

سیستمی که مانند انسان رفتار می کند را با ذکر مثال تشریح کنید؟ یک مثال از یک سیستم هوش مصنوعی که مانند انسان رفتار 1 هستند. این سیستم ها به کمک الگوریتم های یادگیری عمیق و پردازش زبان (NLP) می کند، سیستم های پردازش زبان طبیعی طبیعی، قادرند به صورت خودکار و هوشمند به داده های متنی و گفتاری پاسخ دهند. به عنوان مثال، سیستم های تشخیص سخنرانی می توانند صداهای گفتار را تشخیص داده و به صورت خودکار متن آن را تبدیل کنند. همچنین، سیستم های ترجمه ماشینی نیز قادرند به صورت هوشمند و با استفاده از الگوریتم های یادگیری عمیق، متون را از یک زبان به زبان دیگر ترجمه کنند. این سیستم ها با استفاده از الگوریتم های پردازش زبان طبیعی و یادگیری عمیق، به صورت خودکار و هوشمند با داده های زبانی کار می کنند و

رفتار مانند انسان را نشان می دهند

هدف از تفکر عاقلانه چیست و چه آورده ای در پی خواهد داشت؟ هدف از تفکر عاقلانه در هوش مصنوعی، ایجاد سیستم هایی 2 است که قادر به تفکر، انتخاب و اتخاذ تصمیمات هوشمندانه با توجه به شرایط محیطی و وضعیت فعلی باشند. این نوع از هوش مصنوعی می تواند به ما در حل مسائل پیچیده، تصمیم گیری های استراتژیک و پیش بینی های دقیق کمک کند. به عنوان مثال، سیستم های تصمیم گیری هوشمند در حوزه مالی می توانند با استفاده از تفکر عاقلانه، به صورت خودکار و هوشمند سرمایه گذاری های موفق را شناسایی کرده و تصمیمات مناسب را اتخاذ کنند. این نوع از هوش مصنوعی، آورده های مهمی در پی خواهد داشت و

اجزای عامل و وظیفه عامل را با رسم شکل و تابع نویسی بررسی کنید؟ در هوش مصنوعی، عامل به معنای هر نوع سیستم 3 هوشمند است که قادر است با محیط ارتباط برقرار کند و به طور خودکار عمل کند. عامل می تواند یک ربات، یک نرم افزار یا حتی یک سیستم هوشمند مبتنی بر وب باشد. اجزای عامل عبارتند از

حسگرها: حسگرها وظیفه تهیه ورودی های حسی از محیط را دارند. آنها می توانند داده های مانند تصاویر، صداها، دما و فشار را دریافت کنند. حسگرها اطلاعات را به صورت دیجیتال یا آنالوگ به عامل ارائه می دهند. عملگرها: عملگرها وظیفه انجام اقدامات فیزیکی یا محاسباتی را دارند. آنها می توانند موتورها، فعال کننده ها، صفحه نمایش و سایر ابزارهایی باشند که عامل برای تعامل با محیط استفاده می کند. محیط: محیط شامل هر چیزی است که عامل در آن عمل می کند. محیط ممکن است فیزیکی یا مجازی باشد و می تواند شامل اشیاء، سایر عامل ها و سیستم های خارجی نیز باشد.

عملکرد: عملکرد عامل، تابعی است که مشخص می کند که عامل در مواجهه با ورودی های خاص چه اقداماتی را انجام دهد. به عبارت دیگر، عملکرد تعیین می کند که عامل چگونه به ورودی ها پاسخ دهد. برنامه: برنامه شامل الگوریتم ها و قوانینی است که عامل برای انجام وظایف خود استفاده می کند. برنامه می تواند به صورت قوانین قابل خواندن توسط انسان یا به صورت کدهای قابل اجرا توسط رایانه باشد.

دانش: دانش شامل اطلاعاتی است که عامل از محیط و تجربه های گذشته خود به دست می آورد. دانش می تواند شامل مدل هایی از محیط، قوانین و الگوریتم ها و تجربیات قبلی باشد. به طور کلی، توابع عامل به این صورت می تواند نشان داده شود

def agent_function(percept): state = update_state(percept) action = select_action(state) return action

وضعیت عامل را بر اساس update_state ورودی حسگر است که توسط عامل دریافت می شود. تابع percept در این تابع، عملیات مورد نیاز برای وضعیت فعلی را انتخاب می کند. سپس عمل انتخاب شده select_action به روز می کند و تابع percept

یک مدل است که در هوش مصنوعی به منظور توصیف وظایف PEAS را برای ربات فضانورد و فوتبالیست تشریح کنید؟ 4 Performance) و عملکرد عامل ها استفاده می شود. این مدل شامل چهار عنصر اصلی است که به ترتیب عبارتند از: عملکرد (Sensors) و حسگرها (Actuators)، عملگرها (Environment)، محیط (measure)

این عنصر مشخص می کند که عامل چه هدفی را در انجام وظایف خود دنبال می کند و (Performance measure) عملکرد عملکرد عامل را ارزیابی می کند. برای ربات فضانورد، عملکرد ممکن است شامل انجام مأموریت های فضایی مانند بررسی و بررسی سیارات، جمع آوری نمونه های سنگی، انجام آزمایشات علمی و حفظ ایمنی در فضا باشد. برای فوتبالیست، عملکرد می تواند شامل زدن گل، پاس دادن دقیق، دفاع موثر و کسب برتری در بازی ها باشد. محیط عبارت است از همه چیزی که عامل در آن عمل می کند. برای ربات فضانورد، محیط ممکن است (Environment) محیط شامل فضا، سیارات، سیستم های فضایی و اجسام مختلف دیگری مانند ماهواره ها و فضاپیماها باشد. برای فوتبالیست، محیط ممکن است شامل زمین فوتبال، تیم حریف، تماشاگران، توپ و سایر عوامل بازی باشد.

این عنصر شامل ابزارها و وسایلی است که عامل برای تعامل با محیط استفاده می‌کند. برای ربات (Actuators) عملگرها فضانورد، عملگرها ممکن است شامل موتورها، دستگاه‌های جمع‌آوری نمونه، دستگاه‌های اندازه‌گیری و دستگاه‌های مخابره باشد. برای فوتبالیست، عملگرها می‌توانند شامل پاها برای زدن گل، دستها برای پاس دادن و دستگاه‌های مخابره با هم‌تیمی‌ها باشند. این عنصر شامل حسگرهایی است که عامل برای دریافت اطلاعات از محیط استفاده می‌کند. برای ربات (Sensors) حسگرها فضانورد، حسگرها ممکن است شامل دوربین‌ها، سنسورهای حرارتی، سنسورهای شیمیایی و سایر حسگرها باشد. برای فوتبالیست، حسگرها می‌توانند شامل چشمان برای دریافت اطلاعات بصری، گوشها برای شنیدن صداها و پوست برای دریافت اطلاعات لمسی به ما کمک می‌کند تا وظایف و عملکرد ربات فضانورد و فوتبالیست را بهبود بخشیم و سیستم‌های PEAS باشند. استفاده از مدل

هوشمندی را برای آنها طراحی کنیم
 طبق شبکه‌کد زیر چرا عامل مبتنی بر جدول به شکست مواجه می‌شود؟ راهکارهای پیشنهادی خود را نام برده و مختصری در 5 خصوص هر کدام توضیح دهید؟ عامل مبتنی بر جدول به شکست می‌خورد زیرا در این حالت، عامل فقط بر اساس تاریخچه‌ی مشاهدات قبلی خود عمل می‌کند و هیچ‌گونه فرایند تصمیم‌گیری یا یادگیری دینامیک ندارد. این در حالی است که در بسیاری از مسائل واقعی، محیط پویا و تغییرپذیر است و عامل باید قادر به انطباق با تغییرات محیط باشد. برای رفع این مشکل و ارتقاء عامل مبتنی بر جدول، می‌توان از راهکارهای زیر استفاده کرد

، عامل را قادر به یادگیری SARSA و Q-learning استفاده از الگوریتم‌های یادگیری تقویتی: الگوریتم‌های یادگیری تقویتی، مانند راهحل بهینه برای مسئله می‌کنند. این الگوریتم‌ها بر اساس مفهوم پاداش و تنبیه عمل می‌کنند و با بهبود تصمیم‌گیری عامل، عملکرد آن را بهبود می‌بخشند

استفاده از شبکه‌های عصبی: شبکه‌های عصبی قادر به تعمیم داده‌ها و یادگیری الگوهای پیچیده هستند. با استفاده از این روش، عامل قادر به تصمیم‌گیری در مورد وضعیت‌های مختلف می‌شود و می‌تواند با تغییرات محیط سازگار شود
 با استفاده از تئوری احتمال و (MDP) استفاده از الگوریتم‌های تصمیم‌گیری مارکوف: الگوریتم‌های تصمیم‌گیری مارکوف تصمیم‌گیری بهینه، عامل را در تصمیم‌گیری‌های پیچیده راهنمایی می‌کنند. این الگوریتم‌ها مبتنی بر مدلی از محیط هستند و با استفاده از توابع ارزش و عملکرد، به عامل کمک می‌کنند تا راهحل بهینه را پیدا کند. با استفاده از این راهکارها، عامل مبتنی بر جدول قادر خواهد بود با محیط پویا و تغییرپذیر تعامل کند و عملکرد بهتری داشته باشد

حالت به وسیله مکان عامل و مکانهای کثیف تعیین (states): (دنیای جاروبرقی رباتوجه به فرموله سازی مسئله تشریح کنید؟ حالتها 6 حالت. $n \times n$ میشود. عامل در یکی از دو مکان است که هر کدام ممکن است کثیف باشند یا نباشند. پس $2 \times 2 \times 2 = 8$ حالت وجود دارد. یعنی در این محیط جاروبرقی فقط سه فعالیت میتواند انجام (action): حالت شروع: هر حالتی میتواند به عنوان حالت شروع باشد. فعالیتها دهد: 1: حرکت به سمت چپ 2: حرکت به سمت راست 3: عمل مکش از مون هدف: بررسی میکند آیا تمام مکانها تمیز است یا خیر. هزینه ی مسیر: تعداد مراحل موجود در مسیر، هزینه ی مسیر است

جستجوی عمقی را با رسم مرحله به مرحله شرح دهید و در نهایت کارایی الگوریتم را با چهار معیار اندازه‌گیری بیان کنید؟ 7
 یک الگوریتم جستجو در گراف است که با استفاده از استک به صورت عمق به جستجوی (Depth-First Search) جستجوی عمقی گره هدف می‌پردازد. در این الگوریتم، عملیات جستجو از ریشه شروع می‌شود و به صورت پیوسته به سمت پایین درخت جستجو حرکت می‌کند تا به گره هدف برسد یا دیگر قابلیت حرکت نداشته باشد. در هر مرحله، یک گره از استک خارج شده و اعمال ممکن در آن گره بررسی می‌شود. در صورتی که گره هدف یافت شود، جستجو متوقف می‌شود و در غیر این صورت، جستجو به عمق بیشتر ادامه می‌یابد. مراحل جستجوی عمقی به صورت زیر است

قرار دادن گره شروع در استک

تکرار این مراحل تا زمانی که استک خالی شود

گرفتن گره فعلی از استک

بررسی آیا گره فعلی گره هدف است یا خیر

در صورت برابر بودن گره فعلی با گره هدف، جستجو متوقف می‌شود و مسیر به گره هدف برگشت داده می‌شود

در غیر این صورت، تمام فرزندان گره فعلی را به استک اضافه می‌کنیم

در صورتی که استک خالی شود و گره هدف پیدا نشود، جستجو به نتیجه "ناموفق" می‌رسد. کارایی الگوریتم جستجوی عمقی می‌تواند با استفاده از چهار معیار اندازه‌گیری شود

زمان اجرا: زمانی که الگوریتم برای پیدا کردن گره هدف نیاز دارد

حافظه: میزان حافظه مورد نیاز برای ذخیره و مدیریت استک و گره‌های جستجو شده

پیچیدگی زمانی: تعداد گره‌ها و یال‌هایی که باید بررسی شوند تا به گره هدف برسیم

کیفیت جواب: میزان بهیگی جواب‌هایی که الگوریتم جستجوی عمقی به دست می‌دهد. با توجه به معیارهای فوق، الگوریتم جستجوی عمقی معمولاً در مسائلی که گراف جستجو از اندازه بزرگی برخوردار است و تعداد عمق جستجو کم است، کارایی خوبی دارد. اما در مسائلی که عمق جستجو بیشتر است و ممکن است در درخت جستجو به گره‌های تکراری برخورد کند، ممکن است به شکست ∞ برخورد و کارایی پایینی داشته باشد.

جست و جوی عمقی را با رسم مرحله به مرحله شرح دهید و در نهایت کارایی الگوریتم را با چهار معیار اندازه گیری بیان کنید ؟

جستجوی عمقی، عمیق ترین گره را بسط میدهد، جستجواز عمیق ترین سطح درخت جستجو ادامه می یابد، و قتیوقتی گره ها بسط داده استفاده میکند. در این صف LIFO شدند از مرز حذف میشوند و جستجو به عمیق تری ن گره بعدی برمی گردد. جستجوی عمقی از صف جدیدترین گره تولید شده، برای بسط دادن انتخاب میشود، این گره باید عمیق ترین گره بسط نداده شده باشد. جستجوی عمقی: کامل بودن : خیر ، مگر اینکه فضای حالت محدود باشد و حلقه تکرار وجود نداشته باشد . بهینه بودن : خیر ، چون کامل نیست . پیچیدگی است . BF باشد به مراتب بدتر است / در بسیاری از مسائل سریعتر از جست و جوی d خیلی بزرگتر از m ، اگر $m^b(O)$ زمانی ، در زمان عقبگرد حافظه آزاد می شود . ∞ (8 ضمن بررسی : $O(1bm+)$ پیچیدگی حافظه الگوریتم جستجوی درختی شبه کد زیر را بررسی کنید که استراتژی در کدام از 4 توابع ، پیاده سازی شده است ، توابع را نام برده و عملکرد هر یک را بیان کنید در الگوریتم جستجوی درختی ، حالت شروع در ریشه درخت قرار می گیرد، انشعابها، فعالیتها و گره ها، حالت‌های موجود هستند. ابتداء ریشه را بررسی میکنیم که آیا حالت هدف است یا خیر در صورتی که حالت هدف نبودان را بسط میدهم تا مجموعه ی جدیدی از حالت‌ها به وجود آید، بعد از آن حالت‌های یکی بررسی کرده تا زمانی که به آخرین گره برسیم که هیچ فرزندی ندارد پس سراغ گره ها میرویم و یکی یکی بررسی میکنیم پس از آن گره هایی که مارا به هدف نمیرسانند حذف میکنیم و این روش ادامه پیدا میکند تا به هدف برسیم. استراتژی های متفاوتی برای رسیدن به حالت هدف وجود دارد. استراتژی مادر اینجائین است که یک گره اولین خانه را: (first remove) (کاندید را بررسی کن اگر هدف نبود آن را بسط بده، آنقدر این کار را تکرار کن تا به هدف برسی. تابع وقتی به هدف : (expand) (آیا به هدف رسیدیم؟ خیر. یک گره باتوجه به استراتژی انتخاب کن تابع: (test goal) (میکند تابع fringe insert بسط بده و نتایج را به جستجو اضافه کن. استراتژی در تابع fringe گره های فرزند را در: (insert) (نرسیدیم گره ها را بسط بده. تابع پیاده سازی شده است. ∞ 9 شبه کد زیر مربوط به کدام جست و جوی ناآگاهانه می باشد ، از مزایای کدام جست و جو های دیگر بهره برده است ، با ترسیم شکل توضیح دهید ؟ این شبه کد مربوط به جستجوی عمقی تکرار شونده است ، که این الگوریتم از لحاظ زمانی از مرتبه جستجوی اول سطحی است و از لحاظ پیچیدگی حافظه از مرتبه جستجوی اول عمق بهره میبرد. جست و جوی عمقی تکراری ، یک استراتژی کلی است . این الگوریتم با شروع از مقدار صفر به عنوان عمق محدود ، مقدار آن را به تدریج اضافه می کند مانند یک و .. تا اینکه هدفی پیدا شود

عمق مربوط به عمیق ترین گره هدف است . این الگوریتم از مزایای d برسد ، که d هدف وقتی پیدا می شود که عمق محدود به جست و جوی عمقی و جست و جوی عمق عرضی استفاده می کند فواید مربوط به این دو الگوریتم را با هم ترکیب می کند . این الگوریتم برای تعیین عمق محدود است که جست و جو با عمق محدود را با حدود صعودی تکرار می کند و زمانی خاتمه می یابد که جوابی را برگرداند که این عمل نشان می دهد جوابی وجود ندارد . failure پیدا شود یا جست و جو با عمق محدود مقدار ∞ 10 شش نوع جست و جو های ناآگاهانه جدول زیر را به تفکیک ، با چهار معیار قابل دسترس باشد . d مربوطه به اختصار شرح دهید ؟ 1 (جست و جوی سطحی کامل بودن : بله / شرط : جواب بهینه در عمق b محدود باشد . بهینه بودن : بله / شرط : مسیر ها فاقد هزینه باشند . پیچیدگی زمانی : گره ریشه حداکثر دارای b فاکتور انشعاب d گره وجود دارد / با فرض اینکه جواب در عمق b فرزند است بنابراین در سطح دوم 2 b فرزند است / هر فرزند نیز حداکثر دارای باشد در بدترین حالت جواب باید در سمت راست ترین گره باشد پیچیدگی حافظه : هم مرتبه پیچیدگی زمانی است . 2 (جست و جو با هزینه یکنواخت کامل بودن : بله / شرط : جواب در عمق قابل دسترس باشد . هزینه ها مقدار مثبت داشته باشند . بهینه بودن : بله / است . در بدترین e هزینه مسیر بهینه است . فرض شود هزینه هر عمل حداقل c شرط : کامل باشد . پیچیدگی زمانی : فرض شود پیچیدگی حافظه : هم مرتبه پیچیدگی زمانی است . 3 (جست و جوی عمقی کامل بودن : $O(b^a)$ حالت . است زمانی پیچیدگی خیر / شرط : مگر اینکه فضای حالت محدود باشد و حلقه تکرار وجود نداشته باشد . بهینه بودن : خیر / زیرا کامل نیست . باشد به مراتب بدتر است . در بسیاری از مسائل سریعتر از جست و جوی d خیلی بزرگتر از m است، اگر $m^b(O)$: پیچیدگی زمانی با DF در زمان عقبگرد حافظه آزاد می شود . 4 (جست و جوی عمقی محدود در حقیقت $O(bm)$ است . پیچیدگی حافظه: $BF + 1$ آنگاه کامل اما غیر بهینه $L < d$ آنگاه غیر کامل است . اگر $L < d$ است . تعیین در همه مسائل امکان پذیر نمی باشد . اگر L عمق محدود جست و جوی عمق ی تکراری کامل $O(5b)$: پیچیدگی حافظه $O(b)$ آنگاه کامل و بهینه است . پیچیدگی زمانی : $L = 1$ است . اگر زمانی : $O(b^d)$. بودن : بله / شرط : حلقه تکرار وجود نداشته باشد . بهینه بودن : بله / اگر هزینه مسیر ها با هم برابر باشد جست و جوی دو طرفه کامل بودن : بله / شرط : استفاده کردن از جست و جوی سطری بهینه $O(6b)$: پیچیدگی پیچیدگی حافظه حافظه یی : $O(b^d/2)$ زمانی پیچیدگی: $O(b^d/2)$ بودن : بله / شرط : استفاده کردن از جست و جوی سطری

search با جست و جوی حریصانه SLD h را با توجه به جدول A* * 11 جست و جوی Greedy یعنی $n(g)$ با رسم درختی به طور کامل توضیح داده و تفاوت ها را با دلیل ذکر کنید ؟ در این روش گره هارا با ترکیب $n(f)$ هزینه $n(h)$ یعنی $n(h)+n(g)=n(f)$. یعنی هزینه رسیدن از این گره به گره هدف ارزیابی می کند $n(h)$ هزینه رسیدن به گره و راداشته باشد. شناخته شده ترین $n(g)$ و $n(h)$ است. پس باید به گره ای فکر کنیم که کمترین n برآورد شده ی ارزانتترین جوار از طریق $n(g) = n(h) + f(n)$: جستجوی آگاهانه ایده: از بسط گرهای که به صرفه به نظر نمیرسند، اجتناب میکند هزینه تخمینی از گره شروع تا هدف با عبور از $n(f)$ تا هدف n هزینه تخمینی از گره $n(h)$ هزینه واقعی از گره شروع تا گره کامل و بهینه و بهینه موثر است. مرتبه زمانی و مکانی آن نمایی است A* جستجوی nگره

گره ای را بسط م دهد که به هدف نزدیکتر باشد . این جستجو کامل نیست چون حلقه تکرار دارد : $n(h) = n(f)$: جستجوی حریصانه . یعنی هزینه واقعی است $n(g)$ در A* است . تفاوت الگوریتم حریصانه $(m^b o)$ و بهینه هم نیست و مرتبه زمانی و مکانی آن مینیمم ترین گره انتخاب شده و به آن مینیمم هزینه A جستجو را بهینه و کامل میکند. جستجوی حریصانه زود تصمیم می گیرد ، اما در A واقعی اعتماد میکند. 12 الگوریتم زیر را شرح دهید و با توجه به جدول و شکل سوال است که در آن : 1) بهترین گره برگ و بهترین جانشین برای آن RBFS 11 با رسم درخت جست و جو توضیح دهید ؟ این الگوریتم انتخاب شود . 2) اگر مقدار بهترین گره برگ از جانشین آن بیشتر شد، آنگاه به مسیر جانشین عقبگرد شود . 3) در حین عقبگرد، است . از تولید تعداد A ID جستجوی به مراتب موثرتری از RBFS . روزرسانی شود . 4) گره جانشین بسط داده شود $n(f)$ مقدار $(bd o)$ قابل پذیرش باشد، بهینه است . پیچیدگی حافظه $n(h)$ اگر A* بسیار زیادی گره به دلیل تغییر عقیده رنج می برد . مانند است . پیچیدگی زمانی به کیفیت تابع هیوریستیک و میزان تغییر عقیده بستگی دارد . 13 چند نوع تابع هیوریستیک را می توان برای پازل اعداد معرفی کرد ، با رسم شکل بررسی کنید ؟ تابع هیوریستیک قابل پذیرش 1 از هر کاشی می تواند به هر خانه ... h هر کاشی می تواند به هر جایی منتقل شود 2.1h version relax طریق نسخه ساده شده از مساله هزینه راه حل برای مکعب روییک را تخمین میزند . ابداع تابع هیوریستیک قابل پذیرش (3) از ABSolver.. همسایه منتقل شود تجربه : حل تعداد بسیار زیادی از مساله (experience from learning) طریق یادگیری از تجربه 14 سه راه حل جهت ابداع تابع هیوریستیک نام برده و شرح دهید ؟ 1) از طریق نسخه . هر کاشی می تواند به هر خانه همسایه منتقل شود H هر کاشی می تواند به هر جایی منتقل شود . H2 ساده شده از مساله 1 هزینه راه حل برای مکعب روییک را تخمین میزند . 2) از طریق نسخه کوچکتر از مساله 3) از طریق یادگیری از ABSolver تجربه تجربه : حل تعداد بسیار زیادی از مساله 15 انواع جست و جوی محلی را نام ، پرتو محلی ، ژنتیک الگوریتم جست و جوی محلی تپه نوردی : SA برده و ایده هر یک را بیان کنید ؟ جست و جوی تپه نوردی ، این الگوریتم حلقه ای است که در جهت افزایش مقدار حرکت می کند (به طرف بالای تپه) . وقتی به قله ای رسید که هیچ همسایه ای این الگوریتم نسخه ای از تپه نوردی اتفاقی است و پایین آمدن : SA از آن بلند تر نیست خاتمه می یابد . الگوریتم جست و جوی محلی پذیرفته شده و با گذشت طمان کمتر اتفاق می annealing از تپه مجاز است . حرکت به طرف پایین و به آسانی در اوایل زمانبندی افتد . الگوریتم جست و جوی پرتو محلی : نگهداری فقط یک گره در حافظه ، واکنش افراطی نسبت به مسئله محدودیت حافظه است حالت که به طور تصادفی تولید شدند ، شروع k حالت را نگهداری می کند . این الگوریتم با k. این الگوریتم به جای یک حالت ، می کند . در هر مرحله تمام پسین های همه حالت ها تولید می شوند . اگر یکی از آن ها هدف بود ، الگوریتم متوقف می شود ؛ وگرنه بهترین پسین را انتخاب و عمل را تکرار می کند . الگوریتم جست و جوی محلی ژنتیک : این الگوریتم شکلی از جست و جوی پرتو اتفاقی است که در آن ، حالت های پسین از طریق ترکیب دو حالت والد تولید می شوند . در مقایسه با انتخاب طبیعی ، مثل جست و جوی پرتو اتفاقی است ، با این تفاوت که اینجا با تولید مثل جنسی سروکار داریم نه غیر جنسی . این الگوریتم همانند حالت که به طور تصادفی تولید شدند شروع می کند که به آن جمعیت گفته می شود k جست و جوی پرتو محلی ، با مجموعه ای از 16 الگوریتم زیر را شرح داده و انواع آن را نام برده و بررسی کنید ؟ الگوریتم بال مربوط به الگوریتم جست و جوی محلی تپه نوردی می باشد . این الگوریتم حلقه ای است که در جهت افزایش مقدار حرکت می کند (به طرف بالای تپه) . وقتی به قله ای رسید که هیچ همسایه ای از آن بلند تر نیست خاتمه می یابد . در این الگوریتم درخت جست و جو را نگهداری نمی کند . لذا ساختمان داده گره فعلی فقط باید حالت و مقدار تابع هدف را نگهداری کند . تپه نوردی به همسایه های حالت فعلی نگاه می کند . مثل تالش برای یافتن قله کوه اورست در مه گرفتگی غلیظ ، در حالی که دچار فراموشی هستید . تپه نوردی گاهی جست و جوی محلی حریصانه نام دارد زیرا بدون اینکه قبال فکر کند به کجا برود ، حالت همسایه خوبی را انتخاب می کند . . تپه نوردی معمول به سرعت به جواب پیش می رود ، زیرا به راحتی می تواند حالت بد را بهبود ببخشد