

الگوریتم سنجاقک: یک تکنیک جدید بهینه‌سازی فراابتکاری برای حل مسائل تک هدف،

گسسته، و چند هدفه

چکیده

یک تکنیک هوشمند جدید ارائه شده است که آن الگوریتم سنجاقک نام دارد (DA). الگوریتم اصلی DA از رفتارهای ایستا و پویا متشکل از رفتار سنجاقک در طبیعت است. دو مرحله ضروری برای بهینه‌سازی، اکتشاف و بهره‌برداری، با تعامل اجتماعی سنجاقک‌ها که در جستجوی غذا و دوری از دشمنان در هنگام پیدایش هستند، طراحی شده است. این مقاله نسخه‌های چند هدفه و باینری را که به ترتیب نسخه چند هدفه DA (BDA) و چند هدفه DA (MODA) نامیده می‌شود را ارائه داده است. الگوریتم‌هایی که چندین توابع آزمون ریاضی و یک مطالعه موردی را که به صورت کیفی و کمی سنجیده می‌شوند را این مقاله ارائه کرده است. نتایج DA و BDA که باعث بهبود الگوریتم‌ها می‌شود و به صورت تصادفی مسائل را مقداردهی اولیه کرده است و به بهترین وجه بهینه‌سازی سراسری را انجام داده است، و نتایج بسیار دقیقی را از الگوریتم‌های دیگر در این ادبیات را ارائه کرده‌اند. نتایج MODA نشان می‌دهد که این الگوریتم بهترین راه‌حل پارتو را برای طیف گسترده‌ای از مسائل چند هدفه ارائه می‌دهد. مجموعه‌ای از طرح‌ها مسائل زیردریایی پروانه را برای نشان دادن محاسن MODA و برای حل مسائل واقعی که هنوز در اصل پارتو ناشناخته هستند را به کار می‌برند. توجه داشته باشید که الگوریتم‌های BDA، DA و MODA به صورت عمومی در سایت <http://www.alimirjalili.com/DA.html> قابل دسترس است.

کلمات کلیدی: بهینه‌سازی، بهینه‌سازی چند هدفه، بهینه‌سازی محدود شده، بهینه‌سازی باینری، معیار، هوش گروهی، تکاملی، الگوریتم‌های بهینه‌سازی ذرات، الگوریتم ژنتیک

1 معرفی

طبیعت پر از رفتارهای اجتماعی برای انجام وظایف مختلف است. گرچه هدف نهایی تمام افراد و رفتارهای جمعی زنده ماندن است، اما موجودات به دلایل مختلفی: به صورت شکار، دفاع، راه رفتن و غذا خوردن در گروه‌ها، گله‌ها، مدارس باهم تعامل دارند. به عنوان مثال، گرگ بسته به یکی از تعاملات اجتماعی برای شکار به صورت اختصاصی سازماندهی شده است. گرگ‌ها به پیروی از رهبر به صورت اجتماعی برای شکار شیطانی در مراحل مختلف تمایل دارند: تعقیب شکارچیان، چرخاندن خزه‌ها، آزار و اذیت قارچ‌ها و حمله به قارچ‌ها [1،2]. یک نمونه از دفاع جمعی مدارس ماهی‌ها در اقیانوس‌ها است. هزاران ماهی یک مدرسه را ایجاد می‌کنند و با هشدار دادن به یکدیگر از شکارچیان جلوگیری می‌کنند و کار شکارچیان را دشوار می‌کنند [3]. اکثر شکارچیان به منظور تقسیم این مدارس به زیر مدارس با حمله به آنها و در نهایت شکار کردن افرادی که جدا شده‌اند، تکامل پیدا می‌کنند.

ناوبری یکی دیگر از دلایل ازدحام بعضی از موجودات است. پرندگان بهترین نمونه‌هایی از این رفتارهای هستند که در بین قاره‌ها و در گله‌ها به راحتی مهاجرت می‌کنند. ثابت شده است که پیکربندی v-shaped پرواز به شدت موجب صرفه‌جویی در انرژی و به همان اندازه باعث توزیع در میان افراد گله می‌شود. آخرین باری نیست، که خوردن غذا یکی دیگر از دلایل اصلی تعاملات اجتماعی بسیاری از گونه‌ها در طبیعت است. مورچه‌ها و زنبورها بهترین نمونه‌هایی از رفتار جمعی با هدف تغذیه است. ثابت شده است که مورچه‌ها و زنبورها قادر به یافتن و علامت‌گذاری کوتاه‌ترین مسیر لانه / کندو به منبع غذا هستند [5]. آنها هوشمندانه غذاها را جستجو می‌کنند و مسیر استفاده از فرومون را برای اطلاع دادن و هدایت دیگران نشان می‌دهند.

بسیار جالب است که موجودات موقعیت مطلوب را پیدا می‌کنند و وظایف را به صورت موثر در گروه‌ها انجام می‌دهند. واضح و روشن است که آنها در طول این قرن‌ها تکامل پیدا کرده‌اند تا رفتارهای مطلوب و کارآمد را دریابند. بنابراین، کاملاً منطقی است که از آنها برای حل مسائلمان الهام می‌بخشیم. این همان هدف اصلی مطالعه است که هوش گروهی نام دارد (SI)، که توسط بنی و وانگ در سال 1989 ارائه شده است [6]. SI اشاره به اجرای مصنوعی / شبیه‌سازی هوش جمعی و اجتماعی گروهی از موجودات زنده در طبیعت دارد [7]. محققان در این زمینه سعی

کردند قوانین محلی را برای تعاملات بین افراد که به هوش اجتماعی هدایت شده‌اند، درک کنند. از آنجا که هیچ واحد کنترل متمرکزی برای هدایت افراد وجود ندارد، یافتن قوانین ساده میان برخی از آنها می‌تواند رفتار اجتماعی کل جمعیت را شبیه‌سازی کند.

الگوریتم بهینه‌سازی کلونی مورچه (ACO) یکی از اولین تکنیک‌های SI است که اطلاعات اجتماعی مورچه‌ها را در هنگام پرورش یک کلونی مورچه تقلید می‌کند [8،9]. این الگوریتم از واقعیت ساده الهام گرفته است که هر مورچه می‌تواند مسیر خود را به سمت منابع غذایی خارج از لانه توسط فرومون نشان دهد. زمانی که مورچه یک منبع غذایی را پیدا می‌کند، به لانه می‌رسد و مسیر فرومون را نشان می‌دهند تا دیگران مسیر را پیدا کنند. زمانی که مورچه‌های مورچه‌های دیگر علائم فرومون را می‌بینند مسیر را دنبال می‌کنند و فرومون‌های خود را ترک می‌کنند. کلید اصلی این است که آنها ممکن است راه‌های مختلفی را برای منابع غذایی داشته باشند. از آنجا که یک مسیر طول می‌کشد تا مورچه‌ها سفر کنند، با این وجود، فرومون با میزان بالاتر قبل از آنکه توسط مورچه‌های دیگر مشخص شوند تبخیر می‌شود. بنابراین، کوتاه‌ترین مسیر به سادگی با دنبال کردن مسیر با سطح قوی فرومون و رها کردن مسیرهای با سطح فرومون ضعیف به دست می‌آید. دورینگو اولین بار از این مفاهیم ساده الهام گرفته و الگوریتم شناخته شده ACO را ارائه کرد [10].

الگوریتم بهینه‌سازی ذرات (PSO) یکی دیگر از الگوهای شناخته شده SI است. فرآیند رفتاری این الگوریتم ناوبری گله‌های پرند را تقلید می‌کند و توسط ابرهات و کندی ارائه شده است [11]. الهام بخش اصلی قوانین ساده تعاملات بین پرندگان است: پرندگان تمایل دارند مسیر پرواز خود را در جهت مسیر فعلی خود، بهترین محل منبع غذایی را به دست آورند و بهترین محل غذا را که تاکنون یافت شده است را حفظ کنند [12]. الگوریتم PSO به سادگی این سه قاعده را تقلید می‌کند و ذرات را به سوی بهترین راه حل‌های بهینه توسط هر یک از افراد و به طور همزمان هدایت می‌کنند.

الگوریتم کلونی زنبور عسل (ABC) یکی دیگر از الگوریتم‌های مبتنی بر سیگنال جدید و محبوب است. این الگوریتم دوباره رفتار اجتماعی جوجه‌های زنبور عسل را در هنگام تغذیه شهد توسط کارابوگا ارائه کرده است [13]. تفاوت این

الگوریتم در مقایسه با ACO، و PSO تقسیم زنبورهای عسل دیده‌بانی، ناظر، و کارگر است [14]. زنبورهای کارگر مسئول یافتن منابع غذایی و اطلاع رسانی به دیگران با یک رقص ویژه هستند. علاوه بر این، تماشاگران رقص را تماشا می‌کنند، و یکی از آنها را انتخاب می‌کند و مسیر را به سمت منابع غذایی انتخاب و آن را دنبال می‌کنند. زنبورهای دیده‌بانی منابع غذایی رها شده را کشف می‌کنند و آنها را جایگزین منابع جدید می‌کنند. از آنجا که تعداد قابل توجهی از محققان تلاش کردند این الگوریتم‌ها را در زمینه‌های گوناگون مورد استفاده قرار دهند مسائل مربوط به آنها حل شده است [15-20]. کاربرد موفقیت آمیز این الگوریتم‌ها در علم و صنعت نشان دهنده شایستگی‌های تکنیک مبتنی بر SI در عمل است. دلایل مربوط به استفاده از مزایای الگوریتم‌های مبتنی بر SI عبارتند از: اولاً، تکنیک‌های مبتنی بر SI در طول مراحل تکرار در فضای جستجو به صورت اطلاعاتی ذخیره می‌شوند، در حالی که این اطلاعات توسط الگوریتم‌های تکاملی (EA) نسل به دور ریخته می‌شوند. ثانیاً، پارامترهای کنترل در الگوریتم مبتنی بر SI کمتر وجود دارند. سوماً، الگوریتم SI مبتنی بر اپراتورها در مقایسه با الگوریتم EA مجهز هستند. سرانجام، تکنیک‌های مبتنی بر SI از انعطاف پذیری سود می‌برند، و به راحتی برای مسائل در زمینه‌های مختلف به کار می‌روند.

علی‌رغم تعداد قابل توجهی از نشریات اخیر در این زمینه [21-29]، در طبیعت هنوز رفتارهای پرخاشگری که مورد توجه قرار نگرفته‌اند وجود دارد. یکی نوع حشرات متفن وجود دارد که به ندرت رنج می‌برند. از آنجایی که ادبیات مطالعاتی برای شبیه‌سازی هوش فردی و اجتماعی سنجاقک‌ها وجود ندارد، این مقاله برای اولین بار ابتدا ویژگی‌های اصلی سنجاقک را پیدا می‌کند. سپس یک الگوریتم براساس مشخصات ارائه می‌شود. قضیه ناهارخوری رایگان (NFL) [30] از انگیزه این کار برای ارائه بهینه‌ساز استفاده می‌کند، به این دلیل که این الگوریتم ممکن است الگوریتم‌های دیگری را برای برخی از مسائل نتواند حل کند. ادامه مقاله به شرح زیر است:

بخش 2 الهام و پایه‌های بیولوژیکی مقاله را ارائه می‌دهد. مدل‌های ریاضی و الگوریتم DA در بخش 3 ارائه شده‌اند. همچنین این بخش نسخه‌های باینری و چند هدفه DA را ارائه می‌کند. یک مطالعه جامع تطبیقی در چندین معیار

توابع و یک مطالعه موردی واقعی در بخش 4 برای تایید عملکرد الگوریتم‌های DA، BDA و MODA ارائه شده است. سرانجام، بخش 5 نتیجه کار است و برخی از دستورالعمل‌ها برای مطالعات آینده را نشان می‌دهد.

2 الهام بخش

سنجاقک‌ها (Odonata) حشرات تفننی هستند. حدود 3000 گونه مختلف این حشره در سراسر جهان وجود دارد [31]. همانطور که در شکل 1 نشان داده شده است، چرخه حیات سنجاقک شامل دو مرحله اصلی است: نیچه و بالغ. آنها بخش عمده‌ای از طول عمر خود را در نیچه می‌گذرانند و دگرگون می‌شوند تا بزرگسال شوند [31].



شکل 1. a. سنجاقک واقعی، b چرخه زندگی سنجاقک‌ها (تصویر چپ حسن نیت مهرداد مانی است که از سایت

www.mehrdadmomeny.com قابل دسترس است).

سنجاقک‌ها به عنوان شکارچیان کوچک در نظر گرفته می‌شوند و تقریباً تمام حشرات کوچک طبیعت را شکار می‌کنند. نیچه سنجاقک‌ها روی سایر حشرات دریایی و حتی ماهی‌های کوچک هم وجود دارد. واقعیت جالب در مورد سنجاقک‌ها این است که آنها منحصر به فرد و رفتار گروهی نادری دارند. سنجاقک‌ها فقط برای دو هدف اهمیت دارند: شکار و مهاجرت. در ابتدا ازدحام ایستا نام داشتند (تغذیه)، و بعداً ازدحام پویا نامگذاری شده‌اند (مهاجرت). در ازدحام ایستا، سنجاقک‌ها گروه‌های کوچکی را تشکیل می‌دهند و در یک منطقه کوچک پرواز می‌کنند تا سایر شکارچیان پرواز به عنوان مثال پروانه‌ها و پشه‌ها را شکار کنند [32]. جنبش‌های محلی و تغییرات ناگهانی در مسیر پرواز، ویژگی‌های اصلی ازدحام ایستا است. با این حال، در ازدحام پویا، شمار زیادی از سنجاقک‌ها برای مهاجرت در یک مسیر در مسافت‌های طولانی می‌چرخند [33].

الگوریتم اصلی DA از رفتارهای ازدحام ایستا و پویا حاصل می‌شوند. این دو رفتار روحیه بسیار شبیه به دو مرحله اصلی بهینه‌سازی با استفاده از فراشناختی دارد: اکتشاف و بهره‌برداری. سنجاک‌ها زیرگروهی را در بیش از مناطق مختلف به صورت ایستا که هدف اصلی آن فاز اکتشاف است را ایجاد می‌کنند. با این حال، در ازدحام ایستا، دره‌های بزرگ و در امتداد یک جهت، که در مرحله بهره‌برداری مطلوب است، پرواز می‌کنند. این دو مرحله به صورت ریاضی در بخش زیر اجرا می‌شوند.

3 الگوریتم سنجاک

3.1 اپراتورهایی برای اکتشاف و بهره‌برداری

بر طبق نظر رینولدز، رفتار ازدحام به سه اصول ابتدایی زیر اشاره دارند [34]:

- جداسازی، که به اجتناب از برخورد ایستا افراد به همسایگی افراد دیگر اشاره دارد.
- هماهنگی، که تطبیق سرعت افراد در همسایگی افراد دیگر را نشان می‌دهد.
- انسجام، که به گرایش افراد نسبت به مرکز همسایگی جرم اشاره دارد.

هدف اصلی هر گونه ازدحام زنده ماندن است، بنابراین تمام افراد باید منابع غذایی و دشمنان بی‌نظیر را جذب کنند. با توجه به این دو رفتار، پنج عامل اصلی در ارتقاء موقعیت افراد در سموم وجود دارد که در شکل 2 نشان داده شده است.

هر یک از این رفتارها به صورت ریاضی به شرح زیر هستند:

عمل جداسازی، با استفاده از فرمول زیر محاسبه می‌شود [34]:

$$(3.1) S_i = - \sum_{j=1}^N X - X_j$$

X موقعیت فعلی فرد است، X_j نشان دهنده موقعیت j ام در همسایگی فرد است، و N تعداد افراد همسایه است.

ترازبندی به صورت زیر محاسبه می‌شود:

$$(3.2) A_i = \frac{\sum_{j=1}^N V_j}{N}$$

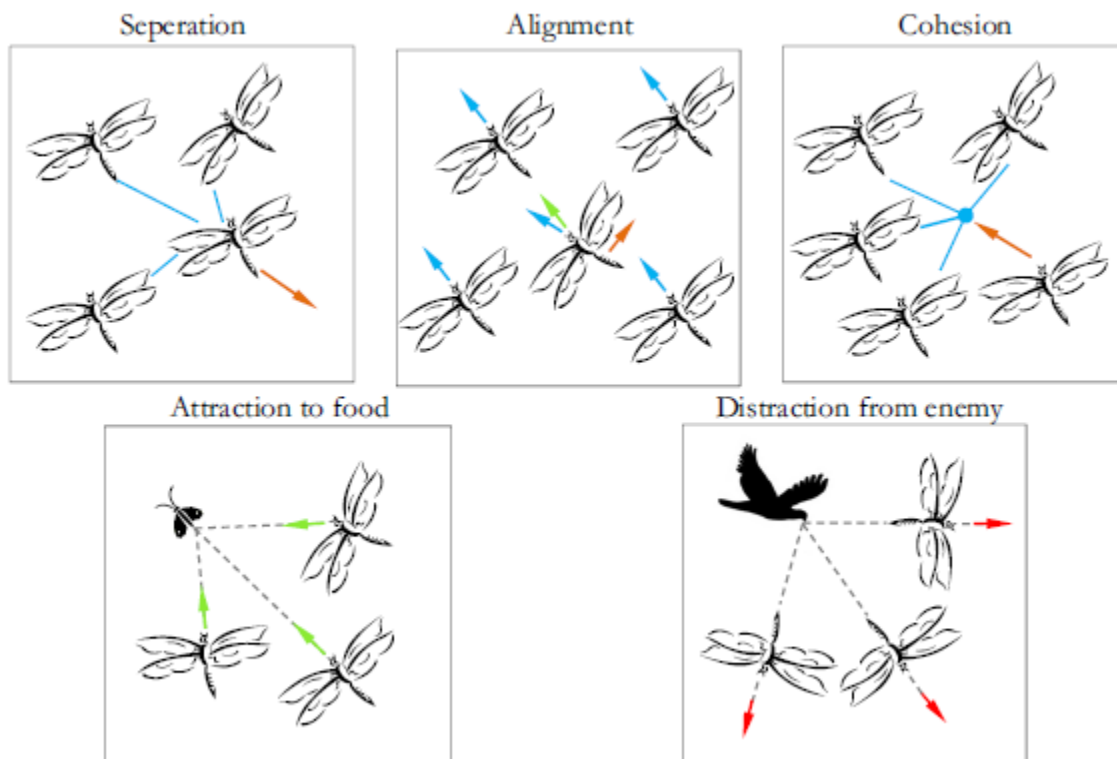
X_j ، j ام سرعت یک فرد همجوار را نشان می‌دهد.

انسجام به صورت زیر محاسبه می‌شود:

$$(3.3) C_i = \frac{\sum_{j=1}^N X_j}{N} - X$$

X موقعیت فعلی فرد است، N تعداد همسایه‌ها است، و X_j ، ژام موقعیت همسایه فرد را نشان می‌دهد.

جذابیت نسبت به منابع غذایی به صورت زیر محاسبه می‌شود:



شکل 2. الگوهای اصلاحی اولیه بین گروهی از افراد

$$(3.4) F_i = X^+ - X$$

X موقعیت فعلی فرد است، و X^+ موقعیت منبع غذایی را نشان می‌دهد.

دور زدن دفاعی بیرون دشمن به صورت زیر محاسبه می‌شود:

$$(3.5) E_i = X^- + X$$

X موقعیت فعلی فرد است، و X^- موقعیت دشمن را نشان می‌دهد.

رفتار سنجاك‌ها در این مقاله به نظر می‌رسد که ترکیبی از پنج اصلاح الگو است. برای به روز رسانی موقعیت سنجاك‌های مصنوعی در فضای جستجو و شبیه‌سازی حرکت آنها، دو بردار در نظر گرفته می‌شود: مرحله (DX) و موقعیت (X). مرحله بردار به طور مشابه‌ای بردار سرعت PSO و الگوریتم DA براساس چارچوب الگوریتم PSO توسعه داده است. مرحله بردار حرکت سنجاك‌ها را نشان می‌دهد و به صورت زیر تعریف می‌کند (توجه داشته باشید که مدل به روز رسانی موضع سنجاك مصنوعی را در یک بعد تعریف می‌کند، اما روش دیگری را می‌تواند برای گسترش در ابعاد بالاتر معرفی کند):

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (3.6)$$

S وزن جداسازی را نشان می‌دهد، S_i ، A_i وزن جداسازی را نشان می‌دهد، a تراز وزن است، A تراز A_i فرد را نشان می‌دهد، C وزن انسجام را نشان می‌دهد، C_i انسجام A_i فرد است، f عامل غذا است، F_i ، A_i منبع غذایی فرد است، e عامل دشمن است، E_i ، A_i موقعیت فرد دشمن است، w وزن اینرسی است، و t مقیاس تکرار است. پس از محاسبه مرحله بردار، موقعیت بردارها به صورت زیر محاسبه می‌شوند:

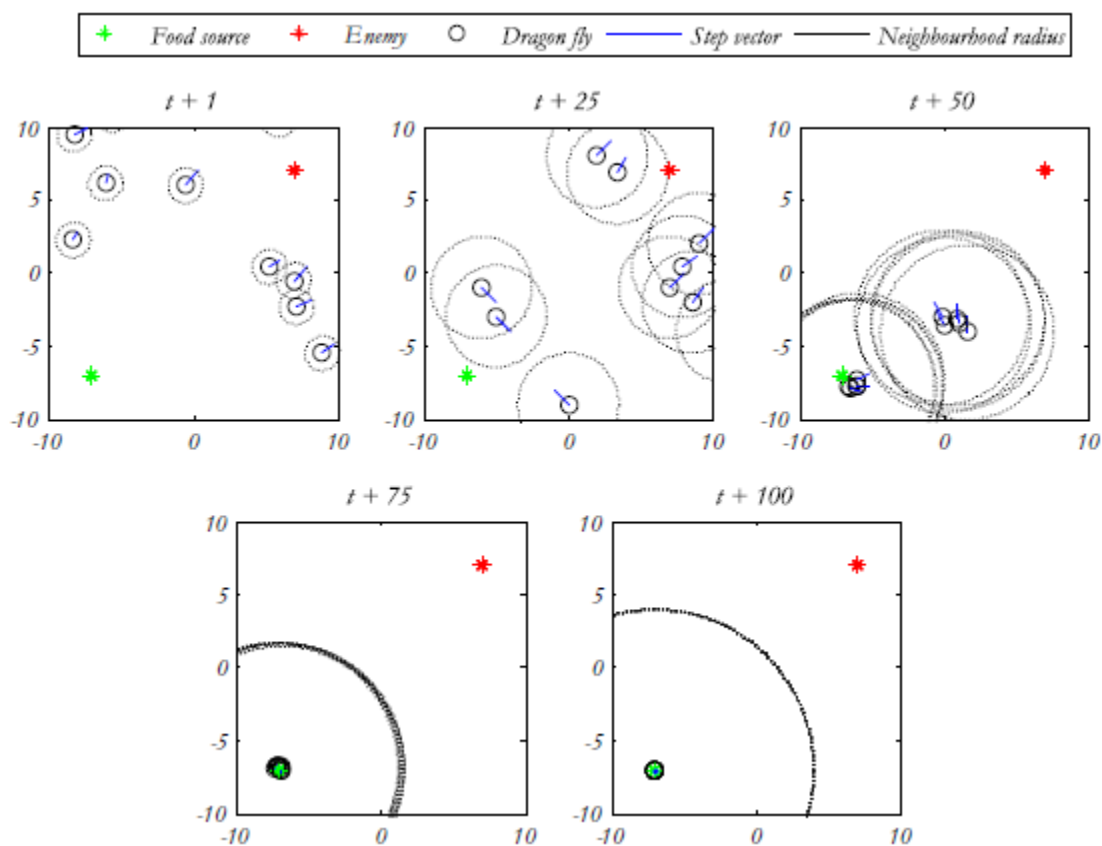
$$(3.7) X_{t+1} = X_t + \Delta X_{t+1}$$

که t تکرار فعلی است.

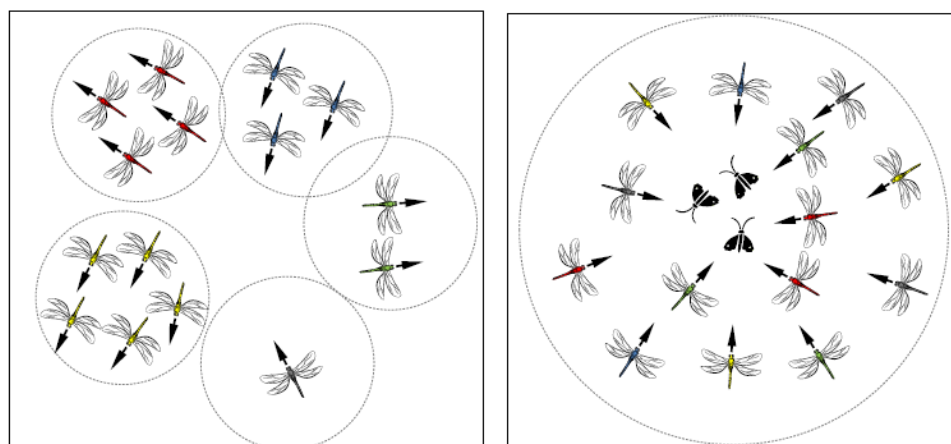
با تجزیه، هم‌ترازی، انسجام، مواد غذایی، و عوامل دشمن (s, a, c, e)، رفتارهای مختلف کاوشگر و استعمارگر می‌توانند در طول بهینه‌سازی به دست آیند. همسایگان سنجاك‌ها بسیار مهم هستند، بنابراین همسایگی (دایره در 2 بعد، کره در یک فضای سه بعدی، یا ابرکره در یک فضای n بعدی) با شعاع خاصی فرض می‌شوند که در اطراف هر یک از سنجاك‌های مصنوعی هستند. یک مثال از رفتار ازدحام سنجاك‌ها با افزایش شعاع همسایگی با استفاده از مدل ریاضی در شکل 3 نشان داده شده است.

همانطور که در شکل زیر دیده می‌شود، سنجاك‌ها فقط دو نوع ازدحام را نشان می‌دهند: ایستا و پویا که در شکل 4 نشان داده شده است. همانطور که در این شکل دیده می‌شود، سنجاك‌ها تمایل دارند پرواز خود را هماهنگ کنند در حالی که جدایی و انسجام پویا را در یک ازدحام حفظ می‌کنند. با این حال، در ازدحام ایستا، هم سطحی بسیار کم است در حالی انسداد برای حمله به شکار بسیار بالا است. بنابراین، در هنگام کاوش فضای جستجو و هم‌ترازی

پایین و انسجام بالا در زمان بهره‌برداری از فضای جستجو ما کاوش فضا را با مقیاس بالا و کم بودن وزن را تخصیص می‌دهیم. برای انتقال بین اکتشاف و بهره‌برداری، شعاع همسایه‌ها با تعداد تکرارها بسیار مناسب است. راه دیگر برای تعادل اکتشاف و استثمار این است که در هنگام بهینه‌سازی، عوامل ازدحام (s,a,c,f,e,w) به صورت سازگار تنظیم شوند.



شکل 3. رفتار ازدحامی پرواز سنجاقک‌های مصنوعی ($w=0.9-0.2$, $s=0.1$, $a=0.1$, $c=0.7$, $f=1$, $e=1$)



شکل 4. ازدحام سنجاقک‌های ایستا در مقابل پویا

در اینجا سوال می‌تواند در مورد نحوه همگرایی سنجاقک‌ها در زمان تضمین بهینه‌سازی باشد. سنجاقک‌ها نیاز به تغییر وزن به صورت سازگار برای انتقال از فضای جستجوی اکتشاف به فضای جستجو بهره‌برداری دارند. همچنین فرض می‌شود که سنجاقک‌ها تمایل بیشتری برای دیدن و تنظیم مسیر پرواز به عنوان یک فرآیند بهینه‌سازی دارند. به عبارت دیگر، منطقه همسایگی افزایش پیدا می‌کند و بدین ترتیب در مرحله نهایی بهینه‌سازی به یک گروه تبدیل می‌شود تا به بهترین وجه به گروه سراسری تبدیل شود. منبع غذایی و دشمن از بهترین و بدترین راه حل‌ها انتخاب می‌شوند تا کل راه‌حل‌های بهینه را کشف کند. این باعث همگرایی مناطق امید بخش فضای جستجو و واپاشی مناطق بیرونی فضای جستجو می‌شود.

برای بهبود تصادفی، رفتار تصادفی و اکتشاف سنجاقک‌های مصنوعی، آنها زمانی که هیچ راه حل برای همسایه وجود ندارد باید در اطراف فضای جستجو با استفاده از پرواز تصادفی (پرواز لووی) پرواز کنند. در این حالت موقعیت سنجاقک‌ها با استفاده از معادله زیر به روز رسانی می‌شود:

$$(3.8) X_{t+1} = X_t + Lévy(d) * X_t$$

t تکرار فعلی است، و d ابعاد بردارهای موقعیت است.

پرواز Le'vy به صورت زیر محاسبه می‌شود [35]:

$$(3.9) Lévy(x) = 0.01 * \frac{r_a * \sigma}{|r_2|^{\frac{1}{\beta}}}$$

که r_1 و r_2 دو عدد تصادفی در بازه [1,0] است، و b یک ثابت است (و برابر با 1.5 است) و r به صورت زیر محاسبه می‌شود:

$$(3.10) \sigma = \left(\frac{\Gamma(1+\beta) * \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) * \beta * 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}}$$

$$\Gamma(x) = (x-1)!$$

3.2 مسائل تک هدفه الگوریتم DA

الگوریتم DA با ایجاد مجموعه‌ای از راه حل‌های تصادفی برای یک مسئله بهینه‌سازی، شروع به بهینه‌سازی می‌کند. در واقع، موقعیت و بردارهای مرحله‌ای سنجاقک‌ها با مقادیر تعریف شده در مرزهای پایین و بالای متغیرها تنظیم می‌شوند. در هر تکرار، موقعیت و مرحله هر نوع سنجاقک با استفاده از معادلات به روز رسانی (3.7) / (3.8) و (3.6) به روز رسانی می‌شود. برای به روز رسانی بردارهای X و DX ، محدوده هر نوع سنجاقک با محاسبه فاصله اقلیدسی بین تمام سنجاقک‌ها و انتخاب N انتخاب می‌شود. فرآیند به روز رسانی موقعیت به طور تکراری ادامه پیدا می‌کند تا معیار آخر مورد رضایت باشد. شبه کد الگوریتم DA در شکل 5 ارائه شده است.

```

Initialize the dragonflies population  $X_i$  ( $i = 1, 2, \dots, n$ )
Initialize step vectors  $\Delta X_i$  ( $i = 1, 2, \dots, n$ )
while the end condition is not satisfied
    Calculate the objective values of all dragonflies
    Update the food source and enemy
    Update  $w, s, a, c, f$ , and  $e$ 
    Calculate  $S, A, C, F$ , and  $E$  using Eqs. (3.1) to (3.5)
    Update neighbouring radius
    if a dragonfly has at least one neighbouring dragonfly
        Update velocity vector using Eq. (3.6)
        Update position vector using Eq. (3.7)
    else
        Update position vector using Eq. (3.8)
    end if
    Check and correct the new positions based on the
    boundaries of variables
end while

```

شکل 5. شبه کد الگوریتم DA

لازم به ذکر است که تفاوت اصلی بین الگوریتم DA و PSO در تفکر جداسازی، هم‌ترازی، انسجام، جاذبه، توزیع، و راه رفتن به صورت تصادفی است. اگرچه برخی از آثار در ادبیات وجود دارد اما تلاش برای ادغام به صورت یکپارچه، همبستگی و انسجام PSO [38-36]، در این مقاله رفتار ازدحام را به صورت ناگهانی با تمام عوامل امکان پذیر برای مدلسازی افراد را در نظر می‌گیرد. مفاهیم ازدحام پویا و ایستا کاملاً جدید هستند. مدل ارائه شده این کار کاملاً متفاوت از PSO بهبود یافته و در ادبیات فعلی در بالا ذکر شده است.

3.3 مسائل باینری الگوریتم DA (BDA)

بهینه‌سازی در یک فضای جستجوی باینری بسیار متفاوت از یک فضای پیوسته است. در فضای جستجوی پیوسته، عوامل جستجوی DA قادر به به روز کردن موقعیت خود با اضافه کردن بردارهای مرحله به بردار موقعیت هستند. در فضای جستجو باینری، موقعیت عامل‌های جستجو را نمی‌توان با اضافه کردن بردارهای مرحله به X به روز کرد به این دلیل که بردارهای موقعیت عامل‌های جستجو فقط می‌توانند با مقادیر 0 یا 1 مشخص شوند. با توجه به شباهت DA و دیگر تکنیک‌های SI، روش‌های فعلی برای حل مسائل باینری در ادبیات و برای این الگوریتم به راحتی قابل استفاده هستند.

با توجه به میرجیلی و لوئیس [39]، ساده‌ترین و موثرترین روش برای تبدیل یک تکنیک مداوم SI به یک الگوریتم باینری بدون تغییر ساختار استفاده از یک تابع انتقال است. سرعت توابع انتقال (مرحله) را به عنوان ورودی دریافت می‌کند و یک عدد را در بازه $[0,1]$ باز می‌گرداند، و تغییر موقعیت‌ها را به صورت احتمالی تعیین می‌کند. خروجی چنین توابعی به طور مستقیم متناسب با مقادیر بردار سرعت هستند. بنابراین، مقادیر بزرگ برای سرعت عامل جستجو، به احتمال زیاد موقعیت خودشان را به روز رسانی می‌کنند. این روش تغییرات ناگهانی ذرات را با مقادیر بزرگ سرعت به طور مشابه‌ای بهینه‌سازی و شبیه‌سازی می‌کند (شکل 6). دو مثال از توابع انتقال در ادبیات شکل 6 نشان داده شده است [39-41].

همانطور که در این شکل دیده می‌شود، دو نوع توابع انتقالی وجود دارد که عبارتند از: S شکل در مقابل V شکل. به گفته صارمی و همکارانش [40] عملکردهای انتقال V شکل بهتر از توابع S شکل هستند به این دلیل که آنها مقادیر ذرات 0 یا 1 را تحمیل نمی‌کنند. برای حل مسائل باینری با الگوریتم BDA، تابع انتقالی زیر مورد استفاده قرار می‌گیرد [39]:

$$(3.11) T(\Delta x) = \left| \frac{\Delta x}{\sqrt{\Delta x^2 + 1}} \right|$$

این توابع انتقال در ابتدا برای محاسبه احتمال تغییر موقعیت تمام سنجاقک‌های مصنوعی مورد استفاده قرار می‌گرفته است. سپس فرمول موقعیت جدید زیر به روز رسانی می‌شود و برای موقعیت عوامل جستجو در فضاهاى جستجو باینری مورد استفاده قرار می‌گیرد، به طوری که داریم:

$$(3.12) X_{t+1} = \begin{cases} \sim X_t & r < T(\Delta x_t + 1) \\ X_t & r \geq T(\Delta x_t + 1) \end{cases}$$

References

1. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
2. Muro C, Escobedo R, Spector L, Coppinger R (2011) Wolf-pack (*Canis lupus*) hunting strategies emerge from simple rules in computational simulations. *Behav Process* 88:192–197
3. Jakobsen PJ, Birkeland K, Johnsen GH (1994) Swarm location in zooplankton as an anti-predator defence mechanism. *Anim Behav* 47:175–178
4. Higdon J, Corrsin S (1978) Induced drag of a bird flock. *Am Nat* 112(986):727–744
5. Goss S, Aron S, Deneubourg J-L, Pasteels JM (1989) Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76:579–581
6. Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: Dario P, Sandini G, Aebischer P (eds) *Robots and biological systems: towards a new bionics? NATO ASI series*, vol 102. Springer, Berlin, Heidelberg, pp 703–712
7. Bonabeau E, Dorigo M, Theraulaz G (1999) *Swarm intelligence: from natural to artificial systems*. Oxford University Press, Oxford
8. Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover F, Kochenberger GA (eds) *Handbook of metaheuristics*. International series in operations research & management science, vol 57. Springer, USA, pp 250–285
9. Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *Syst Man Cybern Part B Cybern IEEE Trans* 26:29–41
10. Colomi A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: *Proceedings of the first European conference on artificial life*, pp 134–142
11. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the sixth international symposium on micro machine and human science*, pp 39–43
30. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *Evolut Comput IEEE Trans* 1(1):67–82
31. Thorp JH, Rogers DC (2014) *Thorp and Covich's freshwater invertebrates: ecology and general biology*. Elsevier, Amsterdam
32. Wikelski M, Moskowitz D, Adelman JS, Cochran J, Wilcove DS, May ML (2006) Simple rules guide dragonfly migration. *Biol Lett* 2:325–329
33. Russell RW, May ML, Soltesz KL, Fitzpatrick JW (1998) Massive swarm migrations of dragonflies (Odonata) in eastern North America. *Am Midl Nat* 140:325–342
34. Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. *ACM SIGGRAPH Comput Gr* 21:25–34
35. Yang X-S (2010) *Nature-inspired metaheuristic algorithms*, 2nd edn. Luniver Press
36. Cui Z, Shi Z (2009) Boid particle swarm optimisation. *Int J Innov Comput Appl* 2:77–85

12. Eberhart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In: Proceedings of the 2001 congress on evolutionary computation, pp 81–86
13. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. In: Technical report-tr06, Erciyes university, engineering faculty, computer engineering department
14. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Optim* 39:459–471
15. AlRashidi MR, El-Hawary ME (2009) A survey of particle swarm optimization applications in electric power systems. *Evolut Comput IEEE Trans* 13:913–918
16. Wei Y, Qiqiang L (2004) Survey on particle swarm optimization algorithm. *Eng Sci* 5:87–94
17. Chandra Mohan B, Baskaran R (2012) A survey: ant colony optimization based recent research and implementation on several engineering domain. *Expert Syst Appl* 39:4618–4627
18. Dorigo M, Stützle T (2010) Ant colony optimization: overview and recent advances. In: Gendreau M, Potvin J-Y (eds) *Handbook of metaheuristics*. International series in operations research & management science, vol 146. Springer, USA, pp 227–263
19. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42:21–57
20. Sonmez M (2011) Artificial Bee Colony algorithm for optimization of truss structures. *Appl Soft Comput* 11:2406–2418
21. Wang G, Guo L, Wang H, Duan H, Liu L, Li J (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. *Neural Comput Appl* 24:853–871
22. Wang G-G, Gandomi AH, Alavi AH (2014) Stud krill herd algorithm. *Neurocomputing* 128:363–370
23. Wang G-G, Gandomi AH, Alavi AH (2014) An effective krill herd algorithm with migration operator in biogeography-based optimization. *Appl Math Model* 38:2454–2462
24. Wang G-G, Gandomi AH, Alavi AH, Hao G-S (2014) Hybrid krill herd algorithm with differential evolution for global numerical optimization. *Neural Comput Appl* 25:297–308
25. Wang G-G, Gandomi AH, Zhao X, Chu HCE (2014) Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. *Soft Comput*. doi:[10.1007/s00500-014-1502-7](https://doi.org/10.1007/s00500-014-1502-7)
26. Wang G-G, Guo L, Gandomi AH, Hao G-S, Wang H (2014) Chaotic krill herd algorithm. *Inf Sci* 274:17–34
27. Wang G-G, Lu M, Dong Y-Q, Zhao X-J (2015) Self-adaptive extreme learning machine. *Neural Comput Appl*. doi:[10.1007/s00521-015-1874-3](https://doi.org/10.1007/s00521-015-1874-3)
28. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
29. Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl*. doi:[10.1007/s00521-015-1870-7](https://doi.org/10.1007/s00521-015-1870-7)
30. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: *Neural networks, 1995. Proceedings, IEEE International conference on*, pp 1942–1948
31. John H (1992) *Holland, adaptation in natural and artificial systems*. MIT Press, Cambridge
32. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evolut Comput* 1:3–18
33. van den Bergh F, Engelbrecht A (2006) A study of particle swarm optimization particle trajectories. *Inf Sci* 176:937–971
34. J. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: *Systems, man, and cybernetics, 1997. computational cybernetics and simulation, 1997 IEEE international conference on*, pp 4104–4108
35. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2010) BGSA: binary gravitational search algorithm. *Nat Comput* 9:727–745
36. Kadrovach BA, Lamont GB (2002) A particle swarm model for swarm-based networked sensor systems. In: *Proceedings of the 2002 ACM symposium on applied computing*, pp 918–924
37. Cui Z (2009) Alignment particle swarm optimization. In: *Cognitive informatics, 2009. ICCI'09. 8th IEEE international conference on*, pp 497–501
38. Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evolut Comput* 9:1–14
39. Saremi S, Mirjalili S, Lewis A (2014) How important is a transfer function in discrete heuristic algorithms. *Neural Comput Appl*:1–16
40. Mirjalili S, Wang G-G, Coelho LDS (2014) Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Comput Appl* 25:1423–1435
41. Mirjalili S, Lewis A (2015) Novel performance metrics for robust multi-objective optimization algorithms. *Swarm Evolut Comput* 21:1–23
42. Coello CAC (2009) Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored. *Front Comput Sci China* 3:18–30
43. Ngatchou P, Zarei A, El-Sharkawi M (2005) Pareto multi objective optimization. In: *Intelligent systems application to power systems, 2005. Proceedings of the 13th international conference on*, pp 84–91
44. Branke J, Kaußler T, Schmeck H (2001) Guidance in evolutionary multi-objective optimization. *Adv Eng Softw* 32:499–50
45. Coello Coello CA, Lechuga MS (2002) MOPSO: A proposal for multiple objective particle swarm optimization. In: *Evolutionary computation, 2002. CEC'02. Proceedings of the 2002 congress on*, pp 1051–1056
46. Coello CAC, Pulido GT, Lechuga MS (2004) Handling multiple objectives with particle swarm optimization. *Evolut Comput IEEE Trans* 8:256–279
47. Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *Evolut Comput IEEE Trans* 3:82–102
48. Digalakis J, Margaritis K (2001) On benchmarking functions for genetic algorithms. *Int J Comput Mathematics* 77:481–506
49. Molga M, Smutnicki C (2005) Test functions for optimization needs. Test functions for optimization needs. <http://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf>
50. Yang X-S (2010) Test problems in optimization. *arXiv preprint arXiv:1008.0549*
51. Liang J, Suganthan P, Deb K (2005) Novel composition test functions for numerical global optimization. In: *Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pp 68–75
52. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A et al (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. In: *KanGAL Report*, vol 2005005
53. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8:173–195
54. Sierra MR, Coello Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance. In: Coello Coello CA, Hernández Aguirre A, Zitzler E (eds) *Evolutionary multi-criterion optimization. Lecture notes in computer science*, vol 3410. Springer, Berlin, Heidelberg, pp 505–519
55. Van Veldhuizen DA, Lamont GB (1998) Multiobjective evolutionary algorithm research: a history and analysis (Final Draft) TR-98-03
56. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolut Comput IEEE Trans* 6:182–197
57. Carlton J (2012) *Marine propellers and propulsion*. Butterworth-Heinemann, Oxford