

درس ۱۶:

الگوریتم فراابتکاری انجماد تدریجی

تهیه شده توسط گروه بهینه‌یاب



www.behinehyab.com

مقدمه

محاسبه راه حل بهینه *Optimal solution* برای اکثر مسایل بهینه سازی 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 بهینه سراسری 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، آئرودینامیک،

دینامک 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 برنامه ریزی، برنامه ریزی 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 است:

- ✓ بهینه سازی کلونی مورچگان *Ant colony optimization*
- ✓ بهینه سازی 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*
- ✓ بهینه سازی 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* و 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 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 راه‌حل‌های خوب به جستجو می‌پردازد.

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.

نحوه استفاده از حافظه

بعضی از الگوریتم‌های فراابتکاری از قبیل انجماد تدریجی بدون حافظه هستند و حرکتهای قبلی را در جایی ذخیره نمی‌کنند. در مقابل الگوریتم انجماد تدریجی از یک حافظه که بعضی از اطلاعات را در طول جستجو بدست می‌آورد، ذخیره 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 راه‌حل اولیه و اجراهای متفاوت، تنها یک جواب نهایی بدست می‌آید و در حالی که در الگوریتم‌های تصادفی، با داشتن یک راه‌حل اولیه و اجراهای متفاوت، ممکن است که جواب‌های نهایی متفاوتی بدست آید.

تکراری در مقابل حریمانه

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 حریمانه 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 انجماد تدریجی که الگوریتمی است که از طبیعت الهام گرفته است و در رده الگوریتم‌های احتمالی، و تکراری بدون حافظه قرار می‌گیرد صحبت می‌شود.

معرفی الگوریتم انجماد تدریجی

الگوریتم انجماد تدریجی یا الگوریتم گرم و سرد کردن شبیه سازی شده یک روش تصادفی است که مکانیزم آماری جهت یافتن جواب مسایل بهینه سازی استفاده می‌کند. این الگوریتم که به اختصار با **SA** یا **Simulate Annealing** نمایش داده می‌شود بر مبنای دو قاعده از فیزیک آماری عمل می‌نماید.

قاعده اول بر این اساس است که وقتی تعادل ترمودینامیکی به یک دمای مشخص رسید، احتمال یک سیستم فیزیک برای داشتن سطح انرژی E با فاکتور بالتزمن **Boltzmann Factor** که رابطه آن $e^{\frac{-E}{k_B T}}$ که در آن k_B توزیع بالتزمن در دمای مربوطه باشد، متناسب است. براساس این قاعده، احتمال پذیرش جواب‌ها در دماهای مختلف متفاوت بوده و متناسب با تابع معرفی شده می‌باشد.

قاعده دوم نحوه رسیدن به تعادل ترمودینامیکی در یک دمای مشخص را بیان می‌کند. اگر یک تغییر شکل منجر به افزایش تابع هدف یا انرژی به اندازه ΔE بشود، این تغییر با احتمال $e^{\frac{-\Delta E}{T}}$ پذیرفته می‌شود. این پذیرش از طریق تولید یک عدد تصادفی با توزیع یکنواخت در فاصله بین ۰ و یک و مقایسه آن با تابع تعریف شده انجام می‌گردد. در صورتی که مقدار عدد به دست آمده از تابع توزیع یکنواخت کوچکتر از تابع تعریف شده باشد، تغییر شکل پذیرفته می‌شود و در غیر این صورت، تغییر شکل پذیرفته نمی‌شود.

در دمای بالا و مراحل اولیه الگوریتم، مقدار $e^{\frac{-\Delta E}{T}}$ نزدیک به یک بوده و در نتیجه اکثر حرکت‌ها پذیرفته می‌شود و الگوریتم رفتاری شبیه به یک جستجوی تصادفی را خواهد داشت. در دماهای پایین و اواخر الگوریتم، مقدار $e^{\frac{-\Delta E}{T}}$ به صفر نزدیک می‌شود و اکثر جواب‌های بدتر رد می‌شوند و تنها جواب‌های خوب پذیرفته می‌شوند و شانس جابه‌جایی یک جواب خوب با یک جواب بدتر کاهش می‌یابد. این تغییر رویه الگوریتم، خروج الگوریتم از جواب بهینه محلی را تضمین می‌کند و از طرف دیگر، کاهش احتمال پذیرش جواب بدتر با کاهش دما، موجب تضمین همگرایی SA می‌شود.

در این الگوریتم، x_1 معرف یک راه حل اولیه امکان پذیر است که محدودیت‌های مدل را رعایت نموده است. $F(x)$ مقدار تابع هدف به ازای x بوده و x_c یک راه حل جاری در هر مرحله می‌باشد. $V(x_c)$ مجموعه‌ای است که برای نشان دادن همسایه‌های x_c به کار می‌رود. P یک عدد تصادفی می‌باشد که در زمان برخورد با جواب بدتر به منظور بررسی پذیرش یا عدم پذیرش آن جواب به کار رود. در **قدم اول** الگوریتم، یک راه حل اولیه ایجاد و مقدار تابع هدف آن محاسبه می‌گردد. در این قدم، راه حل اولیه به عنوان راه حل جاری x_c و بهترین راه حل x^* نیز در نظر گرفته می‌شود. البته، این دو راه حل در طول الگوریتم تغییر خواهد یافت. همچنین در ابتدای الگوریتم، مقدار تابع هدف راه حل اولیه، به عنوان مقدار تابع هدف راه حل جاری $F(x_c)$ و بهترین راه حل $F(x^*)$ در نظر گرفته می‌شود. در این قدم دمای اولیه T_0 به عنوان دمای جاری یا همان T_c در نظر گرفته می‌شود.

در **قدم دوم**، برای هر راه حل جاری، یک راه حل همسایه که به صورت $x_{c'}$ در همسایگی آن پیدا شده و مقدار تابع هدف آن به صورت $F(x_{c'})$ محاسبه می‌گردد. در صورتی که همسایه جدید مقدار تابع هدف جاری را بهبود بدهد یا برابر آن باشد، راه حل همسایه به عنوان راه حل جاری پذیرفته می‌شود که در این صورت تغییرات $x_c \leftarrow x_{c'}$ و $F(x_c) \leftarrow F(x_{c'})$ انجام می‌شود. در غیر این صورت، اگر راه حل همسایه منجر به بدتر شدن تابع هدف جاری گردد، یعنی $F(x_c) < F(x_{c'})$ ، همسایه با تابع احتمال $e^{\frac{-\Delta F}{T_c}}$ که در آن $\Delta F = F(x_{c'}) - F(x_c)$ پذیرفته می‌شود. برای این منظور مقدار تصادفی $p \sim U(0,1)$ تولید شده و

سپس با تابع احتمال $e^{\frac{-\Delta F}{T_c}}$ مقایسه می گردد. در صورتیکه مقدار P کمتر از تابع احتمال باشد، راه حل همسایه به عنوان راه حل جاری پذیرفته می شود و در غیر این صورت راه حل جاری تغییر نمی کند.

بعد از یافتن هر راه حل جدید که منجر به بهبود راه حل جاری گردد، آن راه حل با بهترین راه حل مقایسه می گردد. اگر این راه حل از بهترین راه حلی که تا به حال پیدا شده بهتر باشد، $F(x_{c'}) \leq F(x^*)$ (به دنبال کمینه سازی هستیم)، جایگزین بهترین راه حل $x_{c'} \rightarrow x^*$ و $F(x_{c'}) \rightarrow F(x^*)$ می گردد. این قدم تا زمانی ادامه می یابد تعداد مشخصی از جستجوها انجام شود.

در **قدم سوم**، بعد از اتمام یک زنجیره (شامل تعدادی مشخص جستجو است که معمولاً برابر با یک در نظر گرفته می شود)، کاهش دما اتفاق خواهد افتاد. معمولاً برای کاهش دما از یک ضریب ثابت $0 < r < 1$ استفاده می شود. رابطه کاهش دما در این حالت به صورت $T \leftarrow rT$ تعریف می شود.

در **قدم چهارم**، شرط توقف الگوریتم بررسی می گردد. در صورتی که شرط توقف برآورده شود، الگوریتم متوقف و در غیر این صورت به **قدم دوم** بر می گردد. شرط توقف های مختلف می توان تعریف کرد. یک شرط توقف رایج، رسیدن دمای الگوریتم به یک دمای انتهایی یا همان T_f می باشد که به صورت $T \leq T_f$ تعریف می گردد.

برای مطالعه ادامه این درس، حل تمرین و پیاده سازی الگوریتم به زبان ویژال بیسیک، جزوه این درس را تهیه کنید.

برای دریافت بسته‌های آموزشی گروه **بهینه‌یاب** شامل جزوه، فیلم و فایل آرایه آموزش این درس 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**