



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

پروژه دوم هوش مصنوعی
رشته علوم کامپیوتر

الگوریتم A^*

نگارش
علیرضا مختاری

استاد درس
مهدی قطعی

استاد کارگاه
بهنام یوسفی مهر

شهریور 1403

چکیده

در این تمرین، به بررسی و پیاده‌سازی الگوریتم‌های جستجو برای یافتن مسیر بهینه در یک محیط پیچیده مانند هزارتوی بازی تام و جری پرداخته می‌شود. ابتدا مقاله‌ای درباره یک نسخه بهبودیافته از الگوریتم $A-star$ برای مسیریابی وسایل نقلیه خودران مطالعه می‌شود و مفاهیم اصلی و نوآوری‌های آن درک و تحلیل می‌گردد. سپس، چندین الگوریتم جستجوی مختلف شامل جستجوی سطح اول (BFS)، جستجوی هزینه یکنواخت (UCS)، الگوریتم حریمانه، و $A-star$ در محیط تمرین پیاده‌سازی شده و نتایج آن‌ها از نظر زمان اجرا، تعداد گره‌های بررسی‌شده و عمق جستجو مورد مقایسه قرار می‌گیرد.

واژه‌های کلیدی:

جستجوی سطح اول (BFS)، جستجوی هزینه یکنواخت (UCS)، الگوریتم حریمانه، $A-star$ ، مسیریابی، الگوریتم‌های جستجو، وسایل نقلیه خودران

فهرست مطالب

چکیده.....	أ
فصل اول مقدمه.....	3
فصل دوم الگوریتم بهبودیافته ی Star-A برای مسیریابی وسایل نقلیه خودران زمینی.....	5
2-1- مفاهیم اصلی و نوآوری مقاله.....	6
2-2- الگوریتم های بهبود الگوریتم سنتی.....	6
2-3-1- الگوریتم A-Star مبتنی بر راهنما.....	6
2-3-2- الگوریتم A-Star مبتنی بر نقاط کلیدی.....	7
2-3-3- الگوریتم A-Star مبتنی بر گام های متغیر.....	7
2-3-4- نتیجه گیری و مقایسه عملکرد.....	8
2-3- مقایسه با الگوریتم سنتی A-start.....	8
فصل سوم الگوریتم های جستجو.....	11
3-1- الگوریتم های جستجو.....	12
3-2- اثبات Consistent و Admissible.....	13
فصل چهارم جمع بندی و نتیجه گیری و پیشنهادات.....	16
منابع و مراجع.....	18

فصل اول

مقدمه

مقدمه

الگوریتم‌های جستجو از مهم‌ترین ابزارها برای مسیریابی در محیط‌های پیچیده به شمار می‌روند. این الگوریتم‌ها در بسیاری از حوزه‌ها از جمله رباتیک، تدارکات، و بازی‌های ویدیویی مورد استفاده قرار می‌گیرند. هدف اصلی آن‌ها پیدا کردن کوتاه‌ترین یا بهینه‌ترین مسیر بین دو نقطه است که موانع و محدودیت‌های محیط را در نظر بگیرند. در این تمرین، به پیاده‌سازی و تحلیل چندین الگوریتم جستجو از جمله *A-star*، که یکی از الگوریتم‌های قدرتمند و رایج برای مسیریابی بهینه است، پرداخته می‌شود. علاوه بر آن، نسخه‌های بهبودیافته‌ای از این الگوریتم‌ها نیز بررسی خواهند شد تا کارایی و بهینه‌بودن راه‌حل‌ها از جنبه‌های مختلف مورد ارزیابی قرار گیرد.

فصل دوم

الگوریتم بهبودیافته ی Star-A برای مسیریابی وسایل نقلیه خودران زمینی

2-1- مفاهیم اصلی و نوآوری مقاله

- ارائه یک استاندارد ارزیابی جامع: یکی از نوآوری‌های اصلی مقاله معرفی یک معیار ارزیابی جامع برای الگوریتم‌های برنامه‌ریزی مسیر است که به مقایسه عملکرد الگوریتم‌های مختلف و انتخاب بهترین پارامترها کمک می‌کند. این معیار به عنوان ابزاری برای ارزیابی دقیق و عینی عملکرد الگوریتم‌های مختلف به کار می‌رود.
- الگوریتم A-Star مبتنی بر راهنما: این نوآوری از یک راهنما (guideline) استفاده می‌کند که توسط انسان یا برنامه‌ریزی جهانی تولید شده است و به عنوان تابع هیورستیک در طول فرآیند جستجو به کار می‌رود. این راهنما الگوریتم را در انتخاب مسیری مشابه با مسیر ایده‌آل یاری می‌کند و ضعف‌های الگوریتم A-Star کلاسیک را که در پیچ‌ها به مشکل برمی‌خورد، برطرف می‌سازد.
- الگوریتم A-Star مبتنی بر نقاط کلیدی: برای بهبود اجتناب از موانع، از نقاط کلیدی در اطراف موانع استفاده می‌شود. این نوآوری به الگوریتم کمک می‌کند تا قبل از نزدیک شدن به مانع، از آن دوری کند. این رویکرد کارآمدتر از روش‌های سنتی است که معمولاً موانع را دیرتر تشخیص می‌دهند.
- الگوریتم A-Star مبتنی بر گام‌های متغیر: این نوآوری با تغییر اندازه گام‌های جستجو بر اساس محیط، زمان محاسباتی الگوریتم را کاهش می‌دهد. در مناطق باز، گام‌های بزرگ‌تر و در نزدیکی موانع، گام‌های کوچک‌تر انتخاب می‌شوند. این ویژگی باعث افزایش سرعت و کارایی الگوریتم می‌شود.

2-2- الگوریتم‌های بهبود الگوریتم سنتی

در مقاله ارائه شده از سه تا الگوریتم استفاده شده است که باعث بهبود الگوریتم A-star می‌شود در ادامه به توضیح هر کدام از الگوریتم‌ها می‌پردازیم:

2-3-1- الگوریتم A-Star مبتنی بر راهنما

- هدف: برطرف کردن ضعف‌های الگوریتم A-Star سنتی در مسائلی که مسیر دارای پیچ و خم‌های زیادی است یا زمانی که وسیله نقلیه نیاز به پیروی از یک مسیر تعیین‌شده توسط انسان یا برنامه‌ریزی جهانی دارد.

- **توضیح:** در این الگوریتم، یک راهنما یا مسیریابی اولیه که توسط انسان یا برنامه‌ریزی جهانی تولید شده، به عنوان یک مرجع در فرآیند جستجو به کار می‌رود. راهنما در طول فرآیند جستجو مورد استفاده قرار می‌گیرد تا تابع هیورستیک را توسعه دهد و هدف و قصد راننده یا برنامه‌ریزی جهانی را در حرکت وسیله نقلیه منعکس کند.
- **ویژگی کلیدی:** تابع هیورستیک به شکل $F(i) = G(i) + H1(i) \cdot a1 + H2(i) \cdot a2$ تعریف می‌شود که در آن $H1(i)$ فاصله نقطه i از راهنما و $H2(i)$ فاصله از نقطه $g(i)$ در راهنما تا هدف است.
- **مزیت‌ها:** این رویکرد به الگوریتم اجازه می‌دهد تا به مسیر راهنما نزدیک‌تر باشد و از مسیر بهتری در سناریوهایی با پیچ‌های متعدد پیروی کند، برخلاف A-Star سنتی که در این شرایط ممکن است مسیرهای نامناسبی را انتخاب کند.

2-3-2- الگوریتم A-Star مبتنی بر نقاط کلیدی

- **هدف:** بهبود عملکرد اجتناب از موانع در الگوریتم A-Star مبتنی بر راهنما.
- **توضیح:** در مواقعی که راهنما از یک مانع عبور می‌کند، این الگوریتم نقاط کلیدی (Key Points) را در اطراف مانع شناسایی می‌کند. سپس این نقاط به عنوان راهنماهای جدید مورد استفاده قرار می‌گیرند تا وسیله نقلیه بتواند به طور مؤثرتری از موانع اجتناب کند.
- **فرآیند:**
 - اگر در مسیر راهنما مانعی وجود داشته باشد، نقاط کلیدی در اطراف مانع شناسایی می‌شوند.
 - خطوط جایگزین با استفاده از این نقاط کلیدی ایجاد می‌شوند و الگوریتم مسیرهای جایگزین را بررسی می‌کند.
 - مسیر جایگزینی که فاقد مانع است و زمان کم‌تری نیاز دارد، به عنوان مسیر جدید انتخاب می‌شود.
- **مزیت‌ها:** این الگوریتم به الگوریتم اجازه می‌دهد تا سریع‌تر از موانع عبور کند، که در نتیجه از مسیرهای خطرناک جلوگیری می‌کند و مسیری ایمن‌تر و کارآمدتر ارائه می‌دهد.

2-3-3- الگوریتم A-Star مبتنی بر گام‌های متغیر

- **هدف:** کاهش زمان محاسباتی و بهبود عملکرد در محیط‌های باز و نزدیک به موانع.
- **توضیح:** در این الگوریتم، اندازه گام‌های جستجو بر اساس پراکندگی موانع در محیط به صورت پویا تغییر می‌کند. در مناطق باز که موانع کمی وجود دارند، از گام‌های بزرگ‌تر برای افزایش

- سرعت جستجو استفاده می‌شود. در مقابل، در نزدیکی موانع، گام‌های کوچک‌تر به کار گرفته می‌شوند تا دقت بیشتری در اجتناب از موانع داشته باشند.
- **ویژگی کلیدی:** این الگوریتم با ترکیب اندازه گام‌های متغیر، فضای جستجو را با سرعت و دقت بهتری پوشش می‌دهد و باعث کاهش تعداد نقاط توسعه یافته و زمان محاسباتی می‌شود.
- **مزیت‌ها:** این رویکرد بهبود قابل توجهی در کارایی محاسباتی ایجاد می‌کند، به ویژه در محیط‌هایی که به سرعت و واکنش‌پذیری بالا نیاز دارند.

4-3-2- نتیجه‌گیری و مقایسه عملکرد

- الگوریتم‌های بهبود یافته فوق‌الذکر به طور مؤثری مشکلات الگوریتم A-Star سنتی را حل می‌کنند:
 - الگوریتم مبتنی بر راهنما به بهبود هدایت وسیله نقلیه در مسیرهای پیچیده کمک می‌کند.
 - الگوریتم مبتنی بر نقاط کلیدی، دقت اجتناب از موانع را افزایش می‌دهد.
 - الگوریتم مبتنی بر گام‌های متغیر، کارایی زمانی را بهبود می‌بخشد.
- این بهبودها باعث شده‌اند که الگوریتم نهایی A-Star برای کاربردهای وسایل نقلیه زمینی خودکار بسیار کارآمدتر و عملی‌تر شود

2-3- مقایسه با الگوریتم سنتی A-start

1. تطبیق با محیط‌های پیچیده و اجتناب از موانع:

الگوریتم A-Star سنتی: در مواجهه با موانع، الگوریتم سنتی تنها پس از رسیدن به مانع به تغییر مسیر اقدام می‌کند. این امر منجر به انتخاب مسیرهای غیر بهینه می‌شود و باعث افزایش زمان محاسبه و طول مسیر می‌گردد. همچنین در صحنه‌هایی با پیچ‌های تند، الگوریتم سنتی اغلب نمی‌تواند مسیر مناسبی را برنامه‌ریزی کند.

الگوریتم‌های بهبود یافته: با استفاده از الگوریتم A-Star مبتنی بر نقاط کلیدی، مسیرهای بهتری انتخاب می‌شوند، زیرا این الگوریتم می‌تواند زودتر موانع را تشخیص داده و مسیر مناسب‌تری را از آن‌ها

عبور دهد. این امر منجر به بهبود عملکرد در اجتناب از موانع می‌شود و باعث می‌شود مسیرهای تولیدشده توسط الگوریتم بهینه‌تر و کوتاه‌تر باشند.

2. استفاده از راهنمایی‌ها برای تعیین مسیر:

الگوریتم A-Star سنتی: تنها به تابع هیورستیک $F = G + H$ وابسته است که در آن G هزینه مسیر طی شده و H فاصله تا هدف است. این تابع برای تمامی سناریوها ثابت است و نمی‌تواند به‌طور مؤثری اطلاعاتی مانند پیچ‌ها یا جاده‌های غیرمستقیم را در نظر بگیرد.

الگوریتم مبتنی بر راهنما: این الگوریتم از یک راهنمای تعیین شده توسط انسان یا برنامه‌ریزی جهانی استفاده می‌کند که به عنوان تابع هیورستیک به کار می‌رود. این راهنما مسیر مناسب را به الگوریتم نشان می‌دهد و کمک می‌کند تا تصمیمات هوشمندانه‌تری گرفته شود. این ویژگی باعث می‌شود که الگوریتم در شرایط پیچیده مانند پیچ‌های تند یا جاده‌های پرپیچ‌وخم عملکرد بهتری داشته باشد.

3. بهبود عملکرد محاسباتی با استفاده از گام‌های متغیر:

الگوریتم A-Star سنتی: با استفاده از گام‌های ثابت عمل می‌کند، به این معنا که در تمام شرایط، الگوریتم همان اندازه گام را برای حرکت انتخاب می‌کند. این امر منجر به افزایش تعداد محاسبات، به‌ویژه در فضاهای باز، می‌شود.

الگوریتم مبتنی بر گام‌های متغیر: با تنظیم اندازه گام‌ها بر اساس توزیع موانع، این الگوریتم می‌تواند گام‌های بزرگ‌تر در مناطق باز و گام‌های کوچک‌تر در نزدیکی موانع انتخاب کند. این بهینه‌سازی باعث کاهش زمان محاسبات و بهبود سرعت اجرای الگوریتم می‌شود.

تحلیل نتایج آزمایش‌ها:

در آزمایش‌هایی که برای مقایسه عملکرد الگوریتم‌ها انجام شده، الگوریتم‌های بهبود یافته نسبت به الگوریتم A-Star سنتی نتایج بهتری نشان داده‌اند. به‌ویژه در مسیرهای پیچیده، الگوریتم مبتنی بر راهنما توانست مسیرهای بهینه‌تر و کوتاه‌تری ارائه دهد.

تعداد نقاط گسترش یافته توسط الگوریتم بهبود یافته به میزان قابل توجهی کمتر از الگوریتم سنتی بود، که نشان‌دهنده کاهش زمان محاسباتی است.

همچنین، الگوریتم مبتنی بر نقاط کلیدی توانست در برخورد با موانع عملکرد بهتری از خود نشان دهد و به صورت پیش‌دستانه مسیرهایی را انتخاب کرد که از برخورد با موانع جلوگیری کند.

الگوریتم‌های بهبود یافته به طور کلی نسبت به الگوریتم A-Star سنتی در جنبه‌های مختلفی مانند اجتناب از موانع، کارایی محاسباتی، و پایداری در شرایط پیچیده عملکرد بهتری داشته‌اند. این بهبودها باعث می‌شوند که الگوریتم A-Star برای کاربردهای واقعی در وسایل نقلیه خودکار، به ویژه در محیط‌های شهری یا مسیرهای پیچیده، کارآمدتر و قابل اعتمادتر باشد.

فصل سوم

الگوریتم های جستجو

1-3- الگوریتم های جستجو

-	Computing time	Expanded nodes	Depth
BFS	0.37	1595	63
DFS	0.06	255	85
UCS	0.37	1594	63
Greedy	0.06	212	76
A*	1.89	6790	62

- **زمان اجرا:** این معیار نشان دهنده سرعت الگوریتم در پیدا کردن راه حل است. برخی از الگوریتم ها مانند BFS و UCS تضمین می کنند که کوتاه ترین مسیر را پیدا کنند، اما ممکن است زمان اجرای بیشتری نسبت به الگوریتم های حریصانه و A* داشته باشند، که در شرایط خاص سریع تر هستند.
- **تعداد گره های بررسی شده:** این معیار نشان دهنده تعداد گره هایی است که در طول جستجو پردازش می شوند. الگوریتم هایی که به صورت کامل فضای حالت را بررسی می کنند، مانند BFS، معمولاً تعداد بیشتری گره را پردازش می کنند. از سوی دیگر، الگوریتم های حریصانه و A* به دلیل استفاده از هیوریستیک، تعداد کمتری از گره ها را پردازش می کنند.
- **عمق جستجو:** این معیار نشان می دهد که الگوریتم تا چه عمقی در درخت جستجو پیش رفته است تا به راه حل برسد. عمق می تواند بسته به ویژگی های مسئله و نوع الگوریتم متفاوت باشد. الگوریتم های UCS و A* ممکن است به عمق های بیشتری نسبت به BFS و جستجوی حریصانه برسند، اما در عین حال از مسیرهای بهینه پیروی می کنند.

2-3- اثبات Admissible و Consistent

Admissibility (پذیرفتنی بودن)

تعریف: یک تابع اکتشافی $h(n)$ Admissible است اگر برای هر گره n ، مقدار $h(n)$ از هزینه واقعی مسیر بهینه از n به هدف، یعنی $h^*(n)$ کمتر یا مساوی باشد:

$$h(n) \leq h^*(n) \quad \forall n$$

هدف اثبات: نشان دهیم که اگر تابع اکتشافی $h(n)$ پذیرفتنی باشد، الگوریتم A^* همیشه مسیر بهینه را پیدا می کند.

اثبات:

فرض کنیم $h(n)$ تابع اکتشافی پذیرفتنی باشد، یعنی برای هر گره n ، $h(n) \leq h^*(n)$

الگوریتم A^* گره ها را بر اساس مقدار تابع ارزیابی $f(n)=g(n)+h(n)$ انتخاب می کند، جایی که $g(n)$ هزینه مسیر از گره شروع تا گره n است.

فرض کنید A^* مسیر بهینه را پیدا نمی کند و یک مسیر suboptimal (غیر بهینه) را انتخاب می کند. این به این معناست که یک گره n با هزینه $f(n)$ بزرگ تر از مسیر بهینه آن توسط الگوریتم انتخاب شده است.

اما چون $h(n) \leq h^*(n)$ ، مقدار $f(n)=g(n)+h(n)$ برای مسیر بهینه همیشه کمتر یا مساوی از $f(n)$ برای مسیر suboptimal خواهد بود. بنابراین، A^* ابتدا مسیر بهینه را بررسی می کند و آن را انتخاب می کند، نه مسیر suboptimal.

نتیجه گیری: اگر $h(n)$ پذیرفتنی باشد، الگوریتم A^* همیشه مسیر بهینه را پیدا می کند.

Consistency (سازگار بودن)

یک تابع اکتشافی $h(n)$ Consistent است اگر برای هر گره n و هر گره فرزند n' از n ، نابرابری زیر برقرار باشد:

که در آن $c(n, n')$ هزینه رسیدن از گره n به گره فرزند n' است.

هدف اثبات: نشان دهیم که اگر تابع اکتشافی $h(n)$ سازگار باشد، آنگاه A^* هیچ گاه یک گره را بیش از یک بار گسترش نمی دهد و مسیر بهینه را پیدا می کند.

اثبات:

ثابت سازی سازگاری

با فرض سازگار بودن $h(n)$ ، برای هر گره n و هر فرزند n' ، داریم:

$$h(n) \leq c(n, n') + h(n')$$

این نابرابری به معنی آن است که تخمین هزینه از n به هدف از طریق هر فرزند n' ، هیچ گاه از تخمین مستقیم $h(n)$ تجاوز نمی کند.

نتیجه گیری از سازگاری

با استفاده از سازگاری، می توان نشان داد که مسیر بهینه ابتدا بازدید می شود و هیچ گاه نیاز به بازبینی مجدد ندارد. زیرا اگر $h(n)$ سازگار باشد، پس برای هر مسیر بهینه $g(n) + h(n) = f(n)$ کوچک ترین مقدار ممکن را دارد.

عدم بازدید مجدد گره ها

اگر گره‌ای با مسیر کم‌هزینه‌تر یافت شود، نابرابری سازگاری تضمین می‌کند که مسیر جدید هزینه‌ای کمتر یا مساوی مسیر قبلی خواهد داشت، که این امر باعث می‌شود الگوریتم A^* نیازی به بازدید مجدد گره نباشد.

نتیجه‌گیری: اگر $h(n)$ سازگار باشد، الگوریتم A^* هر گره را فقط یک بار گسترش می‌دهد و مسیر بهینه را تضمین می‌کند.

فصل چهارم

جمع بندی و نتیجه گیری و پیشنهادات

جمع بندی و نتیجه گیری

در این مقاله بهبودهایی بر روی الگوریتم سنتی A* برای مسیریابی وسایل نقلیه خودران زمینی معرفی شد. با توجه به نیاز به یافتن مسیرهای بهینه با در نظر گرفتن محدودیت‌ها و موانع پیچیده، الگوریتم‌های پیشنهادی به صورت مؤثری توانستند عملکرد بهتری از خود نشان دهند. به ویژه، با معرفی سه الگوریتم بهبود یافته، امکان یافتن مسیرهای بهینه‌تر و کاهش زمان جستجو فراهم شد.

مقایسه بین الگوریتم‌های بهبود یافته و الگوریتم A* سنتی نشان می‌دهد که این الگوریتم‌ها نه تنها از نظر سرعت اجرا بهبود یافته‌اند، بلکه توانسته‌اند تعداد گره‌های کاوش شده را کاهش دهند و در عین حال عمق جستجو را حفظ کنند. این بهبودها، کاربردهای عملی زیادی در سیستم‌های مسیریابی وسایل نقلیه خودران و دیگر محیط‌های پیچیده دارد.

در نهایت، نتایج ارائه شده نشان می‌دهد که نوآوری‌های مطرح شده در این مقاله می‌توانند راهکارهای عملی بهتری برای مسائل مسیریابی فراهم کنند و به عنوان پایه‌ای برای تحقیقات آینده در این حوزه به کار روند.

منابع و مراجع

- [1] Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach," 4th Edition, Pearson, 2020.
- [2] <https://www.geeksforgeeks.org/>
- [3] <https://stackoverflow.com/>
- [4] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). "A formal basis for the heuristic determination of minimum cost paths".