

درس ۱۸:

الگوریتم فراابتکاری الگوریتم اجتماع مورچگان

تهیه شده توسط گروه بهینه‌یاب



www.behinehyab.com

مقدمه

محاسبه راه حل بهینه یا *Optimal solution* برای اکثر مسایل بهینه‌سازی که در خیلی از زمینه‌های کاربردی و عملی مشاهده می‌گردند، کار دشوار و سختی است. در عمل، معمولاً به راه‌های خوب که از الگوریتم‌های هیوریستیک *Heuristic* یا متاهیوریستیک (همان فراابتکاری) *Metaheuristic* بدست می‌آید، اکتفا می‌گردد. متاهیوریستیک‌ها مجموعه‌ای از فنون بهینه‌سازی تقریبی *Approximate optimization techniques* را که عمدتاً در طول دو دهه گذشته شهرت پیدا کرده‌اند، در بر می‌گیرند. روش‌های فراابتکاری راه‌های قابل قبول در زمان معقول را برای مسایل پیچیده و سخت، در زمینه‌های مهندسی و علوم پایه ارائه می‌نمایند. برخلاف الگوریتم‌های بهینه‌سازی دقیق *Exact optimization algorithms*، فراابتکاری‌ها بهینه بودن جواب‌های بدست آمده را ضمانت نمی‌نمایند.

کلمه متاهیوریستیک از کلمه یونانی "*Heuriskein*" به معنای هنر کشف قواعد جدید برای حل مسایل گرفته شده است. پیشوند "متا" نیز از یک کلمه یونانی به معنای "متدولوژی" سطح بالا گرفته شده است. واژه "متاهیوریستیک" اولین بار توسط گلوور در سال ۱۹۸۶ ارائه گردید. روش جستجوی متاهیوریستیک را می‌توان به صورت متدولوژی‌های عمومی سطح بالایی که می‌توانند به عنوان یک استراتژی راهنما در طراحی هیوریستیک‌های اختصاصی برای حل مسایل بهینه‌سازی تخصصی به کار روند، تعریف کرد.

برخلاف روش‌های دقیق، متاهیوریستیک‌ها (فراابتکاری‌ها) برای مسایل با اندازه‌های بزرگ، کاربرد دشوار دارد و راه‌هایی راضی‌کننده‌ای در زمان معقولی ارائه می‌نمایند. در این الگوریتم‌ها، هیچ‌گونه ضمانتی برای یافتن جواب بهینه‌سراسری یا حدودی از آن وجود ندارد. متاهیوریستیک‌ها (فراابتکاری‌ها) در طول بیست سال گذشته شهرت زیادی پیدا کرده‌اند. کاربرد و استفاده از آن‌ها در خیلی از مسایل، کارایی و اثر بخشی آن‌ها را برای حل مسایل پیچیده و بزرگ نشان می‌دهد. فراابتکاری‌ها در خیلی از زمینه‌ها از قبیل موارد زیر کاربرد دارند:

✓ طراحی مهندسی، بهینه‌سازی توپولوژی، بهینه‌سازی مسایل الکترونیک، آئرونامیک، دینامیک

سیالات، مخابرات، رباتیک

- ✓ یادگیری ماشین و کاوش داده‌ها
- ✓ مدل سازی سیستم‌ها، شبیه سازی و تحقیق در شیمی، فیزیک، بیولوژی، کنترل، سیگنال، و پردازش تصویر
- ✓ مسایل مسیره‌دهی و برنامه‌ریزی، برنامه‌ریزی ربات، مسایل تولید و زمان بندی، حمل و نقل و لجستیک، مدیریت زنجیره تامین
- روش‌های مختلف الگوریتم‌های فراابتکاری یا متاهیوریستیک تا به حال پیشنهاد شده است که به صورت زیر هستند:

- ✓ بهینه‌سازی کلونی مورچگان *Ant colony optimization*
- ✓ بهینه‌سازی کلونی زنبوران *Bee colony*
- ✓ الگوریتم‌های ترتیبی *cultural algorithms*
- ✓ الگوریتم‌های با هم تکاملی *Co-evolutionary algorithms*
- ✓ الگوریتم ژنتیک *Genetic algorithm*
- ✓ جستجوی محلی تکراری *Iterated local search*
- ✓ بهینه‌سازی توده ذرات *particle swarm intelligent*
- ✓ انجماد تدریجی *Simulated Annealing*
- ✓ جستجوی ممنوع *Taboo search*
- ✓ جستجوی همسایگی متغیر *Variable neighbor search*

در طراحی یک فراابتکاری، دو معیار متناقض شامل **کاوش** *Exploration* در فضای جستجو (گوناگونی و تنوع) و **تبعیت** (*Exploitation*) از بهترین راه حل‌های پیدا شده، باید در نظر گرفته شوند. در کاوش در ناحیه‌های جستجو نشده بررسی صورت می‌گیرد. در تبعیت، در ناحیه‌های امید بخش که تا به حال در آن ناحیه یک جواب خوب پیدا شده است بررسی بیشتر صورت می‌گیرد. در صورتیکه به رفتار **کاوش** اهمیت بیشتری داده شود، الگوریتم رفتار تصادفی بیشتری خواهد داشت و به سمت رفتار تصادفی میل می‌کند و در

صورتی که به رفتار **تبعیت** توجه بیشتری شود، الگوریتم از رفتار تصادفی فاصله می‌گیرد و جستجو تنها در محدوده راه حل‌های خوب به جستجو می‌پردازد که می‌تواند به راه حل‌های محلی منجر شود.

معیارهای طبقه بندی زیادی ممکن است برای طبقه بندی فراابتکاری‌ها استفاده شود که در زیر به بعضی از آن‌ها اشاره می‌شود:

الهام گرفته از طبیعت در مقابل عدم الهام از طبیعت

خیلی از الگوریتم‌های فراابتکاری از فرایندهای طبیعی الهام گرفته شده‌اند: از قبیل الگوریتم‌های اجتماع مورچگان و زنبور عسل که از هوش توده‌ای از گونه‌های مختلف مورچگان و زنبوران استفاده می‌کنند. برخی از الگوریتم‌های این چنین نبوده و از یک رفتار منطقی پیروی می‌کنند که می‌توان به الگوریتم جستجوی ممنوع نام برد.

نحوه استفاده از حافظه

بعضی از الگوریتم‌های فراابتکاری از قبیل انجماد تدریجی یا گرم و سرد کردن شبیه سازی شده بدون حافظه هستند و حرکت‌های قبلی را در جایی ذخیره نمی‌کنند. در مقابل الگوریتم جستجوی ممنوع از یک حافظه که بعضی از اطلاعات را در طول جستجو بدست می‌آورد، استفاده می‌کند.

قطعی در مقابل احتمالی

یک الگوریتم قطعی، یک مسئله بهینه‌سازی را از طریق تصمیم‌گیری قطعی حل می‌نماید (برای مثال الگوریتم جستجوی محلی و جستجوی ممنوع). در الگوریتم‌های فراابتکاری احتمالی، بعضی از قواعد احتمالی در فرایند جستجو مورد استفاده قرار می‌گیرد که می‌توان به الگوریتم انجماد تدریجی و الگوریتم ژنتیک اشاره کرد. در الگوریتم‌های قطعی، با داشتن یک راه حل اولیه و اجراهای متفاوت، تنها یک جواب نهایی بدست می‌آید و در حالی که در الگوریتم‌های تصادفی، با داشتن یک راه حل اولیه و اجراهای متفاوت، ممکن است که جواب‌های نهایی متفاوتی بدست آید.

تکراری در مقابل حریصانه

در الگوریتم‌های تکراری، الگوریتم با یک راه حل کامل شروع شده و در هر تکرار راه حل یا راه حل‌ها تغییر پیدا می‌کنند. در الگوریتم‌های حریصانه یک راه حل کامل در اختیار نبوده بلکه با یک راه حل ساخته نشده شروع شده و در هر مرحله، یک متغیر تصمیم از مسئله یک قسمت از راه حل را می‌سازد. اغلب الگوریتم‌های فراابتکاری، الگوریتم‌های تکراری هستند.

در ادامه **الگوریتم اجتماع مورچگان** که الگوریتمی است که از طبیعت الهام گرفته است و در رده الگوریتم‌های احتمالی، و حریصانه با حافظه قرار می‌گیرد صحبت می‌شود.

معرفی الگوریتم اجتماع مورچگان

مورچه‌ها بر روی زمین از حدود ۱۰۰ میلیون سال پیش زندگی می‌کرده‌اند و در حال حاضر جمعیتی در حدود 10^{16} در دنیا دارند. تخمین زده شد که مجموع وزن مورچه‌ها در حدود مجموع وزن انسان‌های روی زمین می‌باشند. این حشرات موجوداتی اجتماعی می‌باشند و در کلونی‌هایی به اندازه‌ای از ۳۰ تا میلیون‌ها فرد زندگی می‌کنند. رفتارهای پیچیده کلونی مورچگان، انسان‌ها را تشویق کرده است که مطالعات متعددی در خصوص رفتارهای اجتماعی و فردی آن‌ها داشته باشند. رفتارهای گروهی از مورچگان در ارتباط با جستجو برای غذا، نظم کارگران، سازماندهی گورستان، مواظبت از بچه‌ها و ساختن لانه مورد مطالعه قرار گرفته است.

رفتارهای پیچیده مورچه‌ها، از رفتارهای جمعی مورچه‌های تکی خیلی ساده و بی‌حیله ناشی می‌شود. یک مورچه براساس اطلاعاتی که از محیط اطراف خود به دست می‌آورد، اقدام ساده‌ای را انجام می‌دهد که پایه‌ای تصادفی دارد.

یکی از اولین رفتارهایی که توسط محققین مورد مطالعه قرار گرفت، توانایی مورچه‌ها در یافتن کوتاهترین مسیر بین یک منبع غذا و لانه آن‌ها بود. بر اساس این مطالعات، اولین الگوریتم از رفتار مورچه

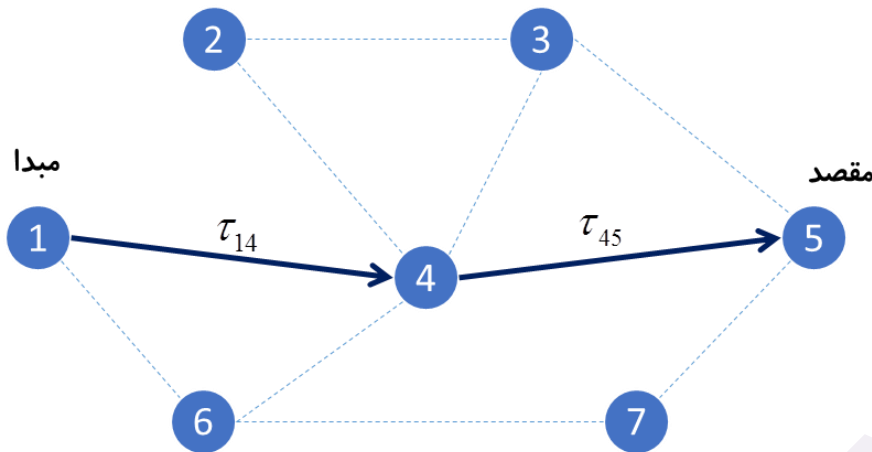
ها در جستجوی غذا توسط **مارکو دوریگو Marco Dorigo** ارائه گردید که خروجی رساله دکترای ایشان در دانشگاه میلان ایتالیا بود.

رفتار مورچه ها در جستجوی غذا

چگونه مورچه ها می توانند کوتاهترین مسیر بین محل غذا و لانه خود را بدون هیچ گونه مکانیسم هدایتی دیداری و مرکزی پیدا کنند؟ مطالعات رفتارهای بعضی از گونه‌های واقعی مورچه ها نشان داده است که یک الگوی تصادفی و شانسی در حرکت مورچه ها برای یافتن غذا وجود دارد. به محض این که یک منبع غذا در محلی در حوالی لانه مورچه ها قرار داده می‌شود، رفتار تصادفی به مرور سازماندهی بیشتری یافته و در نهایت منجر به رسیدن به یک مسیر مشابه توسط مورچه ها به این محل خواهد شد و به صورت جادویی دیده می‌شود که مورچه ها کوتاهترین مسیر را پیدا کرده اند. این رفتار عجیب نتیجه تاثیر مورچه هایی است که منبع غذا را پیدا کرده اند. مورچه هایی که غذا را یافته اند با سایر مورچه ها از طریق یک مکانیسم ارتباطی به نام **اثر فرومون Pheromone Trails** ارتباط برقرار می‌نمایند. زمانی که یک مورچه یک منبع غذا را پیدا می‌کند و مقدار غذا به خانه می‌برد، در مسیر حرکت خود ماده شیمیایی به نام **فرومون** منتشر می‌نماید. بقیه مورچه ها که به دنبال غذا می‌گردند، در هنگام انتخاب مسیر به غلظت فرومون مسیر توجه می‌نمایند. مسیرهای با غلظت فرومون بیشتر، احتمال بیشتری برای انتخاب دارند. هر چقدر مورچه‌های بیشتری از یک مسیر عبور نمایند، غلظت فرومون آن مسیر افزایش یافته و در نتیجه، مورچه‌های بیشتری نیز نسبت به آن مسیر علاقه مند می‌شوند.

ساختار عمومی بهینه سازی اجتماع مورچگان

مساله یافتن کوتاهترین مسیر بین دو راس در یک گراف $G = (V, E)$ است که V مجموعه راس‌های و E ماتریسی است که ارتباطات بین راس ها را بیان می‌کند. این گراف $n_G = |V|$ راس دارد. طول، L^k ، که توسط مورچه k ام ساخته شده است، به صورت فاصله مبدا تا مقصد تعریف می‌گردد. یک گراف نمونه همراه با یک مسیر انتخابی، در شکل زیر نمایش داده شده است. طول مسیر انتخابی برابر با ۲ است. در این حالت، یک غلظت فرومون τ_{ij} به هر یال (i, j) از گراف، تخصیص داده می‌شود.



در الگوریتم استاندارد بهینه سازی اجتماع مورچگان یا همان **SACO** که مخفف **Simple Ant Colony Optimization** است، در ابتدا به هر یال، یک مقدار تصادفی به عنوان فرومون اولیه، τ_{ij}^0 ، تخصیص داده می شود. در ابتدای الگوریتم به علت نبود غلظت فرومون یا مقدار خیلی کم آن، مورچه ها به طور تصادفی مسیرها را انتخاب می نمایند. تعدادی از مورچه ها، $k = 1, \dots, n$ ، در راس مبدا قرار داده می شود. در هر تکرار از الگوریتم **SACO**، هر مورچه به صورت مرحله به مرحله یک مسیر یا راه حل را بین مبدا و مقصد می سازد. در هر راس، هر مورچه فرایند تصمیم گیری را برای تعیین یال بعدی مسیر خود به کار می برد. اگر مورچه k در راس i قرار داشته باشد، راس بعدی $i \in N_i^k$ را بر اساس یک تابع احتمال گذر به شکل زیر تعیین می نماید.

$$p_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)}{\sum_{j \in N_i^k} \tau_{ij}^\alpha(t)} & \text{if } j \in N_i^k \\ 0 & \text{if } j \notin N_i^k \end{cases}$$

که در آن N_i^k مجموعه یال های قابل انتخاب برای مورچه k ام در راس i است. در واقع N_i^k مجموعه یال های همسایه ای راس i است. در صورتیکه $N_i^k = \emptyset$ آنگاه راس پیشین راس i در نظر گرفته می شود و در واقع در صورتیکه مورچه به مسیر بسته برسد، برمی گردد. این مساله منجر به ایجاد یک حلقه می گردد، ولی مشکلی را ایجاد نمی کند زیرا پس از رسیدن به مقصد، در هنگام معرفی مسیر بین مبدا و مقصد، حلقه ها توسط مکانیزمی حذف می گردند. در معادله بالا، α یک مقدار ثابت مثبت برای شدت بخشیدن به اثر

غلظت فرومون است. مقدار بزرگ α ، اهمیت غلظت فرومون را افزایش داده و تاثیر رفتارهای تصادفی را کمتر می کند و منجر به همگرایی سریع به سمت یک بهینه محلی می گردد.

بعد از آن که تمامی مورچه ها، یک مسیر کامل بین مبدا تا مقصد را ساختند و حلقه های احتمالی حذف شدند، هر مورچه ردپای خود را به سمت راس مبدا دنبال می نماید و مقداری فرومون در هر یال (i,j) روی مسیر طی شده، مطابق با قاعده زیر منتشر می نماید.

$$\Delta \tau_{ij}^k(t) \propto \frac{1}{L^k(t)}$$

$L^k(t)$ معرف طول مسیر ساخته شده توسط مورچه k -ام در مرحله زمانی t می باشد. در نتیجه، میزان غلظت فرومون هر مسیر مطابق با قاعده زیر به هنگام می گردد

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \sum_{k=1}^{n_k} \Delta \tau_{ij}^k(t)$$

که در رابطه فوق، n_k تعداد مورچه ها می باشد. با استفاده از رابطه $\Delta \tau_{ij}^k(t)$ مقداری فرومون روی هر یال تولید می گردد و این مقدار بستگی به کیفیت مسیریابی دارد که توسط مورچه ها با استفاده از آن یال ساخته شده است. به عبارت دیگر، هر مورچه مقداری فرومون در هر یال متناسب با کیفیت مسیریابی به جا می گذارد. بعد از یک تکرار، میزان فرومون موجود در یک یال با توجه به مورچه های عبوری از آن یال تغییر می یابد. مقدار چگالی اضافه شده با مقدار قبلی چگالی فرومون آن مسیر طبق معادله $\tau_{ij}(t+1)$ به هنگام می شود.

در SACO ، مقدار کیفیت هر راه حل با یک تابع ساده به صورت عکس طول مسیر طی شده بیان می گردد. البته معیارهای دیگری نیز می توان تعریف کرد. اگر $x^k(t)$ نشان دهنده یک راه حل در مرحله زمانی t باشد، آنگاه $f(x^k(t))$ کیفیت آن راه حل را بیان می کند. اگر مقدار فرومون انتشاری رابطه عکس با طول مسیر داشته باشد در آن صورت، مقدار بیشتر $f(x^k(t))$ موجب می گردد که مقدار کمتری فرومون

در آن مسیر منتشر گردد. بنابراین، یک مسیر طولانی موجب می‌گردد که دارای جذابیت کمتری شوند.

برای مطالعه ادامه این جزوه آموزشی شامل رایه الگوریتم های اجتماع مورچگان و حل یک مثال تشریحی جزوه این درس را تهیه فرمایید.

برای دریافت بسته‌های آموزشی گروه **بهینه‌یاب** به وب سایت ما به نشانی www.behinehyab.com مراجعه کنید.

در صورت هر گونه سوال از طریق ایمیل به نشانی behinehyab@gmail.com و یا بخش تماس با ما وب سایت گروه **بهینه‌یاب** با ما در تماس باشید.

با تشکر از توجه شما

گروه آموزشی **بهینه‌یاب**