

# الگوریتم کلونی زنبور عسل مصنوعی

فاطمه مسکینی<sup>۱</sup>، سید احمد عدالت پناه<sup>۲</sup>

<sup>۱</sup>دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، دانشکده برق و کامپیوتر، موسسه آموزش عالی آیندگان، تنکابن.

<sup>۲</sup>گروه مهندسی کامپیوتر، دانشکده مهندسی برق و کامپیوتر، موسسه آموزش عالی آیندگان، تنکابن.

## مقدمه

الگوریتم زنبور شامل گروهی مبتنی بر الگوریتم جستجو است که اولین بار در سال ۲۰۰۵ توسعه یافت، این الگوریتم شبیه سازی رفتار جستجوی غذای گروههای زنبور عسل است.

در نسخه ابتدایی این الگوریتم، الگوریتم نوعی از جستجوی محلی انجام میدهد که با جستجوی تصادفی (Random) ترکیب شده و میتواند برای بهینه سازی ترکیبی (زمانی که بخواهیم چند متغیر را همزمان بهینه کنیم) یا بهینه سازی تابعی به کار برود.

وجود ویژگی های هوش در زنبورها

(۱) خود سازماندهی

(۲) دنبال کردن منابع غذایی بهتر توسط زنبورهای بیشتر

(۳) ترک منابع غذایی متروک شده

(۴) جستجوی منبع غذایی بهتر

(۵) تقسیم کار

(۶) مشخص کردن جهت فاصله، کیفیت و کمیت منبع یافت شده

## الگوریتم کلونی زنبور عسل مصنوعی

نسخه دیگری از الگوریتم زنبور است. نسخه های مختلفی از این الگوریتم وجود دارد که از رفتار زنبورهای عسل الهام گرفته شده است. دو نسخه بسیار مهم آن: (۱) *Artificial Bee Colony* (۲) *Bees Algorithm* هر دو الگوریتم از یک الگو استفاده میکنند آن هم به این شکل است که در هر کندویی یک تعداد از زنبورها وظیفه پیدا کردن منابع غذایی را دارند که از نظر فاصله در حد معقولی هستند. و این اطلاعات را به بقیه افراد کندو اطلاع میدهند.

## انواع زنبورها

زنبورهای کارگر *Employed bees*

زنبورهای غیر کارگر *Unemployed bees*

زنبورهای ناظر *Onlooker bees*

زنبورهای پیشاهنگ *Scout bees*

زنبوری به نام زنبور پیشاهنگ به منابع غذایی رفته و آنجا را کشف میکند و به کندو برمیگردد و اطلاعات خود را به افراد کندو میدهد. این زنبورهای پیشاهنگ کارشان شناسایی منابع غذایی است. اگر کارشان را به حوزه بهینه سازی ببریم یک سری جواب اولیه هستند اما اینکه در آن منبع غذایی شناخته شده چقدر غذا موجود است و کجا منبع غذایی مرغوب و کجا نامرغوب است وظیفه مجموعه دیگری از زنبورها است که به آن زنبورهای مستخدم (کارگر) می گوئیم. اصطلاح مستخدم خود شاخه های مختلفی دارد یک عده از این زنبورها دقیقا براساس یک سازمانی که بر روی این کندوها حاکم است به این منابع غذایی ارسال میشوند و باید به مسیری که زنبور پیشاهنگ معرفی میکند بروند.

یک عده هم براساس اطلاعاتی که زنبور پیشاهنگ ارائه میکند خودشان تصمیم میگیرند و جذابیت منبع غذایی را درک میکنند و برحسب نیاز به آن منبع غذایی میروند و بیشتر از آنجا غذا تهیه میکنند و بر اساس مطلوبیتی که این منابع دارند به آنها زنبور اختصاص میدهند.

زنبورهای ناظر کارهایی از قبیل: (۱) محاسبه احتمالات (۲) انتخاب سایت (۳) تولید پاسخ جدید (۴) مقایسه را انجام میدهند.

## الگوی موجود در الگوریتم (ABC)

زنبور پیشاهنگ یک پاسخ اولیه را ارسال میکند یعنی یک محل ابتدایی برای سایت‌های غذا وجود دارد که یک عده زنبور به صورت مشخص و الگودار استخدام میشوند بعضی هم به صورت تصادفی و براساس شایستگی سایت‌ها ارسال میشوند.

چرا باید سایتی که شایستگی بیشتری دارد مورد توجه قرار گیرد؟ زیرا وقتی از یک سایت یک نمونه برمی‌داریم که بهتر از سایرین است یعنی یک چیز قابل توجه است و امیدواریم بتوانیم آنجا با سرمایه گذاری بیشتر منبع غذایی بهتر و غنی تری را کشف کنیم، دقیقا مثل فرایند بهینه سازی.

(۱) در بهینه سازی وقتی جوابی داریم که به نظر جواب مناسب است سعی میکنیم بیشتر روی آن جواب سرمایه گذاری کنیم. وقتی یک ایده جالب به ذهن ما میرسد سعی میکنیم آن را پرورش دهیم و نهایتا آن را تبدیل به یک پاسخ مناسبتر کنیم ایده ای که پشت قضیه است این است که از نظر قواعد بهینه سازی یک پاسخ مناسب را مناسبتر کنیم که به آن بهره برداری (**Exploitation**) میگوییم، یعنی توانایی پرورش پاسخهای فعلی برای رسیدن به پاسخهای بهتر.

(۲) **Exploration** یعنی توانایی تولید پاسخهای جدید و البته متفاوت یا آزمایش طرحی که تا به حال مشابه آن را نداشتیم. این مفهوم دقیقا مقابل مفهوم بهره برداری است. در الگوریتم ABC چرا به **Exploitation** نیاز داریم؟ برای اینکه مطمئن شویم همه جا را بررسی کرده‌ایم.

در الگوریتم ABC چرا به **Exploration** نیاز داریم؟ برای اینکه مطمئن شویم به اندازه کافی زمان و انرژی برای بهتر شدن راه حل صرف کرده ایم. ما در بهینه سازی به هردوی اینها نیاز داریم ولی نمیتوانیم به طور همزمان هر دو را داشته باشیم باید کاری کنیم که در ابتدای کار الگوریتم **Exploration** بهتری داشته باشیم و در ادامه کار برای اطمینان بیشتر **Exploitation** بهتری داشته باشیم.

## فرایندهای انتخاب شده در الگوریتم کلونی زنبور عسل مصنوعی

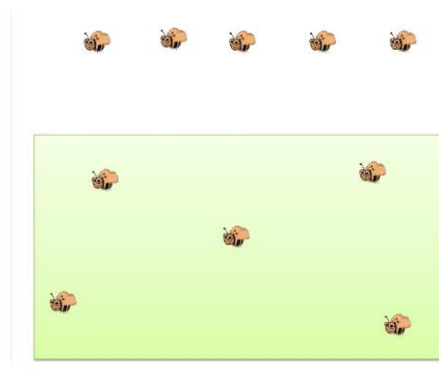
- ۱) فرایند انتخاب سراسری
- ۲) فرایند انتخاب محلی توسط زنبورهای کارگر
- ۳) فرایند انتخاب محلی توسط زنبورهای ناظر
- ۴) فرایند انتخاب تصادفی توسط زنبورهای ناظر
- ۵) فرایند انتخاب تصادفی توسط زنبورهای پیشاهنگ

### مقدار دهی اولیه به منابع غذایی

ابتدا منابع غذایی یا پاسخ های اولیه مساله بصورت تصادفی مقدار دهی اولیه میشوند

$$x_{ij} = x_j^{\min} + \text{rand}(0, 1) (x_j^{\max} - x_j^{\min})$$

### دسته اول : زنبورهای کارگر



هر زنبور کارگر بطور تصادفی یک همسایه انتخاب میکند و از طریق فرمول زیر به سمت آن حرکت میکند.

$$V_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

$$v_{i,j} = x_{i,j} + \phi(x_{i,j} - x_{k,j}) \quad i, k \in \{1, 2, \dots, N\}, \\ j \in \{1, 2, \dots, D\} \text{ and } i \neq k \quad (3)$$

## نکته مهم

اگر موقعیت جدید ( منطقه غذایی جدید) کیفیت بهتری داشت زنبور در منطقه جدید باقی میماند در غیر این صورت به منطق قبلی خود باز میگردد و یک واحد به شاخص محاکمه این زنبور اضافه میشود.

## شاخص محاکمه

این شاخص شمارنده تعداد دفعات متوالی حرکت زنبور با عدم بهبود است.

اگر زنبوری مقدار شاخص محاکمش از حد معین شده بیشتر بشود به این معنی که آن منبع غذایی دیگر منبع مناسبی نیست و باید آنجا را ترک کرد.

## دسته دوم: زنبورهای ناظر



## حرکت زنبورهای ناظر

حرکت زنبورهای ناظر به منابع غذایی با احتمال محاسبه شده از طریق چرخه رولت با استفاده از معادلات زیر و تعیین محله های جدید:

$$P_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j}$$

$$fitness_i = \begin{cases} \frac{1}{1+fit_i}, & fit_i \geq 0 \\ 1 + abs(fit_i), & fit_i < 0 \end{cases}$$

### دسته سوم: زنبورهای پیشاهنگ

زنبورهای پیشاهنگ آن ناحیه که از نظر منبع غذایی نامطلوب شناسایی شدند، ترک و به تصادف ناحیه های دیگری را انتخاب می کنند. در صورتی که یک منبع غذایی بهتر پس از رسیدن شاخص محاکمه به حد تعیین شده یافت نشود، منبع غذایی جدیدی توسط زنبورهای پیشاهنگ به صورت تصادفی با استفاده از معادله زیر مقداردهی می گردد

$$x_{ij} = x_j^{min} + rand(0, 1)(x_j^{max} - x_j^{min})$$

### شرط جایگزینی تابع راه حل بهتر بر اساس مقدار تابع هدف

$$Fitness(f(x)) = \begin{cases} \frac{1}{1+f(x)} & \text{if } f(x) \geq 0 \\ 1 + |f(x)| & \text{if } f(x) < 0 \end{cases}$$

we can see that when  $f(x)$  is larger than the zero but has a very small value, e.g.  $1E-20$ , the fitness value of equation  $1/(1 + 1E-20)$  is rounded up to be 1 ( $1E-20$  is ignored). This will lead the fitness of all solutions to become equal to 1 in the later iterations. In other words, there is no difference between the fitness values that is equal to  $1/(1 + 1E-20)$  and  $1/(1 + 1E-120)$ . Thus, a new solution that gives a better fitness value than the old solution will be ignored and the solution will stagnate at the old solution.