<sub>m</sub>) 時のコード構成音の関係

以下に、マッチングスコアの求め方について説明する。最初に、メロディの各音とコード名の対応付けを行う。ここでは、そのメロディ音が開始し終了するまでの間に存在するコードを、そのメロディ音に対応するコードと規定する。例えば図8(A)に示したメロディの最初の音であるソの音は、それが開始し終了するまでの間に存在するコードはCであるため、対応コードはCとなる。図8(B)は、メロディ音の中にシンコペーションがある場合の対応法を示している。6番目の音であるドの音及び最後の音であるファの音はシンコペーションのリズムを持つ音である。この場合、対応付けられるコードは音が開始した時のコードと終了した時のコードの両方とする。つまり、ドの音に対応するコードはB<sup>b</sup><sub>M7</sub>及びA<sub>m7</sub>、ファの音に対応するコードはA<sub>m7</sub>及びD<sub>m7</sub>の両方である。図8(C)は、メロディ音が開始してから終了するまでに複数のコードを経ている場合を示している。メロディの最初の音であるラの音は、3つのコード(A<sub>m7</sub>, A<sub>7</sub>及びD<sub>m7</sub>)の区間と重なっているが、この場合はこれら3つすべてと対応付けられるものとする。

次に、コード構成音及びスケール(アベイラブルノート、アボイドノート、及びスケール外音)の決定を行う。コー

ド構成音は基本コードの場合は図4に示すように決定すればよく、アレンジされたコードの場合もそのコード名に合わせて音を一意に決定することができる。スケールの一例を図9に示す。この図中で、◎はコード構成音(かつアベイラブルノート)、○はアベイラブルノート、△はアボイドノート、×はスケール外音を示している。スケール音はコード進行等に応じて音楽理論に基づいて決定することができるが、ここでは詳細を述べない。

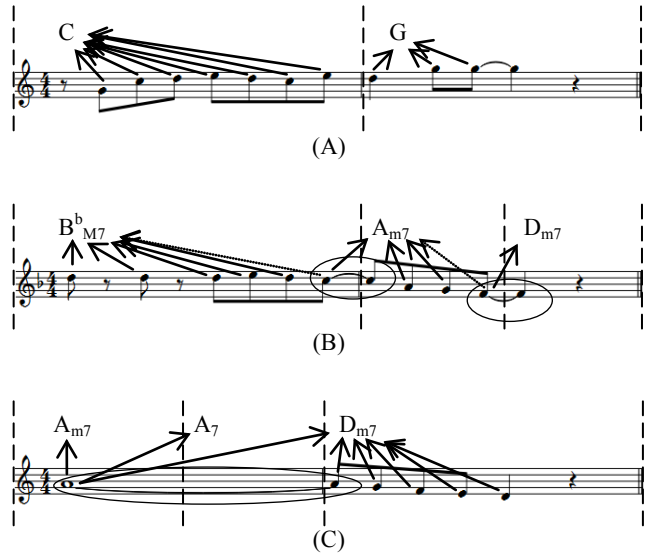


図 8: メロディ音とコードの対応付け例

		メロディノート											
		I	I <sup>#</sup>	II	II <sup>#</sup>	III	IV	IV <sup>#</sup>	V	V <sup>#</sup>	VI	VI <sup>#</sup>	VII
コード	I	◎	×	○	×	◎	△	×	◎	×	○	×	○
	II <sub>m7</sub>	◎	×	◎	×	○	◎	×	○	×	◎	×	△
	III <sub>m7</sub>	○	×	◎	×	◎	△	×	◎	×	○	×	◎
	IV	◎	×	○	×	○	◎	×	○	×	◎	×	○
	V	△	×	◎	×	○	○	×	◎	×	○	×	◎
	VI <sub>m7</sub>	◎	×	○	×	◎	△	×	◎	×	◎	×	○
	VII <sub>m7</sub> <sup>-5</sup>	△	×	◎	×	○	◎	×	○	×	◎	×	◎

図 9: スケールの例

次に、以下に示す7種類の観点より、メロディ音とコードのマッチングスコア計算を行う。

### (1) アベイラブルノート

各メロディ音がアベイラブルノートか否かを判定し、アベイラブルノートであればメロディとコードがマッチしていると判断してスコアに加点を行う。これは、アベイラブルノート以外の音がメロディに含まれると、聴感上の違和感が大きくなるためである。ただし、スケール外音あるいはアボイドノートを経過音等として使用しても構わないた

め、メロディ音がそれらの音の場合でも減点はしない。加減点については単純に1点ずつとしてもよいが、当該音の長さを加減点するようにすると、長さによる重み付けが行われるため、よりよい指標となることが期待できる。なお、メロディ音がシンコペーションあるいは複数のコードにまたがる場合は、対応するすべてのコードにおいてアベイラブルノートか否かを判定し、一つでも満たさないコードがある場合には加減せず、すべてのコードで満たす場合にのみ加減を行うようにする。



図 10: アベイラブルノート (破線) とアボイドノート (実線) で構成されたメロディの例

### (2) コード構成音

メロディ音がコード構成音の場合には加減を行う。これは、メロディ音がアベイラブルノートで構成されていたとしても、コード構成音の一つも含まない場合にはメロディらしさが損なわれるためである。よって、入力メロディ音中に一つもコード構成音が含まれない場合にはスコアの減点を行うこととする。なお、ここでのコード構成音は、3和音(1度, sus4を含む3度, 及び5度の音)と7度のみとし、6度やテンションノートは含まないものとする、よりメロディの安定感につなげることができる。



図 11: コード構成音の一つも含まないメロディの例

### (3) 最後のメロディ音

入力メロディの最後の音がコード構成音の場合にはメロディのフレーズが安定して終了する感覚が得られるため、より良いマッチング結果となることを期待してスコアを加減する。また、最後の音がスケール外音あるいはアボイドノートの場合には、メロディフレーズが非常に不安定な感じになってしまうため、減点を行う。



図 12: 最後のメロディ音がアボイドノートの例

### (4) リズム (休符)

入力メロディ音のうち、あるメロディ音とその次のメロディ音の間に休符がある場合、その2つのメロディ音が共にアベイラブルノートで、かつ少なくともどちらか一方がコード構成音である場合には加減を行う。これは、休符前後の音はメロディフレーズにおける印象が強く残りやすい

ためである。ただし、少なくともどちらか一方がスケール外あるいはアボイドノートの場合には減点を行う。



図 13: 休符の前後の音がともにコード構成音ではない例

### (5) リズム (シンコペーション)

メロディ音のリズムがシンコペーションの場合も(4)と同様に印象に残りやすいため、そのメロディ音または次のメロディ音が共にアベイラブルノートで、かつ少なくともどちらか一方がコード構成音である場合に加減を行う。ただし、少なくともどちらか一方がスケール外あるいはアボイドノートの場合には減点を行う。



図 14: シンコペーションの次のメロディ音がコード構成音の例

### (6) 跳躍

メロディ音がその前後のメロディ音よりも所定の音程(例えば5半音)以上離れている場合、そのメロディ音がコード構成音の場合には加減を行う。ただし、そのメロディ音がアボイドノートまたはスケール外音の場合には減点を行う。



図 15: 跳躍音がコード構成音の例

### (7) 長さ

メロディ音の長さがある所定の長さ(例えば1拍半)より長い場合、そのメロディ音がコード構成音であれば加減を行う。アボイドノートあるいはスケール外音の場合には減点を行う。



図 16: 長いメロディ音がアボイドノートの例

ここまで複数の加減点を行う条件について説明を行ってきたが、あるメロディ音が複数の条件を満たしている場合、加減点は1度だけ行うようにしてもよいし、すべての場合に行うようにしてもよい。また加減点の重み付けについては、それぞれの条件ごとに変えてもよいし、同じ値を用いても構わない。

### 3. 入力メロディから決定されたコード進行及びキーの例

図 1 のメロディを入力とした場合に、本手法により決定されたコード進行及びキー（及び生成された続きのメロディ）の例を図 17 に示す。

Figure 17 displays four musical staves, each representing a different key derived from the melody in Figure 1. Each staff shows a sequence of chords over a melodic line. The keys and their corresponding chord progressions are:

- Key G:** C, G<sub>on</sub>B, A<sub>m</sub>7, D<sub>7</sub>, G, G<sub>7</sub>
- Key C:** D<sub>m</sub>7, G<sub>7</sub>, E<sub>m</sub>7, A<sub>m</sub>7
- Key A<sub>m</sub>:** A<sub>m</sub>9, E<sub>m</sub>7<sub>on</sub>G, F<sub>M</sub>7, E<sub>7</sub>
- Key D<sub>m</sub>:** G<sub>m</sub>7, C<sub>7</sub>, F<sub>M</sub>7, B<sup>b</sup><sub>M</sub>7

図 17: 図 1 のメロディを入力とした場合に本手法によって決定されたコード進行及びキーの例

### 4. おわりに

2 小節のメロディを入力とし、それに違和感なくマッチするコード進行及びキーを決定する方法について述べた。現状は、入力メロディに 2 章の冒頭に述べたような条件を課しており、例えば 16 分音符が含まれていたり 2 小節を超える長さを持つメロディ、あるいは 4 分の 4 拍子以外の拍子を持つメロディには対応しておらず、ユーザが入力すると思われるメロディスタイルのすべてを網羅しているとは言い難い。今後は、さらに幅広いスタイルの入力メロディに対応するとともに、本報告では述べなかった入力メロディを生かしてそれに続くメロディを生成するアルゴリズムについても改善を行いたいと考えている。

### 参考文献

- [1] 深山覚, 中妻啓, 米林裕一郎, 酒向慎司, 西本卓也, 小野順貴, 嵯峨山茂樹: Orpheus: 歌詞の韻律に基づいた自動作曲システム, 情報処理学会研究報告, 2008-MUS-76-30, pp.179-184, 2008
- [2] 南高純一, 猪野真弓, 佐藤邦雄, 森川重則: リズム, 和声, 調性を考慮した自動作曲システム-MAGIC”, 情報処理学会第 37 回全国大会, pp.1960-1961, 1988
- [3] Ian Simon, Dan Morris and Sumit Basu: MySong: Automatic Accompaniment Generation for Vocal Melodies, Proceedings of Computer-Human Interaction, 2008
- [4] MAISTAMUSIC: <http://maistamusic.com/>