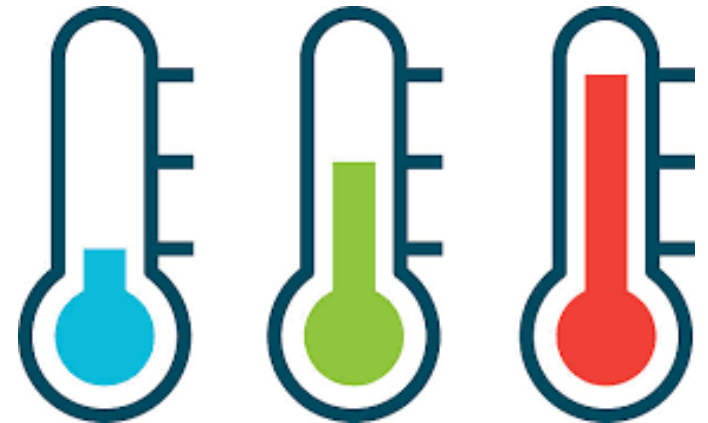
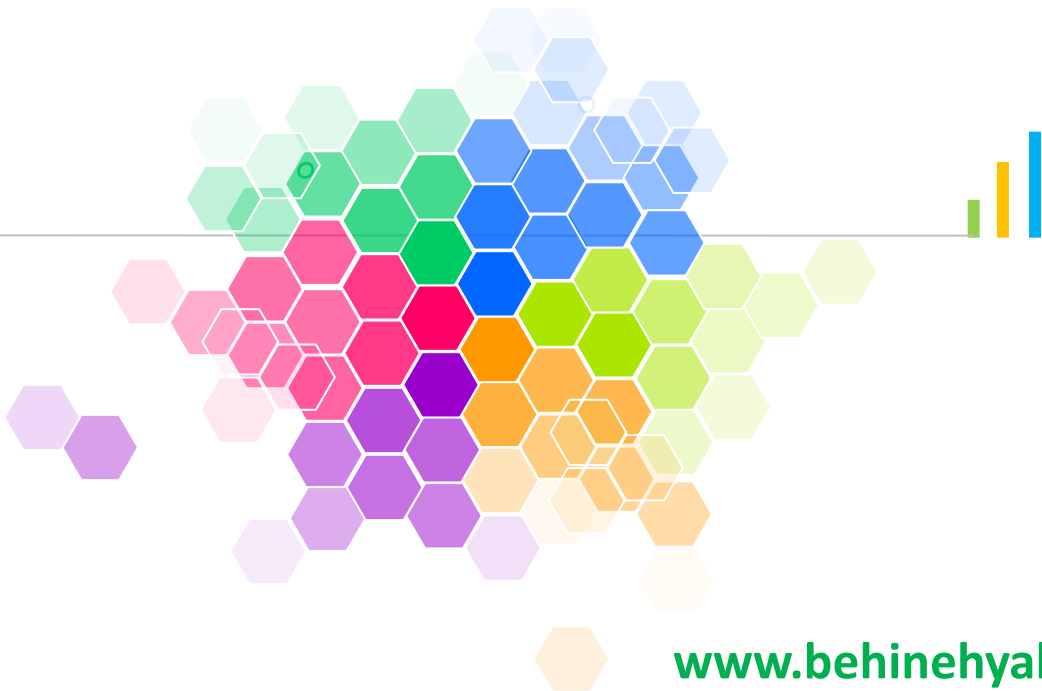


به نام خدا



درس ۱۶: الگوریتم فراابتکاری انجماد تدریجی



فهرست مطالب

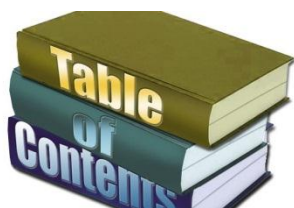


۱ مقدمه الگوریتم فراابتکاری انجاماد تدریجی

۲ اصول الگوریتم انجاماد تدریجی

۳ گام های الگوریتم انجاماد تدریجی

۴ حل یک مثال



مقدمه الگوریتم فراابتکاری انجماد تدریجی

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 راه حل‌های خوب که از الگوریتم‌های هیوریستیک *Heuristic* یا متاهیوریستیک (همان فراابتکاری) *Metaheuristic* بدست می‌آید، اکتفا می‌گردد.

مقدمه الگوریتم فراابتکاری انجماد تدریجی

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 بهینه سازی تخصصی به کار روند، تعریف کرد.

برخلاف روش‌های دقیق، متاهیوریستیک‌ها (فراابتکاری‌ها) برای مسایل با اندازه‌های بزرگ کاربرد دشوار دارد و راه‌حلهایی راضی‌کننده‌ای در زمان معقولی‌ارایه می‌نمایند. در این الگوریتم‌ها، هیچ‌گونه ضمانتی برای یافتن جواب بهینه سراسری یا حدودی از آن وجود ندارد.

مقدمه الگوریتم فراابتکاری انجماد تدریجی



فراابتکاری‌ها در خیلی از زمینه‌ها از قبیل موارد زیر کاربرد دارند:

✓ طراحی مهندسی، بهینه سازی توپولوژی، بهینه سازی مسایل الکترونیک، آئرو دینامیک،

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

مقدمه الگوریتم فراابتکاری انجماد تدریجی



روش های مختلف الگوریتم های فراابتکاری یا متاهیوریستیک تا به حال پیشنهاد شده است که

به صورت زیر است:

✓ جستجوی محلی تکراری *Iterated local search*

✓ بهینه سازی کلونی مورچگان *Ant colony optimization*

✓ بهینه سازی توده ذرات *particle swarm intelligent*

✓ بهینه سازی کلونی زنبوران *Bee colony*

✓ انجماد تدریجی *Simulated Annealing*

✓ الگوریتم های ترتیبی *cultural algorithms*

✓ الگوریتم های با هم تکاملی *Co-evolutionary algorithms*

✓ الگوریتم ژنتیک *Genetic algorithm*

مقدمه الگوریتم فراابتکاری انجماد تدریجی

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.

مقدمه الگوریتم فراابتکاری انجماد تدریجی



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

نحوه استفاده از حافظه

قطعی در مقابل احتمالی

تکراری در مقابل حریصانه

اصول 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 می کند. این الگوریتم که به اختصار با *SA* یا *Simulate Annealing* نمایش داده می شود بر مبنای دو قاعده از فیزیک آماری عمل می نماید.

اصول الگوریتم انجماد تدریجی

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

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.

اصول الگوریتم انجماد تدریجی

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.

اصول الگوریتم انجماد تدریجی

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.

اصول الگوریتم انجماد تدریجی

در این الگوریتم، x_1 معرف یک راه حل اولیه امکان پذیر است که محدودیت های مدل را رعایت نموده است. $F(x)$ مقدار تابع هدف به ازای x بوده و x_c یک راه حل جاری در هر مرحله می باشد. $V(x_c)$ مجموعه ای است که برای نشان دادن همسایه های x_c به کار می رود. P یک عدد تصادفی می باشد که در زمان برخورد با جواب بدتر به منظور بررسی پذیرش یا عدم پذیرش آن جواب به کار رود.

مراحل الگوریتم انجماد تدریجی

در قدم اول الگوریتم، یک راه حل اولیه ایجاد و مقدار تابع هدف آن محاسبه می‌گردد. راه حل اولیه به عنوان راه حل جاری x_c و بهترین راه حل x^* نیز در نظر گرفته می‌شود. 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.

مراحل الگوریتم انجماد تدریجی



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 می گردد. 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 می شود.

مراحل الگوریتم انجماد تدریجی

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 می‌گردد. در صورتیکه

مقدار P کمتر از تابع احتمال باشد، راه حل همسایه به عنوان راه حل جاری پذیرفته می‌شود و در

غیر این صورت راه حل جاری تغییر نمی‌کند.

مراحل الگوریتم انجماد تدریجی

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.

مراحل الگوریتم انجماد تدریجی



عوامل مختلفی در عملکرد الگوریتم انجماد تدریجی موثر است که می‌توان به موارد زیر اشاره

کرد.

۱- دمای اولیه

۲- قاعده کاهش دما

۳- قاعده توقف

دمای اولیه

استفاده از دمای اولیه بالا موجب می‌گردد که الگوریتم رفتاری شبیه به یک جستجوی تصادفی به خود بگیرد و استفاده از دماهای پایین برای دمای اولیه، موجب می‌گردد که الگوریتم تبدیل به یک الگوریتم جستجوی محلی گردد.

مراحل الگوریتم انجماد تدریجی

پذیرش کلیه راه حل ها

در این حالت مقدار دمای اولیه به اندازه ای بزرگ در نظر گرفته می شود که کلیه راه حل ها در طول فاز اولیه ای از الگوریتم پذیرفته شود.

وابسته به انحراف معیار

در این استراتژی، دمای اولیه برابر با $k\sigma$ براساس یک سری آزمایشات اولیه، محاسبه می گردد که در آن σ نشان دهنده انحراف معیار استاندارد توابع هدف جواب های بدست آمده در آزمایشات اولیه بوده و $k = -3 / \ln(p)$ که احتمال پذیرش p مرتبط با ناحیه 3σ در تابع توزیع احتمال می باشد.

مراحل الگوریتم انجماد تدریجی

قاعده کاهش دما

قاعده خطی

$$T_i = T_0 - i \times \beta$$

که در آن T_0 دمای اولیه، i شماره تکرار کاهش دما و β ضریب ثابتی بین صفر و یک می

باشد.

قاعده هندسی

یکی از قواعد مشهور کاهش دما، قاعده هندسی کاهش دما می باشد که به صورت $T_{k+1} = \alpha T_k$

معرفی می گردد و در آن $0 < \alpha < 1$ و α ضریب ثابتی است.

مراحل الگوریتم انجماد تدریجی



قاعده لگاریتمی

این قاعده به صورت تابع زیر تعریف می شود. این قاعده نسبت به دو قاعده دیگر، سرعت کاهش دمای آن آرامتر بوده و باعث همگرایی بیشتری به سمت بهینه سراسری می شود.

$$T_i = \frac{T_0}{\log(i)}$$

مراحل الگوریتم انجماد تدریجی

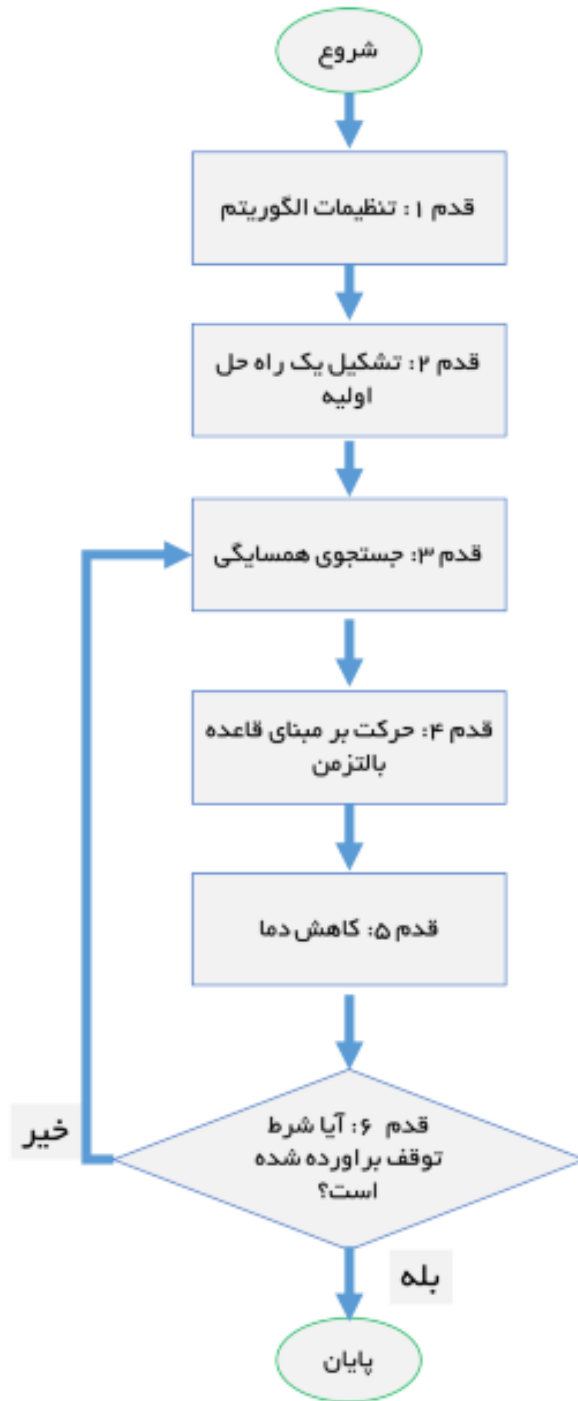
قاعده توقف

رسیدن به یک دمای نهایی

رسیدن دمای الگوریتم به یک دمای انتهایی معروف ترین قاعده توقف می باشد.

عدم بهبود در جواب نهایی

رسیدن تعداد حرکت های بدون بهبود در بهترین حرکت نیز یکی از قواعد توقف مشهور می باشد. در این حالت، فرض بر این است که اگر الگوریتم تا حد معینی به جستجوی خود ادامه دهد و فایده ای نداشته باشد، بهتر است الگوریتم متوقف شود زیرا احتمال یافتن نقطه ای بهتر از بهترین نقطه، خیلی کم است.



حل یک مثال

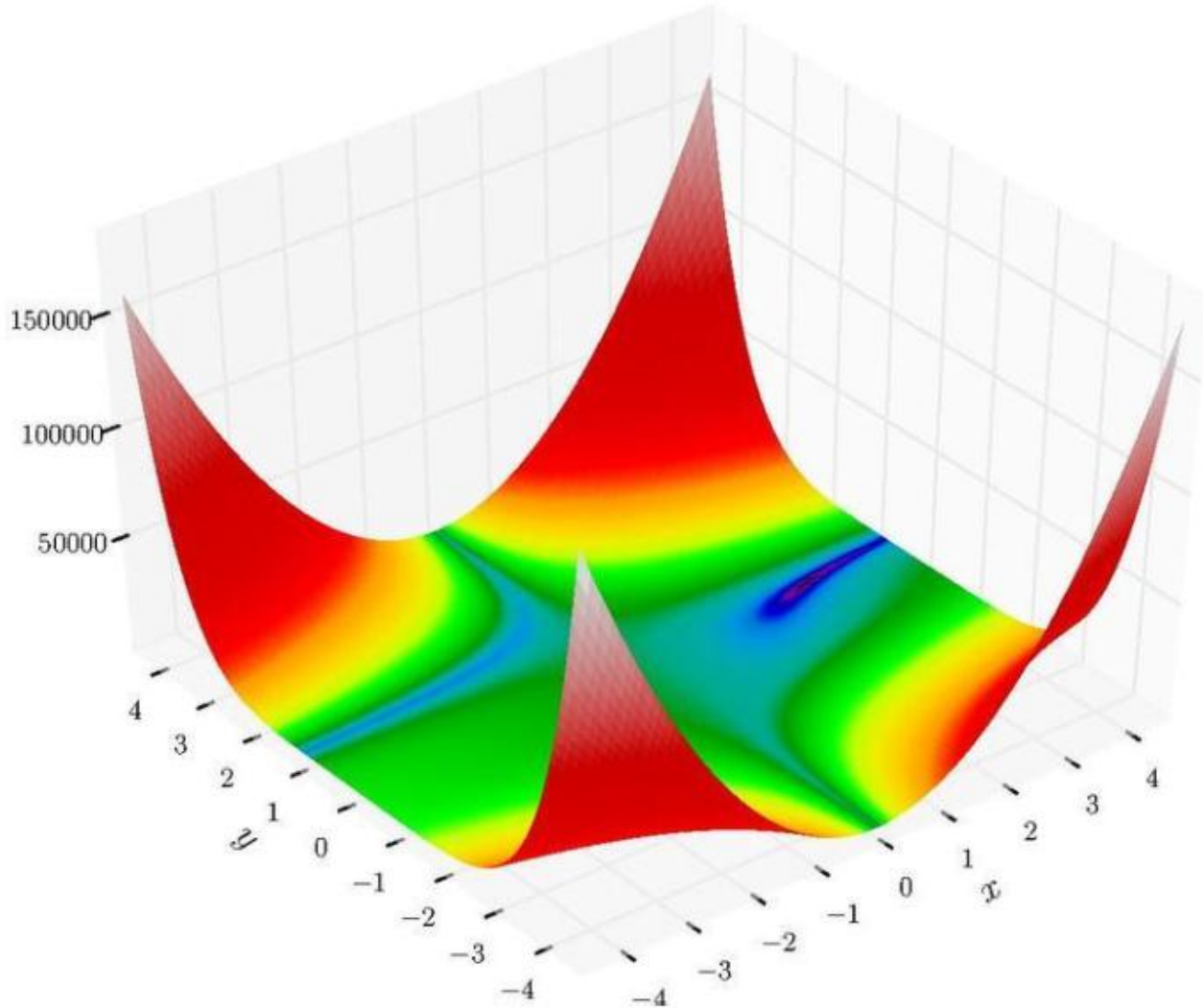
مثال

در این قسمت می خواهیم یک مسئله بهینه سازی بدون محدودیت را با استفاده از روش جستجوی ممنوع حل نماییم. در این مثال به دنبال کمینه کردن تابع هدف زیر هستیم.

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

مقدار بهینه این مدل برابر با صفر و مقدار متغیرهای برابر با $(x^*, y^*) = (3, 0.5)$ هستند.

حل یک مثال



حل یک مثال

قدم ۱: تنظیمات الگوریتم

در این قدم تنظیمات اولیه برای اجرای الگوریتم آورده می شود. دمای اولیه الگوریتم برابر با $T_0 = 1000$ در نظر گرفته می شود. از قاعده کاهش دمای هندسی استفاده می شود و نرخ کاهش دما برابر با $\alpha = 0.99$ لحاظ می گردد. بازه ای تغییرات دو متغیر $x \in [0,10]$ و $y \in [0,10]$ است.

حل یک مثال

قدم ۲: تشکیل یک راه حل اولیه

برای نمایش جواب‌ها و ساخت جواب‌های همسایگی از رویکرد مبنای دو استفاده شد. هر عدد

صحیح در مبنای ده را می‌توان به صورت یک عدد در مبنای ۲ نمایش داد.

$$(x')_{10} = (a_0, a_1, a_2, \dots, a_{n-1}, a_n) \rightarrow x' = \sum_{i=0}^n a_i 2^i$$

در این تمرین، در صورت استفاده از نمایش دودویی هر جواب، دقت جواب بدون رقم اعشار

در نظر گرفته می‌شود و این برای محاسبه جواب بهینه کافی نیست. برای افزایش دقت و تولید

اعداد بین صفر تا ۱۰، اعدادی بر مبنای دو تولید می‌شود که بین صفر تا ۱۰۲۳ است و سپس عدد

تولید شده بر عدد ۱۰۰ تقسیم می‌شود.

حل یک مثال

$$(x'')_2 = (0, 0, 1, 1, 0, 0, 1, 1, 0, 0)$$

$$\begin{aligned} \rightarrow (x')_{10} &= 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7 + 0 \times 2^8 + 0 \times 2^9 \\ &= 4 + 8 + 64 + 128 = 204 \end{aligned}$$

برای تولید یک مقدار برای متغیر x تنها کافی است مقدار تولید شده در بالا را بر ۱۰۰ تقسیم

نماییم. در این صورت مقدار تولید شده برای x برابر با ۲.۰۴ می‌شود.

در انتهای این قدم، مقدار تابع هدف براساس دو متغیر x و y با توجه به تابع هدف زیر

محاسبه می‌شود.

$$f(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2$$

حل یک مثال

قدم ۳: جستجوی همسایگی

۱- یک عدد تصادفی پیوسته بین صفر و یک تولید شود.

۲- اگر عدد تصادفی تولید شده کمتر از ۰,۵ بود آنگاه متغیر x تغییر می کند و در غیر این صورت متغیر y تغییر می کند.

۳- یک عدد تصادفی گسسته بین صفر و نه تولید می شود. عدد انتخاب شده را i می نامیم. در این صورت مولفه a_i اگر صفر است به یک و اگر یک است به صفر تغییر می دهیم.

حل یک مثال

برای مثال همسایگی جواب $(x, y) = (2.04, 7.91)$ را تولید کنید.

$$x = 2.91 \rightarrow (x')_{10} = 291 \rightarrow (x'')_2 = (11001100)_2$$

$$y = 7.91 \rightarrow (y')_{10} = 791 \rightarrow (y'')_2 = (1100010111)_2$$

مقدار عدد تصادفی برای انتخاب یکی از دو متغیر برابر ۰,۵۷ در نظر گرفته می شود. در این صورت متغیر y انتخاب می شود. یک عدد تصادفی بین صفر و نه تولید می شود. مقدار این عدد برابر ۵ می شود. در این صورت مولفه a_5 یا $(11000\underline{1}0111)_2$ که برابر ۱ است انتخاب می شود. در این صورت مقدار ۱ به مقدار صفر تغییر می کند و مقدار جدید برابر با $(11000\underline{0}0111)_2$ می شود. در این صورت مقدار y به صورت زیر محاسبه می شود.

حل یک مثال

$$(y'')_2 = (1100000111)_2 \rightarrow (y')_{10} = 775 \rightarrow y = 7.75$$

در این قدم، جواب کنونی $(x, y) = (2.04, 7.91)$ به جواب همسایگی $(x_n, y_n) = (2.04, 7.75)$

می‌رویم.

در انتهای این قدم مقدار تابع هدف جواب جدید محاسبه می‌شود و آن را f_n می‌نامیم.

حل یک مثال

قدم ۴: حرکت بر مبنای قاعده بالتزمن

اگر $f_n < f$ باشد در این صورت جواب فعلی برابر با $(x_c, y_c) = (x_n, y_n)$ می شود. اگر

$f_n \geq f$ باشد مقدار $e^{\frac{\Delta f}{T}}$ که در آن $\Delta f = f_n - f$ است محاسبه می شود. یک عدد تصادفی بین

صفر و یک، r ، تولید می شود. اگر $r < e^{\frac{\Delta f}{T}}$ باشد آنگاه $(x_c, y_c) = (x_n, y_n)$ می شود و این به این

معنا است که با احتمال r به جواب بدتر حرکت می کنیم. و در غیر این صورت جواب کنونی تغییر

نمی کند.

حل یک مثال

قدم ۵: کاهش دما

در این مثال از قاعده هندسی برای کاهش دما استفاده می‌شود. به این صورت که دما جدید از رابطه زیر محاسبه می‌شود.

$$T = 0.99 \times 1000 = 990$$

قدم ۶: آیا شرط توقف برآورده شده است؟

شرط‌های توقف مختلفی می‌توان ارایه کرد. در این مثال زمانی الگوریتم به پایان می‌رسد که دما از ۰٫۰۱ کمتر شود.

حل یک مثال

با توجه به موارد ذکر شده اکنون می‌توان جواب بهینه تابع $f(x,y)$ را محاسبه کرد. برای پیاده سازی الگوریتم گرم و سرد کردن شبیه سازی شده از زبان **Visual Basic for Application** که زبان برنامه نویسی در نرم افزارهای **Office** است استفاده می‌کنیم. این زبان همان زبان برنامه نویسی **Visual Basic** است ولی این امکان در آن فراهم شده است که از قابلیت های **Office** در این زبان بهره برد. در کنار این زبان برنامه نویسی از قابلیت های **Excel** برای نمایش جواب ها استفاده می‌شود.

با تشکر

راه های ارتباطی با ما

www.behinehyab.com

behinehyab@gmail.com

