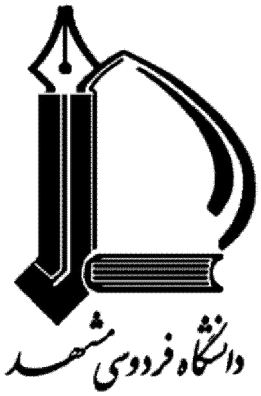
بسم الله الرحمن الرحيم



دانشکده مهندسی برق

رساله کارشناسی ارشد برق- کنترل

کنترل پارامترهای استراتژی الگوریتم های تکاملی در یک محیط غیر ایستا با استفاده از کنترلر فازی احتمالی تطبیقی

اساتید راهنما :

دکتر محمد رضا اکبرزاده توتونچی

دکتر محمد باقر نقیبی سیستانی

تهیه و تنظیم :

مجید مازوچی

تابستان ۱۳۸۹

خدایا به من زیستنی عطا کن که در لحظه مرگ، بر بی ثمری لحظه هایی که برای زیستن گذاشته ام حسرت نخورم...

(دکتر شریعتی)

تقدیم به پدر و مادر بزرگوارم که گام برداشتن در فراز و نشیب زندگی را به من آموختند و روح، فکر و ایمانم را پرورش دادند.

تشکر و قدردانی

حال که به یاری خداوند متعال توفیق تکمیل این پایان نامه نصیبم گردید بر خود لازم می دانم که از زحمات، تلاشهاو راهنمائیهای دلسوزانه اساتید بزرگوارم آقایان دکتر اکبرزاده توتونچی و دکتر نقیبی سیستانی که در طول انجام این پروژه راهنما و مشوق من بودند، کمال تشکر را نموده و امتنان قلبی خود را از بابت تمامی زحمات ایشان بیان می نمایم.

همچنین از تمامی اساتید محترمی که در دوران تحصیل از محضر علمی ایشان بهره برده ام، قدر دانی می نمایم و سلامتی انان را از خداوند متعال خواستارم.

در دوره تحصیل در بخش تحصیلات تکمیلی کنترل، از امکانات آزمایشگاه محاسبات شناختی بهره ی بسیار برده ام؛ بی شک، این امکانات تسریع کننده ی تلاش من در این دوره بوده است، سپاس بر گردانندگان این آزمایشگاه.

مصاحبت با دوستان و هم دوره ای های عزیزم، القا کننده ی دلگرمی و توان مضاعف در من بوده است. لطف و مرحمت الهی بر همه ی آنان، بویژه: آقایان روحانی منش، ولی زاده؛ خانم ها تاتاری و سید حجازی.

چکیده: در تمام الگوریتم های بهینه سازی مبتنی بر جمعیت، مسئله ی برقراری تعادل بین اکتشاف و استفاده از نتایج بدست آمده، مهمترین مسئله ای است که الگوریتم با آن رو به رو است. تا کنون الگوریتم های ژنتیکی تطبیقی بسیاری که بصورت دینامیک به کنترل بعضی پارامترهای استراتژی الگوریتم می پردازند، ایجاد و معرفی شده اند. هدف اصلی آنان بهبود رفتار الگوریتم مزبور، در ایجاد تعادل بین اکتشاف و بهره وری بوده که در نتیجه منجر به بهبود عملکرد الگوریتم می گردد. مسئله ی MAB بطور گسترده برای بررسی کردن مسئله ی ایجاد تعادل بین اکتشاف و بهره وری مورد استفاده قرار می گیرد. چالش انگیز ترین انواع MAB، انواع غیر ایستای آنان است که عامل در آن با افزایش پیچیدگی در شناسایی تغییرات در محیط نیز روبروست. در این رساله، با یک مسئله ی MAB غیر ایستا، زمان گسسته، با افق زمانی محدود، تعداد اکشن های محدود و پاداشی تصادفی با توزیع نرمال مواجه هستیم. در ادامه، برای حل این مسئله یک الگوریتم ژنیک کد حقیقی بر اساس کنترلر فازی احتمالی تطبیقی معرفی و مورد استفاده قرار می گیرد. با استفاده از نتایج شبیه سازی شده ی بدست آمده توسط رهیافت های ارائه شده در این رساله، این نتیجه ی مهم حاصل می گردد که استفاده از کنترلر های فازی احتمالی تطبیقی برای کنترل پارامترهای الگوریتم های ژنتیک، بنا بر بررسی های آماری باعث بهبود در ایجاد تعادل بین بهره وری و اکتشاف شده و در نتیجه عملکرد الگوریتم مزبور را بهبود می بخشد.

واژگان کلیدی: الگوریتم های تکاملی، کنترلر فازی احتمالی، تعادل بین بهره وری و اکتشاف، الگوریتم های فازی تکاملی، کنترل پارامترهای استراتژی، محیط غیر ایستا، Multi-armed Bandit

فهرست عنوان ها

فصل اول: مقدمات ۶

مقدمه ۶

طبقه بندی تکنیکهای کنترل ۱۰

کنترل پارامترها بصورت قطعی ۱۱

کنترل پارامترها بصورت تطبیقی ۱۱

کنترل پارامترها بصورت خود- تطبیقی ۱۱

ملاک مطلق ۱۲

ملاک نسبی ۱۲

کارهای انجام شده دراین پایان نامه ۱۵

فصل دوم: مروری بر الگوریتمهای مبتنی بر جمعیت ۱۷

مقدمه ۱۷

الگوریتمهای بهینه سازی مبتنی بر جمعیت ۱۸

روشهای کلاسیک مبتنی بر جمعیت ۲۲

بهینه سازی تکاملی ۲۳

روش جستجوی سیمپلکس ۲۴

روشهای جستجوی تصادفی تطبیقی ۲۵

روشهای تکاملی ۲۶

الگوریتم محاسبات تکاملی، بطور عام ۲۷

فرم نمایش جواب ۲۸
جمعیت اولیه ۳۰
اپراتورهای الگوریتمهای تکاملی ۳۰
محاسبه ی ارزش تابع هدف ۳۰
اپراتور انتخاب ۳۱
اعمال دگرگونی در جمعیت جواب ۳۲
اپراتور تولید مجدد ۳۳
اپراتورهای تولید مجدد پارامتر حقیقی ۳۴
اپراتور جهش ۴۰
الگوریتم ژنتیک پارامتر حقیقی ۴۴
الگوریتمهای تکاملی گسترده در مقابل حالت ماندگار ۴۵
تکامل تفاضلی ۴۸
استراتژی تکاملی ۴۹
استراتژیهای تکاملی- وفقی ۵۰
فصل سوم: مروری بر روشهای بهبود الگوریتمهای تکاملی ۵۲
مقدمه ۵۲
رفتارهای الگوریتمهای تکاملی ۵۲
روشهای بهبود عملکرد الگوریتمهای تکاملی ۵۴
الگوریتمهای تکاملی با چندین جمعیت ۵۶
الگوریتمهای تکاملی و پردازش موازی ۵۶
رهیافت چند جمعیتی ۵۸

تجزیه ی مسئله ۵۹
اصلاح اپراتورها در الگوریتمهای تکاملی ۵۹
الگوریتمهای هایبرید ۶۱
منطق فازی و الگوریتمهای تکاملی ۶۲
شبکه ی عصبی و الگوریتمهای تکاملی ۶۵
روشهای محاسبات سخت و الگوریتمهای تکاملی ۶۶
سن در الگوریتمهای تکاملی ۶۷
ارث بری در الگوریتمهای تکاملی ۶۸
فصل چهارم: مروری بر کنترل پارامترها یا لگوریتمهای تکاملی ۷۰
مقدمه ۷۰
تنظیم پارامترها یا لگوریتمهای تکاملی ۷۱
طبقه بندی تکنیکهای کنترل ۷۵
کنترل پارامترها بصورت قطعی ۷۶
کنترل پارامترها بصورت تطبیقی ۷۶
کنترل پارامترها بصورت خود- تطبیقی ۷۶
ملاک مطلق ۷۷
ملاک نسبی ۷۸
فصل پنجم: الگوریتمهای تکاملی فازی ۸۰
مقدمه ۸۰
منطق فازی ۸۰
الگوریتمهای تکاملی– فازی ۸۲

تعریف ورودی هاو خروجی ها ۸۳
مقادیر تنوع در محیط ژینوتایپ..... ۸۴
GDMs برپایه ی فاصله ی اقلیدسی ۸۴
GDMs برپایه ی مقادیر پخششدگی ۸۵
مقادیر تنوع در محیط فینوتایپ ۸۵
تعریف پایگاه داده..... ۸۵
تعریف پایگاه قواعد ۸۶
فصل ششم: فازی احتمالی و یادگیری تقویتی ۹۰
مقدمه..... ۹۰
منطق فازی احتمالی ۹۰
مجموعه فاز ی احتمالی..... ۹۱
قوانین فازی احتمالی..... ۹۲
یادگیری تقویتی ۹۴
الگوریتم QL..... ۹۴
الگوریتم $\varepsilon -$ ۹۶
فصل هفتم: کنترلر فازی احتمالی تطبیقی (APFC) ۹۷
مقدمه..... ۹۷
کنترلر فازی احتمالی تطبیقی در یک نگاه..... ۹۸
فصل هشتم.....۱۰۱
بکارگیری (APFC) برای ایجاد توازن و بهبود عملکرد الگوریتم ژنتیک در مسئله (Multi -Armed Bandit) در یک محیط غیر ایستا.....۱۰۱
مقدمه.....۱۰۱

بهینه سازی مسئله ی Multi-Armed Bandit ۱۰۱
الگوریتم ژنتیک ۱۰۳
امتزاج ۱۰۵
جهش ۱۰۵
اپراتورRepair ۱۰۵
اپراتور نخبه گرایی ۱۰۵
شبیه سازی ۱۰۶
نتایج شبیه سازی الگوریتم تکاملی (T-GA) ۱۰۶
الگوریتم ژنتیک هایبرید با کنترلر فازی (FLC-GA) ۱۰۸
تعریف ورودی ها و خروجی ها ۱۱۰
تعریف پایگاه داده ۱۱۱
تعریف توابع عضویت ۱۱۲
تعریف پایگاه قواعد ۱۱۷
شبیه سازی ۱۱۸
نتایج شبیه سازی الگوریتم تکاملی هایبرید با کنترلر فازی (FLC-GA) ۱۱۹
الگوریتم ژنتیک هایبرید با کنترلر فازی احتمالی (PF-GA) ۱۲۳
شبیه سازی ۱۲۵
نتایج شبیه سازی الگوریتم تکاملی هایبرید با کنترلر فازی احتمالی (PF-GA) ۱۲۵
الگوریتم ژنتیک هایبرید با کنترلر فازی احتمالی تطبیقی (APFC-GA) ۱۲۸
شبیه سازی ۱۳۰
نتایج شبیه سازی الگوریتم تکاملی هایبرید با کنترلر فازی احتمالی تطبیقی (APFC- GA) ۱۳۰

فصل نهم: نتایج و پیشنهادات ۱۳۳

نتیجه گیری ۱۳۳

پیشنهادات ۱۳۵

مراجع ۱۳۶

فصل اول

مقدمات

مقدمه :

بهینه سازی عبارت است از تنظیم ورودی های یک تابع ریاضی، یا یک آزمایش بطوری که خروجی یا نتیجه ی آن بیشینه یا کمینه شود. در مسائل بهینه سازی، ورودی شامل تعدادی پارامتر است. از فرآیند یا تابع به عنوان تابع هزینه یا تابع برازندگی[1] نام برده می شود و خروجی نیز هزینه یا برازش نام دارد. اکثر روشهای بهینه سازی با جستجو در فضای هزینه[2] به یافتن پاسخ بهینه می پردازند.

در بکارگیری روشهای کلاسیک بهینه سازی با رهیافت نقطه به نقطه[3] ، اغلب با مشکلات زیادی روبرو می شویم. در بهینه سازی تحلیلی با رهیافت نقطه به نقطه که با یک جواب حدسی شروع شده و جواب ها بصورت مرحله به مرحله به روز می شوند تا نهایتا به نزدیکی یک نقطه ی بهینه برسیم، برای نمونه در روش لاگرانژین[4]، با معادل صفر قرار دادن گرادیان یک تابع ($\nabla = 0$) و محاسبه ی مشتق دوم آن، نقاط کمینه یا بیشینه ی آن تابع را تعیین می کنیم. یکی از معایب روش بهینه سازی با این رهیافت نقطه به نقطه، معین نبودن این مسئله است، که کدام نقطه بهینه سرتاسری است و کدام محلی؛ مشکل دیگر این رهیافت این است که فقط در بهینه سازی توابع با پارامتر های پیوسته به کار می رود، که با گسسته شدن و افزایش میزان پیچیدگی و تعداد پارامتر ها این روش کارایی خود را از دست می دهد.

اکنون این سوال مطرح می شود که با توجه به عیوب روش های کلاسیک، آیا روش مطمئنی برای بدست آوردن پاسخ بهینه ی سرتاسری برای یک مسئله در حالتی که تعداد پارامتر ها و پیچیدگی مسئله بسیار زیاد بوده و از طرف دیگر زمان در دست محدود باشد، وجود دارد؟ خوشبختانه پاسخ این سوال مثبت است و در طول چهار دهه ی اخیر، رهیافت های بهینه سازی چند نقطه ای[5] که به تعدادی از متدهای کلاسیک و متدهای بهینه

---

1 Fitness function
2 تمام مقادیر ممکنی که تابع هزینه می تواند اختیار کند
3 Point-by-Point
4 Lagrangian method
5 Multi Point

سازی مبتنی بر جمعیت[۶] الهام گرفته شده از طبیعت، همانند الگوریتم های تکاملی ، الگوریتم اجتماع پرندگان ، اجتماع مورچگان تقسیم می شود، نتایج بسیار خوبی را در این زمینه به دست داده اند. برای نمونه، استفاده از الگوریتم های تکاملی در دو دهه ی اخیر بطور چشمگیری افزایش پیدا کرده است. این الگوریتم ها، با داشتن قابلیت حل مسائل بهینه سازی در حالت سرتاسری، قابلیت خود را در بهینه سازی مسائل ناپیوسته، مشتق ناپذیر، نویزی ، چند هدفی و همچنین مسایلی مانند ارزیابی موسیقی که دارای تابع برازندگی خاصی برای ارزیابی نیست، به اثبات رسانیدند.

اما در کنار این قابلیت ها، الگوریتم های مبتنی بر جمعیت و بخصوص الگوریتم های ژنتیک که مورد بحث این پروژه است، از مشکلات حادی، نظیر همگرایی زودرس[۷] یا رکود[۸] و مسئله ی قابلیت اطمینان[۹] رنج می برند [Bac96]؛ که از رسیدن الگوریتم به یک جواب رضایت بخش در یک زمان معقول و منطقی ممانعت می کند که به دلیل نبود تعادل بین جستجو و بهره برداری به وجود می آیند، که با تنظیم پارامترها ی این الگوریتم ها می توان بر این مشکلات فائق آمد.

بنابر این، مسئله ی تنظیم پارامتر های الگوریتم های تکاملی[۱۰] یک امر بسیار مهم برای داشتن عملکرد خوب در این گونه الگوریتم ها است. در واقع، پیدا کردن یک ترکیب خوب از پارامترهای موجود در اپراتورهای الگوریتم های تکاملی، مسئله ای است که از مدت ها قبل بصورت یک چالش برای این گونه الگوریتم ها وجود داشته است. اما از طرف دیگر، مسئله ی اصلی که وجود دارد، این است که هر الگوریتم تکاملی خاص، شامل اجزاء و مولفه های خاص خود، از قبیل: اپراتورهای نمایش[۱۱]، انتخاب[۱۲]، تولید مجدد[۱۳] ، جهش[۱۴] و ... است؛ در نتیجه، چارچوب الگوریتم دارای پارامترهای نامشخص بسیاری است.

برای نمونه، یک الگوریتم ژنتیک ساده ممکن است با استفاده از نمایش باینری، امتزاج یکنواخت[۱۵]، جهش بیت-فلیپ، انتخاب مسابقه ای[۱۶] و برای مثال بصورت جایگذاری نسل به نسل[۱۷] تعریف شده باشد. که به هر حال برای تعریف دقیق تر، می بایست اطلاعات و جزئیات بیشتری مشخص و بیان شود؛ برای مثال، اندازه ی جمعیت، نرخ جهش، نرخ امتزاج و اندازه ی مسابقه، که به این گونه اطلاعات و جزئیات، پارامترهای الگوریتم یا پارامتر های استراتژی[۱۸] گفته می شود. بنا بر این، بدست آوردن مقادیر مناسب پارامترهای استراتژی، برای داشتن یک

6Population based optimization
7Premature Convergence
8Stagnation
9Reliability
10Evolutionary Algorithms
11Representation
12Selection
13Reproduction
14Mutation
15Uniform Crossover
16Tournament
17Generational Replacement
18Strategy Parameters

الگوریتم کارآمد و قابل اجرا، امری ضروری است. در نتیجه، مقادیر این پارامترها باید با دقت انتخاب شوند تا الگوریتم قادر به پیدا کردن جواب بهینه، نزدیک به بهینه[۱۹] و یا جوابی قانع کننده برای کاربر[۲۰] باشد؛ که به هر حال، بدست آوردن مقادیر درست و مناسب برای پارامتر های استراتژی امری بسیار دشوار و مشکل است.

بطور کلی دو فرم اصلی برای تنظیم کردن مقادیر پارامترها[۲۱] وجود دارد: کنترل پارامترها[۲۲] و کوک کردن پارامترها[۲۳]. از یک طرف، کوک کردن پارامترها، یعنی بهترین مقادیر را پیش از اجرای برنامه انتخاب کرده، که این مقادیر در حین اجرای برنامه ثابت می مانند؛ در ادامه نشان خواهیم داد که انتخاب و قرار دادن پارامترهای استراتژی از پیش و بصورت استاتیک و ثابت، امری نادرست است. اما از طرف دیگر، کنترل پارامترها که جایگزین مناسبی برای کوک کردن پارامترهاست، یعنی انتخاب یک مقادیر ابتدایی و تا حد توان مناسب برای پارامترهای استراتژی پیش از اجرای الگوریتم، که این مقادیر در حین اجرای الگوریتم با مکانیزم خاص و از پیش تعریف شده ای، به منظور بهبود عملکرد الگوریتم، تغییر نموده و بهبود داده می شوند.

کوک کردن پارامترها یک رهیافت ابتدایی و معمولی برای استفاده در الگوریتم های تکاملی است. در واقع، کوک کردن[۲۴] مقادیر پارامتر های استراتژی یک الگوریتم تکاملی، بر اساس تجربه و سعی و خطا برای مقادیر مختلف پارامترها در الگوریتم مزبور و سپس انتخاب بهترین ترکیب از این مقادیر، بر اساس این امر که کدام ترکیب نتیجه ی بهتری در عملکرد الگوریتم برای مسئله ی در دست ارائه می دهد، انجام می شود. به هر حال بدیهی است که، این رهیافت که قسمت عمده ی آن مبتنی بر سعی و خطا است، بسیار وقت گیر و زمان بر است. تصور کنید که الگوریتم تکاملی با پنج پارامتر استراتژی و چهار مقدار پیشنهادی و کاندید برای هر کدام داریم، در نتیجه باید ۶۲۵ ترکیب مختلف و برای هر ترکیب حداقل ۱۰۰ اجرای مجزا انجام دهیم، که به عبارتی ۶۲۵۰۰ اجرا برای انتخاب پارامترهای استراتژی مناسب[۲۵] برای الگوریتم مزبور نیاز است.

بطور خلاصه، مشکلات و نقاط ضعف تکنیکی، در رهیافت کوک کردن پارامترها بر اساس سعی و خطا و تجربه بصورت زیر است:

۱. بطور کلی، پارامترها از یکدیگر مجزا نیستند و سعی در امتحان تمامی ترکیب های مختلف بصورت سیستماتیک و از روی یک روش معین، از لحاظ عملی غیر ممکن است.
۲. پروسه ی کوک کردن پارامتر ها، فرآیندی زمان بر است، حتی اگر پارامترها مجزا و مستقل از هم و بی توجه به ارتباطات و وابستگی هایشان به یکدیگر، بهینه شوند.

19Sub optimal
20User
21Setting Parameters
22Parameters Control
23Parameters Tuning
24Tuning
25که هیچ ضمانتی برای مناسب بودن وجود ندارد

۳. برای مسئله ی داده شده، مقادیر بدست آمده توسط رهیافت کوک کردن پارامترها، حتی اگر تلاش بسیاری برای بدست آوردن آن ها انجام گرفته باشد، هیچ ضمانتی وجود ندارد که لزوما بهینه و بهترین مقادیر موجود باشند.
۴. بعضی قضایای تئوریک، مانند تئوری NFL[۲۶].

به هر حال باید توجه داشت که فرآیند تکامل در الگوریتم تکاملی بطور ذاتی و غریزی، فرآیندی دینامیک[۲۷] است، که در نتیجه استفاده از پارامترهای ثابت که مقادیر آن ها در حین فرآیند تغییر نکرده و ثابت است با ذات این مسئله در تضاد است. از طرف دیگر، از لحاظ تئوری و عملی، این امر که مقادیر مختلف برای پارامتر های استراتژی برای وضعیت ها و شرایط متفاوت یک پروسه ی تکاملی مناسب اند، نشان داده شده است [Jai00] ، [Eib07]. برای نمونه، نرخ جهش بالا در نسل های اولیه مناسب بوده و به جستجوی[۲۸] بهتر در فضای مسئله کمک کرده و نرخ جهش کوچک در نسل های انتهایی به همگرایی به یک کروموزوم زیربهینه[۲۹] کمک می کند، که خود گواهی بر این مسئله است که استفاده از پارامترهای استراتژی بصورت استاتیک می تواند منجر به عملکرد ضعیف الگوریتم تکاملی مزبور گردد.

یک راه واضح برای غلبه بر این نقاط ضعف و محدودیت ها استفاده از تابع ( ) به جای پارامتر ثابت است، که شماره ی نسل و یا هر مقدار مربوط به زمان سپری شده، است. به هر حال، فرایند پیدا کردن پارامتر های دینامیک بهینه ی ( ) حتی ممکن است از فرایند پیدا کردن یک مقدار پارامتر بهینه ی استاتیک نیز سخت تر باشد.

رهیافت دیگر، بر پایه ی این نظر که پیدا کردن مقادیر مناسب برای پارامترهای استراتژی یک الگوریتم تکاملی مسئله ای پیچیده و مشکل است، معرفی شده است؛ در واقع، این مسئله به دلیل پیچیدگی و مبهم بودن از آن گونه مسائلی است که الگوریتم های تکاملی نسبت به باقی روش های بهینه سازی موجود، در حل آن، از خود عملکرد بهتری نشان می دهند. این امر، دقیقا همان ایده ی استفاده از یک الگوریتم تکاملی، برای تنظیم پارامترهای استراتژی موجود در الگوریتم تکاملی دیگر به منظور حل مسئله ی در دست است؛ در واقع، در این رهیافت از دو الگوریتم تکاملی بطور هم زمان استفاده می شود، که اولی برای حل مسئله ی در دست و دومی که به آن متا الگوریتم تکاملی[۳۰] گفته می شود، به منظور تنظیم پارامترهای استراتژی الگوریتم تکاملی اول، است [Eib08]. هر چند این رهیافت، به این گونه نیز به کار می رود که بطور کلی فقط از یک الگوریتم تکاملی استفاده کرده و خود الگوریتم وظیفه ی اصلاح و تعدیل پارامتر های استراتژی خود را دارد.

26No Free Lunch
27Dynamic
28Exploration
29Suboptimal
30Meta EA

در تمام الگوریتم های تکاملی و به طور کلی، الگوریتم های بهینه سازی مبتنی بر جمعیت، مسئله ی برقراری تعادل بین اکتشاف[۳۱] و استفاده از نتایج بدست آمده[۳۲]، مهمترین مسئله ای است که الگوریتم با آن رو به رو است[Gol89]. در واقع جمله ی "استفاده از نتایج بدست آمده " به این موضوع بر می گردد، که چه مقدار از جواب های خوب بدست آمده در تکرار های متوالی استفاده کنیم و دومین جمله "اکتشاف" به این موضوع بر می گردد که چه مقدار به دنبال جواب های بهتر دیگری در فضای جستجو بگردیم. تعدادی از مطالعات در این زمینه[Gol93]، [Thi93]روش هایی برای تعیین کردن این دو مقدار در الگوریتم های ژنتیک ارائه دادند، و بطور واضح نشان دادند که برقراری تعادل بین این دو کمیت نیازی ضروری است.

حال در ادامه به طبقه بندی تکنیک های موجود برای کنترل پارامترهای الگوریتم های تکاملی می پردازیم.

طبقه بندی تکنیک های کنترل

برای طبقه بندی[۳۳] تکنیک ها ی کنترل پارامتر های استراتژی در الگوریتم های تکاملی، باید چندین نکته و منظر را در نظر گرفت.

۱. چه اجزا یا پارامترهایی قرار است تغییر کنند؟ (برای مثال: تابع ارزیابی، نرخ جهش، نرخ امتزاج، پارامترهای اپراتورها)
۲. این تغییر، چگونه قرار است اتفاق بیافتد؟ (به عبارت دیگر، بصورت قطعی[۳۴]، بر پایه ی فیدبک[۳۵] و بصورت وفقی یا خود- وفقی[۳۶])
۳. بر اساس چه فیدبک یا ملاکی[۳۷] قرار است، تغییرات اتفاق بیافتد؟ (برای مثال، تنوع[۳۸] موجود در جمعیت، عملکرد[۳۹] اپراتورها و ...)

حال به بحث در مورد نکات مطرح شده می پردازیم.

برای طبقه بندی تکنیک های کنترل پارامتر از منظر "چه اجزا و پارامترهایی قرار است تغییر کند؟" ضروری است که در مورد لیست تمام اجزا اصلی الگوریتم های تکاملی به توافقی دست پیدا کنیم؛ برای این منظور، اجازه دهید اجزا نام برده شده در زیر را به عنوان اجزا اصلی الگوریتم های تکاملی در نظر بگیریم:

31Exploration
32Exploitation
33Classifying
34Deterministic
35Feedback Based Heuristic
36Self Adaptive
37Evidence
38Diversity
39Performance

- اپراتورهای دگرگونی و تابع توزیع احتمالی آن ها
- جمعیت (اندازه، توپولوژی، ...)
- نحوه ی نمایش افراد

توجه داشته باشید که هر جز می تواند بصورت پارامتری باشد، که تعداد پارامترها بطور واضح مشخص نیست؛ برای مثال، پارامترهای جمعیت در هنگامی که بیش از یک جمعیت در الگوریتم تکاملی وجود دارد، می تواند شامل سایز زیر جمعیت ها[۴۰]، نرخ مهاجرت[۴۱] و غیره باشد.

اما از منظر "این تغییرات، چگونه قرار است اتفاق بیافتد؟ "، می توان روش های موجود برای تغییر پارامتر ها را در سه رده طبقه بندی کرد :

کنترل پارامترها بصورت قطعی[۴۲]

این روش به این گونه است که، مقدار پارامترهای استراتژی توسط یک قانون قطعی تغییر می کنند. این قانون، پارامترهای استراتژی را با یک روش ثابت و از پیش تعیین شده ای (که توسط کاربر مشخص شده است.) بدون استفاده از هیچ گونه فیدبکی از فضای جستجو، اصلاح می کند.

کنترل پارامترها بصورت تطبیقی[۴۳]

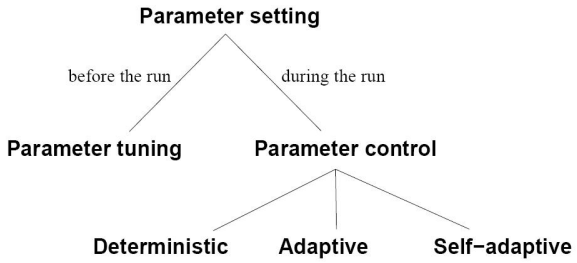
این روش به این گونه است که، تعدادی فیدبک از فضای جستجو به عنوان ورودی به مکانیزمی که وظیفه ی تغییر پارامترهای استراتژی را دارد، وارد می شود. تفاوت اصلی این روش با روش قبل، در استفاده ی مکانیزم به روز رسانی از یک فیدبک خارجی است.

کنترل پارامترها بصورت خود- تطبیقی[۴۴]

در این حالت، پارامترهای استراتژی که قرار است تعدیل شوند، به درون کروموزوم ها کد شده و تحت تاثیر مستقیم اپراتورهای دگرگونی و ماژول های دیگر قرار می گیرند؛ که بهترین مقادیر این پارامترهای کد شده، منجر به ایجاد افراد بهتر شده که گرایش بیشتری برای باقی ماندن در جمعیت و تولید فرزندان جدید دارند.

شکل (۲) بطور خلاصه طبقه بندی مزبور را نشان می دهد.

40Sub-Population
41Migration
42Deterministic Parameter Control
43Adaptive Parameter Control
44Self- Adaptive Parameter Control



شکل (۱-۱) - طبقه بندی کلی، رهیافت های موجود تنظیم پارامتر های استراتژی در الگوریتم های تکاملی

اما سومین منظر که بر اساس آن، رهیافت های کنترلی طبقه بندی می شوند، این است که " بر اساس چه فیدبک یا ملاکی[۴۵] قرار است، تغییرات اتفاق بیافتد؟ " [Smi98] . بطور کلی، رهیافت های کنترلی، برای رسیدن به عملکرد بهتر در الگوریتم های تکاملی، سعی در مانیتور کردن وضعیت پروسه ی جستجو، از قبیل تنوع، عملکرد اپراتور ها، کیفیت و غیره، دارند. اطلاعاتی که از چنین پروسه ی مانیتورینگی بدست می آید، به عنوان فیدبک برای تعدیل و اصلاح پارامترهای استراتژی بکار برده می شود، که بطور کلی دو نوع ملاک یا فیدبک در پروسه ی مانیتورینگ مورد استفاده قرار می گیرد.

ملاک مطلق[۴۶]

ما هنگامی در مورد ملاک مطلق بحث می کنیم که، مقادیر پارامترهای استراتژی بوسیله ی یک و یا تعدادی قاعده و قانون که از پیش مشخص و تعریف شده اند، اصلاح و تعدیل شوند؛ و البته تفاوت این روش با کنترل پارامتر قطعی در این نکته نهفته است که در کنترل پارامتر بصورت قطعی، قواعد بر اساس یک راه انداز قطعی[۴۷] (برای مثال زمان سپری شده) اعمال می شوند، که در واقع این راه انداز قطعی یک فیدبک واقعی که دارای اطلاعات با ارزشی از وضعیت فرایند جستجو باشد، نیست؛ در حالی که منظور از فیدبک یا ملاک مطلق، یک فیدبک از فضای جستجو با اطلاعات با ارزش از وضعیت جستجو است.

ملاک نسبی[۴۸]

در حالت استفاده از ملاک نسبی، پارامترهای استراتژی متفاوت بر اساس برازندگی فرزندانی که بر پایه ی آن ها تولید می شوند، مقایسه شده و بهترین پاداش بیشتری کسب می کند. مسیر و دامنه ی تغییرات پارامترهای استراتژی بطور قطعی مشخص نبوده، بلکه به عملکرد بقیه ی پارامترها وابسته است؛ البته واضح و بدیهی است که، در هر لحظه لازم است که بیش از یک مقدار برای هر پارامتر استراتژی موجود باشد.

45Evidence
46Absolute Evidence
47Deterministic Trigger
48Relative Evidence

بطور خلاصه می توان گفت، سه ملاک اصلی برای طبقه بندی روش های موجود برای تعدیل پارامترهای استراتژی الگوریتم های تکاملی در حین اجرا، به قرار زیر هستند:

۱. چه اجزا یا پارامترهایی قرار است تغییر کنند؟
۲. این تغییر، چگونه قرار است اتفاق بیافتد؟
۳. بر اساس چه فیدبک یا ملاکی[۴۹] قرار است، تغییرات اتفاق بیافتد؟

بنابراین، طبقه بندی موجود دارای ۳ بعد است؛ اولین این ابعاد، بعد اجزا است که دارای اجزای مختلف و دلخواه است که توسط طراح مشخص می گردد، ۲ بعد دیگر از این ابعاد، به ترتیب دارای ۳ (قطعی، تطبیقی و خود- تطبیقی) و ۲ طبقه ی (مطلق و نسبی) است [Eib07].

ترکیب های مختلف دو بعد آخر در جدول (۱) ارائه شده است.

جدول (۱-۱) - طبقه بندی تنظیم پارامترهای استراتژی در الگوریتم های تکاملی – انواع کنترل پارامترها از نقطه نظر دو بعد نوع و ملاک

	Deterministic	Adaptive	Self- adaptive
Absolute	✓	✓	-
Relative	-	✓	✓

مجموع رشته های مختلف هوش مصنوعی ما را به رهیافت های کنترلی نوینی هدایت می کند؛ که این رهیافت ها بصورت الگوریتم های هایبرید مورد استفاده قرار می گیرند. بنا بر تعریف، الگوریتم های هایبرید از ترکیب دو یا چند تکنیک مختلف بوجود می آیند. الگوریتم های تکاملی با موفقیت با روش های دیگر موجود در محاسبات نرم[۵۰]، نظیر منطق فازی و شبکه های عصبی و محاسبات سخت[۵۱] نظیر سیمپلکس و گرادیان نزولی بطور هایبرید مورد استفاده قرار گرفته است. اما دلیل اصلی برای استفاده ی الگوریتم تکاملی و این گونه روش ها بطور هایبرید، تلفیق توانایی های تکنیک های مختلف به قصد فائق آمدن بر ضعف ها و مشکلات موجود در یکدیگر است.

یکی از این رهیافت ها شامل منطق فازی است، مهمترین ویژگی منطق فازی این است که یک راه موثر برای انتقال دانش انسانی به فرم قابل فهم برای ماشین را بوجود می آورد. از دیگر ویژگی های مهم منطق فازی

49Evidence
50Soft Computing
51Hard Computing