

ساخت هوشمند موسیقی با الگوریتم ژنتیک مبتنی بر جهش یکنواخت موتیف

فریا نصیری مفخم^۱ و راضیه روغنی زاده^۲

۱- استادیار، دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران

fnasiri@eng.ui.ac.ir

۲- دانش آموخته کارشناسی فناوری اطلاعات، دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران

r_roghanizadeh@yahoo.com

چکیده: امروزه دنیای موسیقی و هوش مصنوعی به واسطه پژوهش‌های انجام شده در هر دو حوزه به هم نزدیک‌تر شده‌اند. ساخت موسیقی با استفاده از راه‌کارهای موجود در هوش مصنوعی، زمینه پژوهشی جدیدی را ایجاد کرده است. ساخت خودکار موسیقی علاوه بر اینکه ما را در شناخت بهتر چگونگی تفکر موسیقایی انسان یاری خواهد کرد، کمک شایان توجهی به آهنگ‌سازان و موسیقی‌دانان در بهبود تئوری‌های موسیقی با استفاده از قدرت محاسباتی کامپیوترها می‌کند و آهنگ‌سازان را نیز از آهنگ‌سازی‌های «نُت به نُت» رها خواهد ساخت. در این پژوهش، به ساخت موسیقی به شکل خودکار پرداخته می‌شود. ابتدا، با استفاده از زنجیره مارکوف و سیستم‌های لیندن‌مایر، یک سیستم آهنگ‌سازی خودکار پیاده‌سازی می‌شود که به دلیل ماهیت روش‌های مورد استفاده، به شکل بدون تنوع به تولید موسیقی می‌پردازد. ولی بر اساس الگوریتم ژنتیک با استفاده از جهش یکنواخت و توجه به برانزنگی قطعه براساس آکوردها، نتایج سریع به سمت قطعات بهتر می‌روند. خلاقیت در ساخت موسیقی فراتر از مرزهای کنونی هوش مصنوعی است و برای دستیابی به نتایج بهتر، کار بسیاری در این زمینه هنوز پیش رو است.

واژه‌های کلیدی: آکورد، الگوریتم ژنتیک، زنجیره‌ی مارکوف، ساخت موسیقی، سیستم لیندن‌مایر، هوش مصنوعی.

۱- مقدمه

آن را هنر می‌نامند. آهنگ‌سازی مانند هر فعالیت هنری دیگر، سرشار از انتخاب‌های آزادانه است که به وسیله آن، احساسات، خیالات یا الهامات بیان می‌شوند و نیاز به خلاقیت، دانش و توانایی‌های خاصی دارد. بسیاری از مکانیسم‌های دخیل در ساخت موسیقی هنوز ناشناخته هستند و بعضی دیگر نیز آن چنان پیچیده‌اند که ابزارهای کامپیوتری فعلی با وجود رشد و تحولی چشمگیر در مدت زمانی کوتاه، قادر به مدیریت آن‌ها نیستند. کامپیوترها آهنگ‌سازان را از آهنگ‌سازی نوت به نوت می‌توانند رها کنند و به آن‌ها اجازه دهند که تئوری‌های موسیقی و آهنگ‌سازی را به عنوان یک فرایند جستجو کنند بدون آن‌که نگران نت‌های واقعی باشند. فرآیندهای ادراکی که شامل منطق، دانش و احساسات هستند، گرچه به سختی توسط یک کامپیوتر نمایش داده می‌شوند، برای پژوهش‌های هوش

موسیقی یکی از مقوله‌هایی است که درک و هوش انسان بر دلیل زیبایی آن تسلط کامل ندارد. موسیقی، هنر بیان احساسات به وسیله اصوات است و حالت درونی انسان و موجودات زنده را متحول می‌کند. ارسطو موسیقی را یکی از شاخه‌های ریاضی می‌دانسته ولی چون همه ویژگی‌های موسیقی مانند ریاضی مسلم و غیرقابل تغییر نیست و ذوق و قریحه سازنده و نوازنده در آن دخالت دارد،

^۱ تاریخ ارسال مقاله : ۱۳۹۲/۵/۲۸

تاریخ پذیرش مقاله : ۱۳۹۳/۳/۳

نام نویسنده مسئول : فریا نصیری مفخم

نشانی نویسنده مسئول : ایران - اصفهان - دانشگاه اصفهان - دانشکده مهندسی کامپیوتر - گروه مهندسی فناوری اطلاعات

زمانی مشخص انجام می‌دهد. به طول هر تناوب کامل طول موج می‌گویند که واحد آن متر است. هر چقدر طول موج کمتر باشد صدا «زیر» تر و هرچه بیشتر باشد صدا «بم» تر است.

دامنه: این ویژگی بیانگر حداکثر مسافتی است که جسم مرتعش از نقطه تعادل خود به طرفین طی می‌کند. هرچه دامنه صدا بزرگ‌تر باشد صوت قوی‌تر است.

شیوش: همان رنگ یا طنین صوتی است که با توجه به شکل موج صدا تغییر می‌کند.

در مبحث تئوری موسیقی، صدا دارای ویژگی‌های زیر است [۶، ۷]:

نواک: واضح است که صداهای موسیقایی می‌توانند در سطح‌های مختلف زیر و بمی حاصل شوند. در خط موسیقی نواک را با کمک «حامل» نشان می‌دهند (شکل ۱). نت‌های با نواک مختلف بسته به نواک آن در بین و روی خطوط قرار می‌گیرند.

دیرند: مدت زمانی را که هر صدای موسیقایی ادامه می‌یابد دیرند گویند. روشن است که یک آهنگ موسیقی از صداهایی تشکیل شده که بیشتر ارزش‌های زمانی متفاوتی دارند، یعنی برخی کوتاه‌تر و برخی کشیده‌تر هستند. در خط موسیقی دیرندهای مختلف را با شکل مخصوص در هر نت نشان می‌دهند.

شدت: همان قوی و ضعیف بودن اصوات نسبت به هم است. در خط موسیقی برای نمایش آن‌ها از علائمی به شکل حروف استفاده می‌شود.

طنین و رنگ: همان صداهای مختلف است که شنیده می‌شود و هر ساز رنگ خاص خود را دارد.



شکل (۱): خط حامل

نت‌های موسیقی: برای همه صداهای موسیقی، از بم‌ترین تا زیرترین آن‌ها، فقط هفت نام وجود دارد. این نام‌های هفت‌گانه در برخی کشورهای غربی با کلمات تک‌هجایی، و در برخی به شکل الفبایی تلفظ می‌شوند شکل (۲). وسعت صداهای اصلی موسیقی، یعنی پهنه‌ای از

مصنوعی بسیار مناسب بوده و بنابراین، استفاده از ماشین‌ها برای آهنگ‌سازی موضوع بسیار جالبی برای پژوهش و بررسی است.

تکنیک‌های مختلف هوش مصنوعی که برای درک توانایی‌های موسیقایی انسان در زمینه آهنگ‌سازی با کامپیوتر به کار برده شده‌اند در دو گروه تحلیل آهنگ و آهنگ‌سازی طبقه‌بندی می‌شوند. تحلیل کردن آهنگ شامل استخراج اطلاعات از خود آهنگ به منظور یادگیری قوانین یا به دست آوردن مدلی برای توصیف قطعات موسیقی است. آهنگ‌سازی، تولید موزیک جدید از روی قوانین است. در موسیقی ابتدا قطعاتی ساخته شده و سپس، قوانین از قطعات خلق شده استخراج می‌شود، اما در آهنگ‌سازی خودکار این فرآیند به شکل معکوس است یعنی از قوانین شناخته شده در موسیقی در ایجاد قطعه‌های جدید استفاده می‌شود. تلاش‌های اولیه در آهنگ‌سازی الگوریتمیک شامل انتخاب کردن نت‌ها (به شکل اصلی نواک‌ها و ریتم‌ها) با یک سری محدودیت‌ها به منظور تولید موزیک بود. در این حوزه نتایج محدودی به دست آمد اما در پیشرفت کارهای بعدی نقش خارق‌العاده داشت [۱] و در پژوهش‌های بین رشته‌ای بسیاری دنبال می‌شود [۲-۵]. در این مقاله، یک آهنگ‌ساز خودکار با استفاده از راهکارهای موجود در هوش مصنوعی پیاده‌سازی و بهبود این راهکارها بررسی می‌شود.

۲- تئوری موسیقی

آهنگساز در ساختن موسیقی، صداها و عوامل صوتی را به کار می‌بندد. صوت، پدیده‌ای که انسان آن را به یاری دستگاه شنوایی خود درک می‌کند، در نتیجه ارتعاش یک جسم تولید می‌شود و در محیط مادی مانند هوا یا آب به شکل موج انتشار می‌یابد و به گوش ما می‌رسد. حرکت موجی صدا دارای ویژگی‌های زیر است [۶]:

بسامد: تعداد نوسان در مدت زمانی مشخص را بسامد گویند. اگر زمان اندازه‌گیری نوسان‌ها یک ثانیه باشد، واحد آن هرتز است. هر قدر بسامد بیشتر باشد صدا «زیر» تر و هرچه کمتر باشد صدا «بم» تر است.

طول موج: جسم مرتعش هر تناوب کامل را در مدت

جدول (۱): دیرند و شکل نت‌ها [۶]

نمایش نت	نام نت	ارزش زمانی
	گرد، کامل	چهار واحد زمانی
	سفید	دو واحد زمانی
	سیاه	یک واحد زمانی
	چنگ	۱/۲ واحد زمانی
	دولاچنگ	۱/۴ واحد زمانی
...

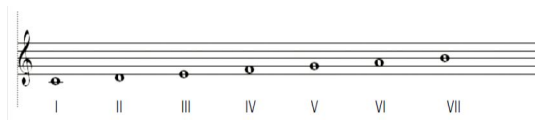
فاصله: با توجه به اینکه نت‌ها به شکل زیر و بم هستند، میان هر دو نت فاصله‌ای وجود دارد. بنابراین، فاصله در موسیقی عبارت از مسافت میان هر دو نت است. واحد فواصل موسیقی به شکل پرده و نیم پرده است. فواصل نت‌ها به شکل خوشایند و ناخوشایند به گوش ما می‌رسند و این خود یکی از عوامل اصلی ساخت موسیقی است [۶، ۷].

گام: عبارت است از تعدادی اصوات پی در پی که با فاصله‌های مشخص و حساب شده به دنبال هم قرار می‌گیرند و آخرین نت آن یک اکتاو بالاتر از نت اول است. نام هر گام با توجه به نت اول و فواصل بین نت‌های آن تعیین می‌شود. در شکل (۳) گام دو ماژور دیده می‌شود [۶، ۷].



شکل (۳): گام دو ماژور [۶]

ویژگی و هویتی است که آن را از سایر درجات متمایز می‌کند. در موسیقی همین تغییر درجات است که سبب ایجاد آهنگ می‌شود [۶، ۷].



شکل (۴): گام دو ماژور به همراه درجات آن [۶]

می‌شود. به این شکل که شروع آکورد با نت دوم (معکوس اول) یا نت سوم (معکوس دوم) آن باشد. در شکل ۵ چند آکورد به همراه نوع، اسم و نوع معکوس آن آمده است [۶، ۷].

بم‌ترین تا زیرترین اصوات، متجاوز از ۶۰ صوت را در بر می‌گیرد. در حالی که نت‌ها فقط هفت نام دارند. در واقع در این پهنه گسترده پس از هفتمین نت، تمام آن‌ها از آغاز تکرار می‌شود. به عبارت دیگر، در ادامه پهنه اصوات موسیقی همواره پیش از هر نت «دو»، نت «سی» و پس از هر نت «سی»، نت «دو» قرار دارد. به فاصله بین دو نت همنام متوالی اکتاو می‌گویند [۶، ۷].

سی	لا	سل	فا	می	ر	دو
Si	La	Sol	Fa	Mi	Re	Do
B	A	G	F	E	D	C

شکل (۲): نام نت‌های موسیقی [۶]

شکل نت‌ها: به منظور نمایش دیرند نت‌ها برای هر یک از شکل ویژه‌ای استفاده می‌شود جدول (۱). با توجه به این که نت سیاه ارزش یک واحد زمانی دارد، نت‌های بالاتر ارزش دو برابری و نت‌های پایین‌تر ارزش یک دوم برابر دارند [۶، ۷].

هر نت در گام دارای درجه‌است به طوری که نت اول درجه ۱ و به همین ترتیب تا درجه ۷ ادامه می‌یابد که با اعداد یونانی نمایش داده می‌شود. هر یک از درجات دارای

آکورد: به چند صدا که در یک زمان با هم شنیده می‌شوند و به عبارتی، به نواختن همزمان چند نت با هم کورد (یا آکورد) می‌گویند. امروزه بسیاری از آهنگ‌ها را با آکورد می‌نوازند با افزودن نت‌هایی به فاصله سوم و پنجم به یک درجه از یک گام، کورد به شکل‌های مختلف ایجاد

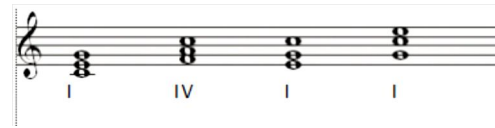
۳-۱- معرفی راهکارهای هوش مصنوعی در

موسیقی

سیستم‌هایی که با استفاده از هوش مصنوعی به آهنگ‌سازی می‌پردازند از راهکارهای متعددی بهره می‌گیرند. بیشتر راهکارهای استفاده شده عبارت هستند از: مدل‌های ریاضی، سیستم‌های بر پایه دانش، گرامرها، مدل‌های تکاملی، سیستم‌های یادگیرنده، و سیستم‌های ترکیبی [۸].

مدل‌های ریاضی: فرآیندهای تصادفی و به ویژه زنجیره‌ی مارکوف به شکل گسترده‌ای برای آهنگ‌سازی الگوریتمیک استفاده شده‌اند [۵]. آهنگ‌سازسایرناتیک که به وسیله‌ی ایمز و دومینو در سال ۱۹۹۲ ساخته شد یک مثال بارز از چنین سیستمی است. این برنامه قطعه‌های مختلف از ژانرهای مختلف را ترکیب می‌کند. یکی از ویژگی‌های جالب این سیستم این است که اول ریتم ملودی را با استفاده از زنجیره‌های مارکوف استنباط می‌کند و سپس، زیر و بمی‌های صدا را در مرحله بعد انتخاب می‌کند. همچنین، مدل‌های محاسباتی وجود دارند که بر پایه سیستم‌های غیرخطی آشوب یا توابع تکراری هستند ولی کیفیت خروجی این سیستم‌ها مورد تردید است. زیرا برخلاف همه مدل‌های دیگر دانش موسیقایی آن‌ها از انسان یا کارهای انسان مشتق نشده است. مهم‌ترین معایب این سیستم‌ها این است که برای شبیه‌سازی یک سبک موسیقی، باید قطعه‌های بسیاری تحلیل و احتمالات آن‌ها پیدا شود و دیگر اینکه در نظر گرفتن معیارهای سطح بالاتر موسیقی بسیار دشوار است [۱، ۸].

سیستم‌های بر پایه دانش: بیشتر سیستم‌های هوش مصنوعی سیستم‌های بر پایه دانش هستند. استفاده از این سیستم‌ها برای معرفی ساختارها یا قوانین صریح، منطقی به نظر می‌رسد. مزیت اصلی آن‌ها این است که منطقی روشن دارند و به عبارتی می‌توانند انتخاب اعمالشان را توضیح دهند [۹]. برای مثال، هارمونیزه کردن معمولاً به عنوان یک مسئله رضایت‌مندی محدود^۱ در نظر گرفته می‌شود که با



کورد اول کورد اول کورد اول کورد اول
دوماژور دوماژور فاماژور دوماژور
معکوس دوم معکوس اول اصلی اصلی

شکل (۵): نمونه آکورد [۶]

موتیف: در تئوری موسیقی کوچک‌ترین واحد معنایی موسیقی، موتیف نامیده می‌شود. موتیف مانند یک سلول زنده در بدن جانداران است، یعنی کوچک‌ترین واحد دارای نغمه‌های مختلف با وزن مشخص. تمام موسیقی از موتیف شکل می‌گیرد. توسط قواعد موسیقی نه تنها می‌توان فقط با تکرار موتیف جمله موسیقایی ساخت، بلکه می‌توان با تکرار وارونه قرینه یک موتیف و بسیاری حالات ریاضی آهنگ تولید کرد. در شکل (۶) موتیف یک جمله موسیقی مشخص شده است [۶، ۷].



شکل (۶): نمونه موتیف در یک جمله موسیقی [۶]

۳- مروری بر مفاهیم و آهنگ‌سازی کامپیوتری

با هوش مصنوعی

اولین نمونه آهنگ‌سازی با اعداد تصادفی کار موزارت است که به Dice Game مشهور بوده و شامل به هم چسباندن قطعات موسیقایی کوچک به شکل تصادفی و تولید یک قطعه‌ی موسیقی است. همین ایده ساده، اساس بخشی از روش‌های نوین در ساخت موسیقی توسط الگوریتم‌های هوش مصنوعی است [۱].

با توجه به ارتباط تنگاتنگ موسیقی با احساس و ذوق آهنگ‌ساز، مشکل اصلی طراحی الگوریتم‌های ساخت موسیقی، برقراری رابطه‌ای ریاضی برای تعیین میزان مقبولیت یک قطعه موسیقی است.

¹ Constraint Satisfaction Problem

سیستم‌های یادگیرنده: سیستم‌های یادگیرنده سیستم‌هایی هستند که به شکل عمومی یک دانش از قبل تعریف شده به شکل قانون یا قید ندارند ولی به جای آن از طریق مثال‌ها یاد می‌گیرند [۱۴]. می‌توان این سیستم‌ها را بر اساس روشی که اطلاعات را نگه می‌دارند به دو گروه اصلی تقسیم کرد:

شبکه‌های عصبی مصنوعی^۳ (ANN): ANN‌ها در سال‌های اخیر به شکل گسترده‌ای در زمینه موسیقی استفاده شده و موفق بوده‌اند. موزار در سال ۱۹۹۴ با استفاده از ANN ملودی‌هایی تولید کرد که قابل قبول اما فاقد یک انسجام درونی و کلی بودند [۸، ۱].

یادگیری ماشین^۴ (ML): ML‌ها در این زمینه زیاد استفاده شده‌اند. ویدمور در سال ۱۹۹۲ برای هارمونیزه کردن ملودی‌ها از ML استفاده کرد. کار کوپ هم می‌تواند در این دسته قرار بگیرد. با اینکه خیلی جالب است که سیستمی از طریق مثال یاد بگیرد اما این یادگیری در موارد بسیاری نیاز به پالایش دارد زیرا مثال‌های متعددی می‌توانند با هم تضاد داشته باشند [۸، ۱].

سیستم‌های هیبرید: این سیستم‌ها از روش‌های مختلف هوش مصنوعی استفاده می‌کنند. سیستم‌های هیبرید معمولاً پیچیده هستند به ویژه در مواردی که سیستم‌ها به سختی با هم ترکیب شده‌اند [۸، ۱]. در نهایت، عیب اساسی در سیستم‌ها این است که با آهنگ‌سازی به عنوان یک کار حل مساله برخورد می‌کنند تا مساله‌ای که با خلاقیت همراه بوده و فرایندی با مفهوم است.

۳-۲- پروژه‌هایی در زمینه آهنگ‌سازی با کامپیوتر

هیلر و ایساکسون و بیکر از اولین کسانی بودند که موفق شدند به وسیله کامپیوتر آهنگ‌سازی کنند [۸]. در سال ۱۹۵۷ آن‌ها توانستند با استفاده از کامپیوتر Iliac Iliac suite را تولید کنند. هیلر و ایساکسون کوشش کردند تا فرایند آهنگ‌سازی را روی کامپیوتر شبیه سازی کنند. آن‌ها

محدودیت‌هایی مانند محدوده‌ی صدا و هدایت صدا و غیره درگیر است. روی و پچت در سال ۱۹۹۸ از برنامه‌نویسی منطقی محدود^۱ و تکنیک‌های رضایت‌مندی محدود برای هارمونیزه کردن استفاده می‌کنند. اگرچه سیستم‌های بر پایه دانش یک روش مناسب به نظر می‌رسد، اما این‌ها هنوز هم مشکلات زیادی دارند. یک دلیل مهم این است که استنباط دانش در زمینه‌های مفهومی و انتزاعی مانند موسیقی دشوار است و به دلیل اینکه در حوزه موسیقی هزاران استثنا در هر قانون می‌تواند وجود داشته باشد این سیستم‌ها بسیار پیچیده هستند [۸، ۱].

گرامرها: ایده‌ی وجود داشتن یک گرامر از موسیقی احتمالاً به اندازه‌ی ایده خود گرامر قدیمی است [۵]. یک مثال معروف در این زمینه، EMI^۲ کار دیوید کوپ است که بر اساس یافتن الگوها از کارهای یک آهنگ‌ساز و سپس، ساخت یک قطعه جدید بر اساس کارهای داده شده بود. اما برخلاف بیشتر موسیقی‌ها، گرامرها سلسله مراتبی هستند و درباره‌ی معنای موسیقی ساخته شده نیز هیچ ادعایی نمی‌توانند داشته باشند [۸، ۱].

مدل‌های تکاملی: ثابت شده که الگوریتم‌های ژنتیک برای جستجو در فضاها بزرگ بسیار کارآمد هستند [۲۴]. می‌توان تلاش‌های انجام گرفته در این زمینه را براساس تابع ارزیابی آن‌ها به دو گروه اصلی تقسیم کرد. در گروه اول از یک تابع ارزیابی عینی برای ارزیابی کروموزوم‌ها استفاده می‌شود [۱۰، ۱۱]. البته عملکرد این الگوریتم‌های ژنتیک هیچ‌گاه نمی‌تواند رفتار آدمی را شبیه‌سازی کنند. در گروه دوم، از یک انسان به عنوان تابع ارزیابی استفاده می‌شود که به این نوع از الگوریتم ژنتیک، الگوریتم ژنتیک تعاملی گفته می‌شود [۱۲]. مشکل اساسی این نوع از الگوریتم ژنتیک، ایجاد یک گلوگاه در ارزیاب است زیرا او باید تمام کروموزوم‌ها را بشنود و امتیاز دهد. این الگوریتم‌های ژنتیک درباره فرایند ذهنی آهنگ‌سازی چیز زیادی برای گفتن ندارند زیرا تمام منطق ارزیابی به شکل دست نیافتنی در ذهن آن فرد کد شده است [۱۳].

³ Artificial Neural Network

⁴ Machine Learning

¹ Constraint Logic Programming

² Experiments in Musical Intelligence

یک سوال دیگر که پژوهشگران هوش مصنوعی باید پاسخ دهند این است که «آیا می‌خواهیم خلاقیت را شبیه‌سازی کنیم یا نتایج حاصل از خلاقیت را؟». عیب بیشتر مدل‌های محاسباتی در این است که موسیقی تولید شده بی‌معناست. کامپیوترها احساس ندارند و برخلاف آدم‌ها نمی‌خواهند با موسیقی خود چیزی را بیان کنند. پس چگونه می‌توان مفاهیمی مانند معنا را در سیستم‌های الکترونیکی گنجانند؟

۳-۴- نظریه آشوب و سیستم‌های دینامیک غیر خطی

در سال‌های اخیر، رفتار سیستم‌های دینامیک غیرخطی باعث ایجاد علاقه در استفاده از آن‌ها به عنوان الگوریتم‌های تولید نوت در موسیقی شده است [۱۶]. این سیستم‌ها بیانگر رفتارهای بسیاری از سیستم‌هایی هستند که در طبیعت یافت می‌شوند مانند هوا، ترکیب مایعات، ضربان قلب انسان و سایر آن. به منظور تولید داده‌هایی که بتوانند برای موسیقی استفاده شوند این سیستم‌های معادلات در یک فرایند تکرار شونده قرار می‌گیرند که به موجب آن راه حل محاسبه می‌شود و برای تکرار بعدی به عنوان ورودی به معادلات داده می‌شود. این راه حل یک نقطه را در فضای n بعدی نشان می‌دهد که n بیانگر تعداد متغیرهاست. به مجموعه این راه حل‌ها در طول زمان، یک چرخش از سیستم گفته می‌شود. این سیستم‌ها می‌توانند در تکرارها سه نوع رفتار از خود نشان دهند، که بیشتر به عنوان جاذب‌ها شناخته می‌شوند. دلیل این نام‌گذاری این است که یک سری نقاط را نشان می‌دهند که مقادیر سیستم به آن گرایش دارند. این سه نوع رفتار به شرح زیر هستند [۱]:

ثابت: که در آن تمام نقطه‌ها در چرخش یکی هستند؛

نوسانی: که در آن تمام نقاط به یک مجموعه تکراری از نقاط تعلق دارند؛

بی‌نظم: که به نظر می‌رسد که چرخش‌ها رفتاری تصادفی دارند و هیچ وقت دو بار یک نقطه را ملاقات نمی‌کنند و به وضوح رفتار ثابت هیچ نتیجه موسیقایی جالبی را تولید نمی‌کند.
رفتار نوسانی می‌تواند تکرارهای جالبی را تولید کند به

راهکار تولید، اصلاح و انتخاب را برای شبیه‌سازی فرآیند آهنگ‌سازی به کار بردند. این راهکار از یک سری روش به منظور تولید یک سری مواد اولیه و خام استفاده می‌کند. سپس، روش‌های متعددی را برای دستکاری این مواد تولید شده به کار می‌بندد و سرانجام یک سری قوانین انتخاب را به کار می‌برد تا موارد مناسب را برای آهنگ‌سازی انتخاب کند. Illiac Suite از طریق روش حل تولید و آزمون به آهنگ‌سازی می‌پردازد. این برنامه به شکل تصادفی نت‌ها را با استفاده از زنجیره‌ی مارکوف تولید می‌کند. نت‌های تولید شده در مرحله بعد به وسیله هیورستیک قوانین آهنگ‌سازی کلاسیک آزمایش می‌شوند و فقط نت‌هایی که قوانین را ارضا می‌کنند باقی می‌مانند. اگر تمام نت‌ها شکست بخورند در ارضای قوانین یک برگشت به عقب ساده انجام می‌شود و دوباره از نقطه‌ای که قوانین ارضا شده به تولید آهنگ می‌پردازد. هیلر و ایساکسون هیچ‌گونه احساس و بیانی را در کار خود در نظر نگرفتند. پروژه دیگر آهنگ‌سازی کار مورر بود. مورر با استفاده از اصول آهنگ‌سازی ملودی ساده‌ای ایجاد و سپس، با تکرار این ملودی‌های ساده آهنگ‌سازی می‌کرد. کار او بیشتر بر اساس شبیه‌سازی اصول آهنگ‌سازی بود [۱۵]. مورالس مانزارانس در سال ۱۹۷۰ سیستمی به نام SCIB را توسعه داد که ساخت موسیقی را براساس حرکات بدن انجام می‌داد. این سیستم اطلاعات را از سنسورهای بدن انسان می‌گرفت و بی‌درنگ براساس اطلاعات دریافتی موسیقی مناسبی را ایجاد می‌کرد. یکی دیگر از سیستم‌ها EMI است که توسط دیوید کوپ در سال ۱۹۸۹ ابداع شد و موسیقی را بر اساس سبک یک آهنگ‌ساز می‌ساخت. او دو قطعه از آهنگ‌ساز مورد نظر را تحلیل و الگوها، قوانین و محدودیت‌هایی که آن آهنگ‌ساز در قطعه خود داشت را استخراج می‌کرد و بر اساس آن‌ها به ساخت آهنگ می‌پرداخت [۸، ۱].

۳-۳- خلاقیت محاسباتی

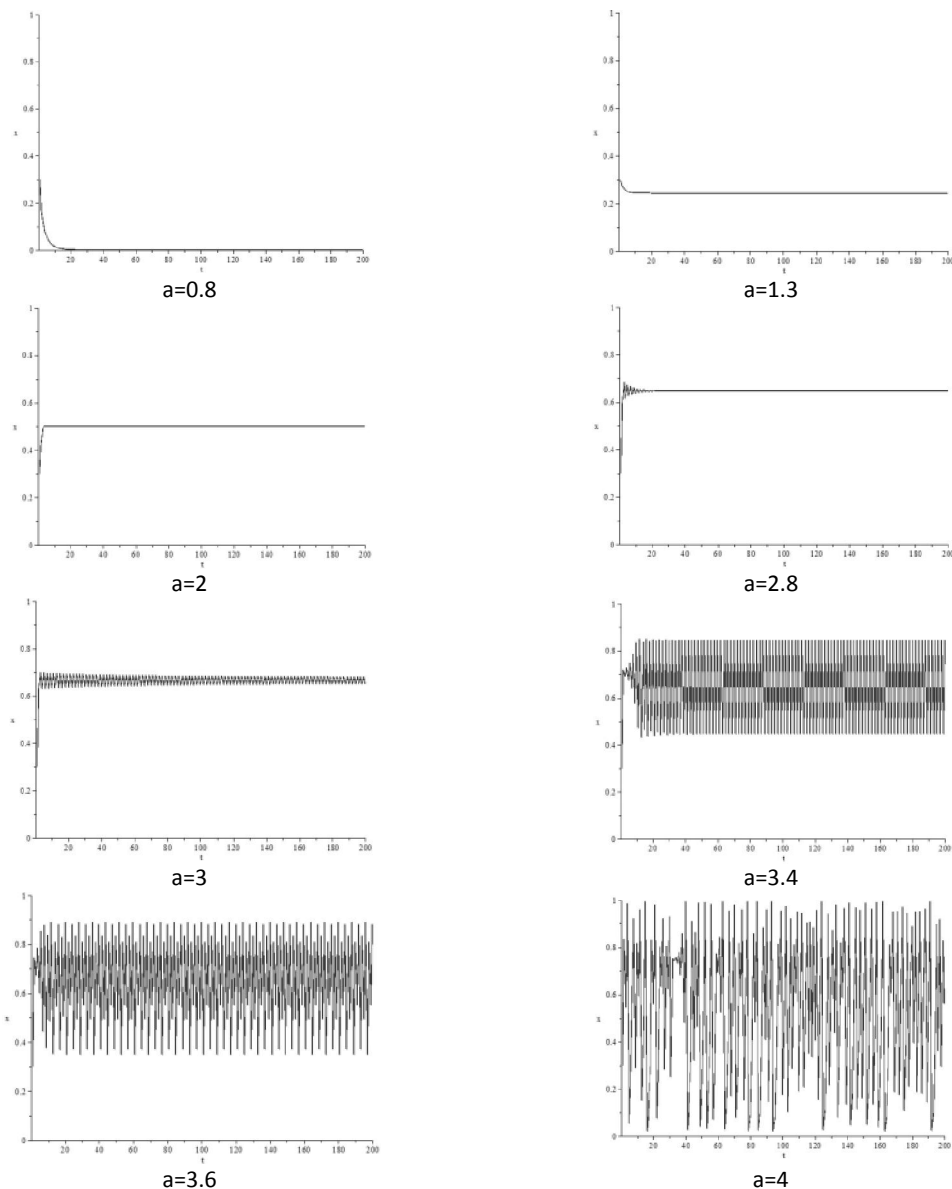
ممکن است سخت‌ترین کار، گنجاندن مفهوم خلاقیت در سیستم‌ها باشد. شبیه‌سازی خلاقیت به این دلیل مشکل است که به طور واضح نمی‌دانیم که خلاقیت چیست [۸].

موسیقی به این قضیه نگاه شود این موضوع می تواند به عنوان تغییرات در تم موسیقی دیده شود و به این دلیل است که چرخش های آشوب برای تولید اجزای موسیقی مناسب هستند. مقادیری که تولید می شوند همبستگی زیادی با گذشته شان دارند اما همیشه چیز جدیدی تولید می شود. مدل جمعیت نگاشت لجستیک یا معادله لورنز معادله (۱)، یک نمونه از این نوع سیستم هاست [۱].

$$x_{n+1} = ax_n(1 - x_n) \quad (۱)$$

شرط آن که دوره آن به اندازه کافی بزرگ باشد. ولی در عمل، مجموعه کمابیش کوچک است و باعث می شود ملودی های چرخشی ایجاد شود که فقط بین تعداد کمی از نواک ها چرخش می کنند. رفتار آخرین گروه است که بیشترین جذابیت را در موسیقی دارد.

جاذب های یک چرخش آشوب، یک بازه از مقادیر مشخص را تعریف می کنند که سیستم بین آن ها سرگردان است و بیشتر به مقادیر شبیه به هم برمی گردند اما دقیقاً به همان مجموعه از نقطه ها باز نمی گردند. وقتی از دید



شکل (۷): رفتار معادله لورنز [۱۷]

محدودیت آن در نمایش موسیقی چند بخشی اشاره کرد که دارای بیش از یک خط ملودی از نمادهای کورد هستند [۱]. در این مقاله از روش نمادگذاری ABC برای نمایش نت‌های موسیقی استفاده می‌شود. در شکل ۲، هفت کاراکتر ASCII مربوط به نت‌ها مشخص شده است. این علائم تنها ویژگی زیری و بمی اصوات را نشان می‌دهند که با توجه به نیاز این مقاله کافی خواهند بود.

۴- پیاده‌سازی سیستم ساخت موسیقی

در این بخش ابتدا پیاده‌سازی سیستم آهنگ‌سازی بر پایه احتمال و زنجیره مارکوف ارایه می‌شود. در این روش، نت‌های موسیقی را به شکل حروف در نظر گرفته، بر اساس الگو و قطعه‌ای که به عنوان ورودی داده شده گرامری را تشکیل داده و به شکل تصادفی آهنگ ساخته می‌شود. همان‌طور که گفته شد صوت دارای ویژگی‌های مختلفی است که در این روش تنها آهنگ بر اساس زیری و بمی اصوات و تغییر در آن ایجاد می‌شود. البته می‌توان با اعمال تغییرات جزئی الگوریتم را در مورد دیگر ویژگی‌های صوت نیز به کار برد. نکته دیگر در مورد این روش آن است که آهنگ ساخته شده در وسعت یک اکتاو است.

مراحل کار در این روش به این شکل است که ابتدا با استفاده از زنجیره مارکوف و ماتریس انتقال احتمال، یک مدل احتمالی از قطعه موجود ساخته می‌شود. سپس، با کمک سیستم لیندن‌مایر^۲ [۵]، الگویی از قطعه موجود به شکل گرامری به دست آمده، و پس از ترکیب با ماتریس انتقال احتمال، به شکل تصادفی یک قطعه موسیقی ساخته می‌شود. سپس، از روشی پیشرفته‌تر به نام الگوریتم ژنتیک بهره گرفته می‌شود که بر اساس پدیده طبیعی تکامل نسل‌ها استوار است. در این روش سعی بر این است تا با ترکیب رشته‌های ابتدایی قطعاتی که ویژگی‌های مطلوب دارند حاصل شود. بنابراین، یکی از مراحل کار تعریف تابعی است که معیاری از کیفیت موسیقی تولید شده را ارایه دهد.

این معادله یک معادله ساده یک بعدی است که برای مدل‌کردن رشد جمعیت در طول زمان استفاده می‌شود. این معادله خاص با اینکه ساده است، به ازای تکرارهای مختلف، یک مجموعه رفتار غنی به ازای مقادیر مختلف a به وجود می‌آورد. همان‌گونه که در شکل (۷) مشاهده می‌شود، برای مقادیر a بین صفر و یک، سیستم به نقطه $x=0$ همگرا می‌شود. برای تمام مقادیر a بین ۱ و ۳ متغیر x به نقطه ثابت $1-a$ همگرا می‌شود. برای مقادیر بالاتر از ۳، چرخه‌هایی وجود دارد که تمام مقادیر x بین مقادیر ۲، ۴ و ۸ چرخش می‌کنند. به این آبشار هارمونیک می‌گویند و به نقطه‌ای منجر می‌شود که به ازای مقادیر خاصی از a رفتار سیستم بی‌نظم می‌شود [۱، ۱۸، ۱۹]. این مقادیر که به رفتار آشوبی منجر می‌شوند مناسب‌ترین منطقه برای تفسیرهای موسیقایی هستند [۱].

۳-۵- روش نمایش نمادگذاری ABC

راه‌های مختلفی برای نمایش موسیقی در رایانه وجود دارد. قالب‌های معروفی چون MIDI، فایل‌های صوتی مختلف مانند Wave، WMA و MP3 و فایل‌های گرافیکی برای نمایش نت‌های نوشته شده است. برای آهنگ‌سازی به کمک رایانه که هدف این مقاله است باید نت‌ها را به شکل مشخص همراه ویژگی‌های آن‌ها نمایش داد و پردازش کرد. روش‌هایی که برای این کار مناسب هستند استفاده از قالب‌های گرافیکی یا فرمت MIDI است. یک روش ساده دیگر، نت نویسی به شکل ABC است که کار با آن ساده است و آن را می‌توان به هر قالبی تبدیل کرد. روش نمادگذاری ABC از یک ایده نت‌نویسی که در اواخر قرن ۱۹ میلادی استفاده می‌شده برگرفته شده است. در این روش از حروف A تا G برای نمایش نت‌ها استفاده می‌شود که نشان‌دهنده نواک نت‌هاست و سایر ویژگی‌ها را با علائمی به همراه آن نمایش می‌دهند. با ظهور رایانه مشخص شد که این روش قابلیت‌های زیادی برای استفاده در چنین محیطی دارد زیرا که هر نت توسط استاندارد ASCII^۱ قابل نمایش است. از معایب آن می‌توان به



شکل (۸): نمایش نت‌های قطعه مورد نظر

AACADAEDCCECGCGGBGCGDCFFAFCFCBAA
AAGCAAAAGEAAAAAGCAAAAAA

شکل (۹): نمایش ABC قطعه شکل ۸

	A	B	C	D	E	F	G
A	$\frac{16}{23}$	۰	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{1}{23}$	$\frac{3}{23}$
B	$\frac{1}{2}$	۰	۰	۰	۰	۰	$\frac{1}{2}$
C	$\frac{1}{4}$	$\frac{1}{12}$	$\frac{1}{12}$	۰	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{4}$
D	$\frac{1}{3}$	۰	$\frac{2}{3}$	۰	۰	۰	۰
E	$\frac{1}{4}$	۰	$\frac{1}{2}$	۰	۰	$\frac{1}{4}$	۰
F	$\frac{1}{4}$	۰	$\frac{1}{2}$	۰	۰	$\frac{1}{4}$	۰
G	۰	$\frac{1}{8}$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{8}$	۰	$\frac{1}{8}$

شکل (۱۰): ماتریس انتقال احتمال مربوط به قطعه شکل (۸) و

(۹)

برای مثال چون نت E، ۴ بار در ورودی مشاهده شده پس مخرج خانه‌های سطر E برابر ۴ است. فقط نت‌های A، C و F به ترتیب ۱، ۲ و ۱ بار بعد از نت E رخ داده‌اند. بنابراین، خانه‌های (E,A)، (E,C) و (E,F) برابر $\frac{1}{4}$ ، $\frac{1}{2}$ و $\frac{1}{4}$ هستند.

سیستم لیندن‌مایر: این سیستم یک سیستم بازنویسی موازی است که یک گرامر رسمی را معرفی می‌کند و از آن بیشتر برای مدل‌کردن رشد گیاهان استفاده می‌شود ولی به دلیل قابلیت آن در مدل‌کردن شکل یک نوع از ارگانیسم می‌توان از آن در ساخت قطعه موسیقی به شکل تصادفی و گرامری استفاده کرد. در این روش برخلاف روش بازنویسی متوالی گرامرهای چامسکی که نمادها پی‌درپی یکی در هر لحظه در رشته جایگزین می‌شوند، بازنویسی بر مجموعه‌های موازی از عناصر رشته بعد از رخداد آن‌ها در نمایش رشته انجام می‌شود [۵]. سیستم لیندن‌مایر دارای ویژگی بازگشتی است و می‌تواند به سمت تولید سیستم‌هایی مشابه سیستم کلی و با اجزایی شبیه به هم حرکت کند. سیستم لیندن‌مایر را می‌توان به شکل $G = (V, W, P)$ توصیف کرد، که:

V: الفبا، یک مجموعه از عناصر که می‌تواند جایگزین

۴-۱- الگوریتم آهنگ‌سازی تصادفی

زنجیره مارکوف: زنجیره مارکوف دنباله‌ای از متغیرهای تصادفی است که همه آن‌ها دارای فضای نمونه‌ای یکسان هستند اما توزیع احتمالات آن‌ها می‌تواند متفاوت باشد و در ضمن هر متغیر تصادفی در یک زنجیره مارکوف تنها به متغیر قبل از خود وابسته است [۵]. فرض کنید مجموعه S مجموعه کل حالت‌هایی باشد که یک سیستم می‌تواند اختیار کند و فرض کنید در هر مرحله از زمان، سیستم یا تغییر حالت می‌دهد یا همان حالت قبل را حفظ می‌کند. ماتریس انتقال احتمال P از S به این شکل تعریف می‌شود که اگر سیستم در یک مرحله زمانی در حالت S (عضو S) باشد، آن‌گاه احتمال آن که در مرحله بعدی زمانی در حالت t (عضو S) باشد $P(s,t)$ است. احتمال انتقال حالت از S در t از زمان و مسیر طی شده تا رسیدن به حالت S مستقل است. توجه داشته باشید که $P(s,s)$ احتمال باقی ماندن در حالت S در یک مرحله زمانی است. احتمال پیمودن هر مسیر نیز حاصل ضرب احتمال‌های هر مرحله زمانی است. چنین سیستمی را یک زنجیره مارکوف می‌نامند که با $M(S,P)$ نمایش داده می‌شود. زنجیره مارکوف را می‌توان به کمک گراف یا ماتریس انتقال احتمال نمایش داد. در اینجا از ماتریس انتقال احتمال استفاده می‌شود که سطر و ستون‌های آن نت‌های موسیقی است و هر خانه آن احتمال قرارگیری یک نت (ستون) پس از نت دیگر (سطر) را نشان می‌دهد. برای محاسبه احتمال‌ها تعداد کل اتفاقات یک نت را به عنوان کل حالات و تعداد اتفاقاتی که یک نت خاص پس از آن قرار گرفته به عنوان تعداد حالت‌های پیش‌آمد نت در نظر گرفته می‌شود. احتمال قرارگیری یک نت خاص پس از نت مورد نظر برابر است با حاصل تقسیم تعداد حالات پیش‌آمد به تعداد کل حالات. برای مثال اگر نت C، ۱۴ بار در آهنگ وجود داشته باشد و ۴ بار نت C پس از A قرار گرفته باشد، در خانه (A,C) ماتریس انتقال احتمال، مقدار $\frac{14}{4}$ قرار می‌گیرد.

در شکل (۸) یک قطعه موسیقی را در نظر گرفته، و در شکل (۹) نمایش ABC آن پیاده شده است. پس از تحلیل آن، ماتریس انتقال احتمال مربوطه محاسبه و به صورت شکل (۱۰) نشان داده می‌شود.

شوند؛

۴-۲- پیاده‌سازی الگوریتم آهنگ‌سازی تصادفی

بر اساس توضیحات ارائه شده، روند پیاده‌سازی این الگوریتم به این شکل است که قطعه ورودی را به شکل نمایش ABC از کاربر دریافت کرده، پس از تحلیل، قوانین تولیدی و احتمال آن‌ها محاسبه می‌شود. سپس، با توجه به طول قطعه درخواستی و نت آغازین با تکرار قوانین تولیدی آهنگ‌سازی انجام می‌شود. روند نمای این الگوریتم در شکل (۱۲) مشاهده می‌شود.

برای پیاده‌سازی این الگوریتم نیاز به تولید ماتریس انتقال احتمال و همچنین، گرامرهای استخراجی از قطعه مورد نظر است. در واقع این برنامه دارای دو قسمت کلی است که ابتدا قطعه ورودی را خوانده و با تحلیل آن ماتریس انتقال احتمال را به وجود می‌آورد. سپس، با توجه به دو شاخص طول قطعه درخواستی و نت شروع کننده و بر اساس گرامر به دست آمده از قسمت قبل به شکل تصادفی به تولید آهنگ می‌پردازد.



شکل (۱۱): یک قطعه حاصل از الگوریتم تصادفی بر اساس

قطعه مورد نظر

جدول (۲): گرامر حاصل از سیستم لیندن‌مایر و زنجیره

مارکوف برای مثال یاد شده در شکل ۱۱

p1: A	$\frac{1^6}{24}$	A
	$\frac{1}{24}$	C
	$\frac{1}{24}$	D
	$\frac{1}{24}$	E
	$\frac{1}{24}$	F
	$\frac{3}{24}$	G
p2: B	$\frac{1}{2}$	A
	$\frac{1}{2}$	G
...
p7: G	$\frac{1}{8}$	B
	$\frac{1}{2}$	C
	$\frac{1}{8}$	D
	$\frac{1}{8}$	E
	$\frac{1}{8}$	G

W : یکی از عناصر که حالت اولیه را تشکیل می‌دهد؛ و P یک مجموعه از قوانین تولیدی که دارای دو رشته هستند (مقدم‌ها و تالی‌ها). یک قانون به شکل $A \rightarrow B$ نمایش داده می‌شود که A از مقدم‌ها و B از تالی‌هاست. برای هر سمبل A در V که در سمت چپ قوانین تولیدی در P ظاهر نشده، قانون تولیدی $A \rightarrow A$ به شکل پیش‌فرض در نظر گرفته می‌شود و به آن ثابت می‌گویند.

قوانین سیستم لیندن‌مایر به شکل تکراری به کارگرفته می‌شوند. از حالت اولیه برای هر تکرار می‌توان قوانین مختلفی را به شکل همزمان به اجرا درآورد. تفاوت سیستم لیندن‌مایر با گرامر رسمی در همین مورد است. برای استفاده از این سیستم در آهنگ‌سازی، گرامر G به شکل (V, W, P) در نظر گرفته می‌شود که V به شکل مجموعه نت‌های موسیقی $\{A, B, C, \dots, G\}$ ، و حالت اولیه W بر اساس موسیقی مورد نظر از یک نت دلخواه انتخاب می‌شود. مجموعه قوانین P با استفاده از قطعه ورودی و تحلیل آن تولید می‌شود. بدین شکل که سمت چپ عملگر را یک نت موجود در نظر گرفته، و برای سمت راست آن یک یا چند نت انتخاب شده است.

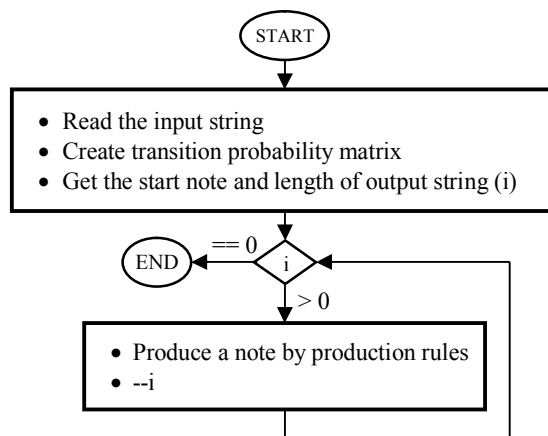
روش آهنگ‌سازی تصادفی با استفاده از زنجیره

مارکوف و سیستم لیندن‌مایر: با توجه به توضیحات ارائه شده در مورد زنجیره مارکوف و سیستم لیندن‌مایر می‌توان آن‌ها را به گونه‌ای ترکیب کرد که برای تولید موسیقی تصادفی بهینه باشد [۵]. در این روش قوانین تولیدی گرامر سیستم لیندن‌مایر با توجه به زنجیره مارکوف و ماتریس انتقال احتمال تولید می‌شود. در واقع برای هر درایه ماتریس یک قانون تولیدی در نظر گرفته می‌شود که سمت چپ آن سطر ماتریس و سمت راست آن ستون ماتریس باشد و احتمال درایه مورد نظر به عنوان احتمال به کارگیری آن قانون تولیدی به کار می‌رود. بدین ترتیب با تشکیل گرامر مورد نظر با توجه به قطعه ورودی و به کارگیری قوانین آن از حالت آغازین که همان نت شروع قطعه است به شکل تصادفی قطعه موسیقی جدیدی ایجاد می‌شود. در جدول (۲) قوانین تولیدی و احتمال به کارگیری آن‌ها در مورد قطعه‌ای که پیش‌تر یاد شده بود آورده شده است.

نت پس از نت مورد نظر درباره انتخاب یا عدم انتخاب آن به عنوان نت بعدی تصمیم گیری می شود. مقایسه عدد تصادفی دوم با میانه آن ($1/2$) کمک می کند تا در ساخت موسیقی تنوع بیشتری حاصل شود. به این شکل که اگر برای مثال نت آغازین D باشد و پس از D با احتمال $1/3$ به C و با احتمال $2/3$ به G رفته باشد، ابتدا دو عدد تصادفی بین ۰ تا ۱ تولید می شوند و بررسی می شود که آیا عدد اول از $1/3$ کوچک تر است یا خیر. در صورتی که این گونه نباشد سراغ عدد بعدی یعنی $2/3$ می رود و به همین ترتیب در صورت وجود گزینه های بعدی. در هر مرحله که شرط بزرگ تر بودن احتمال حضور یک نت پس از نت مورد نظر (D) نسبت به عدد تصادفی اول برقرار شد، این که آیا عدد تصادفی دوم از $1/2$ بزرگ تر است یا خیر، کنترل می شود. در صورت مثبت بودن پاسخ نت بعدی مشخص شده و یک نت جدید در انتهای رشته خروجی اضافه می شود. در غیر این صورت روند ادامه می یابد تا شروط بیان شده صدق کنند.

در صورتی که قوانین فراگیری شده کامل نباشند و الگوریتم هیچ انتخابی برای نت بعدی نداشته باشد پیغام خطای "Not enough rule to continue the process!" در صورتی که رشته ورودی DAAFEGAFGB و نت آغازین C باشد. با توجه به شرایط موجود قطعاً انتخاب بعد از C، نت B خواهد بود زیرا که C فقط یک بار آمده و پس از آن هم به B رفته است. حال هیچ قانونی برای نت بعد از B وجود ندارد زیرا آن هم فقط یکبار در انتها آمده است. در پایان، در صورتی که خطایی در مراحل میانی رخ نداده باشد رشته خروجی نمایش داده شده و با استفاده از تابع نواختن آهنگ، قطعه ساخته شده نواخته می شود.

نواختن آهنگ: تابع نواختن آهنگ برای تولید صدای نت ها در محیط Matlab نوشته شده است. Matlab امکان پخش یک آرایه به شکل یک فایل صوتی را از طریق تابع sound یا soundsc فراهم کرده است. بنابراین، کافی است یک آرایه تک فرکانس سینوسی با فرکانس نت های مختلف ساخته و پخش شود. این تابع یک ورودی دارد که



شکل (۱۲): روندنمای الگوریتم آهنگسازی تصادفی

تولید ماتریس انتقال احتمال: در پیاده سازی با استفاده از Matlab، تابعی با سه ورودی نوشته شده است. اولین ورودی یک رشته است که از کاراکترهای A تا G تشکیل می شود و برای طول آن محدودیتی وجود ندارد. دومین ورودی طول قطعه خروجی را مشخص می کند و آخرین ورودی هم یک کاراکتر از A تا G است که مشخص می کند قطعه خروجی با چه نتی شروع شود. تعداد تکرار هر نت و تعداد تکرار هر یک از نت ها پس از ظهور هر نت مورد نظر محاسبه می شود. در آخر با تقسیم تعداد تکرار هر نت پس از یک نت خاص به کل دفعات تکرار آن نت ماتریس انتقال احتمال به دست می آید.

تولید تصادفی آهنگ: قسمت بعدی کد مربوط به ساخت موسیقی بر اساس گرامر به دست آمده است. این کار به شکل یک حلقه تکراری است که نخستین گام آن جایگذاری نت تعریف شده به عنوان نت آغازین در رشته خروجی است. در صورتی که نت شروع کننده نتی باشد که برای آن قانونی تعریف نشده باشد یک پیغام خطا به کاربر داده می شود. برای مثال اگر رشته ورودی CDABGAGADDFAB باشد و نت آغازین E، مشخص است که نت E در رشته ورودی حضور ندارد و هیچ قانونی برای آن ساخته نشده است تا در مورد انتخاب نت بعدی بتوان تصمیمی گرفت. در چنین حالتی کاربر با پیغام "No rule has been made for initial note!" مواجه می شود. سپس، به شرطی که پیغام خطا رخ ندهد با کمک یک عدد تصادفی و مقایسه آن با احتمال حضور یک

خودکار خیلی محدود اما دارای پتانسیل بسیاری است. علاوه بر آن، مفهوم استفاده از روش‌های استخراج الگو، برای به دست آوردن قطعات اولیه یا موتیف‌های اصلی، قطعاتی از موسیقی را می‌سازد که در متون به شکل گسترده موشکافی نشده‌اند. این تا حدی شگفت‌آور است زیرا آهنگ‌سازان از موتیف‌ها برای ساخت آهنگ در طول قرن‌ها استفاده کرده‌اند. مسئله ساخت موسیقی بر پایه مجموعه‌ای از موتیف‌ها، احتمالاً اندکی بالاتر از مرزهای قابلیت‌های هوش مصنوعی است. از این رو این زمینه پژوهشی افق جدیدی را در ساخت خودکار موسیقی نمایان می‌کند. در این بخش، قطعات موسیقی را بر پایه مجموعه‌ای از موتیف‌های در حال تحول با استفاده از الگوریتم ژنتیک شکل داده و با بهبود تدریجی آن‌ها اقدام به آهنگ‌سازی می‌شود.

شرح نحوه عملکرد الگوریتم ژنتیک: الگوریتم ژنتیک یک روش بهینه‌سازی تصادفی بر پایه بهبود تکاملی در یک جمعیت با استفاده از انتخاب و تکثیر است. این الگوریتم که از طبیعت الهام گرفته شده، بر روی جمعیت کروموزوم‌ها عمل می‌کند. هر کروموزوم شامل تعدادی ژن است و هر ژن یکی از شاخص‌هایی که باید بهینه شود را نشان می‌دهد [۲۱، ۲۲]. ابتدا یک سری موتیف (کروموزوم) با طول ۱۶ نت به شکل تصادفی تولید می‌شود که ساختار هر یک به صورت شکل (۱۳) است.

می‌تواند اعداد بین ۱ تا ۷ متناظر با نت‌های C (دو) تا B (سی) باشد. فرکانس استاندارد پخش نوا در Matlab و یکی از استانداردهای کارت‌های صوتی و سیستم عامل‌ها ۸/۱۹۲ KHz است. براساس همین فرکانس ابتدا یک آرایه یکنواخت ساخته می‌شود. اکتاو پنجم که صدای مناسبی از لحاظ زیر و بمی ایجاد می‌کند انتخاب می‌شود. بر اساس نت ورودی مقدار n که فرکانس صوت را مشخص می‌کند مقاردهی می‌شود. در مرحله بعد باید فرکانس موج سینوسی محاسبه شود. فرکانس کلیدهای یک پیانو با ۸۸ کلاویه به شکل معادله (۲) بدست می‌آید که در آن، n شماره کلاویه پیانو است [۲۰]. فرکانس همه کلاویه‌های یک پیانو در جدول ۳ آمده است.

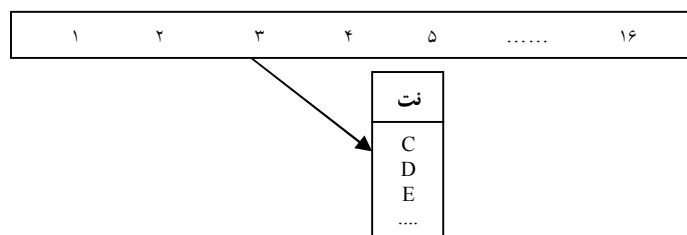
$$f(n) = 440 \times (\sqrt[12]{2})^{n-49} \quad (2)$$

هر اکتاو حاوی ۱۲ نت است و فرکانس برای اکتاو پنجم ۳۲ برابر می‌شود. یعنی $f(n) = 440 \times 2^5 \times 2^{\frac{n-49}{12}}$

بر اساس این معادله و ارتباط اکتاوهای متوالی، مقدار فرکانس در متغیر f ذخیره شده و یک موج سینوسی $(\sin 2\pi ft)$ با فواصل زمانی ۱/۸۱۹۲ از روی آن ساخته می‌شود و توسط تابع soundsc پخش می‌شود.

۳-۴ سیستم ساخت موسیقی با استفاده از الگوریتم ژنتیک

در این قسمت به توسعه سیستم خودکار آهنگ‌سازی با الگوریتم ژنتیک پرداخته می‌شود. توسعه آهنگ‌سازهای



شکل (۱۳): کروموزوم‌های مشخص‌کننده موتیف

جدول (۳): فرکانس کلایوهای پیانو

کلایو	نمایش نت	نمایش ABC	فرکانس	کلایو	نمایش نت	نمایش ABC	فرکانس
۸۸	c ^{''''} 5-line octave	C8 Eighth octave	۴۱۸۶/۰۱	۴۴	e'	E4	۳۲۹/۶۲۸
۸۷	b ^{''''}	B7	۳۹۵۱/۰۷	۴۳	d#'/e b'	D#4/E b 4	۳۱۱/۱۲۷
۸۶	a#'/b b ^{''''}	A#7/B b 7	۳۷۲۹/۳۱	۴۲	d'	D4	۲۹۳/۶۶۵
۸۵	a ^{''''}	A7	۳۵۲۰	۴۱	c#'/d b'	C#4/D b 4	۲۷۷/۱۸۳
۸۴	g#'/a b ^{''''}	G#7/A b 7	۳۳۲۲/۴۴	۴۰	c' 1-line octave	C4 Middle C	۲۶۱/۶۲۶
۸۳	g ^{''''}	G7	۳۱۳۵/۹۶	۳۹	b	B3	۲۴۶/۹۴۲
۸۲	f#'/g b ^{''''}	F#7/G b 7	۲۹۵۹/۹۶	۳۸	a#/b b	A#3/B b 3	۲۳۳/۰۸۲
۸۱	f ^{''''}	F7	۲۷۹۳/۸۳	۳۷	a	A3	۲۲۰
۸۰	e ^{''''}	E7	۲۶۳۷/۰۲	۳۶	g#/a b	G#3/A b 3	۲۰۷/۶۵۲
۷۹	d#'/e b ^{''''}	D#7/E b 7	۲۴۹۸/۰۲	۳۵	g	G3	۱۹۵/۹۹۸
۷۸	d ^{''''}	D7	۲۳۴۹/۳۲	۳۴	f#/g b	F#3/G b 3	۱۸۴/۹۹۷
۷۷	c#'/d b ^{''''}	C#7/D b 7	۲۲۱۷/۴۶	۳۳	f	F3	۱۷۴/۶۱۴
۷۶	c ^{''''} 4-line octave	C7 Double high C	۲۰۹۳	۳۲	e	E3	۱۶۴/۸۱۴
۷۵	b ^{''''}	B6	۱۹۷۵/۵۳	۳۱	d#/e b	D#3/E b 3	۱۵۵/۵۶۳
۷۴	a#'/b b ^{''''}	A#6/B b 6	۱۸۶۴/۶۶	۳۰	d	D3	۱۴۶/۸۳۲
۷۳	a ^{''''}	A6	۱۷۶۰	۲۹	c#/d b	C#3/D b 3	۱۳۸/۵۹۱
۷۲	g#'/a b ^{''''}	G#6/A b 6	۱۶۶۱/۲۲	۲۸	c small octave	C3 Low C	۱۳۰/۸۱۳
۷۱	g ^{''''}	G6	۱۵۶۷/۹۸	۲۷	B	B2	۱۲۳/۴۷۱
۷۰	f#'/g b ^{''''}	F#6/G b 6	۱۴۷۹/۹۸	۲۶	A#/B b	A#2/B b 2	۱۱۶/۵۴۱
۶۹	f ^{''''}	F6	۱۳۹۶/۹۱	۲۵	A	A2	۱۱۰
۶۸	e ^{''''}	E6	۱۳۱۸/۵۱	۲۴	G#/A b	G#2/A b 2	۱۰۳/۸۲۶
۶۷	d#'/e b ^{''''}	D#6/E b 6	۱۲۴۴/۵۱	۲۳	G	G2	۹۷/۹۹۸۹
۶۶	d ^{''''}	D6	۱۱۷۴/۶۶	۲۲	F#/G b	F#2/G b 2	۹۲/۴۹۸۶
۶۵	c#'/d b ^{''''}	C#6/D b 6	۱۱۰۸/۷۳	۲۱	F	F2	۸۷/۳۰۷۱
۶۴	c ^{''''} 3-line octave	C6 Soprano C	۱۰۴۶/۵	۲۰	E	E2	۸۲/۴۰۶۹
۶۳	b ^{''''}	B5	۹۸۷/۷۶۷	۱۹	D#/E b	D#2/E b 2	۷۷/۷۸۱۷
۶۲	a#'/b b ^{''''}	A#5/B b 5	۹۳۲/۳۲۸	۱۸	D	D2	۷۳/۴۱۶۲
۶۱	a ^{''''}	A5	۸۸۰	۱۷	C#/D b	C#2/D b 2	۶۹/۲۹۵۷
۶۰	g#'/a b ^{''''}	G#5/A b 5	۸۳۰/۶۰۹	۱۶	C great octave	C2 Deep C	۶۵/۴۰۶۴
۵۹	g ^{''''}	G5	۷۸۳/۹۹۱	۱۵	B ₁	B1	۶۱/۷۳۵۴
۵۸	f#'/g b ^{''''}	F#5/G b 5	۷۳۹/۹۸۹	۱۴	A# ₁ /B b ₁	A#1/B b 1	۵۸/۲۷۰۵
۵۷	f ^{''''}	F5	۶۹۸/۴۵۶	۱۳	A ₁	A1	۵۵
۵۶	e ^{''''}	E5	۶۵۹/۲۵۵	۱۲	G# ₁ /A b ₁	G#1/A b 1	۵۱/۹۱۳۱
۵۵	d#'/e b ^{''''}	D#5/E b 5	۶۲۲/۲۵۴	۱۱	G ₁	G1	۴۸/۹۹۹۴
۵۴	d ^{''''}	D5	۵۸۷/۳۳	۱۰	F# ₁ /G b ₁	F#1/G b 1	۴۶/۲۴۹۳
۵۳	c#'/d b ^{''''}	C#5/D b 5	۵۵۴/۳۶۵	۹	F ₁	F1	۴۳/۶۵۳۵
۵۲	c ^{''''} 2-line octave	C5 Tenor C	۵۲۳/۱۵۱	۸	E ₁	E1	۴۱/۲۰۳۴
۵۱	b ^{''''}	B4	۴۹۳/۸۸۳	۷	D# ₁ /E b ₁	D#1/E b 1	۳۸/۸۹۰۹
۵۰	a#'/b b ^{''''}	A#4/B b 4	۴۶۶/۱۶۴	۶	D ₁	D1	۳۶/۷۰۸۱
۴۹	a ^{''''}	A4 A440	۴۴۰	۵	C# ₁ /D b ₁	C#1/D b 1	۳۴/۶۴۷۸
۴۸	g#'/a b ^{''''}	G#4/A b 4	۴۱۵/۳۰۵	۴	C, contra-octave	C1 Pedal C	۳۲/۷۰۳۲
۴۷	g ^{''''}	G4	۳۹۱/۹۹۵	۳	B ₁₁	B0	۳۰/۸۶۷۷
۴۶	f#'/g b ^{''''}	F#4/G b 4	۳۶۹/۹۹۴	۲	A# ₁₁ /B b ₁₁	A#0/B b 0	۲۹/۱۳۵۲
۴۵	f ^{''''}	F4	۳۴۹/۲۲۸	۱	A ₁₁ , sub-contra	A0 Double Pedal A	۲۷/۵

در تابع ارزیابی را به خود اختصاص می‌دهد و با استفاده از این ویژگی الگوریتم ژنتیک به جستجوی آن می‌پردازد. ورودی به شکل آرایه‌ای از بیت‌ها و امتیازدهی به آن براساس معیار یافتن بیش‌ترین آکورد در آن است. کروموزومی که بیش‌ترین آکورد در آن یافت شود منفی‌ترین عدد خروجی را به خود اختصاص می‌دهد. حلقه‌های تو در تو برای تشخیص الگوهای تعریف شده آکوردهای A تا G نوشته شده است و با یافتن هر آکورد یک امتیاز به کروموزوم داده شده و آکورد یافت شده حذف می‌شود تا دوباره شمارش نشود.

مدل الگوریتم ژنتیک پیش‌تر نیز در ساخت آهنگ مورد توجه قرار گرفته بوده است [۱۰، ۱۱، ۱۳، ۲۱-۲۴]. ولی برخی از آن پژوهش‌ها به توصیف اکتفا کرده و وارد راهکار عملی نشده‌اند. برخی نیز که نتایج عملی ارائه داده‌اند، جهش را در نظر نگرفته‌اند و نیز برای بررسی برازندگی دچار مشکل گلوگاهی به دلیل نیاز به ارزیابی موسیقی تولید شده توسط انسان هستند، یا آنکه به کیفیت موتیف براساس برازندگی آکوردها توجه نکرده‌اند.

بررسی نتایج حاصل از اجرای سیستم پیشنهادی:

اجرای پیاده‌سازی‌های انجام شده در بخش ۲-۴ و ۳-۴ نشان می‌دهد که ترکیب زنجیره‌ی مارکوف و سیستم لیندن‌مایر به دلیل ماهیت روش‌های مورد استفاده، به شکل بدون تنوع به تولید موسیقی می‌پردازد و با شروع از هر نت دلخواه، قطعاتی بدون برازندگی تولید می‌کند. ولی آهنگ تولید شده توسط الگوریتم ژنتیک با استفاده از جهش، به سمت قطعات بهتر می‌رود.

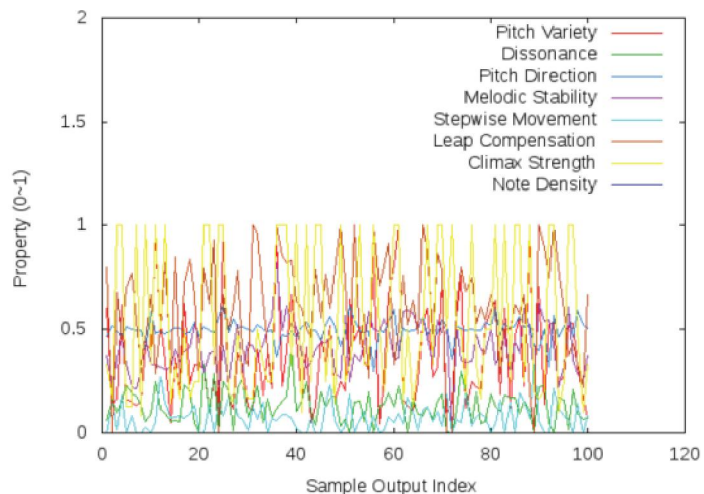
ویژگی‌های سه سیستم به طور جداگانه در پژوهش دیگری [۲۵] که به موازات این پژوهش [۲۶] در حال انجام بوده است، در شکل (۱۴) مقایسه شده است. این شکل نشان می‌دهد که موسیقی حاصل از روش لیندن‌مایر تکرار بسیار دارد و فاقد تنوع موسیقایی است. در مقابل، زنجیره مارکوف تغییرات بسیار سریعی در موسیقی ایجاد می‌کند. این عدم تنوع و تغییرات سریع باعث شده است که قطعه تولید شده فاقد موتیف‌های برازنده باشد. الگوریتم ژنتیک گاهی آهنگ‌های دلنشین‌تری ساخته است ولی همانگونه که در شکل (۱۴) مشاهده می‌شود، برازندگی موسیقی نهایی در طول ۲۵۰ نسل بهبود قابل ملاحظه‌ای نمی‌یابد.

با استفاده از روش‌هایی ساده تعدادی از این کروموزوم‌های اولیه انتخاب شده و با یکدیگر ترکیب می‌شوند. برای ترکیب آن‌ها روش‌های مختلفی وجود دارد که بر اساس نوع مسئله باید بهترین گزینه را انتخاب کرد. در نتیجه این ترکیب، نسل جدیدی شکل می‌گیرد که می‌تواند یک گام به هدف نهایی نزدیک‌تر باشد. با ادامه این روند تا رسیدن به حد مطلوب یا شرط تعداد نسل‌های تولید شده الگوریتم به پایان رسیده و خروجی را ارائه می‌دهد.

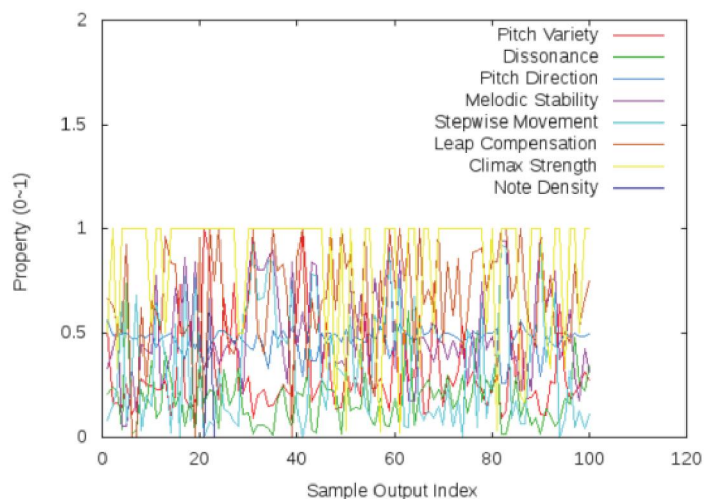
تابع بهینه‌سازی بر اساس الگوریتم ژنتیک: تابع اصلی که وظیفه پیاده‌سازی الگوریتم ژنتیک را بر عهده دارد با ۶ خروجی و ۴ ورودی نوشته شده است. برخی از این ورودی و خروجی‌ها به شکل پیش‌فرض توسط *optimtool* مربوط به الگوریتم ژنتیک در *Matlab* تولید می‌شود که در اینجا مورد نیاز نیستند. از میان خروجی‌ها، کروموزوم منتخب نهایی به همراه مقدار مینیمم نهایی تابع ارزیابی نشان‌دهنده میزان نزدیک بودن کروموزوم نهایی به مقدار ایده‌آل می‌باشد.

تنظیمات مربوط به الگوریتم ژنتیک شامل نوع متغیر جمعیت، میزان جمعیت در ابتدا، تعداد نسل تولید شونده، محدودیت تعداد نسل‌ها، تابع *Crossover* برای ترکیب کروموزوم‌ها و تولید فرزندان جدید و برخی تنظیمات دیگر مقداردهی می‌شوند. تابع اصلی مربوط به الگوریتم ژنتیک در *Matlab* با نام *ga* مشخص شده است و ورودی‌های تابع بهینه‌سازی نوشته شده را به همراه تنظیمات اعمال شده گرفته و خروجی‌های مورد نظر (از جمله مینیمم نهایی تابع ارزیابی) را برمی‌گرداند. با استفاده از این مقدار و مقایسه آن با مقدار ایده‌آل و واقعی مینیمم تابع ارزیابی می‌توان میزان موفقیت الگوریتم را نمایش داد. در پایان با استفاده از تابع نواختن آهنگ، نت‌های کروموزوم نهایی را نواخته و به شکل نمایش *ABC* آن نشان داده می‌شود.

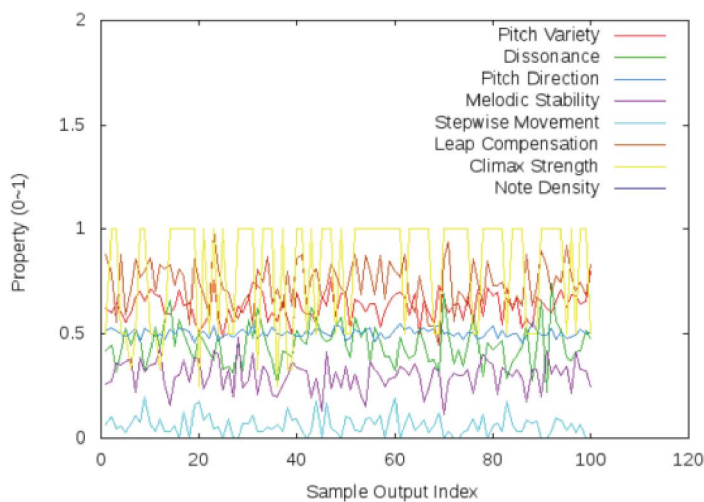
توابع ارزیابی: برای تعریف توابع ارزیابی براساس فواصل بین نت‌ها و با استفاده از مفهوم آکوردها، آکوردهای اصلی A, B, C, D, E, F و G را در نظر گرفته و در صورت وجود هریک از این آکوردها در موتیف‌ها یک امتیاز به مقدار تابع ارزیابی آن موتیف افزوده می‌شود و در نهایت، موتیفی که برازندگی بیشتری داشته باشد به عنوان خروجی برگردانده شده و نواخته می‌شود. این موتیف مقدار مینیمم



(الف)

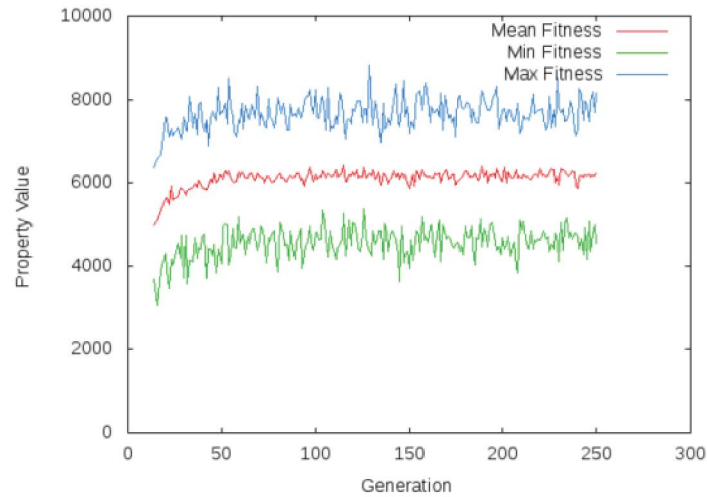


(ب)



(ج)

ساخت هوشمند موسیقی با الگوریتم ژنتیک مبتنی بر جهش یکنواخت موتیف

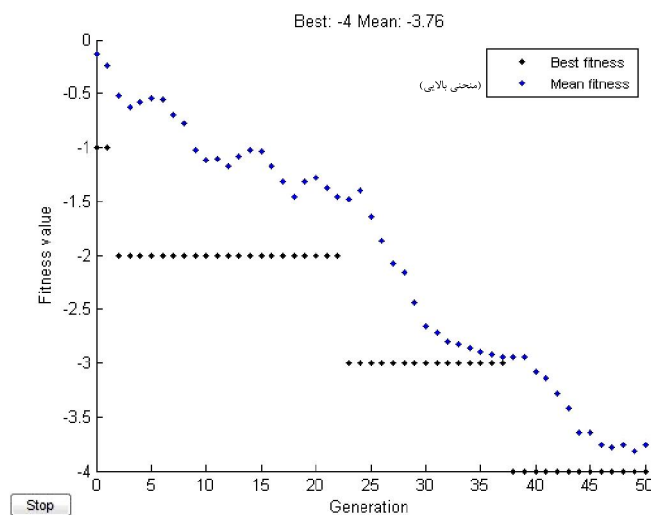


(د)

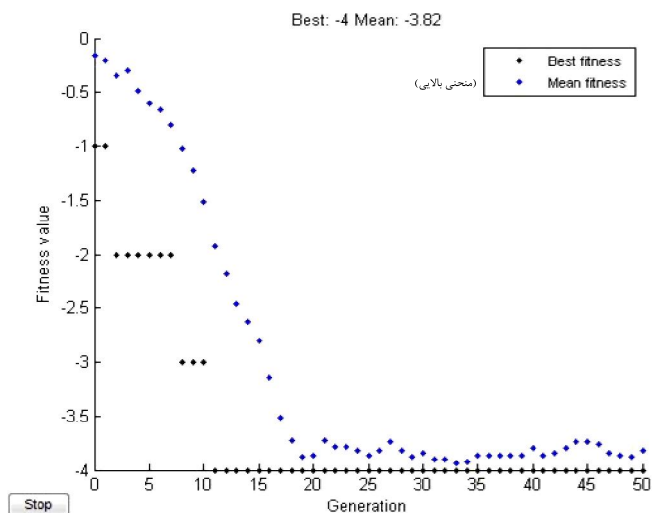
شکل (۱۴): موسیقی حاصل از اجرای (الف) گرامر لیندنمایر، (ب) زنجیر مارکوف، (ج) الگوریتم ژنتیک در ۲۵۰ نسل [۲۵]

نسل نشان می‌دهد. برای جهش به دو گونه گوسی و یکنواخت عمل شده است. مشاهده می‌شود در پیاده‌سازی که جهش به روش گوسی بوده است، اجرا در نسل ۳۸ به امتیاز ۴ (پیدا شدن ۴ آکورد در قطعه) رسیده‌است در حالی که روش یکنواخت نتیجه بهتری ارائه داده است و در نسل ۱۱ این هدف حاصل شده است.

نتایج پیاده‌سازی الگوریتم ژنتیک با به‌کارگیری جهش و تابع ارزیابی با برگزیدن قطعاتی که حاوی آکوردهای بیشتری باشند، نشان داده است که کیفیت موتیف‌های ایجاد شده در طول نسل‌ها بهبود داده می‌شود و قطعاتی برانزده تولید می‌شود. شکل (۱۵) و (۱۶)، نمودار مربوط به کیفیت موتیف‌های تولید شده بر اساس معیار بالا را در طول ۵۰



شکل (۱۵): کیفیت موتیف‌های تولید شده طی تولید نسل‌های پی‌درپی با جهش گوسی



شکل (۱۶): کیفیت موتیف‌های تولید شده طی تولید نسل‌های پی در پی با جهش یکنواخت

۵- نتیجه‌گیری و چشم‌اندازهای آینده

هدف این مقاله، ضمن مرور ادبیات این حوزه پژوهشی بین‌رشته‌ای روبه‌رشد، پیاده‌سازی عملی برخی کاربردهای هوش مصنوعی در موسیقی بود. آهنگ‌سازان در طول قرن‌ها از موتیف‌ها برای ساخت آهنگ استفاده کرده‌اند. ولی ساخت موسیقی بر پایه‌ی مجموعه‌ای از موتیف‌ها، در متون به شکل گسترده موشکافی نشده‌است. به لحاظ نت به نت بودن، تفاوت خوشایندی موسیقی ساخته شده کاملاً قابل تشخیص توسط سیستم شنوایی انسان نیست و در معرض خطا است. بنابراین، در این پژوهش، به جای استفاده از ارزیابی توسط انسان، به برازندگی موتیف‌ها مبتنی بر مشاهده آکوردها توجه شد.

در این راستا ترکیبی از سیستم تصادفی لیندن‌مایر و زنجیر مارکوف و نیز الگوریتم ژنتیک در ساخت موسیقی پیاده‌سازی شد. نتایج نشان داد که به دلیل ماهیت روش‌های مورد استفاده، سیستم اول به شکل بدون تنوع به تولید موسیقی می‌پردازد و با شروع از هر نت دلخواه، قطعاتی بدون برازندگی تولید می‌کند. ولی نتایج الگوریتم ژنتیک با استفاده از جهش و به‌کارگیری تابع ارزیابی با برگزیدن قطعاتی که حاوی آکوردهای بیشتری باشند، نشان داده است که کیفیت موتیف‌های آهنگ تولید شده در طول نسل‌ها بهبود داده می‌شود و قطعاتی برازنده تولید می‌شود. همچنین،

در پیاده‌سازی جهش به روش یکنواخت، تحول و برازندگی مجموعه موتیف‌ها سریع‌تر از روش گوسی حاصل شده است.

پژوهش دیگری می‌تواند برای ایجاد یک قطعه موسیقی متشکل از چندین موتیف، تابع برازندگی را بر روی ترکیب موتیف‌ها تعریف نماید. گرچه آهنگ‌سازی خودکار توسط عامل هوشمند که بتواند تجربه اندوزد، با استفاده از قوانین، الگوها، قطعات موسیقی موجود و استفاده از تجربه انسانی در آهنگ‌سازی می‌تواند به وقوع بپیوندد، ولی فقدان حس ناخودآگاه درونی باعث می‌شود که نتواند با قدرت موسیقایی انسان برابری کند. از این رو، ساخت موسیقی نوآورانه‌ای که برای انسان مطلوب باشد فراتر از مرزهای قابلیت هوش مصنوعی کنونی است. بنابراین، پیشنهاد می‌شود که به مفهوم دوگانگی در موسیقی نیز توجه شده و دوگانگی در مقیاس‌های متفاوت بررسی شود زیرا در نهایت آنچه که موسیقی را برای ما دلنشین می‌سازد نبرد بین کشش‌ها و رهاسازی‌ها در کل موسیقی و حتی در کوچک‌ترین اجزای آن است. احساسات در موسیقی، موضوع مهم دیگری است که در سال‌های اخیر مطرح شده [۲۷] و تولید احساس در ساخت خودکار موسیقی در ادامه این پژوهش مورد توجه قرار گرفته است.

- 1998.
- [13] Jacob, B. L., "Composing with Genetic Algorithms", Proc. of the Int. Computer Music Conf., pp. 452-455, 1995.
- [14] Delgado, M., Fajardo, W., Molinasolana, M., "Inmamusys: Intelligent Multi-Agent Music System", Expert Systems with Applications, Vol. 36, No. 3, pp. 4574-4580, 2009.
- [15] de Mantaras, R. L., Arcos, J.L., "AI and Music: from Composition to Expressive Performance", AI Magazine, Vol. 23, No. 3, p. 43, 2002.
- [16] Alpern, A., "Techniques for Algorithmic Composition of Music", Earth pp. 1-17, 1995.
- [17] Robinson, R.C., "An introduction to dynamical systems: continuous and discrete", Prentice Hall, 2004.
- [18] Lichtenberg, A.J., Lieberman. M.A., "Regular and stochastic motion." Research supported by the US Department of Energy, US Navy, and NSF. New York, Springer-Verlag (Applied Mathematical Sciences. Vol. 38, 516 p., 1983.
- [19] May, R.M., "Simple mathematical models with very complicated dynamics." Nature, Vol. 261, No. 5560, pp. 459-467, 1976.
- [20] McClellan, J. H., Schafer R., Yoder. M., "DSP First: A Multimedia Approach". Prentice Hall, USA. 1998.
- [21] Biles, J. A., "GenJam: A Genetic Algorithm for Genetaring Jazz Solos", Architecture, pp. 131-131, 1994.
- [22] Matic, D., "A Genetic Algorithm for Composing Music", Yugoslav Journal of Operations Research, Vol. 20, No. 1, pp. 157-177, 2010.
- [23] Burton, A.R. Vladimirova, T.R., "Generation of Musical Sequences with Genetic Techniques", Computer Music Journal, Vol. 23, No. 4, pp. 59-73, 1999.
- [24] Nelson, G. L., "Sonomorphs: An Application of Genetic Algorithms to the Growth and Development of Musical Organisms", Proc. of the Fourth Biennial Art Tech. Symposium, pp. 1-15, 1993.
- [25] Yamaguchi, M.H., "An extensible tool for automated music generation", B.Sc. Thesis, Department of Computer Science, Lafayette College, May 2011.
- [26] Roghanizedeh, R., "Artificial Intelligence in Music Composition", B.Sc. Thesis, Department of Information Technology Engineering, University of Isfahan, June 2011.
- [27] Yang, Y.-H., Chen, H.H., Music Emotion Recognition, CRC Press, 2011.
- مراجع
- [1] Nierhaus, G., Algorithmic composition: paradigms of automated music generation. Springer, 2010.
- [2] Julià, J.S., "Identification of Versions of the Same Musical Composition by Processing Audio Descriptions", PhD Thesis, Dept. of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain, 2011.
- [3] Lashari, S.A., Ibrahim R., Senan, N., "Performance Comparison of Musical Instrument Family Classification Using Soft Set", International Journal of Artificial Intelligence and Expert Systems (IJAE), Vol. 3, No. 4, pp. 100-110, 2012.
- [4] Lo, M.Y., "Evolving Cellular Automata for Music Composition with Trainable Fitness Functions", PhD Thesis, School of Computer Science and Electronic Engineering, University of Essex, 2012.
- [5] McCormack, J., "Grammar Based Music Composition", Complex systems, Vol. 96, pp. 321-336, 1996.
- [6] Mansouri, P., Fundamental Theory of Music. 3rd Ed., Karnameh Publishing, 2006.
- [7] Harnum, J., "Basic Music theory, how to read, write, and understand written music", Sol-Ut Press, 2001.
- [8] Papadopoulos G., Wiggins, G., "AI Methods for Algorithmic Composition: A Survey, a Critical view and Future Prospects", AISB Symposium on Musical Creativity, Edinburgh, UK, 1999.
- [9] Sugimoto, T., Legaspi, R., Ota, A., Moriyama, K., Kurihara, S., Numao, M., "Modelling Affective-Based Music Compositional Intelligence with the Aid of ANS Analyses", Knowledge-Based Systems Journal, Vol. 21, No. 3, pp. 200-208, 2008.
- [10] Khalifa, Y.M.A., Al-Mourad, M.B., "Autonomous Evolutionary Music Composer", Genetic and Evolutionary Computation Conf. - GECCO, pp. 1873-1874, 2006.
- [11] Pazos, A., del Riego, A.S., Dorado, J., Caldalda, J.J.R., "Genetic Music Compositor". Proc. of the Congress of Evolutionary Computation, Vol. 2, 1999.
- [12] Johanson, B., Poli R., "GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters", Genetic Prog. Proc. of the Third Annual Conf. CSRP, Vol. 98, No. 13, pp. 181-186,

¹ The American Standard Code for Information Interchange

² Lindenmayer