

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ INFORMATYKI I ZARZĄDZANIA

KIERUNEK: Informatyka

SPECJALNOŚĆ: Inteligentne Systemy Informatyczne

PRACA DYPLOMOWA
MAGISTERSKA

System “improvizujący” solo do zadanego
podkładu muzycznego.

Opracowanie (adaptacja) metod i
implementacja prototypu.

AUTOR:

inż. Dariusz Olszowy

PROMOTOR:

dr Marek Kopel

OCENA PRACY:

SPIS TREŚCI

Spis rysunków	3
Spis tabel	5
Spis listingów	6
Streszczenie pracy	7
Wstęp	8
1. Definicja oraz podstawowe założenia tematu. Wstępny opis podejścia do jego realizacji.	9
1.1. Objaśnienie zakresu tematu pracy i zawartej w nim terminologii	9
1.2. Trudności i problemy tworzenia improwizacji muzycznych	10
1.3. Praktyczne wykorzystanie systemu	10
1.4. Dodatkowe założenia do tematu pracy	11
1.5. Przesłanki skłaniające do realizacji zadania z pomocą maszyny	11
Znajomość fraz muzycznych	12
Umiejętności muzyka	12
Wiedza teoretyczna muzyka	13
1.6. Cechy poprawnej improwizacji	13
1.7. Subiektywizm w ocenie utworu muzycznego	13
I. Wstępny projekt metody improwizacji	
2. Historia i przegląd rozwiązań z dziedziny kompozycji wspomaganej komputerowo	16
3. Sekrety Polihymnii czyli podstawowa terminologia muzyczna	19
3.1. Zapis nutowy	19
3.2. Metrum	19
3.3. Takt	20
3.4. Tempo	20
3.5. Znaki chromatyczne	20
3.6. Ton, półton	20
3.7. Interwały	20
3.8. Skala, gama	21
3.9. Tonacja	21
3.10. Akordy	21
3.11. Triada harmoniczna	22
3.12. Progresja akordów	23

3.13. Triola	23
3.14. Synkopa	23
3.15. Blues	23

II. Szczegółowy opis zaprojektowanej metody improwizacji

4. Architektura systemu	29
4.1. Przykład przebiegu procesu improwizacji	30
5. Szczegółowy opis metody improwizacji oraz jej implementacji w systemie	34
5.1. Narzędzia oraz oprogramowanie użyte podczas tworzenia systemu	34
5.2. Reprezentacja zapisu nutowego w systemie – ABC NOTATION	35
5.3. Sposób reprezentacji głównych elementów muzycznych	38
5.4. Generator taktów	41
5.4.1. Wstęp do algorytmów genetycznych	41
5.4.2. Struktura pojedynczego osobnika	42
5.4.3. Operacje „świadome muzycznie”	43
5.4.4. Funkcja oceny – podobieństwo melodii	44
5.4.5. Ostateczna postać funkcji celu	49
5.4.6. Mutacje	50
5.4.7. Strategie selekcji	52
5.4.8. Warunek stopu algorytmu genetycznego	53
5.5. Etap kompozycji	54
5.6. Etap korekcji	54
5.6.1. Korekcja względem tonacji	54
5.6.2. Korekcja względem progresji akordów	55
5.7. Etap przetwarzania końcowego	58
5.7.1. Przycinanie improwizacji	59
5.7.2. Zbieżność głównej linii melodycznej do toniki	59
5.8. Odtwarzanie oraz zapis improwizacji	60
6. Wyniki działania metody oraz ich ocena	63
6.1. Próba obiektywnej oceny generowanych improwizacji	63
6.2. Analiza wybranej improwizacji	66
7. Potencjał zaprojektowanej metody oraz kierunki jej rozwoju	73
Zakończenie	76
Dodatki	77
A. Opis struktury plików projektu. Jego kompilacja oraz uruchomienie	78
B. Opis graficznego interfejsu użytkownika aplikacji	79
Bibliografia	84

SPIS RYSUNKÓW

3.1	Prezentacja podstawowych aspektów graficznego zapisu nut	25
3.2	Metrum i jego wpływ na budowę taktów	26
3.3	Przykład działania znaków chromatycznych. Dźwięk f został podwyższony o półton do dźwięku fis . Dźwięk c został natomiast obniżony o jeden półton do ces z użyciem bemola. Działanie krzyżyka zostało anulowane dopiero w ostatniej nucie poprzez znak kasownika.	26
3.4	Budowa akordu c -dur oraz jego powiązanie z gamą c -dur	26
3.5	Przykład powszechnie stosowanej bluesowej progresji akordów wprowadzonej przez muzyka Williama Christophera Handy’ego, zbudowanej dla gamy c -dur.	27
3.6	Przykład triol – ćwierćnutowej oraz ósemkowej.	27
3.7	Przykład synkop – bazującej na ósemkach oraz na ćwierćnutach.	27
4.1	Architektura systemu improwizującego.	33
5.1	Zapis poszczególnych wysokości dźwięków w notacji ABC.	36
5.2	Zapis długości trwania nuty G w notacji ABC dla referencyjnej wartości pola L odpowiadającej ósemce.	36
5.3	Wyrażenie przedstawiające zapis pojedynczej nuty w notacji ABC	37
5.4	Przykłady zapisu pauz, synkop oraz triol w notacji ABC.	37
5.5	Sposoby odgrywania podkładu bazującego na akordzie c -dur	40
5.6	Przykład krzyżowania jednopunktowego.	42
5.7	Przykłady tablicowej reprezentacji taktów.	43
5.8	Przykłady krzyżowania tablicowej reprezentacji taktów.	44
5.9	Przykłady uproszczenia tablicowej reprezentacji trioli oraz synkopy do odpowiadającym im form złożonych z podstawowych nut.	45
5.10	Przykład budowy konturu interwałowego.	46
5.11	Przykład budowy konturu wędrówki melodii.	46
5.12	Przykład budowy wektora zróżnicowania rytmu dla różnych wartości parametru N. Najmniejszą wartością rytmiczną jest szesnastka.	47
5.13	Element graficznego interfejsu użytkownika służący do porównywania dwóch taktów pod względem ich konturów.	48
5.14	Wpływ parametrów α , σ oraz μ na kształt bazowej funkcji celu. Jeśli dany parametr nie jest podany, przyjmuje on wartość domyślną: $\alpha = 2$, $\mu = 0$ oraz $\sigma = 0.25$	50
5.15	Ostateczny kształt funkcji celu $f_{cel}(x)$ dla parametrów $\mu = 0.8$, $\sigma_1 = 0.025$ oraz $\sigma_2 = 0.005$	51
5.16	Działanie przednutki krótkiej górnej, notacja oraz jej realizacja w systemie.	51

5.17	Działanie mordentu górnego, notacja oraz jego realizacja w systemie.	52
5.18	Działanie ozdobnika – trioli bazującej na mordencie.	52
5.19	Działanie ozdobnika tworzącego przejście pomiędzy nutami dla jedno- oraz trójelementowego przejścia.	52
5.20	Proces korekcji dwóch taktów względem tonacji <i>d-dur</i>	56
5.21	Proces korekcji taktu względem towarzyszącego mu akordu <i>c-moll</i> (Cm).	58
5.22	Operacja przycinania melodii w ramach PRZETWARZANIA KOŃCOWEGO.	59
5.23	Operacja kończenia improwizacji poprzez zbieg melodii do toniki (e^1).	60
5.24	Okno odtwarzania oraz eksportu wygenerowanej improwizacji.	62
6.1	Okno odtwarzania oraz eksportu wygenerowanej improwizacji.	68
6.2	Kształt funkcji celu $f_{cel}(x)$ użytej podczas generacji przykładowej improwizacji. Parametry funkcji to $\mu = 0.8$, $\sigma_1 = 0.025$ oraz $\sigma_2 = 0.005$	68
6.3	Zapis nutowy przykładowej improwizacji.	69
6.4	Pierwszy przykład podobieństwa improwizacji do bazy zagrywek. Wynik mutacji – dodanie trioli.	70
6.5	Drugi przykład podobieństwa improwizacji do bazy zagrywek.	70
6.6	Trzeci przykład podobieństwa improwizacji do bazy zagrywek. Wynik przycięcia oraz mutacji, która dodała przejście pomiędzy dwiema nutami.	71
6.7	Wyniki korekcji tego samego taktu względem dwóch różnych akordów.	71
B.1	Trzy główne elementy graficznego interfejsu użytkownika.	79
B.2	Zakładka „ Ustawienia improwizacji ” graficznego interfejsu użytkownika.	80
B.3	Druga część zakładki „ Ustawienia alg. genetycznego ” graficznego interfejsu użytkownika.	81
B.4	Pierwsza część zakładki „ Ustawienia alg. genetycznego ” graficznego interfejsu użytkownika.	82
B.5	Zakładki „ Ustawienia odtwarzania/MIDI ” graficznego interfejsu użytkownika.	83
B.6	Panel postępu prac nad improwizacją.	83
B.7	Okno odtwarzania oraz eksportu wygenerowanej improwizacji.	83

SPIS TABEL

1.1	Zestawienie głównych elementów składających się na improwizację muzyczną oraz ich teoretyczne odzwierciedlenie w elementach komputerów.	12
3.1	Zestawienie podstawowych interwałów	21
3.2	Zestawienie podstawowych akordów.	22
5.1	Zestawienie przykładowych, użytych w systemie akordów, skal muzycznych, z których się wywodzą oraz stopni skal, których należy unikać podczas improwizacji wokół akordu.	57
6.1	Ocena jakości wygenerowanej przez system improwizacji muzycznej w porównaniu z dysharmonicznym utworem referencyjnym (kakofonią).	64
6.2	Ocena improwizacji po ponownym jej odsłuchaniu, już bez dysharmonicznego odnośnika.	64
6.3	Wyniki drugiej ankiety.	65
6.4	Zestawienie parametrów algorytmu genetycznego, użytego do wygenerowania przykładowej improwizacji.	67

SPIS LISTINGÓW

5.1	Przykład pliku ABC.	35
-----	-----------------------------	----

STRESZCZENIE PRACY

Niniejsza praca opisuje metodę, która improwizuje solo do zadanego podkładu muzycznego. Metodę tę zaimplementowano w postaci prototypu systemu z graficznym interfejsem użytkownika. Zaprojektowana metoda stara się uchwycić ludzki sposób improwizowania. Architektura systemu bazuje na trzech elementach, które składają się na ludzką improwizację: pamięć taktów, doświadczenie muzyka (ogranie) oraz wiedza teoretyczna muzyka.

Rdzeniem metody jest algorytm genetyczny, który generuje nowe takty na podstawie poprawnego zbioru zagrywek. Algorytm używa „świadomych muzycznie” operatorów krzyżowania oraz mutacji (głównie ozdobniki muzyczne). Nie uniknięto etapu korekcji generowanych taktów względem podkładu.

Opracowano model podobieństwa taktów – niezbędny element funkcji celu alg. genetycznego. Funkcja podobieństwa bazuje na trzech wektorach, które opisują trudne do uchwycenia muzyczne aspekty improwizacji.

Zaprojektowana metoda improwizacji oraz prototyp systemu w sposób wielce zadowalający realizują zadanie ujęte w temacie pracy. W rozdziale siódmym znaleźć można liczne kierunki oraz perspektywy rozwoju rzeczonych metody improwizacji.

This paper describes a method which improvise a solo to a given music track. The method has been also implemented as a prototype system with a graphical user interface. Designed method tries to capture the human way of improvising. The system architecture is based on three elements that imitate the human improvisation: music bars memory, the experience of musician and his music theoretical knowledge.

The core of the method is a genetic algorithm that generates new bars on the basis of a correct set of licks. The algorithm uses, „musically aware” crossover and mutation operators (mainly for musical ornaments). The simple bars–correction phase was not possible to avoid.

A model of similarity bars – an essential element of the genetic algorithm – was developed. The similarity is based on three vectors that describe the elusive aspects of musical improvisation.

Developed improvisation method fulfil main task included in the subject of this paper, and, what is more, the results are very satisfactory. In chapter seven, you can find a number of trends and development prospects of the said method of improvisation.

WSTĘP

Praca niniejsza, najogólniej rzecz biorąc, to próba ujęcia poprzez informacyjny system – dziedziny należącej do sztuk pięknych – muzyki – która, jako wytwór ludzkiego natchnienia, z definicji nie podlega jednoznacznym, ścisłym opisom i regułom. Jednakże sposób utrwalania dźwięku – wytworu ludzkiego ducha – za pomocą bardzo precyzyjnego alfabetu muzycznego, jakim jest bez wątpienia zapis nutowy, sprawił, iż człowiek od wieków ponawia próby sprzężenia dwóch na pozór niepołączalnych dziedzin – muzyki i matematyki, czyli transcendencji ludzkiego talentu i logicznego racjonalizmu jego umysłu.

1. DEFINICJA ORAZ PODSTAWOWE ZAŁOŻENIA TEMATU. WSTĘPNY OPIS PODEJŚCIA DO JEGO REALIZACJI.

1.1. Objaśnienie zakresu tematu pracy i zawartej w nim terminologii

Tematem niniejszej pracy jest opracowanie prototypu oraz budowa systemu improwizującego solo do zadanego podkładu muzycznego. Sekcja rozpocznie się od próby uściślenia kilku podstawowych elementów składających się na rzeczony system.

Przede wszystkim, przez system, rozumieć należy oprogramowanie użytkowe, które ze względu na modułową budowę – wyrażoną poprzez szereg komponentów ściśle ze sobą powiązanych – realizuje pewną funkcjonalność (w tym przypadku – improwizuje). Improwizacja muzyczna, powtarzając za *Słownikiem Języka Polskiego*, jest „spontanizmem komponowaniem utworu muzycznego”. Aby zawęzić rozważania do tematu pracy, należy wyobrazić sobie muzyka improwizującego w towarzystwie pewnego podkładu muzycznego. To swoiste tło improwizacji, utożsamiać należy z mało dynamiczną warstwą muzyczną graną najczęściej przez gitarę klasyczną (akordy) lub gitarę basową (melodia). Podkład ten ma za zadanie prowadzić główną linię melodyczną. Stanowi on pewnego rodzaju kierunek, w którym powinien zmierzać improwizator i którego powinien się trzymać. Zaznaczyć jednak należy, iż muzyk nie jest bezwzględnie ograniczony przez tę warstwę – pozostawia ona pełne pole do popisu improwizatorowi. Nie sposób nie zadać pytania: skoro improwizator powinien mieć na uwadze podkład – prowadzący główną linię melodyczną, to czy jest coś co w podobny sposób determinuje ów podkład? Teoretycznie nie, jednakże tym miejscu nieodzownym wydaje się być wyjaśnienie mechanizmów, które rządzą wieloma improwizacyjnymi występami.

Wspomniany podkład muzyczny jest *de facto* pewnym środkiem konstrukcyjnym stosowanym m. in. przez kompozytorów. Technika prowadzenia podkładu polega na przenoszeniu melodii w górę lub w dół zgodnie z odpowiednimi zasadami, które opierają się na harmonii występującej w muzyce. Gwarantuje ona przyjemną dla ucha ludzkiego wędrówkę melodii, która zazwyczaj dąży do jej naturalnego punktu kulminacyjnego zwanego toniką.

Wędrówki takie, czyli kolejne przejścia melodii, są zazwyczaj przewidywalne. Dla niektórych gatunków muzycznych istnieją powszechnie znane zestawy takich przejść. W bluesie standardem jest dwunastoelementowa struktura, występująca w wielu odmianach. Z tej wiedzy często korzystają muzycy improwizujący – znają oni bowiem strukturę podkładu i przewidują, w którym kierunku przemieści się linia prowadząca (podkład).

Muzyk, który z wprawą improwizuje, odbierany jest często jako osoba ponadprzecięt-

nie uzdolniona. Może to być prawdą, jednak warto zwrócić uwagę, iż jego maestria wynika najczęściej z dużego doświadczenia, czyli tzw. ogrania. Ponadto, stosowane przez niego struktury muzyczne, podobnie jak podkład, są na ogół elementami charakterystycznymi dla danego gatunku muzycznego – stanowią powszechnie znany zestaw tzw. zagrywek.

Choć wielu muzyków improwizuje, bazując na swoich wrodzonych, nieprzeciętnych zdolnościach, to gro z nich opiera się na swej rzetelnie zdobytej wiedzy, technice i osłuchaniu. Mowa tu o wiedzy zarówno z dziedziny teorii muzyki (przewidywanie kolejnych przejść podkładu), jak i pamięci – ogromnej liczbie taktów, zagranej wcześniej. Improwizacja taka, choć niewątpliwie wzbogacona talentem muzyka, jest w dużej mierze techniczną umiejętnością, którą może odtworzyć komputer.

Gwoli zrozumienia poniższych rozważań, należy wyjaśnić również pojęcie „solowej linii melodycznej”. Solo, w terminologii muzycznej, oznacza utwór, czy też linię melodyczną, graną lub śpiewaną przez jednego wykonawcę. Opisywany system generuje improwizacje, w której nie usłyszymy jednocześnie więcej niż jednego dźwięku (nie licząc podkładu muzycznego). Taki środek wyrazu nosi miano homofonii – jednogłosowej melodii, której towarzyszy akompaniament.

Tło, czy też podkład muzyczny, w praktyce przyjmuje wiele postaci. Może być to jedynie zbiór kierunków i odległości, zwanych interwałami, które definiują kolejne kroki akompaniamentu. Równie często spotyka się bezpośredni zapis nutowy np. dla linii basowej, która w wielu utworach pełni rolę tła muzycznego. W ramach niniejszej pracy, przyjęto, iż zadany podkład muzyczny będzie przyjmował tę pierwszą postać, w jej szczególnej formie - progresji akordów.

Reasumując, zakres tematu niniejszej pracy obejmuje zagadnienia dotyczące budowy programu komputerowego, który przy zadanym podkładzie muzycznym, generuje pojedynczą, improwizowaną linię melodyczną.

1.2. Trudności i problemy tworzenia improwizacji muzycznych

Gdy wiemy już czym zajmuje się budowany na potrzeby pracy system, warto wytłumaczyć w jakim celu to robi oraz jakie problemy, związane z improwizacją, rozwiązuje. Zaczniemy od próby uchwycenia trudności improwizacji muzycznej. Wiele osób, niekoniecznie zawodowych muzyków, potrafi wykonywać utwory muzyczne z niezwykle kunsztem i płynnością. Osoby te, mają opanowaną umiejętność czytania nut, a dzięki żmudnym ćwiczeniom, potrafią zagrać swój repertuar niemal doskonale. Czy opanowanie czytania nut, nawet *a vista*¹ oraz płynne odgrywanie wielu utworów muzycznych pociąga za sobą umiejętność improwizacji? Podobne pytanie zadać można w odniesieniu do powszechnej umiejętności – czytania. Czy osoby, które płynnie czytają oraz, co więcej, potrafią cyto-

¹ *A vista* z wł. w dosłownym tłumaczeniu znaczy „na pierwszy rzut oka”; w terminologii muzycznej oznacza pierwsze wykonanie danego utworu muzycznego przez wykonawcę czytającego z nut bez wcześniejszego zaznajomienia się z zapisem nutowym tego utworu.

wać z pamięci fragmenty znakomitych dzieł literackich, potrafiłyby napisać równie znakomity tekst, a nawiązując do podkładu muzycznego, czy zdołałyby osadzić go w danym gatunku literackim? Sytuacja taka jest prawdopodobna, jednak przeciętny czytelnik, który nawet na co dzień obcuje z literaturą piękną, pisarstwo musi opanować podobnie jak inne umiejętności. W większości przypadków jednak nie jest w stanie stworzyć dzieła równego poziomem literackim czytany przez siebie utworom. Podobnie jest z improwizacją muzyczną. Wśród osób grających na wszelkiego rodzaju instrumentach trudno znaleźć osobę posiadającą zdolność improwizacji. Pamiętając, że istnieje wiele gatunków muzycznych, należy zaznaczyć, iż wielu znanych ze zdolności improwizatorskich muzyków specjalizuje się tylko w jednym z nich. Wszystko to sprawia, iż jest to umiejętność wbrew pozorom rzadka.

Ponadto, improwizacja, jest procesem twórczym, podczas którego uzyskiwanie oryginalnych wyników nie jest rzeczą łatwą. A więc mając świadomość faktu, iż niewielu jest muzyków biegłych w sztuce improwizacji, trzeba stwierdzić, że system, który automatyzuje tę czynność, przyczyniłby się do większej dostępności tego rodzaju muzyki, a przede wszystkim do możliwości jej generowania. W tym miejscu należałoby zadać pytanie komu przyda się system, który w sposób automatyczny będzie tworzył improwizacje muzyczne.

1.3. Praktyczne wykorzystanie systemu

Przede wszystkim, system taki, może służyć do nauki improwizacji. Poprzez dobór odpowiednich parametrów aplikacji, można generować improwizowane linie melodyczne do zadanego podkładu. Mogą one posłużyć jako wzór improwizacji dla osób chcących osiągnąć tę umiejętność.

Drugim przykładem zastosowania takiego systemu jest wspomaganie procesu komponowania utworów muzycznych. Improwizowane przez system linie melodyczne mogą być inspiracją dla kompozytorów, lub, przy doborze odpowiednich parametrów, stanowić nowy, gotowy utwór muzyczny, wymagający jedynie kilku poprawek.

Podobnie jak nie każdy muzyk potrafi improwizować, tak nie każda improwizacja zasługuje na miano dobrego utworu muzycznego. Nie byłoby więc rozsądnym wymagać od maszyny, która naśladować ma ludzką kreatywność, aby generowała melodie jedynie na najwyższym poziomie. Okazuje się jednak, że motywy muzyczne, które są przeciętne i nie absorbują uwagi, również mogą być użyteczne. „Muzaki”, bo o nich mowa, to nazwa muzyki funkcjonalnej, która określana jest również mianem muzyki windowej (ang. elevator music), muzyki tła czy też muzyki marketingowej. Głównym zadaniem muzaka jest wypełnienie ciszy w miejscach publicznych (szpitalach, bankach, windach, pracy). Pierwszy zastosował go francuski kompozytor Erik Satie², który, ku zdziwieniu publiczności,

² Erik Satie (ur. 17 maja 1866 w Honfleur, zm. 1 lipca 1925 w Paryżu) – kompozytor francuski. Uznawany za prekursora wielu późniejszych awangardowych kierunków artystycznych oraz za najbardziej ekscentrycznego kompozytora czasów *Fin de siècle*.

podczas przerwy koncertu zalecił ciche odgrywanie specjalnej kompozycji. Równocześnie zachęcał publiczność do rozmów i kontynuowania swych typowych zajęć podczas antraktu. Muzaki są przeznaczone do uprzyjemniania pobytu w miejscach publicznych. Aplikacja generująca improwizacje może służyć do tworzenia rzeczonych muzaków.

Można więc, z całą stanowczością, stwierdzić, iż improwizująca aplikacja będzie szeroko użyteczna. Ponadto, należy podkreślić, iż sama próba wykorzystania komputera do generowania improwizacji, jest niezwykle interesującą koncepcją. Zmierza ona bowiem do uchwycenia psychologicznych mechanizmów ludzkiej kreatywności i odwzorowaniu ich w sztywnych strukturach programów komputerowych. Można by pokusić się o stwierdzenie, iż praca niniejsza to kolejny krok na drodze do zmiany powszechnego przekonania, iż komputer nie będzie nigdy zdolny do twórczego myślenia.

1.4. Dodatkowe założenia do tematu pracy

Zgodnie z podaną już wcześniej definicją, improwizacja jest spontanicznym tworzeniem melodii. Czy należy zatem spontaniczność tą odzwierciedlić w systemie w postaci generowania linii melodycznej w czasie rzeczywistym? Z całą stanowczością należy stwierdzić, iż nie jest to konieczne. Zgodnie ze spostrzeżeniami, działanie systemu w czasie rzeczywistym, nie jest najistotniejszą kwestią. Temat skupia się na samym procesie improwizacji, a nie na technikach, które umożliwiłyby generowanie kolejnych taktów improwizacji w czasie rzeczywistym. Ponadto, podkład muzyczny jest z góry znany. Pozwala to na korzystanie z niego, jak z pewnego rodzaju kontekstu muzycznego. Sam czas generowania improwizacji nie jest krytyczny – zależy od możliwości sprzętu, na którym aplikacja będzie uruchamiana.

1.5. Przesłanki skłaniające do realizacji zadania z pomocą maszyny

Na tym etapie rozważań należałoby odpowiedzieć na pytanie – co w ostateczności skłania do postawienia tezy, iż maszyna może sprostać zadaniu tworzenia improwizacji muzycznej. Konstatacji takiej sprzyja analiza składowych ludzkiego procesu improwizacji, takich jak: wiedza teoretyczna muzyka, jego doświadczenie (ogranie) oraz stopień przygotowania (np. znajomość podkładu, na tle którego aktualnie odbywa się improwizacja).

Wykluczwszy z improwizacji czynnik czysto ludzki jakim jest talent improwizatora, każdy z pozostałych komponentów można śmiało odzwierciedlić w powszechnie wykorzystywanych strukturach programistycznych. I tak, upraszczając tą myśl, można wyróżnić trzy podstawowe elementy improwizacji muzycznej:

- znajomość wielu fraz muzycznych (wzorców),
- umiejętności muzyka,
- wiedza teoretyczna.

Tabela 1.1: Zestawienie głównych elementów składających się na improwizację muzyczną oraz ich teoretyczne odzwierciedlenie w elementach komputerów.

Znajomość fraz muzycznych	
Człowiek	Komputer
Frazy muzyczne utrwalone w pamięci ludzkiej.	Frazy muzyczne, przy ustalonej formie zapisu, w bazie danych.
Umiejętności	
Człowiek	Maszyna
Zdolności muzyka do modyfikacji melodii, płynnego łączenia fraz muzycznych, znajomość reguł modyfikacji, łączenia fraz itp.	Wprowadzanie lokalnych zmian we frazach (np. ozdobników) zapisana jako sekwencyjny zestaw czynności czy też reguła. Automatyzacja w/w procesów.
Wiedza teoretyczna	
Człowiek	Maszyna
Wiedza o tym jak jest zbudowany i do kąd zmierza podkład, jak z nim współbrzmieć aby uniknąć niezgodnych dla ludzkiego ucha brzmień.	Znajomość podkładu muzycznego, odpowiadająca przewidywaniu kolejnych przejść podkładu. Wiedza teoretyczna sprowadzona do zautomatyzowanych procesów.

Zestawienie tych składowych z odpowiadającymi im abstrakcyjnymi pojęciami z dziedziny informatyki, które mogą służyć za ich elektroniczny substytut, zawiera tabela 1.1. Oto szczegółowy opis pojęć w tejże tabeli:

Znajomość fraz muzycznych

Mowa tu o pamięci muzyka, który podczas improwizacji może bazować na fragmentach pochodzących ze znanych mu utworów. Zbiór takich zagrywek, po ustaleniu formatu zapisu, może być z powodzeniem przechowywany przez komputer – zarówno w przeznaczonych do tego celu bazach danych, jak i w pospolitych plikach tekstowych, o nieskomplikowanej strukturze.

Umiejętności muzyka

Przez umiejętności muzyka należy rozumieć tu biegłość w dodawaniu do melodii ozdobników, łączeniu taktów czy też w innych zabiegach mających na celu zbudowanie poprawnej i efektownej improwizacji. Dzięki temu, że większość ozdobników muzycznych ma sprecyzowaną budowę, można odwzorować je w strukturach programistycznych jako zapis poszczególnych przekształceń muzycznych. Przekształcenia te, mogą bazować na prostych regułach, które w łatwy sposób można utrwalić chociażby w programistycznych konstrukcjach sterujących.

Wiedza teoretyczna muzyka

Wiedza teoretyczna muzyka pozwala mu m. in. przewidzieć wędrówkę podkładu muzycznego, pomaga unikać nut, które wraz z podkładem brzmiałyby nieakceptowalnie dla ucha ludzkiego. Część tych informacji jest znana systemowi *a priori* – podkład muzyczny jest zmienną wejściową aplikacji. Większość zasad improwizacji, podobnie jak w przypadku ozdobników melodycznych, można odzwierciedlić w zbiorze reguł, które z kolei można przenieść i utrwalić w pamięci komputera.

Opisane powyżej trzy elementy improwizacji, stanowią rdzeń systemu improwizacyjnego, który bazując na zbiorze poprawnych zagrywek (bazie danych) potrafi je łączyć, przekształcać, modyfikować (zbiór operacji) dbając o ich poprawność i właściwe współbrzmienie względem podkładu muzycznego (reguły).

1.6. Cechy poprawnej improwizacji

Jak podkreślano już wcześniej głównym celem improwizującego muzyka jest tworzenie melodii, którą jest w stanie zaakceptować ludzkie ucho. Współbrzmienia przykre dla ucha ludzkiego to dysonanse. Lista odległości (interwałów) pomiędzy dwoma dźwiękami, które tworzą dysonanse jest znana. Ułatwia to muzykowi ich unikanie, a fakt, iż lista tych interwałów jest skończona i ściśle określona, pozwala odzwierciedlić je w postaci reguł z których korzystać może komputerowy system improwizujący. Reasumując – poprawna improwizacja nie zawiera niezamierzonych dysonansów.

1.7. Subiektywizm w ocenie utworu muzycznego

Zmieniający się przez wieki stosunek twórców i odbiorców muzyki do dysonansów, może zilustrować zagadnienia w tej części pracy. Dla porządku należy rozpocząć od wprowadzenia pojęcia konsonansu – przeciwieństwa dysonansu – współbrzmienia zgodnego dla ucha ludzkiego. Pojęcie konsonansu, szczególnie w muzyce europejskiej, zmieniało się na przestrzeni wieków. Normy muzyki starożytnej do konsonansów zaliczały jedynie kilka interwałów, a większość dysonansów nakazywały unikać. Kanony te zmieniały się z upływem lat, stopniowo uszczuplając listę dysonansów. Muzyka współczesna, w odróżnieniu od utworów klasycznych, dopuszcza, a nawet zakłada istnienie dysonansów. Niegodne brzmienia są wręcz elementami rozpoznawalnymi niektórych gatunków muzycznych. Mowa tu chociażby o tzw. *blue notes*, które wprowadzają do bluesa charakterystyczną nastrojowość. W muzyce harmonia przestaje być priorytetem, czego dowodem są nowo powstające gatunki muzyczne (np. dodekafonia).

Wielu melomanów uznaje stosowanie dysonansów za niedopuszczalne. Równie wielu podziwia kunszt stosowania tego typu konstrukcji muzycznych. Prowadzi nas to do spostrzeżenia, że odbiór utworu muzycznego jest sprawą indywidualną. Ma to niebagatelne

znaczenie w kontekście tworzenia systemu improwizującego. I tak, w wielkim uproszczeniu, po podaniu pewnych parametrów improwizacji, aplikacja generuje zbiór melodii ułożonych w improwizację. Procesy, które mają na celu wygenerowanie tych melodii, muszą kierować się pewną miarą, aby dostarczyć użytkownikowi fraz, które są poprawne i zostaną przez niego odebrane jako przyjemne dla ucha brzmienia. Jednak, jak powiedziano wcześniej, ocena taka jest zawsze subiektywna. Najlogicznieszym wydaje się być zastosowanie reguł tworzenia fraz i przekształceń muzycznych, które gwarantują powstanie melodii poprawnych i akceptowalnych przez większość słuchaczy. Czy takie podejście do problemu pozostawia miejsce dla kreatywności? Niestety jest ona zastąpiona sztywnymi i przewidywalnymi regułami. Zbudowany system stara się uchwycić ludzką kreatywność ale, mając na uwadze istotę ludzkiej pomysłowości oraz przypisany naturze człowieka subiektywizm w ocenach estetycznych, nie jest to rzeczą łatwą.

W niniejszym rozdziale sprecyzowałem temat pracy. Wyjaśniłem również potrzebę budowania systemów improwizujących oraz nakreśliłem głównie problemy, które wynikają z próby uchwycenia przez maszynę tego typu ludzkiej kreatywności. Opisałem także grupę docelową odbiorców stworzonej aplikacji oraz zobrazowałem problem subiektywności odbioru generowanych przez system melodii. Na tle wspomnianych problemów i ograniczeń, opisałem, w sposób ogólny, strukturę systemów odzwierciedlającą proces improwizacji muzycznej.

Część I

Wstępny projekt metody improwizacji

2. HISTORIA I PRZEGLĄD ROZWIĄZAŃ Z DZIEDZINY KOMPOZYCJI WSPOMAGANEJ KOMPUTEROWO

Myślą przewodnią rozdziału jest przegląd dotychczasowych rozwiązań dotyczących wspomaganego komputerowo komponowania muzyki. W jego treści nie zabraknie jednak ciekawostek z szerokiego obszaru muzyki generowanej komputerowo (ang. *computer generated music*). W kilku miejscach dokonano podziału podejść stosowanych w tej dziedzinie. Wskazanie spośród nich tych, które zastosowano przy budowie systemu przeniesiono do rozdziału opisującego zaprojektowaną metodę improwizacji. Tam też znaleźć można uzasadnienia wyborów i ich dokładniejszy opis.

W powszechnym mniemaniu, muzyka, jako dziedzina sztuki – wytwór ludzkiego talentu, nie może mieć powiązań z naukami ścisłymi. Okazuję się jednak, że nie do końca jest to prawdą. Analogię pomiędzy muzyką i matematyką zauważono już w starożytności. Pitagoras (ok. 500 p.n.e.) dokonuje podziału muzyki ze względu na jej harmonię, wskazując przy tym jej pokrewieństwo z matematyką. Ada Byron Lovelace [11], znana z opisanie mechanicznego komputera Charlesa Babbage’a¹, w 1843 roku przewidziała użycie maszyn w komponowaniu muzyki. Z biegiem lat przekonano się, iż wiele aspektów muzycznych można uchwycić w modelach matematycznych.

Zapis nutowy jest w znacznej mierze niezwykle sformalizowany. Obok rozbudowanej struktury, zapis ten ma jeszcze bogatszą semantykę, ujętą często w nieformalnych elementach zapisu nutowego (np. komentarze kompozytora). Wyłączywszy z rozważań te nieusystematyzowane elementy, znaleźć można języki notacji muzycznej (ABCNotation[1], GUIDO [7]), które z powodzeniem oddają nawet skomplikowane konstrukcje muzyczne. Znanym rozwiązaniem jest MIDI (**M**usical **I**nstrument **D**igital **I**nterface), który jako protokół służy do wymiany prostych poleceń pomiędzy instrumentami elektronicznymi. Protokół ten opiera się na przekazywaniu informacji o rozpoczęciu odgrywania nuty (ang. *note on*) oraz jej zakończeniu (ang. *note off*). Informacje te opisują linie melodyczne na 16 kanałach, z rozdzielczością 24 zdarzeń na ćwierćnutę. Choć MIDI jest złożonym protokołem, a rozdzielczość sygnałów zwiększa jego precyzję, nie jest rozwiązaniem idealnym do zapisu czy analizy dzieł muzycznych. Największą wadą protokołu jest fakt, że w danej chwili informacja o dźwięku nie może być rozpatrywana bez uwzględnienia komunika-

¹ Charles Babbage (ur. 26 grudnia 1791 w Teignmouth, zm. 18 października 1871 w Londynie) – angielski matematyk, astronom i mechanik, autor tablic logarytmicznych, konstruktor mechanicznych maszyn liczących.

tów poprzednich. Dany komunikat nie istnieje sam w sobie – musi być interpretowany z uwzględnieniem przeszłych informacji.

Wraz z rozwojem technologii komputerowej znacznie wzrosło zainteresowanie zastosowaniem maszyn do generowania czy też syntezy muzyki. Baggi [3] wyróżnia dwa główne nurty tych zainteresowań:

- komponowanie muzyki oraz analiza dzieł muzycznych,
- synteza dźwięku.

Temat niniejszej pracy zawiera się w pierwszym nurcie, a dokładniej, w jego pierwszej części. Dlatego też, w dalszej części rozdziału, omówione zostaną rozwiązania związanych z tym właśnie obszarem, zwanym *Computer Generated Music*.

W tym miejscu warto wyjaśnić co rozumieją, przez „komponowanie”, autorzy cytowanych prac. Powtarzając za Cope² definiują je jako takie ułożenie dźwięków, zgodne z zamysłem kompozytora, które w ogólnym ujęciu można uznać za przeciwieństwo chaosu – szumu. W odniesieniu zaś do maszyn, przez kompozycję rozumieją sekwencję lub zbiór reguł, instrukcji, operacji do rozwiązania pewnego problemu, zadania w skończonej liczbie kroków poprzez kombinację części muzycznych w całą kompozycję. Mając na uwadze tą definicję można przejść do omówienia najciekawszych rozwiązań z dziedziny komputerowego komponowania muzyki.

Pierwszą, niezwykle oryginalną grupę rozwiązań, stanowią kompozycje interdyscyplinarne, które starają się przypisać wielkości niemuzyczne do zmiennych opisujących dzieło muzyczne. Jako przykład można wymienić niezwykle ciekawy pomysł zastosowania wielkości opisujących ruch planet do opisu rytmu i melodii utworów muzycznych[9]. Równie intrygującą pracę opublikowała Alexjander [2], która współczynnik absorpcji światła przez molekuly DNA powiązała z wysokościami dźwięków generowanych kompozycji.

Nie sposób pominąć grupy rozwiązań probabilistycznych, generujących melodie nuta po nucie (ang. *note by note*). Wiodą tu prym systemy oparte na modelach Markova³, w których korzysta się z macierzy tranzycji aby wyznaczyć prawdopodobieństwo wystąpienia kolejnej nuty. Wadami takich podejść jest konieczność posiadania odpowiednio liczego zbioru uczącego. Gdy jest zbyt mały - generowane są melodie zbliżone do losowych. Zbyt liczny zbiór natomiast – skutkuje powtarzaniem fraz ze zbioru uczącego [18].

Liczną grupę rozwiązań stanowią metody ewolucyjne. Niewątpliwą zaletą zastosowania algorytmów genetycznych (AG) w procesie kompozycji jest otrzymanie nie jednego wyniku lecz całego ich zestawu. Główną różnicą rozwiązań korzystających z algorytmów ewolucyjnych jest sposób oceny kolejnych pokoleń, co jest kluczowym elementem w tego typu algorytmach. I tak, wyróżnić można m.in. ewaluacje deterministyczne, które uży-

² David Cope (ur. w San Francisco, Stany Zjednoczone, 1941 r.) amerykański naukowiec, publicysta i emerytowany profesor muzyki na Uniwersytecie Kalifornijskim w Santa Cruz.

³ Proces Markowa – ciąg zdarzeń, w którym prawdopodobieństwo każdego zdarzenia zależy jedynie od wyniku poprzedniego.

wają funkcji matematycznych do oceny poszczególnych osobników, ewaluacje oparte na opiniach użytkownika oraz ewaluacje bazujące na sieciach neuronowych. Dla przykładu – Orusava [12] opiera działanie AG o decyzje użytkownika, który w sposób subiektywny decydował o tym, które z taktów trafią do kolejnej iteracji. Sieci neuronowe natomiast, mogą służyć do zautomatyzowania procesu oceny, wskazując osobników o odpowiednich, zapamiętanych w strukturach neuronowych, cechach np. w danym stylu muzycznym.

Nie można pominąć tu systemu GENJAM (ang. **Genetic Jammer**) Johna Biles’a [4]. Komputer z tą aplikacją, bazując na algorytmach ewolucyjnych, jest w stanie w czasie rzeczywistym improwizować na temat granych przez użytkownika fraz muzycznych. Tego typu dialogi muzyczne po dodaniu aranżacji⁴ dają w wyniku bardzo efektowne wyniki. Biles podkreśla, iż korzysta ze, specjalnie zaprojektowanych do pracy z muzyką, operacji krzyżowania i mutacji. Razem z algorytmami poprawy taktów względem podkładu daje to niezwykle efekt zbliżony do dialogu dwóch improwizujących muzyków (z ang. trading fours). System GenJam był główną inspiracją podczas tworzenia systemu improwizującego. Na stronie autora⁵ znaleźć można wiele nagrań, w tym wideo, prezentujące działanie systemu.

Wśród rozwiązań korzystających z sieci neuronowych znaleźć można frapujące doświadczenie Thaler’a [19]. Przeprowadził on eksperyment „śmierci klinicznej” sieci neuronowych. Wyłączył on niektóre neurony w sieci, skutkiem czego nie była, jak przewidywało wielu naukowców, dezintegracja sieci, lecz jej adaptacja. Thaler nazwał to zjawisko kreatywnym myśleniem. Pobudzając uszkodzoną sieć neuronową, jej wyjście interpretował m.in. jako melodie.

Autor niniejszej pracy spróbował odtworzyć ten eksperyment dla pamięci autoasocjacyjnych Hopfielda. Wyniki przy uszkodzaniu sieci tego typu przyniosły zadowalające rezultaty. Wadą był brak kontroli tego procesu i wykrycie bezpośredniej zależności pomiędzy uszkodzeniami neuronów, a wynikiem uzyskanym na wyjściu zmodyfikowanej sieci.

Innym istotnym zagadnieniem związanym z tematem jest „analiza utworów”. Nie jest ona główną kwestią niniejszej pracy, jednak warto wspomnieć o jednym z wypracowanych w jej ramach podejść, zwanym lingwistycznym. Cytowana niezwykle często praca Winograda [23] opisuje lingwistyczną analizę harmonii tonalnej utworu. Autor zauważa podobieństwa pomiędzy analizą składni tekstu, a teorią harmonii muzycznej. W uproszczeniu, buduje on drzewa rozkładu bazując m.in. na pewnych następstwach wynikających z harmonii. Tak jak w zdaniach wymaga się aby czasownik był silnie związany z podmiotem, tak niektóre elementy muzyczne (np. akordy) wymagają podobnego współwystępowania. Podejście takie dobrze sprawdza się podczas analiz utworów muzycznych.

⁴ W tym kontekście aranżacją jest nieznaczne rozwinięcie utworu poprzez dodanie linii perkusyjnej oraz dodatkowych brzmień; nadanie mu ostatecznego kształtu.

⁵ Adres strony: igm.rit.edu/~jabics/GenJam.html

Powyżej opisano główne nurty związane z komputerowym komponowaniem muzyki. Scharakteryzowano najciekawszych reprezentantów tych nurtów. Istotnym dopełnieniem każdego opisu powinna być ocena wyników. Niestety, niewiele artykułów zawiera odnośniki chociażby do wybranych aranżacji wygenerowanych utworów. Wyniki do których udało się dotrzeć podzielić można na trzy kategorie. Do pierwszej zaliczyć można melodie, w których nie można odnaleźć elementów wysokopoziomowej kompozycji. Nie są to jednak utwory losowe. Pokazują one, że dane rozwiązanie działa, jednak sam wynik pozostawia wiele do życzenia. Druga grupa to melodie poprawne, a co więcej, widać w nich pewien głębszy zamysł kompozytorski. Niestety wynika to zazwyczaj z ich aranżacji. Jest tak chociażby w przypadku "śpiewających" neuronów Thaler'a, który udostępnia przykładowe utwory o szerokiej aranżacji, jedynie bazując na fragmentach melodii dostarczonej przez sieć neuronową. Najciekawsza jest trzecia grupa przykładów. Są to kompletne utwory; odnajdujemy w nich poprawne melodie na poziomie taktów, pomysł na kompozycję całego utworu oraz elementy aranżacji. Do wyników takich zaliczyć należy utwory nagrane podczas sesji wspomnianego systemu GENJAM, który należy potraktować jako pewnego rodzaju odnośnik, szczególnie, że w pracy niniejszej korzystano również z algorytmów ewolucyjnych.

3. SEKRETY POLIHYMNII CZYLI PODSTAWOWA TERMINOLOGIA MUZYCZNA

Dwa pierwsze rozdziały pracy napisano z myślą o osobach, które nie miały styczności z muzyką od jej teoretycznej strony. Pozostała część pracy wymaga jednak znajomości chociażby podstawowych zagadnień teoretycznych związanych z muzyką. W rozdziale niniejszym postaram się w przystępny sposób przybliżyć czytelnikowi najistotniejsze z tych kwestii.

Nazwa rozdziału nawiązuje do cieszącej się niesłabnącą popularnością książki „Sekrety Polihymnii”¹ [21] Jerzego Waldorfa². To kompendium, obok [17], posłuży jako główne źródło wiedzy w tym rozdziale.

Muzyka, jako pojęcie, odnosi się do ogółu wszelkich kompozycji dźwiękowych, które oddziałują na psychikę człowieka. Podobnie jak było z ludzką mową – muzykę zaczęto odwzorowywać w piśmie. To prowadzi nas do pierwszej definicji – nuty.

3.1. Zapis nutowy

Podstawowym muzycznym elementem piśmiennym jest nuta, która odwzorowuje konkretny dźwięk, a dokładniej - jego wysokość czyli częstotliwość drgań, która go wywołuje. Współcześnie nuta przekazuje nam informacje o wysokości dźwięku oraz czasie jego trwania (w kontekście tempa utworu). Aby uporządkować zapis kolejnych nut korzysta się z pięciolinii. Położenie główki nuty na pięciolinii określa jej wysokość. Wypełnienie główki wraz z typem ogonka opisują czas trwania nuty (względem rytmicznego kontekstu utworu: metrum i tempa). Przeciwnieństwem zagrania nuty jest wstrzymanie się przed odgrywaniem jakiegokolwiek z nich. Przerwa taka nosi miano pauzy i podobnie jak nuta, dzielona jest ze względu na czas jej trwania. Rysunek 3.1 a) przedstawia podział nut i pauz ze względu na czas trwania (odnośnie całej nuty). Rysunek 3.1 b) natomiast pokazuje jak zapis nut na pięciolinii ma się do odpowiednich wysokości na klawiaturze pianina. Oprócz graficznej reprezentacji, nuty na odpowiednich wysokościach posiadają swoje nazwy. Z racji okresowego powtarzania się układu siedmiu bazowych klawiszy na klawiaturze muzycznej (tzw. oktawy), nazwano siedem podstawowych dźwięków, a ich odpowiedniki położone niżej lub wyżej tej siódemki oznacza się poprzez indeks dolny lub

¹ Polihymnia – w mitologii greckiej muza poezji sakralnej i hymnicznej oraz pantomimy.

² Jerzy Waldorff-Preyssh. Nabram (ur. 4 maja 1910 w Warszawie, zm. 29 grudnia 1999 tamże) – polski pisarz, publicysta, krytyk muzyczny i działacz społeczny.

górnym. Przykładowe nazewnictwo, wraz ze zwyczajowymi nazwami oktafów, przedstawia rysunek 3.1 a). Prosty przykład zapisu nutowego przedstawia 3.1 c).

3.2. Metrum

Nuty grupowane są, podobnie jak słowa, w muzyczne „zдания”. Schematem, który określa długość takich fraz (zdań) jest metrum. Zapisywane jest ono jako dwie cyfry, jedna nad drugą. Cyfra dolna oznacza podstawową jednostkę metryczną, cyfra górna natomiast, licznosc tych jednostek. W obrębie pracy posługiwać się będziemy jedynie metrum 4/4 oznaczającym iż fraza trwać będzie cztery ćwierćnuty. Przykład zapisu metrum przedstawiono na rysunku 3.2. Należy zwrócić uwagę na to, iż długości trwania wszystkich nut w obrębie taktu sumują się, zgodnie z metrum, do czterech ćwierćnut (patrz: 3.1a).

3.3. Takt

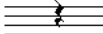
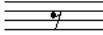
Opisane w poprzednim podpunkcie frazy (zдания) muzyczne zwane są taktami. Takty oddzielane są od siebie pionowymi kreskami. Każdy utwór muzyczny dzieli się na takty powtarzające się bez przerwy. Przykładowy zapis nutowy z podziałem na takty prezentuje rysunek 3.2.

3.4. Tempo

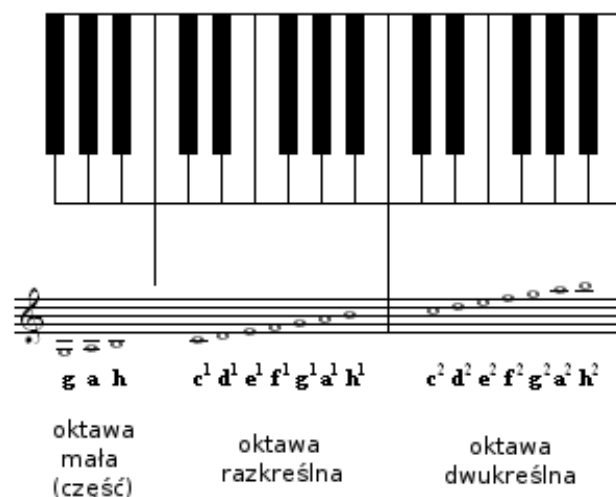
Tempo mówi o tym jak szybko utwór powinien być wykonywany. Mówiąc *ćwierćnuta* odnosimy się do czasu trwania całej nuty. Wiemy jednak tylko, że będzie ona trwała dwa razy dłużej niż półnuta, cztery razy dłużej niż cała nuta itd. Tempo określa, pośrednio, ile powinna trwać przykładowa ćwierćnuta. Dzięki temu wiemy jak szybko grać poszczególne nuty.

3.5. Znaki chromatyczne

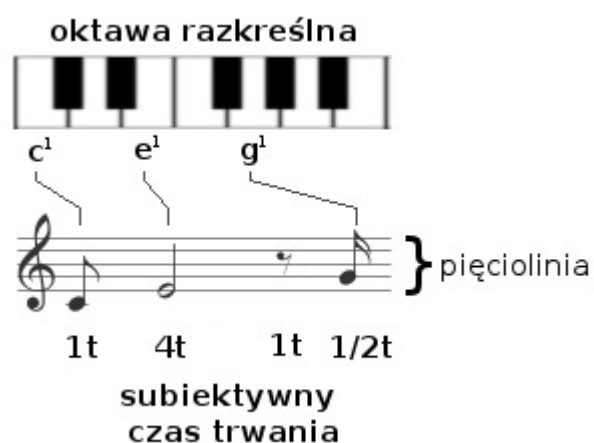
Zaprezentowane w jednej poprzednich sekcji nazwy nut (*c, d, e, f* itd.) odnoszą się do kolejnych białych klawiszy klawiatury muzycznej. Zapis nutowy pozwala „dostać się” do pozostałych – „czarnych” dźwięków poprzez znaki chromatyczne. Znaki te postawione przed nutą oznaczają jej podwyższenie (poprzez *krzyżyk*) lub obniżenie (poprzez *bemol*) o pół tonu czyli na najbliższy klawisz - w tym czarny. Raz użyty znak chromatyczny odnosi się do wszystkich nut o tej samej wysokości w obrębie taktu, albo do zastosowania specjalnego znaku chromatycznego – kasownika. Aby odróżnić nazwy dźwięków od ich zmodyfikowanych odpowiedników, dodaje się postfiks: *-is* dla dźwięków podwyższonych, *-es* dla obniżonych. Rysunek 3.3 prezentuje zastosowanie znaków chromatycznych.

Nazwa	Graficzna reprezentacja	Pauza
	referencyjny czas trwania całej nuty	
cała nuta		
półnuta	 	
ćwierćnuta	   	
ósemka	       	
szesnastka	               	

(a) Podział nut i pauz ze względu na czas trwania



(b) Odwzorowanie pozycji nut na pięciolinii na klawiaturę muzyczną

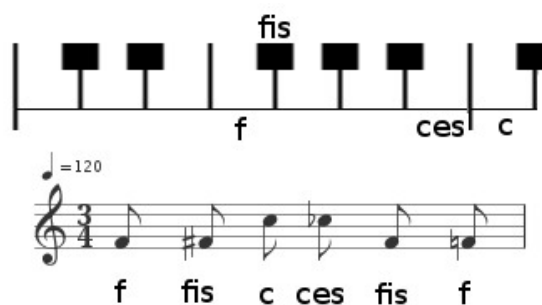


(c) Przykład zapisu nutowego

Rys. 3.1: Prezentacja podstawowych aspektów graficznego zapisu nut



Rys. 3.2: Metrum i jego wpływ na budowę taktów



Rys. 3.3: Przykład działania znaków chromatycznych. Dźwięk *f* został podwyższony o półton do dźwięku *fis*. Dźwięk *c* został natomiast obniżony o jeden półton do *ces* z użyciem bemola. Działanie krzyżyka zostało anulowane dopiero w ostatniej nucie poprzez znak kasownika.

3.6. Ton, półton

Zapis nutowy umożliwia przedstawienie wielu nut. Ich odmienność wynika z różnych wysokości dźwięków (ich częstotliwości), które reprezentują. Na klawiaturze muzycznej, jak i w muzyce europejskiej, najmniejszą odległością pomiędzy dwiema nutami jest tzw. *półton*. Odległość zwana *całym tonem* składa się z dwóch półtonów. Odległość półtonu to odległość pomiędzy dwoma dźwiękami, których częstotliwości pozostają w stosunku:

$$\sqrt[12]{2} \approx 1,0594.$$

3.7. Interwały

Gdy udaje nam się rozpoznać melodię, nie wynika to zazwyczaj z bezpośredniego rozpoznania przez nas wysokości (częstotliwości) każdego dźwięku. Melodię definiują tak na prawdę odległości pomiędzy kolejnymi dźwiękami (i oczywiście ich rytm). Można tę sytuację porównać do rozpoznawania kształtów. Nie liczy się dla nas położenie danej figury (np. względem układu współrzędnych). Do rozpoznania niezbędne są poszczególne krawędzie kształtu i kąty pomiędzy nimi.

Jak już wspomniano, najmniejszą odległością w muzyce europejskiej jest półton. Względem ich liczby definiuje się odległości pomiędzy dźwiękami zwane interwałami. W tabeli

Tabela 3.1: Zestawienie podstawowych interwałów

Nazwa interwału	Liczba półtonów	Dysonans / Konsonans
pryma czysta	0	konsonans
sekunda mała	1	dysonans
sekunda wielka	2	dysonans
tercja mała	3	konsonans
tercja wielka	4	konsonans
kwarta czysta	5	konsonans
tryton	6	dysonans
kwinta czysta	7	konsonans
seksta mała	8	konsonans
seksta wielka	9	konsonans
septyma mała	10	dysonans
septyma wielka	11	dysonans
oktawa czysta	12	konsonans

3.1 znaleźć można zestawienie podstawowych interwałów. W tabeli zaznaczono również, które z interwałów uważane są za dysonanse³

3.8. Skala, gama

Skalą nazywamy układ dźwięków zestawionych według ściśle określonego porządku. Na różnych skalach, jak na różnych bazach, rozwijały się systemy muzyczne poszczególnych epok. Nowożytna muzyka europejska opiera się na dwóch skalach – „dur” i „moll”. Skalę „dur” zbudować można rozpoczynając od dowolnego dźwięku, a następnie poruszając się „w górę” klawiatury kolejno o odległości sekundy: wielkiej, wielkiej, małej, wielkiej, wielkiej, wielkiej oraz małej. Rozpoczynając od dźwięku *c* razkreślnego, poruszać się będziemy po kolejnych białych klawiszach, aż trafimy na *c* dwukreślny i otrzymamy gamę *c*-dur. Gama jest to przeniesienie schematu skali na klawiaturę przy wskazaniu dźwięku początkowego - toniki. Skala „moll” zbudowana jest z również z interwałów sekund. Są to kolejno sekunda wielka, mała, wielka, wielka, mała, wielka, wielka (białymi klawiszami od *a* razkreślnego do *a* dwukreślnego). *Dur* i *moll* to dwie podstawowe skale ośmiodźwiękowe. Istnieje wiele ich modyfikacji. Więcej informacji o użytych przy budowie systemu skalach znaleźć można w odpowiedniej sekcji rozdziału związanego z implementacją.

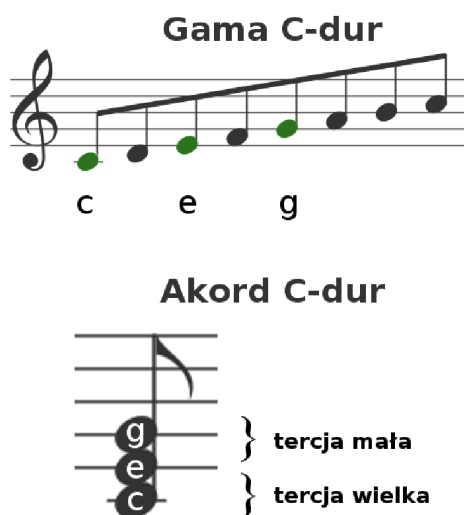
3.9. Tonacja

Tonacją nazywa się przyjętą, w ramach utworu, konkretną gamę durową lub molową. Dźwięki tej gamy są podstawowymi w utworze i definiują jego charakter.

³ Mowa tu o równoczesnym zabraniu dwóch dźwięków. Zabranie kolejno dwóch nut odległych od siebie o dysonansową odległość zazwyczaj nie jest uznawane za dysonans.

3.10. Akordy

Akord to kilka dźwięków w ściśle określonych interwałach brzmiących równocześnie. Podstawowym podziałem akordów jest ich rozróżnienie ze względu na liczbę dźwięków (trójdźwięk - 3 dźwięki, septymowy - 4 dźwięki itd.). Różnorodność akordów wynika bezpośrednio z mnogości skal muzycznych. Akord skonstruować można na schemacie skali, z odpowiedniej liczby dźwięków, biorąc ze skali co drugi element (z pewnymi wyjątkami – np. odnośnie czwartego dźwięku). I tak, dla skali *dur*, zbudować można gamę *c-dur* złożoną z dźwięków: *c, d, e, f, g, a, h*. Akord *c-dur* powstaje poprzez złożenie dźwięków *c*, *e* oraz *g*. Każdy akord określa się również, i tak dzieje się najczęściej, poprzez kolejne interwały występujące pomiędzy jego dźwiękami. Akord *c-dur* to dźwięk podstawowy (pryma akordu), odległy od niego o tercję wielką dźwięk środkowy i ostatni z trzech, odległy od środkowego o tercję małą. Przykład budowy akordu *c-dur* oraz jego powiązanie z gamą *dur*ową obrazuje rysunek 3.4. Akordów, podobnie jak skal muzycznych jest wiele. Używane są one najczęściej do akompaniamentu i posiadają specjalne oznaczenie. Nazwa akordu składa się z dźwięku podstawowego (pierwszego) oraz przyrostka. Zestawienie najważniejszych akordów znaleźć można w tabeli 3.2. Akordy w ostatniej kolumnie zostały zbudowane na dźwięku c^1 . Lista interwałów odzwierciedla piętrową budowę akordów – pierwszy interwał od c^1 jest interwałem w ostatniej linijce. Akord *C11* nie jest podstawowym akordem – został przedstawiony jako interesujący przykład wielodźwięku.

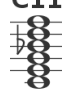


Rys. 3.4: Budowa akordu *c-dur* oraz jego powiązanie z gamą *c-dur*

3.11. Triada harmoniczna

Triadą harmoniczną nazywa się trzy podstawowe trójdźwięki (akordy) zbudowane na pierwszym, czwartym i piątym stopniu gamy. Akordy te to odpowiednio: tonika (T), sub-

Tabela 3.2: Zestawienie podstawowych akordów.

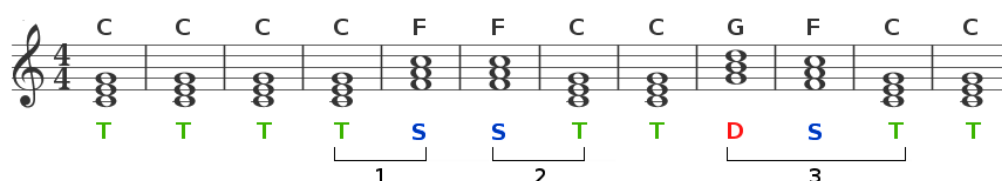
Akord	Przyrostek	Liczba dźwięków	Składowe interwały	Przykład
Trójdźwięki				
dur	- - -	3	tercja mała, tercja wielka	C 
mol	m	3	tercja wielka, tercja mała	Cm 
zwiększony	+	3	tercja wielka, tercja wielka	C+ 
zmniejszony	o	3	tercja mała, tercja mała	Co 
Wielodźwięki				
Akord septymowy durowy z septymą wielką	7	4	tercja mała, tercja mała, tercja wielka	C7 
Akord durowy undecymowy z undecymą czystą (17 półtonów)	11	6	tercja mała, tercja wielka, tercja mała, tercja mała, tercja wielka	C11 

dominanta (S) i dominanta (D). Trójdźwięki te są podstawą budowania harmonii w utworach poprzez *harmonizowanie* – odpowiednie łączenie akordów. Pomiedzy tymi trzema akordami występuje specyficzna więź. Tonika, zwana ośrodkiem ciężenia tonalnego, jest punktem, z którego „wychodzi” oraz do którego dąży melodia. Z toniką łączy się zazwyczaj subdominanta, oddalając nieco melodię od jej centrum. Dominanta z kolei, jest akordem, który kończy pewien etap oddalania się melodii od toniki. Melodia, zatrzymawszy się na dominancie, dąży bardzo wyraźnie do rozwiązania się na tonikę, zazwyczaj wracając do niej poprzez subdominantę.

Tonika, dominanta i subdominanta to również nazwy I, IV i V stopnia danej gamy. Mówiąc gama *c-dur* mamy na myśli gamę durową z toniką (pierwszym dźwiękiem) *c*¹.

3.12. Progresja akordów

Progresję akordów należy roznieć jako szereg akordów, opisanych ich rodzajem oraz stopniem gamy na którym powinny zostać zbudowane. Choć progresja może być dowolna, szeregi akordów budowane są zgodnie z zasadami harmonii, a w szczególności – w oparciu o triadę harmoniczną. Przypomnieć należy, iż progresja taka będzie stanowiła podkład muzyczny, który, co zostało zaznaczone, ma wpływ na prowadzenie głównej melodii. Jeśli zatem zadamy aby progresja zbudowana została z godnie z zasadami harmonizowania, przeniesie się to na główną linię melodyczną. Jedną z bazowych, dwunastotaktowych progresji bluesowych w skali *c-dur* prezentuje rysunek 3.5. Opisano na nim również wędrówkę melodii poprzez tonikę, subdominantę i dominantę. Przejście nr 1 prezentuje odejście od toniki (T) do subdominanty (S). Zaznaczona numerem 2 zmiana akordu sprowadza melodię do jej ośrodka - toniki. Ciąg akordów nr 3 pokazuje nagłe odejście melodii do dominanty (D) i jej naturalny powrót do toniki poprzez subdominantę. Powrót do toniki może w sposób naturalny zakończyć utwór.



Rys. 3.5: Przykład powszechnie stosowanej bluesowej progresji akordów wprowadzonej przez muzyka Williama Christophera Handy’ego, zbudowanej dla gamy *c-dur*.

3.13. Triola

Triola jest strukturą muzyczną, opisującą pewne zachwianie rytmiczne w obrębie zazwyczaj regularnego rytmu utworu. Triola to trzy nuty, występujące w równych odstępach czasu, w obrębie dwudzielnej wartości rytmicznej. I tak, triola ósemkowa oznacza wystąpienie trzech nut w czasie dwóch ósemek; triola ćwierćnutowa – trzech nut w obrębie

czasu trwania dwóch ćwierćnut itd. Triola oznaczana jest cyfrą 3 umieszczoną nad grupą nut (patrz rysunek 3.6).



Rys. 3.6: Przykład triol – ćwierćnutowej oraz ósemkowej.

3.14. Synkopa

Synkopa oznacza zabieg przesunięcia akcentu muzycznego poprzez wydłużenie jednej z dwuelementowej struktury nut. Stosunek długości nut po tym zabiegu to 3:1. Dłuższa nuta jest słaba metrycznie – nieakcentowana, krótsza, znajdująca się w mocnej części struktury, akcentowana. Rodzaj synkopa zależy od składowych długości nut (dwie ósemki, dwie ćwierćnoty itd.) oraz od tego, która nuta występuje pierwsza – dłuższa czy krótsza. Przykłady synkop zawiera rysunek 3.7, na którym akcentowane nuty oznaczone są kolorem zielonym.



Rys. 3.7: Przykład synkop – bazującej na ósemkach oraz na ćwierćnutach.

3.15. Blues

Praktycznie każdy aspekt zbudowanej improwizującej aplikacji jest rozszerzalny. Można zdefiniować dowolną skalę, własne rodzaje akordów, progresję itd. W ramach testów oraz jako temat przewodni wybrano blues jako gatunek, który będzie przedmiotem improwizacji. Rzutuje to na używane w przykładach progresje akordów, wzorcowe takty, skale itp.

Blues jest gatunkiem muzyki, który wywodzi się z muzyki afroamerykańskiej. Rdzenie, jako pieśń, blues miał ściśle określoną budowę, zarówno pod względem harmonicznym jak i tekstowym. Z czasem narodziła się, znana współcześnie, żywiołowa odmiana bluesa – *rhythm and blues*. Została ona przejęta przez przemysł rozrywkowy dając początek muzyce *rock-and-roll*owej czy też *rockowej*.

Wybór bluesa jako „platformy” testowej systemu improwizacyjnego nie ma wpływu na jego budowę ani możliwości – każdy z elementów aplikacji, który został sprametryzowany względem bluesa, może zostać zastąpiony analogicznym, nie związanym z tym gatunkiem muzycznym.

Niektóre z pojęć, wraz z przykładami, zostaną wyjaśnione dokładniej w dalszej części pracy, opisującej w sposób szczegółowy zaprojektowaną metodę improwizacji oraz implementację jej poszczególnych elementów.

Część II

Szczegółowy opis zaprojektowanej metody improvizacji

4. ARCHITEKTURA SYSTEMU

Przed przystąpieniem do prezentacji architektury należy przypomnieć jaki cel przyświecał jej projektowaniu. Przede wszystkim chciano uchwycić złożony proces jakim jest bez wątpienia proces tworzenia improwizacji przez człowieka. Aby sprostac temu zdaniu, wyodrębnione zostały trzy główne składowe rzeczonoego procesu, a następnie odzwierciedlono je w zaimplementowanym systemie. Dla przypomnienia, elementy, o których mowa to:

- pamięć (znajomość wielu fraz muzycznych),
- umiejętności (łączenie fraz, modyfikacja, ozdabianie),
- wiedza z zakresu teorii muzyki.

Aby lepiej zrozumieć strukturę architektury systemu, należy przyjrzeć się dokładniej procesowi ludzkiej, naturalnej improwizacji. Trzeba spojrzeć na jej przebieg z perspektywy muzyka, który, znając kolejny etap progresji, jest tuż przed zagranie następnego taktu. Muzyk, o czym już pisano, musi mieć pewien zarys, czy też szablon melodii, którą ma zamiar zagrać. Nuty te mogą powstać m.in. podczas tzw. *ogrywania* akordu. Jest to nazwa budowania wokół niego melodii używając zestawu znanych zagrywek i szablonów. Są to muzyczne sztamperki oparte na zasadach teorii muzyki, gwarantujące poprawność melodii. Sposób ten jest jednak zbyt mało kreatywny, wręcz banalny, a więc system odtwarzający takie podejście byłby bardzo przewidywalny, nieciekawym, co stoi w sprzeczności z kreatywnością. Podczas tworzenia improwizacji, muzyk posługując się strzępkami swojej pamięci i fragmentami zapamiętanych melodii, może budować całkiem nowe i oryginalne. Im więcej zapamiętanych taktów, tym większe możliwości improwizatora (więcej kombinacji fraz). Zaimplementowany system stara się odzwierciedlić ten drugi sposób generowania melodii.

Często spotykanym rodzajem improwizacji, jest improwizowanie wokół zadanej melodii wzorcowej – tematu. W tym przypadku muzyk musi generować frazy zbliżone do wzorców¹. Powszechnym zabiegiem podczas tego rodzaju improwizacji jest urozmaicanie melodii poprzez dodawanie do niej ozdobników – nieznacznych uatrakcyjnających modyfikacji nie wpływających jednak znacznie na główną melodię. Elementem w architekturze odpowiedzialnym za dostarczenie bazowych taktów, łącznie z ich ozdabianiem, jest GENERATOR TAKTÓW, który na podstawie grupy wskazanych zagrywek, tworzy nowe takty. Robi to poprzez ich łączenie, przeplatanie, ozdabianie itp. Potrafi on przy tym dbać,

¹ Jest to rodzaj tzw. *wariacji*, której istotą jest przetworzenie tematu lub jego motywów.

aby muzyczne frazy były zbliżone do wskazanych wzorców – pozwalając na uzyskanie efektu improwizacji wokół zadanego tematu.

Improwizator, posiadając już pomysł na melodię w obrębie taktu, musi melodię tę skonfrontować z kontekstem improwizacji. Przez kontekst ten należy rozumieć tonację oraz, przede wszystkim, akord, który zostanie zagrany w obrębie taktu. Założono, że melodia stworzona przez muzyka składać się może z elementów pochodzących z wielu różnych fraz. Składowe takty, w wersji oryginalnej, nie musiały towarzyszyć akurat temu akordowi, który występuje w aktualnej improwizacji. Wynika stąd potrzeba dopasowania melodii do kontekstu improwizacji. Odbywa się to poprzez modyfikacje melodii. Zadanie to będzie realizowane przez odpowiedni element oznaczony jako ETAP KOREKCJI.

Gdy system wygeneruje odpowiednią liczbę taktów, które zostaną następnie połączone z progresją, można mówić o zakończeniu procesu improwizacji. Wprowadzono jednak dodatkowe, opcjonalne etapy, które mają na celu wykreowanie improwizacji, w której będzie można dopatrywać się nie tylko poprawności (braku dysonansów) ale również pewnej techniki kompozycji całego utworu. Utwór, w którym odbiorca wyczuwa zamysł kompozytorski, rozumiany jako wysokopoziomą strukturę, ma większą szansę na jego utożsamienie z dziełem człowieka. Etapy te to SKŁADANIE KOMPOZYCJI ORAZ PRZETWARZANIE KOŃCOWE.

Kolejnym pojęciem, którą należy uściślić jest tzw. *wejście* systemu. Główną zmienną, mającą diametralny wpływ na improwizację jest podkład muzyczny czyli progresja akordów. Zakłada się definiowanie jednego akordu na takt. Długość progresji w sposób bezpośredni określa długość improwizacji. Do dodatkowych zmiennych zaliczyć należy tonację utworu (skalę oraz tonikę). Choć większość elementów systemu jest konfigurowalna, to właśnie progresja oraz tonacja są niezbędne do rozpoczęcia improwizacji ².

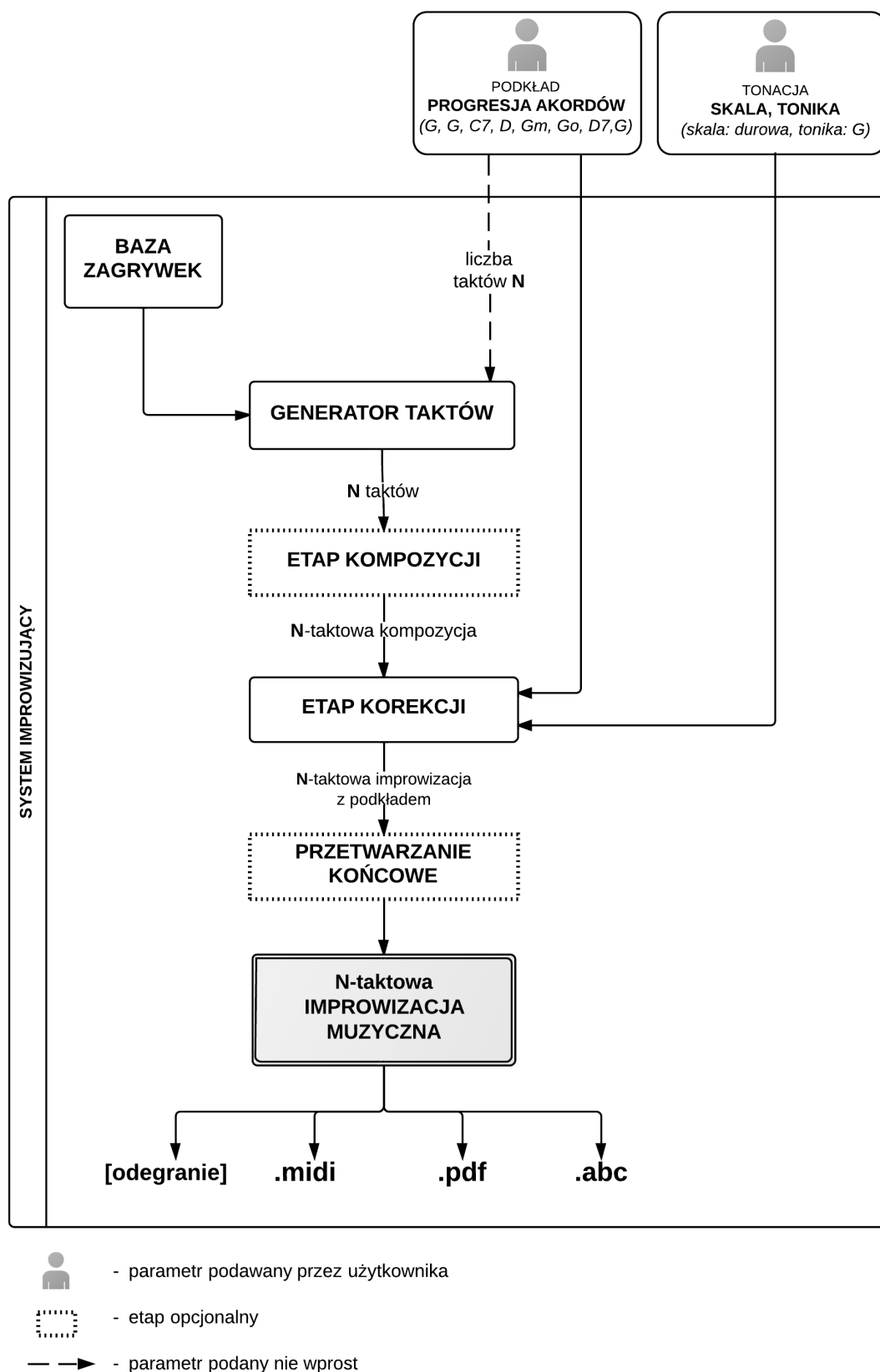
Główne elementy architektury systemu, wraz z przepływem prezentującym proces improwizacji prezentuje rysunek 4.1.

4.1. Przykład przebiegu procesu improwizacji

Gwoli pełnego zrozumienia schematu przedstawiającego architekturę systemu oraz rolę jaką odgrywają w nim poszczególne elementy, przedstawiono poniżej przykład generowania improwizacji. Większość elementów architektury, choć są w pełni konfigurowalne, na potrzeby prezentacji mechaniki systemu, przyjmą wartości domyślne.

Inicjalizacja procesu improwizacji wymaga podania dwóch kluczowych elementów: progresji akordów oraz tonacji. O ile podanie podkładu w postaci progresji akordów jest wymogiem oczywistym, tak wskazanie tonacji w postaci toniki oraz rodzaju skali, takim nie jest. Sama improwizacja jest prowadzona przez podkład, który winien być zbudowany

² Pozostałe parametry przyjmują swoje wartości domyślne.



Rys. 4.1: Architektura systemu improwizującego.

zgodnie z zasadami technik kompozytorskich³. Jednakże częstym wymogiem jest zgod-

³ Zakładając chęć stworzenia zgodnej z zasadami harmonizowania improwizacji; sam podkład może być dowolną, niekoniecznie zgodną z wspomnianymi technikami, progresją akordów.

ność melodii z określoną tonacją. Na tonację składają się: skala muzyczna, będąca pewnego rodzaju szablonem, oraz tonika, która wskazuje pierwszy stopień gamy (ukonkretnienia skali). Wskazana gama determinuje również progresję, będącą, podobnie jak skala, pewnego rodzaju schematem, opierającym się na stopniach gamy. Ponadto system umożliwia wybór pomiędzy dwoma priorytetami, jeśli chodzi o korekcję melodii – POPRAWNOŚĆ WZGLĘDEM PROGRESJI lub POPRAWNOŚĆ WZGLĘDEM TONACJI.

Przechodząc do konkretnego przykładu, założono, że użytkownik wskazał jako tonikę g^1 (g razkreślnie), a *dur* jako skalę muzyczną (zwaną systemem). Dzięki tonice, możemy ukonkretnić skalę, tworząc gamę *g-dur*:

$$g^1 \ a^1 \ h^1 \ c^1 \ d^1 \ e^1 \ f \ i s^1 \ g^2.$$

Kolejne stopnie gamy posłużą m.in. do budowy progresji akordów, która, jak wspomniano, jest pewnego rodzaju szablonem. Przyjęto, że użytkownik zdefiniował progresję w następujący sposób:

$$I \ I \ IV7 \ V \ Im \ Io \ V7 \ I.$$

Każdy z akordów składa się z łacińskiej liczby oraz przyrostka, który odpowiada rodzajowi akordu (patrz: tabela 3.2). Liczba wskazuje stopień gamy, który staje się prymą dla danego akordu. I tak – „I” oznacza akord durowy (brak przyrostka) o prymie będącej pierwszym stopniem gamy *g-dur*. Daje to w wyniku akord *g-dur*. Drugi akord progresji jest identyczny. Trzeci element progresji natomiast, to „IV7”. Zapis ten oznacza akord septymowy durowy oparty o prymę, która jest czwartym dźwiękiem gamy. W rezultacie, trzecim akordem, jest akord *C7* zwany akordem *c-dur* z septymą małą. Ostatecznie, szablon progresji podany przez użytkownika zostaje ukonkretniony do następującej postaci:

$$G \ G \ C7 \ D \ Gm \ Go \ D7 \ G.$$

Na schemacie architektury nie zaznaczono etapu ukonkretniania schematu progresji ponieważ jest on nieistotny z punktu widzenia głównego zadania systemu. Schemat prezentuje już ukonkretnioną progresję, jako zmienną przekazywaną przez użytkownika.

Pierwszy etap generowania improwizacji realizowany jest przez GENERATOR TAKTÓW. Jego zadaniem jest dostarczenie „surowych” taktów, które, po pewnych przeróbkach, staną się improwizacją. Ich liczba – N – nie jest podawana wprost przez użytkownika. Założymy jednak, że jeden akord przypadnie na jeden takt, długość progresji zdefiniuje parametr N . W naszym przypadku N przyjmie wartość 8, bo tyle akordów zawiera progresja podana przez użytkownika. Generator korzysta z BAZY ZAGRYWEK, która odzwierciedla pamięć muzyka. Z fragmentów tych zagrywek składa N taktów, które posłużą jako baza improwizacji. Generator winien spełniać szereg wymogów, zarówno związanych z charakterem

procesu improwizacji, jak i tych, które będą definiowane przez użytkownika. Ostatecznie GENERATOR TAKTÓW przekazuje N fraz do kolejnego etapu.

Z kolei opcjonalny ETAP KOMPOZYCJI, ma za zadanie stworzyć wrażenie, że improwizacja posiada pewien zamysł kompozytorski, jeśli chodzi o układ taktów. Przykładem, na tle którego wyraźnie widać pozytywny wpływ elementów kompozycji na odbiór improwizacji, jest sytuacja, w której się ją pominie. W takim przypadku, GENERATOR TAKTÓW dostarcza osiem **różnych** fraz, które finalnie staną się improwizacją. Ludzkie ucho jest jednak przyzwyczajone do utworów posiadających pewne formy. Nie ważne czy mówimy tu o pieśniach, które zazwyczaj posiadają powtarzalną melodię zwrotek i refrenów, czy *sonatach da chiesa* (wł. sonatach kościelnych) uporządkowanych pod względem tempa. I tak, ETAP KOMPOZYCJI, wprowadza pewną powtarzalność w układzie taktów, sprawiając przy tym, że improwizacja staje się łatwiejsza do odbioru oraz nabiera pewnego rytmu, który jest charakterystyczny dla większości dzieł muzycznych. Użytkownik ma tu również możliwość wyboru własnego szablonu kompozycji. Zakładając jednak domyślne parametry tego etapu, spośród ośmiu taktów przekazanych przez generator, wybrane zostaną cztery frazy oznaczone kolejnymi literami alfabetu – A, B, C oraz D. Mogą one zostać ułożone w następującą, ósmiotaktową kompozycję:

$A \ A \ B \ C \quad A \ A \ B \ D.$

Kompozycja taka nie jest, choć dla uzyskania najlepszego efektu powinna być, skorelowana z progresją akordów. Warto również zauważyć, że dla powtórek taktu A w każdym z widocznych czteroelementowych podciągów akordów, mogą przypaść różne akordy. Daje to ciekawy efekt, w którym każdy z taktów ma nieco inny wydźwięk, zgodny jednak z zasadami harmonizowania wynikającymi z układu progresji. Dzięki ETAPOWI KOMPOZYCJI do kolejnego etapu trafiają więc takty ułożone zgodnie z pewnym zamysłem, co sprzyja pozytywnemu odbiorowi improwizacji.

Zgodnie z uwagą poczynioną w jednej z poprzednich sekcji, nie byłoby rozsądnym oczekiwanie, iż takty złożone z fragmentów fraz różnego pochodzenia, bezbłędnie współbrzmiały z podkładem. Muzyk improwizujący, a w naszym przypadku również i maszyną, tworząc melodię korzysta z pewnego szablonu, który należy zmodyfikować tak aby odpowiednio współbrzmiał z podkładem lub tonacją. Poprawki takie nie zniekształcają jednak znacząco wzorcowej melodii. ETAP KOREKCJI ma za zadanie odzwierciedlić umiejętności muzyka, oparte na wiedzy teoretycznej, aby wprowadzić niezbędne poprawki do poszczególnych taktów w sposób, który skutecznie ograniczy liczbę ewentualnych dysonansów. Poprawki te, jak zostało już wspomniane, mogą odbywać się w dwóch kontekstach – względem tonacji lub względem progresji akordów. ETAP KOREKCJI generuje na swoim „wyjściu” poprawną, odnośnie wskazanych założeń, improwizację. Może ona trafić do drugiego z opcjonalnych etapów, w celu jej dodatkowej obróbki.

Improwizacje, które trafiają do etapu PRZETWARZANIA KOŃCOWEGO są już gotowe do zaprezentowania użytkownikowi. Etap ten umożliwia zdefiniowanie operacji, które podniosą walory muzyczne wygenerowanego utworu. Do operacji takich można zaliczyć m.in. „przycinanie” melodii, która zawiera nuty wyraźnie odbiegające od ich przeciętnej wysokości w obrębie utworu, co nie jest zgodne z powszechnym sposobem prowadzenia linii melodycznej. Ponadto, w etapie PRZETWARZANIA KOŃCOWEGO można umieścić proces, który zadba o odpowiednie zakończenie melodii - jej zbieżność do toniki.

Improwizacja, która opuści blok PRZETWARZANIA KOŃCOWEGO jest gotowa do zaprezentowania użytkownikowi. Uzyskany utwór można odegrać lub zapisać jako:

- plik notacji muzycznej ABC NOTATION,
- plik PDF⁴ zawierający zapis nutowy improwizacji,
- odtwarzalny plik muzyczny MIDI.

Pełniejszy opis poszczególnych etapów znaleźć można w następnym rozdziale.

⁴ PDF (ang. Portable Document Format, przenośny format dokumentu) – format plików służący do prezentacji, przenoszenia i drukowania treści tekstowo-graficznych, stworzony i promowany przez firmę Adobe Systems.

5. SZCZEGÓŁOWY OPIS METODY IMPROWIZACJI ORAZ JEJ IMPLEMENTACJI W SYSTEMIE

Celem pracy jest opracowanie metody improwizacji solo do zadanego podkładu muzycznego. Zakres niniejszej pracy obejmuje również opracowanie (adaptacje) metod i implementację prototypu. W poprzednich rozdziałach zaznajomiono czytelnika z dotychczasowymi realizacjami systemów improwizujących, wprowadzono również niezbędne dla zrozumienia pracy pojęcia z teorii muzyki, przedstawiono szczegółowo architekturę zbudowanej aplikacji.

Rozdział niniejszy w sposób szczegółowy opisuje zaprojektowaną metodę oraz jej poszczególne elementy. Wyjaśnia także zasady działania ważniejszych metod. W części tej uwzględniono również najistotniejsze kwestie z punktu widzenia implementacji.

5.1. Narzędzia oraz oprogramowanie użyte podczas tworzenia systemu

Aplikacja została napisana w języku programowania JAVA w wersji 1.7, której środowisko uruchomieniowe¹ wymagane jest do pracy z systemem. Pliki aplikacji oraz ich organizacja odpowiadają standardowemu archetypowi projektu MAVEN (wersja 2.2.1)², który służy do automatyzacji budowy oprogramowania na platformę JAVA. Projekt można zaimportować do popularnych zintegrowanych środowisk programistycznych (ang. IDE) w tym do ECLIPSE³, który był główną platformą tworzenia systemu.

System bazuje na notacji ABC, która pozwala tekstowo reprezentować zapis nutowy utworów muzycznych. Poniżej wymieniono dedykowane dla tej notacji aplikacje, które współpracują z systemem improwizującym. Są to:

- **ABCJ**⁴ – edytor, odtwarzacz oraz program katalogujący pliki notacji ABC; dostępny jest dla systemów Linux/Unix.
- **abc2midi**⁵ – program uruchamiany z linii komend do tworzenia plików muzycznych MIDI z plików notacji ABC; jeden z bardziej zaawansowanych.
- **abcm2ps**⁶ – program uruchamiany z linii komend do tworzenia plików w uniwersalnym języku opisu stron – PostScript; jest to wyjściowy format do tworzenia plików PDF.

¹ Konkretnie: JAVA(TM) SE RUNTIME ENVIRONMENT (BUILD 1.7.0-B147)

² Strona projektu MAVEN: <http://maven.apache.org/>

³ Dokładniej: ECLIPSE JUNO SR1

⁴ Strona internetowa programu: <http://abcj.ganderband.com/>

⁵ Strona internetowa pakietu programu: <http://abc.sourceforge.net/abcMIDI/>

⁶ Strona internetowa autora programu: <http://moinejf.free.fr/>

- **gs/gswin32**⁷ – program GHOSTSCRIPT odpowiednio dla systemów Linuxowych/Unixowych; program ten jest interpreterem plików PostScript i na ich podstawie umożliwia stworzenie pliku PDF.

Na czas realizacji projektu, jego źródła umieszczono w scentralizowanym systemie kontroli wersji SUBVERSION⁸.

5.2. Reprezentacja zapisu nutowego w systemie – ABC NOTATION

Większość operacji w muzycznym systemie improwizującym dotyczy melodii, a konkretniej, poszczególnych nut lub ich grup. Notacja muzyczna, której rdzeniem jest graficzna reprezentacja muzyki, nie jest przystosowana do obróbki przez maszyny. Jednym z pierwszych zadań w trakcie budowy systemu był wybór sposobu reprezentacji w nim melodii. Autor wybrał notację ABC NOTATION⁹, będącą specjalnym językiem tekstowej reprezentacji notacji muzycznej. Do jej zalet zaliczyć należy czytelność, ogromne możliwości jeśli chodzi o zapis struktur muzycznych oraz dostępność oprogramowania. Wśród wspomnianej gamy różnorodnych aplikacji znaleźć można edytory, odtwarzacze oraz konwertery plików ABC do plików dźwiękowych czy też graficznej reprezentacji nutowej. W niniejszej sekcji czytelnik zostanie zapoznany z podstawami notacji ABC (kompletny opis standardu znaleźć można w [1]).

Pliki notacji ABC powinny posiadać rozszerzenie *.abc*. Jeden plik zawierać może kilka odrębnych melodii (ang. tunes). Nazywany jest wtedy książką z melodiami (ang. tune book). System zakłada, iż jeden plik ABC zawiera jedną melodię. Listing 5.1 przedstawia przykładową zawartość pliku abc. Listing ten posłuży jako przykład w dalszej części rozdziału.

```

1 X:1
2 T:Polskie Drogi (melodia z serialu telewizyjnego)
3 C:Andrzej Kurylewicz
4 M:3/4
5 L:1/8
6 Q:1/4=90
7 K:G
8 B2 cB eg | b2 b2 b2 |]
```

Listing 5.1: Przykład pliku ABC.

Plik notacji składa się z dwóch części – sekcji nagłówkowej oraz nut. Nagłówki zawierają informacje dotyczące kontekstu utworu: jego tonację, metrum, tempo itp. Każda definicja parametru rozpoczyna się od litery oraz dwukropka, po którym znaleźć można jego wartość. Pojedynczy nagłówek zakończony jest przejściem do nowej linii. Sekcja

⁷ Strona internetowa programu: <http://www.ghostscript.com/>

⁸ Strona internetowa systemu: <http://subversion.apache.org/>

⁹ Strona internetowa notacji: <http://abcnotation.com/>

- **X** – definiuje numer utworu w obrębie jednego pliku ABC.
- **T** – jest skrótem od tytułu (ang.title) i pozwala określić nazwę utworu.
- **C** – jest skrótem od angielskiego słowa *composer* i określa kompozytora melodii.
- **M** – pozwala określić metrum utworu. Pierwsza cyfra oznacza podstawową jednostkę metryczną, cyfra druga – licznosc tych jednostek.
- **L** – pozwala wskazać domyślną długość wprowadzanych nut. Zapis „1/4” oznacza ćwierćnutę, „1/8” – ósemkę, itd.
- **Q** – określa tempo utworu poprzez określenie częstotliwości występowania danej nuty na minutę. Zapis „1/4=90” oznacza tempo dziewięćdziesięciu ćwierćnut na minutę.
- **K** – wskazuje tonację utworu poprzez podanie toniki oraz skali muzycznej. Zapis bazo-
wej nuty odpowiada zapisowi nuty w ciele pliku ABC. Można używać znaków chroma-
tycznych: # jako odpowiednik krzyżyka, b – bemola. Tonika może zostać wzbogacona
o przyrostek definiujący skalę muzyczną (jego brak wskazuje skalę durową). Może
być to m dla skali molowej, Dor dla doryckiej czy też Mix dla miksolidyjskiej. Istnieje
możliwość podania wartości parametru K również „none”, co oznacza nie korzystanie
z żadnej tonacji. Pole K powinno znajdować się na końcu sekcji nagłówkowej.

Ósma linia kodu na listingu przykładu, to tzw. ciało utworu czyli jego zapis nutowy. Każda podstawowa nuta przypisana jest odpowiedniej, zaczerpniętej z oryginalnego nazewnictwa, literze. Rysunek 5.1 prezentuje odwzorowanie nut na odpowiadające im litery w notacji ABC.



Rys. 5.1: Zapis poszczególnych wysokości dźwięków w notacji ABC.

Istotnym elementem zapisu nutowego, poza wysokością dźwięków, jest czas ich trwania. Notacja ABC nie przewiduje bezwzględnego sposobu wyrażania czasu trwania nut.

Jako punkt odniesienia bierze ona nutę podaną w polu L sekcji nagłówkowej¹⁰. Wprowadzenie pojedynczej nuty oznacza, iż trwać ona będzie tyle ile referencyjna wartość pola L. Aby skrócić czas trwania nuty, jako przyrostek należy „pomnożyć” lub „podzielić” czas jej trwania przez odpowiednią cyfrę lub ułamek. Rysunek 5.2 zawiera przykład takich operacji na zapisie pojedynczej nuty.



Rys. 5.2: Zapis długości trwania nuty G w notacji ABC dla referencyjnej wartości pola L odpowiadającej ósemce.

Rysunek 5.3 prezentuje wyrażenie opisujące zapis pojedynczej nuty w notacji ABC. Nawiasy kwadratowe, podobnie jak w języku wyrażeń regularnych, wskazują opcjonalną grupę znaków. Pierwsza, zielona sekcja, pozwala dodać przykładowej nucie G znak chromatyczny w postaci odpowiedniego prefiksu. Pominęto kasownik który nie jest używany w systemie improwizującym. Druga sekcja, koloru pomarańczowego, to nuta G. Należy pamiętać, iż może ona posiadać prefiks lub sufiks precyzujący jej przynależność do konkretnej oktawy (małej, razkreślnej lub dwukreślnej). Ostatnia, niebieska sekcja, odpowiada za sprecyzowanie czasu trwania nuty. Można go powielić poprzez „pomnożenie” przez cyfrę (d) lub „podzielić” przez nią za pomocą ukośnika.

[_ ^][,]G['][/] [d]

Rys. 5.3: Wyrażenie przedstawiające zapis pojedynczej nuty w notacji ABC

Pauzy w notacji reprezentowane są przez znak z. Czas trwania pauz modyfikowany jest w sposób analogiczny do nut.

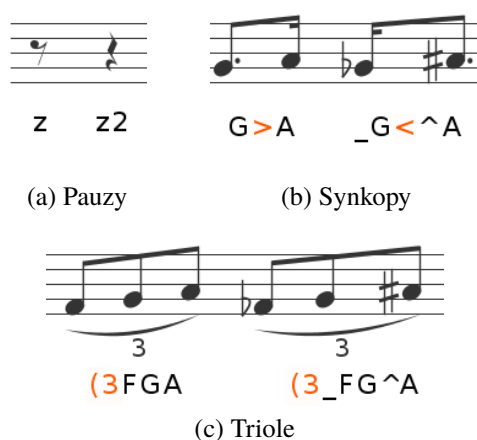
Podział na takty odbywa się poprzez wprowadzenie znaku pionowej linii: |. Oznaczenie końca utworu, to odpowiednio: ||.

Spośród wielu konstrukcji muzycznych oferowanych przez notację ABC wybrano dwie, które pozwalają na urozmaicenie utworu. Pierwszą z nich jest synkopa. Czas jej trwania opowiada dwóm jednostkom referencyjnym, określonym w nagłówku poprzez pole L. Konstrukcja synkopy w notacji ABC polega na połączeniu znakiem < lub > dwóch, niezmodyfikowanych pod względem czasu trwania, nut. Dźwięk wskazany jako „większy” przez znak nierówności jest dźwiękiem akcentowanym (krótszym).

Drugą konstrukcją muzyczną, uwzględnioną w systemie, jest triola. Podobnie jak synkopa, czas jej trwania pokrywa się z dwiema nutami referencyjnymi. Reprezentacja trioli rozpoczyna się od dwóch znaków: (3. Trzy kolejne nuty są traktowane jako należące do

¹⁰ W przypadku braku pola L przyjmuje ono wartość domyślną – 1/8 odpowiadającą ósemce.

trioli. Rysunek 5.4 prezentuje przykłady pauz, synkop oraz triol w notacji ABC. Kolor pomarańczowy składni wskazuje elementy charakterystyczne dla danej konstrukcji.



Rys. 5.4: Przykłady zapisu pauz, synkop oraz triol w notacji ABC.

Melodia w postaci pliku ABC jest analizowana przez system. Każdy element, począwszy od nuty, a skończywszy na skomplikowanych konstrukcjach, wyłuskiwany jest dzięki odpowiednim wyrażeniom regularnym. Podczas implementacji prototypu zdecydowano się korzystać z dwóch, urozmaicających rytm konstrukcji muzycznych – synkopy oraz trioli. System ignoruje pozostałe konstrukcje. Wynika to w sposób bezpośredni z zasady, na której opiera się wyszukiwanie podciągów w tekście za pomocą wyrażeń regularnych. W związku z powyższym przewidziano rozbudowę systemu o obsługę dodatkowych elementów notacji, która sprowadza się do sformułowania odpowiedniego wyrażenia regularnego, uwzględniającego nowy element w odpowiednich miejscach metody.

Notacja ABC pozwala na dodawanie tekstowych komentarzy do każdego z taktów. Można tego dokonać poprzez wprowadzenie tekstu ujętego w cudzysłów na początku taktu. Możliwość ta, choć nie wydaje się istotna z punktu widzenia tematyki pracy, w połączeniu z użytym konwerterem plików ABC, jest niezwykle ważna. Konwerter ów, potrafi wygenerować w sposób automatyczny podkład muzyczny na podstawie skrótowego zapisu akordu, który w tym celu powinien znaleźć się we wspomnianym komentarzu do taktu. Ułatwia to generowanie plików muzycznych, a dodatkowe, omówione w dalszej części pracy możliwości wspomnianego konwertera, pozwalają na wprowadzenie różnorodności do improwizacji.

5.3. Sposób reprezentacji głównych elementów muzycznych

Zaprojektowana metoda improwizująca korzysta z większości omówionych we wstępie pojęć oraz elementów muzycznych. Te drugie są bardzo różnorodne, a ponadto, każdy z nich jest wykorzystywany podczas improwizacji w określony sposób. Elementy te, zostały odzwierciedlone w systemie poprzez typy wyliczeniowe, które znakomicie nadają

się do przechowywania grup różnego rodzaju elementów. Każdy z nich jest definiowany wyłącznie w jego wnętrzu. Pozwala to na uniknięcie błędów wynikających z ręcznej inicjalizacji jego konkretnych instancji. Co więcej, za zastosowaniem typów wyliczeniowych, przemawia charakter implementowanego systemu – jest on prototypem, a więc większość zmiennych (np. dostępne typy akordów), będzie zawierała się w skończonym zbiorze, a ich umieszczenie wewnątrz typu enumeracyjnego zminimalizuje zdecydowanie liczbę błędów.

Język programowania, w którym zaimplementowano metodę improwizacji, oferuje rozbudowane typy enumeracyjne, umożliwiające przechowywanie dowolnie skomplikowanych informacji. Listę omawianych typów enumeracyjnych rozpocznie *nuta*.

Nuta – typ enumeracyjny ABCNote

Każda z nut, która w prototypowej wersji systemu, możliwa jest do zapisania, określana jest względem najniższej z nich. W zaimplementowanym systemie tą referencyjną nutą jest dźwięk *g*, z oktawy małej. Kolejne nuty określone były względem tego właśnie dźwięku. I tak, dla przykładu, *cis* razkresłone, reprezentowane w systemie pod nazwą *C1s*, przechowuje dwie informacje: o ile półtonów odległe jest od wspomnianej nuty referencyjnej (4 półtony) oraz jak reprezentowane jest w notacji ABC NOTATION (ciąg znaków: *_C*). Ponadto, typ enumeracyjny ABCNote zawiera metody, które umożliwiają wyszukiwanie nut po ich półtonowej odległości od referencyjnego dźwięku, ich transpozycję itp.

Interwał – typ enumeracyjny Interval

Wiele z działań przeprowadzanych w ramach generowania improwizacji, operuje na interwałowych odległościach pomiędzy dźwiękami. Każdy z interwałów dostępny jest w typie enumeracyjnym Interval, w ramach którego zdefiniowano wszystkie odległości, począwszy od prymy czystej (typ UNISON_PERFECT), do oktawy czystej (typ OCTAVE_PERFECT). Każdy interwał przechowuje informację jaką odległość, w półtonach, określa.

Rodzaj skali muzycznej – typ enumeracyjny ScaleType

Jak zostało już powiedziane – każda skala muzyczna zdefiniowana jest poprzez interwały występujące pomiędzy kolejnymi jej stopniami. W ten właśnie sposób, poprzez tablicę zawierającą wspomniane interwały (typu Interval), definiowana jest skala muzyczna w zaimplementowanym systemie. Dodatkowo, każdy typ skali zawiera stałą typu *String*, zawierającą jej słowny opis. Typ enumeracyjny ScaleType zawiera 11 rodzajów skal. Znaleźć wśród nich można zarówno te najpopularniejsze (durową, molową), jak i rzadziej używanych (skala zmniejszona, zwiększona, czy też bluesowa skala heptatoniczna).

Omawiany typ enumeracyjny jest jedynie szablonem, na podstawie którego, odpowiednia klasa – fabryka, po wskazaniu toniki, generuje ukonkretnioną skalę (klasa Scale).

Obiekt typu Scale składa się już nie z interwałów, lecz konkretnych dźwięków.

Rodzaj akordu – typ enumeracyjny ChordType

Akordy, składowe podkładu muzycznego, na którym opiera się improwizacja, wynikają bezpośrednio ze struktur skal muzycznych. Dlatego też, definicje akordów w typie wyliczeniowym ChordType nie uwzględniają konkretnych stopni gam, czy też interwałów lecz skale, z których dany wielodźwięk się wywodzi. Opisywany typ enumeracyjny zawiera również szereg dodatkowych informacji. Pierwszą z nich jest przyrostek, który dodany do toniki, stanowi zapis danego akordu (np. „7” dla akordu septymowego). Ponadto, co jest wyjaśnione w dalszej części pracy, każdy akord, w tablicy, definiuje stopnie skali, z którymi współbrzmi wyjątkowo dysonansowo.

Jeden z dodatkowych, wykorzystanych podczas tworzenia systemu programów, używa wspomnianych przyrostków do automatycznego generowania podkładu muzycznego czyli akordów. Niestety nie rozpoznaje on wszystkich dostępnych w systemie wielodźwięków. Dlatego też, każdy akord w obrębie typu ChordType, o ile jest to potrzebne, posiada pole, które zawiera definicję wielodźwięku w postaci kolejnych interwałów, z których się składa. Definicja taka zamieniana jest na specjalny nagłówek MIDI tuż przed przystąpieniem do wygenerowania pliku dźwiękowego. Dzięki temu możliwe jest odegranie podkładu złożonego z akordów nie obsługiwanych przez wspomniany program.

Szablon progresji akordów – typ enumeracyjny ProgressionTemplate

Akordy, omówione w poprzedniej sekcji, służą przede wszystkim do budowania podkładu muzycznego – progresji. Ta z kolei, tworzona jest na podstawie szablonu oraz toniki. Szablony owe reprezentowane są w systemie w ramach typu enumeracyjnego ProgressionTemplate, którego każdy element definiuje tablicę z kolejnymi typami akordów (typ ChordType) oraz stopniami skali, na których powinny one być zbudowane. Specjalna klasa – fabryka, na podstawie szablonu (typ ProgressionTemplate) oraz konkretnej skali muzycznej (klasa Scale), generuje właściwą progresję akordów. Składa się ona z obiektów typu ConcreteChord, czyli ukonkretnionych akordów.

Typ enumeracyjny ProgressionTemplate umożliwia wybór spośród ośmiu progresji akordów. Znaleźć można wśród nich podstawowe, charakterystyczne dla bluesa progresje, np. typ BLUES_BASIC oraz jego odmianę molową – typ BLUES_BASIC_MINOR. Poza standardowymi progresjami warto zwrócić uwagę na znaną miłośnikom bluesa progresję, rozpropagowaną przez klasyka bluesa – W.C.Handy’ego – zwaną *quick-change* (typ HANDY_QUICK_CHANGE_7TH).

Rodzaj akompaniamentu podkładu muzycznego – typ enumeracyjny MIDIChordAccompanimentType

Podkład muzyczny, w kontekście niniejszej pracy, kojarzyć należy z progresją akordów.

Trzeba jednak zaznaczyć, że pomimo rozumienia akordu jako współbrzmienia kilku dźwięków, progresję odgrywać można na wiele sposobów – począwszy od kilkakrotnego odegrania akordu w obrębie taktu, poprzez przeplatanie go z dźwiękiem toniki, a skończywszy na graniu, w określonym porządku, jego kolejnych dźwięków. Dzięki możliwości wyboru sposobu odgrywania podkładu muzycznego, można zwiększyć różnorodność kolejnych improwizacji, a nawet zmniejszyć w ich obrębie stopień słyszalnych współbrzmień dysonansowych.

Wybór sposobu odgrywania podkładu możliwy jest dzięki programowi do konwersji plików ABC do formatu MIDI. Aplikacja, o której mowa – *abc2midi* – pozwala na zdefiniowanie specjalnego nagłówka, który w sposób automatyczny generuje opisany w nim rodzaj podkładu muzycznego. Sprecyzowanie rodzaju odgrywania podkładu sprowadza się do stworzenia ciągu znaków, które reprezentują poszczególne elementy akordu do odegrania. Są to m.in. następujące znaki:

- litera **c** – odegranie całego wielodźwięku,
- litera **f** – odegranie pierwszego dźwięku akordu – toniki z oktawy razkreślnej,
- litera **g** – odegranie pierwszego dźwięku akordu – toniki z oktawy małej (basowe wzmocnienie toniki),
- litera **h** – odegranie drugiego dźwięku akordu,
- litera **i** – odegranie trzeciego dźwięku akordu,
- litera **z** – pauza muzyczna.

Nagłówek umieszcza się przed definicją melodii w pliku ABC. Przyjmuje on następującą postać:

%%MIDI gchord fcfc

gdzie „*%%MIDI gchord*” jest definicją nagłówka, a „*fcfc*” jego parametrem, złożonym z opisanych wyżej znaków.

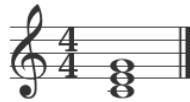
Rysunek 5.5 przedstawia kilka sposobów odgrywania akordu *c-dur* wraz z ciągiem znaków, pozwalającym na jego zdefiniowanie za pomocą wspomnianego nagłówka MIDI.

5.4. Generator taktów

Generator taktów odgrywa istotną rolę w opisywanym systemie. Dostarczane przez niego takty w bezpośredni sposób kształtują melodię improwizacji. Element ten odwzorowuje ludzki sposób improwizowania – kreowania nowych taktów na podstawie fragmentów zapamiętanych melodii. Aby uściślić wyobrażenie o rzeczonym generatorze, wymienimy cechy, które powinien spełniać, aby w jak największym stopniu przypominać pomysłowość improwizującego muzyka.

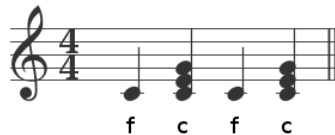
Generator taktów bazuje na skończonej liczbie znanych fraz muzycznych. Zbiór ten reprezentowany jest w systemie jako BAZA ZAGRYWEK. Ponadto, improwizowana linia melo-

Akord c-dur



(a) Bazowy akord
c-dur.

Definicja sposobu odgrywania podkładu:
%%MIDI gchord fcfc



(b) Pierwszy sposób odegrania podkładu.

Definicja sposobu odgrywania podkładu:
%%MIDI gchord ghiihih



(c) Drugi sposób odegrania podkładu.

Definicja sposobu odgrywania podkładu:
%%MIDI gchord gzzc



(d) Trzeci sposób odegrania podkładu.

Rys. 5.5: Sposoby odgrywania podkładu bazującego na akordzie *c-dur*

dyczna powinna powstawać poprzez łączenie elementów pochodzących z **różnych** taktów bazowych. Zadaniem generatora taktów jest również naśladowanie zdolności muzyka w umiejętności upiększania melodii podczas jej tworzenia, poprzez dodawanie ornamentów muzycznych. Równocześnie, generator powinien kierować się preferencjami użytkownika, co do tworzonych taktów – np. generując wariacje wokół zadanego tematu. Kontrola powinna obejmować także swobodę improwizacji, rozumianą przez stopień odejścia od melodii bazowych taktów, czy też liczbę dodanych ozdobników.

Jako generator taktów wybrano algorytm genetyczny, przystosowując go do realizacji specyficznego zadania, jakim jest generowanie kolejnych taktów improwizacji.

Opis implementacji poszczególnych elementów algorytmu genetycznego poprzedza wstęp objaśniający sposób działania rozwiązań z rodziny algorytmów ewolucyjnych.

5.4.1. Wstęp do algorytmów genetycznych

Nazwa „algorytmy genetyczne”, podobnie jak proveniencja stojącego za nimi pomysłu, są ściśle związana z naturą. Algorytm genetyczny próbuje naśladować znane w przyrodzie zjawisko ewolucji. Opiera się on na obserwacji przyrody, z której wynika, że populacje ewoluujących osobników charakteryzują się, rosnącym z pokolenia na pokolenie, przystosowaniem do otaczających je warunków. Algorytm genetyczny, rozpoczynając swe działanie od odpowiednio liczego zbioru przypadkowych rozwiązań danego problemu (populacji), krzyżuje je ze sobą, wybierając do kolejnej iteracji te najlepsze (najlepiej dostosowane). Przeprowadza on dodatkowo operacje mutacji, naśladując istniejące w naturze losowe zjawisko zmiany materiału genetycznego. Zmiany takie nadają, dotkniętym nimi osobnikom, cechy niespotykane w populacji. Cechy te, poprzez swą odmienność, w naturalnych warunkach byłyby niemożliwe do odziedziczenia, a tym samym zaistnienia w obrębie populacji. Mutacje pomimo swojego losowego charakteru potrafią czasami wskazać populacji nowy kierunek rozwoju. Jest to jednak pożądane zjawisko w populacjach, które poprzez podobieństwo większości osobników, utknęły w martwym punkcie – pośród osobników brak nowego materiału genetycznego, a żaden z osobników nie osiągnął dostatecznego poziomu dostosowania (nie osiągnął dostatecznie wysokiej oceny).

Algorytmy genetyczne znajdują zastosowanie w zadaniach optymalizacji. Potrafią one rozwiązać wiele skomplikowanych problemów poprzez połączenie metod przeszukiwania bezpośredniego oraz stochastycznego. Daje im to przewagę nad algorytmami bezpośredniego przeszukiwania w postaci większej niezawodności, a przetwarzanie całej populacji potencjalnych rozwiązań, stawia je ponad większością metod optymalizacyjnych.

Niewątpliwą zaletą algorytmów genetycznych jest dostarczanie, w każdej iteracji, nie jednego, lecz całego zbioru potencjalnych rozwiązań. Pojedyncze rozwiązanie w obrębie populacji zwane jest osobnikiem. Struktura wewnętrzna osobnika, zwana kodowaniem, wynika ze specyfiki zadania, do którego rozwiązania algorytm genetyczny został zastosowany.

Głównym elementem związanym z przeszukiwaniem przestrzeni rozwiązań jest krzyżowanie najlepszych osobników. Operacja taka, najprościej rzecz ujmując, wymienia elementy struktury pomiędzy dwoma osobnikami, tworząc tym samym dwa nowe rozwiązania. Przykład najczęściej spotykanego rodzaju krzyżowania – jednopunktowego – prezentuje rysunek 5.6. Ilustruje on w jaki sposób dochodzi wymieniania się „materiału genetycznego” dwóch rodziców przy jednym punkcie krzyżowania, czego wynikiem jest powstanie dwóch nowych rozwiązań – potomka I oraz II.

Podczas budowy systemu użyto jednego z dostępnych darmowych frameworków za-



Rys. 5.6: Przykład krzyżowania jednopunktowego.

wierających gotowe struktury programistyczne umożliwiające szybkie tworzenie algorytmów genetycznych¹¹.

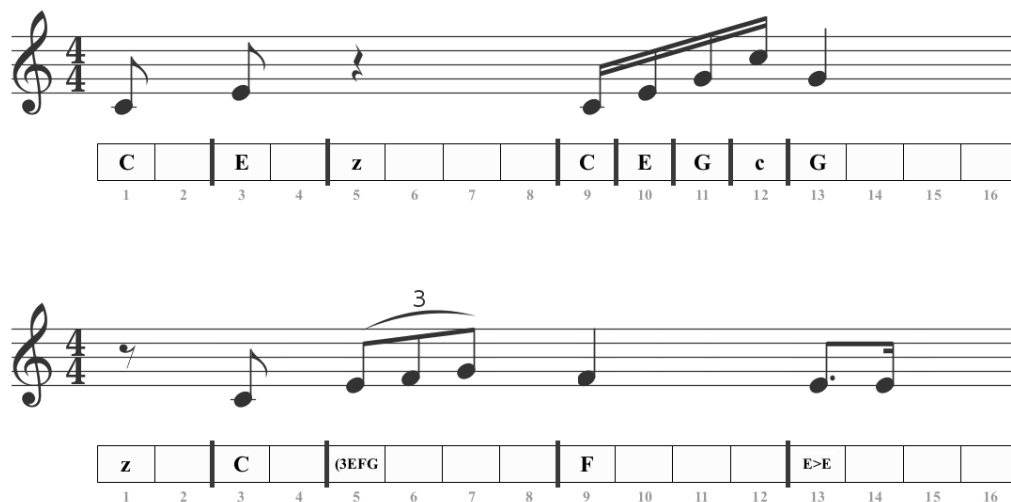
5.4.2. Struktura pojedynczego osobnika

Postać pojedynczego osobnika, dobrana jest najczęściej tak, aby po krzyżowaniu, nowe rozwiązania zgodne były z założeniami dotyczącymi ich struktury. Wynika to z faktu, że krzyżowanie nie ma wiedzy z jakiego rodzaju osobnikami ma do czynienia – punkt „przecięcia” dobierany jest losowo, bez względu na wewnętrzną budowę osobników. Dlatego też, ich struktura musi być zawsze poprawna – bez względu na wybrany punkt krzyżowania. Podczas budowy systemu improwizującego stwierdzono, iż standardowa, tablicowa reprezentacja osobnika byłaby zbyt skomplikowana – należałoby uwzględnić w niej wszystkie użyte struktury muzyczne, dbać o zachowanie odpowiedniego metrum itp. Ponadto niefortunny dobór punktu krzyżowania mógłby spowodować połączenie dwóch nut w środku ich reprezentacji kodowej. Mogłoby to doprowadzić do powstania w potomku nut nie należących do żadnego z rodziców, co nie jest zgodne z założeniem – czerpania melodii ze znanych fraz. Mogłoby również, przez swoją przypadkowość, zaburzyć harmonię nowych taktów. Postanowiono zatem przenieść część ciężaru dbania o poprawność reprezentacji osobników z zaprojektowania ich, odpornej na wspomniane sytuacje, struktury na zdefiniowanie specjalnych operacji tworząc „świadome muzycznie” operatory. Rzeczone procesy działać będą na opisanej wcześniej reprezentacji struktur muzycznych w notacji ABC, a ich „świadomość muzyczna” będzie polegała na radzeniu sobie z problemem niefortunnego dobrania miejsc ingerencji w strukturę fraz muzycznych.

Reprezentacja taktów (osobników) w populacji bazuje na powszechnie stosowanej strukturze tablicowej. Długość tablicy wynika z metrum oraz najdrobniejszej, możliwej do zapisu jednostki czasu trwania nuty. W naszym przypadku, dla metrum 4/4 i szesnastki, jako najkrótszej do zapisania nuty, długość tablicy wyniesie 16. Gdy w danej jednostce czasu rozpoczyna się struktura muzyczna, w odpowiadającej jej komórce tablicy zapisywana jest jej reprezentacja w notacji ABC. Element muzyczny, np. nuta, trwa przez tyle

¹¹ Użyto frameworka Watchmaker Framework. Strona projektu: www.watchmaker.uncommons.org

pustych elementów tablicy (jednostek czasu), aż do zapisu kolejnej struktury lub końca taktu. Przykład takiego zapisu ilustruje rysunek 5.7.



Rys. 5.7: Przykłady tablicowej reprezentacji taktów.

5.4.3. Operacje „świadome muzycznie”

Przykładem wspomnianych, świadomych muzycznie operacji, jest m.in. zaimplementowane krzyżowanie. Działanie to, przed przystąpieniem do rozbicia danego taktu na części, sprawdza czy wylosowany punkt przecięcia nie wypadł w niepożądanym miejscu. Do miejsc takich zaliczyć należy triole i synkopy, które muszą zajmować odpowiednią liczbę wolnych komórek. Należy pamiętać, iż dla zwykłych nut, krzyżowanie może jedynie zmienić czas ich trwania. Nie ma tu więc ryzyka, że potomkowie będą posiadać nieprawidłową strukturę. Przykład bezproblemowego krzyżowania dla zwykłych nut oraz nieprawidłowych wyników tej operacji dla triol, prezentuje rysunek 5.8. Świadomy muzycznie operator krzyżowania, przed przystąpieniem do „rozcięcia” taktu, sprawdza czy nie trafił na triolę lub synkopę. W takim przypadku, przed krzyżowaniem, zamienia je na odpowiadające im struktury złożone ze zwykłych nut. Synkopa, sama w sobie, jest złożona ze zwyczajnych nut – jedynie notacja ABC wprowadza odpowiadający jej element notacji. Nie ma więc problemu z zapisem synkopy używając podstawowych nut. Triola natomiast wprowadza pewną nieregularność do rytmu melodii. Każda z nut w jej obrębie, trwa $\frac{2}{3}$ podstawowej jednostki czasu trwania nuty. Konieczne zatem jest sprowadzenie trzech nut trioli do dwóch, które, już jako zwyczajne, odzwierciedlałyby przebieg melodii w obrębie trioli. Operator krzyżowania zamienia triolę na dwie nuty o podstawowym czasie trwania, biorąc z trioli najwyższy oraz najniższy dźwięk, przy zachowaniu kolejności ich występowania. Rysunek 5.9 prezentuje przykład działania opisaney operacji dla trioli oraz synkopy.

Etap krzyżowania w algorytmie genetycznym odpowiada za tworzenie nowych taktów, bazując na znanych nam frazach muzycznych (populacji początkowej). W każdej z iteracji

Example (a) shows two musical phrases in 3/8 time. The first phrase consists of a quarter note C and a half note E. The second phrase consists of a quarter note F, a quarter note G, and a half note D. The tabular representation shows the notes in a 6-unit grid. For the first phrase, Rodzic I is [C, empty, empty, empty, E, E] and Rodzic II is [F, G, D, F, G, D]. For the second phrase, Potomek I is [C, D, F, G, D] and Potomek II is [F, G, empty, empty, E, E]. A vertical line at position 3 indicates the crossover point.

(a) Bezproblemowe krzyżowanie podstawowych jednostek muzycznych – nut. Kropka przy nucie oznacza dodanie do niej połowy jej czasu trwania.

Example (b) shows a musical phrase in 3/8 time consisting of a triplet of eighth notes (G, F, E) followed by a quarter note D and a half note C. The tabular representation shows the notes in a 6-unit grid. For the first phrase, Rodzic I is [(3GFE), empty, empty, D, C] and Rodzic II is [z, empty, C, D, E]. For the second phrase, Potomek I is [(3GFE), C, D, E] and Potomek II is [z, empty, empty, D, C]. A vertical line at position 3 indicates the crossover point. A note explains that a triplet requires four units of time, but only two are available in the 3/8 time signature.

(b) Nieprawidłowe krzyżowanie z punktem przecięcia wypadającym w obszarze trioli i synkopy.

Rys. 5.8: Przykłady krzyżowania tablicowej reprezentacji taktów.

Example (c) shows two musical phrases. The first phrase is a triplet of eighth notes (C, E, G) followed by a quarter note. The second phrase is a quarter note F followed by a quarter note G. The tabular representation shows the notes in a 4-unit grid. For the first phrase, the simplified representation is [C, empty, G, empty]. For the second phrase, the simplified representation is [F, empty, empty, G].

Rys. 5.9: Przykłady uproszczenia tablicowej reprezentacji trioli oraz synkopy do odpowiadającym im form złożonych z podstawowych nut.

algorytm stara się dobrać w pary najlepsze osobniki. Dla każdej ze wskazanych par – rodziców – wybiera on losowy punkt krzyżowania.

5.4.4. Funkcja oceny – podobieństwo melodii

Powyżej wyjaśniono zagadnienia związane z krzyżowaniem tablicowych reprezentacji taktów. Należałoby teraz sprecyzować, które melodie są dla systemu „najlepsze” czyli w jaki sposób wybiera on z populacji te najlepiej dostosowane osobniki.

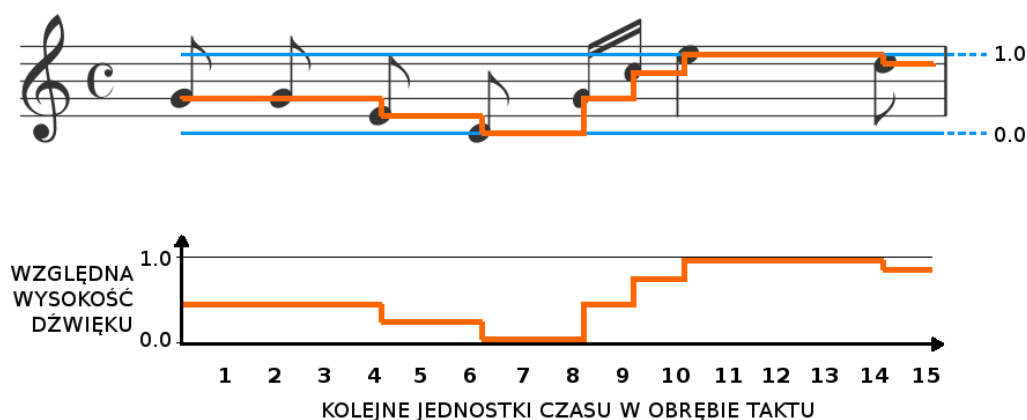
Funkcja oceny jest niezwykle istotnym elementem algorytmu genetycznego. Ocenia ona osobniki w populacji i przyporządkowuje tym samym każdemu z nich numeryczną wartość określającą stopień ich przystosowania, czyli, innymi słowy, w jakim stopniu odpowiadają one naszym oczekiwaniom. Dla najbardziej powszechnego zastosowania algorytmów genetycznych – optymalizacji, funkcja oceny jest po prostu funkcją optymalizowaną, a każdy z osobników, jako jej argument, otrzymuje ocenę równą jej wartości w punkcie, który reprezentuje. Pozostaje pytanie – jak ocenić każdy ze specyficznych osobników, jakimi są niewątpliwie takty, oraz jak ocenę tę odzwierciedlić w wartościach liczbowych? Zgodnie ze wstępnymi założeniami, osobniki w kolejnych iteracjach algorytmu, powinny bazować na początkowym zbiorze zagrywek – być do nich podobne, jednak podobieństwo to nie powinno być zbyt duże. Wynika to z definicji improwizacji, która jest procesem twórczym, a nie odtwórczym. Podobieństwo, o którym mowa, nie powinno być równocześnie zbyt nikłe, gdyż wówczas system odbiegałby od bazowych zagrywek, które oprócz pomysłu na melodie zawierają w swojej strukturze cenne wartości, takie jak harmonia, czy też unikająca dysonansów budowa. Nie ulega więc wątpliwości, że zaimplementowana w systemie funkcja oceny musi bazować na podobieństwie taktów. Co jednak należy rozumieć przez podobieństwo dwóch melodii? Czy odpowiada za nie bezpośrednie podobieństwo melodii równoznaczne z leksykalnym podobieństwem zapisu taktów w notacji ABC? Aby odpowiedzieć na to pytanie należy przypomnieć spostrzeżenie poczynione w oparciu o definicję pojęcia interwału. Melodii, poza rytmem, nie określają bezwzględne wartości wysokości poszczególnych dźwięków, a odległości pomiędzy nimi. Nawet najlepszym śpiewakom zdarza się nie trafić w pierwszy dźwięk melodii – może być on nieznacznie niższy lub wyższy. Nie wpłynie to jednak na rozpoznawalność pieśni, o ile zachowane zostaną rytm i interwały pomiędzy kolejnymi nutami. Ponadto, częstym zabiegiem jest transpozycja utworu – przeniesienie każdej z nut o określony interwał w górę lub w dół. Celem takiej operacji jest dostosowanie melodii do możliwości danego wykonawcy czy też instrumentu. Podobieństwo melodii nie może zatem zostać sprowadzone do porównywania wysokości i czasu trwania poszczególnych nut. Odpowiednim podejściem jest porównywanie kolejnych odległości pomiędzy wysokościami dźwięków melodii.

Jednym ze sposobów opisywania utworu muzycznego jest kod Parsona (ang. Parsons code)[13]. Polega on na opisie melodii z użyciem prostej notacji, która odzwierciedla ruch melodii w obrębie utworu. Notacja ta obejmuje trzy możliwe kierunki wędrówki melodii:

- UP (z ang.: w górę) – skrót **U**,
- DOWN (z ang.: w dół) – skrót **D**,
- REPEAT (z ang.: powtórzenie) – skrót **R**.

Dodatkowym oznaczeniem jest * – pierwszy, referencyjny dźwięk utworu. Opisana w ten sposób melodia nie uwzględnia bezwzględnych wartości wysokości dźwięków, a jedynie stosunek danej nuty do jej poprzedniczki i następczyni. Dzięki temu, kod Parsona nadaje się do identyfikacji i wyszukiwania melodii podobnych do siebie. W systemie zaimplementowano podobny, zwany konturowym, sposób opisu melodii, wzorując się na pracy Zhu i Kankanhalliego [24].

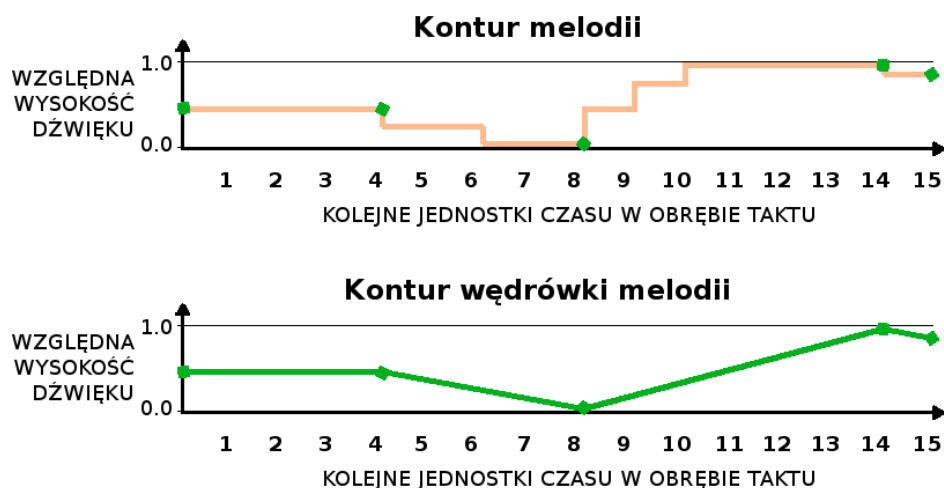
Zaimplementowany kontur, w znormalizowany sposób, opisuje dwa aspekty taktu – wzajemne relacje odległości pomiędzy nutami, oraz, podobnie jak kod Parsona, wędrówkę melodii. Pierwszym etapem tworzenia konturu jest określenie najniższego oraz najwyższego dźwięku w obrębie taktu. Dźwiękom tym przypisuje się odpowiednio wartości 0 oraz 1. Każdą z nut przenosi się następnie na kartezjański układ współrzędnych, którego oś odciętych odzwierciedla zaistnienie danej nuty w obrębie czasu trwania taktu; oś rzędnych natomiast, znormalizowany względem skrajnych wysokości dźwięków, interwał od najniższego dźwięku w takcie. Kontur ten będzie nazywany *konturem interwałowym*. Przykład konstruowania konturu melodii ilustruje rysunek 5.10.



Rys. 5.10: Przykład budowy konturu interwałowego.

Ponadto, na podstawie konturu melodii, budowany jest kontur wędrówki melodii. Kontur ten ma za zadanie wskazać miejsca zmiany wędrówki dźwięków w górę lub w dół oraz zilustrować tempo tych wędrówek (poprzez nachylenie danej części konturu). Nowy rodzaj wykresu zbudowany jest z punktów konturu interwałowego, w których następuje zmiana kierunku wędrówki dźwięków. Punkty te (kolor zielony), wraz z przykładem budowy nowego konturu, znajduje się na rys 5.11. Do punktów wchodzących w skład konturu wędrówki melodii należy zaliczyć również punkt rozpoczynający oraz kończący pierwszy dźwięk taktu.

Opisane kontury są niewrażliwe na transpozycję melodii oraz potrafią skojarzyć ze sobą melodie w których wszystkie interwały zostały w proporcjonalny sposób powiększone lub pomniejszone.



Rys. 5.11: Przykład budowy konturu wędrówki melodii.

Cechą melodii, która nie jest bezpośrednio uchwycona w opisanych konturach, a która może być istotna podczas określania podobieństwa dwóch melodii, jest rytm. Aby uzupełnić informacje wydobyte z taktu o te związane z rytmem, zdefiniowano wektor, który w znormalizowany sposób określa jak bardzo intensywny jest dany takt pod względem mnogości nut w nim występujących. Podstawową zmienną określającą ogólność wektora jest N – liczba mówiąca dla ilu części melodii obliczyć wspomniany wskaźnik. I tak, dla $N=2$, takt zostanie podzielony na dwie równe części, w obrębie których policzony zostanie wskaźnik zróżnicowania rytmu i umieszczony w wektorze cechy. Pojedynczy wskaźnik jest ilorazem liczby nut w danej części taktu do ich maksymalnej możliwej liczby. Wspomniana maksymalna liczba nut wynika bezpośrednio z określonej w systemie najkrótszej możliwej do zapisania nuty. Przykład obliczenia wektora zróżnicowania rytmu dla różnych wartości parametru N znajduje się na rysunku 5.12.

Im większa wartość parametru N tym dokładniejszy opis zróżnicowania taktu pod względem rytmu. Dla $N=1$ wektor jest jednoelementowy i opisuje takt, jako zawierający przeciętną liczbę nut (wartość $\frac{9}{16} \approx 0.56$). Przy N równym 4 można pokusić się o bardziej rozbudowany rytmiczny opis taktu: rozpoczyna się on w miarę szybkim rytmem (0.5) przechodzi do punktu kulminacyjnego (1.0), a następnie, stopniowo malejąc, zatrzymuje się na pojedynczej ćwierćnucie (0.25).

Rysunek 5.13 prezentuje specjalny element graficznego interfejsu użytkownika (ang. GUI) służący do porównywania dwóch taktów pod względem ich konturów. Kontur wędrówki melodii nazwany jest w elemencie GUI *konturem dynamiki* – jest to jego robocza nazwa, która pozostała na tzw. rzutcie ekranu z wcześniejszej wersji aplikacji.

Każdy z trzech sposobów opisu taktu – kontur melodii, kontur wędrówki melodii oraz zróżnicowanie rytmiczne – redukowany jest w systemie do postaci wektora. Kontury stają się szesnastoelementowymi tablicami z odpowiadającymi kolejnym jednostkom czasu wysokościami dźwięków. Wskaźnik zróżnicowania natomiast, jest wektorem sam



N=1 |----- 9 nut
16 nut maksymalnie

Wektor zróżnicowania rytmu: **{0.56}**

N=2 |----- 6 nut
8 nut maksymalnie |----- 3 nuty
8 nut maksymalnie

Wektor zróżnicowania rytmu: **{0.75 , 0.38}**

N=4 |----- 2 nuty
4 nuty maks. |----- 4 nuty
4 nuty maks. |----- 2 nuty
4 nuty maks. |----- 1 nuta
4 nuty maks.

Wektor zróżnicowania rytmu: **{0.5 , 1.0 , 0.5 , 0.25}**

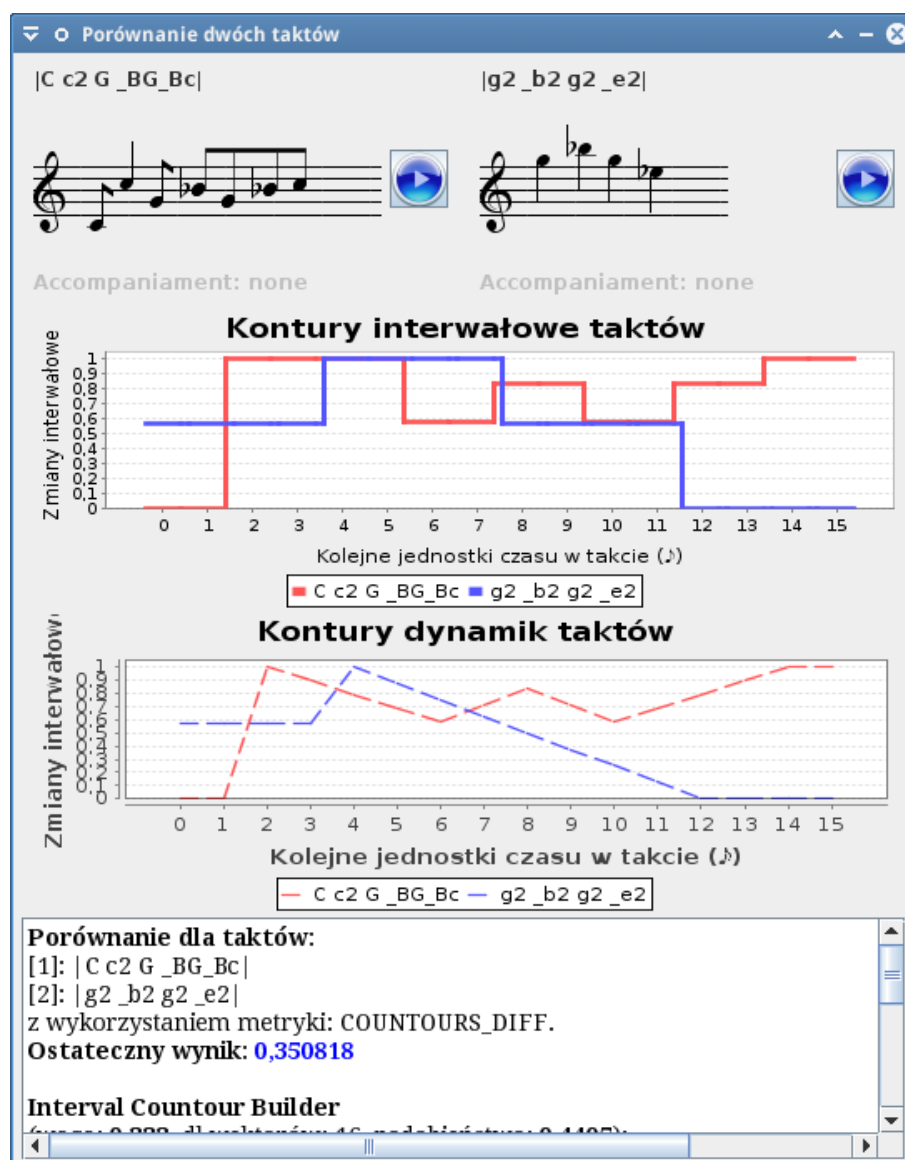
Rys. 5.12: Przykład budowy wektora zróżnicowania rytmu dla różnych wartości parametru N. Najmniejszą wartością rytmiczną jest szesnastka.

w sobie. Postać wektorowa pozwala na sprowadzenie czynności obliczania podobieństwa taktów do kalkulowania odległości pomiędzy dwoma wektorami. Jeżeli wektory są nieznacznie od siebie oddalone, można mówić o podobieństwie melodii. Element systemu odpowiedzialny za porównywanie melodii może korzystać ze wszystkich wskaźników jednocześnie. Oblicza on odległości pomiędzy odpowiadającymi sobie modelami taktów, a następnie liczy ich średnią. Aby kontrolować wpływ poszczególnych modeli na ostateczny wskaźnik podobieństwa, każdemu z nich przypisuje się wagę, która biera pod uwagę podczas liczenia średniej ważonej wyników, czyli finalnej wartości podobieństwa.

Powyżej przedstawiono trzy modele, które mogą opisywać takty, a tym samym służyć do ich porównywania. Podkreślono, iż są one niewrażliwe na transponowanie lub „rozciąganie” melodii. Dodatkowo, ich normalizacja uodparnia je na wpływ różnic w rozpiętościach zestawianych ze sobą melodii. Ponadto, wskaźniki same w sobie, są interesującymi sposobami opisu poszczególnych aspektów melodii.

5.4.5. Ostateczna postać funkcji celu

Normalizacja wektorów modeli sprawia, iż ostateczna wartość podobieństwa jest liczbą z przedziału $< 0; 1 >$. Jak zostało już wcześniej powiedziane, algorytm genetyczny wybiera do kolejnej iteracji najlepsze osobniki. Czyni on z nich dodatkowo rodziców nowej części populacji. Pozwala to na stopniowy wzrost średniej wartości przystosowania w populacji. Można śmiało powiedzieć, iż kolejne iteracje algorytmu genetycznego, wyraża-



Rys. 5.13: Element graficznego interfejsu użytkownika służący do porównywania dwóch taktów pod względem ich konturów.

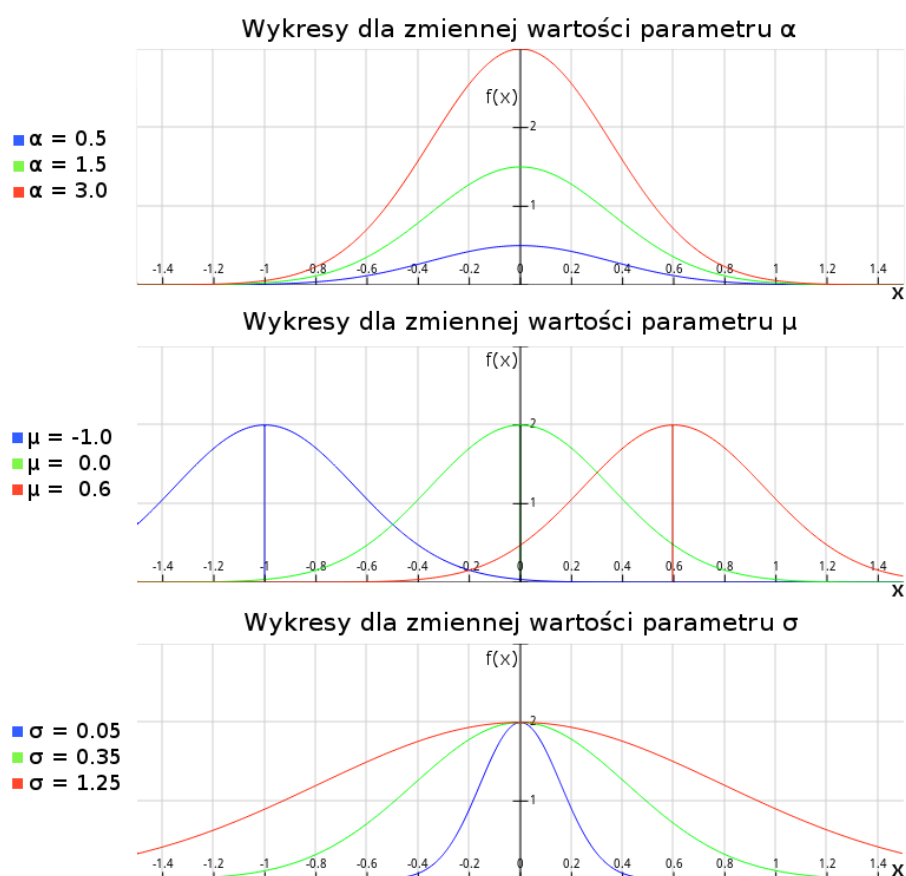
jąc to w średniej wartości przystosowania populacji, zmierzają w kierunku wskazywanym przez funkcje celu. Podobieństwo melodii może maksymalnie osiągnąć wartość równą 1. Oznacza ona, że dwa takty są identyczne. Specyfika algorytmu genetycznego sprawia, że, poprzez maksymalizację tej wartości, będzie on dążył, pomimo przeszkód w postaci krzyżowania i mutacji, do stworzenia osobników identycznych do tych z grupy referencyjnej. Można mu w tym przeszkodzić w dwojaki sposób – ograniczając liczbę iteracji lub definiując specyficzną funkcję celu. Niewielka liczba iteracji nie pozwala algorytmowi osiągnąć maksymalnej wartości przystosowania - ma on na to zbyt mało iteracji. Funkcja celu natomiast, jako argument, powinna przyjmować wartość podobieństwa i osiągać maksimum dla jej, wskazanego przez użytkownika, poziomu. Wraz ze zbliżaniem się do pożądanego stopnia podobieństw, a wartość funkcji celu powinna dynamicznie rosnąć. Dzięki funkcji

celu spełniającej podane kryteria, algorytm genetyczny nie będzie dążył do maksymalnej wartości podobieństwa, a jedynie do osiągnięcia jej pewnego progu. Przykładowo, dla 75% maksymalnej wartości podobieństwa, można mówić, iż melodie są jedynie inspirowane wzorcami – jest to jedno z naszych założeń odnośnie muzycznych fraz dostarczanych przez generator taktów. Progiem tym można manipulować otrzymując takty z większym lub mniejszym podobieństwem do wskazanej grupy.

System improwizujący, mając na uwadze powyższe spostrzeżenia, wymaga podania liczby iteracji algorytmu genetycznego oraz definiuje, bazującą na podobieństwie, funkcję celu. Funkcja ta, przyjmuje kształt zbliżony do rozkładu Gaussa, składa się jednak z dwóch ramion o różnych parametrach. Kształt bazowej – symetrycznej funkcji, o której mowa, opisany jest następującym wzorem:

$$f(x) = \alpha e^{-(x-\mu)^2/\sigma} \quad (5.1)$$

gdzie e oznacza stałą Eulera, a α , σ oraz μ są parametrami funkcji. Parametr α określa maksymalną wartość funkcji, μ położenie wierzchołka względem osi odciętych, a σ nachylenie ramion. Rysunek 5.14 prezentuje wpływ wymienionych parametrów na kształt opisanej funkcji.



Rys. 5.14: Wpływ parametrów α , σ oraz μ na kształt bazowej funkcji celu. Jeśli dany parametr nie jest podany, przyjmuje on wartość domyślną: $\alpha = 2$, $\mu = 0$ oraz $\sigma = 0.25$.

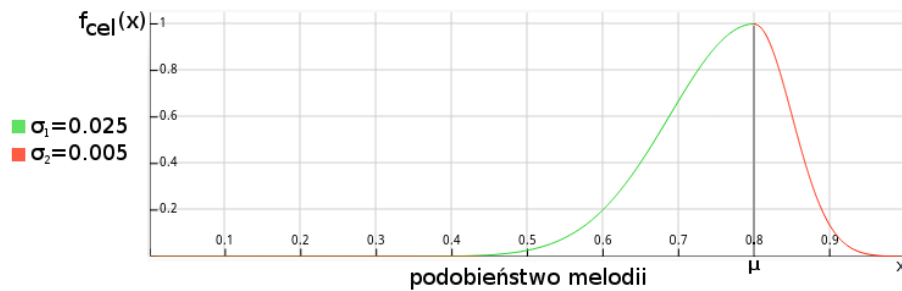
Zaimplementowana funkcja celu, podobnie jak znormalizowane podobieństwo melodii, przyjmuje wartości z przedziału $< 0; 1 >$. Wartość parametru α przyjmie zatem wartość 1. Argumentem funkcji będzie podobieństwo, a więc dobierając parametry σ oraz μ dbać należy aby w skrajnych punktach na osi odciętych (0 oraz 1), funkcja celu przyjmowała jak najmniejsze wartości. Odzwierciedla to mechanizm unikania zbyt wielkiego lub niedostatecznego podobieństwa osobników do wzorców. Najistotniejszym parametrem jest σ , który określa nachylenie ramion. Opisana funkcja jest symetryczna względem prostej prostopadłej do osi odciętych, przecinającej ją w punkcie μ . Nie odpowiada to do końca założonym oczekiwaniom. Przede wszystkim, algorytm genetyczny powinien unikać zbyt wielkiego podobieństwa osobników do wzorców. Nachylenie ramienia funkcji celu bliższego wartości 1 na osi odciętej powinno być większe niż nachylenie drugiego ramienia. Unika się tym samym tworzenia taktów zbyt podobnych do grupy referencyjnej, w mniejszym stopniu tolerując te podobne do wzorców. Ostatecznie funkcja celu $f_{cel}(x)$ została złożona z dwóch sparametryzowanych części opisanej funkcji bazowej i przyjęła postać:

$$f_{cel}(x) = e^{-(x-\mu)^2/\sigma}, \text{ gdzie} \quad (5.2)$$

$$\sigma = \sigma_1 \text{ dla } x \leq \mu$$

$$\sigma = \sigma_2 \text{ dla } x > \mu$$

Parametr α pominięto ponieważ przyjął on wartość 1. Rysunek 5.15 prezentuje przykładową postać funkcji celu. Osiąga ona maksimum dla wartości podobieństwa równej 0.8. Część czerwona, z parametrem $\sigma_2 = 0.005$, maleje szybciej, co utrudnia powstawanie taktów identycznych z grupą wzorców. Zielona część krzywej jest łagodniejsza ($\sigma_1 = 0.025$), a tym samym sprawia, że algorytm genetyczny daje szansę taktom, które są znacznie oddalone od maksimum (wartości $\mu = 0.8$).



Rys. 5.15: Ostateczny kształt funkcji celu $f_{cel}(x)$ dla parametrów $\mu = 0.8$, $\sigma_1 = 0.025$ oraz $\sigma_2 = 0.005$.

5.4.6. Mutacje

Celem mutacji jest wprowadzanie losowych zmian w obrębie osobnika. Zmiany takie mają zróżnicować populację, w której, zmodyfikowana w ten sposób struktura osobnika,

może wskazać nowy kierunek ewolucji. Etap mutacji następuje po krzyżowaniu, a prawdopodobieństwo jego zaistnienia dla konkretnego osobnika jest, podobnie jak w przyrodzie, znikome.

W zaimplementowanym systemie improwizującym, mutacje idą w parze z krzyżowaniem – są one celowym zabiegiem o niemałym prawdopodobieństwie zaistnienia. Mutowanie ma odzwierciedlić umiejętność ozdabiania melodii przez muzyka. Każda z tych operacji zaimplementowana jest jako klasa dziedzicząca po abstrakcyjnej klasie *AbstractMutation*. Klasa ta, wymaga określenia ile miejsca potrzebuje dany ozdobnik muzyczny. Informację tę może wykorzystać, przy aktywnym trybie dodawania ozdobników, jedynie jeśli jest dla nich miejsce w takcie. Ponadto, sama metoda mutacji *performMutation*, jako kontekst przekazuje dwie istotne informacje – jaka nuta poprzedza oraz następuje po obszarze przeznaczonym na mutację. Kontekst ten pozwala na tworzenie ozdobników, które w większym stopniu współgrają z modyfikowaną melodią.

Często spotykanym ornamentem muzycznym jest tzw. przednutka krótka górna (z wł. *acciaccatura*). Polega ona na dodaniu, przed głównym dźwiękiem, dowolnego, wyższego od niego, interwału. Interwałem tym jest najczęściej sekunda, a czas dodawanej nuty bardzo krótki. W przypadku implementowanego systemu, najkrótszą nutę definiuje określone odgórnie, najmniejsze okno czasowe w takcie. Sam ornament polega na dodaniu nuty o sekundę wyższej od ozdabianej. Czas jej trwania to jedno wspomniane okno czasowe. Przykład dodania przednutki krótkiej górnej ilustruje rysunek 5.16. Znaleźć na nim można zapis ozdobnika w standardowej notacji muzycznej oraz jego realizację w systemie.



Rys. 5.16: Działanie przednutki krótkiej górnej, notacja oraz jej realizacja w systemie.

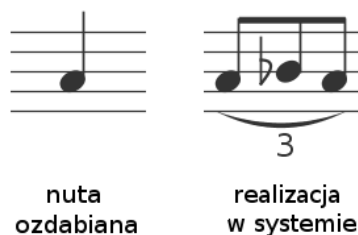
Kolejnym z wybranych do implementacji ozdobników jest mordent górny. Polega on na szybkim przejściu z nuty bazowej na sąsiadującą i powrocie na nutę początkową. Nuta sąsiadująca w przypadku górnego mordentu, to nuta odległa od bazowej o półton w górę. Przykład mutacji mordentowej wraz z jego notacją oraz realizacją w systemie znajduje się na rysunku 5.17.

Mutacją, która stara się wprowadzić zróżnicowanie rytmiczne do taktu, jest dodanie trioli bazującej na mordencie. Przed wybraną nutą dodawana jest triola złożona kolejno z: dźwięku bazowego, wyższego od bazowego o sekundę wielką oraz ponownie – bazowego. Po ostatnim elemencie trioli następuje przejście do dźwięku, na którym opiera się muta-



Rys. 5.17: Działanie mordentu górnego, notacja oraz jego realizacja w systemie.

cja – bazowego. Jest on przez to dodatkowo akcentowany. Przykład mutacji przedstawia rysunek 5.18.



Rys. 5.18: Działanie ozdobnika – trioli bazującej na mordencie.

Kolejnym ornamentem jest struktura muzyczna, która tworzy przejście pomiędzy dwiema nutami z użyciem ustalonej liczby równych interwałowo kroków. Przykład działania ozdobnika ilustruje rysunek 5.19.



Rys. 5.19: Działanie ozdobnika tworzącego przejście pomiędzy nutami dla jedno- oraz trójelementowego przejścia.

Zdarza się, iż po kilku iteracjach algorytmu genetycznego dochodzi do zjawiska powtarzania się jednej nuty w kilkuelementowych grupach. Wynika to z niefortunnego przebiegu krzyżowania, które, operując na zmutowanych taktach, łączy je w sposób niepożądany. Aby przeciwdziałać takim powtórzeniom dodano specjalny rodzaj mutacji, łączący kilka kolejnych wystąpień tej samej nuty w jej jednoelementowego reprezentanta.

5.4.7. Strategie selekcji

Istotnym etapem działania algorytmu ewolucyjnego jest wybór najlepszych osobników jako tych, które w niezmienionej formie przetrwają do następnego pokolenia oraz staną się rodzicami dla jego nowej części. Strategie selekcji, zwane mechanizmami próbkowania, opisują w jaki sposób dochodzi do kompletowania nowego pokolenia. Zaimplementowany algorytm genetyczny oferuje wybór spośród trzech najpopularniejszych metod selekcji.

Pierwszą z nich jest metoda rankingowa. Wstępnym etapem tego próbkowania jest stworzenie niemalejącego szeregu z osobników, biorąc pod uwagę wartość funkcji oceny jakie otrzymały. Określona część najlepszych rozwiązań staje się rodzicami i poprzez krzyżowanie zapełnia puste miejsce po usuniętych osobnikach.

Kolejną z metod próbkowania jest metoda ruletki. Rozpoczyna się ją od budowy wirtualnego „koła fortuny”, na którym, każdemu z osobników, przyznawany jest wycinek proporcjonalny do jego wartości przystosowania na tle wyników całej populacji. Następnie „kręci” się kołem określoną liczbę razy, losując grupę rodziców, którzy poprzez krzyżowanie generują pozostałą część populacji. Niewątpliwą wadą tej metody jest fakt, iż wraz ze wzrostem przystosowania osobników, ich wycinki na kole upodabniają się do siebie i sens losowania za pomocą ruletki, sprowadza się do losowego wyboru osobników z populacji.

Ostatnią z dostępnych metod jest selekcja turniejowa. W próbkowaniu tym, populację dzieli się na k -elementowe grupy, spośród których – jako rodzic – zostaje wybrany osobnik o wyższym dopasowaniu.

5.4.8. Warunek stopu algorytmu genetycznego

Algorytmy genetyczne służą do rozwiązywania konkretnych problemów. Niektóre z nich mają osiągnąć zaplanowany cel, zaś inne, takie jak np: optymalizacja, założonego *a priori* wyniku nie posiadają. Warunkiem zatrzymania pracy algorytmu jest najczęściej otrzymanie osobnika, posiadającego wymaganą wartość przystosowania. Równie często, szczególnie dla zadań optymalizacyjnych, określa się liczbę iteracji, po których algorytm powinien się zatrzymać. Choć wartość zastosowanej w systemie funkcji celu może być optymalizowana, to nietypowy charakter poszukiwanego przez nas rozwiązania, przemawia za zastosowaniem warunku stopu opierającego się na liczbie iteracji. Wynika to z faktu, iż nie poszukujemy jednego, najlepszego względem funkcji oceny taktu, a ich całej grupy. Funkcja celu powinna być jedynie pewnym drogowskazem podczas kompletowania kolejnej populacji. Ponadto, użycie świadomych muzycznie operatorów, pozwala nam liczyć na powstanie taktów poprawnych muzycznie. Innymi słowy: operacje krzyżowania i mutacji oraz użyta funkcja celu, gwarantują poprawne muzycznie odejście od taktów bazowych już w pierwszej iteracji; kolejne mogą zaowocować zwiększeniem średniego przystosowania osobników. Z kolei ich znaczna liczba, może doprowadzić do powstania taktów, które poprzez zbyt wiele modyfikacji zmniejszyły swoje przystosowanie, upodabniając się do siebie, a przez to zmniejszając drastycznie szanse na wzrost wartości funkcji oceny. Należy odejść od postrzegania zaimplementowanego algorytmu ewolucyjnego, jako elementu optymalizacyjnego. Chociaż wywodzi się on z rodziny tego typu rozwiązań, w tym przypadku, jest operacją mającą na celu kilkukrotne „przemieszanie” taktów, kierując się ich podobieństwem do wskazanych wzorców.

Generator taktów – jako element opisanej w niniejszej pracy architektury systemu – ma za zadanie dostarczenie grupy taktów, które posłużą jako główny materiał improwizacyjny. W tym celu został wykorzystany algorytm genetyczny, który ze względu na specyfikę zadania, odpowiednio zmodyfikowano. Populacja początkowa, o określonej przez użytkownika liczności, złożona jest z taktów przechowywanych w bazie zagrywek lub pliku podanym przez użytkownika. W kolejnych etapach, za pomocą przystosowanych do działania na taktach operatorów, algorytm genetyczny łączy ze sobą najlepsze osobniki, kierując się przy tym specjalnie zaprojektowaną funkcją celu. Funkcja ta ma za zadanie wskazywać takty, które w jak największym stopniu, bazują na wskazanej grupie fraz referencyjnych, a także są do nich podobne we określonym przez użytkownika stopniu. Algorytm wykonywany jest jedynie kilka razy. Wynika to z postrzegania algorytmu genetycznego nie jako mechanizmu optymalizacyjnego, a jako swoisty generator, łączący i modyfikujący populację początkową, kierujący się przy tym wzorcami czyli tematem improwizacji.

5.5. Etap kompozycji

Frazy muzyczne, które tworzone są przez generator taktów, trafić mogą do opcjonalnego etapu kompozycji. Ma on za zadanie stworzyć układ fraz, które, sprawiałyby wrażenie użycia pewnych technik kompozytorskich podczas naturalnego improwizowania. Zabieg taki zmniejsza stopień odczucia sztuczności utworu.

Proces kompozycji wymaga podania szablonu, według którego szereguje się poszczególne takty. Schemat ten jest ciągiem znaków, które oznaczają poszczególne takty, n.p.:

A B A C.

Opisana przez ten szablon kompozycja, składa się z trzech różnych taktów: A, B oraz C. Fraza A pojawia się w improwizacji dwukrotnie – jako pierwszy oraz trzeci takt. Przypomnieć tu należy, iż dla uzyskania najlepszego efektu, kompozycja taka powinna być skorelowana z progresją akordów.

W sytuacji, gdy wymagane jest stworzenie improwizacji złożonej z większej liczby taktów niż podany schemat kompozycji, system powiela szablon układu taktów tyle razy aby jego długość pokrywała się z wymaganą liczbą fraz.

5.6. Etap korekcji

W rozdziale dotyczącym historii dziedziny kompozycji wspomaganej komputerowo stwierdzono, że już Pitagoras zauważył podobieństwo pomiędzy muzyką a matematyką. Dziedzina sztuki, jaką jest muzyka, posiada szerokie zaplecze teoretyczne, dokładnie opisujące reguły nią rządzące. Z drugiej jednak strony, nie sposób zrezygnować z postrzegania muzyków, jako osób posiadających szczególny dar, nie wynikający bezpośrednio z wiedzy teoretycznej – wielu znakomitych artystów nie ma wykształcenia muzycznego.

W podobny sposób należy spojrzeć na ETAP KOREKCJI. Algorytm genetyczny, a w szczególności świadome muzycznie operatory, zaprojektowane są tak, aby tworzyć poprawne melodie, a źródło improwizacji, jakim jest baza zagrywek, również składa się z poprawnych taktów. Pomimo to, jak zostało już kilkakrotnie podkreślone, nie można zakładać, iż melodia otrzymana od GENERATORA TAKTÓW będzie, zgodnie dla ucha, współbrzmiała z podkładem muzycznym. ETAP KOREKCJI jest procesem, który ma za zadanie zmodyfikować improwizację w sposób, który zminimalizuje liczbę dysonansów w utworze.

W systemie zaimplementowano dwa rodzaje korekcji taktów. Oba z nich opierają się na prostym mechanizmie unikania dysonansów względem pewnego kontekstu muzycznego. W pierwszym przypadku kontekst ten jest szerszy – dotyczy tonacji utworu, w drugim – węższy i dotyczy pojedynczego akordu progresji.

5.6.1. Korekcja względem tonacji

Tonację można rozumieć, jako pewne zależności zachodzące pomiędzy stopniami gamy. Wynikają one z użytego w utworze systemu – zazwyczaj *dur* lub *mol*. Utwór muzyczny, który napisany jest zgodnie z daną tonacją – gamą – przejmuje od niej pewne cechy. Mówi się, że tonacja nadaje charakter utworowi.

Jedną z niezbędnych do rozpoczęcia procesu improwizacji zmiennych, jest tonika. Dźwięk ten jest referencyjną nutą, pozwalającą na ukonkretnienie progresji akordów. Wraz z progresją, można podać rodzaj skali muzycznej, która, również po ukonkretnieniu, stanie się tonacją utworu. O ile dla progresji, tonika, ma zauważalny wpływ na jej charakter, to dla tonacji, która również wynika z toniki, wpływ ten nie musi być odczuwalny. Wynika to z faktu, iż generator taktów nie kieruje się tonalnością podczas tworzenia linii melodycznej. Dopiero ETAP KOREKCJI, podczas poprawek może sugerować się wskazaną przez użytkownika tonacją (patrz: architektura systemu - rys. 4.1). Należy zaznaczyć, iż kierowanie się tonacją podczas poprawek w melodii improwizacji przynosi zadowalające efekty tylko w przypadku, gdy progresja akordów z nią współgra, w sensie harmonii muzycznej. Choć korekcja względem tonacji nie jest podstawowym sposobem poprawek melodii, jej opis na wstępie ułatwi zrozumienie głównego rodzaju korekcji – względem progresji akordów.

Mniej wyrafinowane utwory, o których powiedzieć można, że są utrzymane w określonej tonacji, składają się jedynie z dźwięków wchodzących w skład konkretnej gamy. W ten sposób cechy tonacji – jej charakter – przenoszone są na dany utwór. Proces korekcji, znając wchodzące w skład tonacji dźwięki, „podciąga” do nich nuty improwizacji leżące poza nią. Ponieważ znakomita większość skal muzycznych opiera się na interwałach nie większych niż sekunda wielka, również wspomniane „podciąganie” dźwięków nie będzie obejmowało odległości większych niż jeden ton (sekunda wielka). System, tuż przed przystąpieniem do korekty, tworzy tabelę poprawek. W jej skład wchodzi parę nut – wszystkie możliwe do reprezentacji w systemie, oraz powiązane z nimi poprawki. Korzystanie z takiej tablicy usprawnia działanie aplikacji pod względem szybkości. Jako przykład korek-

cji improwizacji względem tonacji posłużą dwa takty, poprawione będą względem gamy *d-dur*, co ilustruje rysunek 5.20.

Wszelkie poprawki bazowej improwizacji otrzymanej od GENERATORA TAKTÓW opierają się na podanej przez użytkownika tonacji – *d-dur*. Nuty, z oktawy małej, wchodzące w skład gamy *d-dur* przedstawia rysunek 5.20 (a). Wygenerowanie obiektu typu Scale (z ang. skala), który reprezentuje tę gamę w systemie, jest jednym z pierwszych etapów korekcji. Znając dozwolone w obrębie tonacji dźwięki, system przygotowuje tablicę, która posłuży do poprawek. W jej pierwszym wierszu znajdują się wszystkie możliwe do reprezentacji w systemie dźwięki – nazywane dalej bazowymi. Poniżej każdego z nich, w drugim wierszu, znaleźć można nuty, które poprawiają dźwięk bazowy na najbliższy, wyższy od danego. Tablica przedstawiona na rysunku 5.20 (b) zawiera elementy nadmiarowe – mapowania dźwięków, które już znajdują się w gamie *d-dur* (zaznaczone kolorem pomarańczowym). Aby nie przeprowadzać redundantnych operacji podczas poprawiania, zaznaczone na rysunku elementy tabeli, zostają usunięte, dzięki czemu poprawne, bazowe dźwięki pozostaną na swoim miejscu. Z kolei rysunek 5.20 (c) przedstawia dwa takty, względem których przeprowadzona zostanie korekcja. Każdy z dźwięków opisywanych fraz, sprawdzany jest pod względem występowania w pierwszym wierszu tabeli korekcji. Na rysunku 5.20 (d) zaznaczono nuty z przykładowych taktów, które kwalifikują się do poprawki – zostały znalezione w pierwszym wierszu tabeli. Dźwięki te zostały poprawione zgodnie z tablicą poprawek, czyli zgodnie ze schematem:

$$f^1 \rightarrow fis^1, ges^1 \rightarrow g^1, gis^1 \rightarrow a^1$$

gdzie $a \rightarrow b$ oznacza podmianę nuty *a* na nutę *b*. Wynik przeprowadzonych poprawek znaleźć można w części (e) rysunku 5.20. Podczas poprawek należy pamiętać, iż zaimplementowany system w specyficzny sposób traktuje znaki chromatyczne. Ich działanie nie obejmuje całego taktu, a jedynie nutę przy której się znajdują.

5.6.2. Korekcja względem progresji akordów

Drugim rodzajem poprawiania improwizacji, jest korekcja względem progresji akordów. Metoda ta daje najlepsze efekty, ponieważ każda poprawka wynika bezpośrednio z akordu towarzyszącego danemu taktowi, a współbrzmienie tych dwóch akordów jest źródłem dysonansów, których system stara się unikać. W tym przypadku, tablica poprawek jest przygotowywana, wielokrotnie – dla każdego z akordów, a nie jedynie dla tonacji. Uwzględnianie progresji akordów jest zatem lepszym sposobem podczas korekcji improwizacji. Wynika to również z faktu, iż podana przez użytkownika tonacja, która *de facto* nie ma wpływu na proces generacji taktów, nie musi wynikać z jego wiedzy o harmonii muzycznej, a więc może nie odpowiadać progresji akordów. Wpływa to jednak na pojawianie się dysonansów w improwizacji.



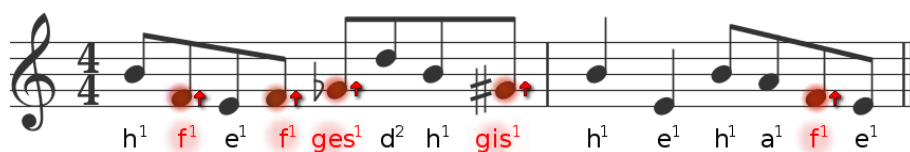
(a) Tonacja – gama *d-dur*, względem której dokonywane będą przykładowe poprawki.

Nuta bazowa	...	c¹	cis¹ des¹	d¹	dis¹ es¹	e¹ fes¹	f¹ eis¹	fis¹ ges¹	g¹	gis¹ as¹	a¹	ais¹ b¹	h¹ ces¹	c²	cis² des²	d²	...
Nuta poprawiona	...	cis¹	d¹	d¹	e¹	e¹	fis¹	fis¹	g¹	a¹	a¹	h¹	h¹	cis²	d²	d²	...

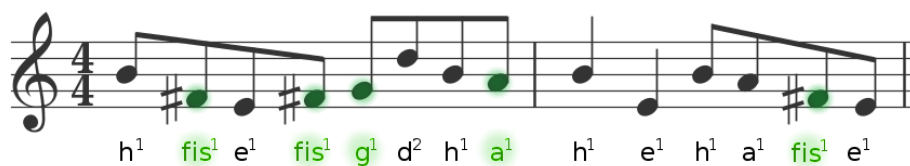
(b) Fragment tablicy poprawek stworzonej na podstawie gamy *d-dur*.



(c) Takty poddane korekcji względem tonacji *d-dur*.



(d) Dźwięki skorygowane względem tonacji *d-dur*.



(e) Poprawione takty.

Rys. 5.20: Proces korekcji dwóch taktów względem tonacji *d-dur*.

Tablica poprawek, w przypadku tonacji, budowana jest bezpośrednio z jej dźwięków, zaś w przypadku akordów, proces ten jest bardziej skomplikowany. Akord budowany jest na podstawie konkretnej gamy, z której bierze się odpowiednią ilość nieparzystych stopni (patrz: rysunek 3.4). Na podstawie rzeczonych gam, dla każdego akordu, budowana jest tablica poprawek, która służy do korekcji nut w obrębie określonego taktu. Ponieważ akordy wynikają bezpośrednio z typów skal muzycznych, każdy z nich można powiązać z odpowiadającą mu skalą (tabela 5.1). Zauważyć należy, że nawet po poprawkach taktu względem gamy, z której wywodzi się akord jemu towarzyszący, nie można całkowicie wyeliminować dysonansów. Wynika to ze zjawiska występowania współbrzmień odległych o jeden stopień od składowych akordu, a jak wiemy, w większość skal, odległość pomiędzy kolejnymi stopniami jest sekundą wielką lub małą. Oba te interwały należą do dysonansów. Wyeliminowanie spośród uzyskanych dźwięków tych, które tworzą dysonanse z akordem progresji, zmniejszyłoby liczbę dozwolonych dźwięków do 4 lub 5.

Tabela 5.1: Zestawienie przykładowych, użytych w systemie akordów, skal muzycznych, z których się wywodzą oraz stopni skal, których należy unikać podczas improwizacji wokół akordu.

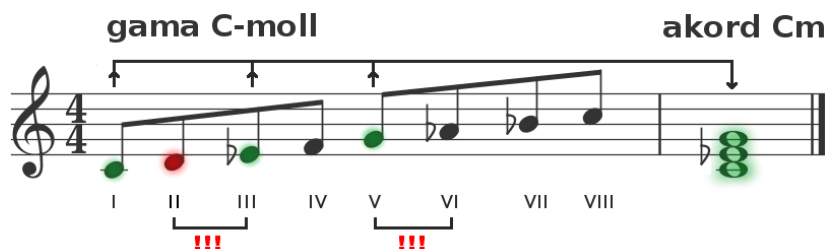
Rodzaj akordu [przyrostek]	Skala muzyczna	Unikane stopnie skali
durowy [<i>brak</i>]	skala durowa	IV
molowy [m]	skala molowa	II
zwiększony [+]	skala zwiększona	<i>brak</i>
zmniejszony [o]	skala zmniejszona	II i IV
septymowy durowy [7]	skala durowa	IV

Improwizacja, która składałaby się z tak niewielkiej liczby dźwięków byłaby zbyt jednolita. Aby skutecznie eliminować dysonanse oraz nie uszczuplać przy tym w nadmierny sposób liczby dozwolonych dźwięków, każdy typ akordów w systemie, definiuje stopnie skali, które należy usunąć z listy dźwięków dostępnych dla improwizacji. Na listach tych znajdują się zazwyczaj stopnie gamy, które wraz z dźwiękami akordu tworzą sekundy małe – są one interwałami w większym stopniu dysonansowymi, niż sekundy wielkie. Pozostałe „dysonansowe” stopnie skali, mogą występować w improwizacji, ponieważ nie są tak rażące jak interwały sekundy małej. Za przykład posłuży akord *c-moll* (Cm), względem którego dokonana została poprawka jednego taktu. Najistotniejsze elementy/etapy korekty ilustruje rysunek 5.21. Ilustracja – 5.21 (a) – prezentuje gamę *c-moll* oraz akord Cm, który składa się z jej I, III oraz V stopnia. Na rysunku zaznaczono również występujące w obrębie gamy interwały sekund małych, które, jak wspomniano, są odległościami szczególnie dysonansowymi. Obie te odległości występują w bezpośrednim sąsiedztwie dźwięków wchodzących w skład akordu *c-moll*. Systemowa definicja akordu typu *moll* wskazuje II stopień skali jako ten, którego należy unikać. Został on wybrany ponieważ tworzy dysonans z I stopniem skali – toniką, która jest dźwiękiem bazowym tonacji oraz progresji. Istotnym jest aby dźwięk ten brzmiał w improwizacji w możliwie wyraźny i „czysty” sposób, nie niszcząc tym samym naturalnej harmonii progresji, w której tonika odgrywa ważną rolę. W tym przypadku w obrębie taktu, na który przypadł akord Cm, obowiązuje tabela poprawek zaprezentowana w części (b) rysunku 5.21. Zbudowana ona jest podobnie jak w poprzednim przykładzie – z dźwięków gamy *c-moll*, z której wywodzi się akord Cm. Czerwonym kolorem zaznaczony jest, wykluczony przez definicję akordu molowego, dźwięk d^1 , który poprawiony został na kolejny, dozwolony dla akordu Cm, dźwięk – es^1 . Tablicą poprawek dla danego akordu należy posługiwać się w ten sam sposób, jak podczas korekty względem tonacji. Jediną różnicą jest fakt, iż każdy takt zawiera poprawki specyficzne dla akordu, z którym współbrzmi. Rysunek 5.21 (c) obrazuje takt bazowy – poprawiany, oraz frazę po korekcie z użyciem tabeli poprawek. Czerwony kolor wskazuje nuty spoza dozwolonych, zielony – ich poprawione odpowiedniki. Zastosowano

następujące korekty:

$$fis^1 \rightarrow g^1, e^1 \rightarrow f^1, a^1 \rightarrow b^1$$

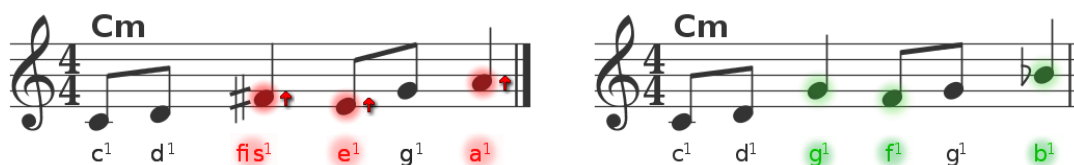
gdzie $a \rightarrow b$ oznacza podmianę nuty a na nutę b .



(a) Gama *c-moll* oraz zbudowany na jej podstawie akord molowy *c-moll* (Cm). Czerwone wykrzykniki wskazują interwały sekund małych pomiędzy stopniami gamy.

Nuta bazowa	...	c^1	cis^1 des^1	d^1 dis^1	es^1 dis^1	e^1 fes^1	f^1 eis^1	fis^1 ges^1	g^1 gis^1	gis^1 as^1	a^1	ais^1 b^1	h^1 ces^1	c^2	...
Nuta poprawiona	...	c^1	d^1	d^1	es^1	f^1	f^1	g^1	g^1	as^1	b^1	b^1	c^2	c^2	...

(b) Fragment tablicy poprawek stworzonej na podstawie gamy *c-moll*. Czerwona ramka wskazuje dodatkowo wykluczony przez akord dźwięk szczególnie dysonansowy – d^1 , poprawiany na dźwięk es^1 .



(c) Takt poddany korekcji – po lewej oraz wynik poprawek – po prawej.

Rys. 5.21: Proces korekcji taktu względem towarzyszącego mu akordu *c-moll* (Cm).

Korekcja względem progresji akordów jest metodą poprawek znacznie efektywniejszą. Wynika to z faktu, iż kontekst, który brany jest tu pod uwagę, uwzględnia główne źródło dysonansów - akord. W przypadku poprawek względem tonacji, korekcja odbywa się jedynie względem konkretnej gamy. Brak dysonansów wynika z założenia, że progresja akordów przystosowana jest do odpowiedniego współbrzmienia ze wskazaną tonacją. Niestety, założenie to, jest w większości przypadków błędne, a ponadto, progresja idealnie współbrzmiąca z daną tonacją, jest najczęściej nieciekawa muzycznie.

Wadą korekcji improwizacji względem progresji akordów, jest ciągła zmiana tonacji, względem której poprawiane są kolejne takty. Daje to w wyniku melodie, które nie zawsze utrzymane są w jednej tonacji, choć mogą posiadać wspólny system – np. gdy wszystkie akordy progresji są durowe lub molowe. Charakter utworu, w przypadku korekcji względem poszczególnych akordów, nadawany jest przez progresję, która jako tło muzyczne prowadzi główną linię melodyczną. W większości przypadków progresje akordów zgodne są z zasadami harmonii, co, poprzez poprawki, przenosi się na improwizację i w pozytywny

sposób wpływa na odbiór wygenerowanego utworu. Zjawisko to równoważy wspomnianą powyżej wadę korekcji tego typu.

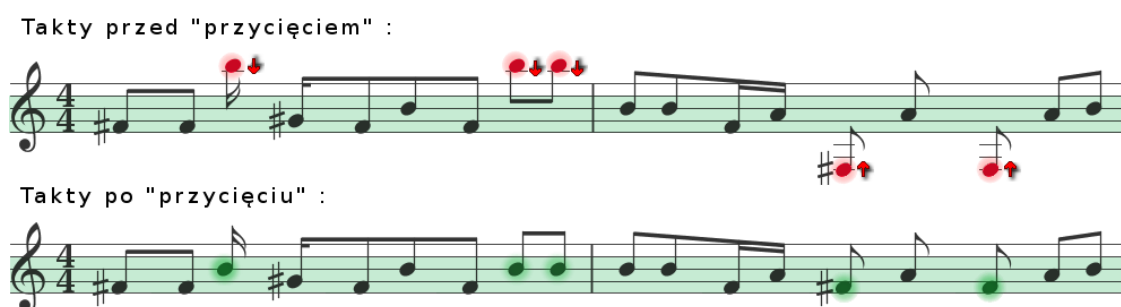
5.7. Etap przetwarzania końcowego

Improwizacja, poddana korekcji, jest w efekcie utworem gotowym do zaprezentowania użytkownikowi. Innymi słowy – system zakończył już główny etap działania – stworzył improwizację solo, współbrzmiającą w odpowiedni sposób z zadaniem podkładem muzycznym. Architektura systemu uwzględnia jednak dodatkowy etap, a mianowicie – PRZETWARZANIE KOŃCOWE. Podczas jego trwania, możliwe jest przeprowadzenie dodatkowych operacji wykonywanych na improwizacji. Mają one na celu jej ulepszenie, bez ingerencji w melodię, która mogłaby mieć znaczący wpływ na jej charakter lub byłaby sprzeczna z poprawkami dokonanymi podczas ETAPU KOREKCJI. Opisane poniżej operacje wchodzi w skład ETAPU PRZETWARZANIA KOŃCOWEGO.

5.7.1. Przycinanie improwizacji

W sekcji poświęconej ETAPOWI KOREKCJI, przedstawione zostały fragmenty tabel poprawek. Kompletne tablice obejmują trzy oktawy, których reprezentację umożliwia system. Poprawki dla pozostałych, nieprzedstawionych w przykładach nut, bazują na tonacjach, które zwyczajowo prezentuje się w obrębie oktawy razkreślnej. Pominięte w ilustracjach korekty, są analogiczne do tych w obrębie oktawy razkreślnej, różnica polega na ich przeniesieniu do odpowiedniej oktawy. I tak – jeśli w oktawie razkreślnej poprawka nakazuje podmianę dźwięku ges^1 na g^1 , to dźwięk ges z oktawy dwukreślnej, czyli ges^2 , poprawiany jest na dźwięk g również z oktawy dwukreślnej czyli g^2 . Improwizacja po korekcji jest więc poprawna w obszarze należącym do każdej z uwzględnionych w systemie oktaw. Jaki ma to wpływ na improwizację? Poprawność dla każdego możliwego dźwięku nie wymusza na GENERATORZE TAKTÓW ani na etapie korekcji trzymania się danej oktawy, czy też wyznaczonego zakresu wysokości dźwięków. Nuty improwizacji mogą obejmować swoją rozpiętością kilka oktaw i, jak zostało powiedziane, każda z tych nut będzie odpowiednio poprawiona. Jednak zbytnia rozpiętość wysokości dźwięków w utworze nie jest zjawiskiem pożądanym, szczególnie jeśli interwały pomiędzy kolejnymi dźwiękami są zbyt wielkie – melodia traci w ten sposób płynność oraz spójność. Niestety, opisany w jednym z poprzednich rozdziałów, mechanizm krzyżowania taktów, może sprzyjać występowaniu tego typu zjawisk w miejscach łączenia się melodii pochodzących z dwóch różnych fraz. Operacja przycinania ma za zadanie sprowadzenie dźwięków, które wychodzą poza ustaloną rozpiętość, do odpowiedniego obszaru melodii. Jedynym parametrem tej metody jest liczba półtonów, które, liczone od dolnej granicy – toniki, wyznaczają górny limit wysokości dźwięku. Dźwięk, który znajduje się poza wyznaczonym w ten sposób obszarem, jest obniżany lub podwyższany o odległość oktawy czystej, w sposób, który umieści go we wspomnianym obszarze lub zmniejszy jego odległość od

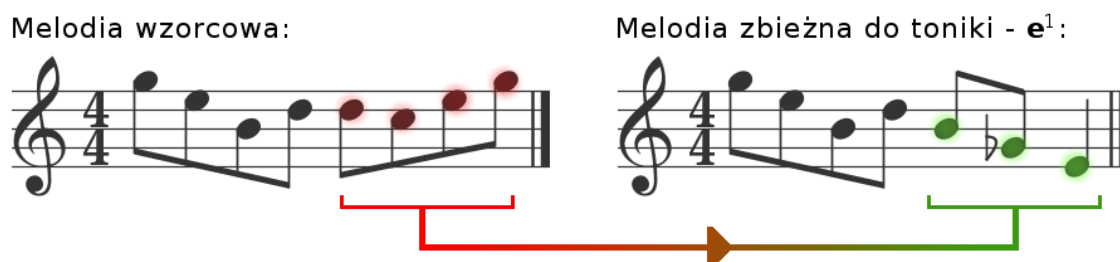
niego. Użycie oktawy czystej wynika z założenia o nie ingerowaniu w znaczny sposób w melodie utworu – dźwięk przeniesiony o oktawę jest tym samym dźwiękiem, znajdującym się jednak w innej oktawie. Przykład przycinania ilustruje rys. 5.22. Wskazane kolorem czerwonym nuty, zostały przeniesione o interwał oktawy w górę lub w dół, aby znaleźć się w obszarze przeciętnej wysokości dźwięków danej melodii, oznaczonym kolorem niebieskim.



Rys. 5.22: Operacja przycinania melodii w ramach PRZETWARZANIA KOŃCOWEGO.

5.7.2. Zbieżność głównej linii melodycznej do toniki

Podczas precyzowania pojęcia progresji akordów, zauważono, iż występują one zazwyczaj w postaci szablonów, które związane są zarówno z dziedziną harmonii, jak i gatunkiem muzycznym, do którego należą. Progresje takie, składają się z określonej liczby taktów, z których ostatni, lub kilka ostatnich, w sposób wynikający z zasad harmonii, zbiegają się do akordu w tzw. funkcji toniki (akordu opartego o pierwszy dźwięk tonacji). Zabieg taki sprawia, iż podkład kończy się w naturalny sposób. Poprawki improwizacji względem akordów sprawiają, iż melodia również przechodzi do tonacji zgodnej z ostatnim akordem. Niestety, melodia nie zakończy się na dźwięku toniki; stać się może tak jedynie za sprawą przypadku. Aby utwór kończył się w naturalny sposób, należy poprawić końcówkę ostatniego taktu w taki sposób, aby melodia na kilku ostatnich dźwiękach zbliżała się do toniki i, finalnie, zakończyła się na niej. Ostatni dźwięk nie powinien również należeć do najkrótszych – dłuższa nuta spotęguje wrażenie naturalnego zakończenia utworu. Rysunek 5.23 prezentuje przykład poprawki ostatniego taktu tak, aby jego melodia zbiegała się do dźwięku toniki (dźwięk e^1). Omawiany tu proces modyfikuje połowę ostatniego taktu (kolor zielony), wprowadzając w tym obszarze pasaż od ostatniej nuty w pierwszej połowie frazy (d^2) do toniki. Pasaż ten (kolor czerwony), składa się z dwóch nut o domyślnej długości (ósemki) oraz ostatniej dłuższej nuty – ćwierćnutowej toniki. Akord progresji przypadający na ten takt jest nieistotny – ważne jest jedynie to, aby zbudowany był w oparciu o tonikę.



Rys. 5.23: Operacja kończenia improwizacji poprzez zbieg melodii do toniki (e^1).

5.8. Odtwarzanie oraz zapis improwizacji

Improwizacja wygenerowana przez system, jest w nim reprezentowana jako obiekt typu ABCTune. Przechowuje on m.in. nagłówki pliku ABC, ciało improwizacji oraz progresję akordów. Ponadto, zawiera on informacje dotyczące rodzaju instrumentów, akompaniamentu, poziomów głośności, mających wpływ na odtwarzanie improwizacji. Informacje te mogą zostać połączone ze sobą, tworząc gotowy do zapisu plik notacji ABC. Dokonać tego można, przekazując obiekt ABCTune do metody buildTune(ABCTune abcTune) klasy ABCTuneFileBuilder, która realizuje jeden ze wzorców kreacyjnych – zwanego budowniczym. Zapisany plik w notacji ABC, jest wyjściowym elementem dla dedykowanych dla notacji ABC aplikacji, umożliwiających zapis improwizacji w kilku różnych formatach.

Jednym z wariantów zapisu improwizacji, jest jej zapis w postaci notacji muzycznej do pliku PDF. Możliwe jest to dzięki dwóm programom:

- ABCM2PS, który konwertuje plik ABC do języka opisu stron PostScript oraz
- GHOSTSCRIPT (GS/GSWIN), który umożliwia stworzenie pliku PDF na podstawie opisu w języku PostScript.

Plik PDF jest wygodnym w przeglądaniu, gotowym do druku dokumentem.

Aplikacja umożliwia również zapis improwizacji w postaci pliku dźwiękowego MIDI. Służy do tego program ABC2MIDI, który na podstawie pliku ABC tworzy plik dźwiękowy MIDI. Program ABC2MIDI umożliwia zdefiniowanie rodzaju instrumentów dla głównej linii melodycznej oraz podkładu wraz z określeniem poziomu głośności każdego z nich. Pozwala również na wskazanie rodzaju akompaniamentu podkładu muzycznego. Ponieważ informacje takie nie należą do standardu notacji ABC, w pliku typu .ABC należy dodać specjalne nagłówki, które są pomijane przez aplikacje do obróbki plików ABC, lecz rozpoznawalne przez program ABC2MIDI. Każdy nagłówek znajduje się w nowej linii i rozpoczyna się ciągiem %%MIDI, po którym następuje nazwa parametru oraz jego argumenty – każdy oddzielony spacją. Przykładowe nagłówki %%MIDI program 65, %%MIDI chordprog 32 oraz %%MIDI bassprog 32, określają rodzaje instrumentów, odgrywających odpowiednio – główną linię melodyczną, akordy występujące w progresji, linię basową w progresji, o ile rodzaj akompaniamentu taką uwzględni. Po każdym z nagłówków powinien następować numer tzw. programu, czyli instrumentu, który został wybrany do odegrania wskazanej

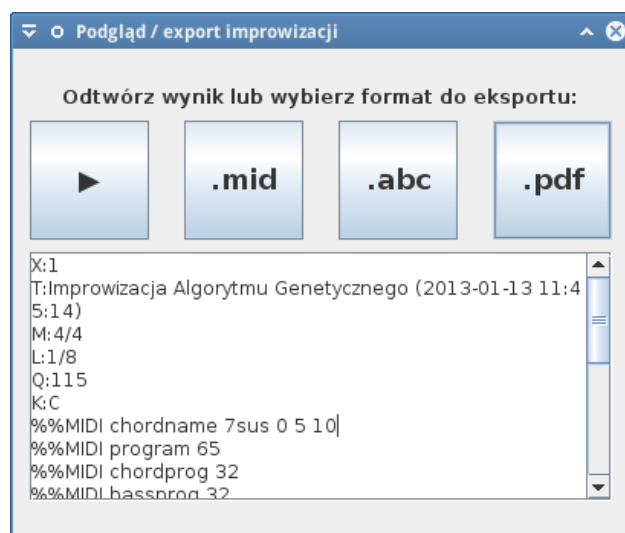
warstwy improwizacji. Lista możliwych do wyboru instrumentów zawiera ponad 120 pozycji, wśród których znaleźć można zarówno instrumenty muzyczne, takie jak: fortepian, akordeon, gitara akustyczna, saksofon altowy, jak i efekty dźwiękowe – aplauz, wystrzał broni palnej, itp. System improwizujący, w typie enumeracyjnym Programs uwzględnił kilkanaście z nich. Dwa kolejne nagłówki: `%%MIDI chordvol 55` oraz `%%MIDI bassvol 60`, określają głośność, kolejno, dla akordów progresji oraz linii basowej, o ile występuje. Parametry liczbowe tych nagłówków, określają procentowo głośność danego elementu improwizacji, względem głównej linii melodycznej, której poziom głośności domyślnie przyjmuje wartość 100. Jednym z dodatkowych nagłówków jest `%%propagate-accidentals not`, który sprawia, że program ABC2MIDI podczas generowania pliku MIDI nie propaguje znaków chromatycznych na cały takt – stosuje je jedynie do nut, przy których owe znaki się znajdują. Listę nagłówków MIDI zamykają `%%MIDI chordname` oraz `%%MIDI gchord`. Pierwszy z nich pozwala na zdefiniowanie nowego rodzaju akordu, drugi natomiast, określa rodzaj akompaniamentu. Oba zostały dokładniej omówione w poprzednich rozdziałach.

Reasumując, użytkownik, po zakończeniu procesu generowania improwizacji ma możliwość utrwalenia jej w postaci:

- pliku typu ABC,
- zapisu notacji muzycznej w postaci dokumentu PDF,
- pliku dźwiękowego MIDI.

Ponadto, aplikacja pozwala na odtworzenie improwizacji bez uprzedniego jej zapisu. Umożliwia to klasa natywnego odtwarzacza NativeMIDIPlayer.

Po zakończeniu procesu generowania improwizacji, w przykładowej aplikacji z graficznym interfejsem użytkownika, wyświetlane jest okno z podglądem muzycznej notacji głównej linii melodycznej oraz okno przedstawione na rysunku 5.24, pozwalające na odtworzenie improwizacji lub jej zapisu do wyżej wymienionych formatów. Zawiera ono dodatkowo pole tekstowe z improwizacją w postaci notacji ABCNOTATION.



Rys. 5.24: Okno odtwarzania oraz eksportu wygenerowanej improwizacji.

6. WYNIKI DZIAŁANIA METODY ORAZ ICH OCENA

6.1. Próba obiektywnej oceny generowanych improwizacji

Przed przystąpieniem do omówienia zakresu zagadnień uwzględnionych w tytule niniejszego rozdziału, można by pokusić się o refleksję ogólniejszej natury. A mianowicie, należało by zastanowić się nad możliwością obiektywnej oceny wytworów jakiejkolwiek dziedziny sztuki. Jak powszechnie wiadomo, nawet profesjonaliści, z dużą wiedzą teoretyczną i doświadczeniem, różnią się w ocenach co do estetyki i szeroko pojętej wartości wytworów sztuki. Trzeba również pamiętać, że artyści coraz częściej wykorzystują najnowsze technologie, których elementy stają się integralną częścią naturalnego procesu twórczego. To z pewnością utrudnia jeszcze bardziej obiektywną, krytyczną analizę i ocenę tego rodzaju dzieł.

Wracając do głównego tematu tej części pracy, trzeba podkreślić, że generowane przez zbudowany system utwory muzyczne, nie wymagają profesjonalnej oceny znawców przedmiotu. Chodzi tu, przede wszystkim, o wrażenia estetyczne przeciętnego odbiorcy muzyki użytkowej. Dla porządku przypomnijmy raz jeszcze, że wyniki pracy zaprojektowanego systemu, to nic innego, jak konkretne utwory muzyczne – w tym przypadku – improwizacje. Niezwykle istotnym dla możliwości praktycznego wykorzystania rzeczonych utworów, ma ich ocena pod względem wartości muzycznej dla przeciętnego człowieka, czyli potencjalnego ich odbiorcy.

Aby zweryfikować przydatność tworzonych przez system improwizacji i zobiektywizować ich ocenę pod względem czysto estetycznym, posłużono się tu statystycznym narzędziem badawczym, jakim jest ankieta. W ankiecie brały udział 23 osoby. Większość ankietowanych to studenci w wieku od 20 do 24 lat, a także 3 osoby 50-ladne oraz jedna osoba w wieku 74 lat. Aby ułatwić ankietowanym zadanie, czyli ocenę, w pierwszym etapie odtworzono im wygenerowaną przez system improwizację w kontraście do kakofonicznej, rozstrojonej melodii, złożonej z fragmentów różnorodnych utworów muzycznych, odegranych na tle popularnej bluesowej progresji akordów. Zestaw odpowiedzi na pierwsze pytanie "Która z kompozycji, w Pani/Pana odczuciu, jest lepsza?" wyglądał następująco:

- improwizacja była znacznie lepsza,
- improwizacja była lepsza,
- oba utwory reprezentują ten sam poziom,
- kakofonia była lepsza,
- kakofonia była znacznie lepsza.

Przy czym zaznaczyć należy, iż mogące sugerować oceny określenia „improwizacja” oraz

Tabela 6.1: Ocena jakości wygenerowanej przez system improwizacji muzycznej w porównaniu z dysharmonicznym utworem referencyjnym (kakofonią).

Odpowiedź	Liczba głosów
Improwizacja była dużo lepsza.	8
Improwizacja była lepsza.	14
Oba utwory reprezentują ten sam poziom.	1
Kakofonia była lepsza.	0
Kakofonia była znacznie lepsza.	0

„kakofonia” nie były użyte w odpowiedziach – zastąpiono je słowami „pierwszy utwór” oraz „drugi utwór”. Ponadto, kolejność odtwarzanych słuchaczom melodii była różna – część osób, jako pierwszą usłyszała improwizację, pozostałym zaprezentowano najpierw kompozycję dysharmoniczną.

Pierwszy etap badania miał na celu ocenę jakości improwizacji w kontraście do nieprzyjemnej dla ludzkiego ucha dysonansowej kompozycji. Nie pozwala to, rzecz jasna, na niezależną obiektywną ocenę wygenerowanej przez aplikację melodii. Dlatego też improwizacja odtwarzana była ankietowanym jeszcze raz, po czym poproszono o ponowną ocenę utworu w skali od 1 do 5.

Przed przystąpieniem do analizy wyników ankiety, zaleca się odsłuchanie melodii, które były prezentowane ankietowanym¹. Wśród plików znajduje się wspomniana kompozycja dysharmoniczna. Złożona ona jest z różnych, lekko zmodyfikowanych, zagrywek bluesowych, które nie starają się współgrać z podkładem. Rozstrojony utwór to plik o nazwie 2.Przykład.mp3, który w dalszej części rozdziału nazywany będzie kakofonią.

Zestawienia odpowiedzi osób ankietowanych znajdują się w tabelach 6.1 i 6.2. Tabele wyraźnie pokazują, iż system generuje harmoniczne improwizacje, które są pozytywnie odbierane przez większość ankietowanych (średnia ocena ≈ 3 ; odch. stand. ≈ 1). Zaznaczyć należy, iż uczestnikom ankiety odgrywane były pliki w formacie MIDI, który nie oferuje barw dźwięków najwyższej jakości. Najważniejszym jednak spostrzeżeniem, jest fakt, iż system z powodzeniem generuje poprawne, przyjemne dla ludzkiego ucha improwizacje. Można zatem stwierdzić, że realizuje on swoje założone na wstępie zadanie – improwizowanie solo do zadanego podkładu muzycznego. Należy zdawać sobie sprawę, iż melodie prezentowane ankietowanym, wygenerowane zostały z użyciem domyślnych parametrów improwizacji. Ponieważ metoda improwizacji jest prototypem, pozostaje wiele obszarów, które pozwalają na konfigurację i rozbudowę większości elementów systemu. Więcej informacji na temat potencjału, który tkwi z systemie, a który wydobyć można za pomocą badań i konfiguracji, znajduje się w dalszej części rozdziału.

Choć wyniki omówionej ankiety są zadowalające, zaznaczyć należy, iż była ona prze-

¹ Znaleźć je można w folderze 1.Przykłady_improwizacji znajdującym się wśród plików dołączonych do pracy.

Tabela 6.2: Ocena improwizacji po ponownym jej odsłuchaniu, już bez dysharmonicznego odnośnika.

Ocena	Liczba głosów
5	2
4	5
3	9
2	6
1	1

prowadzana na podstawie pierwszych wyników działania zaprojektowanej metody improwizowania. Co więcej, ze względu na zestawienie improwizacji z kakofonią, stwierdzić możemy jedynie, iż generowane melodie są bardziej akceptowalne dla ludzkiego ucha niż dysharmoniczne utwory.

Aby uwiarygodnić i zobiektywizować maksymalnie ocenę wartości estetycznej wygenerowanych przez system improwizacji, przeprowadzono drugi etap badań ankietowych. Postawiono pytanie czy opracowana metoda jest w stanie wygenerować utwory, które mogą konkurować z bluesowymi melodiami. W tym celu, posługując się jednym z charakterystycznych dla tzw. *delta blues*, utworem skomponowanym przez bluesmana Toby'ego Wine'a², ponownie przeprowadzono ankietę, w której tym razem proszono o próbę wskazania, która z dwóch melodii³ jest dziełem człowieka, a którą wygenerowano za pomocą komputera. Podobnie jak w poprzednim przypadku, posłużono się nie jedną, lecz dwiema przykładowymi improwizacjami stworzonymi za pomocą zaprojektowanej metody. Jedna z nich zestawiana była, w losowej kolejności, z bluesem Toby'iego Wine'a. Wspólnymi elementami melodii była progresja akordów, a co za tym idzie – liczba taktów. Zdecydowano się również na ten sam typ akompaniamentu. Ankieta przeprowadzona została na grupie 15 osób, reprezentujących przekrój wiekowy podobny, jak w poprzedniej grupie ankietowanych. Tabela 6.3 zawiera wyniki przeprowadzonej ankiety. Każdy z ankietowanych, po wskazaniu melodii, którą uważa za dzieło ludzkiego talentu, proszony był dodatkowo o ocenę stopnia trudności w podjęciu decyzji, czyli w rozróżnieniu źródła utworów. Do wyboru były trzy odpowiedzi:

- „Nie miałem/am problemów z podjęciem decyzji – wyraźnie słyszeć różnicę w poziomach utworów.” zwaną dalej **bezproblemową decyzją**,
- „Miałem/am problemy z podjęciem decyzji; utwory reprezentują zbliżony poziom.” zwaną dalej **problemem z decyzją**,
- „Nie słyszeć różnicy w jakości utworów; wybór był losowy (tzw. *strzał*).” zwaną dalej **brakiem różnicy**.

Wyniki ankiety są niezwykle interesujące. Po pierwsze, aż siedem osób ($\approx 47\%$) wska-

² Toby Wine – nowojorski gitarzysta, kompozytor, aranżer oraz nauczyciel; absolwent Akademii Muzycznej „Manhattan”.

³ Melodie użyte w ankiecie znaleźć można w folderze 3_II_Ankieta wśród plików dołączonych do pracy.

Tabela 6.3: Wyniki drugiej ankiety.

Odpowiedź	Poprawne wskazanie ludzkiej improwizacji	Wskazanie komputerowej improwizacji jako ludzkiej	Suma głosów
Bezproblemowa decyzja	4	3	7
Problem z decyzją	2	2	4
Brak różnic	2	2	4
Suma głosów:	8	7	

zało improwizację wygenerowaną przez zaprojektowaną metodę jako dzieło człowieka. Pozwala to na stwierdzenie, iż generowane melodie mogą stawać w szranki z utworami skomponowanymi przez człowieka. Ponadto, warto zwrócić uwagę, że nawet osoby, które były pewne podjęcia swej decyzji (7 osób), aż w trzech przypadkach uznały improwizację komputerową za dzieło człowieka. Choć wyniki są niezwykle obiecujące, należy wskazać kilka czynników, które mogły wpłynąć na taką właśnie decyzję ankietowanych. A mianowicie: oba, prezentowane ankietowanym utwory, były plikami MIDI, które, z racji użycia dosyć prymitywnych barw instrumentów i nierozbudowanego akompaniamentu, sprowadziły obie melodie do podobnego poziomu pod względem aranżacji. Choć, z jednej strony, pozwala to na sprawiedliwą ocenę obu utworów, z drugiej, uznana kompozycja ludzka, może wiele stracić podczas odgrywania jej za pomocą uproszczonych barw dźwięków. Istotnym czynnikiem jest również fakt, że utwór Toby'iego Wine'a jest improwizacją, która nie posiada kompozycji w sensie, w którym rozumiana jest ona w obrębie tej pracy. Kolejne frazy, w rzeczonym utworze, mają wspólną tonację oraz temat; nie są one jednak ułożone w sposób, który zastosowany jest w tworzeniu improwizacji przez zbudowany system. Większość ludzi przyzwyczajona jest do typowej budowy utworów, która uwzględnia powtórzenia, a tym samym posiada pewien rytm. Mogło mieć to wpływ na wskazywanie melodii muzyka bluesowego jako tej sztucznej – posiadającej nietypową strukturę. Ponadto trzeba pamiętać, iż grupa osób ankietowanych złożona z przeciętnych, nieosłuchanych odbiorców muzyki. To również mogło mieć niebagatelny wpływ na wyniki ankiety.

Należy stwierdzić, że wynik ankiety, jest miarodajny w odniesieniu do przeciętnego odbiorcy, który w większości przypadków uznał improwizacje wygenerowane za pomocą zaprojektowanej metody za równie dobre jak stworzone przez muzyka.

6.2. Analiza wybranej improwizacji

Poniżej opisana jest jedna z wygenerowanych improwizacji. Wskazano również istotne parametry i zmienne, które użyte były do jej wygenerowania. Uruchomienie metody odbyło się za pośrednictwem graficznego interfejsu użytkownika, opisanego w dodatku A.

Głównymi zmiennymi przekazywanymi przez użytkownika do aplikacji są: tonika wraz ze skalą muzyczną oraz szablon progresji akordów. Ponieważ zastosowaną korekcją była korekcja względem akordów, podana przez użytkownika skala nie była brana pod uwagę podczas tworzenia improwizacji; pominięto ją również w dalszej części sekcji. Wskazaną toniką był dźwięk e^1 . Spośród kilku dostępnych szablonów improwizacji wybrano następujący:

$$I, IV, I, I7, IV, IV7, I, I7, V, IV, I, I7$$

Występuje on w systemie pod nazwą HANDY_QUICK_CHANGE_7TH. Nazwa ta pochodzi od nazwiska znanego muzyka bluesowego Williama Christophera Handy’ego, na którego progresji bazowano tworząc jej, wyżej wymienioną, popularną wariację zwaną „quick-change” (z ang.: szybka zmiana). Nazwa ta odnosi się do nietypowego, przedwczesnego wprowadzenia subdominantowego akordu, który występuje jako drugi element progresji. Warto zauważyć, iż progresja ta opiera się na akordach triady harmoniczej oraz zbiega do trójdźwięku opartego na tonice. Jest to zgodne z podstawowymi zasadami nauki łączenia akordów – harmonii. Za pomocą toniki, szablon ukonkretniony został do następującego szeregu akordów:

$$E A E E7 \quad A A7 E E7 \quad B A E E7.$$

Znając tonikę oraz progresję, system rozpocząć może etap generowania taktów improwizacji. Do jego uruchomienia niezbędne jest określenie liczby taktów, z którymi składać ma się improwizacja. Wynika to bezpośrednio z długości progresji i, w tym przypadku, wynosi 12 taktów.

Podstawą algorytmu genetycznego, który pełni rolę generatora, jest baza zagrywek spośród których wybrana zostanie populacja początkowa. Bazą, która posłużyła do stworzenia analizowanego utworu, było 28 taktów wybranych z publikacji zawierającej zagrywki bluesowe [20]. Rysunek 6.1 ilustruje wspomniane, ponumerowane zagrywki. Plik dźwiękowy zawierający wspomniane zagrywki znaleźć można wśród plików dołączonych do pracy⁴.

Tabela 6.4 zawiera parametry algorytmu genetycznego użyte podczas generowania improwizacji.

Jak już wspomniano, bazę zagrywek stanowiło 28 taktów, których melodie mają charakter bluesowy. Spośród nich, wylosowana została grupa 20 fraz, które stanowiły populację początkową algorytmu genetycznego. Funkcja oceny bazowała na modelu podobieństwa, który brał pod uwagę kontur interwałowy, kontur wędrówki melodii oraz wektor zróżnicowania rytmicznego taktów; z wagami o wartościach, odpowiednio: 0.65, 0.2 oraz 0.15. Odległość pomiędzy dwoma modelami liczona była za pomocą uogólnionej miary

⁴ Plik zagrywki_bazowe w folderze 2_Analizowana_improwizacja. Rozpoczęcie każdego taktu sygnalizowane jest dźwiękiem perkusyjnym.

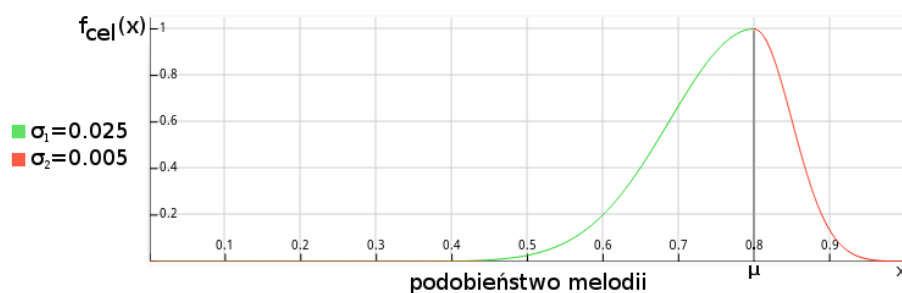
Tabela 6.4: Zestawienie parametrów algorytmu genetycznego, użytego do wygenerowania przykładowej improwizacji.

Baza zagrywek	
28 zagrywek bluesowych	
Populacja początkowa AG	
20 taktów wylosowanych z bazy zagrywek	
Model podobieństwa	
Element składowy	Waga
kontur interwałowy	0.65
kontur wędrówki melodii	0.20
wektor zróżnicowania rytmu (N=4)	0.15
Funkcja celu – parametry	
Parametr	Wartość
μ	0.800
σ_1	0.025
σ_2	0.005
Wzorce	
5 wylosowanych spośród bazy zagrywek	
Strategia selekcji	
rankingowa	
Operacja krzyżowania	
jednopunktowa	
Operatory mutacji	
Rodzaj	P-stwo wystąpienia
Dodanie mordentu	0.15
Dodanie przednutki krótkiej górnej	0.20
Dodanie trioli mordentowej	0.20
Dodanie przejścia pomiędzy 2 nutami	0.15
Scalanie nut	0.15
Liczba iteracji algorytmu genetycznego	
3	



Rys. 6.1: Okno odtwarzania oraz eksportu wygenerowanej improwizacji.

Minkowskiego z parametrem $m = 0.5$. Funkcja oceny przyjęła kształt odpowiadający temu z rysunku 6.2.



Rys. 6.2: Kształt funkcji celu $f_{cel}(x)$ użytej podczas generacji przykładowej improwizacji. Parametry funkcji to $\mu = 0.8$, $\sigma_1 = 0.025$ oraz $\sigma_2 = 0.005$.

Temat improwizacji, niewskazany przez użytkownika, został stworzony z pięciu wylosowanych z bazy zagrywek taktów.

Wybrano rankingową strategię selekcji.

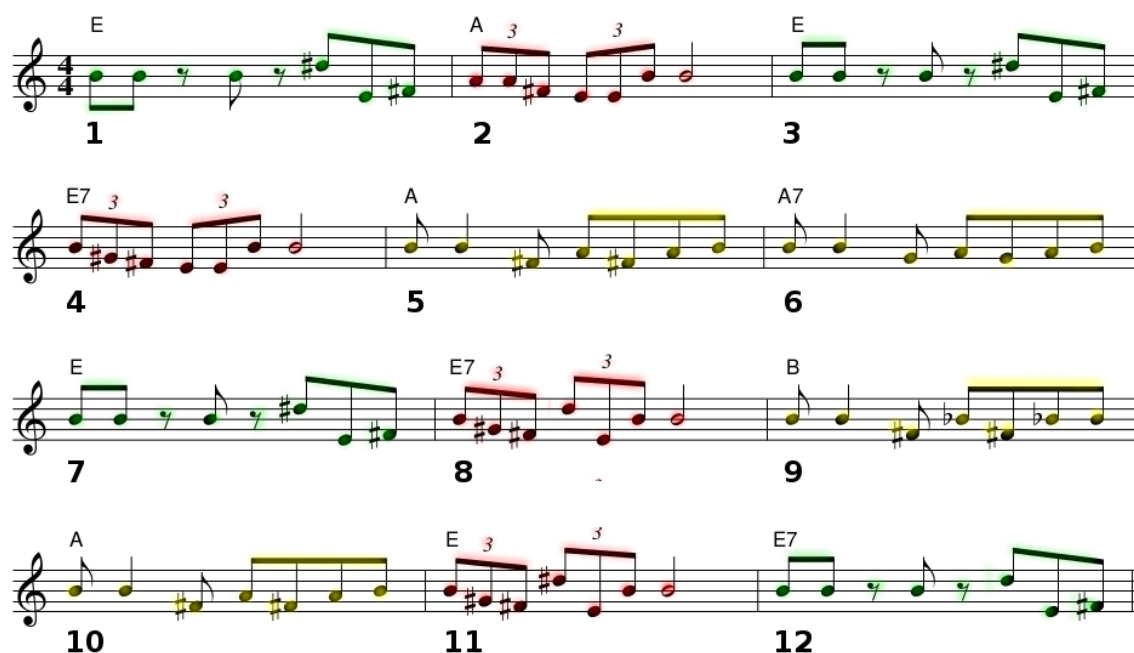
Wśród świadomych muzycznie operatorów znalazło się jednopunktowe krzyżowanie oraz pięć typów mutacji z prawdopodobieństwami wystąpienia odpowiadającymi tym z tabeli.

Podkład w każdym z taktów, przyjął czteroelementową postać, kolejno: toniki, akordu, toniki i akordu. Odpowiada to definicji *fcfc* w nagłówku MIDI opisującym sposób odgrywania podkładu.

Użyto następującego szablonu kompozycji:

ABAB CCAB CCBA

Składa się on z trzech różnych taktów: A, B, oraz C. Zostały one wybrane spośród najlepszych osobników ostatniej populacji algorytmu genetycznego. Algorytm przeprowadził trzy iteracje. Rysunek 6.3 przedstawia zapis nutowy wygenerowanej melodii. Każdy z czterech typów wspomnianego szablonu oznaczony jest innym kolorem.



Rys. 6.3: Zapis nutowy przykładowej improwizacji.

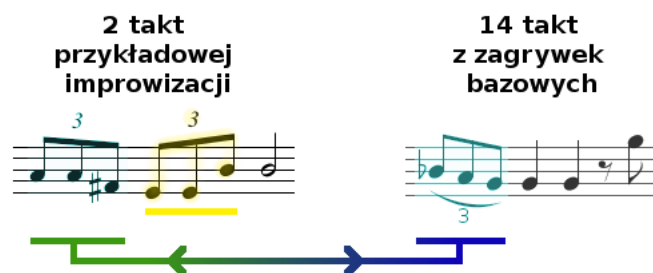
Dodatkowo, wygenerowana improwizacja, w postaci pliku mp3, znajduje się wśród plików załączonych do pracy⁵. Przed przejściem do kolejnej części sekcji zaleca się odsłuchanie wspomnianego pliku.

Liczba iteracji generatora taktów jest niewielka. Wynika to z chęci zaobserwowania zjawiska wzorowania się aplikacji na bazie zagrywek. Po trzech iteracjach istnieje duża szansa, iż niektóre takty będą składać się z fragmentów zagrywek na tyle dużych, iż będzie można wskazać frazy muzyczne, z których dany takt się składa. Omawiany przykład

⁵ Plik Analizowany_przykl_MP3 w folderze 2_Analizowana_improwizacja

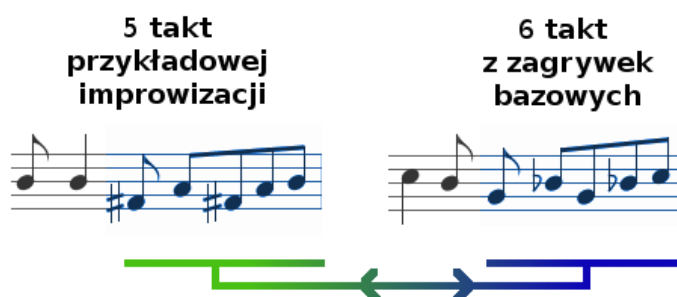
zawiera trzy takty, w których można dopatrywać się fragmentów z początkowej bazy zagrywek.

Rysunek 6.4 przedstawia podobieństwo pomiędzy drugim taktem przykładowej improwizacji, a czternastą frazą z bazy zagrywek. Podobieństwo to widać w pierwszych triolach fragmentów. Należy pamiętać, iż różnice w poszczególnych dźwiękach porównywanych triol, wynikają z poprawek, które przeprowadzone zostały podczas etapu korekcji. Ponadto, w takcie improwizacji, zauważyć można triolę, która jest efektem przeprowadzenia jednego z rodzajów mutacji (zaznaczona kolorem żółtym).



Rys. 6.4: Pierwszy przykład podobieństwa improwizacji do bazy zagrywek. Wynik mutacji – dodanie trioli.

Drugim przykładem, zilustrowanym na rysunku 6.5, są dwa takty, które w znacznej części są niemal identyczne. Zaznaczone fragmenty, choć różnią się nieznacznie wysokościami, posiadają ten sam kontur interwałowy – można mówić w tym przypadku o transpozycji fragmentu o odległość jednego półtonu.



Rys. 6.5: Drugi przykład podobieństwa improwizacji do bazy zagrywek.

Znaczna część podobieństw, pomiędzy improwizacją i bazą zagrywek, jest trudna do wyśledzenia. Wynika to zarówno z wielokrotnej operacji krzyżowania, mutacji oraz, co ilustruje rysunek 6.6, operacji przycinania przeprowadzanej podczas etapu przetwarzania końcowego. Oryginalny zapis piątego taktu przykładowej improwizacji nie jest podobny do żadnego z taktów bazowych. Gdy jednak weźmie się pod uwagę operację przycinania i, zaznaczone kolorem czerwonym nuty transponuje o oktawę w dół, można zauważyć pewne podobieństwo wstępujące pomiędzy piątym taktem improwizacji, a szóstym taktem

z zagrywek bazowych. Ponadto, warto zwrócić uwagę, na zaznaczoną kolorem żółtym mutację, która pomiędzy nutami *dis*¹ i *fis*¹ dodała przejściową nutę *e*¹.



Rys. 6.6: Trzeci przykład podobieństwa improwizacji do bazy zagrywek. Wynik przycięcia oraz mutacji, która dodała przejście pomiędzy dwiema nutami.

Według schematu kompozycji, drugi oraz ósmy takt improwizacji, przed etapem korekcji, były identyczne. Każdy z nich został jednak poprawiony względem innego akordu; pierwszy względem akordu *a-dur*, ósmy natomiast, względem akordu septymowego durowego – *E7*. Rzeczone takty ilustruje rysunek 6.7. Dodatkowo, nuta oznaczona kolorem zielonym, uległa przycięciu i w drugiej z zaprezentowanych fraz, została przeniesiona o oktawę w górę – na pozycję dźwięku *d*².



Rys. 6.7: Wyniki korekcji tego samego taktu względem dwóch różnych akordów.

Generator taktów, w przypadku analizowanej improwizacji, bazował na zagrywkach bluesowych. Warto zastanowić się czy omawiana melodia przejęła cechy tych zagrywek.

Choć brak w niej charakterystycznych dla tego gatunku dysonansowych „niebieskich” nut, to niektóre z części melodii nawiązują do elementów bluesa. Mowa tu o przesunięciu rytmu z akcentowanych ćwierćnotowych części taktu, jak również o triolach, które w specyficzny sposób urozmaicają rytm. Niestety, aby zachować charakter zagrywek, trzeba w sposób świadomy dostosować metodę improwizacji. Mogłaby ona np. przeprowadzać mutację w ograniczony sposób, co zapobiegało by zbyt wielkiemu zróżnicowaniu rytmicznemu, które może szybko rosnąć w kolejnych iteracjach algorytmu genetycznego, co nie jest charakterystyczne dla bluesa.

Wygenerowany utwór, jest świetnym przykładem, jak etap kompozycji może współgrać z progresją akordów, sprawiając, iż improwizowany utwór jest odpowiednio zróżnicowany. Sama kompozycja, bazuje jedynie na trzech taktach. Ich układ jednak, podobnie jak rymy w poezji, nadają rytm melodii. Dodatkowo, te same takty mogą być poprawiane względem różnych akordów. Słuchacz dostrzeże pomiędzy nimi podobieństwo, a dodatkowo wyłapując zmianę akordu, która jest zgodna z zasadami harmonii, ma on wrażenie naturalnej wędrówki melodii. Wszystko to, zgodnie z założeniami, ma pozytywny wpływ na odbiór utworu.

7. POTENCJAŁ ZAPROJEKTOWANEJ METODY ORAZ KIERUNKI JEJ ROZWOJU

Wyniki przeprowadzone wśród potencjalnych odbiorców muzyki użytkowej pokazują, że zaprojektowana metoda nie tylko realizuje zamierzone zadanie, ale robi to w sposób co najmniej zadowalający. Rzeczona metoda oraz system, który ją implementuje, zostały zaprojektowane w sposób, który umożliwia dowolną jego rozbudowę i kastomizację. Co więcej, wygenerowane utwory powstały w wyniku działania jedynie prototypu systemu, w którym, spośród wielu możliwości, zaimplementowano jedynie ich część. W modularności rozwiązania, jak i w architekturze opracowanej metody, tkwi potencjał, który pozwala na osiągnięcie jeszcze lepszych rezultatów przez dostrojenie jego parametrów. Właśnie tym zagadnieniom poświęcony jest niniejszy rozdział.

Baza zagrywek muzycznych, będąca w systemie odpowiednikiem pamięci muzyka, stanowi swojego rodzaju materiał, przetwarzany przez zaprojektowaną metodę improwizacji. Jego ilość oraz jakość ma istotny wpływ na wyniki improwizacji. Frazy, wchodzące w skład bazy zagrywek, mogą być dobierane według subiektywnych gustów muzycznych, czy też według kryterium przynależności do danego gatunku muzycznego. To, w jaki sposób dobór melodii wpływa na kształt końcowej improwizacji, jest pierwszym z wartych zbadania obszarów działania zaprojektowanej metody. Wspomniane badania powinny obejmować takie aspekty bazy zagrywek jak:

- licznosc,
- zróznicowanie taktów względem: rytmu, rozpiętości melodii itp.,
- przynależność fraz składowych do konkretnego gatunku muzycznego.

Każde ze wskazanych zagadnień może mieć istotny wpływ na jakość improwizacji.

Na wewnętrzną reprezentację melodii w systemie wybrano notację ABC, która, rzecz jasna, obejmuje wszelakie konstrukcje muzyczne – począwszy od metrum czy też podziału na takty, a skończywszy na strukturach takich jak triole czy synkopy. Prototyp systemu uwzględnia jedynie kilka wybranych struktur muzycznych. Ogranicza to w znaczny sposób różnorodność i złożoność melodii, zarówno w bazie zagrywek początkowych, jak i w uzyskiwanych improwizacjach. Zwiększenie liczby rozpoznawanych przez metodę konstrukcji muzycznych wpłynęłoby w istotny sposób na różnorodność i niepowtarzalność utworów. W prototypowej wersji, system korzysta jedynie z kilku struktur muzycznych, co skutkuje postrzeganiem melodii jako wtórnych i banalnych. Zaprojektowana aplikacja pozwala w łatwy sposób rozszerzyć pulę akceptowanych elementów notacji ABC. Należy w

tym celu rozbudować wyrażenie regularne, które je wydobywa z tekstowego zapisu utworów, a następnie w wybranych miejscach – wykorzystać. Warto rozbudować system w ten właśnie sposób – zabieg taki zwiększyć można różnorodność improwizacji lub umożliwić wskazanie metodzie grupy struktur, które powinna preferować podczas improwizacji, naśladując tym samym dany styl czy też gatunek muzyczny.

Jakość oraz wartość estetyczna generowanych improwizacji muzycznych, uzależniona jest od wielu parametrów, np. akordu, skali muzycznej, metrum, itd. Przykładem mogą być tu akordy, które występują w wielu odmianach¹. Wielodźwięki występują w zaimplementowanym prototypie w postaci jednego z typów enumeracyjnych. Pomimo tego, iż jest on przystosowany do łatwej rozbudowy, ujęto w nim jedynie dziewięć, najpopularniejszych typów akordów. Rozszerzenie go o dodatkowe akordy, przełożyłoby się na wzbogacenie i urozmaicenie podkładu muzycznego.

Pozostałe, uwzględnione poniżej, elementy systemu mają również znaczny potencjał rozwojowy.

- **Szablon progresji akordów**

Progresje akordów można zbudować w systemie od podstaw - wskazując kolejne konkretne akordy, które się na nią składają. Drugim, wygodniejszym sposobem tworzenia podkładu, jest skorzystanie z gotowych szablonów progresji. System oferuje ich osiem najpopularniejszych odmian. Każda z nich wywodzi się z jednego gatunku muzycznego – bluesa. Rozszerzenie systemu o kolejne szablony w oczywisty sposób zwiększy jego możliwości.

- **Pula dostępnych dźwięków**

Każda, rozpoznana w pliku notacji ABC nuta, reprezentowana jest w systemie poprzez jej odpowiednik dostępny w odpowiednim typie enumeracyjnym. Prototyp systemu umożliwia reprezentację 29 nut – od dźwięku *G* z oktawy małej, po *c* trzykreślne. Dodanie do wspomnianego typu kolejnych nut, zwiększy zarówno różnorodność improwizacji, jak i rozpiętość dźwięków, na których działać będą mogły operatory muzyczne generatora taktów.

- **Metrum**

Podobnie jak dla puli dostępnych w systemie dźwięków, rozszerzenie liczby typów metrum, pozwoli zarówno na zwiększenie różnorodności generowanych taktów, jak i charakteru całych utworów. Możliwe będzie również wykorzystanie w systemie kolejnych gatunków muzycznych, które często opierają się na frazach o ściśle określonym metrum.

- **Styl odgrywanego akompaniamentu**

¹ Wpisując w przeglądarce internetowej frazę „chords generator” trafić można na wiele generatorów akordów. W ramach testu wybrano jeden z pierwszych wyników – generator oferował wybór spośród 28 typów akordów.

W typie enumeracyjnym, który zawiera w sobie definicje rodzajów odgrywania podkładu muzycznego, znaleźć można trzynaście elementów. Podkreślić należy, że sposób odgrywania podkładu, ma niebagatelny wpływ na charakter utworu. Rodzaj odgrywanego podkładu może również w znaczny sposób wpłynąć na liczbę dysonansów w improwizacji. Warto rozszerzyć wspomnianą listę o kolejne sposoby odgrywania tła muzycznego oraz przeprowadzić badania, które mogłyby każdemu z nich przypisać stopień w jakim „ukrywa” bądź wzmacnia dysonanse w utworze.

Głównym celem zaprojektowanej metody jest generowanie improwizacji. Najczęstszym formatem, w jakim są one finalnie prezentowane, jest MIDI, który pozwala na natychmiastowe odsłuchanie utworu i, tym samym, jego ocenę. System umożliwia wybór instrumentów (tzw. programów), które odgrywają poszczególne linie melodyczne. Niestety, jakość barw instrumentów nie jest najwyższa, co może skutkować negatywną oceną melodii. W plikach dołączonych do niniejszej pracy,² znaleźć można zapis improwizacji w postaci pliku MIDI oraz jej prostą aranżację w jednym z wielu programów do tworzenia muzyki. Porównanie obu plików pokazuje, że rozszerzenie systemu o moduł, który byłby w stanie w sposób automatyczny aranżować improwizacje, w znaczny sposób mógłby zwiększyć ich jakość, a co najmniej, zminimalizować negatywny wpływ słabej jakości barwy dźwięków na odbiór wygenerowanych utworów.

Istotnym elementem zaprojektowanej metody improwizacji jest algorytm genetyczny. Jego składowe mogą stanowić wyjątkowo interesujący przedmiot badań. Większość parametrów opisujących rzeczony algorytm może mieć istotny wpływ na finalną postać improwizacji. Wśród elementów, które są standardowymi elementami dostrajania algorytmów ewolucyjnych, wyróżnić należy te, które są specyficzne dla zaprojektowanego rozwiązania, a które w większości pozwalają na rozbudowę i optymalizację.

Dobór odpowiedniego kształtu funkcji celu jest ważnym elementem parametryzowania metody improwizacji. Równie istotny, jednocześnie bardziej interesujący, jest model opisu taktów, który służy do ich porównywania. Jak zostało już wielokrotnie podkreślone, stopień podobieństwa melodii jest niezwykle trudny do uchwycenia. Wieloaspektowość muzyki utrudnia stworzenie uniwersalnego modelu opisu utworów, który uwzględniałby większość istotnych elementów dla tej dziedziny sztuki. Zaprojektowany komparator taktów pozwala na dodawanie kolejnych modułów, które mogą służyć do opisu taktów w sposób, który przyczyniłby się do zwiększenia precyzji w określaniu stopnia podobieństwa pomiędzy taktami. Podkreślić należy, że nawet bez wspomnianej rozbudowy, samo już badanie zaprojektowanych modeli, pozwoliłoby na poprawę wyników systemu improwizującego.

² Folder *2_Analizowana_improwizacja* zawiera improwizację w formacie MIDI – plik *Analizowany_przykl_MP3*. W podfolderze *Aranżacja* znaleźć można jej aranżacje.

Kolejnym obszarem, w którym tkwi potencjał rozwojowy systemu, są „świadome” muzycznie operatory algorytmu genetycznego oraz działania podejmowane podczas etapu przetwarzania końcowego. Zwiększenie liczby rodzajów rzeczonych operacji, w oczywisty sposób poszerzyłoby możliwości systemu. Ponadto poprzez odpowiedni dobór prawdopodobieństw ich użycia, można pokusić się o próbę naśladowania stylów improwizacji konkretnych muzyków. Wspomnianą „świadomość” muzyczną operatorów, można spróbować wynieść na kolejny poziom, uwzględniając w ich działaniu tonację utworu, jego metrum czy nawet dotychczasowy przebieg melodii.

Rozbudowa, wchodzących w skład systemu elementów, w oczywisty sposób zwiększa jego możliwości. O potencjale zaprojektowanej metody, a tym samym możliwościach zbudowanego prototypu, świadczy fakt, iż operując wyłącznie na podstawowych, ograniczonych elementach, osiągnięto tak zadowalające wyniki. Warto również zwrócić uwagę na możliwości architektury systemu, która pozwala na jego bezproblemowy rozwój. W połączeniu z możliwościami ulepszenia pozostałych elementów, zaprojektowany system, daje szerokie pole do dalszych badań, a w perspektywie może być bazą dla wypracowania znacznie lepszych metod improwizacji.

ZAKOŃCZENIE

Celem niniejszej pracy było zaprezentowanie metody oraz implementującego ją prototypu, których zadaniem jest generowanie muzyki użytkowej, a dokładniej rzecz ujmując – tworzenie solowych improwizacji muzycznych do zadanego podkładu muzycznego.

Autorowi chodziło przede wszystkim o stworzenie powszechnie dostępnego źródła muzyki użytkowej, naśladowującego ludzki proces twórczy.

W podsumowaniu, należałoby stwierdzić, iż cel badawczy został osiągnięty, a mianowicie, w efekcie pracy systemu, uzyskano improwizacje muzyczne o wartości estetycznej co najmniej akceptowalnej dla ludzkiego ucha, co potwierdzają wyniki badań ankietowych, przeprowadzonych wśród potencjalnych odbiorców tego typu muzyki.

Niezwykle ważkim aspektem podjętego tu tematu jest możliwość praktycznego wykorzystania efektów pracy systemu – czyli wygenerowanych utworów muzycznych.

W tym miejscu można by zadać pytanie, jakie argumenty przemawiają za celowością sztucznego generowania muzyki użytkowej, w obliczu potencjału twórczego muzyków. Jest to przede wszystkim łatwość uzyskiwania określonych kompozycji za pomocą programu komputerowego, permanentna gotowość systemu do generowania muzyki, bez konieczności oczekiwania na wenę twórczą, duża wydajność w tworzeniu melodii, nieosiągalna dla człowieka, związana z tym bezpośrednio możliwość unikania monotonii przez częstą zmianę odtwarzanych utworów bez ich powtarzania.

Niebagatelny jest też aspekt ekonomiczny. Pozyskiwana za pomocą programu muzyka, nie wymaga opłat za jej odtwarzanie ani też zgody twórcy.

A zatem wracając do zasadniczej kwestii, dotyczącej możliwości zastosowania generowanych sztucznie kompozycji, należałoby przytoczyć konkretne przykłady wykorzystywania muzyki, o której tu mowa. I tak, trzeba zdawać sobie sprawę, że wciąż rośnie liczba miejsc, w których muzyka pełni rolę tła wykonywanych przez człowieka rozmaitych czynności. Są to: hipermarkety, galerie handlowe, zakłady usługowe, restauracje itp.

Od dawna wiadomo, że muzyka stymuluje nastrój i silnie oddziałuje na psychikę człowieka. Muzykoterapia, to uznana dziedzina terapii, posługująca się, jako narzędziem utworami muzycznymi, indywidualnie dobranymi do potrzeb i wrażliwości pacjenta. To również pole do zagospodarowania przez utwory generowane za pomocą zaprojektowanej metody. I tu znów permanentna dostępność, łatwość pozyskiwania i mnogość, a także możliwość indywidualnego kształtowania na potrzeby konkretnej osoby, przemawiają za szerokim wykorzystaniem tego narzędzia we wspomnianej tu dziedzinie terapii.

Podobne, jak w przypadku muzykoterapii, cechy systemu, a w szczególności możli-

wość dostrajania i mnogość generowanych melodii, pozwalają na wykorzystanie aplikacji przez szukających inspiracji muzyków.

Reasumując, przedstawione powyżej przykłady, wskazują na szerokie możliwości praktycznego wykorzystania muzyki generowanej przez zaprojektowaną aplikację. Fakt ten zdaje się świadczyć za zasadnością podjętej tu pracy nad stworzeniem rzeczzonego systemu.

Dodatki

A. OPIS STRUKTURY PLIKÓW PROJEKTU. JEGO KOMPILACJA ORAZ URUCHOMIENIE

Projekt został podzielony na moduły. Jest to zgodne ze standardem, który narzuca użyte do automatyzacji budowy oprogramowania narzędzie – APACHE MAVEN. Oprogramowanie to, dostępne pod wskazanym adresem¹, jest niezbędne do stworzenia uruchamialnych plików zaimplementowanego systemu.

Pliki projektu znaleźć można w folderze *Prototyp systemu improwizującego*, który znajduje się pośród plików dołączonych do niniejszej pracy.

Poniżej zamieszczony jest opis kolejnych kroków, które należy wykonać aby uruchomić interfejs graficzny aplikacji.

1. Należy przejść do głównego katalogu projektu o nazwie ROOT. Znajdując się w tej lokacji, należy wykonać polecenie „mvn clean install”. Zainstaluje one wszystkie moduły do lokalnego repozytorium MAVENA.
2. Następnie, należy przejść do katalogu „gui” i wewnątrz niego wykonać polecenie „mvn assembly:single”.
3. Po pomyślnym zakończeniu poprzedniej operacji w podkatalogu „target” znajdować się powinien uruchamiany plik JAVY o nazwie „gui-0.0.1-SNAPSHOT-jar-with-dependencies.jar”. Plik ten jest gotowy do uruchomienia poprzez dwukrotne kliknięcie lub wykonanie na nim polecenia „java -jar gui-0.0.1-SNAPSHOT-jar-with-dependencies.jar”.

Uwaga: program MAVEN wymaga połączenia z internetem. Jest ono niezbędne do pozyskania wszelkich niezbędnych do uruchomienia aplikacji bibliotek, które przechowywane są w zewnętrznym repozytorium Apache. Ściągnięcie wspomnianych plików odbędzie się jednokrotnie.

Uwaga: jedna z używanych bibliotek – *simmetrics* – nie jest uwzględniona w internetowym repozytorium MAVEN’a. Należy ją zainstalować ręcznie. Skrypt stworzony do uruchomienia w dowolnym systemie linuxowym znajduje się w katalogu *Maven_dodatkowe_pliki/simmetrics* i nosi nazwę *installInMVNLocalRepo.sh*. Na jego podstawie można stworzyć skrypt bądź wydobyć odpowiednie komendy do zainstalowania biblioteki np. w systemie WINDOWS.

¹ Strona projektu: www.maven.apache.org

Przypomnienie: aplikacja wymaga zainstalowanego środowiska uruchomieniowego JAVA.

B. OPIS GRAFICZNEGO INTERFEJSU UŻYTKOWNIKA APLIKACJI

Na podstawie zaprojektowanej metody powstał prototyp systemu – framework umożliwiający generowanie improwizacji. Aby ułatwić prezentację jego możliwości, stworzono graficzną nakładkę, która umożliwia generację utworów oraz zmianę **większości** parametrów improwizacji.

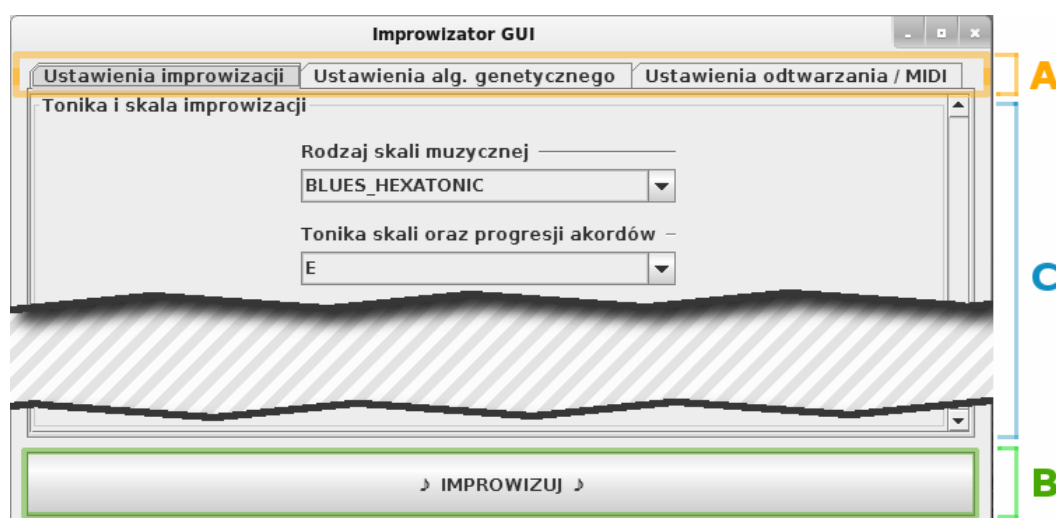
Poniżej znajduje się opis graficznego interfejsu użytkownika. Wskazano również parametry, na których warto skupić się podczas pierwszych eksperymentów z systemem.

Przypomnienie: aplikacja wymaga zainstalowanego środowiska uruchomieniowego JAVA.

Uwaga: podczas implementacji skorzystano z gotowego frameworku pozwalającego na tworzenie algorytmów genetycznych. W niektórych sytuacjach, ze względu na wykorzystywaną wielowątkowość, staje się on przyczyną problemów z wygenerowaniem improwizacji. Wynika to najprawdopodobniej z faktu, iż rzeczony framework nie jest silnie rozwijanym projektem.

Aplikacja składa się z trzech głównych elementów (rysunek B.1). Należą do nich:

- trzy zakładki z ustawieniami generacji improwizacji (litera **A** na rysunku),
- część okna z ustawieniami konkretnej zakładki (litera **C** na rysunku),
- przycisk uruchamiający generowanie improwizacji (litera **B** na rysunku).



Rys. B.1: Trzy główne elementy graficznego interfejsu użytkownika.

Zakładka **Ustawienia improwizacji** (rysunek B.2) zawiera główne ustawienia kolejnych etapów improwizacji¹. . Wśród parametrów rzeczzonej zakładki, wyróżnić należy te podstawowe:

- progresję akordów (element A na rysunku),
- rodzaj akompaniamentu (element B na rysunku),
- szablon kompozycji (element C na rysunku),
- rodzaj korekcji (element D na rysunku).

Rys. B.2: Zakładka „Ustawienia improwizacji” graficznego interfejsu użytkownika.

Zakładka **Ustawienia alg. genetycznego** zawiera wszelkie istotne dla generatora taktów parametry (rysunki: B.4 oraz B.3). Spośród nich, warto zwrócić uwagę przede wszystkim na:

¹ Wyjątkiem jest generator taktów – algorytm genetyczny – ma on osobną zakładkę – *Ustawienia alg. genetycznego*.

- typ strategii selekcji (element B na rysunku B.3),
- operatory GA (głównie mutacje, które odpowiadają za ozdobniki; element B na rysunku B.3),
- liczbę iteracji algorytmu genetycznego (element C na rysunku B.3).

Strategia selekcji GA

☐ Rank selection
☒ **Tournament selection** 0,8 A
☐ Roulette selection

Operatory GA

☒ Krzyżowanie n-punktowe n: 1
☒ Mutacja 'Mordent' p: 0,15
☒ Mutacja 'Przednutka' p: 0,2
☒ Mutacja 'Triola' p: 0,25 B
☒ Mutacja 'Portamento' p: 0,15
☒ Mutacja 'Scalanie' p: 0,1 def: 1;2;3
☐ Mutacja 'Reverse' p: 0,005

Parametry iteracji GA

Ile iteracji przeprowadzić? _____
 5 C
 Nazwa improwizacji _____
 Improwizacja Algorytmu Genetycznego (2013-01-22 17:46:58)

Rys. B.3: Druga część zakładki „Ustawienia alg. genetycznego” graficznego interfejsu użytkownika.

Ostatnia zakładka – „Ustawienia odtwarzania/MIDI” – pozwala przede wszystkim na dobór instrumentów oraz głośności poszczególnych linii melodycznych. Jest ona przedstawiona na rysunku B.5.

Po wciśnięciu przycisku „**IMPROWIZUJ**”, aplikacja przystąpi do generowania utworu. Może to potrwać kilka sekund, dlatego, w tym czasie, graficzny interfejs zostaje zablokowany i wyświetlony jest panel sygnalizujący trwanie prac nad improwizacją (rys. B.6).

Po zakończeniu procesu generowania improwizacji, w przykładowej aplikacji z graficznym interfejsem użytkownika, wyświetlane jest okno z podglądem muzycznej notacji głównej linii melodycznej oraz okno przedstawione na rysunku B.7, pozwalające na utworzenie improwizacji (kolor zielony) lub jej zapisu kilku formatów. Zawiera ono dodatkowo pole tekstowe z improwizacją w postaci notacji ABCNotation.

Populacja początkowa GA (aka wiedza / ograniczenie)

Źródło wiedzy/taktów

☒ Z wewnętrznej bazy zagrywek

☐ Z pliku

Wczytaj plik

...

Maksymalna liczba taktów w populacji początkowej

10

Funkcja dopasowania GA

Modele oraz ich wagi

☒ Model interwałowy w: 1

☒ Model kier. melodii w: 1

☒ Model rytmu w: 1 par: 4

Metryka odległości

(parametr m metryki Minkowskiego:)

0.5

Parametry kształtu f-cji dopasowania

DEFAULT_MINK_05_NOT_SIMMETRIC

parametr a: 1

parametr b: 0,2

parametr cL: 0,025

parametr cR: 0,015

Wzorce

☒ Losowo z populacji pocz., nie więcej niż: 5

☐ Modele jako wzorce:

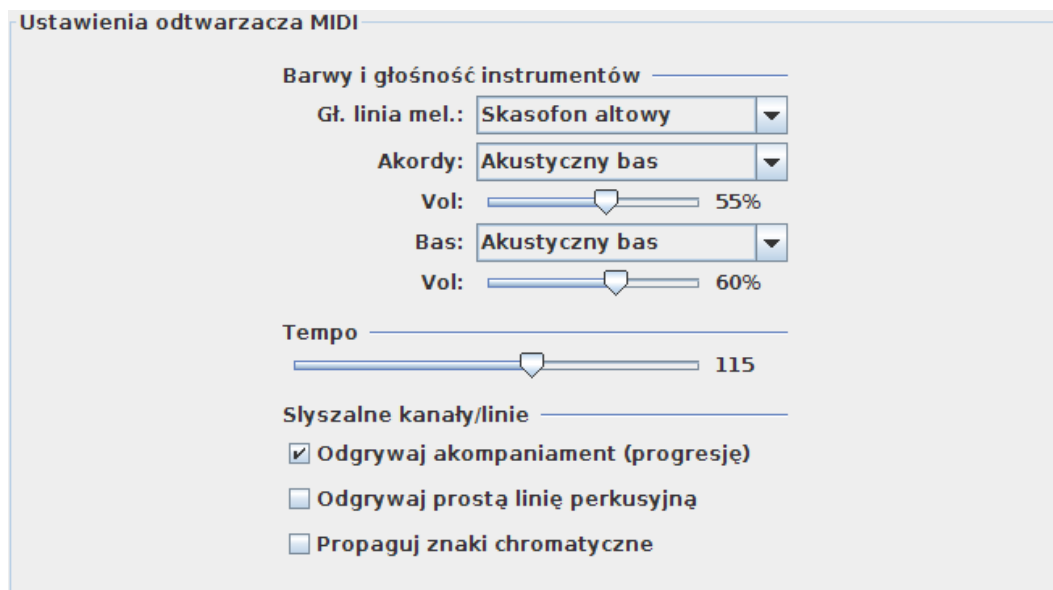
Kontur interwałowy:

Kontur kier. melodii:

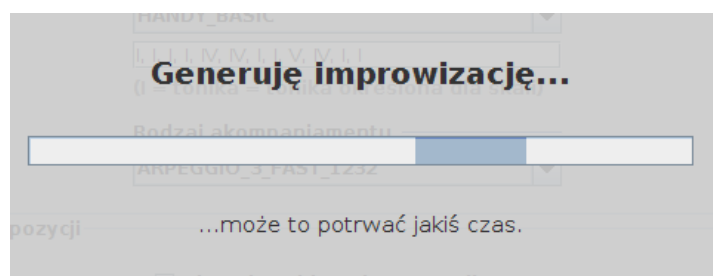
Kontur rytmu:

Rys. B.4: Pierwsza część zakładki „Ustawienia alg. genetycznego” graficznego interfejsu użytkownika.

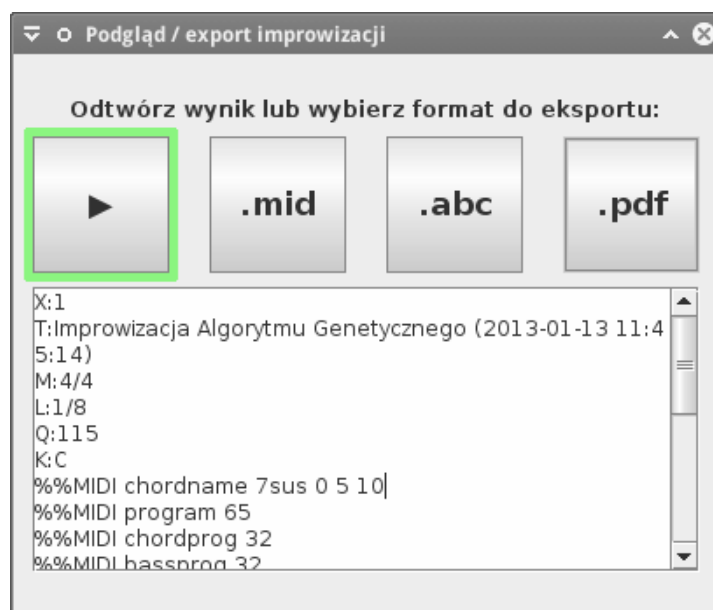
Autor niniejszej pracy zachęca do eksperymentowania!



Rys. B.5: Zakładki „Ustawienia odtwarzania/MIDI” graficznego interfejsu użytkownika.



Rys. B.6: Panel postępu prac nad improwizacją.



Rys. B.7: Okno odtwarzania oraz eksportu wygenerowanej improwizacji.

BIBLIOGRAFIA

- [1] *The abc music standard 2.1 (dec 2011)*, <http://abcnotation.com/wiki/abc:standard:v2.1>. 2012.
- [2] Alexjander, S., *The infrared frequencies of dna bases, as science and art*, IEEE Engineering In Medicine and Biology. 1999.
- [3] Baggi, L., *Computer-generated music*, IEEE Computer. 1991.
- [4] Biles, J.A., *Evolutionary computation gets a gig*, Proceedings of the 2002 Conference for Information Technology Curriculum. 2002.
- [5] Burton, A.R., Vladimirova, T.R., *Generation of musical sequences with genetic techniques*, Comput. Music J. 1999, tom 23, 4.
- [6] Eck, D., Schmidhuber, J., *A First Look at Music Composition using LSTM Recurrent Neural Networks*, Rap. tech., IDSIA. 2002.
- [7] Hoos, H., Hamel, K., *Guido music notation version 1.0: Specification part i, basic guido*, Technical Report TI. 1997, tom 20.
- [8] Kłys, D., Zabierowski, W., *Aplikacja wykorzystująca komunikaty midi, odzwierciedlenie danych w postaci prostego zapisu nutowego*. 2004. Konferencja Sieci i Systemy.
- [9] Lorrain, D., *A panoply of stochastic 'cannons'*, Computer Music Journal 3. 1980.
- [10] Michalewicz, Z., Nahorski, Z., *Algorytmy genetyczne + struktury danych = programy ewolucyjne* (WNT, 2003).
- [11] O'Connor, J., Robertson, E., *Augusta ada king, countess of lovelace*, <http://www-history.mcs.st-andrews.ac.uk/Biographies/Lovelace.html>. 2012.
- [12] Orusava, T., Unehara, M., *Composition of music using human evaluation*, IEEE International Fuzzy Systems Conference. 2001.
- [13] Parsons, D., *Directory of Tunes and Musical Themes* (Spencer Brown, 1975).
- [14] Pope, S.T., *Fifteen years of computer-assisted composition*. 1994.
- [15] Ricanek, K., Homaifar, A., Lebby, G., *Genetic algorithm composes music*, w: *System Theory, 1993. Proceedings SSST* (IEEE, 1993).
- [16] Schmuckler, M.A., *Testing Models of Melodic Contour Similarity*, Music Perception. 1999, tom 16, 3.
- [17] Sikorski, K., *Harmonia*, tom 1 (Polskie Wydawnictwo Muzyczne, Kraków, 1976).
- [18] Smoliar, S., *Algorithms for musical composition: A question of granularity*, Computer. 1991, tom 24.
- [19] Thaler, S., *Neural networks 101*, Servo Magazine. 2005.
- [20] Villegas, G., *Piano blues*, (zasób elektroniczny). 2012.
- [21] Waldorf, J., *Sekrety Polichymnii* (Iskry, Warszawa, 1957).
- [22] Wine, T., *1001 Blues Licks* (Cherry Lane Music Company, 2003).

- [23] Winograd, T., *Linguistics and the computer analysis of tonal harmony*, Journal of Music Theory. 2005, tom 12, 1.
- [24] Zhu, Y., Kankanhalli, M., *Similarity matching of continuous melody contours for humming querying of melody databases, w: of Melody Databases,” International Workshop on Multi-media Signal Processing, USVI (2002).*