



سؤال ۱. Simulated Annealing

الگوریتم Simulated Annealing به این صورت عمل می‌کند که برای دماهای بالا، جست‌وجوی ش کاملاً تصادفی است. هر چه دما را بیش‌تر کاهش بدهیم، فرآیند جست‌وجو بیش از پیش دقیق می‌شود. بنابراین، دو چیز در این فرآیند بسیار مهم است:

۱. شروع با دمای بالا (برای ارزیابی ویژگی‌های ناخالص تابع هدف) و تکامل آن به صورتی که به دمای پایین‌تر برسد.

۲. با توجه به طبیعت تصادفی بودن این الگوریتم، به‌طوری تعریف شده است که احتمال حرکت رو به پایین (هر چند با مقدار اندک بزرگ‌تر از صفر) وجود دارد. این عامل سبب می‌شود تا احتمال گیر کردن الگوریتم در ماکسیمم‌های محلی را تا حدی از بین می‌برد. چون همیشه احتمال بیرون پریدن از آ وجود دارد.

- الف) اگر دما را به سرعت کاهش دهیم، باعث می‌شود که فرآیند بازپخت به خوبی صورت نگیرد. این عمل‌کرد باعث می‌شود تا به یک وضعیت suboptimal برسیم که می‌تواند یک ماکسیمم محلی یا یک سطح صاف باشد.
- ب) اگر دمای مثبت اولیه مقداری کوچک باشد، می‌تواند منجر به محدود شدن فضای مدل در نقطه‌ی شروع شود. اما اگر مقدار آن بزرگ باشد باعث افزایش تعداد random walk ها و iteration ها می‌شود.
- ج) اولین node که همان current است را برمی‌گرداند و الگوریتم به پایان می‌رسد.

سؤال ۲. Hill Climbing

- (آ) 2^n
- (ب) n
- (ج) هدف: مجموعه‌ی همه‌ی Conjunction of Disjunctions مقدارش True شود. به‌طور دقیق‌تر برای هر State تعداد عبارت‌هایی که True (یا Satisfy) شده‌اند.
- (د) در این مثال، اگر به کدام از پرانتزها (از چپ به راست) عددی از ۱ تا ۵ نسبت دهیم، مقداری که برای هر کدام از آنها به دست می‌آید، به شرح زیر است:

$$1 \rightarrow False, 2 \rightarrow True, 3 \rightarrow True, 4 \rightarrow True, 5 \rightarrow False$$

با توجه به قسمت (ج)، هدف ما این است که مقدار همه‌ی آنها را صحیح کنیم. بنابراین به تصادف باید مقدار یکی از متغیرهای A یا D را برابر با True قرار دهیم تا به حالت بعدی برویم.

• (ه)

سؤال ۳. Genetic Algorithm

مدل‌سازی: برای مدل کردن این مسئله با الگوریتم ژنتیک، یک رشته‌ی به طول n خواهیم داشت (که در آن n ، تعداد رأس‌های گراف است). عدد «۱» در جایگاه i ام، نشان‌دهنده‌ی حضور رأس i در مجموعه‌ی Vertex Cover است و عدد «۰» به منزله‌ی عدم حضور آن.

تابع fitness: تفاضل تعداد کل رأس‌ها از تعداد رأس‌هایی که عضو Vertex Cover نیستند و مقدار «۱» دارند.
تابع selection: همانند آنچه که در اسلاید شماره «۵» درس برای n -Queens داشتیم، چون در این جا با رشته‌ها کار می‌کنیم، یک عدد در محدوده‌ی ۱ تا $n-1$ انتخاب می‌کنیم و نام آن را x می‌گذاریم. با توجه به اعدادی (درصدهایی) که به کمک تابع fitness به دست آوردیم، دو به دو دسته‌بندی می‌کنیم.

تابع cross-over: همانند آنچه که در اسلاید شماره «۵» درس برای n -Queens داشتیم، از ۱ تا x کاراکتر از رشته‌ی اول را $x+1$ تا n رشته‌ی دوم را جابه‌جا می‌کنیم. دو رشته‌ی جدید از ترکیب به دست می‌آید.

تابع mutation: همانند آنچه که در اسلاید شماره «۵» درس برای n -Queens داشتیم، یکی از کاراکترها را تصادفی تغییر می‌دهیم.

سؤال ٤ . Local Search

- (آ)
- (ب)