

MCST (Monte Carlo Search Tree):

:استفاده از تصادفی برای انتخاب حرکت

از اصل تصادفی مونته کارلو استفاده می‌کند. با انتخاب تصادفی حرکت‌ها و شبیه‌سازی‌های چند بار بازی، MCST الگوریتم الگوریتم تقریباً بهترین حرکت را بر اساس احتمالات به دست می‌آورد.

:تصمیم‌گیری در شرایط پیچیده

به کمک شبیه‌سازی‌های تصادفی تصمیم‌گیری در شرایط MCST در بازی 2048 که ممکن است به سرعت پیچیده شود، الگوریتم پیچیده و پیش‌بینی حرکات آینده را تسهیل می‌کند.

Expectimax:

:مدل‌سازی احتمالات حرکات مختلف

از اصل احتمالاتی استفاده می‌کند و حرکات مختلف را با در نظر گرفتن احتمالات آنها مدل‌سازی می‌کند. این به Expectimax الگوریتم این امکان را می‌دهد که بر اساس احتمالات بهترین حرکت را انتخاب کند.

:انعطاف در تصمیم‌گیری

به بازیکن این امکان را می‌دهد که در تصمیم‌گیری‌هایش انعطاف داشته باشد. با در نظر گرفتن احتمالات، Expectimax الگوریتم بهترین تصمیم را با توجه به وضعیت فعلی بازی اتخاذ می‌کند.

در بازی 2048 به دلایلی انجام می‌شود. Expectimax یا MiniMax در الگوریتم‌های جستجوی مانند (Depth) محدود کردن عمق در زیر توضیحاتی به زبان فارسی در مورد این محدودیت‌ها آورده شده است:

:پیچیدگی محاسباتی

با افزایش عمق جستجو، تعداد حالات مختلفی که باید بررسی شوند نیز افزایش می‌یابد. این موضوع منجر به افزایش پیچیدگی محاسباتی می‌شود که ممکن است زمان بسیار زیادی برای تصمیم‌گیری در هر حرکت به ما اختصاص یابد. با محدود کردن عمق، می‌توان زمان اجرای الگوریتم را بهبود بخشید.

تبادل بین دقت و سرعت

محدود کردن عمق یک تضاد بین دقت تصمیم‌گیری و سرعت اجرای الگوریتم ایجاد می‌کند. به دلیل محدودیت منابع محاسباتی، توازن بین این دو مسئله بسیار مهم است. با افزایش عمق، دقت تصمیم‌گیری افزایش می‌یابد اما همزمان زمان اجرا افزایش می‌یابد.

اورفیتینگ (Overfitting):

در صورتی که عمق جستجوی الگوریتم بسیار زیاد باشد، الگوریتم ممکن است به تصمیم‌گیری بر اساس موارد خاص و نهایی برسد که ممکن است در واقعیت بهینه نباشد. این ریسک اورفیتینگ، یعنی تنظیم الگوریتم به‌طور خاص برای داده‌های آموزشی و نه داده‌های واقعی، با افزایش عمق جستجو بیشتر افزایش می‌یابد.