

Za veliku pomoć u rješavanju nedoumica na području glazbene problematike skladanja melodija zahvaljujem se Vlatki Mihelić.

Za pomoć i podršku u svim ostalim segmentima zahvaljujem se ocu Dušku i stricu Paji Mišljenčević. Uz njih je izrada ovog rada predstavljala iskreno zadovoljstvo i zabavu.

Uvod

1960. godine I. Rechenberg u svome djelu „Evolucijske strategije“ iznosi ideju evolucijskog računarstva. Ideja je prihvaćena od strane istraživača na području računalne znanosti, te se počinje intenzivno proučavati.

Genetski algoritmi su rezultat istraživanja John-a Holland-a i grupe njegovih kolega i sudenata sa sveučilišta. To su algoritmi koji svoj princip rada crpe iz prirodne evolucije. Najčešće se koriste se kod problema traženja optimalnih rješenja problema.

Ovaj završni rad koristi genetski algoritam za generiranje najskladnije melodije koja se gradi usporedbom sa nekom zadanom melodijom. Što znači najskladnija melodija? Najskladnija po kojem kriteriju? Kako se uopće određuje sklad nekog glazbenog djela? Koje su posebnosti primjene genetskog algoritma u području generiranja glazbe? Postoje li neka ograničenja u primjeni i koji su kompromisi morali biti uvedeni? Odgovori na ova i još mnoga pitanja slijede u nastavku teksta.

Problem skladanja melodija

Skladanje melodija po mnogočemu se razlikuje od ostalih problema koje rješava genetski algoritam. Glavna razlika koja taj problem čini veoma kompliciranim a ujedno i veoma intrigantnim jest u tome što glazba nije uniformna i ne može se strogo definirati matematičkim formulama. Mnogi se s ovom tvrdnjom možda neće složiti. Naročito kada postoje tolike studije koje nalaze poveznice upravo između glazbe i matematike. U glazbenoj školi djeca imaju predmet pod nazivom „solfeggio“. Svako dijete koje ima matematičkog talenta zasigurno ima odličnu ocjenu iz tog predmeta. Barem iz onog teoretskog djela. No, je li to dovoljno? Je li Mozart uistinu bio vrsan matematičar i je li to ključ njegove genijalnosti?

Ovaj završnom rad nudi odgovor i na ovo pitanje pomalo filozofske prirode. Želi se pokazati da temelj koji skladbu poput Beethovenove «Ode radosti» ili Vivaldijevih «Godišnjih doba» drži na okupu jest upravo teoretske prirode i zahtjeva mnogo proračuna koji su sasvim slični matematički formulama. Ali ono što takvoj skladbi daje njenu besmrtnost je nešto što niti najbolji matematičar ovoga svijeta ne može formalno definirati. I upravo u tome se krije sav čar glazbene umjetnosti i sva njena mističnost i ljepota.



Slika 1. Serenada

Evolucija u prirodi

Charles Darwin je ustanovio da je glavni pokretač evolucije prirodna selekcija. Priroda određuje uvjete za život. One jedinke koje imaju bolji genetski materijal prilagodit će se tim uvjetima i opstati, dok će slabije jedinke s vremenom izumrijeti. Kako jedna vrsta izumire, druga vrsta križanjem stvara nove potomke i na taj način broj živih bića uvijek ostaje barem približno konstantan. Povremene mutacije koje se u prirodi javljaju mijenjaju izgled i osobine živih bića i dodatno pridonose procesu evolucije na Zemlji.

1.1. Genetski materijal – geni, kromosomi i molekula DNA

Naše se tijelo sastoji od šest bilijuna stanica, od kojih svaka u sebi nosi 23 para homolognih kromosoma. Unutar svakog para homolognih kromosoma jedan je kromosom naslijeđen od oca, a drugi od majke. Homologni kromosomi sadrže gene za ista svojstva na istom mjestu, tj. lokusu. Npr. na jednom od tih parova nalazi se gen koji određuje boju očiju. Razlika je u tome što gen naslijeđen od oca može određivati plavu boju očiju a od majke smeđu boju očiju. Od tih svojstava samo će se jedno ispoljiti ovisno o tome koje svojstvo je dominantno a koje recesivno. U konkretnom slučaju potomak će imati smeđe oči jer je to dominantno svojstvo.

Strukturu molekule DNA otkrili su sredinom 20. st. Watson i Creek. DNA je molekula koja se sastoji od određenog slijeda nukleotida. Nukleotid je dušična baza na koju je vezan šećer deoksiriboza i fosfatna skupina. Dva lanca DNA se spajaju po principu komplementarnosti dušičnih baza (adenin se spaja sa timinom, a gvanin s citozinom). Određeni slijed parova dušičnih baza čini gen, odnosno prenosioca nasljednih svojstava.

Kromosomi su transportni oblik molekule DNA koja se nalazi u svakoj tjelesnoj stanici živog bića. Svaka molekula DNA dugačka je otprilike dva metra. Prilikom diobe molekula DNA se mora replicirati kako bi svaka od novonastalih stanica mogla dobiti po jedan njen primjerak. Odmah je jasno da neće biti jednostavno odvojiti dvije molekule dugačke dva metra na različite polove sićušne stanice. Zbog toga se DNA mora sažeti u svoj transportni oblik, već spomenuti kromosom.

1.2 Križanje i mutacija

Križanjem se nasljedna svojstva prenose sa parentalne generacije na potomstvo. Pritom roditelji na svoju djecu prenose genetski materijal, tj. svoje dobre i loše osobine. Slučajna promjena nekih svojstava jedinke naziva se mutacija. U genetskom algoritmu operacija mutacije je unarna operacija jer se promjene gena vrše nad jednom jedinom jedinkom. Za razliku od mutacije, operacija križanja je operacija višeg reda jer se nove jedinke stvaraju kombinacijom gena više različitih jedinki iz parentalne skupine.

Genetski algoritmi

Genetski algoritmi pripadaju skupini evolucijskih algoritama. Uz njih, u evolucijske algoritme ubrajaju se i genetsko programiranje, evolucijsko programiranje i evolucijske strategije. Navedeni algoritmi svoju ideju crpe iz procesa evolucije u prirodi koju koriste za rješavanje problema optimizacije. Simulacija evolucijskog procesa odvija se korištenjem svih važnih čimbenika procesa evolucije u prirodi. Kod genetskog algoritma ti čimbenici su sljedeći: populacija (jedinke), genetski operatori (križanje i mutacija), funkcija dobrote koja predstavlja sposobnost preživljavanja jedinke u prirodi, te selekcijski mehanizam. Promoviranjem željenih osobina jedinke moguće je usmjeravati evoluciju u željenom smjeru i na taj način pronaći optimalno rješenje problema čak i u slučajevima kada ne postoji eksplicitno rješenje.

Primjena genetskih algoritama moguća je u slučajevima kada se problem može opisati kao pretraživanje ili optimizacija proizvoljnih podataka, te je poznat način mogućeg mjerenja uspješnosti svakog pojedinog rješenja. Iako genetski algoritam pretražuje i optimizira zadani skup, rješenje ne predstavlja uvijek optimum procesa, te je potrebno pažljivo konstruirati okruženje evolucije kako bi se zaista dobili rezultati bliski rješenju. Prvi korak u konstrukciji ispravnog evolucijskog okruženja je razumijevanje svih njegovih dijelova (populacija, dobrotu, selekcija, operatori) te njihovih međusobnih interakcija.

1.1. Populacija

Populacija je skup jedinki iste vrste smještenih na nekom području. Kako dio populacije stari i umire, tako se razmnožavanjem stvaraju novi potomci i veličina populacije u svakoj generaciji ostaje konstantna. U genetskom algoritmu populacija predstavlja skup jedinki od kojih je svaka jedinka potencijalno rješenje zadanog problema. Početna populacija može biti odabrana slučajnim odabirom ili nekim drugim optimizacijskim postupkom. Odumiranje slabijih jedinki koje se nisu uspjele prilagoditi novim životnim uvjetima i opstanak jedinki koje su to uspjele u genetskim se algoritmima provodi uporabom takozvane funkcije dobrote (detaljnije objašnjeno u nastavku). Razmnožavanje jedinki koje su se svojim dobrim svojstvima uspjele probiti provodi se operacijom križanja. Iz iteracije u iteraciju jedinke u populaciji poprimaju sve poželjnija svojstva. Algoritam obično završava dosezanjem zadanog broja iteracija. Kada je uvjet završetka ispunjen, iz dobivene populacije odabire se najbolja jedinka i ona predstavlja rješenje optimizacijskog problema.

1.2. Funkcija dobrote

Kao što je već spomenuto u prethodnom poglavlju, funkcija dobrote predstavlja prirodnu okolinu koja vrši selekciju nad jedinkama. Što je jedinka bolje prilagođena okolini u kojoj živi, tj. što joj je veća dobrotu, veće su joj i šanse za preživljavanje. Ovdje treba napomenuti da ključ problema nije odmah u prvoj iteraciji odabrati jedinke s najvećom dobrotom jer jedinke koje su po tom pitanju prosječne nose neka bitna svojstva koja nije poželjno izgubiti. Dakle, izbacuju se one jedinke koje imaju ekstremno nisku razinu dobrote unutar promatrane populacije.

Odabir adekvatne funkcije dobrote obično je ključan problem kod implementacije genetskog algoritma. Također, obzirom da se radi o funkciji koja se u algoritmu najviše koristi, funkcija dobrote trebala bi biti što je moguće jednostavnija i brža.

1.3 Genetski operatori

Kako bi se osigurao napredak populacije potrebno je osmisliti načine stvaranja novih jedinki. Cilj je osiguranje opstanka genetskog materijala boljih jedinki, tj. onih koje posjeduju veći koeficijent dobrote i na taj način promovirati što više željenih osobina. Promatrajući evoluciju živih bića nailazimo na dva načina stvaranja novog genetskog materijala: križanje i mutacija.

1.3.1 Križanje

Postupak križanja imitira spolnu reprodukciju dviju jedinki (roditelji), prilikom čega dolazi do zamjene jednog dijela roditeljskog gena onime drugoga. Novonastala jedinka (dijete) posjeduje genetski kôd obaju roditelja, te kombinaciju njihovih osobina. Križanjem dviju jedinki koje posjeduju pozitivne osobine očekuje se dijete koje će također posjedovati pozitivne osobine, tim više što je moguće nadomještanje negativnih osobina nekog od roditelja genima onog drugog čije su osobine bolje na nepovoljnom segmentu.

1.3.2 Mutacija

Mutacija je trajna promjena genetskog materijala, najčešće uzrokovana vanjskim čimbenicima. Smatra se jednim od predujeta evolucije jer se procesom prirodnog odabira u populaciji nakupljaju mutacije koje omogućuju bolju prilagodbu uvjetima

okoliša i time poboljšavaju vjerojatnost preživljavanja jedinke koja ih nosi te prenosi na sljedeće generacije.

Mutacija se odvija nad pojedinom jedinkom. Obično nakon postupka križanja svako dijete prolazi kroz proces mutacije s proizvoljnim parametrom (najčešće malim) vjerojatnosti. Ukoliko je jedinka odabrana za mutaciju bira se proizvoljan dio genetskog koda jedinke koji će biti zamijenjen nasumičnim nizom (ili drugom strukturom u ovisnosti o prikazu jedinke).

1.4 Selekcija

Selekcija je posljedica generacije velikog broja jedinki. Kako nije moguće očuvati sve jedinke u populaciji mora se odrediti način odabira jedinki koje će opstati, te onih koje ćemo ukloniti, tj. koje će uginuti. Iako je cilj da populacija sadrži samo jedinke pozitivnih svojstava, prerana konvergencija kao posljedica selekcije samo najboljih jedinki odražava se negativno na uspješnost algoritma. Postoji nekoliko različitih vrsta selekcija koje pokušavaju riješiti taj problem.

Neke od popularnijih tehnika selekcije su jednostavna, turnirska, eliminacijska i elitizam kao zasebno svojstvo koje se dodatno može primjeniti nad svakom selekcijom. Slijedi kratko objašnjenje svake od navedenih selekcija.

1.4.1 Jednostavna selekcija

Jednostavna selekcija generira populaciju istog broja jedinki korištenjem dobrote svake jedinke kao vjerojatnosti odabira. Što je veća dobrota jedinke to je veća vjerojatnost da će ta jedinka biti odabrana, ali ujedno se pruža mogućnost da se odaberu i jedinke manje dobrote.

Ukoliko neka jedinka posjeduje velik iznos dobrote proporcionalna je vjerojatnost njene prisutnosti veći broj puta u novoj populaciji. Pojavljivanje duplikata u populaciji pokazalo se kao nepoželjan fenomen koji usporava algoritam te su osmišljene tehnike koje ih nastoje ublažiti. Jedan od načina je da se jedinke poredaju uzlazno po vrijednosti dobrote, te se vjerojatnost odabira jedinke računa položajem u nizu i visini dobrote. Rezultat ove korekcije je ujednačavanje vjerojatnosti odabira jedinki, dok se u isto vrijeme zadržava veća vjerojatnost pojave jedinki s većom dobrotom.

1.4.2 Turnirska selekcija

Turnirska selekcija simulira turnire u kojima se pobjednici dodaju u skup jedinki za razmnožavanje. Broj sudionika turnira predstavlja selekcijski pritisak koji kontrolira brzinu konvergencije. Ukoliko je broj sudionika veći manja je vjerojatnost odabira

jedinki koje posjeduju manji koeficijent dobrote, te je brža konvergencija prema rješenju. Brza konvergencija dvosjekli je mač koji često dovodi do suboptimalnog rješenja, stoga je važno pažljivo odrediti selekcijski pritisak.

Eliminacijska turnirska selekcija nakon nasumičnog odabira proizvoljnog broja jedinki uklanja najlošiju jedinku nadomještajući je djetetom dviju slučajno odabranih preživjelih jedinki.

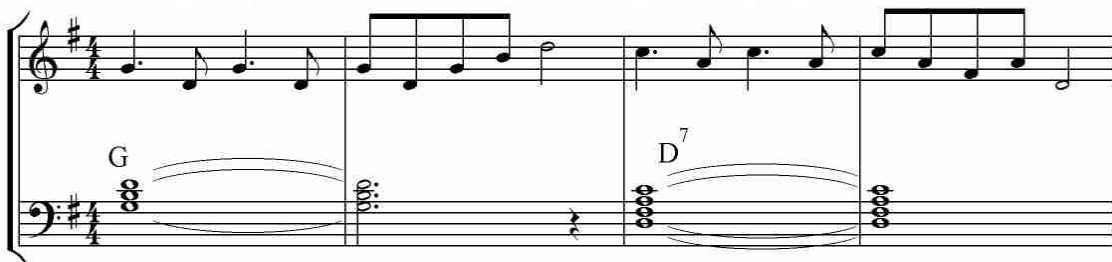
1.4.3 Eliminacijska selekcija

Eliminacijska selekcija temelji se na izbacivanju loših jedinki radije nego odabiru boljih pri generiranju nove populacije. Kako bi odabrala lošu jedinku izračunava se kazna svake jedinke koja predstavlja vjerojatnost odabira jedinke za izbacivanje. Kazna jedinke je razlika maksimalne dobrote i dobrote jedinke. Ovim načinom odabira jedinke za izbacivanje iz postupka selekcije u potpunosti se uklanja jedinka s najvećom dobrotom, pošto ona ima kaznu nula. Zbog očuvanja broja jedinki u populaciji nakon izbacivanja stvara se nova jedinka nekim od genetskih operatora te nastavlja proces selekcije.

1.4.4 Elitizam

Kako se dobra rješenja ne bi izgubila uporabom genetskih operatora ili eliminacijom tijekom selekcije javlja se potreba za zaštitom najbolje jedinke od izmjena ili eliminacije. Zaštita najbolje jedinke naziva se elitizam i osigurava da se svaka nova generacija kreće prema globalnom optimumu. Ukoliko je populacija velika, prilikom pretrage za najboljom jedinkom uvodi se nepotreban overhead koji može dodatno usporiti rad algoritma.

Osnovni glazbeni pojmovi



Slika 2. Izvadak iz Mozartove „Male noćne muzike“

Slijedi opis pojmova koji su nužni za shvaćanje kako bi se mogla shvatiti i sama ideja završnog rada. Pojmovi su većinom jako pojednostavljeni i objašnjeni u vrlo kratkim crtama, te popraćeni primjerima kako bi se čitatelju što više olakšalo njihovo razumijevanje.

1.1 Ton

«Ton je slušni doživljaj pravilnog titranja tijela. Visinu i boju možemo mu pouzdano odrediti uhom. Zahvaljujući tome, ton je daleko čišći element u glazbenim djelima negoli šum, jer omogućuje melodiju i harmoniju.»¹

1.2 Melodija

«Melodija, u najširem i najjednostavnijem smislu, niz je tonova različite visine, što svoja obilježja stječe prema načinu organizacije.»¹

¹ Mutak, V., Petrović, T. Kompendij glazbenih pojmova. Zagreb: Vlastita naklada autora, 1991.

1.3 Nota

«Tonove zapisujemo notama koje upisujemo u crtovlje.»¹ Nota je osnovna građevna jedinica svake melodije. U glazbi postoji sedam osnovnih nota: «c», «d», «e», «f», «g», «a» i «h». Nakon note «h» ponovno slijedi nota «c» koja se od prethodne razlikuje u visini. Ona započinje novi niz od sedam tonova. Da bi se nizovi razlikovali dodaju im se brojevi indeksi. Tako postoje note «c1», «c2», «c3» itd. Tonski sustav dijeli se u grupe od po sedam navedenih tonova. Ta grupa kao melodijska cjelina uključuje i osmi ton i naziva se oktava tonskog sustava. Oktave se najlakše mogu uočiti pri pogledu na klavirske tipke koje su prikazane na Slici 3.

Svaka nota može biti povišena ili snižena, pri čemu se noti dodaje sufiks -is odnosno -es. Npr. povišenjem note «c» dobiva se «cis», a njenim sniženjem dobiva se nota «ces». Povišenjem se stupanj note povećava za pola stupnja, a sniženjem se umanjuje za pola. Pri povišenju note stavlja se znak povišilice, a pri sniženju se stavlja znak snizilice. Ako se nota želi vratiti u prvobitni oblik, tj. želi joj se maknuti sufiks stavlja se znak razrješilice.

Osim po boji tona i visini, note se razlikuju i po trajanju. «Razliku u trajanju označavamo različitim izgledom nota ili stanka (pauza), pa razgovor o trajanju može postati razgovor o vrijednostima kojima se određeno trajanje prikazuje. Suvremeni sustav trajanja, to jest sustav notnih vrijednosti, započinje cijelom notom ili stankom. Svaka je sljedeća vrijednost note ili stanke u sustavu upola kraća od prethodne. Matematički rečeno, sustav notnih vrijednosti nastaje podjelom cijele note na broj dijelova koji su članovi geometrijske progresije broja dva s kvocijentom dva (2, 4, 8, 16, 32, 64 itd.). Notne vrijednosti, isto kao i stanke kojima na određeno vrijeme prekidamo tijekom zvukova, dobivaju ime u odnosu prema cijeloj noti (stanki) čija su polovina, četvrtina, osmina, šesnaestina, tridesetdrugina, šezdesetčetvrtina itd., pa ih stoga nazivamo polovinkom, četvrtinkom, osminkom, šesnaestinkom, tridesetdruginkom, šezdesetčetvrtinkom itd.»²

Trajanje cijele note iznosa je četiri dobe, dok se trajanje svake iduće note dijeli sa dva. Tako je trajanje polovinke iznosa dvije dobe, trajanje četvrtinke iznosa jedne dobe itd.

1.4 Stupanj i polustupanj, odnosno stepen i polustepen

Svaku notu jedinstveno identificira njezina frekvencija. «Da bismo slušajući dva tona zamijetili razliku u visini, omjer njihovih frekvencija ne smije biti manji od 1,003. Omjer frekvencija susjednih nota u tonskom sustavu europske umjetničke glazbe znatno je veći i iznosi 1,05946. Takav razmak, kao najmanji praktički upotrebljiv razmak između susjednih tonova tonskog sustava europske umjetničke glazbe, nazivamo

² Petrović, T. Nauk o glazbi. Zagreb: Tihomir Petrović, 1998.

polustepen. Iz naziva se jasno vidi da se radi o polovici većeg razmaka koji nazivamo stepen.»³

1.5 Intervali

Ne treba imati veliku glazbenu naobrazbu da bi se moglo konstatirati da neke melodije imaju lijep, uhu ugodan zvuk, dok neke zvuče poput stare, parne lokomotive. Kako to postići? Kako osigurati da melodija koja se sklada bude zvučna? Sada u priču ulaze intervali. «Interval je razmak između dvaju tonova različite visine.»⁴ Konsonantni intervali svojim nastupom stvaraju ugođaj mirnoće i stabilnosti. Oni među njima koji su stabilniji pripadaju skupini savršenih konsonanata, a ostali pripadaju skupini nesavršenih konsonanata. Disonantni intervali stvaraju ugođaj nesklada i nemira, odnosno neke napetosti koju treba riješiti. Koji intervali su konsonantni a koji disonantni? I koliko uopće intervala postoji?

U glazbi postoji osam intervala: prima, sekunda, terca, kvarta, kvinta, seksta, septima i oktava. Njihove veličine predstavljene su u Tablici 1. Prima je interval koji postoji samo u teoriji, budući se radi o razmaku između dvije iste note. «Intervale koji se približno podjednako učestalo javljaju u dvije veličine, različite za pola stepena razvrstavamo na velike i male. To su sekunde, terce, sekste i septime. Intervale koji se javljaju u jednoj veličini, uz eventualnu iznimku, zovemo čisti. To su kvarta, kvinta i oktava. Kvartu od tri stepena zovemo povećanom, a kvintu od tri stepena smanjenom.»⁵

Tablica 1. Intervali u glazbi

<i>Interval</i>	<i>Veličina intervala</i>
sekunda	½ ili 1 stepen
terca	1½ ili 2 stepena
kvarta	2½ , s jednom iznimkom od 3 stepena
kvinta	3½ , s jednom iznimkom od 3 stepena
seksta	4 ili 4½ stepena
septima	5 ili 5½ stepena
oktava	6 stepena

³ Mutak, V., Petrović, T. Kompendij glazbenih pojmova. Zagreb: Vlastita naklada autora, 1991.

⁴ Petrović, T. Nauk o glazbi. Zagreb: Tihomir Petrović, 1998.

⁵ Mutak, V., Petrović, T. Kompendij glazbenih pojmova. Zagreb: Vlastita naklada autora, 1991.

1.5 Ljestvica

«Ljestvica je skup različitih tonova određene međusobne hijerarhije. Tonovi u ljestvici nanizani su uzlazno ili silazno bez ponavljanja već uvrštenih tonova nakon što se ispuni čista oktava.»⁶ Dva najčešće korištena oblika ljestvice su durska i molska ljestvica.

Primjer uzlazne dur ljestvice koja započinje tonom «c»:



Primjer silazne dur ljestvice koja završava tim istim tonom «c», a započinje tonom «c» koji je od prethodnog viši za oktavu:



1.2 Tonalitet

«Uzročnopoljedično povezivanje tonova odnosno akorda postupno postaje sustav, način uspostave sklada među tonovima, što nazivamo tonalitet. Tonalitet je dakle sustav odnosa akorda i/ili tonova u skladbi. Temelji se na njihovoj uzročnopoljedičnoj ovisnosti i međusobnoj srodnosti. Slušno središte tonaliteta (ton kojem sve vodi, od kojega se sve mjeri) jest tonika – ton kojim ujedno počinje ljestvica od čijih su tonova izgrađeni melodija i akordi skladbe. Po tom tonu dobivaju ime i ljestvica koju započinje i tonalitet kojem je taj ton središte. Ako je riječ, primjerice, o tonu C, valjalo bi za skladbu kazati da je «tonalitetna, u durskoj/molskoj ljestvici s početnim tonom c. Ovaj se izraz u praksi skraćuje u «skladba je u C-dur/mol-tonalitetu» ili «skladba je u C-duru/molu.»⁷

Dur ljestvice moraju imati polustupnjeve između 3. i 4. i 7. i 8. tona u ljestvici i da bi se to osiguralo uvode se povisilice, odnosno snizilice. Konkretno, G-dur ima povisilicu «fis» i ona se piše na početku crtovlja, desno od violinskog ključa.

U završnom radu melodije se skladaju u G-duru i koristi se sljedećih dvanaest nota: «c1», «d1», «e1», «fis1», «g1», «a1», «h1», «c2», «d2», «e2», «fis2» i «g2». Note se prikazane na slici ispod.



⁶ Petrović, T. Nauk o glazbi. Zagreb: Tihomir Petrović, 1998.

⁷ Petrović, T. Nauk o glazbi. Zagreb: Tihomir Petrović, 1998.

1.4 Takt

Svaka skladba sastoji se od taktova. Na Slici 2. takt predstavlja okomitu crtu koja dijeli melodiju na dijelove. Ta crta nije tu postavljena sasvim slučajno. U svakom taktu može se nalaziti točno određen broj doba (notnih trajanja), a taj broj zadaje se mjerom.

1.5 Mjera

Oznaka mjere nalazi se na samom početku skladbe u obliku dvije brojke postavljene jedna ispod druge. Mjera govori koliko doba se smije nalaziti u jednom taktu. U skladbama ovog završnog rada koristi se četvero-četvrtinska mjera, što znači da se u svakom taktu mogu nalaziti četiri dobe, a da svaka doba vrijedi jednu jedinicu trajanja. To znači da se u taktu može nalaziti najmanje jedna nota u trajanju od četiri dobe (cijela nota), a najviše šesnaest nota u trajanju od $\frac{1}{4}$ dobe (šesnaestinka).

Vrednovanje pojedinih harmonijskih nizova i suzvuka

U okviru završnog rada osmišljen je algoritam za vrednovanje nota koji će poslužiti za računanje već spomenutih intervala između nota. Vrednovanjem se note ne kategoriziraju u „dobre“ i „loše“, već se svakoj noti dodjeljuje njena jedinstvena vrijednost (jedinstveni broj za raspoznavanje). Kako vrednovati pojedinu notu u melodijskom nizu? Algoritam je najbolje najprije objasniti na poznatom primjeru iz svakodnevnog života. Kako vrednovati slovo „c“ u abecedi? Najlogičnije bi bilo vrednovati svako slovo u abecedi u odnosu na početno slovo, slovo „a“. Tako slovo „c“ poprima vrijednost 3, slovo „e“ vrijednost 9, a slovo „ž“ vrijednost 30.

Analogno ovom algoritmu, svakoj se noti dodjeljuje vrijednost koja se računa kao broj polustupnjeva za koje je ona udaljena od note „c1“.



Slika 3. Klavijature

Na slici su prikazane klavijature sa dvije oktave. Prva nota «c» neka je, radi lakšeg praćenja, označena kao «c1», a druga kao «c2». Polustupanj je udaljenost bijele tipke do najbliže crne tipke. Slučaj koji je vidljiv kod nota «e» i «f», gdje nema crnih tipki naziva se prirodni polustupanj i njegova vrijednost jednaka je ranije navedenom polustupnju. Iz slike je vidljivo da je ton „e1“ udaljen od tona „c1“ za 4 polustupnja, pa je njegova vrijednost 4. Ton „c2“ udaljen je za od tona „c1“ za 12 polustupnjeva pa je njegova vrijednost 12, itd.

Interval se u ovom algoritmu računa kao apsolutna razlika vrijednosti dviju susjednih nota i predstavlja udaljenost između te dvije note. Odnos između algoritamski izračunatih udaljenosti i intervala priložen je u Tablici 2.

Tablica 2. Algoritamski izračun intervala

<i>Interval</i>	<i>Algoritamska udaljenost</i>
prima	0
mala sekunda	1
velika sekunda	2
mala terca	3
velika terca	4
povećana kvarta	5b
čista kvarta	5a
smanjena kvinta	6
čista kvinta	7
mala seksta	8
velika seksta	9
mala septima	10
velika septima	11
oktava	12

Elementi genetskog algoritma primijenjeni u završnom radu

Već je rečeno da se skladanje melodija u mnogočemu razlikuje od većine ostalih problema koje genetski algoritam rješava. Prije izrade završnog rada autor je isti problem razvijao primjenom genetskog algoritma kakav je objašnjen ranije u tekstu. To je klasičan oblik genetskog algoritma koji se koristi za probleme optimizacije. Dakle u priču su, uz uvijek neophodnu populaciju i jedinku bili uključeni operatori križanja i mutacije i koristila se jednostavna selekcija. Rezultati nisu bili zadovoljavajući. Zaključak svega bio je da funkcija dobrote i genetski operatori ne mogu biti proizvoljno odabrani jer se očito radi o vrlo osjetljivom području koje zahtjeva visoku preciznost. Uzevši u obzir količinu vremena koja je stavljena na raspolaganje za izradu završnog rada, jedina opcija bila je pronaći ih gotove u obliku koji je testiran i daje uspješne rezultate. Posvetivši se detaljnijem istraživanju tog područja, između mnogo zanimljive literature odabran je rad Ronalda Hochreitera pod nazivom «Audible Convergence for Optimal Base Melody Extension with Statistical Genre-Specific Interval Distance Evaluation». Od autora je preuzeta ideja za funkciju dobrote i genetske operatore. Stoga se u završnom radu rabi pomalo specifičan genetski algoritam, tj. genetski algoritam se morao prilagoditi problemu koji rješava. U nastavku slijede detaljni opisi primjenjenih elemenata genetskog algoritma i neke modifikacije koje je nad njima bilo nužno provesti.

1. Jedinka

Jedinka predstavlja jednu skladbu u G-duru. Sastoji se od četiri takta, a svaki takt sastoji se od nota. Već je rečeno da su skladbe pisane u četvero-četvrtinskoj mjeri, pa treba voditi računa o pravilima popunjavanja taktova. Uvedeno je jedno malo ograničenje. Početna nota, tj. nota u prvom taktu ne smije biti cijela nota jer se tada za taj takt ne može izračunati ocjena intervala (budući u tom taktu tada ne postoje barem dvije note koje bi zatvarale neki interval). Također, treba paziti da se note nasumce odabiru ovisno o prethodnoj popunjenosti takta. Tako se na početku takta, dok je on još prazan, mogu birati sve note za popunjavanje (kod prvog takta mogu sve note osim cijele). Kada je takt popunjen s barem jednom notom više se ne može birati bilo koja nota. Naprimjer, ako je za prvu notu odabrana osminka (čija je vrijednost $\frac{1}{2}$ dobe) tada se u taj isti takt više ne može staviti cijela nota (čija je vrijednost 4 dobe) jer u protivnom ukupna vrijednost unutar takta iznosi $4\frac{1}{2}$ dobe, a dozvoljen iznos je 4 dobe. U Tablici 3. je dan prikaz nota koje se u završnom radu koriste i njihovih pripadajućih vrijednosti, a u Tablici 4. je dan prikaz mogućih notnih trajanja:

Tablica 3. Korištene note i njihove vrijednosti

<i>Nota</i>	<i>Vrijednost</i>
c1	0
d1	2
e1	4
fis1	6
g1	7
a1	9
h1	11
c2	12
d2	14
e2	16
fis2	18
g2	19

Tablica 4. Korištena notna trajanja

<i>Naziv</i>	<i>Trajanje (izraženo u dobama)</i>
cijela nota	4
polovinka	2
čtvrtinka	1
osminka	0.5
šesnaestinka	0.25

2. Populacija

Početna populacija generirana je slučajnim odabirom. Sastoji se od 30 jedinki. Nakon stvaranja populacije ona se sortira po rastućim vrijednostima funkcije dobrote. Funkcija dobrote ovdje je zapravo funkcija kazne tako da manja vrijednost označava bolju jedinku. Budući je početna populacija generirana sasvim slučajno, bez ikakvih optimizacijskih metoda, za očekivati je da će dobrota tih jedinki biti loša.

3. Referentna jedinka



Slika 4. Wolfgang Amadeus Mozart

Da bi se mogla izračunati dobrota jedinke potrebno je naći neku melodiju koja skladno zvuči i koja će se koristiti za usporedbu sa svakom od jedinki iz populacije. Bitno je spomenuti da jedinka teži referentnoj jedinki samo na temelju skladnosti intervala. Osim toga, ono što također čini veliki segment svake poznate skladbe jest njen ritam. To se unutar ovog završnog rada nije razmatralo.

Program mijenja jedinke primjenjujući nad njima navedene genetske operatore. Nakon toga jedinka se evaluira i dobivaju se izračunati parametri. Ti parametri uspoređuju se s parametrima referentne jedinke. Što su im parametri sličniji, jedinka ima veće šanse za opstanak. Za referentnu jedinku u ovom je radu odabrana upravo Mozartova «Mala noćna muzika», tj. prva četiri takta te melodije. Zašto je od svih skladbi u G-duru odabrana baš ta?

Kada se izračunaju intervali između nota Mozartova skladba daje intervale koji su uvjerljivo najbolji. Što znači najbolji? Najviše savršenih konsonanata. Najskladniji raspored Mozartovih intervala u nastavku će biti i dokazan na samom primjeru.

4. Funkcija dobrote

4.1 Evaluacija pojedine jedinke

Svaka generirana jedinka evaluira se u sljedeća četiri koraka:

1. korak evaluacije: Postavljanje vrijednosti

Iznad svake note u crtovlju zapiše se njezina vrijednost prema Tablici 3.

2. korak evaluacije: Izračun intervala

Između svake dvije susjedne note u taktu računaju se intervali. Na prijelazu taktova također se računaju intervali između zadnje note u n-tom taktu i prve note u n+1 taktu. Valja primijetiti da će broj izračunatih intervala u svakom taktu osim u prvom biti jednak broju nota u tom taktu. U prvom taktu će broj intervala biti za jedan manji budući prva nota u melodiji nema svoju prethodnicu s kojom bi zatvarala neki interval.

3. korak evaluacije: Ocjenjivanje intervala

U ovom se koraku dodjeljuju ocjene intervalima izračunatim u 2. koraku evaluacije. Ocjene se dodjeljuju prema sljedećoj tablici:

Tablica 5. Ocjene intervala

<i>Intervalne kategorije</i>	<i>Intervali</i>	<i>Ocjena</i>
Savršeni konsonanti	0, 5, 7, 12	1
Nesavršeni konsonanti	3, 4, 8, 9	2
Disonanti	1, 2, 6, 10, 11	3

Ocjene dodjeljene intervalima zapravo su kazne (*engl. penalty*). Tako najbolji intervali, tj. savršeni konsonanti dobivaju najmanju kaznu dok oni najlošiji, tj. disonanti dobivaju najveću kaznu.

Povećana kvarta je interval između nota „c“ i „fis“. Smanjena kvinta je interval između nota „c“ i „ges“. Budući među notama koje se koriste nema note „ges“, tako se smanjena kvinta neće nikada niti pojaviti. Stoga se u završnom radu ne koriste vrijednosti 5a za čistu kvartu i 5b za povećanu kvartu. Zbog izostanka smanjene kvinte uvedeno je pojednostavljenje. Vrijednost 5 predstavlja čistu kvartu, a vrijednost 6, koja u Tablici 2. predstavlja smanjenu kvintu, predstavlja povećanu kvartu. Postupak je opravdan budući su veličine intervala povećane kvarte i smanjene kvinte jednake i obje iznose 3 stepena.

4. korak evaluacije: Izračun aritmetičke sredine i varijance

Računa se aritmetička sredina i standardno odstupanje (varijanca) ocjena iz 3. koraka. Aritmetička sredina i varijanca računaju se posebno za svaki takt.

Formula za aritmetičku sredinu:

$$a = \frac{1}{n} \sum_{i=1}^n x_i$$

Ovo je općepoznata formula. „n“ predstavlja broj ocjena u taktu dok „x_i“ predstavlja pojedinu ocjenu koja se sumira.

Formula za varijancu:

$$\sigma^2 = \frac{1}{n} \sum_{i=0}^n (x_i - a)^2$$

4.2 Evaluacija referentne jedinke

Kao što je već rečeno, referentna jedinka je Mozartova «Mala noćna muzika» i njezinim ocjenama teži svaka generirana jedinka. Slijedi evaluacija prva četiri takta «Male noćne muzike». Time će se dobiti aritmetička sredina i varijanca za usporedbu s ostalim jedinkama i za izračun funkcije dobrote.

Prije samog izračuna treba objasniti uvođenje jedne male modifikacije u Mozartovo djelo i razlog njezinog uvođenja.

Prva četiri takta «Male noćne muzike»:



Slika 5. „Originalni“ Mozart

Prva četiri takta «Male noćne muzike» mogu se podijeliti u dvije glazbene fraze. Svaka od njih traje dva takta. «Glazbena fraza je zapravo manja glazbena cjelina, melodijskog karaktera»⁸. Laički govoreno, to je kao jedna kratka, dvotaktna melodija. Posljednja nota prve fraze je nota «d2», dok je prva nota druge fraze nota «c2» Interval koji spaja dvije fraze upravo je interval između navedene dvije note. Radi se o velikoj

⁸ Petrović, T. Glazbeni oblici. Zagreb: Stuba, 1992.

sekundi. Njena algoritamska vrijednost je 2, a pogledom u Tablicu 5. vidljivo je da taj interval ima najveću kaznu, tj. kaznu iznosa 3. Upravo ovo kvarilo je željeni iznos aritmetičke sredine i varijance za treći takt. Kako to izbjeći?

Ronald Hochreiter predlaže dva rješenja:

1. rješenje: Uporaba elemenata umjetne inteligencije
2. rješenje: Računanje intervala samo unutar, a ne i na prijelazima taktova

Niti jedno od ovih rješenja nije uzeto kao opcija.

Primjena drugog rješenja donijela bi u algoritam više štete nego koristi. Zašto? Zato jer je ovaj slučaj «Male noćne muzike» zapravo iznimka, a ne generalno pravilo. Da se uzela neka druga skladba, velika je vjerojatnost da se navedeni problem uopće ne bi niti pojavio. S druge strane, ako se primjeni drugo rješenje algoritam može izgenerirati vrlo disharmonične jedinke koje na svakom prijelazu takta imaju neki disonantni interval.

Primjena prvog rješenja zahtjeva možda i više posla nego sam završni rad, a možda i više znanja nego što ga osoba nakon tri godine studija ima. Izmisлити neki oblik umjetne inteligencije za raspoznavanje najrazličitijih mogućih fraza u glazbi? Veoma komplicirano.

Stoga je autor primjenio svoju osobnu verziju rješenja: kao referentna jedinka zapravo se neće koristiti «originalni Mozart» već «modificirani Mozart» koji kao prvu notu u trećem taktu nema notu «c2» već notu «d2». A to je ista nota kao i posljednja nota u drugom taktu. Također treba naglasiti da su kod izračuna izostavljene pauze iz Mozartove melodije budući one ionako nemaju nikakav doprinos u računanju parametara.

Prva četiri takta uz primijenjenu modifikaciju:



Slika 6. „Modificirani“ Mozart

Modifikacija je uistinu malena, te na prvi pogled većini možda nije ni vidljiva. Melodijska razlika također nije toliko velika. Opravdanje za ovaj postupak leži u tome što je izbor referentne jedinke sasvim proizvoljan. U programu se korisniku nudi opcija postavljanja referentne melodije po želji. Jedini uvjet koji ona mora zadovoljavati jest da je napisana u G-duru. Dakle, ako je unutar ovog završnog rada za referentnu jedinku odabran upravo «modificirani» Mozart, to je odluka autora. Netko drugi može postaviti drugačiju referentnu jedinku i prema njoj mijenjati jedinke u svojoj populaciji.

Gore navedena evaluacija jedinke biti će još jasnija kada se kao primjer izvede evaluacija prva četiri takta «Male noćne muzike».

1. korak: Postavljanje vrijednosti

Musical notation showing fingerings for a sequence of notes: 7, 2, 7, 2, 7, 2, 7, 11, 14, 14, 9, 12, 9, 12, 9, 6, 9, 2.

2. korak: Izračunavanje intervala

Intervali su označeni zelenom bojom. Sada valja uočiti da se interval između zadnje note u taktu i prve u idućem taktu prenosi na idući takt.

Musical notation showing intervals (green numbers) for a sequence of notes: 5 5, 5, 5 5 5 4 3, 0, 5 3, 3, 3 3 3 3 7.

3. korak: Ocjenjivanje intervala

Ocjene intervala (tj. dodijeljene im kazne) na slici su označene crvenom bojom.

Musical notation showing interval ratings (red numbers) for a sequence of notes: 1 1, 1, 1 1 1 2 2, 1, 1 2, 2, 2 2 2 2 1.

4. korak: Izračun aritmetičke sredine i varijance

Tablica 6. Parametri referentne jedinice

Redni broj takta	Aritmetička sredina	Varijanca
1. takt	1.0	0.00
2. takt	1.4	0.24
3. takt	1.5	0.25
4. takt	1.8	0.16

4.3 Funkcija dobrote (engl. fitness function)

Funkcija dobrote računa se kao euklidska udaljenost pojedine jedinke od referentne (Mozartove) jedinke. Ispravniji naziv zapravo bi bio «funkcija kazne», ali u genetskim algoritmima uvriježeno je koristiti naziv «funkcija dobrote».

Potpuna formula:

$$f = \sum_{i=1}^b \lambda_m \|\mu_i, \text{AritmetičkaSredina}(b_i)\| + \lambda_v \|\sigma_i^2, \text{Varijanca}(b_i)\|$$

Suma ide od 1 do b , gdje je „ b “ broj taktova u skladbi. Oznake μ_i i σ_i^2 predstavljaju aritmetičku sredinu i varijancu «Male noćne muzike» po taktovima. AritmetičkaSredina(b_i) i Varijanca(b_i) predstavljaju aritmetičku sredinu i varijancu generirane jedinke po taktovima.

Dakle, za svaki takt generirane jedinke računa se udaljenost između aritmetičke sredine te jedinke i Mozartove jedinke, te udaljenost između varijance te jedinke i Mozartove jedinke. Budući da se radi u jednodimenzionalnom prostoru, euklidska udaljenost je zapravo apsolutna razlika ta dva broja. Broj taktova jedinki je konstantan i uvijek iznosi 4.

Parametri λ_m i λ_v služe za podešavanje važnosti aritmetičke sredine ili varijance (u slučaju da je za funkciju dobrote aritmetička sredina tri puta značajnija od varijance postavlja se da je $\lambda_m = 3$, a $\lambda_v = 1$). U programu su obje vrijednosti λ_m i λ_v postavljene na 1.

5. Genetski operatori

Najveća razlika između genetskog algoritma primjenjenog u završnom radu i klasičnog genetskog algoritma vidljiva je upravo kod genetskih operatora. Naime, operator križanja uopće se ne koristi. Umjesto toga, programu su na raspolaganju četiri operatora mutacije. Kod sva četiri operatora mijenja se samo notna vrijednost, a ne i trajanje. U protivnom bi došlo do pogreške u mjeri.

1. operator mutacije: Zamjena dviju susjednih nota

Metoda slučajno odabire poziciju p unutar jedinke. Zatim note na pozicijama p i $p+1$ zamjene mjesta. Pozicija p ne smije biti posljednja nota u jedinki jer ona nema s kime zamijeniti mjesto.

2. operator mutacije: Promjena stupnja note za slučajno odabrani interval

Stupanj note zapravo je isto što i njezina vrijednost. Drugim riječima, ovom mutacijom mijenja se vrijednost note za neki slučajno odabrani broj u rasponu od 0 (interval prime) do 12 (interval sekunde). Taj se broj može ili dodati ili oduzeti od postojeće notne vrijednosti. Ovaj operator mutacije je najkompliciraniji od sva četiri operatora, a evo i zašto. Gledajući vrijednosti nota u Tablici 3. na prvi je pogled vidljivo da je najmanja vrijednost note 0 (nota «c1»), dok je najveća vrijednost note 19 (nota «g2»). Vrijednosti ostalih nota nalaze se negdje između navedene dvije vrijednosti. Međutim, lako se uočava da neke vrijednosti nedostaju. To su naprimjer vrijednosti 1, 3, 5 itd. Dakle, notnoj vrijednosti ne može se dodati ili oduzeti bilo koji broj od 0 do 12 jer tada postoji velika vjerojatnost da se dobije nota koja nije u G-duru i nije jedna od nota koje se u ovom algoritmu koriste za generiranje jedinke. Zato je ovaj genetski operator malo drugačije osmišljen.

Najprije se slučajnim odabirom uzme jedna nota unutar jedinke. Zatim se provjerava odabrana nota i sve vrijednosti za koje se njezin stupanj može promijeniti, a da je dobivena nota i dalje u okviru G-dura. Vrijednosti dolaze s predznakom „+“ ili „-“ ovisno o tome dodaje li se ona ili oduzima od trenutnog stupnja note. Ako postoji više takvih vrijednosti, prema jednakoj vjerojatnosti odabira uzima se jedna od njih.

Proračun vrijednosti za svaku notu priložen je u Tablici 7.

Tablica 7. Dopuštene vrijednosti za promjenu stupnja svake od korištenih nota

<i>Nota</i>	<i>Vrijednost note</i>	<i>Mogući intervali za dodavanje/oduzimanje</i>
c1	0	+2, +4, +6, +7, +9, +11, +12
d1	2	+2, +4, +5, +7, +9, +10, +12, -2
e1	4	+2, +3, +5, +7, +8, +10, +12, -2, -4
fis1	6	+1, +3, +5, +6, +8, +10, +12, -2, -4, -6
g1	7	+2, +4, +5, +7, +9, +11, +12, -1, -3, -5, -7
a1	9	+2, +3, +5, +7, +9, +10, -2, -3, -5, -7, -9
h1	11	+1, +3, +5, +7, +8, -2, -4, -5, -7, -9, -11
c2	12	+2, +4, +6, +7, -1, -3, -5, -6, -8, -10, -12
d2	14	+2, +4, +5, -2, -3, -5, -7, -8, -10, -12
e2	16	+2, +3, -2, -4, -5, -7, -9, -10, -12
fis2	18	+1, -2, -4, -6, -7, -9, -11, -12
g2	19	-1, -3, -5, -7, -8, -10, -12

Jednostavnom provjerom može se pokazati da se uz pomoć tablice dobivaju adekvatne note.

3. operator mutacije: Promjena stupnja note za oktavu

Kada se stupanj note promijeni za oktavu nota mijenja broj u svome imenu. Primjerice, poveća li se nota „c1“ za oktavu ona prelazi u notu „c2“. Također, smanji li se nota „c2“ za oktavu ona prelazi u notu „c1“. Pogledom na Tablicu 3. opet je vidljivo da se određene note ne mogu povećati ili smanjiti za oktavu. Konkretno, radi se o notama «a1» i «h1».

Note koje se povećavaju za oktavu, tj. dodaje im se vrijednost 12: c1, d1, e1, fis1 i g1.

Note koje se umanjuju za oktavu, tj. oduzima im se vrijednost 12: c2, d2, e2, fis2 i g2.

4. operator mutacije: Zamjena skupine nota između slučajno odabrane početne i završne točke

Unutar jedinke odabiru se dvije note koje predstavljaju početnu i konačnu točku nekog podintervala. Zatim se note unutar tog podintervala zrcalno preslikavaju. Tako

zadnja nota podintervala postaje prva nota, a prva nota postaje zadnja. Predzadnja nota postaje druga nota, a druga nota postaje predzadnja itd. Valja napomenuti da se kod ovog operatora mutacije također ne mijenja trajanje note već samo njezino ime kako se ne bi poremetila mjera. Slika 7. pruža još bolje objašnjenje navedenog operatora:



Slika 7. Četvrti operator mutacije

Note zaokružene istom bojom su one koje se zamjenjuju. Na slici je dan primjer neparnog broja nota unutar podintervala. U tom slučaju note se zrcalno preslikavaju u odnosu na središnju notu koja ostaje nepromijenjena. Ako se unutar podintervala nalazi paran broj nota tada se svaka nota mijenja na već objašnjen način.

6. Selekcija

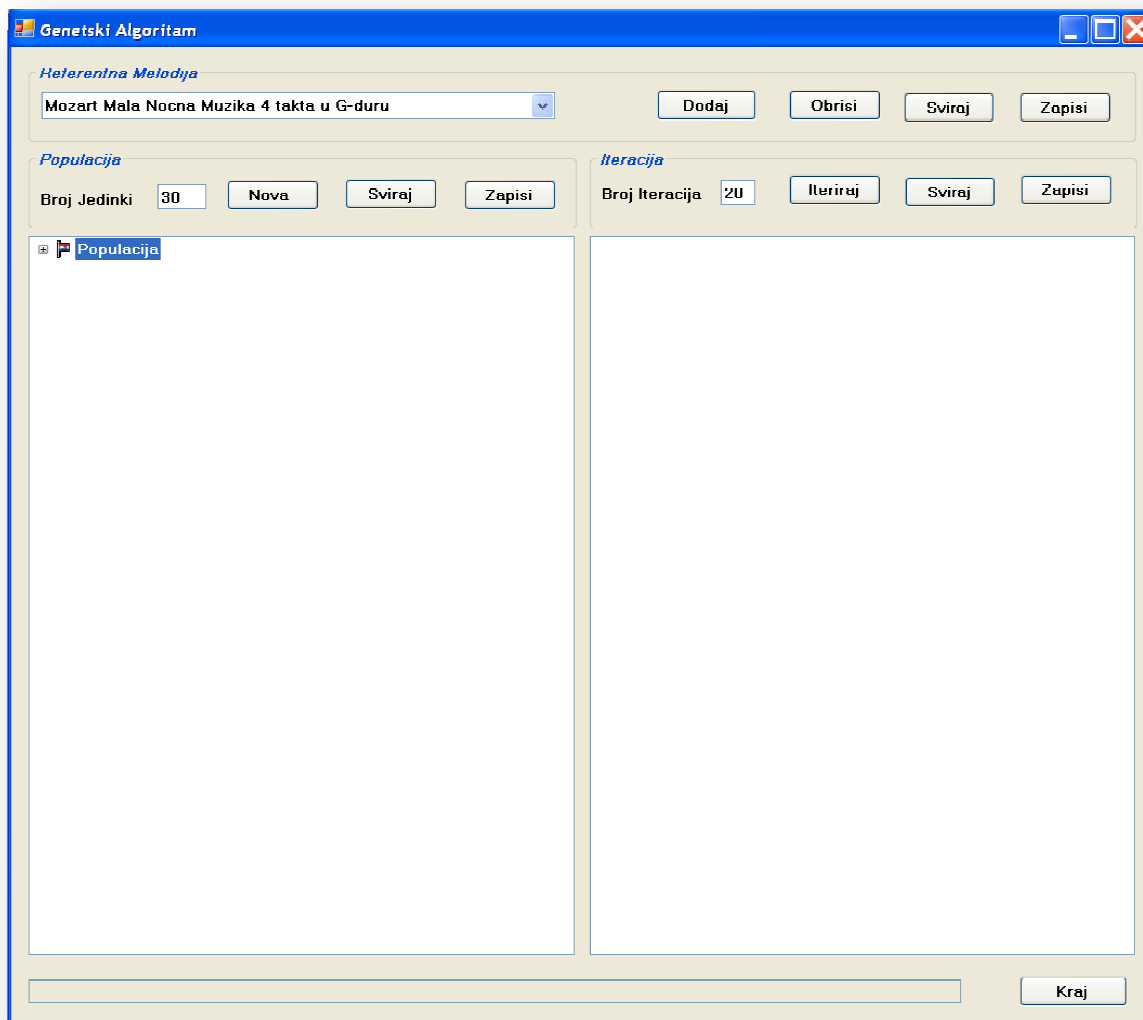
Ne koristi se niti jedna od spomenutih selekcija što je još jedna od posebnosti ovog genetskog algoritma. Već je spomenuto da je nakon generiranja početna populacija sortirana po rastućim vrijednostima funkcije dobrote. Jedinka je tim bolja što joj je dobrotu manja budući da se zapravo radi o funkciji kazne. Pri kreiranju iduće populacije najprije se u nju kopira deset najboljih jedinki iz prethodne populacije. Idućih deset jedinki dobiva se tako da se iz prethodne populacije uzme najbolja jedinka i nad njom se obavi pet mutacija. To se ponavlja deset puta kako bi se dobilo deset jedinki. Posljednjih deset jedinki dobiva se uzimanjem druge najbolje jedinice iz prethodne populacije i primjenjujući nad njom deset mutacija. Opisani postupak također se ponavlja deset puta. Time je populacija popunjena i završena je prva iteracija. Broj iteracija postavljan je na dvadeset. Kod testiranja programa vidi se da se funkcija kazne kroz iteracije smanjuje, što znači da se intervali popravljaju. Veličina populacije, broj iteracija, broj najboljih jedinki iz prethodne populacije koji se kopira u iduću, broj mutacija najbolje i druge najbolje jedinice, te broj mutiranih jedinki može se u programu mijenjati po želji. Međutim, veličina populacije ne smije biti manja od 30. Također, kod raspodjele jedinki za iduću populaciju trećina najboljih iz prethodne populacije odmah se kopira, trećina se dobiva mutacijama najbolje jedinice iz prethodne populacije i posljednja trećina dobiva se mutacijama druge najbolje jedinice iz prethodne populacije. U autorovom programu je veličina populacije 30 jedinki, pa su trećine skupine od 10 jedinki. Ako je veličina populacije broj koji nije djeljiv s tri, za trećine se uzimaju integer vrijednosti broja podijeljenog s tri.

Programsko rješenje

Program je razvijen i testiran u Microsoft Visual Studio 2008 okruženju uporabom programskog jezika C#. Korištena je objektno orijentirana paradigma. Program je izveden u dvije verzije: jedna konzolna aplikacija i jedna Windows Forms aplikacija koja je razvijena prije svega radi jednostavnijeg testiranja, a bazirana je na istim klasama kao i konzolna. Na CD-u je priložena verzija sa Windows Forms aplikacijom koja je ujedno i mnogo jednostavnija za korisnički rad. U nastavku najprije slijedi detaljno objašnjenje korištenja aplikacije kako bi korisnik mogao po želji postavljati vlastite parametre i koristiti program. Nakon toga slijedi opis grafičkog prikaza jedinke pomoću programa LilyPond i audio zapis jedinke pomoću autorove implementirane funkcije.

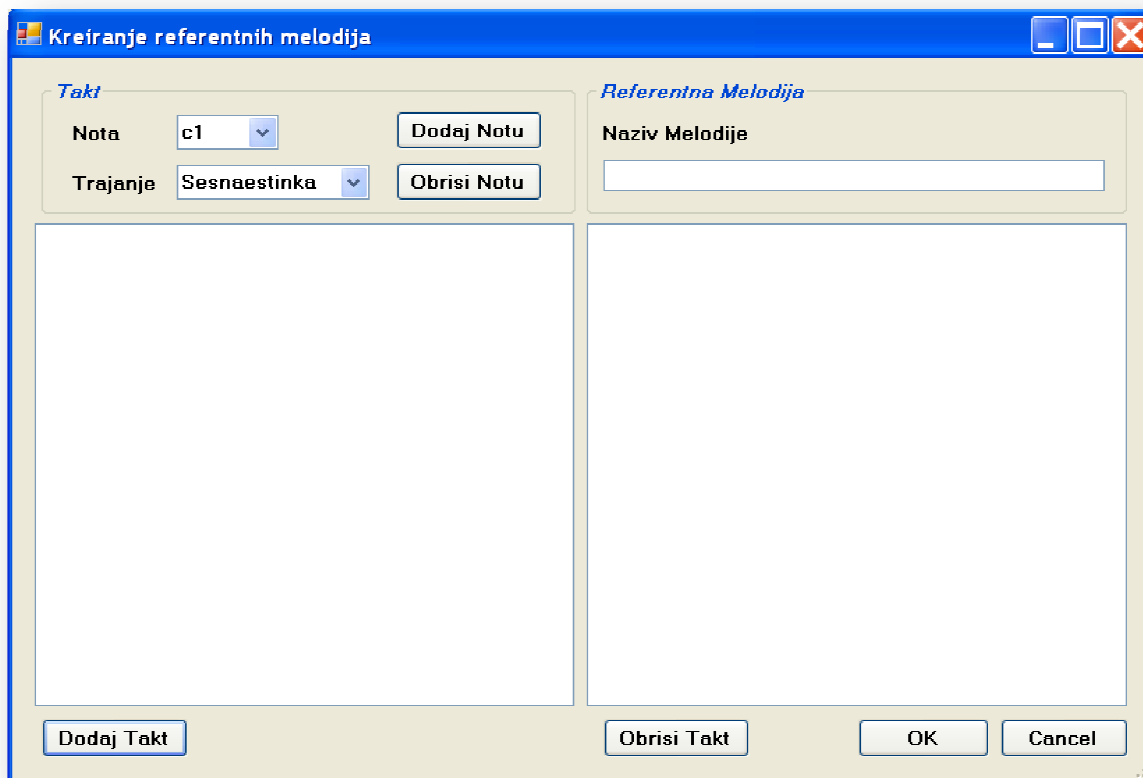
1. Windows Forms aplikacija

Sučelje aplikacije je podijeljeno u tri osnovne grupe: „Referentna Melodija“, „Populacija“ i „Iteracija“. Radi lakšeg uočavanja one su označene plavom bojom, kao što se može vidjeti na Slici 8. Na dnu se nalazi tipka „Kraj“. Pritiskom na navedenu tipku program završava s radom. Glavninu sučelja čine lijevo i desno polje prikaza koja pripadaju grupi „Populacija“, odnosno „Iteracija“. U lijevom se polju ispisuju jedinke početne populacije, koja je generirana slučajnim odabirom. U desnom polju ispisuje se najbolja jedinka iz svake iteracije.



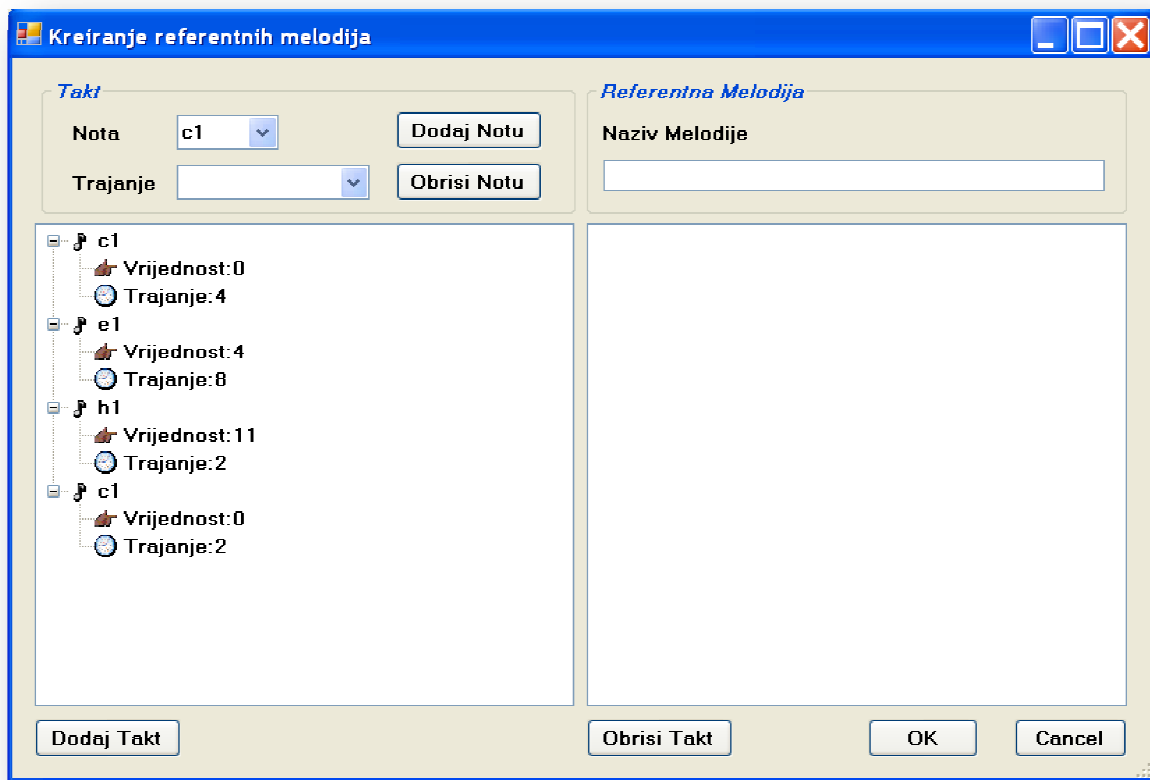
Slika 8. Izgled Windows Forms aplikacije prilikom pokretanja programa

Referentna melodija bira se iz liste postojećih. Korisnik može kreirati vlastite referentne melodije po želji i uspoređivati rezultate. Melodije se kod kreiranja automatski spremaju u datoteke, a naknadno se mogu i brisati izborom na listi i pritiskom na tipku „Obrisi“. Na početku izvođenja programa u padajućem izborniku se pokazuje lista spremljenih melodija. Inicijalno je spremljena samo referentna melodija „Mozart Mala Nočna Muzika 4 takta u G-duru“. Da bi se kreirala nova referentna melodija treba kliknuti na tipku „Dodaj“, nakon čega se otvara novi ekran prikazan na Slici 9.



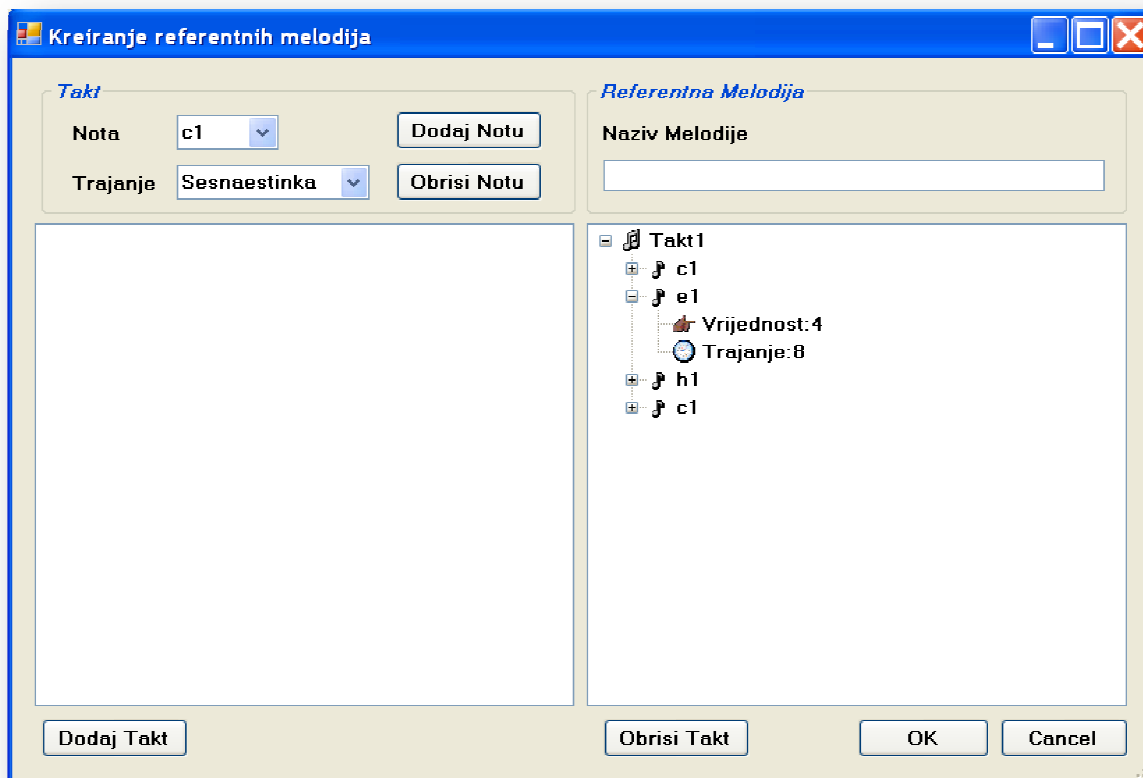
Slika 9. Primjer kreiranja referentne melodije

Za kreiranje referentne melodije odabire se nota i njezino trajanje. Pritiskom na tipku „Dodaj Notu“ ona se upisuje u takt. Postupak se ponavlja za sve note unutar jednog takta. Program vodi računa o trajanjima nota koje se unose (ako se unijela polovinka i četvrtinka u taj se takt ne može više unijeti cijela nota ili polovinka te program ta trajanja neće niti ponuditi u padajućem izborniku) kao i o popunjenosti takta (ukupno trajanje takta mora biti 16 s time da ovdje trajanje u vrijednosti 1 predstavlja notu šesnaestinku). Evo kako izgleda jedan primjer:



Slika 10. Primjer kreiranja takta referentne melodije

Pritiskom na tipku „Dodaj Takt“, takt se seli u desno polje. Ukoliko takt nije do kraja popunjen javlja se odgovarajuća poruka. Gornji je takt popunjen i pritiskom na tipku „Dodaj Takt“ dobiva se sljedeće:



Slika 11. Primjer iskreiranog takta referentne melodije

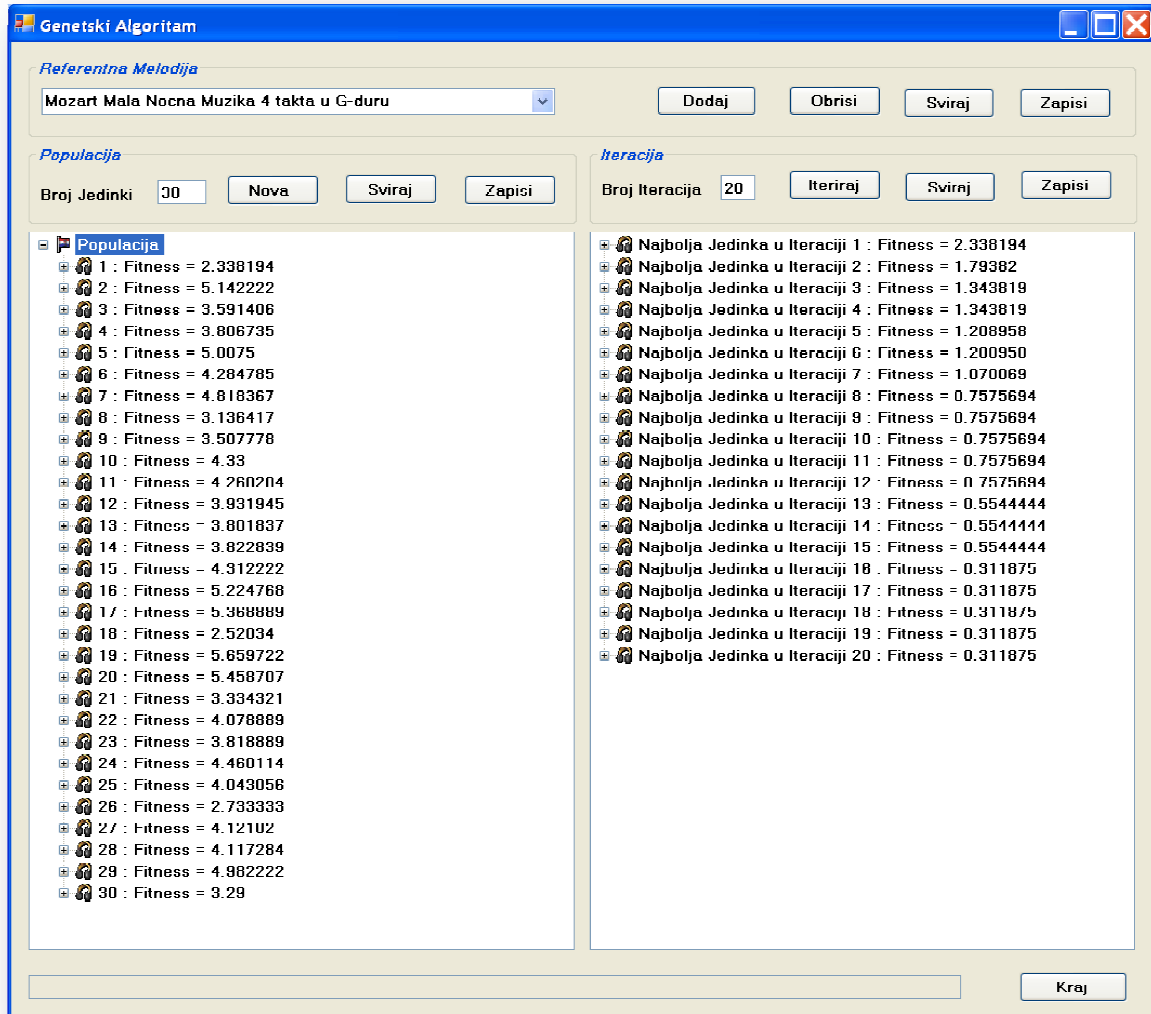
Na isti se način kreiraju i preostali taktovi. Kada je postupak završen, referentna melodija se sprema pritiskom na tipku „OK“. Pri tome treba odabrati naziv melodije. Pošto se kreirana melodija automatski sprema u datoteku koja nosi isti naziv kao i ime melodije, za naziv nije dozvoljeno koristiti specijalne znakove (npr. ;,;, \). Postupak dodavanja referentnih melodija vrlo je koristan za detaljnije testiranje programa. U grupi „Referentna Melodija“ nalaze se još i tipke „Sviraj“ i „Zapisi“ koje će biti objašnjene nešto kasnije.

Grupa „Populacija“ sastoji se od polja za unos broja jedinki unutar populacije i tipki „Nova“ za kreiranje nove populacije te „Sviraj“ i „Zapisi“ koje će biti opisane u nastavku. Inicijalno, pri pokretanju programa automatski se kreira početna populacija. Tipka „Nova“ služi za kreiranje nove početne populacije i prvenstveno je namijenjena za testiranje. Pritiskom na tu tipku u lijevom polju prikaza pojavit će se nova populacija, a istovremeno će se obrisati jedinke u desnom polju. (Slika 8.).

Grupa „Iteracija“ sastoji se od polja za unos broja iteracija i tipki „Iteriraj“ za pokretanje iteracija te „Sviraj“ i „Zapisi“ koje će biti opisane kasnije. Pritiskom na tipku „Iteriraj“ program započinje proces popravljavanja početne populacije, a rezultat se prikazuje na desnom polju prikaza u vidu najbolje jedinke u svakoj iteraciji.

Lijevo i desno polje za prikaz izvedena su u stablastoj strukturi pa se tako svaka jedinka, bilo početne populacije, bilo jedinke dobivene iteracijama može detaljnije prikazati raščlanjivanjem na taktove, note i konačno notne vrijednosti i trajanja. Korištene su ikone radi boljeg estetskog dojma grafičkog prikaza.

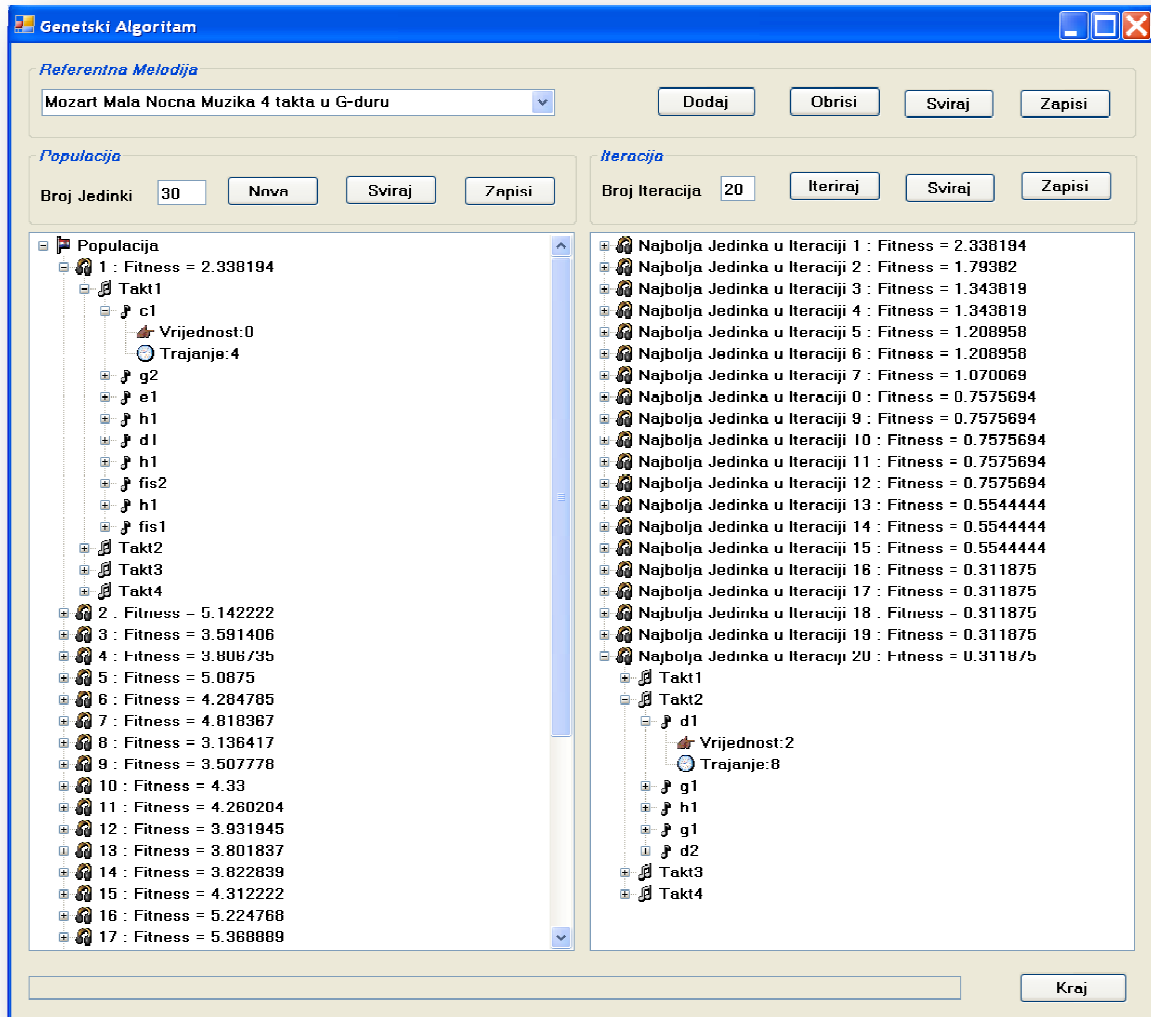
Uporabom objašnjenog sučelja vrlo je jednostavno testirati razvijeni program. Slika 12. prikazuje rezultate jednog takvog testa.



Slika 12. Test primjer

U lijevom polju prikaza nalazi se početna populacija. U desnom polju nalaze se rezultati 20 iteracija. Kao što je vidljivo sa slike, u početnoj populaciji vrijednosti funkcije dobrote kreću se od minimalne vrijednosti 2.33 do maksimalne vrijednosti 5.65. Na desnoj strani najbolja jedinka u prvoj iteraciji ima funkciju dobrote 2.33, a u posljednjoj iteraciji 0.311. Dakle, vidi se poboljšanje koje je nastalo primjenom opisanog algoritma.

Pored prikaza jedinki i populacije moguće je prikazati i pojedinu jedinku sa svim njenim detaljima, tj. taktovima i notama. Na Slici 13. prikazana je prva jedinka iz početne populacije (lijevo polje prikaza) i najbolja jedinka nakon 20 iteracija (desno polje).



Slika 13. Detaljan prikaz jedinke

1.1 Grafički prikaz jedinke

Svaku je jedinku moguće prikazati i u grafičkom, tj. notnom zapisu. Za prikaz u notnom zapisu koristi se program LilyPond. Program kao ulaznu informaciju uzima jednostavnu tekstualnu datoteku. Tekstualna datoteka treba biti napisana u točno definiranom formatu. Ovdje se neće ulaziti u detalje LilyPond formata već će samo biti prikazan ulaz za najbolju jedinku iz programa sa Slike 13. (jedinke u lijevom dijelu prozora).

```
\include "deutsch.ly"
{
\key g \major
\time 4/4
\clef treble
h'4 fis'16 h'16 fis'4 fis'16 d'8 g'16 g'16 g'16 d'2 g'8 h'8 g'8 d''8 g'8 d'2
h'8 g'16 e''8 h'16 d'16 g'16 h'16 e'4 c'8 a'4 fis''8 a'16
}
```

Prve četiri linije su kontrolne: Odabrana je njemačka notacija (`\include "deutsch.ly"`) jer se u njoj koristi nota „h“ (u pretpostavljenoj, engleskoj notaciji se umjesto note „h“ koristi oznaka „b“, što je pomalo zbunjujuće budući je nota „b“ zapravo snižena nota „h“). Nadalje, skladba je pisana u G-dur-u (`\key g \major`) uz četveročetvrtinski takt (`\time 4/4`). Kontrolna linija `\clef treble` definira da se na početku skladbe nalazi violinski ključ (osim violinskog ključa postoje još i bas ključ, tenor ključ, alt ključ itd.).

Program omogućuje da se odabrana jedinica (bila ona iz početne populacije, iz liste jedinica nakon iteracija ili pak referentna melodija) zapiše u LilyPond formatu i spremi na željeno mjesto na disku. To se postiže odabirom jedinice i pritiskom na odgovarajuću tipku „Zapisi“. Pri tome je pozicija za spremanje na disku prikazana na samom dnu sučelja. Nakon toga može se aktivirati LilyPond (dvostrukim klikom miša na zapisanu jedinku) i dobiti notni zapis u .PDF formatu. Na Slici 14. se nalazi takav zapis već spomenute najbolje jedinice.



Slika 14. LilyPond zapis jedinice

1.2 Audio zapis jedinke

Za potpunu ocjenu melodije najbolje je jedinku i odsvirati. LilyPond nažalost pruža samo grafički notni zapis iz ulaznog tekstualnog formata. Stoga se autor posvetio proučavanju čitavog niza programa koji na različite načine sviraju neku melodiju. No, ispalo je da su svi ti programi vrlo općeniti i pokrivaju čitavo područje muziciranja, pa su samim time i vrlo složeni (jedan od takvih je i program pod nazivom „Mozart“).

Polazeći od dvije jednostavne premise da je svaka nota definirana frekvencijom i trajanjem, te je programu na raspolaganje stavljen poprilično mali skup nota, za potrebe sviranja razvijen je vlastiti program. On u nizu nota prepoznaje pojedine note i njihova trajanja te ih zatim jednostavnom Windows komandom Beep i izvodi.

Trebalo je, dakle, pronaći frekvencije svake note i odrediti vrijeme trajanja pojedinih nota (cijela nota, polovinka, četvrtinka itd.).

Tablica 8. Notne frekvencije

<i>Nota</i>	<i>Frekvencija (Hz)</i>
c1	261.63
d1	293.66
e1	329.63
fis1	369.99
g1	392.00
a1	440.00
h1	493.88
c2	523.25
d2	587.33
e2	659.26
fis2	739.99
g2	783.99

Za računanje vremenskog trajanja note treba najprije ponoviti da cijela nota traje četiri dobe, polovinka polovinu toga, četvrtinka četvrtinu itd. S duge strane treba odabrati tempo koji je definiran brojem udaraca u minuti, s time da jedan udarac predstavlja trajanje jedne četvrtinke. U literaturi se može naći sljedeća formula za računanje vremenskog trajanja nota u milisekundama:

$$t = \frac{\textit{trajanje}}{\textit{rezolucija}} \times 60 \times 1000 \times 4 \times \frac{1}{\textit{tempo}}$$

Najčešće se za tempo uzima vrijednost 120. Razlomak trajanja i rezolucije za cijelu notu iznosi 1, a za svaku iduću ta se vrijednost umanjuje za pola (kod polovinke iznosi 0.5, kod četvrtinke 0.25 itd.). Trajanja u milisekundama su sljedeća:

cijela nota: $t = 2000$ ms

polovinka: $t = 1000$ ms

četvrtinka: $t = 500$ ms

osminka: $t = 250$ ms

šesnaestinka: $t = 125$ ms

Uz ova dva podatka (frekvencija i trajanje) može se pozvati sistemska funkcija `Beep` i tako na vrlo jednostavan način odsvirati niz nota. Upravo na taj način implementirana je funkcija sviranja. Odabirom jedinke i pritiskom na odgovarajuću tipku „Sviraj“ uzima se odabrana jedinka i njezine se note sviraju na opisan način.