

الگوریتم تکامل تفاضلی

این الگوریتم یک الگوریتم تکاملی مبتنی بر جمعیت است که رفتارش به صورت تصادفی است و با استفاده از یک سری پاسخ پیشنهادی کار خود را شروع می کند و طی یک سری تکرارهای متوالی سعی می کند نتیجه ی بهتری برای یک مسئله ی بهینه سازی که ما به دنبال حل آن هستیم، به دست آورد. این الگوریتم شباهت های زیادی به سایر الگوریتم های تکاملی مثل ژنتیک و... دارد اما تفاوت هایی هم با این الگوریتم ها دارد که مهمترین آن ها در شیوه ی منحصر به فردی است که الگوریتم DE برای تولید پاسخ های جدید استفاده می کند. مکانیزمی که الگوریتم های تکاملی به طور کلی برای بهبود پاسخ ها و ایجاد پاسخ های جدید به کار می گیرند شامل سه اپراتور اساسی است: **selection** ، **mutation** ، **crossover**.

غالباً با انجام یکی از این اپراتورها یعنی **crossover** یا **mutation** جمعیت جدیدی ایجاد می شود. الگوریتم تکامل تفاضلی در مورد ترتیب اعمال این اپراتورها و شیوه ی اعمال جهش با سایر الگوریتم ها متفاوت است. در DE برخلاف سایر الگوریتم های تکاملی که ابتدا **crossover** انجام می شود و سپس **mutation**، ابتدا با استفاده از عملگر جهش یک پاسخ موقت ایجاد می شود و سپس با استفاده از عملگر تقاطع یک پاسخ قطعی ایجاد می شود. در این الگوریتم شیوه ی انجام جهش هم متفاوت است. در الگوریتم های تکاملی دیگر با استفاده از یک مدل احتمالی و با استفاده از یک توزیع جهش را انجام می دهیم که این تابع توزیع کاملاً مشخص است. اما در این الگوریتم از یک توزیع مشخصی برای نمونه برداری و ایجاد گام جهش استفاده نمی شود، بلکه تفاضل و اختلاف بین پاسخ هایی که تاکنون به آن رسیدیم برای ایجاد پاسخ های جدید و جهش مورد استفاده قرار می گیرد. این اختلاف بین اعضای جمعیت اطلاعات مفیدی در مورد تابع هدف و مسئله ی بهینه سازی ای که ما به دنبال حل آن هستیم دارد. مثلاً در الگوریتم ژنتیک وقتی فرزندان با والدین مقایسه می شوند، یک عده از اعضای جمعیت دور ریخته می شوند و اضافی هستند، اما در بین این اعضای جمعیت هم اطلاعات مفیدی است که ما از آنها استفاده نمی کنیم، اگر بخواهیم از این اطلاعات استفاده کنیم، یک راه، استفاده از تفاضل و اختلاف بین اعضای جمعیت است که می تواند باعث افزایش قابلیت جستجوی الگوریتم شود و نتایج بهتری را در برداشته باشد.

فاصله میان اعضای جمعیت

به این ترتیب چون عمل جهش و گام های جهش با استفاده از این اختلاف ها تعیین می شود، دیده می شود که در DE رفته رفته با همگرایی این گام های حرکت کمتر می شود، یعنی دقت الگوریتم در بررسی یک سری از نواحی افزایش می یابد. چون در ابتدا اعضای جمعیت پراکنده هستند فاصله ی بین اعضا بیشتر است و الگوریتم باید فضای بزرگی را جستجو کند. این فاصله ی بین اعضای جمعیت اطلاعات مفیدی در

مورد پراکندگی اعضای جمعیت دارد که می‌توان با استفاده از این فاصله ی بین اعضای جمعیت، طول و گام و شدت جهش را تعیین کنیم. به این صورت که اگر اعضای جمعیت نسبتاً از هم دور باشند اصولاً اعضا باید گام‌های بزرگتری را بردارند. برعکس اگر یک همگرایی نسبی اتفاق بیفتد، باید فاصله کمتر باشد. به عنوان یک جمع بندی در الگوریتم DE این که چطور طول گام و طول جهش مشخص می شود را می توانیم به این صورت بگوییم که: ابتدا دو عضو از جمعیت به تصادف انتخاب می شوند. سپس اختلاف این دو عضو که به آن بردار تفاضل گویند محاسبه می شود. این بردار تفاضل در یک پارامتری به نام پارامتر مقیاس ضرب می شود که حاوی اطلاعاتی از طول و جهت جهش است. این بردار به عنوان یک بردار جهش در نظر گرفته می شود و یکی از اعضای جمعیت به این صورت انتخاب شده و جابجا می شود.

مزایای استفاده از بردار تفاضل

اطلاعات موجود در اعضای جمعیت به طور کامل مورد استفاده قرار می گیرند و به عبارتی دور ریز اطلاعاتمان کمتر می شود، البته این بدین معنا نیست که از همه ی اطلاعات به طور کامل استفاده کنیم، بلکه نسبت به آن حالتی که در الگوریتم ژنتیک داشتیم، اطلاعات بیشتری مورد استفاده قرار می گیرد و منجر می شود تا فرآیند جستجو و بهینه سازی هوشمندانه تر دنبال شود.

مراحل الگوریتم تکامل تفاضلی

۱- تعریف پارامترهای مسئله و الگوریتم: پارامترهای مسئله می تواند شامل تعریف تابع هزینه، تعداد پارامترهای مجهول یا متغیرها و پارامترهای تصمیم‌گیری باشد.

۲- مقداردهی اولیه^۱: قبل از اینکه جمعیت بتواند مقداردهی اولیه شود، باید حد بالا و پایینی برای هر پارامتر مشخص شود. وقتی حدود مقادیر اولیه مشخص شد، یک مولد عدد تصادفی به هر پارامتر بردار، یک مقدار در محدوده مجاز تعیین شده اختصاص می‌دهد.

۲- جهش^۲: پس از مقداردهی اولیه، DE جهش می‌یابد و جمعیت را دوباره ترکیب می‌کند. جهش تفاضلی، یک بردار تفاضل را که بصورت تصادفی نمونه‌برداری شده است، به بردار سومی اضافه می‌کند. معادله (۱-۲) نشان می‌دهد که چگونه این سه بردار تصادفی متفاوت را ترکیب کنیم تا یک بردار جهش، $V_{i,g}$ ، تولید کنیم.

$$V_{i,g} = x_{i_0,g} + F \times (x_{i_1,g} - x_{i_2,g}) \quad (1-2)$$

¹ Initialization

² Mutation

که در معادله فوق فاکتور مقیاس^۳، $F \in (0,1^+)$ ، یک عدد حقیقی مثبت است که سرعتی را که جمعیت نمو می کند، کنترل می کند.

۳- تقاطع^۴: در DE هر بردار از جمعیت جاری با یک بردار جهش آمیخته می شود و یک پاسخ موقت تولید می کند. در ادامه این پاسخ موقت را با بردار جمعیت جاری با استفاده از اپراتور crossover بصورت زیر ترکیب می کنیم.

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if (rand}_j(0,1) \leq C_r \text{ or } j=j_{\text{rand}}) \\ x_{j,i,g} & \text{o.w} \end{cases}$$

که احتمال Crossover، $C_r \in [0,1]$ ، یک مقدار تعریف شده توسط کاربر است که بخشی از مقادیر پارامتر که از جهش کپی شده است را کنترل می کند.

۴- انتخاب^۵: اگر مقدار تابع هدف بردار آزمایش، $u_{i,g}$ ، کوچکتر یا مساوی مقدار تابع هدف بردار هدف، $X_{i,g}$ ، باشد، در نسل بعد بردار آزمایش جایگزین بردار هدف می شود. در غیر اینصورت، پارامتر مربوط به بردار هدف برای حداقل یک نسل بیشتر در همان مکانش در جمعیت باقی می ماند.

$$X_{i,g+1} = \begin{cases} u_{i,g} & \text{if (} f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{o.w} \end{cases}$$

وقتی جمعیت جدید بکار گرفته شد، تا زمانی که نقطه یا مقدار بهینه مکان یابی شود یا معیار از قبل تعیین شده برای توقف (مثلا تعداد نسل ها به مقدار از پیش تنظیم شده g_{\max} برسد) ارضا شود، فرایند جهش، باز ترکیب و انتخاب تکرار می شود.

۵- شرایط خاتمه: انواع شرایط خاتمه که می توان در الگوریتم های تکاملی و نه صرفا در DE در نظر گرفت. ممکن است تعداد معینی از نسل ها یا تکرارها مدنظر باشد. یا رسیدن به یک مقدار تابع هزینه ی خاص مدنظر باشد. ممکن است الگوریتم را از نظر زمانی محدود کنیم یا معیارهای دیگر.

تاثیر پارامترهای موجود در الگوریتم تکامل تفاضلی

³ Scaling Factor

⁴ Crossover

⁵ Selection

اندازه جمعیت: تغییر اندازه جمعیت تاثیر مستقیمی در فرآیند جستجو دارد. اگر اندازه جمعیت بیشتر باشد پاسخ های بیشتری بررسی می شود و قابلیت جستجوی الگوریتم افزایش می یابد ولی بار محاسباتی زیادی تحمیل می شود.

پارامتر مقیاس: این مقدار یا حتما باید عددی مثبت باشد یا حتما باید منفی باشد، یعنی نباید در طول الگوریتم تغییر علامت دهد. چون در غیر اینصورت طول جهش ها نمی تواند میانگین صفر داشته باشد و الگوریتم را از حالت میانگین صفر بودن خارج می کند، میانگین صفر بودن به این معنی است که می خواهیم یک مقدار به اعضای جمعیتی که تاکنون به دست آوردیم وفادار باشیم در غیر اینصورت باعث ناپایداری الگوریتم می شود و منجر می شود که جمعیت به جای دیگر فضا منتقل شود. با کاهش پارامتر مقیاس طول گام های جهش کمتر می شود و در نتیجه زمان بیشتری لازم است تا الگوریتم فضا را جستجو کند و این امر منجر به کند شدن همگرایی می شود اما از طرفی دقت الگوریتم افزایش می یابد.

احتمال تقاطع: بیشتر شدن این احتمال باعث می شود تا سهم پاسخ های موقت بیشتر شود، این تنوع پاسخ های جدید را افزایش می دهد و منجر می شود تا سرعت همگرایی افزایش یابد ولی ممکن است الگوریتم در دام بهینه های محلی بیفتد. از طرف دیگر کم شدن آن نیز فرآیند جستجو را مقاوم⁶ می کند. بنابراین روند الگوریتم کند می شود و زمان بیشتری را صرف می کند. پس احتمال تقاطع نه باید خیلی کم و نه خیلی زیاد باشد و این انتخاب کاملا به نوع مسئله و ابعاد فضا بستگی دارد.

⁶ Robust