

卒業論文

Profit Sharing を用いた自動作曲

5年 情報工学科 18番

小島 京輔

指導教員 東海林智也

目次

1.	目的	1..2
2.	自動作曲システムの構築	3..11
2.1.	システムの概要（流れ図を書く）	3
2.2.	度数分布の作成	3..5
2.3.	Profit Sharing による音符作成	5..10
2.3.1.	Profit Sharing のサイクル	6
2.3.2.	報酬関数	7
2.3.3.	Profit Sharing の評価関数	8
2.3.4.	メロディ作成での使用	8
2.3.5.	メロディとなる条件	9..10
2.4.	音データの作成	10
2.5.	度数分布を用いた音の長さの決定	11
3.	実験	12..
3.1.	実験環境	12
3.2.	結果	12..13
3.3.	考察	13
4.	まとめ	14
	参考文献	15

1 章 目的

近年コンピュータが身近になりパソコンを用いて曲を作ることが出来るようなソフトウェアを簡単に入手できるようになった。それに伴い楽器を演奏できない人でも、楽器を演奏することなく曲を作ることが出来るようになった[1]。その結果、趣味で作曲を行うような人も現れたが、曲を作るためには音楽についての知識は必要不可欠であり、本人の感性や経験の問題などもあり素人が曲を作るということが難しいという状況に変わりはない。そこで本研究では、音楽についての知識の有無に関わりなく誰もが簡単に曲を作ることが出来るような自動作曲を行うツールを作成することをめざす。

自動作曲を行う手法は種々存在する。例えば、『知識ベースを用いる手法』、『あらかじめ用意されたパターンを組み合わせる手法』、『数値データを与えて音楽理論に基づいて編集する手法』、などがあげられる[2]。最近では多項式などの数学的手法を利用した自動作曲の研究や、脳波や、DNAなどの生体のゆらぎと呼ばれるものを用いた自動作曲の研究も行われている。さらには図形とその色彩から音符を生成したり、動画の場面変化から音符を生成するなどの自動作曲の研究や、遺伝的アルゴリズムを用いた自動作曲も行われている。

このうち『知識ベースを用いる手法』とは、既存の曲から『音符の遷移』、『リズム』、『コード進行』などを取り出して知識ベースを作成し、それをもとに曲を作成する手法である。

また、『あらかじめ用意されたパターンを組み合わせる手法』とは、あらかじめ音符の遷移やリズムの変化のパターンを決めてデータベースを作成し、そのパターンをランダムにあるいはあるルールに基づいて組み合わせる手法である。

さらに、『数値データを与えて音楽理論に基づいて編集する手法』とは、乱数などの数値データを音楽理論として用意したルールを用いて音符やリズムなどに換えて曲を作る手法である。

昨年度までの従来研究では、『知識ベースを用いる方法』を自動作曲の方法として選択した。その理由としては数学的知識がそれほど必要ないこと、生体のゆらぎのようなものを測定する必要がないことなどがあげられるが、最も大きな理由は知識ベースに知識ベースを作成する際に使用した曲の特徴を保存することが出来ることであり、これによってジャンルに関する知識がない人が自動作曲ソフトを使用する際に迫られることがあるジャンル選択をなくすことが出来ると考えたからである。このような理由より昨年度の研究では Q 学習を用いて自動作曲を行った。その結果、出力されたメロディは不自然な音階の遷移こ

そ取り除けたものの、自然な曲とは言えないものになってしまった。また、Q学習を用いると学習が終了するまでに大幅な時間がかかった。以上より音楽的な出力を得られるようにすること、学習に必要な時間の短縮を行うことなどの課題が残った。

そこで、本研究では音楽的な出力を得られるようにすることと、さらに学習に必要な時間の短縮することの2点について改善を行う。その手法として今回は、強化学習の手法の1つである **Profit Sharing** を用いることとした[3][4]。**Profit Sharing** は報酬関数を変化させることによって価値の与え方をいくつか選択でき、ルールの価値を一括して更新するので学習が速いという特徴がある。

本研究では知識ベースに既存の曲から抽出した特徴を記憶させ、知識ベースのなかにあるデータを **Profit Sharing** を用いてコンピュータに学習させることによって、知識ベースが持つデータの中でも最適な組み合わせを見つけてそれをメロディとして出力させる。さらに、シミュレーションを行うため考察に基づいたプログラムの作成とその評価を行う。

2章 自動作曲システムの構築

自動作曲システムを構築するに当たって、本研究では、『知識ベースを用いる方法』を自動作曲の方法として利用する。

2.1. システムの概要

自動作曲システムを構築する流れ図を以下に示す。(図1)

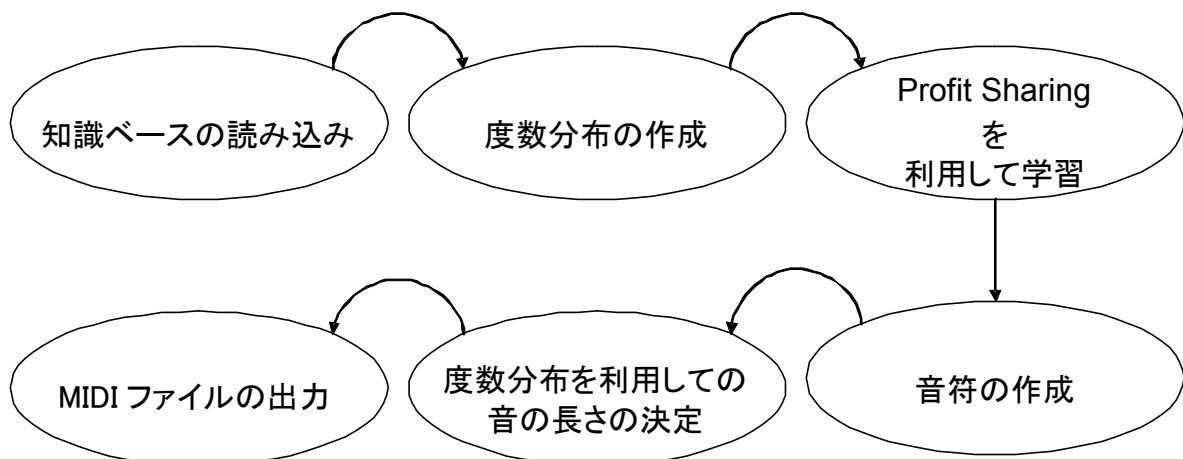


図1 システムを構築するための流れ

2.2. 度数分布の作成

昨年度までの従来研究では、度数分布を音程の遷移回数記録した度数分布と、音符の遷移回数記録した度数分布を知識ベースとして用いて作成した。例えば以下のような音程進行(図2)があったとき、それぞれの遷移を調べて頻度を数えて表を作った(表1、表2)。

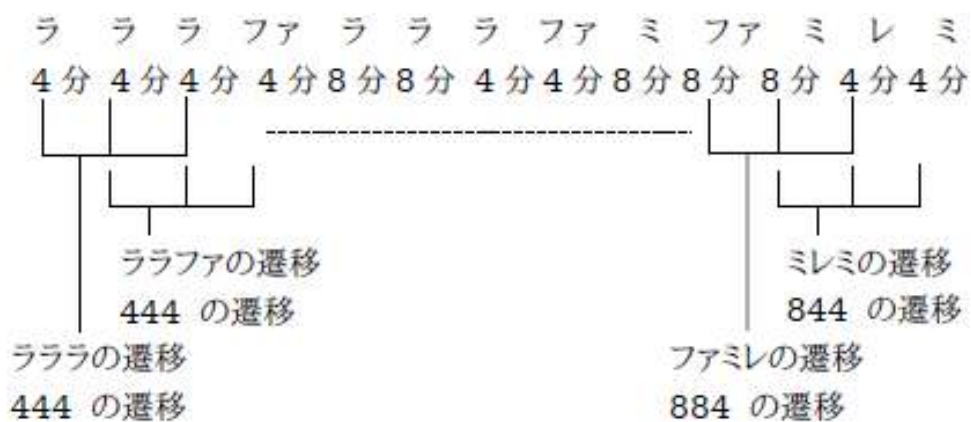


図 2 音程と音符の遷移

1 音め	2 音め	3 音め	回数
ラ	ラ	ラ	2
ラ	ラ	ファ	2
ラ	ファ	ラ	1
ファ	ラ	ラ	1
ラ	ファ	ミ	1
ファ	ミ	ファ	1
ミ	ファ	ミ	1
ファ	ミ	レ	1
ミ	レ	ファ	1

表 1. 音程の度数分布

1 音め	2 音め	3 音め	回数
4	4	4	2
4	4	8	2
4	8	8	2
8	8	4	2
8	4	4	2
8	8	8	1

表 2. 音符の度数分布

このようにして度数分布表を作成した。ただ、曲は調ごとに根音と言うものが存在しているため、表 1 のようにラ、シ、ドなどの絶対的な音階を用いて処理を行う場合問題が起こる。そのため、曲の根音を基軸とした相対的音階を用いて度数分布表を作成した。表 1 をドを根音とした相対的音階で表すと表 3 の

用になった。ただし、[ド = 0、レ = 1、ミ = 2、ファ = 3、ソ = 4、ラ = 5、シ = 6]のように音階に数字を割り当てた。

1音め	2音め	3音め	回数
ラ-ド(5)	ラ-ド(5)	ラ-ド(5)	2
ラ-ド(5)	ラ-ド(5)	ファ-ド(3)	2
ラ-ド(5)	ファ-ド(3)	ラ-ド(5)	1
ファ-ド(3)	ラ-ド(5)	ラ-ド(5)	1
ラ-ド(5)	ファ-ド(3)	ミ-ド(2)	1
ファ-ド(3)	ミ-ド(2)	ファ-ド(3)	1
ミ-ド(2)	ファ-ド(3)	ミ-ド(2)	1
ファ-ド(3)	ミ-ド(2)	レ-ド(1)	1
ミ-ド(2)	レ-ド(1)	ファ-ド(3)	1

表 3. 相対的音階の度数分布表例

音符の度数分布には手を加えず、そのまま利用していた。このような度数分布を用いることで、不自然な音の遷移をなくし、メロディの変化を滑らかなものに出せると考えられる。

2.3. Profit Sharing による音符作成

Profit Sharing とは強化学習の手法の一つであり経験強化型に分類される [5][6]。本研究では、この Profit Sharing を用いて音符作成を行う。

エージェントが報酬を得たときにその報酬を得るまでに使用したルールを一括して強化を行う。今回の研究では、この Profit Sharing という手法で学習を行わせた。その理由として、Profit Sharing は最適解ではなく、準最適解を求めるため、必ずしも最適な解を求められる保証は無い。そこで、このようなアクセントを入れることで出力されるメロディを自然な流れにできるのではないかと考えた。

2.3.1. Profit Sharing のサイクル

Profit Sharing のサイクルは以下のようなになる。

- (1) エージェントは「環境」から現在の状態を取得する。
- (2) 状態に応じて、実行可能な行動群の中から評価関数に基づいて最もよいと考えられる行動を一つ選択してそれを環境に対して実行する。
- (3) エージェントは環境から行った行動の結果として報酬を得られた場合は(4-1)を実行し、報酬を得られなかった場合は、(4-2)を実行する。
 - (4-1) エージェントは結果として得た評価に基づいて今まで保持した行動に対して評価関数を更新する。
 - (4-2) 現在の状態と実行した行動を保持する。
- (5) 状態の遷移を行う。

このサイクルを図示したものを図3に示す。

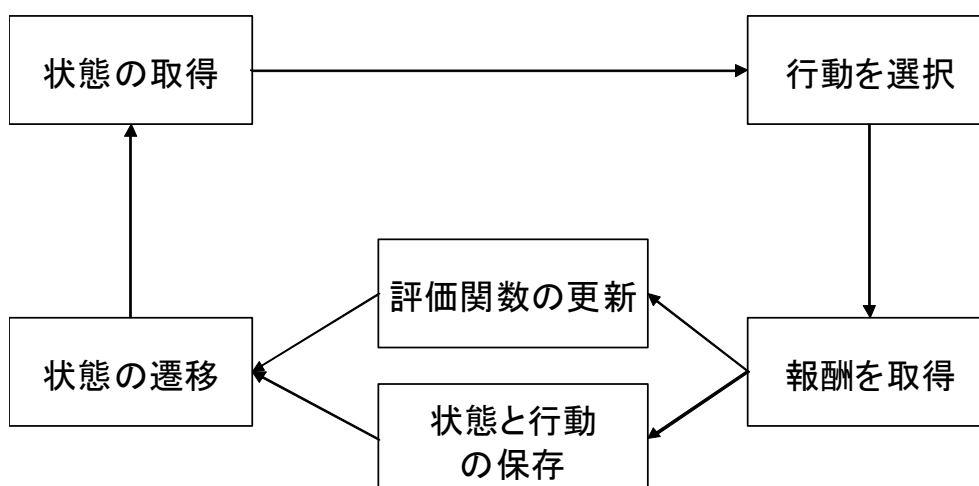


図3 Profit Sharing のサイクル図

2.3.2. 報酬関数

今回用いた Profit Sharing では、報酬関数 $r(t)$ を用いて価値を更新する。報酬関数は、さまざまな関数が考えられ、例えば、下図に示すように、時間 t (ルール実行回数) に依存せず固定値である場合(a)、線形的に増加する場合(b)、逃避増大的な場合(c)などがあげられる。これらの関数を図 4 に図示する。

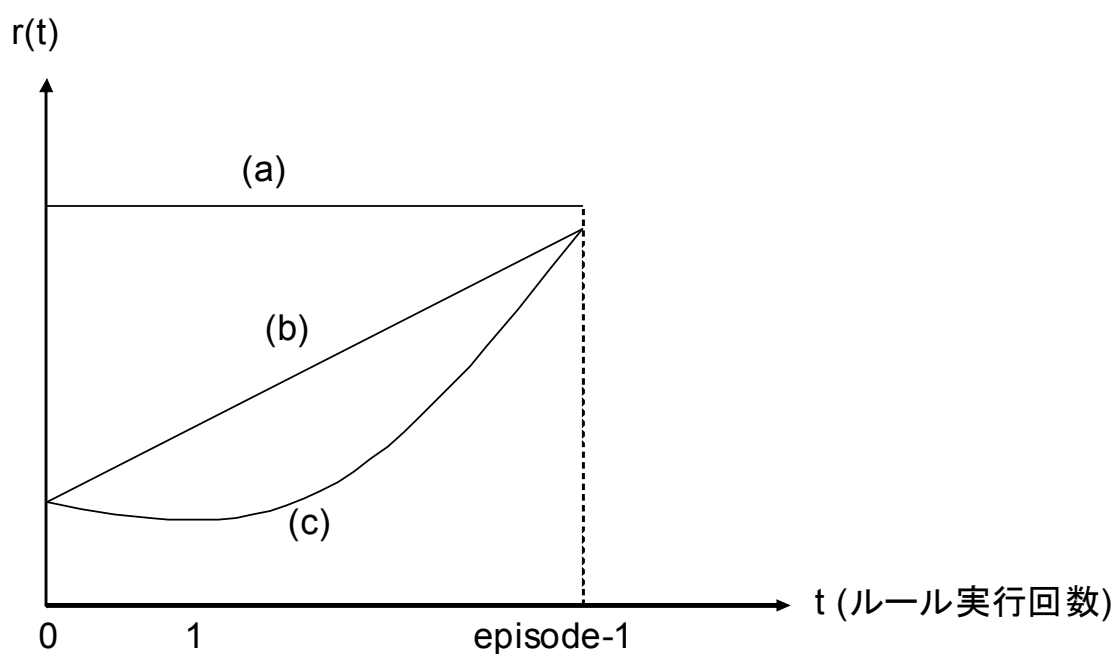


図 4 Profit Sharing の報酬関数

2.3.3. Profit Sharing の評価関数

Profit Sharing の評価関数は、状態と行動から表される関数で表現され、 $Q(s,a)$ となる。ここで[s:状態], [a:行動]とする。エージェントはこの関数をもとに現在の状態 s において評価関数の値が最大となっている行動 a を選択し、環境に対して実行する。この関数は、環境から受け取る行動の価値によって更新されて増減して行き、最終的に報酬関数で設定される値に近い値を持つ用になる。図 5 に評価関数の更新式を記す。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + C_{bid} [r(t) - Q(s_t, a_t)]$$

$Q(s_t, a_t)$: t 時間における価値観数

s_t : t 時間における状態

a_t : t 時間において取った行動

a : 状態 s_t において実行できる行動

C_{bid} : 学習率

図 5 評価関数の更新式

2.3.4. Profit Sharing のメロディ作成への応用

Profit Sharing が準最適解を求められることからメロディの生成にこれを応用する。具体的には度数分布よりデータを与え、音同士をつなぎ合わせてそれをメロディとした。そのメロディがメロディを構成するために必要なルールに従っているようであれば報酬を与え、ルールから外れているようであれば罰を与えることで、作られるメロディを制御しデータより得られるメロディをルールに従う最適なものにできると考えた。

2.3.5 メロディとなる条件

メロディとは音楽を構成する要素の1つであり、『リズム』、『ハーモニー』と並んで音楽の3要素と呼ばれる。この研究では、このメロディを作成することを目的としている。メロディは種々の条件を持ち、クラシックの場合メロディを作る際にいくつかのルールを使用する[7][8][9][10][11]。

また、良いメロディと呼ばれるメロディには様々な条件が存在する。例えば、『音が跳びすぎない』、『調性感やコード感がある』、『リズムカルである』、『繰り返しと変化を適度に含んでいる』などがあげられ、この他にも様々な条件を持っている[12]。この研究では「音が跳びすぎない」という点のみに主点をおいて研究を進めた。

『音が跳びすぎない』ということがどういうことかを説明するため例を示す(図6)。左側の図では前の音と後の音の高さの差が小さくなっている。このような進行を順次進行と呼ぶ。右の図では前の音と後の音の高さの差が大きくなっている。このような進行を跳躍進行と呼ぶ。一般的に五線譜で線1本分以上離れている場合に跳躍進行であるとされる[13][14]。『音が跳びすぎない』と言うことはあまりにも大きな跳躍進行を行ってはいけないということである。

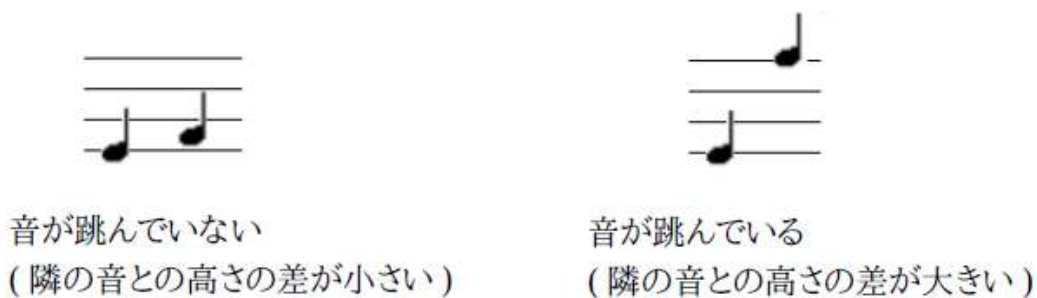


図6 音の高さの差

一般的にメロディの進行は順次進行と跳躍進行によって構成されているが、前の音と後の音の高さの差に大きな変化がなく、音符を線で結んだ場合滑らかな曲線を描く(図7)。このことから人に自然なメロディであると認められるためにはなだらかな変化を持つ『音が跳びすぎないメロディ』を構成する必要があると考えられる。それを実現するために作曲家が用いるような作曲に関するルールを使用した。



図7 メロディの流れの例

クラシックの理論では作曲の基本とされているルールが存在する。この研究では、このルールのうちのいくつかを利用した。以下に基本とされるルールを記す[15]。

1. メロディは順次進行を中心に作り、跳躍進行は3回以上続けない
2. 大きな跳躍進行の後には、逆方向へ進行させる
3. 1つのフレーズの中に音の頂点を2つ以上作らない
4. 大きな跳躍進行の前には、逆方向への順次進行をおく

今回は1、2の教えより、昨年と同様2つのルールを作ってそれを利用することで学習中のエージェントに報酬を与えるか、罰を与えるかを定めることとした。以下にそのルールを示す。

1. メロディは順次進行を中心に作る
2. 大きな跳躍進行の後には、逆方向へ進行させる

2.4. 音データの作成

音データの作成は、エージェントが学習した後に行う。エージェントは移動が行われた後、その移動が正しかったかどうかを報酬あるいは罰で知ることになる。新しいデータを得た結果が今回用いたメロディ作成のためのルール（順次進行中心に作る、跳躍進行の後には逆方向への順次進行を行う）を満たしているならばその音に遷移する度数を報酬として与え、満たしていないならば負の値を罰として与える。こうすることによってエージェントは目的位置に辿り着くまでに得られる報酬を最大にしようとするので、学習を重ねるごとに負の値を得る方向を選択しなくなっていく、また度数が大きい音程を選択するようになる。今回は学習率0.1、負の値として-50を与えた。学習率、負の値は適当に決定した。

2.5. 度数分布を用いた音の長さの決定

音データを作成した後累積分布と Q 学習を用いて音データに当てはめる音符の長さを決定する。長さは長さの遷移を記録した度数分布の度数より累積分布を作成し、乱数を用いて累積分布よりデータを選択した。長さはこのデータの組み合わせが正しいかどうかによって報酬を受け取るか罰を受け取るかを決めた。この際、音データの作成時に利用した方法と同様に、データが持つ「1 音めの音符」、「2 音めの音符」が其々前のデータの「2 音めの音符」、「3 音めの音符」と等しければエージェントは報酬を与えられ、どちらか一方でも異なっているば罰を受ける、というルールを用いた。ただし長さの場合には同じ長さの音符が 5 回以上続く様であれば、その場合似も罰を受けるようにしてある。ここで、与える報酬や罰は、昨年度と同じものを使用した。

3章 実験

今回自動作曲するためプログラム作成の環境を以下に示す。

OS: windows XP

開発ツール: Visual C++ 2005 Express Edition

3.1. 実験方法

度数分布の入力として童謡の楽譜の「ぶんぶんぶん」や「エーデルワイス」を使用した[16]。学習回数は音データの作成時、長さの決定時共に 400 回とした。これは、デバッグ時に音データ、長さ共にこの程度の学習回数で結果の収束が認められたことを理由とする。Profit Sharing におけるパラメータや、報酬などは 2 節に記述したものを使用する。プログラムの実行時は引数として、出力として得られる MIDI ファイルの名前、度数分布のファイル名前をそれぞれ与える。このプログラムを 10 回起動し、その結果得られる MIDI ファイルを被験者 6 名に聞いてもらいメロディと考えられるかどうかを調べる。

3.2. 結果

MIDI ファイルを被験者に聞いてもらった結果、平均して 3.6 名がメロディと考えられると答えた。そのため、今回生成されたメロディは、自然な曲はいえないと考えられる。またエージェントが学習するためにかかった時間は 4/1 短縮された。以下に「ぶんぶんぶん」と「エーデルワイス」の度数分布を用いたときに生成されたメロディの楽譜を図示する(図 8)(図 9)。



図8 「ぶんぶんぶん」を用いて作成した MIDI ファイルの楽譜



図9 「エーデルワイス」を用いて作成した MIDI ファイルの楽譜

3.3. 考察

図8では音符の滑らかさがあまり見られなかった。図9では、単一の音が過度に続く傾向が現れた。この理由として、単一の音が続き過ぎないようにするためのルールの追加を行えば改善できると思われる。また、出力されたメロディは、度数分布作成に使用した楽曲の特徴があまり見られなかった。これもルールの追加を行えば改善できると思われる。

4. まとめ

今回自動作曲ツールを作成するにあたって、学習に必要な時間の短縮することと、より音楽的な出力を得られるようにすることの 2 点について改善を行ったが、学習に必要な時間の改善は、**Profit Sharing** で学習をさせることによって学習時間を短縮させることができた。しかし、出力されたメロディは同一の音の出現頻度が高くなり、まだ自然なメロディとはいえない。その理由として学習するときの報酬値をその度数としていることにより、一定の音符が選択されて価値も高くなったことが原因だと思われる。

今後の課題としては、学習ルールの改善やメロディのルールの改善などをしてより自然なメロディにすることがあげられる。

参考文献

- [1] 熊谷 洋希:Q 学習を用いたメロディの作成
- [2] 遺伝的アルゴリズムによる自動作曲
: http://www.nlab.sogol.ynu.ac.jp/ynu/r_samples/no_10_3.html
- [3] 三上 貞芳, 皆川 雅章 著:強化学習, (森北出版, 2004)
- [4] 松本 啓之亮, 黄瀬 浩一, 森 直樹:知能システム工学入門, (コロナ社, 2003)
- [5] 宮崎 和光, 木村 元, 小林 重信:Profit Sharing に基づく強化学習の理論と応用, 人工知能学会誌, Vol. 14, No. 5, pp. 800-807 (1999)
- [6] 高玉 圭樹:マルチエージェント学習ー相互作用のなぞに迫るー,
(コロナ社, 2003)
- [7] 水谷 友香, キッシー岸田 著: Cdで覚えるやさしい楽譜の読み方 読み方・書き方から作曲・編曲までマスターしよう, (成美堂出版, 2005)
- [8] メロディ
: <http://ja.wikipedia.org/wiki/%E3%83%A1%E3%83%AD%E3%83%87%E3%82%A3>
- [9] 関 和則 著:DTMのための全知識, (リットーミュージック, 2000)
- [10] デイブ・スチュワート 著, 藤井 美保 訳:絶対わかる!曲作りのための音楽理論 必要最小限の理論知識だけでOK!, (リットーミュージック, 2006)
- [11] 御池 鮎樹 著:裏口からのMIDI入門 理論不要の作曲道, (工学社, 2004)
- [12] 勝田 哲司:月刊DTMマガジン 11月号 シリーズ 自動作曲研究所 音階を組み込もう, p. 64, (寺島情報企画, 1999)

[13] 北川 祐 著:基本はドレミファソラシド 絶対わかる!コード理論, (リットーミュージック, 2004)

[14] 意味音:http://imion.jp/index.asp?id=190&b_chkNum=4095

[15] 勝田 哲司:月刊DTMマガジン12月号 シリーズ 自動作曲研究所 クラシックに学ぶメロディの作成技法, p. 110, (寺島情報企画, 2000)

[16] 安田 進:やさしいポピュラーピアノ曲集2, (全音楽譜出版, 1988)