

می شود و در اواخر روند Simulated Annealing ، ممکن است به کار بباید. این مکانیزم به طور میانگین چه تاثیری روی الگوریتم میگذارد؟ وجود چنین مکانیزمی در این بازی نیاز است؟ تشخیص و جلوگیری از گیر افتادن در لوپ: مانند تپه نورده، این مکانیزم را اضافه کنید که بازیکن اگر در یک مسیر تکراری گیر کرد (یعنی در حال تکرار موقعیت های قبلی خود است)، یک حرکت تصادفی انجام دهد تا از لوپ خارج شود و بازی ادامه پیدا کند. به نظر شما این استراتژی در این الگوریتم استفاده خواهد شد؟ آیا ممکن است در این الگوریتم در لوپ گیر کنیم؟

جواب :

به طور کلی این الگوریتم هنگامی که دما بالاست حرکات تصادفی را با احتمال بالا انجام میدهد و در لوپ نمی افتد ولی هرچقدر که دما کمتر میشود پس احتمال تصادفی بودن حرکات نیز کمتر می شود و در نتیجه ممکن است در لوپ یا بهینه محلی بیفتیم و نیاز به یک حرکت رندوم باشد تا مارا از لوپ در بیاورد همانطور که گفته شد ممکن است در دماهای پایین در لوپ گیر کنیم.

اضافه کردن مکانیزم حرکت تصادفی در بن بست: در این مرحله، مکانیزم جهش در مسیرها را بهبود دهید تا اگر مسیری به بن بست خورد (یعنی بازیکن به نقطه ای رسید که نمی تواند به گروگان نزدیکتر شود)، جهش بیشتری در مسیر اتفاق بیفتد تا مسیر تصادفی جدیدی انتخاب شود. به نظر شما پیاده سازی چنین مکانیزمی نیاز است؟ این مکانیزم به طور میانگین باعث بهبود الگوریتم میشود؟

بعضی مواقع کروموزوم های که تولید میشوند مشابه هستند و در یک لوپ میفتقیم و نیاز است تا با یک حرکت رندوم از لوپ خارج شویم.(البته ساختار میوتیشن تا حد زیادی ما را از انجام این عمل بی نیاز می کند)

تشخیص و جلوگیری از گیر افتادن در لوپ: اگر در نسل های جدید، ژن ها (مسیرها) تمایل به تکرار داشتند یا بازیکن در یک مسیر لوپ گرفتار شد، با استفاده از جهش تصادفی مسیرها را تغییر دهید تا از تکرار مسیر های ناموفق جلوگیری شود. بازیکن باید بتواند از موقعیت های تنگ با انتخاب یک موقعیت همسایه تصادفی که توسط مانع اشغال نشده، فرار کند. فکر میکنید این مکانیزم به طور میانگین باعث بهبود الگوریتم میشود؟

بله.(در جواب سوال قبل توضیح داده شد)

• در کدام یک از این الگوریتم ها احتمال گیر افتادن در بهینه محلی بیشتر است؟

تپه نورده

احتمال گیر افتادن در بهینه محلی در همه الگوریتم های جستجو وجود دارد. در تپه نورده، به دلیل انتخاب حرکات بر اساس کمترین فاصله تا هدف، احتمال گیر افتادن در بهینه محلی بیشتر است. در مقابل، الگوریتم SA و ژنتیک به دلیل داشتن مراحل تصادفی (مثل پذیرش راه حل های ضعیفتر یا جهش) انعطاف بیشتری برای خروج از این حالت دارد.

• هر الگوریتم چگونه با این مشکل مقابله می کند؟ آیا تپه نورده با مکانیزم تصادفی به درستی از گیر افتادن در لوپ جلوگیری می کند؟

در الگوریتم تپه‌نوردی، می‌توان با استفاده از حرکات تصادفی یا تنظیم شرط توقف، از تکرار لوب جلوگیری کرد.
در الگوریتم SA تنظیم کاهش دما به طور تدریجی می‌تواند کمک کند تا به رامحل بهتری برسد و از گیر افتادن جلوگیری کند.
در الگوریتم ژنتیک، جهش‌های تصادفی می‌توانند مسیرهای متفاوتی ایجاد کند و از گیر افتادن در لوب جلوگیری کنند.

• کدام الگوریتم‌ها در یک محیط پیچیده و پر از موانع عملکرد بهتری دارند؟

الگوریتم ژنتیک به دلیل توانایی در تولید نسل‌های جدید با ویژگی‌های بهبودیافته، در محیط‌های پر از موانع عملکرد بهتری دارد.

کدام یک از الگوریتم‌ها سریع‌تر به یک نتیجه نهایی می‌رسد؟

سرعت رسیدن به نتیجه وایسته به شرایط محیط و تعداد موانع است، اما به طور کلی، تپه‌نوردی به دلیل ساده‌تر بودن روند خود در شرایط محیطی ساده‌تر، سریع‌تر به نتیجه می‌رسد. الگوریتم ژنتیک به دلیل پیچیدگی تولید نسل‌ها و جهش‌ها، کندتر از تپه‌نوردی است.
البته در محیط‌های پیچیده‌تر و دارای موانع بیشتر احتمال موقتیت الگوریتم ژنتیک بیشتر است.

کدام الگوریتم شناس بیشتری برای پیدا کردن بهینه جهانی دارد؟

الگوریتم ژنتیک به دلیل داشتن تنوع ژنتیکی و قابلیت ترکیب مسیرهای مختلف، شناس بیشتری برای رامحل بهینه‌ی جهانی دارد. Alg SA نیز می‌تواند با احتمال پذیرش حرکات ضعیفتر، از بهینه‌های محلی عبور کند ولی ژنتیک بهتره .

در هر الگوریتم، چگونه تنظیمات پارامترها (مانند نرخ جهش در الگوریتم ژنتیک یا دما در Simulated Annealing بر عملکرد آن تأثیر می‌گذارد؟ میتوانید این مورد را به صورت عملی پیاده‌سازی کرده و نتایج را باهم مقایسه کنید.

نرخ جهش در الگوریتم ژنتیک: افزایش نرخ جهش می‌تواند تنوع بیشتری در جمعیت ایجاد کرده و احتمال پیدا کردن مسیر بهینه را افزایش دهد، اما نرخ بسیار بالا ممکن است به نتایج ناپایدار و تصادفی منجر شود.

دما در Alg SA : تنظیم صحیح دما باعث می‌شود که الگوریتم بتواند در ابتدا حرکات غیر بهینه را پذیرد و به مرور با کاهش دما، به سمت بهینه نزدیک شود. دمای بالا منجر به پذیرش مسیرهای مختلف می‌شود، در حالی که دمای پایین الگوریتم را به انتخاب‌های قطعی‌تر سوق می‌دهد.
و نرخ کاهش دما هرچقدر کمتر باشد احتمال رسیدن به جواب بیشتر می‌شود. ولی اگر نرخ کاهش دما را از یک حد بیشتر کنیم منجر به ناپایداری می‌شود و احتمال زیاد به جواب نمیرسیم.

در داخل کد یک سری لاگ برای مقایسه این مفاهیم تعییه شده است و در صورت نیاز میتوانید از انها استفاده کنید.

در نهایت تعداد گام‌های لازم برای رسیدن به گروگان، زمان اجرای الگوریتم و تعداد برخورد با موانع برای 3 الگوریتم را مقایسه کنید.(در صورت نیاز میتوانید پارامترهای مربوط به ساخت بازی را نیز تغییر دهید).

(1) زمان اجرا

زمان اجرا نشان‌دهنده سرعت هر الگوریتم در رسیدن به جواب است.

- تپه‌نوردی > SA Alg > الگوریتم ژنتیک

یعنی:

- تپه‌نوردی سریع‌ترین الگوریتم است زیرا فقط به دنبال حرکات به سمت هدف است.
- سرد کردن شبیه‌سازی شده به دلیل استفاده از حرکات تصادفی در مراحل ابتدایی، سرعت کمتری دارد.

- الگوریتم ژنتیک به دلیل فرایندهای ترکیب و جهش و ارزیابی جمعیت، زمان اجرای بیشتری دارد.

2) تعداد برخورد با موانع

تعداد برخورد با موانع نشان می‌دهد که هر الگوریتم چقدر در اجتناب از موانع موفق است.

- الگوریتم ژنتیک > SA Alg > تپه‌نوردی

یعنی:

- الگوریتم ژنتیک کمترین تعداد برخورد با موانع را دارد، زیرا با تکامل کروموزوم‌ها به تدریج به مسیرهای بهینه و کم برخورد می‌رسد.
- SA Alg با پذیرش حرکات غیربهینه، امکان عبور از موانع بیشتری را دارد.
- تپه‌نوردی بیشترین برخورد را با موانع دارد، زیرا مستقیماً به دنبال نزدیکترین مسیر به هدف است و اگر مانع در مسیر باشد، به سختی می‌تواند از آن اجتناب کند.

3) تعداد گام‌ها تا رسیدن به گروگان

تعداد گام‌ها نشان می‌دهد که هر الگوریتم چه مقدار بهینه عمل می‌کند تا به گروگان برسد.

- تپه‌نوردی > الگوریتم ژنتیک > SA Alg

یعنی:

- تپه‌نوردی کمترین تعداد گام‌ها را در محیط‌های ساده دارد، زیرا با هر حرکت به سمت گروگان نزدیک می‌شود.
- الگوریتم ژنتیک به دلیل ایجاد مسیرهای بهینه‌تر از طریق جهش و ترکیب، می‌تواند به تعداد گام‌های کمتری نسبت به SA Alg دست پابد.
- SA Alg ، به دلیل پذیرش حرکات تصادفی، معمولاً به تعداد گام بیشتری نیاز دارد.

نتیجه کلی:

تپه‌نوردی سریع‌ترین الگوریتم است ولی در محیط‌های پیچیده تعداد برخورد بالایی دارد.
عملکردی میانگین دارد و برای محیط‌های پیچیده‌تر مناسب است.
SA Alg الگوریتم ژنتیک بهترین الگوریتم از نظر برخورد کم با موانع و یافتن مسیرهای بهینه در محیط‌های پیچیده است، اگرچه زمان بیشتری نیاز دارد.