

حل مسئله نجات گروگان - محمدپویا تراشی - ۴۰۱۵۲۱۱۰۲

۱. تپهنوردی (Hill Climbing)

جزئیات پیاده‌سازی:

- موقعیت فعلی بازیکن با هر خانه همسایه مقایسه می‌شود تا خانه‌ای که کمترین فاصله منتهن (جمع فاصله‌های افقی و عمودی) را با گروگان دارد پیدا شود.
- تابع `choose_successor` مسئول انتخاب بهترین موقعیت بعدی از بین همسایگان است.
- اگر همسایه انتخاب شده هزینه کمتری (یعنی نزدیکتر به گروگان) نسبت به موقعیت فعلی داشته باشد، بازیکن به آنجا حرکت می‌کند. در غیر این صورت، بازیکن در مکان فعلی باقی می‌ماند.

- مزایا: ساده و مؤثر زمانی که مسیر واضحی وجود داشته باشد.

- محدودیت‌ها: احتمال گیر افتادن در مینیمم‌های محلی، به ویژه اگر اطراف بازیکن موانعی وجود داشته باشد که مانع از بهبود بیشتر شود، نسبتاً بالاست که در این حل با استفاده از حرکتهای تصادفی به مسیر خود ادامه می‌دهد.

۲. Simulated Annealing

جزئیات پیاده‌سازی:

- پارامتر دما (Temperature): در ابتدا حرکات به خانه‌های با امتیاز کمتر را برای کمک به فرار از مینیمم‌های محلی مجاز می‌کند.
- نرخ سرد شدن (Cooling rate): به تدریج دما را کاهش می‌دهد، به طوری که الگوریتم بیشتر شبیه به تپهنوردی می‌شود و صرفاً حرکات مطمئن‌تر را انتخاب می‌کند.
- یک موقعیت تصادفی از همسایگان انتخاب می‌شود و هزینه آن با موقعیت فعلی مقایسه می‌شود. اگر موقعیت جدید هزینه کمتری داشته باشد، انتخاب می‌شود. در غیر این صورت، ممکن است با توجه به تابع نمایی موجود که با کاهش دما کمتر می‌شود احتمال پذیرش کمتر می‌شود، پذیرفته شود.
- مزایا: کمک می‌کند تا از مینیمم‌های محلی با اجازه دادن به حرکات با هزینه بالاتر در ابتدا فرار کنیم.
- محدودیت‌ها: عملکرد آن به شدت به نرخ سرد شدن و دمای اولیه وابسته است و ممکن است همیشه به راه‌حل بهینه منتهی نشود. همچنین برای این سوال با توجه به سادگی مسیر عملکرد بهتری نسبت به تپهنوردی ندارد.

۳. الگوریتم ژنتیک (Genetic Algorithm)

جزئیات پیاده‌سازی:

- تولید جمعیت: جمعیت اولیه مسیرها با استفاده از جستجوی DFS تولید می‌شود. هر مسیر نماینده یک راه‌حل است و در ادامه الگوریتم سعی می‌شود این نسل‌های اولیه کوتاه‌تر و بهینه شوند.
- تابع تناسب (Fitness function): تناسب یک مسیر به طور معکوس به طول آن بستگی دارد و به مسیرهای کوتاه‌تر اولویت می‌دهد.
- انتخاب: مسیرها (ژن‌ها) بر اساس توزیع احتمال که توسط تناسب آن‌ها تعیین می‌شود، انتخاب می‌شوند.
- Crossover: مسیرها با استفاده از نقاط مشترک یا همسایه ترکیب می‌شوند و فرزندی تولید می‌شود که ویژگی‌های هر دو والد را به ارث می‌برند و در صورت بهینه بودن هر دو والد، مسیر کوتاه‌تری می‌سازند. شانس این مسیرهای اولیه برای انتخاب به تابع تناسب آن‌ها بستگی دارد.
- جهش (Mutation): بررسی می‌کند اگر در یک مسیر امکان میان‌بر وجود داشت، آن را اعمال می‌کند و باعث جلوگیری از حرکات اضافی می‌گردد.
- مزایا: برای فضاهای مسأله پیچیده مؤثر است و راه‌حلی نزدیک به بهینه را در طول چندین نسل پیدا می‌کند.
- محدودیت‌ها: از نظر محاسباتی گران است و ممکن است نیاز به تنظیم پارامترهایی مانند نرخ جهش و اندازه جمعیت داشته باشد. همچنین در برخی موارد ممکن است تعداد کافی ژن برای نسل اولیه پیدا نشود و در کد موجود راه حلی برای این exception تعبیه نشده است و در صورت طول کشیدن پاسخ برنامه باید مجدد اجرا گردد.

سایر کمکی

۱. get_neighbors

این تابع خانه‌های همسایه (بالا، پایین، چپ، راست) برای یک موقعیت داده شده را تولید می‌کند و خانه‌هایی که خارج از محدوده یا حاوی موانع هستند را فیلتر می‌کند.

۲. in_loop

موقعیت‌های اخیر بازیکن را پیگیری می‌کند تا حلقه‌ها را شناسایی کند.

۳. random_move

یک حرکت تصادفی معتبر انجام می‌دهد تا از یک حلقه خارج شود یا از گیر افتادن در آن فرار کند.

لیستی از موقعیت‌های اخیر بازیکن را نگه می‌دارد تا الگوهای تکراری را شناسایی کند که نشان‌دهنده حلقه‌ها هستند.

پاسخ سوالات موجود در سند صورت مسئله، به ترتیب:

- در صورتیکه عملیات سرد کردن به اندازه کافی آهسته انجام گردد، حتما جواب حاصل می‌شود. اما در غیر این صورت به منظور فرار از نقطه اپتیمم‌های محلی استفاده از حرکات رندم نیاز است.
- در دمای بالا خیر، با توجه به اینکه این الگوریتم حرکات رندم را انجام می‌دهد نیازی به پیاده‌سازی انتخاب رندم وجود ندارد.
- در پیاده‌سازی موجود با توجه به اینکه اعضای نسل اولیه همگی پاسخ‌های این مسئله هستند این اتفاق نخواهد افتاد. در سایر پیاده‌سازی‌ها ممکن است نیاز باشد. البته وجود مواردی همچون جهش تاثیر مشابهی بر عملکرد الگوریتم دارند.
- وجود تابع جهش عملکرد مشابهی را دارد.
- به صورت کلی در الگوریتم تپه‌نوردی احتمال گیر افتادن بیشتر است.
- در تپه‌نوردی با حرکات رندم، در آنیلینگ شبیه‌سازی شده با سرد کردن تدریجی و اجازه به حرکات رندم و در الگوریتم ژنتیک با جهش و ترکیب کردن نسل‌ها. تپه‌نوردی با حرکات رندم ممکن است در صورت وجود موانع پیچیده، مدت زیادی را در محدوده اپتیمم محلی سپری کند اما در نهایت رندم بودن این حرکات و ذخیره مسیر پیموده شده کمک خواهد کرد نجات پیدا کند.
- در حالت عادی Genetic و Simulated Annealing در محیط‌های پیچیده که احتمال گیر افتادن در اپتیمم محلی بیشتر است عملکرد بهتری دارند.
- بسته به نوع بازی، در صورتی که گیر افتادن در لوکال اپتیمم اتفاق نیفتد، تپه‌نوردی بهترین عملکرد و کمترین پیچیدگی را دارد.
- به صورت کلی الگوریتم‌های آنیلینگ و ژنتیک با توجه به اینکه در در اپتیمم‌های محلی گیر نمی‌کنند، در نهایت پاسخ را به دست می‌آورند.
- در صورتی که نرخ سردسازی به اندازه کافی آهسته باشد الگوریتم آنیلینگ حتما جواب گلوبال اپتیمال را پیدا می‌کند. البته این هزینه محاسباتی و زمان را افزایش می‌دهد.

تعداد مراحل مورد نیاز برای رسیدن به جواب در الگوریتم‌های مختلف در ۵ بار اجرا

ژنتیک	آنیلینگ	تپه‌نوردی
۵۱	۲۱۹	۱۶
۸۴	۳۰۰	۱۲
۸۹	۲۶۷	۲۶۶
۶۷	۲۴۷	۲۰۸
۵۶	۵۹	۲۳