باسمه تعالی

موضوع :‌

الگوریتم تپه نوردی

حل مسئله ی هشت وزیر

پاییز ۱۴۰۳ – آبان ماه

درس هوش مصنوعی – دکتر مهدی مصلح

حسین کوشکیان

فهرست

[مقدمه 3](#_Toc184501412)

[الگوریتم تپه نوردی 3](#_Toc184501413)

[حل مسئله ی هشت وزیر 5](#_Toc184501414)

# مقدمه

الگوریتم تپه‌نوردی یکی از روش‌های ساده در هوش مصنوعی است که برای پیدا کردن جواب یک مسئله استفاده می‌شود. ایده اصلی این الگوریتم این است که همیشه به سمت جواب بهتر حرکت کنیم، مثل کسی که برای رسیدن به قله یک تپه، قدم به قدم به سمت بالا می‌رود.

**مسئله هشت وزیر** یک مسئله شناخته‌شده است که در آن باید هشت وزیر شطرنج را روی یک صفحه ۸×۸ طوری بچینیم که هیچ‌کدام نتوانند همدیگر را تهدید کنند. پیدا کردن یک چیدمان درست برای این مسئله، در ظاهر ساده ولی در عمل چالش‌برانگیز است.

در این ارائه، ابتدا الگوریتم تپه‌نوردی و نحوه کار آن را توضیح می‌دهیم و بعد بررسی می‌کنیم که چطور می‌توان از این روش برای حل مسئله هشت وزیر استفاده کرد. هدف ما این است که این مفاهیم را به زبانی ساده و کاربردی ارائه کنیم.

# الگوریتم تپه نوردی

الگوریتم تپه‌نوردی (Hill Climbing) یک روش جستجوی محلی است که برای حل مسائل بهینه‌سازی استفاده می‌شود. این الگوریتم با شروع از یک نقطه اولیه و حرکت به سمت وضعیت‌هایی با مقدار بهتر (بر اساس تابع هدف) عمل می‌کند. در اینجا، مفهوم را با یک مثال ملموس توضیح می‌دهیم:

**مثال: پیدا کردن بلندترین نقطه در یک تپه**

تصور کنید که روی یک تپه قرار دارید و می‌خواهید بلندترین نقطه آن را پیدا کنید. شما فقط می‌توانید وضعیت ارتفاع خود را در هر لحظه ارزیابی کنید. الگوریتم به این شکل عمل می‌کند:

1. **شروع از یک نقطه تصادفی:** حرکت از یک نقطه اولیه روی تپه.
2. **بررسی همسایگی:** اطراف خود را بررسی می‌کنید تا ببینید کدام مسیر شما را به ارتفاع بالاتری می‌برد.
3. **حرکت به بهترین انتخاب:** به سمت مسیری حرکت می‌کنید که بیشترین افزایش ارتفاع را دارد.
4. **توقف در نقطه اوج:** وقتی به جایی برسید که هیچ مسیر اطراف شما ارتفاع بیشتری نداشته باشد، الگوریتم متوقف می‌شود.

**مثال در مسئله هشت وزیر**

حالا این الگوریتم را برای حل **مسئله هشت وزیر** در نظر بگیرید. هدف این است که هشت وزیر را روی یک صفحه شطرنج قرار دهیم به‌طوری‌که هیچ‌کدام یکدیگر را تهدید نکنند. مراحل کار به این صورت است:

1. **حالت اولیه:** هشت وزیر را به‌صورت تصادفی روی صفحه قرار می‌دهیم.
2. **تابع هدف:** تعداد زوج‌های وزیر که همدیگر را تهدید می‌کنند، محاسبه می‌کنیم (هرچه این عدد کمتر باشد، وضعیت بهتر است).
3. **بررسی همسایگی:** وزیرها را به مکان‌های دیگر جابه‌جا می‌کنیم و وضعیت جدید را ارزیابی می‌کنیم.
4. **حرکت به وضعیت بهتر:** به حالتی می‌رویم که تعداد تهدیدها کاهش پیدا کند.
5. **توقف:** اگر به حالتی برسیم که هیچ وزیری دیگری را تهدید نکند (تابع هدف برابر صفر شود)، الگوریتم متوقف می‌شود.

**نقاط قوت و محدودیت‌ها**

الگوریتم تپه‌نوردی سریع و ساده است، اما ممکن است در نقاطی متوقف شود که بهترین جواب نیستند (قله فرعی). برای غلبه بر این محدودیت، می‌توان از روش‌هایی مانند **تپه‌نوردی تصادفی** یا **جستجوی شبیه‌سازی‌شده (Simulated Annealing)** استفاده کرد.

این الگوریتم به دلیل سادگی و کاربردی بودن، در حل بسیاری از مسائل بهینه‌سازی مورد

توجه قرار گرفته است.

# حل مسئله ی هشت وزیر

**مسئله هشت وزیر** یک مسئله کلاسیک است که هدف آن قرار دادن هشت وزیر روی یک صفحه شطرنج ۸×۸ است به‌طوری که هیچ وزیری نتواند دیگری را تهدید کند. وزیر شطرنج می‌تواند در تمام جهات (سطر، ستون و قطرها) حرکت کند، بنابراین در یک چینش صحیح، هیچ دو وزیری نباید در یک سطر، یک ستون یا یک قطر مشترک باشند.

برای حل این مسئله با استفاده از الگوریتم تپه‌نوردی، مراحل را گام به گام توضیح می‌دهیم

**۱. حالت اولیه (Initial State)**

برای شروع، باید یک چیدمان اولیه از هشت وزیر روی صفحه داشته باشیم. برای ساده‌تر کردن مسئله:

* هر ستون دقیقا یک وزیر دارد.
* جایگاه وزیر در هر ستون به‌صورت تصادفی روی یکی از سطرها قرار داده می‌شود.

این روش اطمینان می‌دهد که هیچ وزیری در یک ستون مشترک نیست، اما ممکن است در یک سطر یا قطر مشترک باشند که تهدید ایجاد می‌کند.

**۲. تعریف تابع هدف (Heuristic Function)**

تابع هدف در این مسئله تعداد زوج وزیرهایی است که یکدیگر را تهدید می‌کنند.

* اگر دو وزیر در یک سطر باشند، یک تهدید است.
* اگر دو وزیر در یک قطر (قطر اصلی یا فرعی) باشند، یک تهدید دیگر محسوب می‌شود.

**هدف الگوریتم این است که تابع هدف را به صفر برساند، یعنی هیچ تهدیدی وجود نداشته باشد.**

**۳. بررسی همسایگی (Neighbor States)**

برای بهبود وضعیت فعلی، باید تغییراتی در صفحه ایجاد کنیم تا وضعیت بهتری به دست آید.

* در هر گام، یک وزیر را در ستون خودش به یک سطر دیگر منتقل می‌کنیم (وزیر را جابه‌جا می‌کنیم).
* این تغییر برای هر وزیر و هر سطر ممکن انجام می‌شود و تمامی حالات همسایگی ایجاد می‌شود.

مثال: اگر وزیر اول در ستون اول و سطر ۳ باشد، حالات همسایگی شامل جابه‌جایی آن به سطرهای ۱، ۲، ۴، ۵، ۶، ۷ و ۸ است.

**۴. انتخاب بهترین حرکت (Best Move Selection)**

* تمامی حالت‌های همسایگی بررسی می‌شوند.
* حالتی که تعداد تهدیدها (تابع هدف) را کاهش دهد، انتخاب می‌شود.
* اگر وضعیت جدید بهتر از وضعیت فعلی باشد، به آن حرکت می‌کنیم.

**مثال:**  
اگر تابع هدف در وضعیت فعلی برابر ۴ باشد (۴ جفت وزیر یکدیگر را تهدید می‌کنند) و پس از جابه‌جایی وزیر اول به یک سطر دیگر، تعداد تهدیدها به ۲ کاهش یابد، این حرکت انتخاب می‌شود.

**۵. شرط توقف (Stopping Condition)**

الگوریتم در دو حالت متوقف می‌شود:

1. وقتی تابع هدف برابر صفر شود، یعنی هیچ وزیری دیگری را تهدید نکند (حل مسئله).
2. وقتی به حالتی برسیم که هیچ حرکت دیگری وضعیت را بهبود ندهد (قله محلی).

**مشکل قله محلی (Local Maxima)**

گاهی الگوریتم به وضعیتی می‌رسد که هیچ بهبودی ممکن نیست، اما مسئله هنوز حل نشده است. در این حالت به قله‌ای فرعی رسیده‌ایم، نه قله اصلی. برای رفع این مشکل، می‌توان از روش‌های زیر استفاده کرد:

1. **شروع مجدد تصادفی:** الگوریتم را از یک حالت اولیه جدید شروع می‌کنیم.
2. **تپه‌نوردی تصادفی:** در برخی موارد، به جای انتخاب بهترین حرکت، به‌صورت تصادفی یکی از همسایگی‌ها را انتخاب می‌کنیم.
3. **ترکیب با جستجوی شبیه‌سازی‌شده:** از روش‌های پیشرفته‌تری مانند Simulated Annealing استفاده می‌شود که به خروج از قله محلی کمک می‌کند.

منابع :

GPT – Open AI