به نام خدا

الگوریتم SA

این الگوریتم از نوع محلی است، یعنی ما فقط میتوانیم در اطراف جایی که هستیم جستجو را انجام دهیم، به این شکل که در حالتی که هستیم یک فرزند را به صورت تصادفی انتخاب یا تولید میکنیم، و چک میکنیم اگر این حالت جدید وضعیت حالت قبلی یا پدر یا بهبود میبخشید، آن را گسترش میدهیم درغیر اینصورت مقدار e^(delta(E) / T) حساب میکنیم که یک مقدار بین صفر و یک است. و بعد یک عدد تصادفی بین یک و صفر انتخاب میکنیم که اگر بین مقدار حساب شده و صفر بود آنگاه این حالت را گسترش میدهیم، در غیر اینصورت به سراغ حالت دیگری میرویم. لازم به ذکر است در هر مرحله که حالت جدیدی را تصادفی تولید میکنیم از مقدار T کم میشود و این الگوریتم تا زمانی که T مخالف صفر و بزرگتر است ادامه پیدا میکند. برای مسأله های کمینه سازی مقدار بالا به صورت e^(-delta(E) / T) محاسبه میشود.

شیوه پیاده سازی

برای پیاده سازی این پروژه کلاس ها و متد هایی به شرح زیر پیاده سازی شده اند.

1. State (class)

a. getEvaluation (method)
b. moveQueen (method)
c. getQueenPosition (method)

2. SimulatedAnnealing (class)
a. saAlgoritm (method)
b. randomStart (method)
c. randomState (method)
d. schedule (method)
e. isNewStateValid (method)

مورد اول برای ساختن حالت ها استفاده میشود و شامل یک ارایه دو در دو به عنوان صفحه شطرنج، یک مقدار به عنوان مقدار Evaluation، و یک لیست از سطر و ستون جایگاه وزیر ها در صفحه شطرنج است و همچنین شامل متد هایی برای جابجای وزیر ها و تولید حالت جدید و چاپ کردن صفحه شطرنج و وضعیت وزیرها در آن حالت است.

مورد دوم یک کلاس است که مقدار T و حالتی که فعلا در حال گسترش است را در خود نگه میدارد و همچنین شامل متد هایی است برای اجرای الگوریتم SA، تولید حالت اولیه به صورت

تصادفی، تولید حالت فرزند به صورت تصادفی، چک کردن گسترش یا عدم گسترش حالت تصادفی و کاهش مقدار T.

فرایند کار به این صورت است که ابتدا یک حالت تصادفی توسط متد randomStart برای شروع تولید میشود و سپس اجرای الگوریتم در متد saAlgoritm شروع میشود، ابتدا چک میکنیم ایا به هدف رسیده ایم یا نه اگر رسیده بودیم اجرای الگوریتم متوقف شده و حاصل چاپ میشود در غیر اینصورت یک فرزند به صورت تصادفی توسط متد randomState تولید میشود، و بررسی میشود که ایا برای گسترش دادن معتبر است یا نه (در متد isNewStateValid، یعنی بررسی میشود که ایا گره تولید شده گسترش داده شود یا نه. سپس گره جدید در صورت پایان بررسی به صورت موفق، با گره قبلی جایگزین میشود و مقدار T کاهش توسط متد schedule میابد و فرایند برای گره جدید تکرار میشود، در غیراینصورت مقدار T کاهش میابد و فرایند دوباره اجرا میشود تا فرزند تصادفی دیگری تولید شود. فرایند بالا تا زمانی ادامه پیدا میکند که یا به یک حالت جواب برسیم یا مقدار T به صفر برسد.