Archive of SID

الگوریتم فراابتکاری خفاش برای مسائل برنامه‌ریزی عدد صحیح

سمیه بخته

گروه ریاضی، دانشکده علوم ریاضی، دانشگاه شهید مدنی آذربایجان.
دانشجوی دکتری.
bakhtehs@yahoo.com

چکیده

الگوریتم خفاش یکی از الگوریتم‌های فراابتکاری می‌باشد که بر اساس اصل انعکاس صدای خفاش و دریافت صدا توسط این پرنده عمل می‌کند. این الگوریتم برای حل مسائل بهینه‌سازی پیوسته مناسب است و به طور مستقیم نمی‌توان برای حل مسائل برنامه‌ریزی عدد صحیح به کار برد. لذا در این مقاله هدف این است که این الگوریتم برای حل مسائل برنامه‌ریزی عدد صحیح توسعه یابد. الگوریتم صحیح خفاش(IBA) بر روی چند تابع استاندارد اجرا شده و نتیجه عملکرد آن با الگوریتم‌های فراابتکاری خفاش(BA)، ازدحام ذرات(PSO) و ژنتیک(GA) مورد مقایسه قرار گرفته است. نتایج آزمایشات نشان می‌دهند که الگوریتم IBA از دقت بیشتری برای یافتن نقطه بهینه برخوردار است.

کلمات کلیدی الگوریتم خفاش؛ بهینه‌سازی؛ برنامه‌ریزی عدد صحیح.

۱ پیش‌گفتار

الگوریتم‌های فراابتکاری که معمولاً الهام گرفته شده از طبیعت و فرایندهای فیزیکی می‌باشند، در حال حاضر به عنوان یکی از روش‌های قدرتمند برای حل بسیاری از مسائل بهینه‌سازی پیچیده به کار برده می‌شوند. الگوریتم خفاش یکی از الگوریتم‌های فراابتکاری الهام گرفته از طبیعت است که در سال ۲۰۱۰ توسط آقای یانگ معرفی گردید[۲]. این الگوریتم بر اساس اصول زندگی خفاش‌ها طراحی شده است. خفاش‌ها تنها پستانداران با بال هستند که برای شکار طعمه از انعکاس صدا استفاده می‌کنند. تاکنون الگوریتم BA برای مسائل بهینه‌سازی دودویی[۱] و برای مسائل بهینه‌سازی چند هدفی [۳] به کار گرفته شده است. با این وجود، بسیاری از مسائل بهینه‌سازی گسسته وجود دارند که قابل حل با الگوریتم‌های فراابتکاری موجود نمی‌باشند. لذا در این مقاله هدف این است که نوع صحیح الگوریتم خفاش ارائه گردد. برای اطلاع از پیشرفت‌های دیگر الگوریتم BA خواننده به [۴] ارجاع داده می‌شود.

۲ الگوریتم خفاش

الگوریتم بهینه‌سازی خفاش الهامی از خصوصیات ردیابی خفاش‌های کوچک در جستجوی شکار می‌باشد. به طوری که خفاش‌های کوچک می‌توانند در تاریکی مطلق با انتشار صدا و دریافت آن به شکار طعمه‌های خود بپردازند. برای توسعه این الگوریتم از سه قانون آرمانی زیر استفاده می‌شود:

- همه خفاش‌ها از انعکاس صدا برای تشخیص فاصله استفاده می‌کنند و تفاوت بین مواد غذایی و موانع پیشرو را می‌دانند.
- پرواز خفاش‌ها به طور تصادفی با سرعت v_i در مکان x_i با فرکانس ثابت f_{min} و طول موج مختلف λ و بلندی صوت A_0 به منظور شکار طعمه صورت می‌گیرد. همچنین آنها می‌توانند به طور خودکار امواج پخش شده و نرخ پالس‌های ارسالی خود را ($r \in [0,1]$) با توجه به نزدیکی شکارشان تنظیم کنند.
- با توجه به اینکه ممکن است بلندی صدا در بسیاری از جهات مختلف متفاوت باشد لذا فرض می‌کنیم که بلندی صدا از R_0(بیشترین مقدار) تا R_{min}(کمترین مقدار) متغیر می‌باشد.

طبق قوانین بیان شده، مکان x_i^t و سرعت v_i^t برای هر خفاش مجازی iام در تکرار t و همچنین فرکانس f_i به صورت زیر محاسبه می‌گردد:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad (۱)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*) \quad (۲)$$

$$x_i^t = x_i^{t-1} + v_i^t \quad (۳)$$

www.SID.ir

Archive of SID

که در آن $\beta \in [۰, ۱]$ یک بردار تصادفی با توزیع یکنواخت می‌باشد و x^* بهترین مکان فعلی است که در هر تکرار پس از مقایسه با موقعیت خفاش‌های مجازی انتخاب می‌شود. معمولاً فرکانس f را با $f_{min} = ۰$ و $f_{max} = ۱۰۰$ در نظر می‌گیرند. در هر تکرار، در جستجوی محلی یکی از جواب‌ها به عنوان بهترین جواب‌ها انتخاب شده و موقعیت جدید هر خفاش به طور محلی با گام تصادفی به صورت زیر به روز می‌شود:

$$x_{new} = x_{old} + \epsilon A^t. \qquad (۴)$$

که در آن $\epsilon \in [-۱, ۱]$ یک عدد تصادفی بوده و $A^t =< A_i^t >$ میانگین بلندی صدای خفاش‌ها در تکرار t می‌باشد. همچنین بلندی صدای A_i و نرخ پالس ارسالی r در هر تکرار به صورت زیر به روز می‌شود:

$$A_i^{t+۱} = \alpha A_i^t, \qquad r_i^{t+۱} = r_i^۰[۱ - exp(-\gamma t)] \qquad (۵)$$

که در آن α و γ مقدار ثابت می‌باشند و برای هر $۰ < \alpha < ۱$ و $r > ۰$، وقتی $t \to \infty$ داریم: $r_i^{t+۱} \to r_i^۰$ و $A_i^{t+۱} \to ۰$.
با توجه به مطالب ارائه شده در بالا، الگوریتم خفاش به صورت الگوریتم ۱ خلاصه می‌شود:

الگوریتم ۱۴ الگوریتم خفاش.

۱. جمعیت اولیه خفاش‌ها x_i، $i = ۱, \ldots, n$ را ایجاد کنید.

۲. به ازای $i = ۱, \ldots, n$ سرعت v_i، فرکانس f_i در x_i، نرخ پالس r_i و بلندی صدای A_i را ایجاد کنید.

۳. قرار دهید $t = ۱$.

۴. موقعیت‌ها را مقایسه کرده و جواب بهینه را بیابید.

۵. موقعیت‌های جدید موقتی با تنظیم فرکانس و به روز کردن سرعت همه خفاش‌ها ایجاد کنید(معادلات (۱)-(۳)).

۶. اگر $\text{rand} > r_i$، آنگاه یک جواب در میان بهترین جواب‌ها با گام تصادفی کرده و با استفاده از معادله (۴) یک جواب محلی در اطراف بهترین جواب انتخاب شده ایجاد کنید.

۷. یک جواب جدید با پرواز تصادفی تولید کنید.

۸. اگر $rand < A_i$ و $f(x_i) < f(x^*)$، آنگاه جواب‌های جدید را بپذیرید و طبق معادله (۵)، r_i را افزایش و A_i را کاهش دهید.

۹. خفاش‌ها را مرتب کرده و بهترین جواب x^* را پیدا کنید.

۱۰. اگر t به ماکزیمم مقدار خود رسید الگوریتم را متوقف کنید در غیر این صورت قرار دهید $t = t + ۱$ و به ۴ بروید.

۳ الگوریتم صحیح خفاش

الگوریتم خفاش برای حل مسائل بهینه‌سازی پیوسته معرفی شده است[۲]. اما در بسیاری از مسائل لازم است که جستجو در فضای گسسته انجام پذیرد که در اینجا نوع صحیح الگوریتم خفاش ارائه می‌گردد. برای طراحی الگوریتم IBA، با توجه به فضای جستجوی خفاش‌ها، برخی مفاهیم پایه‌ای سرعت و موقعیت خفاش‌ها به روز می‌شود.
با توجه به اینکه در فضای گسسته فقط با اعداد صحیح سر و کار خواهیم داشت، لذا هدف این است که ارتباطی مانند معادله (۳)، بین سرعت و موقعیت خفاش‌ها در فضای گسسته نیز ایجاد شود. ایده اصلی به این صورت است که بعد از اینکه جمعیت تصادفی اولیه خفاش‌ها ایجاد شد، ابتدا قسمت صحیح و اعشاری موقعیت ذرات را در حافظه جداگانه ذخیره کرده که به ترتیب با نماد Ix_i^t و Fx_i^t نشان داده می‌شوند سپس با استفاده از یک تابع انتقال، قسمت اعشاری موقعیت ذرات را به صفر یا یک تبدیل خواهیم کرد. آقای میرجلیلی و همکارانش، یک تابع انتقال برای سرعت در فضای جستجوی دودویی تعریف کرده‌اند که به صورت یک تابع احتمال در بازه $[۰, ۱]$ محدود شده است[۱]. تابع انتقال و موقعیت به روز شده به صورت زیر می‌باشد:

$$V(v_i^k(t)) = |\frac{۲}{\pi} \arctan(\frac{\pi}{۲} v_i^k(t))| \qquad (۶)$$

www.SID.ir

Archive of SID

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if rand} < V(v_i^k(t+1)) \\ x_i^k(t) & \text{if rand} \geq V(v_i^k(t+1)) \end{cases} \quad (۷)$$

که در آن $x_i^k(t)$ و $v_i^k(t)$ به ترتیب نشان‌دهنده موقعیت و سرعت i امین ذره در تکرار t و در k امین بعد است. همچنین $(x_i^k(t))^{-1}$ عکس مولفه $x_i^k(t)$ و rand یک عدد تصادفي در بازه $[0, 1]$ می‌باشد.
بنابراین با استفاده از تابع انتقال بیان شده، قسمت اعشاری مکان جدید را با استفاده از فرمول (۷) به روز می‌کنیم. در نهایت موقعیت جدید خفاش‌ها به صورت عدد صحیح $x_i^t = Ix_i^t + Fx_i^t$ به روز می‌گردد.

۴ نتایج عددی

الگوریتم IBA بر روی چهار تابع استاندارد در فضای ۱۰۰ بعدی اجرا و نتایج عملکرد آن با الگوریتم‌های فراابتکاری دیگر مقایسه شده است. در تمامی الگوریتم‌ها ماکزیمم تعداد تکرار برابر ۵۰۰ و اندازه جمعیت اولیه برابر ۴۰ می‌باشد. بهترین نتایج حاصل از اجرای الگوریتم‌های مورد نظر در ۱۰ بار تکرار در جدول ۱ آمده است. لازم به ذکر است که مقدار بهینه توابع f_1،...، f_3 برابر صفر و مقدار بهینه f_4 برابر ۴,۶۸۷- می‌باشد. کد مربوط به الگوریتم‌های مذکور در نرم افزار مطلب نسخه (R۲۰۱۲a)۷۳۹ .۰ ۱۴. ۷ نوشته شده و در کامپیوتر شخصی با پردازشگر Intel(R) core(TM)i۵ CPU ۲٫۵GHZ و حافظه ۶GB اجرا شده است. نتایج عددی حاصل نشان می‌دهند که الگوریتم IBA نسبت به الگوریتم‌های دیگر، عملکرد بسیار بهتر و دقت بیشتری برای یافتن جواب بهینه مسئله دارد.

جدول ۱: بهترین جواب‌های حاصل از اجرای الگوریتم‌ها.

Function	Rang		f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	PSO	0.00015952
		GA	0.15785
		BA	1.6388e-07
		IBA	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	PSO	367.9463
		GA	83.3283
		BA	31.0545
		IBA	0
$f_3(x) = -\sum_{i=1}^{n} \sin(x_i)(\sin(\frac{ix_i^2}{\pi}))^{20}$	[-32,32]	PSO	7.4851
		GA	6.0942
		BA	2.9026
		IBA	8.8818e-16
$f_4(x) = -\sum_{i=1}^{n} \sin(x_i)(\sin(\frac{ix_i^2}{\pi}))^{20}$	[0,π]	PSO	-3.5348
		GA	-3.9918
		BA	-4
		IBA	-4.4787

مراجع

[1] S. A. Mirjalili, S. M. Mirjalili and X. S. Yang. *Binary bat algorithm*, Neural Computing and Applications **25.3-4** (2014), 663-681.

[2] X. S. Yang, *A new metaheuristic bat-inspired algorithm*, Nature inspired cooperative strategies for optimization (NICSO 2010), Springer Berlin Heidelberg (2010), 65-74.

[3] X. S. Yang, *Bat algorithm for multi-objective optimisation*, International Journal of Bio-Inspired Computation **3.5** (2011), 267-274.

[4] X. S. Yang and X. He. *Bat algorithm: literature review and applications*, International Journal of Bio-Inspired Computation **5.3** (2013), 141-149.

”

www.SID.ir