

باسمه تعالی

موضوع :

گراف رنگ آمیزی و هشت وزیر با الگوریتم زنبور

پاییز ۱۴۰۳ - آذر ماه

درس هوش مصنوعی - دکتر مهدی مصلح

حسین کوشکیان

فهرست

- 3 مقدمه
- 4 راه حل الگوریتم زنبور عسل
- 5 مسئله ی هشت وزیر
- 6 مثال عددی برای توضیح:
- 7 گراف رنگ آمیزی چیست؟

مقدمه

الگوریتم‌های الهام‌گرفته از طبیعت در دهه‌های اخیر به عنوان راه‌حل‌های کارآمد برای مسائل پیچیده در حوزه بهینه‌سازی مورد توجه محققان قرار گرفته‌اند. یکی از این الگوریتم‌ها، **الگوریتم زنبور عسل** است که از رفتار طبیعی زنبورهای عسل در جستجوی غذا الهام گرفته شده است. این الگوریتم بر پایه تعادل میان **کاوش** (Exploration) و **بهره‌برداری** (Exploitation) طراحی شده است و تلاش می‌کند تا با جستجوی محلی و جهانی، راه‌حل‌های بهینه یا نزدیک به بهینه را در مسائل مختلف پیدا کند.

در این الگوریتم، زنبورها به دو دسته تقسیم می‌شوند: **زنبورهای کارگر** که وظیفه جستجوی منابع غذا در مکان‌های مختلف را دارند و **زنبورهای دیده‌بان** که وظیفه ارزیابی نتایج و هدایت سایر زنبورها به سمت بهترین منابع را بر عهده دارند. این مکانیزم با انتخاب هدفمند و تقویت نواحی موفق، الگوریتم را به سمت بهینه‌سازی کارآمد هدایت می‌کند. کاربرد الگوریتم زنبور عسل در مسائلی مانند **گراف رنگ‌آمیزی** و **مسئله هشت وزیر** از اهمیت ویژه‌ای برخوردار است. در مسئله گراف رنگ‌آمیزی، هدف یافتن کمترین تعداد رنگ برای رنگ‌آمیزی رأس‌های گراف است، به گونه‌ای که هیچ دو رأس مجاور رنگ یکسان نداشته باشند. در مسئله هشت وزیر نیز باید موقعیت هشت وزیر بر روی صفحه شطرنج به گونه‌ای تعیین شود که هیچ دو وزیری همدیگر را تهدید نکنند. هر دو مسئله به دلیل فضای جستجوی بزرگ و محدودیت‌های خاص، از جمله مسائل NP-Complete محسوب می‌شوند.

الگوریتم زنبور عسل به دلیل توانایی در کاوش مؤثر فضای جستجو و بهینه‌سازی پاسخ‌ها، گزینه‌ای مناسب برای حل این مسائل است. این الگوریتم با استفاده از مکانیزم‌هایی مانند انتخاب مسیر بر اساس احتمال، بهبود تدریجی راه‌حل‌ها، و تقویت مسیرهای موفق، می‌تواند راه‌حل‌های کارآمدی برای این مسائل ارائه دهد. در این تحقیق، به بررسی و تحلیل نحوه عملکرد الگوریتم زنبور عسل در حل این دو مسئله پرداخته خواهد شد.

الگوریتم زنبور عسل (Bees Algorithm)

الگوریتم زنبور عسل از رفتار طبیعی زنبورها در جستجوی غذا الهام گرفته شده است. ابتدا تعدادی نقطه به صورت تصادفی در فضای جستجو انتخاب می‌شود که هر نقطه یک راه حل ممکن است. این نقاط بر اساس تابع هدف ارزیابی شده و بهترین نقاط به عنوان نقاط برتر انتخاب می‌شوند. برای این نقاط، زنبورهای کارگر جستجوی محلی انجام می‌دهند تا راه حل بهتری پیدا کنند. هم‌زمان زنبورهای دیده‌بان نقاط جدیدی را به صورت تصادفی در فضای جستجو بررسی می‌کنند. نقاط به روز شده، مجدداً ارزیابی می‌شوند و فرآیند تا زمانی که به شرط توقف (مانند رسیدن به راه حل بهینه) برسد، تکرار می‌شود. این الگوریتم با ترکیب جستجوی محلی و جهانی، به بهبود تدریجی راه حل‌ها و جلوگیری از گیر افتادن در بهینه محلی کمک می‌کند.

راه حل الگوریتم زنبور عسل

فرض کنید می‌خواهیم بهترین مکان برای پیدا کردن گل‌های بیشتر را پیدا کنیم (شبیه مسئله بهینه‌سازی).

1. شروع با نقاط اولیه

چند مکان (نقطه) تصادفی را انتخاب می‌کنیم. این نقاط مثل جاهایی هستند که زنبورها می‌توانند گل‌ها را جستجو کنند.

2. ارزیابی نقاط

تعداد گل‌های هر مکان را می‌شماریم. مثلاً:

○ نقطه A: 5 گل

○ نقطه B: 8 گل

○ نقطه C: 2 گل

3. انتخاب نقاط خوب

مکان‌هایی که بیشترین گل را دارند (مثل B و A) انتخاب می‌کنیم.

4. فرستادن زنبورها به نقاط خوب

زنبورهای بیشتری به نقاط خوب (مثل B) می‌فرستیم تا در اطراف آن مکان جستجو کنند و ببینند آیا نقطه‌ای با گل بیشتر پیدا می‌شود یا نه.

5. کاوش نقاط جدید

تعدادی زنبور دیگر را به مکان‌های کاملاً جدید می‌فرستیم (کاوش جهانی) تا شاید مناطق پرگل جدیدی پیدا کنند.

6. به‌روزرسانی نقاط

اگر زنبورها در نقاط اطراف یا مکان‌های جدید گل‌های بیشتری پیدا کردند، آن نقاط را جایگزین نقاط قدیمی می‌کنیم.

7. تکرار فرآیند

مراحل بالا را بارها تکرار می‌کنیم تا بهترین مکان ممکن (با بیشترین گل) پیدا شود.

مسئله ی هشت وزیر

تعریف مسئله و مقداردهی اولیه

- هر راه‌حل یک چیدمان از 8 وزیر روی صفحه شطرنج است (مثلاً: هر وزیر در یک ستون قرار دارد و موقعیت آن در هر ردیف مشخص می‌شود).
- تابع هدف: (Fitness Function) تعداد برخوردهای بین وزیرها (هدف ما کمینه کردن این مقدار است، یعنی به صفر رسیدن برخوردها).
- تعداد زنبورهای کارگر و زنبورهای دیده‌بان را مشخص می‌کنیم.
- تعدادی چیدمان اولیه به صورت تصادفی تولید می‌کنیم.

ارزیابی اولیه

- برخوردهای هر چیدمان را محاسبه می‌کنیم.
- چیدمان‌هایی که برخورد کمتری دارند، به عنوان چیدمان‌های برتر (Elite Solutions) انتخاب می‌شوند.

جستجوی محلی توسط زنبورهای کارگر

- برای هر چیدمان برتر، زنبورهای کارگر شروع به جستجوی محلی می‌کنند.
 - جابه‌جایی وزیرها در ستون‌های مختلف انجام می‌شود (مثلاً وزیر ردیف اول را به ستون دیگری منتقل می‌کنیم).
 - برخوردهای جدید را محاسبه می‌کنیم و چیدمان بهتری را جایگزین می‌کنیم.

جستجوی جهانی توسط زنبورهای دیده‌بان

- زنبورهای دیده‌بان در نقاط تصادفی دیگر فضای جستجو به دنبال چیدمان‌های کاملاً جدید هستند.
- این جستجو به جلوگیری از گیر افتادن در نقاط بهینه محلی کمک می‌کند.

به روزرسانی چیدمان ها

- اگر چیدمان های بهتری پیدا شدند، جایگزین چیدمان های قدیمی می شوند.
- زنبورهای بیشتری به چیدمان های خوب اختصاص داده می شوند تا جستجوی دقیق تری انجام دهند.

تکرار فرآیند

- این مراحل را برای چندین تکرار انجام می دهیم.
- وقتی به چیدمانی برسیم که تعداد برخوردها صفر باشد، راه حل مناسب پیدا شده است.

مثال عددی برای توضیح:

1. شروع با چیدمان تصادفی:

$[2, 4, 6, 8, 3, 1, 7, 5]$ $[2, 4, 6, 8, 3, 1, 7, 5]$ $[2, 4, 6, 8, 3, 1, 7, 5]$
هر عدد نشان دهنده ستون وزیر در هر ردیف است.
تعداد برخوردها:

2. انتخاب چیدمان های برتر:

چیدمان های دارای برخورد کمتر (مثلاً برخوردهای 3 یا 2) به عنوان نقاط برتر انتخاب می شوند.

3. جستجوی محلی:

در یک چیدمان برتر:

$[2, 4, 6, 8, 3, 1, 7, 5]$ $[2, 4, 6, 8, 3, 1, 7, 5]$ $[2, 4, 6, 8, 3, 1, 7, 5]$
جابه جایی وزیر در ردیف سوم به ستون 5 باعث کاهش برخوردها می شود:
 $[2, 4, 5, 8, 3, 1, 7, 5]$ $[2, 4, 5, 8, 3, 1, 7, 5]$ $[2, 4, 5, 8, 3, 1, 7, 5]$
تعداد برخوردها:

4. جستجوی جهانی:

تولید چیدمان کاملاً جدید:

$[1, 3, 5, 7, 2, 4, 6, 8]$ $[1, 3, 5, 7, 2, 4, 6, 8]$ $[1, 3, 5, 7, 2, 4, 6, 8]$
تعداد برخوردها:

5. به روزرسانی:

چیدمان های بهتر جایگزین چیدمان های قبلی می شوند و تکرار ادامه می یابد.

نتیجه نهایی:

بعد از چندین تکرار، به چیدمانی می رسیم که هیچ برخوردی ندارد:

$[1, 5, 8, 6, 3, 7, 2, 4]$ $[1, 5, 8, 6, 3, 7, 2, 4]$ $[1, 5, 8, 6, 3, 7, 2, 4]$
این یک راه حل بدون برخورد برای مسئله هشت وزیر است.

گراف رنگ آمیزی چیست؟

گراف رنگ آمیزی یک مسئله در ریاضیات گسسته و علوم کامپیوتر است که هدف آن تخصیص رنگ به رأس های گراف به گونه ای است که:

- هیچ دو رأس مجاور رنگ یکسان نداشته باشند (شرط همسایگی).
- تعداد رنگ ها حداقل باشد (هدف اصلی مسئله).

مثال ساده از مسئله گراف رنگ آمیزی:

فرض کنید یک گراف با 4 رأس داریم:

- رأس AAA به BBB و CCC وصل است.
- رأس BBB به AAA و DDD وصل است.
- رأس CCC به AAA و DDD وصل است.
- رأس DDD به BBB و CCC وصل است.

هدف: کمترین تعداد رنگ برای رنگ آمیزی این گراف را پیدا کنیم.

حل مسئله گراف رنگ آمیزی با الگوریتم زنبور عسل

1. تعریف مسئله و مقداردهی اولیه:

- هر راه حل یک توزیع رنگ برای رأس ها است.
 - مثال $[1,2,1,2]$ [1, 2, 1, 2]: رأس ها به ترتیب با رنگ های 1 و 2 رنگ آمیزی شده اند.
- تابع هدف:
 - تعداد رأس هایی که شرط همسایگی را نقض می کنند.
 - هدف: رساندن این مقدار به صفر.
- تعداد زنبورهای کارگر و دیده بان تعیین می شود.
- چندین راه حل اولیه به صورت تصادفی تولید می شود.

2. ارزیابی راه حل های اولیه:

راه حل ها ارزیابی می شوند و تعداد نقض های شرط همسایگی محاسبه می شود:

- $[1,2,1,2]$ [1, 2, 1, 2]: 0 نقض
- $[1,1,2,2]$ [1, 1, 2, 2]: 2 نقض
- $[2,2,1,1]$ [2, 2, 1, 1]: 2 نقض

راه حل های برتر با کمترین نقض انتخاب می شوند.

3. جستجوی محلی توسط زنبورهای کارگر:

- برای هر راه حل برتر، زنبورها رنگ برخی از رأس ها را تغییر می دهند.
 - مثال $[1,1,2,2]$ [1, 1, 2, 2]:
 - رنگ رأس AAA از 1 به 2 تغییر داده می شود. $[2,1,2,2]$ [2, 1, 2, 2]:

▪ نقض کاهش می‌یابد.

4. جستجوی جهانی توسط زنبورهای دیده‌بان:

- زنبورهای دیده‌بان راه‌حل‌های کاملاً جدید تولید می‌کنند.
 - مثال $[1,2,2,1][1,2,2,1][1,2,2,1]$:
- ارزیابی می‌شود و ممکن است بهبود یابد.

5. به‌روزرسانی راه‌حل‌ها:

- بهترین راه‌حل‌های پیدا شده جایگزین راه‌حل‌های قبلی می‌شوند.
- اگر نقضی باقی نمانده باشد، حل مسئله تمام می‌شود.

6. تکرار فرآیند:

مراحل بالا تا زمانی که گراف بدون نقض رنگ‌آمیزی شود، تکرار می‌شوند.

نتیجه:

در نهایت، برای گراف مثال بالا، یکی از راه‌حل‌های بهینه:

$[1,2,1,2][1,2,1,2][1,2,1,2]$

این راه‌حل از 2 رنگ استفاده می‌کند و هیچ دو رأس مجاور رنگ یکسان ندارند.

هدف از استفاده الگوریتم زنبور عسل: این روش ترکیبی از جستجوی محلی (بهبود تدریجی) و جستجوی جهانی (کاوش مناطق جدید) است که کمک می‌کند بهترین راه‌حل بهینه با کمترین رنگ‌ها پیدا شود.