سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران
جزیره کیش، خلیج فارس، ایران
۱۹ الی ۲۱ اسفند ۱۳۸۶
پوستر

جهش سازگار پذیر در استراتژی تکامل با استفاده از یادگیری Q فازی

سید سعید معصوم زاده
دانشجوی کارشناسی ارشد
دانشکده رایانه و فناوری اطلاعات دانشگاه آزاد اسلامی قزوین، ایران
saeid@masoumzadeh.com

کورش مشگی
دانشجوی کارشناسی
دانشکده رایانه و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر تهران، ایران
kouroshmeshgi@aut.ac.ir

سعید شیری قیداری
استادیار
دانشکده رایانه و فناوری اطلاعات، دانشگاه صنعتی امیر کبیر تهران، ایران
shiry@aut.ac.ir

چکیده:

الگوریتمهای تکاملی یا به طور خاص استراتژی تکامل دارای پارامترهایی می باشند که با تغییر آنها در یک مسئله نتایج متفاوتی کسب می‌شود. روش پیشنهادی ما سعی میکند پارامتر σ در عملگر جهش با استفاده از نویز گوسین را سازگار پذیر کند. روش پیشنهادی در مسائل بهینه سازی برای پیدا کردن مینیمم توابع مورد آزمایش قرار گرفته و برتری نتایج آن نسبت به روشهای دیگر نشان داده شده است.

واژهای کلیدی

استراتژی تکامل، جهش سازگار پذیر، یادگیری تقویتی فازی

۱- مقدمه:

یکی از ارتقاءهای پیشنهاد شده بر روی الگوریتمهای تکاملی، کاستن از تنظیمات آنهاست. در میان پارامترهایی که باید تنظیم شوند، پارامترهای عملگرهای بازترکیبی و جهش و نرخ آنها نیز به چشم می‌آید. این تنظیمات عملگرها معمولاً در روند وفق دادن توجه زیادی می‌طلبند چرا که مجاری اصلی اکتشاف و وسعت[1] بشمار می‌روند. ارتقای عملگرهای الگوریتم تکاملی، کارایی را در هر دو زمینه کیفیت و راه حل پیدا شده با توجه به زمان مصرفی بالا می‌برد[1]. در ابتدای مطرح شدن این پیشنهاد، دیونگ مجموعه‌ای از مسائل بهینه سازی توابع عددی را به عنوان «بسته آزمون دیونگ» معرفی کرد و یک تنظیم بهینه را با استفاده از روش خود برای آنها پیدا کرد[2]. گریفستات از خود الگوریتم ژنتیک اعمال شده روی بسته دیونگ استفاده کرد که از روش دیونگ عملکرد بهتری دارد[3]. سپس روش جستجوی کورکورانه نیز توسط دیویس[4] و دیوید و همکاران[5] روی بسته دیونگ اعمال شد. تمرکز این روش روی تنظیم بهینه نرخ عملگرها می‌باشد. بهرحال از آنجا که تئوری، برای پیش بینی کردن تنظیمات ناکافی بود، مکانیزمهای تطبیقی مورد توجه قرار گرفت. در این مقاله عملگر جهش مورد بررسی قرار گرفته است.

مطالعات گذشته نشان داده‌اند که تغییر دادن احتمال جهش، نسبت به استفاده از جهش با نرخ ثابت کاراتر است (ر.ک. به مراجع [2، 3، 5- 10 از مرجع[11]). شایان ذکر است که در زمینه تطابق در مؤلفه بازترکیبی الگوریتمهای تکاملی نیز کارهایی صورت گرفته است که از جمله آنها می‌توان به الگوریتم تقاطع دودویی شبیه سازی شده[12]، الگوریتم فاصله همینگ زیاد و الگوریتم فاصله همینگ کم[7] اشاره کرد.

به طور کلی دو خط سیر برای تحقیقات در زمینه نیازمندیهای اکتشاف وگسترش جهش وجود دارد[1]: ۱- تغییرات با مرور زمان ۲- تغییرات در جهت سازگاری. دسته اول شامل الگویتمهایی شبیه روش گرما سازی شبیه سازی شده[۲] است که در آن طی نسلها احتمال جهش از ۰/۵ تا صفر کاهش می‌یابد[13]، یا چند برنامه زمانی متفاوت برای کوره تعیین می‌کند تا احتمال جهش را تغییر دهد[14]. دسته دوم که به نام روش بر خط نیز نامگذاری شده‌اند[15]، از اطلاعات حالت فعلی الگوریتم تکاملی مانند اطلاعات کل جمعیت، افراد جمعیت، یا مؤلفه‌های الگوریتم تکاملی برای تنظیم آن استفاده می‌کند. الگوریتمهای تکاملی با پارامترهای غیر ثابت بر مبنای مکانیسم تطبیق به سه زیرکلاس تقسیم می‌شوند[6، 16]:

۱- کنترل پارامتر پویا: در این طرح یک زمانبندی اکیداً نزولی برای کنترل پارامتر تهیه می‌شود که در آن با تولد نسلهای جدیدتر، احتمال جهش بر اساس یک راهبرد معین کاهش می‌یابد. بهترین نتایج این کلاس تا کنون از طرح باخ و شوتز[17] بدست آمده است.

۲- کنترل پارامتر خود انطباق: این راه حل از خود روش تکاملی برای تکامل پارامترها استفاده می‌کند. مبنای عملکرد این روش، رابطه غیرمستقیم بین کنترل پارامتر مناسب و افراد با شایستگی بالا از جمعیت است. خود تطبیقی در مسائل بهینه سازی پیوسته بوسیله استراژی تکامل و برنامه نویسی تکاملی نمایش موفقی داشته است[8، 10]. یکی از الگوریتمهای این کلاس (که روش ما با آن مقایسه شده‌است)، روش جهش گوسی خود انطباقی است. در این روش به هر کروموزوم یک ژن که مقدار آن انحراف معیار جهش گوسی را کنترل می‌کند، اضافه شده است. این ژن اجازه می‌یابد تا خود نیز در بازترکیبی و جهش شرکت یابد، ولی مستقیماً در محاسبه شایستگی شرکت نمی‌یابد [23].

۳- کنترل پارامتر سازگار پذیر: در طی فرایند بهینه سازی، می‌توان اطلاعات بازخوردی را در مورد چگونگی پیشرفت و خوب بودن پارامترها استخراج کرد. از نخستین نمونه‌های این روش، «قانون

نسبت جهشهای موفق را به کل جهشها نزدیک ۰/۲ نگه دارد، و اینکار را با افزایش طول گام، زمانی که نسبت بالاتر از ۰/۲ است و بالعکس انجام می‌دهد. یکی از روشهای جالب در این کلاس، روش GA-RL است که مسئله انتخاب عملگر را در چهارچوب یادگیری تقویتی مدل می‌کند[19]. این مدل از دو بخش تشکیل شده است: یک الگوریتم ژنتیک و یک عامل یادگیری تقویتی از نوع یادگیری Q که اطلاعات الگوریتم ژنتیک و جمعیت آنرا دریافت می‌کند و احتمال انتخاب هر عملگر را در الگوریتم ژنتیک دستکاری می‌کند.

کارایی اغلب راهبردهای تنظیم پارامتر ناایستا در مقابل روشهای با نرخ ثابت بهتر است، لیکن این روشها نقاط ضعفی نیز دارند. روش کنترل پویای پارامتر، نیاز به یک زمانبندی تعبیه شده دارند که مطابق آن پارامتر معمولاً کاهش می‌یابد. روشهای خود انطباق به چنین جدولی نیاز ندارند لیکن غالباً پیچیده‌اند و به راحتی اثبات نمی‌شوند[11]. روشهای سازگار پذیر به بازخوردی نیازمند هستند تا پارامتر را مطابق آن تغییر دهند. این بازخورد معمولاً بسیار نویزی است و نیز به سادگی بدست نمی‌آید.

با بررسی روشهایی که باعث سازگار پذیر شدن جهش می گردند، سه سطح از سازگاری مشاهده می‌شود:

۱- سازگار پذیری عملگرها: از میان روشهای گوناگون جهش که به عنوان عملگرهای جهش شناخته می‌شود، الگوریتم عملگری را انتخاب می‌کند که بهترین پاسخ را بدهد. در این میان احتمال هر عملگر تطبیق می‌یابد. ایده سازگاری عملگرها طی بهینه سازی در مرجع[20] مرور شده است. دیویس الگوریتمی را پیشنهاد داده است که به عملگرهایی که در تشکیل فرزندان شایسته تر دخیل بوده‌اند، اعتبار تخصیص می‌دهد[21]. در روش ADOPP که توسط جولستورم ارائه شده است، احتمال هر عملگر پس از هر جهش یا بازترکیبی ژنتیکی سازگارتر می‌شود[9]. تاسون و راس نیز یک روش کشف کننده برای تخصیص احتمال هر عملگر با توجه به محصول آن(شایستگی متوسط فرزندان در برابر والدین) پیشنهاد کرده‌اند و به این نتیجه رسیده‌اند که احتمالهای ابتدایی عملگرها مهمترین فاکتورها در تعیین کارایی هستند[22]. الگوریتم سازگارپذیر ژنتیک نیز از دو الگوریتم ژنتیک استفاده می‌کند. الگوریتم سطح مسئله در فضای مسئله به دنبال راه حلهای بهینه موجود می‌گردد. به طور همزمان الگوریتم سطح عملگر، از میان ۱۷ عملگر فهرست شده در[1]، الگوریتم مناسب را برای سطح مسئله انتخاب می‌کند.

۲- سازگار پذیری پارامتر احتمال: جهش یا عدم جهش یک ژن مسئله‌ایست که توسط یک پارامتر احتمال تصمیم گیری می‌شود. روش کنترل سازگارپذیر نرخ جهش با بهره ثابت از یادگیری تصادفی منهتن برای تصمیم گیری اینکه پارامتر باید کاهش یا افزایش یابد استفاده می‌کند. این تصمیم بسیار نویزی است و دو پارامتر آزاد در فرایند کنترل تطبیقی تضعیفی نرخ جهش، با گامهای جسورانه تر روش قبل را بهبود بخشیده است. این الگوریتم برای ملایم کردن نوسانات شدید پارامتر احتمال که از گامهای بلند ناشی می شود، از پارامتر آزاد جدیدی به نام عامل تضعیف استفاده می‌کند. این عامل در هر جهش به والد اعمال می‌شود و با اینکار الگوریتم به اکتشاف دعوت خواهد شد[11].

۳- سازگار پذیری پارامترهای درونی عملگرها: هر عملگر ممکن است از یک یا چند پارامتر آزاد در درون خود بهره ببرد. با تنظیم این پارامترها می‌توان کارایی را تا حد زیادی بالا برد. الگوریتمی که در این مقاله به آن پرداخته می‌شود به این دسته تعلق دارد تا پارامتر σ که یک پارامتر درونی در عملگر جهش در استراتژی تکامل می باشد را سازگار پذیر کند.

۲- استراتژی تکامل

یکی از انواع الگوریتمهای تکاملی استراتژی تکامل می‌باشد .عمده ترین اختلاف الگوریتمهای تکاملی با یکدیگر در نحوه بازنمایی آنها می‌باشد. که همین اختلاف باعث تغییر در عملکرد عملگرهای تنوع در اینگونه الگوریتمها گردیده است. نحوه بازنمایی در استراتژی تکامل بصورت بردار اعداد حقیقی می‌باشد از اینرو عملگرهای جهش و بازترکیبی در این الگوریتم متفاوت از الگوریتمهای دیگر می باشد[18]. جدول (۱) گزارش فنی استراتژی تکامل را ارائه می دهد.

جدول۱- گزارش فنی استراتژی تکامل

بردار اعداد حقیقی	بازنمایی
گسسته یا واسطه	باز ترکیبی
انحراف (نویز) گوسین	جهش
تصادفی یکنواخت	انتخاب والدین
(μ,λ) or $(\mu+\lambda)$	انتخاب بازماندگان

۲-۱-عملگر جهش:

به علت نحوه بازنمایی استراتژی تکامل، عملگر جهش از یک نویز تصادفی ایجاد شده از توزیع نرمال با متوسط صفر و انحراف معیار σاستفاده می‌کند. الگوریتم کلی به این ترتیب می باشد که نویز تصادفی ایجاد شده با یک احتمال خاص به هر یک از ژنوم های کروموزوم اضافه می‌شود[23].

Chromosome: $\langle x_1 , \dots , x_n \rangle$

$$x'_i = x_i + N(0, \sigma) \qquad (1)$$

تنها پارامتر متغیر در فرمول بالا پارامتر σ می باشد. تغییر پارامتر σ درهر لحظه از وضعیت حل مسئله می‌تواند باعث بهبود در جواب یا بدتر شدن پاسخ به مسئله گردد. از اینرو می‌توان گفت که پارامتر انتخابی پارامتری نادقیق می باشد.

یکی از مهمترین متد های یادگیری تقویتی[24، 25]، یادگیری Q می باشد که توسط واتکینز [26] معرفی شد. در یادگیری Q بجای انجام یک نگاشت از حالت به مقادیر حالتها، نگاشتی از زوج حالت/عمل[۴] به مقادیری که Q-value نامیده می‌شوند، انجام می‌گیرد.

۳-۱- تابع Q

به هرزوج **<حالت ، عمل>** یک مقدار Q(s,a) نسبت داده می‌شود. این مقدار عبارت است از مجموع پاداشهای دریافت شده وقتی که از حالت S شروع کرده، عمل a را انجام داده، بدنبال آن خط مشی موجود را دنبال کرده باشیم.

برای یادگیری تابع Q می‌توان از جدولی استفاده کرد که هر ورودی آن یک زوج <s,a> به همراه تقریبی است که یادگیر از مقدار واقعی Q بدست آورده است. مقادیر این جدول با مقدار اولیه تصادفی(معمولاً صفر) پر می‌شود. عامل بطور متناوب وضعیت فعلی S را تشخیص داده و عملی مثل a را انجام می‌دهد. سپس پاداش حاصله r(s,a) و همچنین حالت جدید ناشی از انجام عمل، s'=d(s,a) را مشاهده می‌کند. مقادیر جدول با استفاده از رابطه (۲) در بخش ۴ تغییر می‌کنند.

۴- یادگیری Q فازی :

در حالت کلی RL به طور مؤثر برای حل مسائل با تعدادی از وضعیتها و عملهای گسسته با ابعاد کوچک استفاده می‌شود و هنگامی که ابعاد وضعیتها و عملهای گسسته بزرگ می‌شود، اندازه جدول جستجو[۵] به حدی بزرگ می‌شود که از نظر زمان محاسباتی الگوریتم بسیار کند می‌گردد. از طرفی هنگامیکه وضعیتها یا عملها بصورت پیوسته مطرح می‌گردند، استفاده از جدول جستجو امکان پذیر نمی‌باشد. برای حل این مشکل از روش یادگیری Q فازی ارائه شده در [25] استفاده می‌کنیم.

$$\tilde{Q}_{t+1}(s,a) = \quad 0<\alpha<1$$
$$\tilde{Q}_t(s,a) + \alpha\,[r_{t+1} + \eta \max_{a'} \tilde{Q}_t(s',a') - \tilde{Q}_t(s,a)] \qquad (2)$$

۴-۱- یادگیری Q فازی برای فضای عمل گسسته

ساختار FQL شامل اجزای زیر میباشد:

- هر قانون R_i در پایگاه قوانین یک مجموعه از اعمال گسسته را که با آن پیوند خورده با نام U^i در بر دارد؛
- هر k امین عمل در هر قانون، $U^i(k)$، یک فاکتور کیفیت[۶] $W^i(k)$ (وزن) دارد، که از طریق فاز یادگیری عمل تنظیم خواهد شد. بنابراین هر قانون برداری از وزنها خواهد داشت، W_i شامل شایستگی عملهای پیوند خورده با آن قانون خواهد بود؛
- خروجی TS-FIS محاسبه خواهد شد:$\tilde{Q}_t(X_t,U(X_t))$؛
- مقدار کیفیت بهینه،Q-value ، برای وضعیت جاری که از طریق رابط فازی محاسبه خواهد شد:$Q^*_t(X_t)$.

FQL بوسیله استفاده از TS-FIS[۷] تابع Q-value را برای وضعیت-عمل جاری تقریب می‌زند. در فاکتورهای کیفیت قوانین، e-Greedy مثل برآیند کننده قوانین عمل خواهد کرد، تابع Q-value همراه با Q-value بهینه که از روش e-Greedy محاسبه می‌شود، برای محاسبه TD-error استفاده می شوند. بعد از آن بر اساس TD-error و بروز شدن قانون یادگیری TD، وزنهای عمل بروز خواهند شد. عامل سپس عمل را بر اساس کیفیت عمل[۸](وزنها) از اعمال مختلف موجود در مجموعه عمل U_i از هر قانون همراه با مکانیزم اکتشاف کردن/وسعت بخشیدن که Double e-Greedy نامیده می‌شود، انتخاب می کند.

۴-۲- اساس FQL

فرض کنید که X_t و $A(X_t)$ بترتیب وضعیت ورودی و مجوعه قوانین فعال در مرحله زمانی t می باشند و همچنین $\alpha_{Ri}(X_t)$ ارزش درستی قانون R_i بعد از ورود X_t می باشد. تقریب برای Q-value جاری از وضعیت- عمل جاری می‌تواند بصورت زیر نوشته شود:

$$\tilde{Q}_t(X_t , U(X_t)) = \frac{\sum_{Ri\in A(X_t)} \alpha_{R_i}(X_t)\, w_t^i(U_t^i)}{\sum_{Ri\in A(X_t)} \alpha_{R_i}(X_t)} \qquad (3)$$

U_t^i عمل e-Greedy قانون R_i در مرحله زمانی t و $U_t(X_t)$ بیشینه عمل e-Greedy می باشد. سپس مقدار کیفیت بهینه، Q-value، از وضعیت جاری از رابطه زیر بدست می آید:

$$Q_t^*(X_t) = \frac{\sum_{Ri\in A(X_t)} \alpha_{R_i}(X_t) \left[\max_{a\in U^i} w_t^i(a)\right]}{\sum_{Ri\in A(X_t)} \alpha_{R_i}(X_t)} \qquad (4)$$

با داشتن مقدار محاسبه شده Q-value، TD-error مستقیماً توسط رابطه زیر بدست می آید:

$$\tilde{\varepsilon}_{t+1} = r_{t+1} + \eta\, Q_t^*(X_{t+1}) - \tilde{Q}_t(X_t , U(X_t)) \qquad (5)$$
$$0 \le \eta \le 1$$

یادگیری بروز رسانی قانون با الگوریتم TD(0) برای تقریب تابع Q-value به صورت زیر میباشد:

$$w_{t+1}^i(U_t^i) = w_t^i(U_t^i) + \tilde{\varepsilon}_{t+1}\alpha_{R_i} \qquad (6)$$
$$\forall R_i \in A(X_t).$$

بعد از رویه بالا، انتخاب عمل با توجه به وزنهای عملها در هر مجموعه عمل U_i پیوند خورده با هر قانون در یک شرایط رقابتی توسط

کنترلی بر روی این انتخاب خواهد داشت.

$$U_t^i = U^i(k) \Big| EE(U^i(k)) = \max_{a \in U^i} \{EE(a)\} \qquad (7)$$

$$Where\, EE(a) = w_t^i(a) + \frac{\theta}{e^{n_t(a)}}, \theta > 0.$$

θ یک ضریب تاثیر مثبت در اکتشاف و $n_t(a)$ تعداد دفعات انتخاب عمل a از ابتدا تا کنون می باشد.

$$U_t(X_t) = \max_{R_i \in A(X_t)} \{EE(U_t^{i^*}).\alpha_{R_i}(X_t)\} \qquad (8)$$

۵- جهش سازگار پذیر با استفاده از FQL :

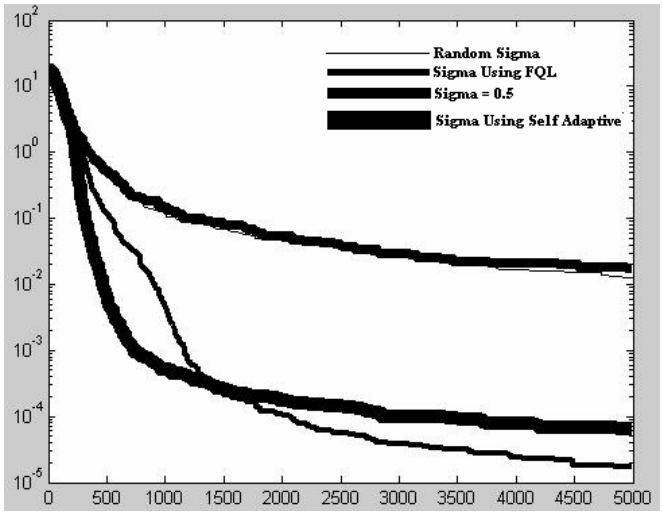
در این مقاله آنچه باعث سازگار پذیر شدن عملگر جهش در استراتژی تکامل گردیده، انطباق پذیری پارامتر σ در شرایط پویای مسئله انتخاب شده جهت بهینه سازی می باشد . در این قسمت ابتدا برای سازگار پذیر کردن عملگر جهش، آن را در قالب یک مدل استاندارد یادگیری تقویتی مطرح می کنیم.

- یک بازه پیوسته از حالتهای محیط S؛ شامل مقدار تابع شایستگی و اختلاف مقدارتابع شایستگی در زمان t با مقدار تابع شایستگی در زمان t-1؛
- یک مجموعهٔ گسسته با ابعاد بزرگ ازواکنشهای عامل: A ؛ شامل یک مجموعه مقادیر گسسته از مقدار پارامتر σ؛
- پاداش r ، شامل درصد میزان تغییر مقدار تابع شایستگی است که به ازای انجام عمل، عامل دریافت می کند و به عنوان یک بازخورد در مسئله ما مطرح است.

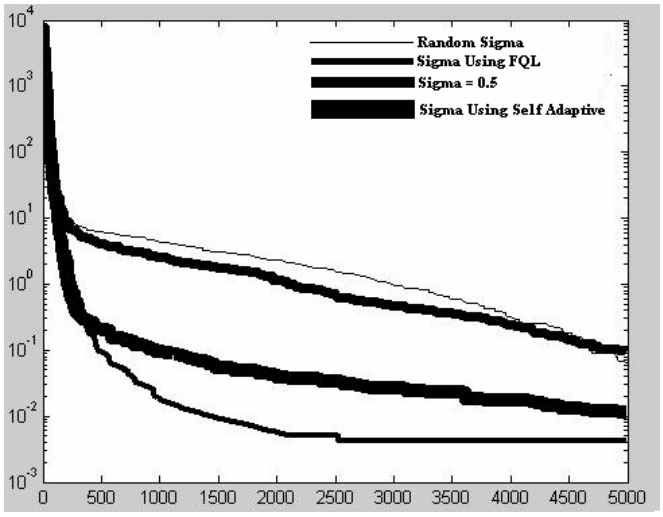
در پیاده سازی توسط FQL، حالتهای محیط S ، به عنوان ورودی به FIS داده می شود. ورودی بعد از فازی سازی و آتش کردن قوانین و تاثیر فاکتورهای کیفیت اعمال پیوند خورده با هر قانون مقدار Q-value را تقریب می‌زند. سپس مقدار کیفیت بهینه ، Q-value محاسبه می‌شود، با داشتن مقدار محاسبه شده Q-value، TD-error و بعد از آن وزنها یا فاکتور های کیفیت عملها بروز می‌شود. و سپس عمل a که در حقیقت مقدار پارامتر σ می‌باشد انتخاب می‌شود.

۶- نتایج عملی:

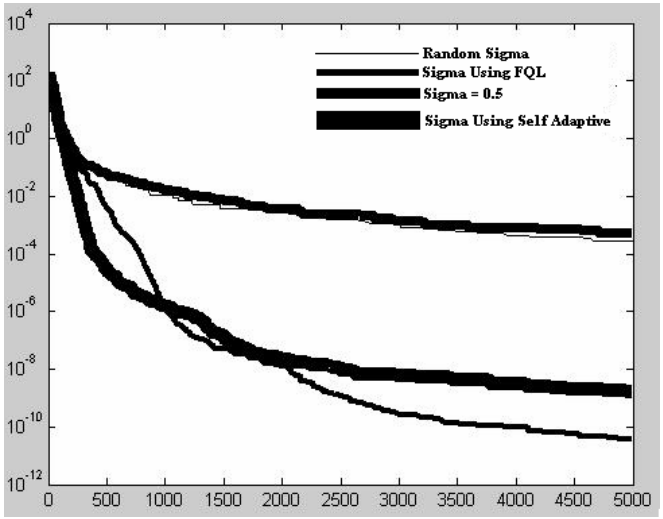
جهش سازگار پذیر ارائه شده برای پیدا کردن مینیمم عمومی، بر روی سه تابع شفر[۹]، آکلی[۱۰] و روزنبروک[۱۱] استفاده شده است. نتایج نشان می‌دهد که عملکرد جهش سازگار پذیر با استفاده از یادگیری Q فازی بطور کلی نسبت به عملکرد سه روش سیگما ثابت، سیگما تصادفی، و سیگما خود انطباق برتری داشته است(شکل ۲). شکلهای (۱) و (۳) نشان می‌دهد که اگرچه جهش سازگار پذیر با استفاده از FQL در ابتدا کندتر از جهش با روش خود انطباق بوده است، ولی با گذشت زمان داشته است.



شکل۱ - بررسی تابع Ackley



شکل۲ - بررسی تابع Rosenbrock



شکل۳- بررسی تابع Sphere

۷- نتیجه گیری:

روشهای با نرخ ثابت کارایی بهتری را از خود نشان می دهند. روشهای خود انطباق اگرچه باعث غیر ثابت شدن پارامتر می گردند، اما بازخورد مستقیمی از خوب یا بد بودن مقادیر پارامتر دریافت نمی‌کنند. اما در روشهای سازگار پذیر در طی فرایند بهینه سازی، می‌توان اطلاعات بازخوردی را در مورد چگونگی پیشرفت و خوب بودن پارامترها استخراج کرد اما از طرفی به این نکته باید توجه داشت که بازخورد (در روش پیشنهادی ما، پاداش) برای هر مسئله‌ای به سادگی بدست نمی-آید. ما معتقدیم که انتخاب بازخورد مناسب در روشهای سازگار پذیر باعث می شود تا این روشها نتیجه بهتری را نسبت به روش خود انطباق حاصل کنند.

۸- مراجع :

[1] S.Y. Koh, "An Adaptive Genetic Algorithm for Permutation Based Optimization Problems," in *AIAI2005- Second IFIP Conference on Artificial Intelligence Applications and Innovations* Beijing, 2005.

[2] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," University of Michigan,Ph.D. Dissertation, 1975.

[3] J. Grefenstette, "Optimization of control parameters for genetic algorithms." vol. 16: IEEE Press Piscataway, NJ, USA, 1986, pp. 122-128.

[4] L. Davis ,*Handbook of genetic algorithms*: Van Nostrand Reinhold New York, 1991.

[5] J. D. Schaffer, R. A. Caruana, L. J. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1989, pp. 51-60.

[6] T. Back, "The Interaction of Mutation Rate, Selection, and Self-Adaptation Within a Genetic Algorithm. Paralel Problem Solving from Nature 2," Elsevier, Amsterdam, 1992.

[7] B. K. S. Cheung ,S. Y. Yuen, and C. K. Fong, "Enhancement in Performance of Genetic Algorithm for Object Location Problem," in *8 thInternational Conference on Control, Automation, Robotics* 2004, pp. 692–697.

[8] D. B. Fogel, *Evolutionary computation: toward a new philosophy of machine intelligence*: IEEE Press Piscataway, NJ, USA, 1995.

[9] B. A. Julstrom, "What Have You Done for Me Lately? Adapting Operator Probabilities in a Steady-State Genetic Algorithm," in *6th Intl. Conf. on Genetic Algorithms and their Applications* :Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1995, pp. 81-87.

[10] H. P. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*: John Wiley & Sons, Inc. New York, NY, USA, 1993.

[11] D. Thierens, "Adaptive mutation rate control schemes in genetic algorithms." vol. 1, 2002.

[12] K. Deb and H. G. Beyer, "Self-Adaptive Genetic Algorithms with Simulated Binary Crossover." vol. 9: MIT Press, 2001, pp. 197-221.

simulated annealing convergence theory onto the simple genetic algorithm," Ph.D. Thesis , University of Florida, 1991.

[14] T. C. Fogarty, "Varying the probability of mutation in the genetic algorithm," in *third international conference on Genetic algorithms* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1989, pp. 104-109.

[15] W. M. Spears, "Adapting crossover in a genetic algorithm," in *5th Conference on Evolutionary Programming*. vol. 25, 1992.

[16] R. Hinterding, Z. Michalewicz, and A. E. Eiben, "Adaptation in evolutionary computation: a survey," in *IEEE International Conference on Evolutionary Computation*, 1997, pp. 65-69.

[17] T. Back and M. Schutz, "Intelligent mutation rate control in canonical genetic algorithms," in *International Syposium on Methodologies for Intelligent systems*, 1996, pp. 158–167.

[18] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*: Frommann-Holzboog, 1973.

[19] J. E. Pettinger and R. M. Everson, "Controlling genetic algorithms with reinforcement learning," p. 692.

[20] J. E. Smith and T. C. Fogarty, "Operator and parameter adaptation in genetic algorithms," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 1, pp. 81-87, 1997.

[21] L. Davis, "Adapting operator probabilities in genetic algorithms," in *third international conference on Genetic algorithms* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1989, pp. 61-69.

[22] A. Tuson, "Adapting Operator Probabilities in Genetic Algorithms,Master Thesis ",UK: Department of Artificial Intelligence, University of Edinburg, 1995.

[23] R. Hinterding, "Gaussian mutation and self-adaption for numeric genetic algorithms," in *Evolutionary Computation, 1995., IEEE International Conference on*, Perth, WA, Australia, 29 Nov-1 Dec 1996, pp. On page(s): 384.-

[24] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement Learning: A Survey " *Journal of Articial Intelligence*, pp. 237-285, 1996.

[25] A. F. Naeeni, "Advanced Multi-Agent Fuzzy Reinforcement Learning," in *Computer Science Department*: Dalarna University, 2004-November-03.

[26] C. Watkins and P. Dayan, "Q-learning." vol. 8: Springer, 1992, pp. 279-292.

زیرنویس‌ها:

[1] Exploration and Exploitation
[2] Simulating Annealing
[3] Q-Learning
[4] State/Action
[5] Lookup Table
[6] Quality Factor
[7] Takagi-Sugeno Fuzzy Inference System
[8] Quality Action
[9] Sphere
[10] Ackley
[11] Rosenbrock