

1. مسئله N-Queen با استفاده از الگوریتم ژنتیک

حل مسئله N-Queen با استفاده از الگوریتم ژنتیک

هدف N-Queens Problem قرار دادن N ملکه در صفحه شطرنج N x N است ، به گونه ای که هیچ ملکه ای با دیگران در تضاد نباشد.

توضیحات:

ژن: یک فرد با مجموعه ای از متغیرها مشخص می کند. در واقع مجموعه وزیرها هست.

کروموزوم: ژن ها به یک رشته متصل می شوند و یک کروموزوم (محل) تشکیل می دهند. کروموزوم مجموعه ای از پارامترها است که راه حل پیشنهادی برای مسئله ای را تعریف می کند که الگوریتم ژنتیک با آن حل میشود.

جمعیت: مجموعه تمام راه حل ها

عملکرد تناسب اندام: جفت وزیرهای غیر حمله کننده) مثلاً برای $N = 6$ ، $F_{max} = 6C2 = 6 * 5/2 = 15$

Crossover: همچنین ترکیب که، یک عملگر ژنتیکی است که برای ترکیب اطلاعات ژنتیکی دو والدین برای تولید فرزندان جدید استفاده می شود. در اینجا برای حرکات وزیرها هست.

جهش: یک یا چند مقدار ژن را از حالت اولیه در یک کروموزوم تغییر می دهد

چگونه الگوریتم ژنتیک مسئله n-queen را حل می کند؟

مرحله 1: یک کروموزوم تصادفی تولید می شود

مرحله 2: مقدار تناسب اندام کروموزوم محاسبه می شود

مرحله 3: اگر تناسب اندام با F_{max} برابر نیست

مرحله 4: تولید کروموزوم جدید (کراس اوور) از بهترین کروموزوم تصادفی انتخاب شده

مرحله 5: جهش ممکن است اتفاق بیفتد

مرحله ششم: کروموزوم جدید به جمعیت اضافه شده است

مرحله 2 تا 6 را تکرار کنید تا یک کروموزوم (محلول) با مقدار $Fitness = F_{max}$ پیدا شود.

2. الگوریتم شبیه سازی شده یک الگوریتم بهینه سازی جستجو جهانی تصادفی است.

که از تصادفی بودن به عنوان بخشی از فرآیند جستجو استفاده می کند. این الگوریتم را برای توابع هدف غیرخطی مناسب می کند در حالی که دیگر الگوریتم های جستجوی محلی به خوبی کار نمی کنند.

مانند الگوریتم جستجوی محلی صعود از تپه تصادفی ، این یک راه حل واحد را اصلاح می کند و منطقه نسبتاً محلی فضای جستجو را جستجو می کند تا زمانی که اپتیمای محلی واقع شود. برخلاف الگوریتم تپه نوردی ، ممکن است راه حل های بدتر را به عنوان راه حل فعلی بپذیرد.

احتمال پذیرش راه حل های بدتر از ابتدای جستجو بسیار زیاد شروع می شود و با پیشرفت جستجو کاهش می یابد و به الگوریتم این فرصت را می دهد که ابتدا منطقه را برای بهینه جهانی تعیین کند ، از اپتیمای محلی فرار کند ، سپس از خود به سمت اپتیمای صعود کند.

در الگوریتم شبیه سازی بهینه سازی شبیه سازی شده برای بهینه سازی است.

در تابع `make_move` ، بازنویسی یک همسایه تصادفی را از کل فضای جستجو انتخاب کردیم. که چقدر خوب عمل می کند تا در یک محله محلی گسترش یافته و از یک نقطه فعلی جستجو کند.

و تابع رندم استفاده کردیم تا نقاط محلی و بهینه را پیدا کند.

.