Zachodniopomorski Uniwersytet Technologiczny
w Szczecinie
Wydział Informatyki

mgr inż. Łukasz Mazurowski

Streszczenie rozprawy doktorskiej

Model systemu algorytmicznego komponowania muzyki na podstawie wzorców

Promotor rozprawy:
prof. dr hab. inż. Antoni Wiliński

Recenzenci rozprawy:

prof. dr hab. inż. Dariusz Kania
Wydział Automatyki, Elektroniki i Informatyki
Politechnika Śląska

dr hab. inż. Przemysław Klęsk
Wydział Informatyki
Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

Szczecin, 2017

Spis treści

1. Aktualność problemu 2
2. Przedmiot badań 3
3. Metody naukowe stosowane podczas wykonywania badań 4
4. Główny cel rozprawy 5
5. Zadania do wykonania 5
6. Wartość teoretyczna 6
7. Wartość praktyczna 6
8. Akceptacja wyników przez społeczność 6
9. Główne osiągnięcia pracy 7
10. Struktura i układ pracy 7
11. Koncepcje algorytmicznego komponowania muzyki 9
11.1. Komponowanie bez definiowania formy generowanego utworu 11
11.2. Komponowanie uwzględniające opis formy generowanego utworu 17
12. Spis publikacji własnych 26
13. Podsumowanie 26
Literatura 29

1. Aktualność problemu

Systemy algorytmicznego komponowania muzyki można pogrupować, uwzględniając rodzaj reguł komponowania, wykorzystanych w procesie generowania muzyki. Wg [47] systemy te można podzielić na dwie grupy. Pierwszą stanowią rozwiązania z regułami jawnymi, w których kompozytor definiuje z góry zasady komponowania utworu. Drugą grupę stanowią systemy z regułami niejawnymi, w których reguły komponowania pozyskiwane są z przykładów (fragmentów bądź całych utworów muzycznych) za pomocą wybranych metod przetwarzania i pozyskiwania informacji. W tab. 1 przedstawiono klasyfikację systemów algorytmicznego komponowania muzyki analizowanych przez autora pod względem jawności i niejawności reguł komponowania zastosowanych w algorytmie generującym muzykę.

W nawiązaniu do przedstawienia różnych podejść z dziedziny algorytmicznego komponowania muzyki zasadne wydaje się wskazanie pewnej luki w prezentowanych rozwiązaniach jaką jest tzw. muzyka tła (*background music*), będąca jednym z najbardziej interesujących obszarów stosowania muzyki użytkowej. Prezentowane w autorskiej dysertacji koncepcje z założenia funkcjonują jako narzędzia generujące muzykę tła, która odnosi się do "*muzyki jakiegokolwiek rodzaju, która jest odtwarzana podczas danej czynności, w taki sposób, aby człowiek aktywnie nie angażował się w słuchanie tej muzyki*" (na podstawie [24]). Ten typ muzyki odgrywa drugorzędną rolę w życiu człowieka, jednak według różnych badań ma wpływ na wiele aspektów naszego zachowania.

Badania prowadzone przez Kämpfe i innych w [36] skupione były na analizie wpływu muzyki na różne zadania wykonywane przez człowieka. Autorzy zasugerowali, że chociaż globalne wyniki analizy nie wykazują znaczącego wpływu na efektywność zadań, to szczegółowe badania pozwalają jednak zidentyfikować zarówno pozytywne jak i negatywne oddziaływanie na analizowane czynności. Cockerton i inni w [15] śledzili pozytywny i negatywny wpływ słuchania muzyki podczas badania funkcji poznawczych (rozwiązywania testu kognitywnego). Także Chaudhury i inni w [12] zaprezentowali rolę muzyki w procesie przeprogramowania aktywności mózgu. Autorzy wywnioskowali, że muzyka może wywoływać różne mechanizmy funkcjonowania mózgu m.in. proces nauczania oraz pamiętania.

Przedstawione analizy są jedynie częścią badań prowadzonych na świecie, mających znaczenie nie tylko poznawcze ale również komercyjne. Autor przypuszcza, że system generujący muzykę tła adekwatną do zadanego rodzaju towarzyszącej jej czynności, byłby narzędziem użytecznym rynkowo, pozwalającym na generowanie odpowiedniego gatunku muzyki w zależności od aktualnych potrzeb klienta-słuchacza. Na bazie analizy istniejących rozwiązań opisanych w literaturze oraz niewypełnionej luki badawczej związanej z systemami algorytmicznego komponowania muzyki tła, powstały koncepcje systemów algorytmicznego komponowania muzyki na podstawie wzorców.

Autor podejmując opisywaną w tej pracy problematykę kierował się, oprócz własnymi fascynacjami, także a może i przede wszystkim aktualnością prowadzonych badań. Ośrodki takie jak *MIT*, *Columbia University* [51], *University College London* [37] czy *Griffith University* [52], oferując w ramach prowadzonych kursów tematykę związaną z muzyką generatywną, kształtują kierunek rozwoju tej gałęzi nauki. W odczuciu autora dorobek badawczy przedstawiony w tej pracy pozostaje w zgodzie z

tym narastającym trendem, niemniej jednak pozostawiając w dalszym ciągu szerokie pole do doskonalenia, rozszerzania i poprawiania uzyskanych autorskich metod tworzenia muzyki generatywnej.

Rozwiązanie opisane w	**Bazuje na**	**Wykorzystanie jawnych regułach**	**Wykorzystanie niejawnych regułach**
[38]	automaty komórkowe	X	-
[7]	gramatyki	X	X
[3], [4], [5]	algorytm genetyczny	X	-
[43]	algorytm genetyczny	X	-
[45]	algorytm genetyczny	X	-
[32]	algorytm genetyczny	X	-
[10]	algorytm roju cząstek (PSO)	X	-
[33]	algorytm mrówkowy	X	-
[41], [50]	wirtualny model systemu chemicznego	X	-
[55], [54]	sztuczne sieci neuronowe	X	-
[42]	drzewa dezycyjne	-	X
[28], [29]	parsowanie przyrostowe i drzewa PST	-	X
[17]	modele statystyczne	-	X
[25]	ukryte modele Markowa (HMM)	-	X
[30]	łańcuchy Markowa	-	X
[53]	łańcuchy Markowa	-	X
[13], [27]	sztuczne sieci neuronowe	-	X
[19], [18], [20], [21], [22], [16]	sieci ATN i MATN	X	X

Tab. 1: Podział systemów algorytmicznego komponowania muzyki ze względu na jawność i niejawność wykorzystanych reguł komponowania. Opracowanie własne.

2. Przedmiot badań

Przedmiotem badań jest opracowanie autorskich koncepcji algorytmicznego komponowania muzyki (akceptowalnej przez człowieka i być może użytecznej rynkowo), w których opracowano autorskie metody przetwarzania treści muzycznych, bazujące na metodach sztucznej inteligencji. W pracy wykorzystane zostały metody probabilistyczne (np. generowanie progresji akordów) jak i eksploracji danych (estymacja tonacji wzorca, rozkład klas długości dźwięków wzorca, okienkowanie wzorca w celu generowania melodii i linii basowej).

3. Metody naukowe stosowane podczas wykonywania badań

Podczas opracowania metod algorytmicznego komponowania muzyki na podstawie wzorców wykorzystano metody naukowe opisane w tab. 2.

Metoda	**Wykorzystana w celu**
metoda analizy literatury	wskazania kierunku dyskusji naukowej i wykazania luki badawczej
metoda estymacji tonacji Krumhansl-Kessler'a [14]	oszacowania tonacji wzorca
model Parncutt'a czasowych akcentów dźwięku [46]	obliczenia histogramu klas wysokości dźwięków wzorca
metoda statystyczna (rozkład zmiennej czasu trwania dźwięków wzorca)	obliczenia histogramu klas długości dźwięków wzorca
obserwacja zmiany wartości wysokości dźwięku	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany wartości długości wzorca	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany parametrów zbiorów wygenerowanych i wykorzystanych wzorców	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany struktury tablicy możliwych powtórzeń wzorca w kompozycji	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany struktury opisującej formę kompozycji	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany struktury opisującej skalę muzyczną	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany struktury opisującej lokalizację dźwięków w obrębie wzorca	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany struktury opisującej przesunięcia (offsety) dźwięków w obrębie wzorca	analizy wpływu zmian wartości parametru na postać wynikowej kompozycji muzycznej
obserwacja zmiany liczby i czasu trwania akordów w obrębie wzorca	analizy wpływu zmian wartości parametrów na postać wynikowej kompozycji muzycznej

Tab. 2: Metody naukowe stosowane podczas wykonywania badań. Opracowanie własne.

4. Główny cel rozprawy

Celem pracy jest opracowanie modelu komputerowego systemu komponowania muzyki użytkowej (posiadającej wartość rynkową), w którym do generowania oryginalnych utworów muzycznych wykorzystuje się metody autorskie oparte na metodach sztucznej inteligencji.

Po analizie dotychczasowych rozwiązań w zakresie algorytmicznego komponowania utworów oraz konstruowania maszyn do generowania muzyki sztucznej a także na podstawie badań autorskich algorytmów stawia się hipotezę, że, bazując na metodach sztucznej inteligencji, możliwe jest opracowanie takich algorytmów komponowania utworów muzycznych, których efektem będą utwory atrakcyjne w odbiorze i posiadające strukturę poprawną w aspekcie teorii muzyki. W powyższej hipotezie badawczej zakłada się, że system komponowania bazował będzie na wzorcowych utworach stanowiących muzyczną bazę wiedzy oraz na metodach wyszukiwania i modyfikacji sygnatur muzycznych (komponentów utworu), wykorzystywanych w kompozycji wynikowej. Przez model w ujęciu metodologicznym autor rozumie narzędzie kognitywne umożliwiające badanie i opisywanie rozmaitych aspektów struktury i funkcjonalności systemu komponowania muzyki. Model taki będzie krokiem pośrednim do syntezy metody lub metod generowania muzyki sztucznej prowadzącej w przyszłości do budowania złożonego systemu informatycznego. Przez system ten autor rozumie z kolei rozbudowaną instalację sprzętowo-programową, której budowa nie jest celem niniejszej pracy. Praca ta może być inspiracją, pierwszym krokiem do budowy i doskonalenia takiego systemu.

W przypuszczeniu, że utwory będą atrakcyjne w odbiorze kryje się wiele subiektywizmu i ta cecha utworów będzie wymagała specyficznego potwierdzenia. Generalnie trudno zdefiniować bezwzględną atrakcyjność utworu muzycznego, z pewnością jest to niemożliwe, można jednak posiłkować się miarami pośrednimi – statystycznie mierzonym zainteresowaniem lub rynkowymi np. wartością sprzedaży. Drugi wymieniony istotny postulat – zgodność z przyjętymi kanonami teorii muzyki, jest znacznie prostszy do dowiedzenia i zapewnienia.

5. Zadania do wykonania

W ramach opracowania metod algorytmicznego komponowania muzyki na podstawie wzorców muzycznych wykonano następujące zadania:

1. przeprowadzono analizę istniejących rozwiązań z dziedziny algorytmicznego komponowania muzyki pod kątem wyszukania luki badawczej,
2. zdefiniowano komponenty procesu generowania muzyki tła w ujęciu systemowym,
3. opracowano metodę generowania kompozycji muzycznych na podstawie wzorców bazującą na mini-generatorach,
4. zaimplementowano metodę z pkt. 3 w środowisku Matlab w celu przeprowadzenia eksperymentów,
5. przeprowadzono szereg eksperymentów, których wyniki były analizowane przez autora i upubliczniane w ramach działań popularyzatorskich,
6. na podstawie metody z pkt. 3 oraz w odpowiedzi na uwagi środowiska odbiorców kompozycji zdefiniowano metodę generowania kompozycji na podstawie wzorców, uwzględniającą opis struktury kompozycji wynikowej,

7. zaimplementowano metodę z pkt. 6 w środowisku Matlab w celu przeprowadzenia eksperymentów,
8. empirycznie zweryfikowano hipotezę postawioną w pracy.

6. Wartość teoretyczna

Wartość teoretyczna rozprawy jest wynikiem opracowania dwóch metod algorytmicznego komponowania muzyki na podstawie wzorców, w ramach których dokonano sformalizowania aspektów zadań procesu komponowania muzyki (tab. 3 na str. 10) związanych z:
— generowaniem melodii,
— rozszerzaniem idei melodycznych,
— harmonizacją,
— hybrydowym podejściu do budowy struktury utworu.
Teoretyczną wartością dodaną w tym względzie jest zdefiniowanie formalizmów wzorca muzycznego, mini-generatora oraz kompozycji wyjściowej a także algorytmów:
— generowania melodii na podstawie sekwencji akordów,
— generowania linii basowej na podstawie melodii,
— generowania wzorca muzycznego za pomocą mini-generatora.

7. Wartość praktyczna

Wartość praktyczną dysertacji stanowią liczne przykłady publikacji wygenerowanych kompozycji w ramach różnych wydarzeń, w tym z udziałem mediów. Należy w tym miejscu wspomnieć o prezentowaniu rzeczywistych kompozycji i własnych koncepcji autora na antenie Polskiego Radia Szczecin (audycja "Trącić Myszką" [1], [2]) oraz podczas różnych lokalnych festiwali i wydarzeń z branży IT ("rabbIT" oraz "Noc Naukowców"). Podczas wymienionych wydarzeń autor z wykorzystaniem sprzętu multimedialnego oraz prototypu własnego systemu algorytmicznego komponowania muzyki oddziaływał na uczestników pokazów, a także dawał możliwość kreowania własnych koncepcji muzycznych opisując i udostępniając autorskie metody generowania kompozycji. Akcje promowania wyników badań były prowadzone również wśród najmłodszych słuchaczy czego przykładem może być Festiwal Nauki i Sztuki w Stargardzie, podczas którego autor prezentował możliwość algorytmizacji procesu tworzenia muzyki za pomocą komputera [35]. Popularyzowanie autorskiej koncepcji odbywało się również przy okazji wydarzeń kulturalnych takich jak wernisaże fotografii, w których tło muzyczne stanowiły kompozycje utworzone za pomocą autorskich metod generowania muzyki.

8. Akceptacja wyników przez społeczność

Czasopisma i materiały konferencyjne, w których recenzenci oceniali wyniki badań cząstkowych:
— *Metody Informatyki Stosowanej*, Polska Akademia Nauk (2008),
— *Modele inżynierii teleinformatyki.* Wybrane zastosowania 4, Wydawnictwo Uczelniane Politechniki Koszalińskiej (2009),

— *Modele inżynierii teleinformatyki*. Wybrane zastosowania 5, Wydawnictwo Uczelniane Politechniki Koszalińskiej (2010),
— *Przegląd Elektrotechniczny*, Wydawnictwo Czasopism i Książek Technicznych SIGMA-NOT Spółka z o.o (2012),
— *IEEE Xplore Digital Library*, IEEE (2012),
— *Journal of Applied Computer Science*, Politechnika Łódzka (2012),
— *Przegląd Elektrotechniczny*, Wydawnictwo Czasopism i Książek Technicznych SIGMA-NOT Spółka z o.o (2015),
— *AIP Conference Proceedings* 1648, AIP Publishing (2015).

Konferencje na których zaprezentowano oraz przedyskutowano wyniki badań cząstkowych:
— *Krajowa Konferencja Studentów i Młodych Pracowników Nauki*, Darłowo (2009),
— *Krajowa Konferencja Studentów i Młodych Pracowników Nauki*, Darłowo (2010),
— *International Conference System Modelling and Control*, Łódź (2011),
— *Advanced Computer Systems*, Międzyzdroje (2012),
— *International Symposium on Multimedia Applications and Processing* (w ramach konferencji *FedCIS*), Wrocław (2012),
— *International Conference on Numerical Analysis and Applied Mathematics*, Rodos (2014).

9. Główne osiągnięcia pracy

Do głównych autorskich osiągnięć pracy należy zaliczyć:
— opracowanie metod algorytmicznego komponowania muzyki na podstawie wzorców w aspekcie muzyki tła, wariancie zakładającym tworzenie kompozycji o strukturze ciągłej oraz strukturze zbudowanej na podstawie zdefiniowanej formie opisu jej składowych,
— opracowanie algorytmu generowania wzorców muzycznych na podstawie zdefiniowanych parametrów wejściowych dla tego procesu,
— opracowanie algorytmu generowania linii melodycznej na podstawie wejściowej sekwencji akordów,
— opracowanie algorytmu generowania linii basowej na podstawie danej sekwencji melodycznej.

10. Struktura i układ pracy

Rozprawa doktorska liczy 195 stron oraz składa się z pięciu rozdziałów a także glosariusza, spisu rysunków, spisu tablic, załącznika w postaci listingu kodów źródłowych oraz załącznika opisującego zawartość dołączonej do pracy płyty CD. W rozprawie ulokowano glosariusz zawierający 47 pojęć. Ponadto praca zawiera 58 rysunki, 34 tabele oraz 24 listingi kodów. Bibliografia liczy 108 pozycji.

W rozdziale 1 dokonano teoretycznego wprowadzenia do zagadnienia algorytmicznego komponowania muzyki oraz przeanalizowano historycznie istotne rozwiązania z tej dziedziny, ujęte w literaturze światowej. Dla potrzeby prawidłowego zrozumienia przez Czytelnika problematyki zawartej w pracy, autor dokonał przeglądu głównych zagadnień związanych z teorią muzyki, w wymiarze wystarczającym do przyswojenia treści zawartych w pracy. Podsumowaniem dokonanego

przeglądu rozwiązań i odniesienia do problemu badań jest sformułowanie celu pracy doktorskiej oraz hipotezy badawczej.

W rozdziale 2 przedstawiono koncepcję algorytmicznego komponowania muzyki bez definiowania formy utworu (muzyka tła o strukturze ciągłej) oraz koncepcję uwzględniającą formalny opis struktury kompozycji. Ponadto przedstawiono główne komponenty procesu komponowania muzyki, zaproponowane przez Bilesa w [40] (rozdział 2.1).

Rozdział 2.1 opisuje pierwszy wariant komponowania, który zakłada generowanie kompozycji muzycznej, bazujące na generowaniu wzorców muzycznych i łączeniu ich w jedną kompozycję. Wzorce muzyczne są generowane za pomocą parametrów wejściowych, takich jak długość wzorca, wektor opisujący sekwencję wysokości dźwięków motywu muzycznego, wektor opisujący możliwe rozmieszczenie dźwięków w ramach wzorca, itp. Kompozycja muzyczna tworzona jest natomiast z wygenerowanych wcześniej wzorców, których możliwość połączeń definiuje macierz, utworzona na podstawie analizy właściwości poszczególnych wzorców (ich podobieństwa).

Rozdział 2.2 opisuje drugi wariant komponowania, który bazuje na opisie formy utworu końcowego (liczbie i kolejności łączenia głównych fragmentów utworu) oraz procesie generowania kompozycji, wykorzystującym wygenerowane wcześniej wzorce muzyczne, których łączenie uwzględnia zdefiniowaną wcześniej formę kompozycji. Proces generowania wzorców ma charakter odmienny, ponieważ w odróżnieniu od pierwszego wariantu, głównym elementem nie jest motyw muzyczny, a wygenerowana (na podstawie reguł) progresja akordów, dla której tworzona jest melodia i dalsze partie instrumentalne.

Rozdział 2.3 porusza aspekt doboru instrumentów muzycznych oraz syntezy kompozycji wynikowej za pomocą syntezatora programowego.

W rozdziale 3 autor skupił się na prezentacji przykładowej implementacji prototypów koncepcji przedstawionych w rozdziałach 2.2 i 2.3. Rozdział 3.1 stanowi opis kolejnych elementów implementacji koncepcji generowania muzyki o strukturze ciągłej wraz ze szczegółowym opisem kodów źródłowych zawartych w dodatku do pracy. Analogicznie rozdział 3.2 zawiera opis implementacji koncepcji generowania muzyki, uwzględniającej opis jej struktury. W rozdziale 3.3 przedstawiono etap syntezy kompozycji z wykorzystaniem oprogramowania SynthFont.

Eksperymenty i wyniki badań zostały szczegółowo opisane w rozdziale 4. W rozdziale 4.1 autor przedstawił eksperymenty dotyczące generowania muzyki tła bez definiowania formy kompozycji. W ramach badań skupiono się na obserwacji następujących parametrów:

— modyfikacji długości dźwięków muzycznych,
— modyfikacji długości wzorca,
— modyfikacji liczności zbiorów wygenerowanych oraz wykorzystanych w kompozycji wzorców muzycznych,
— modyfikacji struktury zawierającej liczby możliwych powtórzeń wzorca w kompozycji.

Rozdział 4.2 zawiera opis obserwacji następujących parametrów:

— modyfikacji struktury opisującej formę kompozycji,
— modyfikacji struktury opisującej skalę muzyczną wykorzystaną w procesie komponowania,
— modyfikacji struktury opisującej możliwe przesunięcia czasowe dźwięków wchodzących w skład wzorców muzycznych,

— modyfikacji struktury opisującej możliwe długości dźwięków wykorzystane w budowie wzorca,
— modyfikacji liczby i czasu trwania akordów.

W rozdziale 5 zaprezentowano konkluzje oraz perspektywy rozwoju i wdrożeń koncepcji omawianych w dysertacji. Ujęto w nim proces publikowania otrzymanych wyników w aspekcie naukowego rozwoju autora. W obrębie dalszych perspektyw opisano aplikację "Wirtualne Laboratorium Muzykologa" opracowaną przez autora w ramach projektu TEWI [49], która może stanowić wstęp do utworzenia złożonego systemu algorytmicznego komponowania muzyki generatywnej na podstawie wzorców muzycznych pozyskanych z wejściowego zbioru gotowych utworów. We wnioskach końcowych autor dokonał podsumowania obu koncepcji komponowania oraz wykazał aktualność problemu naukowego.

11. Koncepcje algorytmicznego komponowania muzyki

Przed przystąpieniem do opisywania autorskich koncepcji systemów algorytmicznego komponowania muzyki, należy w sposób formalny przedstawić szereg zadań składających się na proces komponowania muzyki, ujęty przez Bilesa w [40] na stronach 34-36 i stanowiący przedmiot badań niniejszej dysertacji. Zadania te zostały podzielone na grupy, których struktura została przedstawiona w tab. 3. Porządek przedstawionych zadań procesu komponowania został przyjęty arbitralnie przez Bilesa i nie odzwierciedla struktury tworzenia muzyki według modelu kaskadowego.

Ważnym aspektem procesu komponowania w systemach algorytmicznego komponowania muzyki jest generowanie motywu melodii (tab. 3). Częstym podejściem jest wykorzystanie generatora melodii, który, bazując na zdefiniowanych bądź wyuczonych regułach generowania, wytwarza krótkie (najczęściej kilku- bądź kilkunastonutowe) fragmenty monofoniczne. Dodatkowo generowanie melodii może zostać rozbite na podzadania dotyczące tworzenia (w sposób niezależny) sekwencji wartości rytmicznych oraz wysokości dźwięków. Niektóre systemy oferują możliwość jednoczesnego generowania sekwencji rytmiczno-częstotliwościowych (np. [3]).

Część systemów algorytmicznego komponowania muzyki generuje jedynie sekwencje rytmiczne, najczęściej o długości jednego taktu, który jest traktowany jako wzorzec, a następnie umieszczany i powielany w różnych częściach kompozycji. Systemy takie swoją funkcjonalność opierają na idei tzw. maszyny perkusyjnej (z ang. *drum machine*) - urządzeniu, które przechowuje w pamięci (lub syntezuje w czasie rzeczywistym) próbki dźwięków instrumentów perkusyjnych, wykorzystywanych następnie do tworzenia jedno- bądź wielotaktowych wzorców zapętlonych w celu uzyskania podkładu perkusyjnego. W takich systemach wysokości dźwięku nie są interpretowane jako częstotliwości lecz jako określone identyfikatory instrumentów wykorzystywanych podczas komponowania taktu.[40]

W procesie komponowania kolejnym ważnym zagadnieniem jest rozszerzanie idei melodycznych (tab. 3), które ogólnie można zdefiniować jako serie transformacji motywu początkowego, wykorzystane do generowania motywu docelowego. Zdefiniowane serie transformacji mogą generować wariacje motywu (np. [8], [9], [23]), wzorce rytmiczne ([11]) lub łączyć krótkie motywy w dłuższe linie melodyczne ([9], [8]). Punktem wejściowym w takich systemach jest motyw składający się z kilku lub kilkunastu nut, który jest następnie wykorzystany do tworzenia dłuższej struktury.

Generowanie motywów melodii	Sekwencji wysokości dźwięków bez rytmu; Sekwencji rytmicznych bez określenia wysokości dźwięków; Sekwencji dźwięków z informacją o wysokości i rytmie.
Rozszerzenie ideii melodycznych	Generowanie wariacji i melodii kontrapunktowych; Łączenie fragmentów melodii w dłuższe elementy.
Harmonizacja	Generowanie partii harmonii; Generowanie zmian akordów.
Aranżacja	Definiowanie indywidualnych partii dla danego muzyka; Definiowanie instrumentów przypadających na indywidualne partie.
Struktura	Wykorzystywanie wysokiego poziomu opisu (top-down); Budowanie lub ewaluacja struktury od podstaw (bottom-up); Podejście hybrydowe w definiowaniu struktury (top-down i bottom-up).

Tab. 3: Podstawowe zadania procesu komponowania (na podstawie [40]). Opracowanie własne.

Harmonizacja (tab. 3) stanowi ważny element procesu komponowania muzyki i w ramach różnych systemów algorytmicznego komponowania polega najczęściej na tworzeniu głosu sopranowego, altowego, tenorowego oraz basowego dla danej melodii. Trudność polega na tym, że wszystkie wymienione głosy muszą być traktowane jako indywidualne melodie, z jednoczesnym zachowaniem harmonii wertykalnej (harmonii pomiędzy melodią oraz pozostałymi głosami). Można więc potraktować zagadnienie harmonizacji melodii jako proces planowania (problem optymalizacyjny) bazujący na zdefiniowanych wcześniej regułach następstwa dźwięków w czasie oraz ich rozmieszczenia w pionie (wybrzmiewania w tej samej chwili, z zachowaniem zasad harmonii). W przypadku gdy wejściem do systemu jest zarówno melodia (do której ma zostać dobrana harmonizacja) oraz progresja akordów, problem jest relatywnie prosty ([34]). Sytuacja komplikuje się, gdy wejściem systemu komponowania jest jedynie melodia. W takim przypadku stosuje się zazwyczaj metody bazujące na systemie reguł lub sieciach neuronowych, jak np. [44].

Aranżacja (tab. 3) stanowi problem doboru fragmentów utworu do określonych partii instrumentów muzycznych. Częściowo problem ten jest rozwiązywany w systemach rozbudowujących (rozszerzających) melodie oraz harmonizujących wejściowy materiał muzyczny. Istnieją jednak systemy ([9]), w których wyodrębniony został osobny moduł funkcjonalny odpowiedzialny za aranżację (tzw. aranżer).

Ostatnim i prawdopodobnie najtrudniejszym zagadnieniem procesu komponowania muzyki jest tworzenie logicznej struktury utworu (tab. 3). Niektóre systemy algorytmicznego komponowania muzyki poruszają się w płaszczyźnie niskiego poziomu opisu kompozycji (na ogół na poziomie opisu pojedynczej frazy utworu). Istnieją również systemy, które są w stanie generować w pełni rozwinięte kompozycje, poruszając się po różnych płaszczyznach opisu struktury utworu (np. od frazy

po formę) [8]. Systemy takie mają najczęściej charakter interaktywny i pozwalają (lub często wymagają) zdefiniowania przez użytkownika struktury na najwyższym poziomie opisu kompozycji w sposób manualny ([48]) bądź częściowo zautomatyzowany ([8], [9]). Kształtowanie struktury w systemach komponowania jest realizowane również z wykorzystaniem różnych technik i metod obliczeń ewolucyjnych takich jak agenty, automaty komórkowe lub PSO (optymalizacja rojem cząsteczek) ([6], [31]).

Przedstawiona analiza procesu komponowania ujęta przez Bilesa w [40] pozwala autorowi dokonać próby ulokowania swojej koncepcji systemu algorytmicznego komponowania muzyki na tle innych rozwiązań jako systemu, w którym:

— sekwencja melodii generowana jest z uwzględnieniem informacji na temat wysokości i długości dźwięków ją tworzących;
— rozszerzenie melodii polega na łączeniu jej fragmentów w dłuższe elementy, z zastosowaniem zdefiniowanych reguł;
— generowanie harmonii bazuje na regułach zmian akordów związanych, na bazie których tworzona jest melodia wiodąca;
— etap aranżacji ujęty jako zdefiniowanie instrumentów do określonych partii utworu pozostaje pod kontrolą użytkownika;
— struktura utworu definiowana jest z poziomu wysokiej abstrakcji opisu formy kompozycji (top-down), a możliwe połączenia fragmentów utworu w jedną kompozycję wynikają z analizy właściwości tych fragmentów (bottom-up).

11.1. Komponowanie bez definiowania formy generowanego utworu

Generowanie muzyki nieposiadającej skrystalizowanej formy (tzn. początku, rozwinięcia i zakończenia) jest zagadnieniem dość typowym dla muzyki tła, której cechą charakterystyczną są powtarzające się fragmenty muzyczne i ich wariacje. Taki sposób generowania muzyki może być porównany do generowania dużej tekstury obrazu z relatywnie małych fragmentów. [39]

W przyjętej koncepcji systemu wyjściowa kompozycja muzyczna tworzona jest z mniejszych fragmentów muzycznych, zwanych wzorcami muzycznymi. Wzorzec jest w tym podejściu zdefiniowany jako macierz nut zaproponowana w [26], której przykład zaprezentowano w tab. 4, przy czym dodatkowo macierz ta posiada stałą i jednakową dla wszystkich wierszy wartość długości dur_W nuty (w jednostkach bpm, i sekundach). Uproszczenie to jest podyktowane sposobem generowania wzorca.

Onset (bpm)	Dur. (bpm)	MIDI ch.	Pitch	Vol.	Onset (sec.)	Dur. (sec.)
0	0.4167	1.00	60.00	100.00	0	0.25
0.4167	0.4167	1.00	62.00	100.00	0.25	0.25
0.8333	0.4167	1.00	64.00	100.00	0.50	0.25
1.2500	0.4167	1.00	65.00	100.00	0.75	0.25
1.6667	0.4167	1.00	67.00	100.00	1.00	0.25
2.0833	0.4167	1.00	69.00	100.00	1.25	0.25
2.5000	0.4167	1.00	71.00	100.00	1.50	0.25
2.9167	0.4167	1.00	72.00	100.00	1.75	0.25

Tab. 4: Przykładowa macierz nut w środowisku Matlab. Opracowanie własne.

Przykładową macierz o rozmiarze [8x7], opisującą oktawę od dźwięku $C4$ (60) do $C5$ (72), przedstawiono w tab. 4. Każdy wiersz macierzy reprezentuje sparametryzowany dźwięk muzyczny. Kolejne kolumny zawierają wartości parametrów danego dźwięku. W pierwszej kolumnie znajdują się wartości rozpoczęcia dźwięku w jednostkach bpm, związanych z tempem odgrywania utworu. Druga kolumna zawiera wartości długości dźwięków w jednostkach bpm. W kolumnie trzeciej występują numery wykorzystanego przez dany dźwięk kanału *MIDI* (standard *MIDI* zakłada jednoczesne wykorzystanie 16 kanałów, przy czym w przytoczonym przykładzie z tab. 4 wykorzystywany jest jedynie kanał nr 1). Kolumna czwarta zawiera wartości wysokości dźwięków zapisane w postaci liczb *MIDI*, które odpowiadają poszczególnym częstotliwościom dźwięków wykorzystywanych w muzyce (ich zakres jest ograniczony od 0 do 127). W kolumnie piątej znajdują się wartości głośności dźwięków zawarte w przedziale od 0 (cisza) do 127 (najgłośniej). Kolumny szósta i siódma zawierają wartości przedstawione w kolumnach pierwszej i drugiej wyrażone w sekundach.

Macierz nut przedstawia zatem reprezentację pewnej sekwencji komend *MIDI*, która w przyjętym podejściu analizuje jedynie ograniczony zbiór komunikatów *MIDI* przechowywanych w postaci pliku *MIDI*.

Wzorzec muzyczny W jest zdefiniowany jako macierz nut MN:

$$W = MN_{m\times 7}, m = \begin{cases} \frac{l_W}{dur_W} & \text{jeżeli } \frac{l_W}{dur_W} \in N \\ \left\lfloor \frac{l_W}{dur_W} \right\rfloor & \text{jeżeli } \frac{l_W}{dur_W} \notin N \end{cases}. \tag{1}$$

gdzie:
— m_{MN} oznacza liczbę wierszy macierzy nut (MN),
— l_W oznacza zadeklarowaną długość wzorca w jednostkach bpm,
— dur_W jest wartością długości dla wszystkich dźwięków zawartych w macierzy MN.

Mini-generator (oznaczony jako Mg) jest podstawowym komponentem w opisywanej koncepcji autorskiego systemu algorytmicznego komponowania muzyki na podstawie wzorców muzycznych. Definicja mini-generatora Mg została przedstawiona poniżej jako:

$$Mg = \{K, MM, WPN, dur_W, l_W, t\}, \tag{2}$$

gdzie:
— K oznacza klasę mini-generatora,
— MM - motyw muzyczny na bazie którego generowany jest wzorzec muzyczny W,
— WPN - wektor pozycji (wierszy macierzy MN) nut, w których mają być rozmieszczone dźwięki z motywu MM,
— dur_W - wartość długości dla wszystkich dźwięków generowanego wzorca W określona w jednostkach bpm,
— l_W- długość generowanego wzorca W określona w jednostkach bpm,
— t - tempo generowanego wzorca W określone w jednostkach bpm.

Występujące w wyrażeniu (2) oznaczenie K definiuje klasę do jakiej należy mini-generator Mg. Przynależność do danej klasy determinuje sposób wykorzystania mini-generatora Mg w procesie generowania wzorca muzycznego. W prezentowanym systemie zdefiniowane zostały następujące klasy mini-generatorów:
— generator monofoniczny,
— generator polifoniczny,

— generator perkusyjny.
Generator monofoniczny wykorzystywany jest do generowania monofonicznych fragmentów muzycznych takich jak np. melodia główna czy monofoniczna linia basowa. Generator polifoniczny generuje polifoniczne fragmenty muzyczne takie jak akord muzyczny. Generator perkusyjny to szczególna klasa mini-generatora, którego generowane sekwencje muzyczne (monofoniczne i polifoniczne) są interpretowane jako sekwencje perkusyjne.

Mini-generator *Mg* do generowania muzyki wykorzystuje strukturę, która stanowi kombinację motywu muzycznego wcześniej zdefiniowanego przez użytkownika. W podstawowej wersji prezentowanego systemu motyw stanowi wektor liczb naturalnych zawartych w przedziale od 0 do 127, traktowanych jako wysokość dźwięku w standardzie *MIDI*. Zbiór możliwych kombinacji zdefiniowanego motywu muzycznego stanowi macierz *KMM*. Macierz *KMM* składa się z k-elementowych kombinacji C n-elementowego motywu muzycznego *MM*, rozłożonych w macierzy o wymiarach m wierszy (liczba kombinacji) na k kolumn (liczba elementów motywu muzycznego)

$$KMM = \begin{bmatrix} {}_1C_k^n \\ {}_2C_k^n \\ \vdots \\ {}_mC_k^n \end{bmatrix}, k <= n. \tag{3}$$

Z macierzy *KMM* losowana jest dla mini-generatora *Mg* kombinacja motywu muzycznego zdefiniowana jako *MM*. Mini-generator *Mg* może również przyjmować *MM* w postaci stałej zdefiniowanej liczby naturalnej. Tak sparametryzowany *Mg* generuje fragmenty muzyczne, w których wszystkie dźwięki posiadają tę samą wysokość równą stałej *MM*. W zależności od klasy K oraz rodzaju *MM* (wektor lub skalar) zdefiniowane zostały różne reguły generowania wzorców.

Za rozmieszczenie dźwięków muzycznych w odpowiednich miejscach macierzy nut generowanej przez mini-generator odpowiada wektor pozycji dźwięków *WPN*. Struktura wektora *WPN* jest generowana na podstawie wyrażenia:

$$WPN = \{n : rs : l\}, n, rs < l. \tag{4}$$

Parametr n jest dowolną liczbą naturalną mniejszą od zdefiniowanej liczby wierszy l macierzy nut. Stała rs jest liczbą naturalną określającą krok wypełnienia wektora *WPN*. Dla przykładu: mając zdefiniowaną liczbę wierszy l (dźwięków) w macierzy jako 32, parametr n ustawiony na wartość 1 oraz stałą rs ustawioną na 4, wektor *WPN* będzie miał postać $WPN = \{1, 5, 9, 13, 17, 21, 25, 29\}$, co oznacza, że tylko dla zdefiniowanych pozycji *WPN* macierzy nut mini-generatora *Mg* może osadzać dźwięki muzyczne podczas tworzenia wzorca muzycznego. Rozmieszczenie dźwięków w macierzy wpływa na rytmikę wzorca muzycznego generowanego przez mini-generator i wykorzystanego następnie do generowania wyjściowej kompozycji.

Zdaniem autora najciekawsze efekty pracy omawianego systemu algorytmicznego komponowania muzyki uzyskuje się poprzez zmiany, a właściwie dopasowanie wszystkich wartości parametrów systemu. Aby potwierdzić swoją tezę autor zaprezentował kilka przykładowych wyników wygenerowanych na podstawie różnych konfiguracji ustawień. Wszystkie kompozycje znajdują się na płycie CD dołączonej do dysertacji.

Omawianym dalej przykładem jest kompozycja $s1-415.flac$, do syntezy której wykorzystano 6 kanałów *MIDI* z jedynie 2 instrumentami banku *Evanessence*2, tj. *Ukulele* oraz *Latin*(*Loops*). W procesie komponowania autor zdefiniował następujące wartości parametrów systemu:
— $dur = 0.25$ - długość dźwięku w macierzy NM wyrażona w sekundach,
— $pattern_l = 32$ - liczba wierszy (dźwięków) macierzy NM,
— $n_pattern = 128$ - liczba wzorców muzycznych wygenerowanych przez system,
— $u_pattern = 64$ - liczba wzorców muzycznych wykorzystanych w kompozycji,
— $rt = [4]$ - tablica rt powtórzeń.
Bazowym motywem dla wzorców muzycznych jest wektor wysokości dźwięków zdefiniowany w postaci: $p =< 50, 48, 50, 53, 55, 43, 41, 43, 36, 38 >$, który posłużył do budowy macierzy KMM. Macierz KMM składa się z trzech "części" zawierających kombinacje wygenerowane na podstawie trzech wersji wektora p: podstawowej, zwiększonej o wartość 4 (linia 10) oraz zwiększonej o wartość 7. Tak więc górna "część" macierzy KMM zbudowana jest na podstawie kombinacji wektora p, środkowa stanowi kombinacje wektora $p_1 =< 54, 52, 54, 57, 59, 47, 45, 47, 40, 42 >$ a dolna - kombinacje wektora $p_2 =< 57, 55, 57, 60, 62, 50, 48, 50, 43, 45 >$.

Konfiguracja mini-generatorów wykorzystanych w procesie komponowania została przedstawiona w tab. 5. Mini-generatory od Mg_1 do Mg_{11} należą do klasy generującej partie instrumentów perkusyjnych, natomiast mini-generatory od Mg_{12} do Mg_{18} generują partie melodii, przy czym w tym wypadku są to partie odtwarzane przez jeden instrument muzyczny. Dla urozmaicenia części rytmicznej kompozycji, autor zdefiniował dynamiczną strukturę motywów PM dla wzorców Mg_9, Mg_{10} i Mg_{11}.

Dla mini-generatora Mg_9 motyw PM przyjmuje postać wektora pm, zbudowanego na podstawie implementacji:

```
step = fix(pattern_l/ fix(rand(1)*rhytmic_step(7)+1));
vec = rhytmic_value{7}(1:step:end);
temppattern(vec,4) = [38:1:38+length(vec)-1];
```

Podobnie dla mini-generatora Mg_{10} motyw PM przyjmuje postać wektora $\widetilde{pm}$, zbudowanego na podstawie implementacji:

```
step = fix(pattern_l/ fix(rand(1)*rhytmic_step(7)+1));
vec = rhytmic_value{7}(1:step:end);
temppattern(vec,4) = [35:1:35+length(vec)-1];
```

Mini-generator Mg_{11} generuje dźwięki na podstawie wektora $\widehat{pm}$, zdefiniowanego wg. implementacji:

```
step = fix(pattern_l/fix(rand(1)*rhytmic_step(7)+1));
vec = rhytmic_value{1}(1:step:end);
temppattern(vec,4) = ones(1,length(vec))*38;
```

Dla tak zdefiniowanych mini-generatorów utworzona została architektura generowania kompozycji wg schematu przedstawionego na rys. 1. Bloki oznaczone przez Mg_x symbolizują zdefiniowane mini-generatory biorące udział w procesie generowania wzorca muzycznego. Od każdego bloku odchodzi pewna liczba odcinków

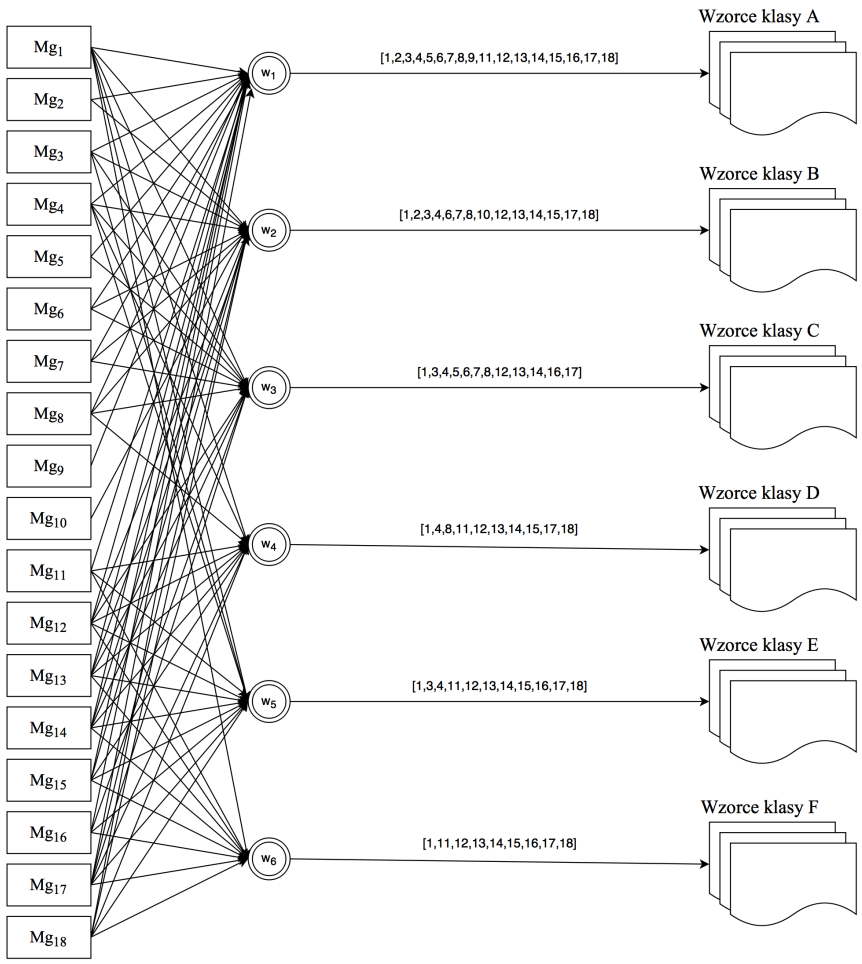
Mini-generator	Klasa	*rhytmic_step*	*rhytmic_value*	PM	Kanał *MIDI*
Mg_1	D	32	[1,5,9,...,29]	36	10
Mg_2	D	32	[2,5,8,...,32]	36	10
Mg_3	D	16	[2,5,8,...,32]	40	10
Mg_4	D	32	[1,2,...,32]	42	10
Mg_5	D	32	[2,5,8,...,32]	51	10
Mg_6	D	16	[16,20,24,28,32]	47	10
Mg_7	D	16	[16,19,22,...,31]	48	10
Mg_8	D	16	[1,5,9,...,29]	54	10
Mg_9	D	8	[17,21,25,29]	pm	10
Mg_{10}	D	8	[17,21,25,29]	$\widetilde{pm}$	10
Mg_{11}	D	8	[1,5,9,...,29]	$\widehat{pm}$	10
Mg_{12}	M	32	[2,6,10,...,30]	z KMM	1
Mg_{13}	M	8	[2,6,10,...,30]	z Mg_{12}	2
Mg_{14}	M	32	[1,6,11,...,31]	z Mg_{12}	3
Mg_{15}	M	16	[2,4,6,...,32]	z Mg_{12}	4
Mg_{16}	M	32	[3,6,9,...,30]	z Mg_{12}	5
Mg_{17}	M	32	[1,2,3,...,32]	z Mg_{12}	6
Mg_{18}	M	16	[2,5,8,...,32]	z Mg_{12}	7

Tab. 5: Parametry zdefiniowanych mini-generatorów dla kompozycji $s1 - 415.flac$. Opracowanie własne.

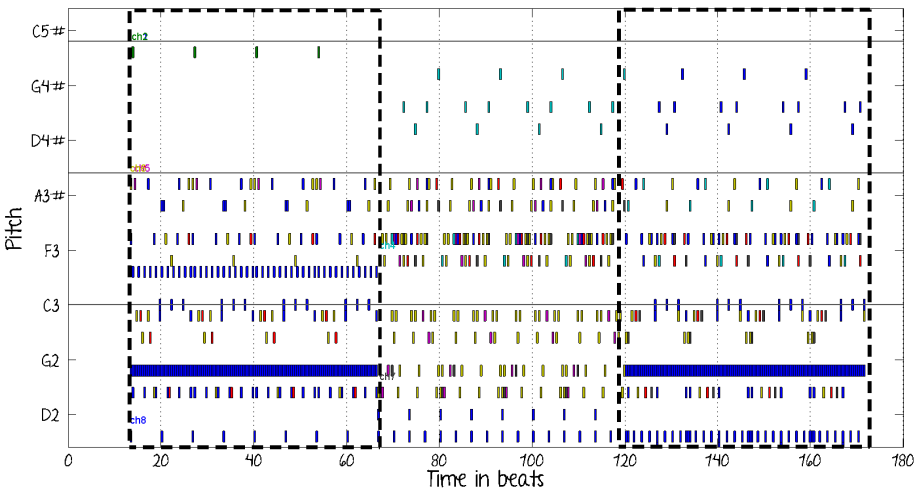
łączących się w jednym z sześciu węzłów w symbolizujących proces konkatenacji wzorca muzycznego.

Wygenerowane wzorce muzyczne posiadają strukturę złożoną z maksymalnie 32 dźwięków muzycznych, natomiast w kompozycji końcowej występują w postaci grup zawierających cztery powtórzenia wzorca. Właściwość tę widać szczególnie na rys. 2 przedstawiającym fragment kompozycji $s1 - 415.flac$ zbudowany z trzech wzorców. Podobieństwo w budowie jest zauważalne przede wszystkim pomiędzy pierwszą a ostatnią częścią przedstawianego fragmentu, w których mini-generatory partii perkusyjnych utworzyły zbliżone struktury rytmiczne (oznaczone na rys. 2).

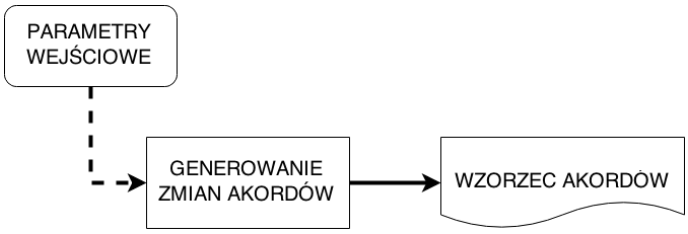
Podobieństwa w budowie poszczególnych grup wzorców są słyszalne również w kolejnych częściach przedstawianej kompozycji. Dodatkowo występujące po sobie kolejne fragmenty oscylują wokół podobnego motywu muzycznego (zbudowanego na podstawie wektora p, p_1 lub p_2). Utworzona kompozycja spełnia definicję muzyki tła (podanej we wcześniejszej części pracy) jednak, w opinii autora, jej "powtarzalność" i swego rodzaju "chaotyczność" mogą stanowić o słabych cechach prezentowanej koncepcji systemu. Podobne cechy komponowanej muzyki można usłyszeć w innych kompozycjach, umieszczonych na dołączonej do pracy płycie CD. Biorąc pod uwagę powyższe stwierdzenia oraz analizując wyniki przeprowadzonych eksperymentów, autor zaproponował inną, odmienną koncepcję algorytmicznego komponowania muzyki, uwzględniającą definicję formy generowanej kompozycji.



Rys. 1: Schemat generowania architektury klas wzorców przez zdefiniowane mini-generatory Mg dla kompozycji $s1 - 415.flac$. Opracowanie własne.



Rys. 2: Wykres pianolowy fragmentu kompozycji $s1 - 415.flac$ z zaznaczonymi grupami wzorców wykazującymi podobieństwo budowy. Opracowanie własne.



Rys. 3: Generowanie zmian akordów. Opracowanie własne.

11.2. Komponowanie uwzględniające opis formy generowanego utworu

W poprzednim rozdziale poświęconym przeprowadzonym badaniom nad użytecznością algorytmicznego systemu komponowania muzyki na podstawie wzorców, autor wyeksponował rolę mini-generatora, jako podstawowej funkcjonalnej części koncepcji tworzenia muzyki bez określania jednoznacznej formy wynikowej kompozycji. Tak wykreowana sztuczna twórczość posiada znamiona muzyki tła, w której dopatrzyć się można teoretycznie nieskończonej struktury, ograniczonej wartościami parametrów systemu. To nieograniczenie formy lokuje omawiany system w dziedzinie systemów komponowania raczej prymitywnych, w których zaszyta muzyczna "wiedza teoretyczna" jest minimalna. Z tego względu autor podjął próbę poszerzenia opisywanej wcześnie koncepcji komponowania o możliwość definiowania finalnej struktury formy kompozycji i jej elementów.

Najważniejsza zmiana dotyczy sposobu generowania wzorców, z których tworzona jest wyjściowa kompozycja muzyczna. Dodatkowo autor zrezygnował z etapu oceny podobieństwa wygenerowanych wzorców, z racji przyjętego mechanizmu generowania i łączenia wzorców w ramach kompozycji.

Biles w swoich rozważaniach na temat modelowania procesu komponowania muzyki poprzez zdefiniowanie elementarnych zadań, które kształtują ten proces wspomniał, że rozmieszczenie zadań ulokowanych w tab. 3 zostało wykonane w sposób arbitralny. System, który w sposób mniej lub bardziej kompleksowy generuje kompozycje muzyczne, może bazować na różnych zadaniach procesu komponowania, nie koniecznie w kolejności przedstawionej w tab. 3. Z tego względu autor w tej części pracy zaproponował odmienne podejście do problemu algorytmizacji procesu komponowania muzyki, bazując na harmonizacji jako wejściowym etapie generowania wzorców.

Na rys. 3 zaprezentowano w sposób uproszczony proces powstawania wzorca zawierającego wygenerowaną sekwencję akordów

$$W_A = \{A_p, n_A, l_A, S, R_z\}. \tag{5}$$

Do parametrów wejściowych (rys. 3) kształtujących proces generowania wzorca sekwencji akordów W_A zaliczone zostały:
— A_p - wygenerowany akord początkowy,
— n_A - liczba akordów, które mają być wygenerowane w sekwencji,
— l_A - długość akordu (w jednostkach bpm),
— S - skala muzyczna, na podstawie której generowana będzie sekwencja,
— R_z - zbiór reguł tworzenia zmian (następczości) generowanych akordów.

Sekwencja akordów (stopnie)	**Sekwencja akordów (nazwy)**
I →IV (kadencja plagalna)	C-dur →F-dur
I →V (kadencja doskonała)	C-dur →G-dur
I →IV →V (triada harmoniczna)	C-dur →F-dur →G-dur
I →IV →V →IV	C-dur →F-dur →G-dur →F-dur
I →V →vi →IV	C-dur →G-dur →a-mol →F-dur
I →ii →IV →V	C-dur →d-mol →F-dur →G-dur
I →ii →IV	C-dur →d-mol →F-dur
I →VI →ii →V	C-dur →A-dur →d-mol →G-dur
I →VI →IV →V	C-dur →A-dur →F-dur →G-dur
IV →I →IV →V	F-dur →C-dur →F-dur →G-dur

Tab. 6: Przykładowe kadencje oparte na tonacji C-dur. Opracowanie własne.

Akord początkowy A_p jest macierzą nut MN zdefiniowaną jako

$$MN = \begin{bmatrix} p_1 m \\ \vdots \\ p_i n \end{bmatrix}, \tag{6}$$

przy czym p określa pojedynczy sparametryzowany dźwięk muzyczny w macierzy MN, natomiast m i n oznaczają stopnie wykorzystanej skali muzycznej (wtedy $m, n \leq 7$) oraz i wyznacza liczbę dźwięków wykorzystanych ($i \geq 3$) w początkowym akordzie A_p.

Parametr n_A określa liczbę kolejnych akordów A wygenerowanych w ramach wzorca W_A. Długość akordu l_A wyraża rozciągnięcie w czasie pojedynczego akordu A w jednostkach bpm.

Szerszego wyjaśnienia wymaga przedstawiony we wzorze (5) zbiór reguł R_z, na podstawie którego budowana jest sekwencja akordów W_A. Zasady generowania sekwencji zostały zapisane w postaci reguł bazujących na kadencjach, które wykorzystują pojęcie *funkcji harmonicznej* budującej lub wygaszającej napięcie w utworze muzycznym.

W tab. 6 przedstawione zostały przykładowe kadencje zbudowane w tonacji C-dur. W kolumnie "Sekwencja akordów (stopnie)" znajdują się sekwencje akordów oznaczonych za pomocą cyfr rzymskich, które odpowiadają stopniom skali. Kolumna "Sekwencja akordów (nazwy)" zawiera przykłady sekwencji akordów dla tonacji C-dur.

Akordy zbudowane na kolejnych stopniach skali w systemie dur-moll tworzą funkcje harmoniczne, wśród których najczęściej występującymi są: *tonika*, *subdominanta* i *dominanta*. Obok wymienionych funkcji w systemie dur-moll występują także tzw. funkcje poboczne, czyli funkcje zbudowane na stopniach skali innych niż I, IV i V. Do tej grupy funkcji należą *medianta* i *submedianta* oraz akordy poboczne zbudowane na II i VII stopniu skali.

Zbiór reguł R_z ujęty we wzorze 5 został przez autora zdefiniowany jako macierz tranzycji (7), w której następstwo pojawienia się określonego akordu j (funkcji harmonicznej) uwarunkowane jest zdefiniowanym prawdopodobieństwem p_{ij} oraz aktualnie wygenerowanym akordem i. W nagłówkach kolumn i wierszy macierzy tranzycji znajdują się cyfry rzymskie, określające funkcje harmoniczne przedstawione w tab.

6. Dla każdego przejścia czyli wygenerowania akordu j będącego następnym w sekwencji po akordzie i, definiowane jest prawdopodobieństwo p_{ij} wystąpienia takiego zdarzenia. W macierzy celowo nie zostały podane konkretne wartości prawdopodobieństwa, gdyż zmieniają się one w zależności od aktualnego stopnia kadencji oraz numeru kolejnego wygenerowanego akordu w ramach progresji. Dla przykładu: prawdopodobieństwo przejścia ze stopnia I do IV stopnia skali w sytuacji gdy akord ten jest jedynym w progresji (początek procesu generowania) jest inne niż w przypadku gdy akord ten został powtórzony w kolejnym kroku (w progresji akordów znajdują się dwa akordy zbudowane na I stopniu skali).

	I	II	III	IV	V	VI	VII
I	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}
II	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{26}	p_{27}
III	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}
IV	p_{41}	p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}
V	p_{51}	p_{52}	p_{53}	p_{54}	p_{55}	p_{56}	p_{57}
VI	p_{61}	p_{62}	p_{63}	p_{64}	p_{65}	p_{66}	p_{67}
VII	p_{71}	p_{72}	p_{73}	p_{74}	p_{75}	p_{76}	p_{77}

Tab. 7: Macierz tranzycji generowania sekwencji akordów. Opracowanie własne.

Budowa kadencji bazująca na przyjętej koncepcji macierzy tranzycji sekwencji akordów pozwala definiować zarówno typowe konfiguracje sekwencji (np. triada harmoniczna: tonika $\rightarrow$ subdominanta $\rightarrow$ dominanta), jak i sekwencje niestandardowe (np. funkcje poboczne oraz tzw. wtrącenia, czyli dodanie dominanty do akordu z istniejącej sekwencji).

Kolejnym krokiem w etapie generowania wzorca wykorzystywanego w przyjętej koncepcji jest konkatenacja wzorca W_A zdefiniowanego we wzorze 5 ze wzorcem melodii W_M wygenerowanej na podstawie sekwencji akordów W_A. Wzorzec melodii W_M jest macierzą nut wygenerowaną na podstawie parametrów przedstawionych w postaci

$$W_M = \{W_A, \{d_1, d_2, \ldots, d_n\}, \{o_1, o_2, \ldots, o_m\}, \{v_1, v_2, \ldots, v_s\}\}, \tag{7}$$

gdzie:

— W_A - wzorzec sekwencji akordów, na podstawie którego wygenerowana będzie melodia,

— $\{d_1, d_2, \ldots, d_n\}$ - wektor możliwych długości dźwięków występujących w melodii,

— $\{o_1, o_2, \ldots, o_m\}$ - wektor możliwych przesunięć dźwięków występujących w melodii,

— $\{v_1, v_2, \ldots, v_s\}$ - wektor możliwych głośności dźwięków występujących w melodii.

Algorytm 1 przedstawia koncepcję generowania melodii bazującą na analizie wzorca W_A sekwencji akordów. Mocnym ograniczeniem narzuconym przez autora jest wykorzystanie podczas tworzenia melodii wyłącznie dźwięków wchodzących w skład analizowanych akordów. Głównym elementem algorytmu 1 jest okienkowanie sygnału (wzorca W_A) oknem O_A o zadanej rozpiętości czasowej $< t_p, t_k >$, które zawiera wydzielony fragment wzorca. Podczas przesuwania okna wzdłuż wzorca W_A generowane są kolejne dźwięki D_M melodii, która ostatecznie przybiera postać macierzy nut wzorca W_M. Jest to podejście częściowo stochastyczne, częściowo deter-

Algorytm 1 Generowanie melodii na podstawie sekwencji akordów. Opracowanie własne.

1. Oblicz długość l_A wzorca W_A,
2. Zdefiniuj pustą macierz nut dla wzorca melodii W_M i linii basowej W_B,
3. Ustaw indeks przeszukiwania i wzorca W_A na początek analizowanego fragmentu: $i = 0$,
4. Zdefiniuj okno analizy wzorca O_A: początek okna: $t_p \leftarrow i$, koniec okna: $t_k \leftarrow i + \Delta t$,
5. Dopóki $i + \Delta t \leq l_A$:
   a) Wybierz z okna analizy wzorca zbiór występujących we fragmencie wysokości dźwięków $\{h_1, h_2, \ldots, h_k\}$,
   b) Ustaw indeks przeszukiwania j okna O_A na: $j \leftarrow t_p$,
   c) Ustaw indeks położenia generowanego dźwięku melodii $r_{D_M} \leftarrow t_p$,
   d) Dopóki $j \neq t_k$:
      i. Dopóki $r_{D_M} \leq t_k$:
         A. Wylosuj z $\{h_1, h_2, \ldots, h_k\}$ wysokość dźwięku h,
         B. Wylosuj z $\{d_1, d_2, \ldots, d_n\}$ długość dźwięku d,
         C. Wylosuj z $\{o_1, o_2, \ldots, o_m\}$ przesunięcie dźwięku o,
         D. Wylosuj z $\{v_1, v_2, \ldots, v_s\}$ głośność dźwięku v,
         E. Utwórz dźwięk muzyczny D_M o wysokości h, długości d, głośności v oraz definiowanym przesunięciem w czasie o,
         F. Dodaj dźwięk muzyczny D_M do macierzy nut wzorca melodii W_M: $W_M \leftarrow \{W_M; D_M\}$,
      ii. Zwiększ indeks r_{D_M} o wartość położenia dźwięku muzycznego o: $r_{D_M} \leftarrow r_{D_M} + o$,
   e) Generuj wzorzec linii basowej W_B na podstawie zawartości wzorca W_M i aktualnego położenia okna O_A,
   f) Zwiększ indeks przeszukiwania j: $j \leftarrow j + i$,
   g) Przesuń indeks przeszukiwania i: $i \leftarrow i + \Delta t$,
   h) Przesuń okno analizy wzorca O_A: $t_p \leftarrow i$, $t_k \leftarrow i + \Delta t$,
6. Zwróć macierz nut wzorca W_M.

ministyczne, ponieważ dźwięki melodii losowane są ze zbioru możliwych dźwięków występujących w akordach, natomiast ich rozłożenie w czasie jest z góry ograniczone występowaniem akordów w macierzy wzorca W_A.

Uzupełnieniem nowej koncepcji generowania wzorca muzycznego jest wprowadzenie mechanizmu tworzenia wzorca linii basowej (oznaczonego jako W_B) na podstawie generowanego wzorca W_M. Wymienioną funkcjonalność, o której wspomniano w kroku 5e w algorytmie 1, przedstawiono szerzej w algorytmie 2.

Generowanie linii basowej polega na fragmentacji linii melodycznej wzorca W_M oraz transponowaniu o m oktawy w dół. Ten prosty zabieg powoduje poszerzenie spektrum częstotliwości dźwięków muzycznych w wygenerowanej kompozycji, a linia basowa jest ściśle związana z melodią główną. Należy przy tym dodać, że dla każdego okna analizy proces generowania linii basowej rozpoczyna się na nowo, co w konsekwencji powoduje urozmaicenie rytmiczne występujących we wzorcu W_B dźwięków.

W ramach przedstawionego procesu generowania wzorców autor założył również

Algorytm 2 Generowanie linii basowej na podstawie melodii. Opracowanie własne.

1. Wybierz fragment wzorca melodii W_M zgodnie z położeniem okna analizy O_A,
2. Zdefiniuj parametr k jako liczbę losową z przedziału $< 1, \ldots, l >$, gdzie l stanowi liczbę wszystkich dźwięków muzycznych w wybranym fragmencie wzorca W_M,
3. Z wybranego fragmentu macierzy nut wzorca melodii wybierz n dźwięków, pomijając co k-ty element,
4. Transponuj o m oktawy w dół dźwięki wybrane w punkcie 3,
5. Dodaj transponowane dźwięki do macierzy wzorca W_B linii basowej.

możliwość stosowania zdefiniowanego wcześniej (2) mini-generatora. Należy jednak przedstawić zasadniczą różnicę pomiędzy wykorzystaniem mini-generatora w koncepcji generowania utworu muzycznego, w którym nie została zdefiniowana forma kompozycji, a koncepcją, w której opis formy utworu stanowi centralny punkt procesu komponowania.

W poprzednim rozdziale mini-generatory zostały przedstawione jako mechanizm generowania wszelkich wzorców muzycznych, wykorzystywanych później w procesie generowania kompozycji wyjściowej. Koncepcja komponowania utworu z uwzględnieniem definiowania jego formy wykorzystuje mini-generatory, w których parametr MM przedstawiony we wzorze (2) przyjmuje postać skalara - wartości odpowiadającej konkretnemu instrumentowi perkusyjnemu zdefiniowanemu w standardzie General Midi. Dodatkowo parametr K oznaczający klasę mini-generatora jest wspólny dla wszystkich mini-generatorów wykorzystanych podczas generowania wzorca i oznaczony jako klasa generatora perkusyjnego.

Istotną zmianą w stosunku do przedstawionej w poprzednim rozdziale koncepcji generowania kompozycji wyjściowej jest pominięcie etapu selekcji wzorców i generowanie utworu muzycznego na podstawie modelu kompozycji $\hat{U}$

$$\hat{U} = \{PW, F = \{i_{f_1}; i_{f_2}; \ldots; i_{f_n}\}, S, I, \{MG_{f_1}; MG_{f_2}; \ldots; MG_{f_n}\}\}, \tag{8}$$

w którym:

— PW - oznacza zbiór parametrów wejściowych kompozycji,
— $F = \{i_{f_1}; i_{f_2}; \ldots; i_{f_n}\}$ - oznacza zbiór identyfikatorów zdefiniowanych fragmentów muzycznych $f_1 \ldots f_n$ opisujący ich rozmieszczenie w kompozycji,
— S - oznacza zbiór kanałów *MIDI* wykorzystanych w kompozycji,
— I - oznacza zbiór instrumentów *MIDI* przypisanych do zdefiniowanych w kompozycji kanałów *MIDI*,
— $\{MG_{f_1}; MG_{f_2}; \ldots; MG_{f_n}\}$ - oznacza zbiór mini-generatorów MG przypisanych do poszczególnych fragmentów kompozycji $f_1 \ldots f_n$, przy czym n stanowi liczbę zdefiniowanych fragmentów kompozycji.

Do parametrów wejściowych PW kształtujących kompozycję wyjściową systemu należą:

— tempo utworu (liczba całkowita, wyrażająca szybkość odtwarzania kompozycji w jednostkach bpm),
— przyjęta skala muzyczna, w której tworzona będzie kompozycja,
— definicje fragmentów muzycznych, z których będzie generowana kompozycja,
— długości i liczby akordów wchodzących w skład zdefiniowanych fragmentów kompozycji.

Skala muzyczna została zdefiniowana jako wektor 8-mio elementowy, w którym znajdują się wartości interwałów kolejnych stopni skali, wyrażone w liczbie półtonów. Pierwszym elementem tego wektora jest liczba 0, która opisuje pierwszy stopień skali - wartość odniesienia dla pozostałych stopni. Ostatni element wektora został zdefiniowany jako interwał oktawy wyrażony wartością 12 półtonów. Przykładowo dla naturalnej skali durowej zdefiniować można następujący wektor interwałów: $< 0, 2, 4, 5, 7, 9, 11, 12 >$. W takim wypadku dla dźwięku $C4$, dla którego wysokość wyrażona wg standardu *MIDI* wynosi 60, otrzymujemy naturalną skalę C-dur w następujący sposób:

$60+ < 0, 2, 4, 5, 7, 9, 11, 12 >=< 60, 62, 64, 65, 67, 69, 71, 72 >$, przy czym ostatnim elementem tak utworzonego wektora jest dźwięk $C5$, należący już do kolejnej oktawy.

Fragment muzyczny stanowi zdefiniowaną część kompozycji przedstawioną w postaci macierzy nut zbudowanej na podstawie wybranych wzorców muzycznych. Formalny zapis fragmentu jest zdefiniowany jako

$$f = \{i_f, S_f, I_f, W_{A_f}, W_{M_f}, W_{B_f}, MG_f\}, \tag{9}$$

gdzie:

— i_f jest identyfikatorem danego fragmentu f,
— S_f jest zbiorem kanałów midi wykorzystanych do wygenerowania fragmentu f,
— I_f stanowi zbiór instrumentów *MIDI* przypisanych do zdefiniowanych we fragmencie f kanałów *MIDI*,
— W_{A_f} jest wzorcem sekwencji akordów przypisanym do fragmentu f,
— W_{M_f} jest wzorcem melodii wygenerowanej na podstawie W_{A_f} i przypisanym do fragmentu f,
— W_{B_f} jest wzorcem linii basowej wygenerowanej na podstawie W_{M_f} i przypisanym do fragmentu f,
— MG_f stanowi zbiór mini-generatorów przypisanych do fragmentu f.

Identyfikator i_f fragmentu służy do jego lokalizowania w zbiorze F (8) opisującym formę generowanej przez system kompozycji muzycznej. Przykładowo dla zdefiniowanych fragmentów kompozycji f_1, f_2, f_3 i f_4 można zdefiniować ich identyfikatory: $i_1 = 1$, $i_2 = 2$, $i_3 = 3$ oraz $i_4 = 4$. Dla tak zdefiniowanych danych zbiór F może przykładowo przybrać postać $F = [1, 1, 2, 2, 2, 2, 3, 3, 2, 2, 2, 2, 3, 3, 4, 4]$. Zbiór tenkolekcję należy analizować od lewej do prawej strony, przy czym kolejność elementów zbioru jest istotna. W przedstawionym przykładzie zbioru F można zauważyć, że fragment f_1 o identyfikatorze $i_1 = 1$ występuję dwukrotnie na początku kompozycji. Następnie naprzemiennie występują dwie grupy fragmentów: czterokrotnie powtórzony fragment f_2 ($i_2 = 2$) oraz dwukrotnie powtórzony fragment f_3 ($i_3 = 3$). Kompozycja kończy się dwukrotnym wystąpieniem fragmentu f_4 o identyfikatorze $i_4 = 4$.

S_f i I_f (10) stanowią podzbiory występujących kolejno w (8) zbiorów: S kanałów *MIDI* oraz I instrumentów *MIDI* wykorzystanych w kompozycji $\hat{U}$. Na tym etapie badań autor założył występowanie w danym fragmencie jednego wzorca sekwencji akordów W_{A_f}, na którym bazuje mechanizm generowania melodii i linii basowej. Istnieje natomiast możliwość generowania więcej niż jednej melodii i linii basowej dla wygenerowanego wzorca W_{A_f}, co zostało szerzej przedstawione w części dysertacji poświęconej eksperymentom i wynikom badań.

Ostatnimi parametrami ujętymi w zbiorze parametrów wejściowych PW są długości akordów (wyrażone w jednostkach bpm) zdefiniowane niezależnie dla każdego fragmentu kompozycji oraz liczby akordów w ramach wzorców sekwencji akordów W_{A_f} związanych z fragmentami kompozycji. Wymienione parametry służą do określenia długości wzorca sekwencji akordów l_{WA_f} wyrażonej wzorem

$$l_{WA_f} = l_A * n, \tag{10}$$

gdzie:
— l_A - długość akordu (w bpm),
— n - liczba akordów generowanych we wzorcu W_{A_f}.
Długość wzorca sekwencji akordów W_{A_f} jest wykorzystywana podczas generowania melodii i linii basowej (algorytm 2). Odpowiednio dobrane l_A oraz n we wzorcach sekwencji umożliwiają generowanie kompozycji w różnym metrum (l_A) oraz z różnym poziomem złożoności fragmentów kompozycji (n).

Na potrzeby eksperymentów zdefiniowane zostały cztery części kompozycji oznaczone kolejnymi liczbami całkowitymi. Poniżej zostały zdefiniowane następujące wartości parametrów systemu:
— $form = [1, 1, 3, 3, 1, 1, 4, 3, 3, 1]$ - wektor definiujący strukturę kompozycji (kolejność poszczególnych części kompozycji),
— $tempo = 160$ - tempo kompozycji wyrażone w jednostce bpm,
— $scale = [0, 2, 3, 5, 7, 8, 10, 12]$ - wektor opisujący skalę muzyczną, w której wygenerowana zostanie kompozycja (w tym przypadku jest to skala naturalna molowa),
— $possible_location = [0, 8]$ - wektor możliwych lokalizacji dźwięków we wzorcach muzycznych,
— $possible_location_snare = [0, 16]$ - j.w. z tym, że zdefiniowany dla konkretnego instrumentu (werbla),
— $possible_offset = [0]$ - wektor możliwych przesunięć dźwięków w czasie,
— $possible_offset_snare = [8]$ - j.w. z tym, że zdefiniowany dla werbla,
— $offset_mel = [0, 0.5]$ - wektor przesunięć dźwięków zdefiniowany dla melodii,
— $possible_duration_mel1 = [0.25]$ - wektor możliwych długości dźwięku dla melodii,
— $possible_duration_mel2 = [0.5]$ - alternatywny wektor możliwych długości dźwięku dla melodii,
— $possible_duration = [0.25, 0.5]$ - kolejny wektor możliwych długości dźwięku,
— $lp = 4$ - długość akordu wyrażona w jednostce bpm,
— $step_1 = 4$ - liczba akordów w ramach wzorca muzycznego,
— $step_2 = 8$ - alternatywna liczba akordów,
— $fill = [36, 50, 48, 47, 45, 43, 40, 49, 57, 66, 65, 40, 39]$ - pomocniczy wektor wykorzystany przez mini-generator do symulacji przejścia perkusyjnego.
W przedstawionej konfiguracji system generuje cztery rodzaje "fragmentów" kompozycji, przy czym nie wszystkie z nich muszą być wykorzystane w wynikowym utworze. Dodatkowo niektóre wartości parametrów, takie jak wektor długości dźwięku czy możliwych przesunięć, zostały zdefiniowane wielokrotnie w celu wprowadzenia urozmaiconych zbiorów wartości wykorzystanych przez różne mini-generatory. Domyślnym akordem dla generowania progresji akordów jest akord molowy $G_3\#$ (składający się z dźwięków: $G_3\#, B_3, D_4\#$), na podstawie którego budowane są kadencje w poszczególnych częściach kompozycji.

Mini-generator	Klasa	*rhytmic_step*	*rhytmic_value*	PM	Kanał *MIDI*
Mg_1	D	pl	onset	36	10
Mg_2	D	pls	onset_s	40	10
Mg_3	D	pl	onset	42	10
Mg_4	D	pl	onset	51	10
Mg_5	D	pl	onset_h1	pm_h	10
Mg_6	D	pl	onset	49	10
Mg_7	D	pl/2	onset_h1	pm_h	10
Mg_8	D	pl/2	onset_h2	pm_h	10
Mg_9	D	pl	onset	54	10

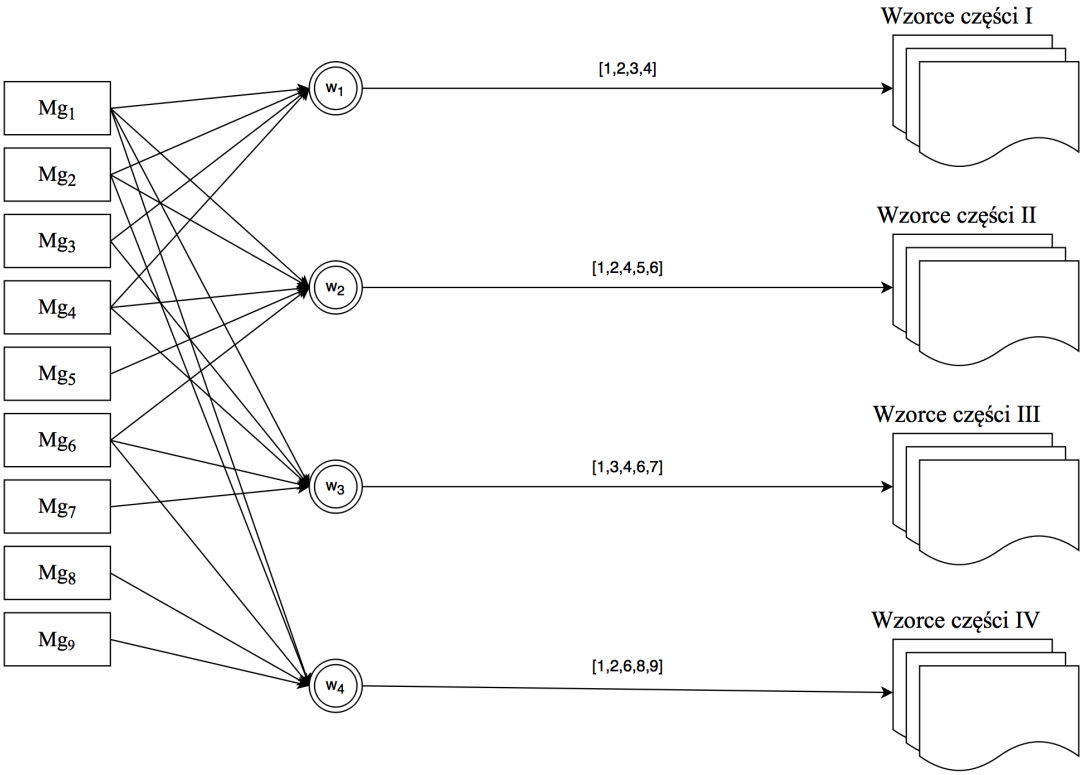
Tab. 8: Parametry zdefiniowanych mini-generatorów. Opracowanie własne.

W przyjętej koncepcji model mini-generatora odgrywa kluczową rolę podczas budowania wzorca muzycznego dla instrumentów perkusyjnych. W tab. 8 przedstawiona została konfiguracja mini-generatorów zdefiniowanych w badanym systemie komponowania, a dla potrzeb przeprowadzonych eksperymentów autor zdefiniował następujące zmienne przedstawione na listingu poniżej (kod Matlab):

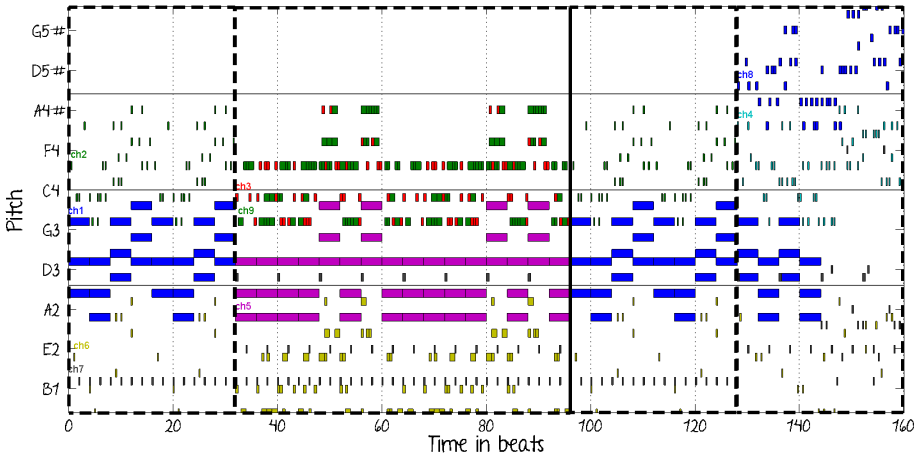
```
lpos = length(possible_location);
lpos_snare = length(possible_location_snare);
lof = length(possible_offset);
lof_snare = length(possible_offset_snare);
...
pl = possible_location(floor(rand(1)*lpos+1));
pls = possible_location_snare(floor(rand(1)*lpos_snare +1));
po = 0;
pos = possible_offset_snare(floor(rand(1)*lof_snare+1));

onset = [1+po:pl:size(notematrix,1)];
onset_s = [1+pos:pls:size(notematrix,1)];
onset_h1 = [1+64:pl:size(notematrix,1)];
onset_h2 = [1+po:pl/2:size(notematrix,1)];

fill = [36,50,48,47,45,43,40,49,57,66,65,40,39];
lfill = length(fill);
pm_h = fill(fix(rand(1,8)*lfill)+1);
```

Zmienne *pl* i *pls* określają wartości parametru *rhytmic_step* losowane z wektorów lokalizacji dźwięków. Zmienne *onset*, *onset_s*, *onset_h*1 i *onseet_h*2 reprezentują wektory parametru *rhytmic_value* - lokalizacji dźwięków w obrębie wzorca muzycznego. Mini-generatory Mg_5, Mg_7 i Mg_8 zostały zdefiniowane do budowania ciągów uderzeń w różne instrumenty perkusyjne (podane w konwencji *MIDI* w wektorze *fill*), których strukturę opisuje zmienna *pm_h*.

Wraz z definicją mini-generatorów z tab. 8 zdefiniowana została architektura tworzenia klas wzorców dla poszczególnych części kompozycji muzycznej. Każdą część z osobna można traktować jako odrębną klasę, dla której generowane są wzorce muzyczne wg połączeń mini-generatorów przedstawionych na rys. 4.



Rys. 4: Schemat generowania architektury klas wzorców przez mini-generatory zdefiniowane w tab. 8. Opracowanie własne.



Rys. 5: Fragment kompozycji wygenerowanej na podstawie ustawień ze strony 23. Opracowanie własne.

Fragment przykładowej kompozycji $k_raw_421.flac$ zbudowanej na podstawie przedłożonej konfiguracji początkowej zaprezentowano na rys. 5. Wskazany wykres pianolowy został podzielony na obszary, w których występują fragmenty zdefiniowane wg parametru $form$. I tak, w pierwszym od lewej strony obszarze znajdują się dwa powtórzenia fragmentu pierwszego, w kolejnym dwa drugiego, w trzecim następuje ponowne dwukrotne wystąpienie fragmentu pierwszego, natomiast w ostatnim obszarze znajduje się fragment czwarty. Dodatkowo rys. 5 ukazuje różne

długości wygenerowanych wzorców muzycznych: fragment trzeci składa się ze wzorców zbudowanych na podstawie ośmiu wystąpień akordów (parametr $step_2$), podczas gdy fragment pierwszy składa się ze wzorców zawierających cztery akordy (parametr $step_1$).

12. Spis publikacji własnych

1. Mazurowski Ł.: *Generative electronic background music system*, AIP Conf. Proc. 1648, 660004 (2015), http://dx.doi.org/10.1063/1.4912880, Rodos, 22-28 września 2014,
2. Mazurowski Ł.: *Algorithmic composition transformational-generative system for background music generation*, Przegląd Elektrotechniczny, R. 91, nr 2, 2015, s. 112-117,
3. Mazurowski Ł.: *Przegląd systemów kształcenia głosu śpiewaczego z wizualnym sprzężeniem zwrotnym*, w *Metody komputerowe w zastosowaniu do analizy i wspierania kształcenia głosu śpiewaczego* (red. E. Półrolniczak), ISBN 978-83-936799-3-5, Stowarzyszenie Przyjaciół Wydziału Informatyki w Szczecinie, 2013, s. 9-24,
4. Mazurowski Ł.: *Pozyskiwanie informacji multimedialnych – bieżące podejścia i zastosowanie*, Firma i Rynek 2013/1 (44), Zeszyty Naukowe Zachodniopomorskiej Szkoły Biznesu w Szczecinie, s. 41-50,
5. Mazurowski Ł., Wiliński A.: *Computer models of similarity in musical pieces based on the multi-dimensional patterns*, Journal of Applied Computer Science, vol. 20, nr 2, 2012, s. 79-94,
6. Mazurowski Ł.: *Computer models for algorithmic music composition*, in Federated Conference on Proceedings of Computer Science and Information Systems (FedCSIS), 2012, IEEE Xplore Digital Library , s. 733-737,
7. Mazurowski Ł.: *An algorithmic composition - the chosen aspects*, Przegląd Elektrotechniczny, Vol 2012, Nr 10b, s. 243-247,
8. Mazurowski Ł.: *Synteza dźwięku w technologii SoundFont*, Modele inżynierii teleinformatyki. Wybrane zastosowania 5, Wydawnictwo Uczelniane Politechniki Koszalińskiej, 2010,
9. Mazurowski Ł.: *Łańcuchy Markowa w systemach generowania muzyki sztucznej*, Modele inżynierii teleinformatyki. Wybrane zastosowania 4, Wydawnictwo Uczelniane Politechniki Koszalińskiej, 2009,
10. Mazurowski Ł., Wiliński A.: *Koncepcja generowania utworu muzycznego przez maszynę - akt sztucznej twórczości opartej na algorytmach genetycznych*, Metody Informatyki Stosowanej 3/2008, PAN, 2008

13. Podsumowanie

W koncepcji algorytmicznego komponowania muzyki na podstawie wzorców z pominięciem formalnego opisu formy generowanej kompozycji autor przeanalizował cztery parametry przyczyniające się do zmiany jej struktury, tj. długość dźwięków muzycznych, długość wzorca, strukturę zbioru wzorców oraz strukturę tablicy powtórzeń wzorców.

Przedstawione wyniki zmian długości dźwięków muzycznych jak i całych wzorców sugerują ich wpływ na powtarzalność fragmentów kompozycji oraz złożoność jej struktury. Przykłady wygenerowanych kompozycji ukazują możliwości systemu w zakresie tworzenia muzyki o zmiennej dynamice i złożoności struktury. Przy przyjętym stałym tempie utworu oraz stałej długości wzorca muzycznego, zmiana parametru długości tworzących go dźwięków powoduje zmianę w dynamice całej kompozycji: im krótsze dźwięki, tym bardziej dynamiczny wydaje się być efekt końcowy pracy systemu. Natomiast w sytuacji gdy przyjęta została stała długość dźwięków muzycznych i tempa kompozycji, zwiększenie długości wzorca powoduje zwiększenie ogólnej złożoności kompozycji - wzorce muzyczne zbudowane są z większej liczby dźwięków, które mogą tworzyć bardziej rozbudowane tematy muzyczne.

Autor w ramach doświadczeń przeprowadził również próby zmian proporcji pomiędzy zbiorem wygenerowanych a wykorzystanych wzorców kompozycji muzycznej oraz zmian budowy tablicy powtórzeń wzorca rt. W wyniku przeprowadzenia kilku procesów komponowania otrzymano wynik nacechowany wielokrotnymi powtórzeniami tego samego wzorca, co odsłania pewną słabość opisywanej koncepcji systemu w przypadku słabo urozmaiconego zbioru wzorców muzycznych. Jednocześnie inne kompozycje ukazują różne aspekty powtórzeń wzorców muzycznych w ujęciu kompletnej kompozycji oraz chaotyczności jej struktury w przypadku rozbudowy tablicy powtórzeń rt.

Przedstawione wyniki badań nad możliwościami systemu algorytmicznego komponowania muzyki na podstawie wzorców muzycznych i opisu formy utworu, pokazują odmienny charakter tworzonej muzyki, w porównaniu z systemem bazującym wyłącznie na mini-generatorach, a którego analizy dokonano w rozdziale 11.1. Wyodrębnionych 6 parametrów sterujących pracą systemu stanowi jedynie podstawę możliwości modyfikowania końcowej formy kompozycji.

Modyfikacja struktury wektora formy razem ze zdefiniowaniem algorytmu generującego poszczególne części kompozycji to podstawowe narzędzia kreujące finalną postać dzieła muzycznego utworzonego przez proponowany system. Zmiana kolejności elementów formy może być pomocna w eksploracji ciekawych i nieszablonowych wyników, a sposób definiowania części kompozycji wprowadza szerokie możliwości kreowania dużych fragmentów formy.

Istotnym elementem kształtującym cześć harmoniczną generowanego utworu jest parametr opisujący skalę muzyczną oraz akord bazowy, na których zbudowane są sekwencje akordów w kompozycji wynikowej. Wynika to ze specyfiki przyjętej koncepcji generowania linii melodycznych oraz linii basowych, które niejako "śledzą" przebieg sekwencji akordu, a budulcem dla nich jest budowa poszczególnych elementów tej progresji. Zatem z pomocą odpowiednio dobranych skali oraz akordu bazowego, można zbudować utwór np. o charakterze smutnym (w tonacjach molowych) lub bardziej radosnym (w tonacjach durowych).

Aspekt rytmiczny jest obok harmonii i melodyki jedną z najważniejszych cech generowanego utworu, dlatego też w przyjętej koncepcji zdefiniowane zostały odpowiednie parametry, pozwalające zmieniać jego strukturę rytmiczną. Modyfikacjom rytmicznym mogą podlegać zarówno poszczególne dźwięki (poprzez zmiany parametrów długości i przesunięć dźwięków w obrębie wzorca muzycznego lub części kompozycji) jak i całe fragmenty muzyczne, których charakterystykę można modyfikować za pomocą odpowiednio dobranych wartości liczby i czasu trwania akordów. Zmiana wymienionych parametrów prowadzić może również do zmiany artykulacji

dźwięków, co w połączeniu z ciekawym brzemieniem instrumentu syntezującego daną partię, może dać ciekawe efekty końcowe.

Parametry zdefiniowane w systemie w różnym stopniu pokrywają domeny muzyki związane z procesem komponowania utworu. Autor prezentując wyniki swoich badań miał na celu jedynie przedstawienie określonych technik sterowania tym procesem w obrębie przyjętych koncepcji.

Nawiązując do komponentów problemu badawczego sformułowanych na str. 5 - celu i hipotezy badawczej należy ocenić, że zostały one osiągnięte. Cel opisany jako *opracowanie modelu komputerowego systemu komponowania muzyki użytkowej (posiadającej wartość rynkową), w którym do generowania oryginalnych utworów muzycznych wykorzystuje się metody autorskie oparte na metodach sztucznej inteligencji,* został bez wątpienia osiągnięty. W pracy zostały szczegółowo opisane owe autorskie metody z podkreśleniem ich nowatorskich aspektów. Autor uważa je za własne, pozbawione wad prawnych i innowacyjne wg często przywoływanego dziś kryterium - są nowe i mają wartość (metodyczną, techniczną i, być może, rynkową).

Hipoteza opisana jako przypuszczenie, że *bazując na metodach sztucznej inteligencji, możliwe jest opracowanie takich algorytmów komponowania utworów muzycznych, których efektem będą utwory atrakcyjne w odbiorze i posiadające strukturę poprawną w aspekcie teorii muzyki*, zdaniem autora także została dowiedziona, przy czym owa teoria muzyki stanowi koncept reguł i struktur zdefiniowanych w ramach struktur określających możliwe sąsiedztwa między różnymi zdarzeniami muzycznymi: dźwiękami, całymi akordami czy fragmentami kompozycji.

Literatura

[1] Trącić myszką. komputerowy kompozytor w domowym studiu nagraniowym. Audycja radiowa, 2011. [dostęp online 04.03.2017: `http://www.myszka.org/archiwum-myszki/tracic-myszka-2011/06112011-komputerowy-kompozytor-w-domowym-studiu-nagraniowym`].

[2] Trącić myszką. synteza i przetwarzanie dźwięku. Audycja radiowa, 2011. [dostęp online 04.03.2017: `http://www.myszka.org/archiwum-myszki/tracic-myszka-2011/17102011-synteza-i-przetwarzanie-dzwieku`].

[3] Biles J. A. Genjam: a genetic algorithm for generating jazz solos. *Proceedings of ICMC'94*, pages 131–137, 1994. University of Michigan Library, MI: MPublishing.

[4] Biles J. A. Genjam: Evolutionary computation gets a gig. In *Third Conference on Information Technology Curriculum*, Rochster, New York, 2002.

[5] Biles J. A. Genjam in transition: from genetic jammer to generative jammer. In *International Conference on Generative Art*, Milan, Włochy, 2002.

[6] Gartland-Jones A. Musicblox: A real-time algorithmic composition system incorporating a distributed interactive genetic algorithm. *Applications of Evolutionary Computing: EvoWorkshops 2003. LNCS*, 2611:490–501, 2003.

[7] Bel B. Migrating musical concepts: an overview of the bol processor. *Computer Music Journal*, 22(2).

[8] Jacob B. Algorithmic composition as a model of creativity. *Organised Sound*, 1.

[9] Jacob B. Composing with genetic algorithms. In *Proceedings of the 1995 International Computer Music Conferencee*, 1995.

[10] Bentley P. Blackwell T.M. Improvised music with swarms. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 2, pages 1462–1467, 2002.

[11] Ariza C. Prokaryotic groove: Rhythmic cycles as real-value encoded genetic algorithms. In *Proceedings of the 2002 International Computer Music Conferencee*, 2002.

[12] Jain S. Wadhwa S. Chaudhury S., Nag T.C. Role of sound stimulation in reprogramming brain connectivity. *Journal of Biosciences*, 38(3):605–614, 2013.

[13] Miikkulainen R. Chen C. C. J. Creating melodies with evolving recurrent neural network. *Proceedings of IJCNN'01*, 2001.

[14] Krumhansl C.L. *Cognitive Foundations of Musical Pitch.* Oxford University Press, New York, 1990.

[15] Dale N. Cockerton T., Moore S. Cognitive test performance and background music. *Perceptual and Motor Skills*, 85:1435–1438, 1997.

[16] Hofstadter D.R. Cope D. *Virtual music: computer synthesis of musical style.* Mass.: MIT Press, Cambridge,, 2001.

[17] Conklin D. Music generation from statistical models. In *Proceedings of the AISB 2003 - Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.

[18] Cope D. *Computers and Musical Style.* A-R Editions, 1991.

[19] Cope D. Recombinant music using the computer to explore musical style. *IEEE Computer*, 24:22–28, 1991. IEEE Computer Society Press Los Alamitos.

[20] Cope D. Computer modeling of musical intelligence in emi. *Computer Music Journal*, 16:69–83, 1992. The MIT Press.

[21] Cope D. *Experiments in musical intelligence.* A-R Editions, 1996.

[22] Cope D. *The algorithmic composer.* A-R Editions, 2000.

[23] Ralley D. Genetic algorithm as a tool for melodic developmenty. In *Proceedings of the 1995 International Computer Music Conferencee*, 1995.

[24] Collins English Dictionary. background music. Dostęp on-line: 26-07-2014.

[25] Ames Ch. Domino M. Understanding music with ai. chapter Cybernetic composer: an overview, pages 186–205. MIT Press, Cambridge, MA, USA, 1992.

[26] Toiviainen P. Eerola T. Mir in matlab: The midi toolbox. *Proceedings of 5th International Conference on Music Information Retrieval*, pages 22–27, 2004.

[27] Correa D.C. et al. Neural network based systems for computer-aided musical composition:supervised x unsupervised learning. *SAC*, pages 1738–1742, 2008. ACM.

[28] Dubnov S. et al. Using machine-learning methods for musical style modeling. *IEEE Computer*, 36(10), 2003.

[29] Lartillot O. et al. Automatic modeling of musical style. *Proceedings of ICMC'02*, 2002.

[30] Schoner B. Farbood M. Analysis and synthesis of Palestrina-style counterpoint using Markov chains. In *International ComputerMusic Conference*, 2001.

[31] Copley P. Gartland-Jones A. The suitability of genetic algorithms for music composition. *Contemporary Music Review*, 22:43–55, 2003.

[32] Tuson A. Geraint Wiggins G. Papadopoulos G., Phon-amnuaisuk S. Evolutionary methods for musical composition. In *In International Journal of Computing Anticipatory Systems*, 1998.

[33] Slimane M. Guéret C., Monmarché N. Ants can play music. In Blum C.-Gambardella L.M. Mondada F. Stützle T. Dorigo M., Birattari M., editor, *Ant Colony Optimization and Swarm Intelligence*, volume 3172 of *Lecture Notes in Computer Science*, pages 310–317. Springer Berlin Heidelberg, 2004.

[34] Ayres L. Horner A. Harmonisation of musical progression with genetic algorithms. In *Proceedings of the 1995 International Computer Music Conferencee*, 1995.

[35] Info-wiadomości. Festiwal nauki i sztuki oraz dzień otwarty w i liceum ogólnokształcącym im. adama mickiewicza w stargardzie, 2014. [dostęp online 08.03.2017: `http://www.wiadomosci.rii.pl/index.php?H=11&HH=9970`].

[36] Renkewitz F. Kampfe J., Sedlmeier P. The impact of background music on adult listeners: A meta-analysis. *Psychology of Music*, 39:424–448, 2011.

[37] University College London. Compgc20 - computer music, 2016. [dostęp online 07.03.2017: `http://www.cs.ucl.ac.uk/students/syllabus/msccs/gc20_computer_music/`].

[38] Hoggar S. McAlpine K., Miranda E.R. Making music with algorithms: A case-study system. *Computer Music Journal*, 23(2).

[39] Jung-Ju C. Min-Joon Y., In-Kwon L. *Background Music Generation Using Music Texture Synthesis*, volume 3166 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004.

[40] Biles J. A. Miranda E. R. *Evolutionary Computer Music*. Springer-Verlag, Londyn, 2007.

[41] Tominaga K. Miura T. An approach to algorithmic music composition with an artificial chemistry. In *Proc. of the German Workshop on Artificial Life*, 2006.

[42] Reagan P. *Computer music generation via decision tree learning. Master Thesis*. Department of Computer Science, Carnegie Mellon University, 1999.

[43] Wiggins G. Papadopoulos G. A genetic algorithm for generation of jazz melodie solos. *Proceedings of STEP'98*, pages 7–9, 1998.

[44] Wiggins G. Phon-Amnuasiuk S. The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system. In *Proceedings of AISB 99*, 1999.

[45] Brown A. R. Opportunities for evolutionary music composition. *Proceedings of the ACMC'02*, pages 27–34, 2001.

[46] Parncutt R. A perceptual model of pulse salience and metrical accent in musical rhytms. *Music Percepction*, 11:409–464, 1994.

[47] Chiu S.C. Shan M.K. Algorithmic compositions based on discovered musical patterns. *Multimedia Tools and Applications*, 46.

[48] Unemi T. Sbeat3: A tool for multi-part music composition by simulated breeding. In *Proceedings of the Eighth International Conference on Artificial Life (A-Life VIII)*, 2002.

[49] TEWI. Wirtualne laboratorium muzykologa, 2013. [dostęp online 07.03.2017: `http://tewi.p.lodz.pl`].

[50] Setomoto M. Tominaga K. An artificial-chemistry approach to generating polyphonic musical phrases. In *Applications of Evolutionary Computing*, volume 4976 of *Lecture Notes in Computer Science*, pages 463–472. Springer Berlin Heidelberg, 2008.

[51] Columbia University. Computer music center, 2017. [dostęp online 07.03.2017: `http://cmc.music.columbia.edu/cmc_courses/`].

[52] Griffith University. Generative music (3710qcm), 2016. [dostęp online 07.03.2017: `https://degrees.griffith.edu.au/Course/3710QCM`].

[53] Fayer M. Verbeurgt K., Dinolfo M. Extracting patterns in music for composition via markov chains. In *Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, IEA/AIE'2004, pages 1123–1132. Springer Springer Verlag Inc, 2004.

[54] Zabierowski W. Accord sequences generation in agreement with harmony tonal rules using cascade artificial neural networks. *Pomiary Automatyka Kontrola*, (11):1394–1396, 2011.

[55] Napieralski A. Zabierowski W. Generation of harmonic sequences in accordance to tonal harmony rules with artificial neural networks. *Elektronika : konstrukcje, technologie, zastosowania*, 50(11):22–24, 2009.

17.05.2017

Mazurski Łukasz