



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

گزارش کتبی پروژه درس هوش مصنوعی

الگوریتم کلونی زنبور مصنوعی

نگارش: سید محمدمهدی احمدپناه^۱

شماره دانشجویی: ۹۰۳۱۸۰۶

دی ۱۳۹۳

¹ smahmadpanah@aut.ac.ir

چکیده:

در سالهای اخیر، الهام گرفتن از طبیعت برای حل مسائل مختلف در دنیای علوم کامپیوتر بسیار رایج شده است. نمونه‌های مختلفی از آن نیز مثل الگوریتم‌های الهام گرفته شده از زندگی جمعی حیوانات مختلف مطرح شده و استفاده می‌شود. این گزارش، به بررسی الگوریتم کلونی زنبور عسل می‌پردازد و پس از معرفی الگوریتم و شبیه‌سازی آن، به مقایسه این الگوریتم با سایر الگوریتم‌های تکاملی می‌پردازد.

در رفتار زنبورهای عسل در دنیای واقعی، می‌توان خودتطبیقی و تقسیم کار را مشاهده کرد. زنبورها به نقش‌های مختلف دیده‌بان، ناظر، مشغول به کار و ... تقسیم می‌شوند و هر یک فعالیتی در راستای یافتن منابع غذایی مورد نیاز کندو را انجام می‌دهند. در این بین، نحوه ارتباط زنبورها با یکدیگر نیز حائز اهمیت است که با پرواز و رقصیدن‌های خاص، به سایر هم‌نوعان خود اطلاعاتی را به اشتراک می‌گذارند و این گونه کندو از وجود منبع غذایی غنی اطلاع می‌یابد.

با الهام از این رفتار و مدل کردن آن، می‌توان به الگوریتم کلونی زنبور مصنوعی یا به اختصار ABC^2 دست یافت. مطرح خواهد شد که این الگوریتم ساده است و دارای انعطاف بالایی به نسبت سایر الگوریتم‌های تکاملی می‌باشد. ضمناً پارامترهای کنترلی موجود در این الگوریتم بسیار کم است و این نکته نیز باعث افزایش کارایی الگوریتم در توابع مختلف می‌شود.

² Artificial Bee Colony

مقدمه:

در سال‌های اخیر، هوش موجود در گروه حیوانات به یک موضوع مورد علاقه برای پژوهشگران زیادی تبدیل شده است. می‌توان این هوش را اینگونه تعریف کرد که "هر تلاش برای طراحی الگوریتمها یا وسایل حل مسئله توزیع شده الهام گرفته شده از رفتار جمعی کلونی اجتماعی حشرات و اجتماعات سایر حیوانات". مثال کلاسیک یک گروه حیوانات، دسته زنبورها در کندو است. با این حال، این مثال‌ها قابل گسترش به سایر سیستم‌های دارای معماری مشابه نیز می‌باشد. یک کلونی مورچگان را نیز می‌توان به عنوان یک دسته در نظر گرفت که عامل‌های انفرادی آنها مورچه‌ها است یا در الگوریتم PSO^3 رفتار اجتماعی گروه پرندگان یا ماهی‌ها مدل شده است.

دو مفهوم اساسی "خود تطبیقی" و "تقسیم کار"، برای داشتن رفتار هوشمند دسته لازم و کافی است.

الف) خود تطبیقی را می‌توان به عنوان یک مکانیزم پویا تعریف کرد که نتیجه در ساختارهای سطح کلی یک سیستم به وسیله تعامل با اجزاء سطوح پایین‌تر خود دارد.

چهار ویژگی پایه که خود تطبیقی به آن وابسته است عبارتند از:

بازخورد مثبت^۴، بازخورد منفی^۵، نوسان‌ها^۶ و تعامل‌های چندگانه^۷.

۱) بازخورد مثبت یک قاعده سرانگشتی رفتاری ساده است که ایجاد ساختارهای مناسب را بهبود می‌دهد. رقص‌های زنبورها می‌تواند مثالی از این باشد.

۲) بازخورد منفی در برابر بازخورد مثبت است و باعث ایجاد موازنه در الگوی جمعی می‌شود. برای جلوگیری از اشباعی که ممکن است بر اثر خصوصیات موجود در شکارچیان، فرسودگی منبع غذا، شلوغی یا رقابت بین منابع غذا پیش بیاید، یک مکانیزم بازخورد منفی لازم است.

۳) نوسان‌ها مانند گام‌های تصادفی، خطاها و تغییر تصادفی وظایف بین افراد در دسته، برای خلاقیت و نوآوری ضروری و حیاتی است. تصادف و تنوع برای ساختارهایی که به دنبال جستجو برای یک راه حل هستند، الزامی است.

۴) به طور کلی، خود تطبیقی به یک تراکم کمینه‌ای از افراد دارای تعامل، که باعث استفاده از نتایج فعالیتهای آنها که به خوبی دیگران است می‌شود، نیاز دارد.

ب) در یک دسته، وظایف مختلفی وجود دارد که همزمان توسط افراد متخصصی انجام می‌شود. این پدیده را تقسیم کار می‌گویند. تقسیم کار باعث می‌شود تا دسته نسبت به تغییر شرایط فضای جستجو واکنش نشان دهد.

³ Particle Swarm Optimization

⁴ Positive Feedback

⁵ Negative Feedback

⁶ Fluctuation

⁷ Multiple Interactions

رفتار دسته زنبورهای عسل:

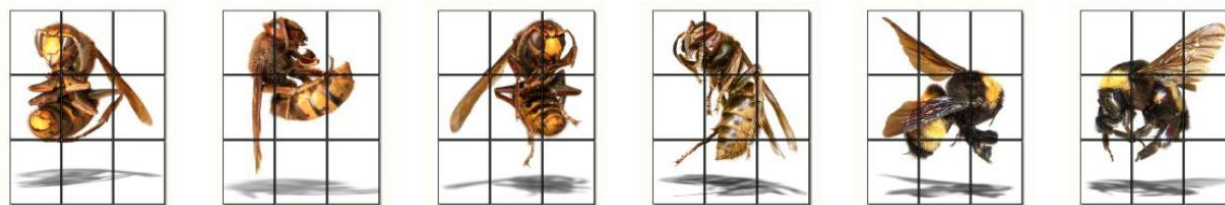
سه جزء ضروری برای انتخاب منبع غذایی مورد نظر: منابع غذایی، شکارچیان مشغول به کار و شکارچیان بیکار است و این مدل دو حالت رفتاری را تعریف می کند: استفاده از یک منبع شهد و رهاسازی یک منبع.

منابع غذایی: ارزش یک منبع غذایی بستگی به عوامل زیادی مثل نزدیکی به کندو، ارزش غذایی و راحتی استخراج انرژی از آن دارد. به منظور سادگی، "سودمندی" یک منبع غذایی با یک کمیت قابل بیان است.

۲) شکارچیان مشغول به کار: آنها به یک منبع غذایی مشخص مرتبط هستند که در آن مشغول به کار هستند. آنها با خود اطلاعاتی درباره آن منبع مشخص، نظیر فاصله و جهت آن نسبت به کندو و سودمندی منبع را حمل می کنند و این اطلاعات را با یک احتمال مشخصی به اشتراک می گذارند.

۳) شکارچیان بیکار: آنها مدام به دنبال یک منبع غذایی برای بهره برداری هستند. دو نوع شکارچی بیکار وجود دارد: دیده بان که محیط اطراف کندو را برای یافتن منابع غذایی جدید جستجو می کند و دیگری، ناظر که در کندو منتظر می ماند و اطلاعات به دست آمده از شکارچیان مشغول را جمع آوری می کند. متوسط تعداد دیده بانها در حدود ۵ تا ۱۰ درصد جمعیت کندو است.

تبادل اطلاعات بین زنبورها مهمترین رویداد در قالب دانش جمعی است. مهمترین بخش کندو با توجه به تبادل اطلاعات، ناحیه رقصیدن^۸ است. ارتباط بین زنبورها با کیفیت منبع غذایی که در ناحیه رقصیدن اتفاق می افتد، مرتبط است. این رقص، یک رقص جنبشی است.



شکل ۱

در حالیکه اطلاعات همه منابع غنی در اختیار یک ناظر، از طریق رقص وجود دارد، احتمالاً او رقص های زیادی را مشاهده می کند و تصمیم می گیرد که خودش را به سودمندترین منبع غذایی مشغول کند.

برای درک بهتر رفتار شکارچیان، درباره شکل شماره ۲ بحث می کنیم. فرض کنید که دو منبع غذایی پیدا شده A و B موجود است. در ابتدای کار، یک شکارچی بالقوه به عنوان یک شکارچی بیکار شروع می کند. آن زنبور هیچ دانشی نسبت به منابع غذایی اطراف کندو ندارد. دو گزینه برای این گونه زنبورها وجود دارد:

۱) می تواند یک دیده بان باشد و خود به خود شروع به جستجوی اطراف برای پیدا کردن غذا کند؛ به علت انگیزه داخلی یا سرنخ خارجی ممکن. (S)

۲) می تواند یک سرباز باشد که بعد از مشاهده رقص های جنبشی، شروع به جستجو برای یک منبع غذایی کند. (R)

⁸ Dancing area

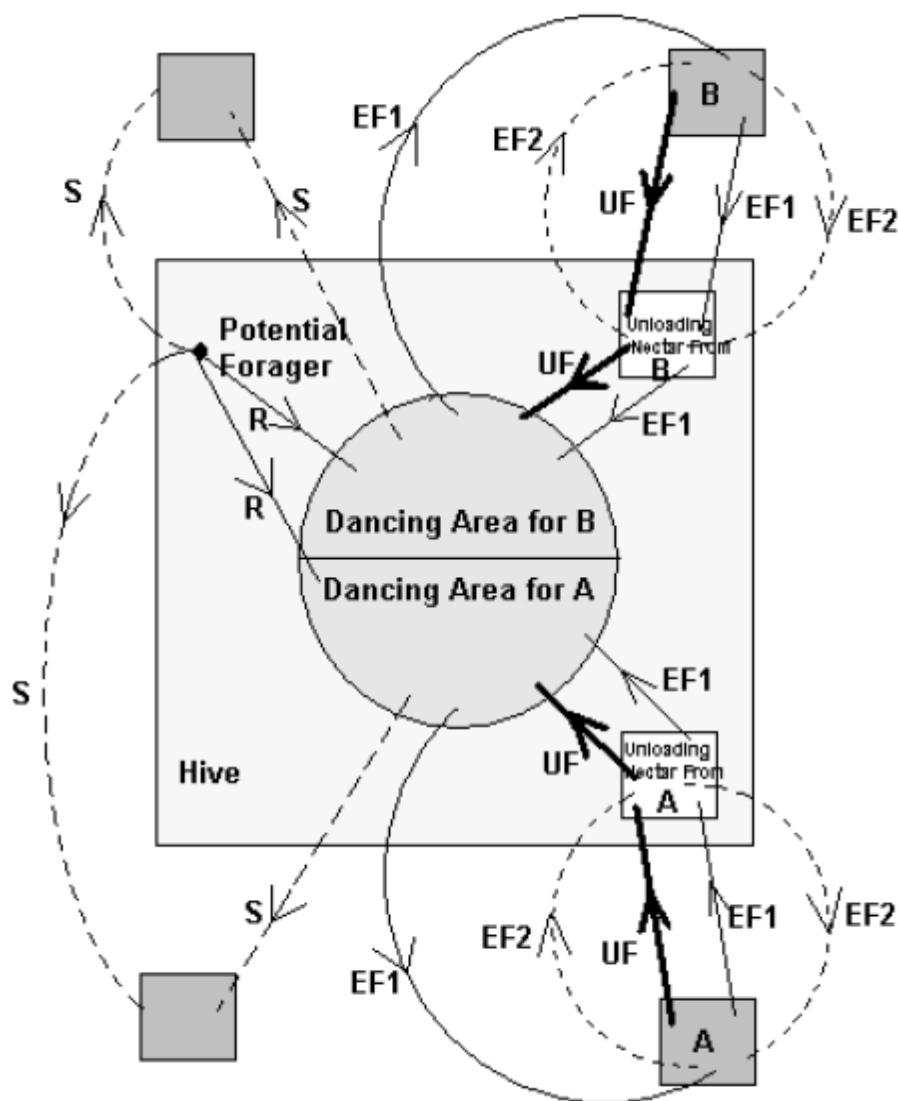
بعد از یافتن مکان منبع غذا، زنبور از توانایی خود برای حفظ موقعیت بهره می گیرد و فوراً شروع به بهره برداری از آن می کند. بنابراین، آن زنبور تبدیل به یک شکارچی مشغول می شود. زنبور شکارچی حجمی از شهد را از منبع جمع آوری می کند و به کندو باز می گردد و آن را در انبار غذا تخلیه می کند. بعد از تخلیه غذا، زنبور این سه گزینه را خواهد داشت:

(۱) به یک زنبور غیرمتعهد بعد از رهاسازی منبع تبدیل شود.

(۲) برقص و نیروی تازه ای را قبل از بازگشت به همان منبع، مشغول کند.

(۳) به شکار در همان منبع بدون مشغول کردن سایر زنبورها، ادامه دهد.

شایان ذکر است که همه زنبورها در یک زمان شروع به شکار نمی کنند. تحریک نشان می دهد که زنبورهای جدید با یک نسبتی به تناسب اختلاف بین تعداد کل زنبورها و تعداد شکارچیان حاضر، شروع به شکار می کنند.



شکل ۲

درباره زنبورهای عسل، ویژگی های اولیه که خودتطبیقی به آن وابسته است بدین شرح است:

بازخورد مثبت: به همان اندازه که مقدار شهد منبع غذایی افزایش می یابد، تعداد ناظرانی که آنها را مشاهده می کنند نیز افزایش می یابد.

بازخورد منفی: فرایند بهره برداری از منابع غذایی فقیر توسط زنبورها متوقف می شود.

نوسانها: دیده بانها یک فرایند جستجوی تصادفی برای یافتن منابع غذایی جدید انجام می دهند.

تعاملهای چندگانه: زنبورها اطلاعات خود را درباره منابع غذایی با سایر زنبورها در ناحیه رقص به اشتراک می گذارند.

شرح الگوریتم:

در این الگوریتم، یک هوش رفتاری خاص یک دسته زنبور عسل (رفتار شکار) مورد توجه قرار گرفته شده است و یک الگوریتم کلونی زنبور مصنوعی جدید شبیه سازی می کند. این رفتار زنبورهای واقعی برای حل مسائل بهینه سازی چندبعدی و چندنمایی توصیف می شود. در این مدل، کلونی زنبورهای مصنوعی شامل این سه گروه می شوند: زنبورهای مشغول، ناظران و دیده بانها. نیمه اول کلونی شامل زنبورهای مصنوعی مشغول و نیمه دوم شامل ناظران است. برای هر منبع غذایی، فقط یک زنبور مشغول وجود دارد. به عبارت دیگر، تعداد زنبورهای مشغول برابر است با تعداد منابع غذایی اطراف کندو. زنبور مشغول که منبع غذایی اش توسط زنبورها تمام می شود، تبدیل به دیده بان می شود. گامهای اصلی الگوریتم به شرح زیر است:

دیده بان ها را به منابع غذایی اولیه ارسال کن

تکرار

زنبورهای مشغول را به منابع غذایی ارسال کن و مقدار شهد آنها را تعیین کن

احتمال ارزش منابع را با اینکه کدام یک از آنها توسط زنبورهای ناظر ترجیح داده می شوند، محاسبه کن

زنبورهای ناظر را به منابع غذایی ارسال کن و مقدار شهد آنها را تعیین کن

فرایند استخراج از منابع تمام شده توسط زنبورها را متوقف کن

دیده بان ها را برای جستجو فضا به طور تصادفی برای یافتن منابع غذایی جدید ارسال کن

بهترین منبع غذایی پیدا شده تا کنون را در حافظه نگه دار

تا زمانی که (نیازها برطرف شده)

هر چرخه جستجو شامل این سه گام است: حرکت زنبورهای مشغول و ناظر به منابع غذایی، محاسبه مقدار شهد و تعیین زنبورهای دیده بان و جهت دهی آنها به منابع غذایی ممکن. یک موقعیت منبع غذایی، بیان کننده یک راه حل برای مسئله ای است که باید بهینه شود. مقدار شهد یک منبع غذایی، با کیفیت راه حل ارتباط دارد. ناظران در منابع غذایی با استفاده از یک احتمال مبتنی بر فرایند انتخاب، قرار می گیرند. همانطور که مقدار شهد یک منبع غذایی افزایش می یابد، احتمال ارزش منبع غذایی ای که توسط ناظران ترجیح داده می شود، نیز افزایش می یابد.

هر کلونی زنبور دارای دیده بانهایی است که جستجوگران آن کلونی به حساب می آیند. جستجوگران هیچ کمکی در حین جستجو برای غذا ندارند. آنها صرفاً به دنبال هر نوع منبع غذایی هستند. در نتیجه این رفتار، دیده بانها توسط هزینه های کم جستجو و یک میانگین پایین در کیفیت منبع غذا مشخص می شوند. بعضی اوقات، دیده بانها تصادفاً منابع غذایی غنی و ناشناخته ای را می یابند. در مورد زنبورهای مصنوعی، دیده بانهای مصنوعی می توانند اکتشاف سریع گروه راه حل های شدنی^۹ را به عنوان یک کار^{۱۰} داشته باشند. در این کار، یکی از زنبورهای مشغول انتخاب می شود و به عنوان زنبور دیده بان تعیین می شود.

⁹ feasible

¹⁰ task

انتخاب توسط یک پارامتر کنترلی به نام "حد"^{۱۱}، کنترل می شود. اگر یک راه حل، یک منبع غذایی اصلاح نشده توسط تعدادی از پیش تعیین شده آزمایش، را بیان کند، پس آن منبع غذایی توسط زنبور مشغولش رها می شود و آن زنبور مشغول، به یک دیده بان تبدیل می شود. تعداد آزمایش ها برای رها سازی یک منبع غذایی برابر است با ارزش حد که یک پارامتر کنترلی مهم در ABC است. در یک فرایند جستجوی قوی، فرایندهای اکتشاف و استخراج حتما باید توأمان انجام شوند. در الگوریتم ABC، در زمانی که زنبورهای ناظر و مشغول، فرایند استخراج را در فضای جستجو انجام می دهند، دیده بانها فرایند اکتشاف را کنترل می کنند.

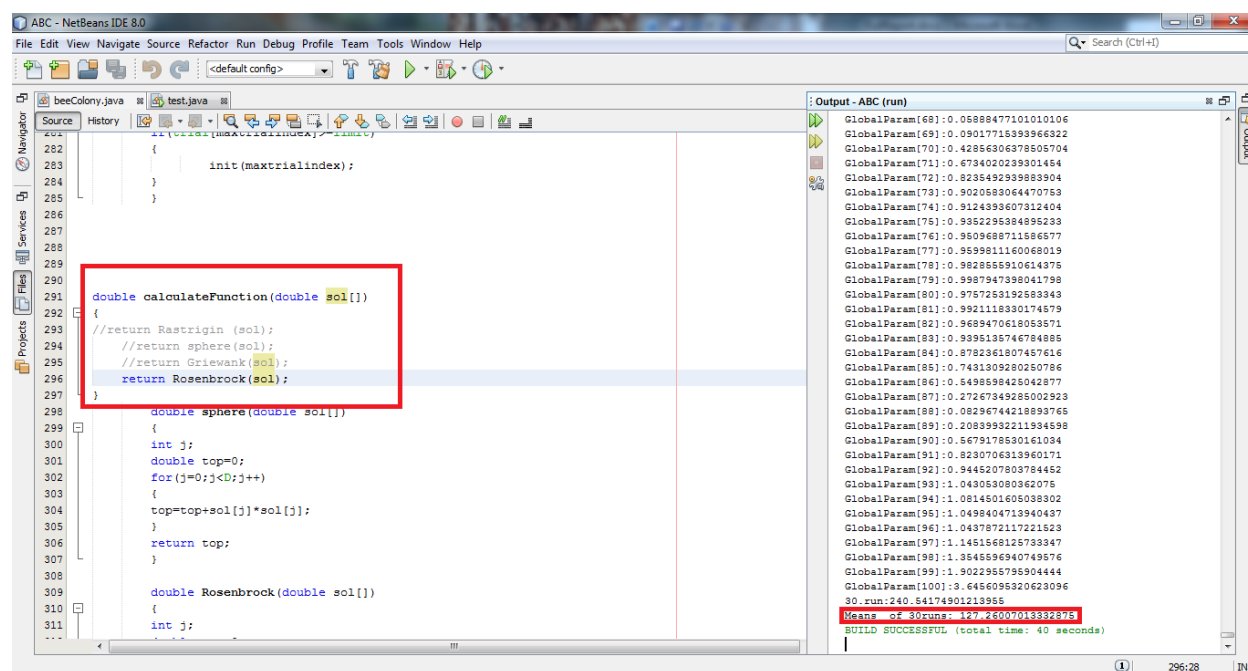
درباره زنبورهای عسل واقعی، نرخ استخدام نیروهای تازه، بیان کننده یک اندازه است که با چه سرعتی دسته زنبورها موقعیت یابی می کنند و منبع غذایی تازه یافته شده را بهره برداری می کنند. فرایند استخدام نیروی تازه مصنوعی می تواند به طور مشابه، بیان کننده اندازه سرعتی باشد که راه حل های شدنی یا راه حل های بهینه در مسئله بهینه سازی، یافته می شوند. بقا و پیشرفت دسته زنبورهای واقعی، بستگی به سرعت یافتن و کارایی استفاده از بهترین منابع غذایی دارد. به طور مشابه، راه حل بهینه مسائل پیچیده مهندسی به نسبت سرعت یافتن راه حل های خوب ارتباط دارد؛ مخصوصا برای مسائلی که لازم است بی درنگ^{۱۲} حل شوند.

¹¹ limit¹² real-time

پیاده‌سازی و نتایج:

نحوه اجرای برنامه ضمیمه شده:

این الگوریتم به زبان‌های مختلف پیاده‌سازی شده است. برنامه پیوست شده، به زبان جاوا می‌باشد که شامل دو کلاس beeColony و test است که بخش اصلی الگوریتم در کلاس اول پیاده شده است و برای اجرای بنچ مارک^{۱۳}های مختلف، تنها لازم است در متد calculateFunction در کلاس beeColony، خطوط کامنت شده را از کامنت در آورده تا مسئله مورد نظر طبق الگوریتم حل شود. در اینجا چهار بنچ مارک Sphere، Griewank، Rosenbrock و پیاده شده است.



شکل ۳

بنچ مارک	میانگین (پس از ۳۰ بار اجرا)
Rastrigin	۱۵,۹۹
Sphere	۷,۱۳
Griewank	۴,۶۰
Rosenbrock	۱۵۳,۷۷

جدول ۱

¹³ benchmark

تحلیل نتایج:

الگوریتم ABC، برای یافتن مینیمم سراسری سه تابع آزمون شناخته شده، اعمال شده است.

یکی از توابع، تابع Sphere است که پیوسته، محدب و درجه یک است. x در بازه $[-100, 100]$ است. مقدار مینیمم سراسری این تابع برابر با صفر است و راه حل بهینه آن $X_{opt} = (x_1, x_2, \dots, x_5) = (0, 0, \dots, 0)$ است.

تابع دوم یک مسئله بهینه سازی معروف به نام دره Rosenbrock است. بهینه سراسری در یک دره صاف سهمی گون باریک و طولانی است. بنابراین، همگرایی بهینه سراسری بسیار دشوار است. متغیرهای تابع به هم وابستگی زیادی دارند و شیبها به طرف بهینه نیست. x در بازه $[-2.048, 2.048]$ است و مقدار مینیمم سراسری برابر با صفر است و راه حل بهینه نیز $X_{opt} = (x_1, x_2) = (1, 1)$ است. بهینه سراسری این تابع تنها بهینه آن است و تابع یک‌نمایی است.

تابع سوم، تابع Rastrigin است که بر اساس تابع Sphere است، به همراه اضافه شدن مدلاسیون کسینوس برای تولید تعداد زیاد کمینه محلی. x در بازه $[-600, 600]$ است و مقدار مینیمم برابر با صفر است. راه حل بهینه برای این تابع، $X_{opt} = (x_1, x_2, \dots, x_{10}) = (0, 0, \dots, 0)$ است.

Functions	Ranges	Minimum Value
$f_1(\vec{x}) = \sum_{i=1}^5 x_i^2$	$-100 \leq x_i \leq 100$	$f_1(\vec{0}) = 0$
$f_2(\vec{x}) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$	$-2.048 \leq x_i \leq 2.048$	$f_2(\vec{1}) = 0$
$f_3(\vec{x}) = \sum_{i=1}^{10} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$-600 \leq x_i \leq 600$	$f_2(\vec{0}) = 0$

جدول ۲

در الگوریتم ABC، بیشترین تعداد چرخه‌ها ۲۰۰۰ در نظر گرفته می‌شود. درصد زنبورهای ناظر و مشغول برابر ۵۰٪ کلونی است و تعداد زنبورهای دیده بان یکی انتخاب شده است. افزایش در تعداد دیده بانها، فرایند اکتشاف را گسترش می‌دهد، در حالیکه افزایش تعداد ناظران روی یک منبع غذایی، فرایند استخراج را گسترش می‌دهد. پارامترها برای این الگوریتم به شرح زیر است:

Control parameters of ABC Algorithm	
swarmsize	20
limit	Number of onlooker bees *Dim.
number of onlookers	50% of the swarm
number of employed bees	50% of the swarm
number of scouts	1

جدول ۳

هر آزمایش، با دانه^{۱۴} های تصادفی ۳۰ بار تکرار شده است و میانگین مقادیر بهترین راه حل های تابع، ذخیره شده است. میانگین و انحراف معیارهای مقادیر تابع در این الگوریتم به شرح زیر است:

Functions	Mean	Std
$f_1(\vec{x})$ (5D Sphere)	4.45E-17	1.13E-17
$f_2(\vec{x})$ (2D Rosenbrock)	0.002234	0.002645
$f_3(\vec{x})$ (10D Rastrigin)	4.68E-17	2.64E-17

جدول ۴

(در مقاله ارجاع داده شده شماره ۲، به طور کامل بنچ مارکها و آزمایشها و نتایج آنها آمده است که مطالعه شد اما در مجال این گزارش نمی گنجد. لذا برای دریافت اطلاعات بیشتر در این خصوص، به این مقاله مراجعه شود.)

¹⁴ seeds

شبه کد الگوریتم با جزئیات:

۱: مقداردهی اولیه جمعیت راه حل ها $X_{i,j}$

۲: ارزیابی جمعیت

۳: چرخه $1 =$

۴: تکرار

۵: تولید راه حل های جدید (موقعیت های منبع غذا) $v_{i,j}$ در همسایگی $x_{i,j}$ ؛ برای زنبورهای مشغول، استفاده از رابطه

$$v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j})$$

(که در آن k یک راه حل در همسایگی i است و Φ یک عدد تصادفی در بازه $[-1,1]$ است.) و ارزیابی آنها

۶: اعمال فرایند انتخاب حریصانه بین x_i و v_i

۷: محاسبه مقدارهای احتمال P_i برای راه حل های x_i به وسیله مقادیر شایستگی آنها با استفاده از رابطه (1).

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (1)$$

برای محاسبه مقادیر شایستگی راه حل ها، می توان از رابطه (2) استفاده کرد:

$$fit_i = \begin{cases} \frac{1}{1 + f_i} & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{cases} \quad (2)$$

نرمال سازی مقادیر P_i بین $[0,1]$

۸: تولید راه حل های جدید (موقعیت های جدید) v_i برای ناظران از راه حل های x_i ، انتخاب شده بر اساس P_i و ارزیابی آنها.

۹: اعمال فرایند انتخاب حریصانه برای ناظران بین x_i و v_i .

۱۰: تعیین راه حل ها (منابع) رهاسده، در صورت وجود، و جایگزینی آن با یک راه حل تولید شده تصادفی x_i برای دیده بان با استفاده از رابطه (3).

$$x_{ij} = \min_j + \text{rand}(0,1) * (\max_j - \min_j) \quad (3)$$

۱۱: نگهداری موقعیت بهترین منبع غذا (راه حل) به دست آمده تاکنون

۱۲: چرخه = چرخه + ۱

۱۳: تا زمانی که (چرخه برابر بیشترین تعداد چرخه شود)

تفاوت این الگوریتم با سایر الگوریتم‌ها:

در این بخش، به مقایسه کارایی الگوریتم ABC با الگوریتم‌های بهینه‌سازی^{۱۵} GA، PSO، DE^{۱۶} و ES^{۱۷} می‌پردازیم. گرچه الگوریتم‌های مطرح شده، هر کدام با توابع هیوریستیک قابل بهبود می‌باشد اما تمرکز این بخش روی نسخه‌های استاندارد است. در آزمایش اول، از تعداد جمعیت یکسان و بیشترین تعداد ارزیابی برای همه مسائل استفاده شد، گرچه این یک حقیقت شناخته شده است که پارامترهای کنترلی به طور مشهودی بر روی کارایی الگوریتم‌ها تاثیر گذار است.

در حالیکه GA و DE از عملگرهای بازترکیبی برای تولید راه حل‌های جدید یا کاندید از راه حل‌های موجود استفاده می‌کنند، الگوریتم ABC این کار را نمی‌کند. این الگوریتم راه حل کاندید را از والدش با یک عملگر ساده مبتنی بر در نظر گرفتن اختلاف بخش‌های مشخص شده تصادفی آن والد و راه حل انتخاب شده تصادفی از جمعیت، به دست می‌آورد. این فرایند، سرعت همگرایی جستجو در مینیمم محلی را افزایش می‌دهد.

در GA، DE و PSO، بهترین راه حل پیدا شده تاکنون همیشه در جمعیت نگه داری می‌شود و می‌تواند برای تولید راه حل‌های جدید در DE و GA و سرعت جدید در الگوریتم PSO به کار برده شود. گرچه، در ABC، بهترین راه حل پیدا شده تاکنون همیشه در جمعیت نگه داری نمی‌شود، در حالیکه ممکن است با یک راه حل تولید شده تصادفی توسط یک دیده بان، جایگزین شود. بنابراین، ممکن است در تولید راه حل‌های آزمایش شده، نقشی نداشته باشد.

هر دو الگوریتم DE و ABC از یک فرایند انتخاب حریصانه بین راه حل‌های کاندید و والد بهره می‌برند. در ABC، مرحله زنبورهای مشغول، یک راه حل آزمایشی برای هر راه حل در جمعیت تولید می‌شود که در DE بدون وابستگی به کیفیت راه حل‌هاست. در مرحله زنبورهای ناظر، راه حل‌های با مقدار شایستگی بالاتر بیشتر برای تولید راه حل‌های آزمایشی مورد استفاده قرار می‌گیرند تا راه حل‌های با مقدار شایستگی پایین. این بدین معناست که جستجو در مدت زمان کوتاه‌تری و در جزئیات انجام می‌گیرد. این فرایند انتخاب، شبیه به انتخاب طبیعت یا انتخاب سابقه‌دار به کار برده شده در GA است.

در GA یا DE، فرایند جهش، یک تغییر در یک بخش تصادفی انتخاب شده از یک راه حل را ایجاد می‌کند تا تنوع لازم را در جمعیت فراهم آورد. در ABC، دو نوع مکانیزم برای کنترل تنوع جمعیت وجود دارد: الف) مانند DE یا GA، یک بخش انتخاب شده تصادفی از والد با یک مقدار مشخص شده تصادفی، برای به دست آوردن یک دنباله راه حل، تغییر می‌یابد. این تغییر نسبتاً کوچک است و برای جستجوی محلی و برای میزان سازی مناسب است. ب) به جای تغییر یک بخش از یک راه حل، یک راه حل از جمعیت حذف و یکی دیگر به طور تصادفی تولید می‌شود و در جمعیت توسط دیده بان قرار داده می‌شود. این مکانیزم قابلیت جستجوی سراسری الگوریتم ABC را فراهم می‌کند و از همگرایی زودرس جلوگیری می‌کند. این ویژگی، وابستگی کارایی الگوریتم به اندازه جمعیت را نیز کم می‌کند. بنابراین، یک تعادل خوب بین فرایند جستجوی محلی انجام شده توسط زنبورهای ناظر مصنوعی و مشغول و فرایند جستجوی سراسری انجام شده توسط دیده بانهای مصنوعی برقرار می‌کند. پس الگوریتم ABC نتایج بهتری را در مسائل چندمنامی و چندمتغیره به نسبت سایر الگوریتم‌های مورد بررسی، به دست می‌آورد.

جدای از بیشترین تعداد ارزیابی و اندازه جمعیت، یک GA استاندارد، سه پارامتر کنترلی (نرخ بازترکیبی، نرخ جهش، گپ نسل)، یک DE استاندارد، دارای حداقل دو پارامتر کنترلی (نرخ بازترکیبی، عامل مقیاس) و یک PSO ساده، دارای سه پارامتر کنترلی (عوامل یادگیری و اجتماعی، وزن سکون) هستند. همچنین، مقادیر حدی برای سرعت‌های V_{max} تاثیر بسزایی بر

¹⁵ Genetic Algorithm

¹⁶ Differential Evolutionary

¹⁷ Evolutionary Strategy

کارایی PSO دارد. الگوریتم ABC تنها یک پارامتر کنترلی "حد"، صرف نظر از اندازه کلونی و تعداد بیشینه چرخه، دارد. گرچه می توان حد را براساس این دو تعریف کرد و تنها دو پارامتر کنترلی تعداد بیشینه چرخه و اندازه کلونی را داشت. متعاقبا، ABC به سادگی و انعطاف DE و PSO است و البته پارامترهای کنترلی کمتری دارد.

ESها از عملگرهای بازترکیبی و جهش برای تولید فرزندان استفاده می کند در حالیکه ABC فقط از عملگر جهش استفاده می کند. گرچه نسخه ابتدایی ES به سادگی ABC است ولی نسخه های پیشرفته تر، پیچیده تر از ABC است. علاوه بر این، همه آنها پارامترهای کنترلی بیشتری را دارند.

با توجه به آزمایش های انجام شده می توان چنین نتیجه گرفت که کارایی الگوریتم ABC بهتر یا شبیه به این الگوریتم ها است، گرچه از پارامترهای کنترلی کمتری بهره می برد و می تواند به نحو موثری، در حل مسائل بهینه سازی چندنمایی و چندبعدی استفاده شود.

نتیجه گیری:

در این گزارش، یک الگوریتم بهینه سازی مبتنی بر هوش رفتاری دسته زنبورهای عسل توصیف شد. این الگوریتم در مقایسه با الگوریتم‌های مبتنی بر دسته موجود، ساده و منعطف است. این الگوریتم همچنین، مقاوم^{۱۸} است. با توجه به نتایج شبیه سازی ها، الگوریتم پیشنهاد شده برای حل مسائل بهینه سازی عددی یک‌نمایی و چندنمایی و چندمتغیره قابل استفاده است.

¹⁸ robust

مراجع:

- [1] D. Karaboga, "An Idea Based On Honey Bee Swarm For Numerical Optimization", Technical Report-TR06, 2005.
- [2] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm" Applied Mathematics and Computation 2009, pp. 108-132.
- [3] The Artificial Bee Colony Algorithm Homepage, <http://mf.erciyes.edu.tr/abc/> , available on 15-Jan-2015.
- [4] GRIEWANK Function, <http://www.sfu.ca/~ssurjano/griewank.html>, available on 17-Jan-2015.
- [5] RASTRIGIN Function, <http://www.sfu.ca/~ssurjano/rastr.html>, available on 17-Jan-2015.
- [6] SPHERE Function, <http://www.sfu.ca/~ssurjano/spheref.html>, available on 17-Jan-2015.
- [7] ROSENBROCK Function, <http://www.sfu.ca/~ssurjano/rosen.html>, available on 17-Jan-2015.