



دانشگاه صنعتی امیرکبیر  
(پلی‌تکنیک تهران)  
دانشکده ریاضی و علوم کامپیوتر

گزارش دوم درس هوش مصنوعی

الگوریتم های مسیریابی و پیمایش گراف

نگارش  
آرمان صالحی

استاد راهنما  
دکتر مهدی قطعی

اسفند 1402

## چکیده

این گزارش خلاصه ای از یک روش برای بهبود الگوریتم مسیریابی A-star است که برای وسایل نقلیه خود راهنما یک روش هندسی از الگوریتم A-star ارائه میدهد که باعث افزایش سرعت و کیفیت الگوریتم سنتی میشود.

## واژه‌های کلیدی:

مسیریابی، وسایل نقلیه خودراهنما، الگوریتم A-star

## صفحه

## فهرست مطالب

چکیده	.....	أ
فصل اول مقدمه	.....	١
فصل دوم مقدمه ای بر مقاله Geometric A-Star Algorithm	.....	٣
2-1- مقدمه	.....	٤
2-2- اصل و کل فرآیند الگوریتم هندسی A-star	.....	٦
2-3- اهداف	.....	8
2-4- چالش ها	.....	8
فصل سوم الگوریتم A-star	.....	٩
3-1- الگوریتم A-star چیست؟	.....	١٠
3-2- نحوه عملکرد الگوریتم A-star	.....	١١
3-3- تابع اکتشافی	.....	١٢
فصل چهارم تحلیل نتایج پیاده سازی الگوریتم ها	.....	13
فصل پنجم نتیجه گیری	.....	13
منابع و مراجع	.....	15
Abstract	.....	16

## فصل اول

### مقدمه

## مقدمه

این گزارش درمورد مقاله Geometric A-Star Algorithm است که به بهبود الگوریتم A\* میپردازد.

الگوریتم A\* یک الگوریتم جستجو در درخت و گراف است که بهترین مسیر بین دو گره را پیدا میکند اما این الگوریتم در محیط های پیچیده مشکلاتی دارد که باعث طولانی تر شدن جستجو می شود، این مقاله اما روشی برای جلوگیری از این طولانی شدن جستجو ارائه می دهد که در این روش مسیر هایی که مناسب نیستند یا دندانه ای هستند را از جستجو حذف میکند.

همچنین الگوریتم A\* را نیز توضیح می دهد که دید بهتری برای درک روش گفته شده به مخاطب می دهد.

## فصل دوم

## مقدمه ای بر مقاله Geometric A-Star Algorithm

## 1-2- مقدمه ای بر مقاله Geometric A-Star Algorithm

امروزه با توسعه سریع اتوماسیون تدارکات انبار و افزایش هزینه نیروی کار، حمل و نقل کالا در بسیاری از حوزه‌ها به سمت سیستماتیک سازی و اتوماسیون در حال توسعه است. وسایل نقلیه خودراهنما<sup>1</sup> (AGV) یکی از ابزار های سیستماتیک سازی است که به طور گسترده در انبارها، کارخانه‌ها، بنادر استفاده می‌شود. یکی از چالش های این وسایل نقلیه خودراهنما پیدا کردن مسیر مناسب و سریع است.

این مقاله یک روش مسیریابی، که بر مبنای الگوریتم A-star است را معرفی میکند. رویکرد این روش برای جلوگیری از مشکلات مسیریابی، برای وسایل نقلیه خودراهنما است که توسط الگوریتم A-star سنتی بوجود می‌آید. الگوریتم A-star که یک الگوریتم اکتشافی<sup>2</sup> است که علاوه بر سرعت خوب در رسیدن به پاسخ نهایی، کارایی خوبی نیز دارد. با این حال وقتی از این الگوریتم در محیط های پیچیده استفاده میشود، گاهی مسیر های متقاطع و دندان اره ای ساخته میشود. تا بحال برای حل این مشکلات دو دیدگاه ارائه شده است که اولی بهینه سازی در طول الگوریتم سرچ است که هزینه زیادی دارد همچنین مسیر چرخش را هموار نمی‌کند و دومی بعد از پایان الگوریتم سرچ است که به تعداد نود ها بستگی دارد و همچنین به تکرار نیز نیاز دارد.

این مقاله مزایای الگوریتم A-Star و الگوریتم درونیابی را ترکیب می کند، آن را از طریق قوانین هندسی بهینه می کند و یک الگوریتم هندسی A-star را ارائه می‌کند. الگوریتم های درونیابی عموماً منحنی های صاف را برای تناسب با کل مسیر اعمال می‌کنند.

در مرحله اول این روش، یک grid map از محیط بندر ساخته می‌شود که الگوریتم A-star بتواند به سرعت یک مسیر بدون برخورد را جستجو کند،

سپس در مرحله دوم این مقاله دو تابع  $P(x, y)$  و  $W(x, y)$  را در نظر گرفته است برای از بین بردن گره هایی که باعث تولید مسیرهای های نامنظم می‌شوند. در مرحله سوم، این روش از طریق درونیابی B-spline مکعبی به همواری کل مسیر دست می یابد. به طور کلی این مقاله به وسایل نقلیه خودراهنما در بنادر می‌پردازد.

<sup>1</sup> Automated Guided Vehicle (AGV)

<sup>2</sup> Heuristic Search Algorithm

با توسعه فناوری جابجایی خودکار، محیط های کاری و وظایف AGV ها نیز در حال توسعه و تغییر هستند، این به این معنی است که AGV ها باید یک مسیر ایمن، بدون برخورد با موانع و عملی را پیدا کنند، در گام اول AGV باید یک درک از موانع ثابت در بندر بدست آورد سپس از یک الگوریتم جستجوی مسیر مناسب از نقطه شروع تا نقطه هدف استفاده می شود که الگوریتم A-star بطور جدی روی کارایی AGV تاثیر میگذارد.

تمرکز اصلی ایجاد یک نقشه از موانع بر اساس درک محیط و سپس پیدا کردن یک مسیر با کمترین هزینه با توجه به موانع است.



## 2-2- اصل و کل فرآیند الگوریتم هندسی A-star

الگوریتم هندسی A-star جستجوی مسیر را از طریق توابع  $P(x, y)$  و  $W(x, y)$  که وظیفه حذف گره های غیرقابل قبول را دارد، گره های اضافه را از لیست بسته حذف میکند که باعث هموار تر شدن مسیر نسبت به الگوریتم سنتی A-star می شود.

الگوریتم هندسی A-star طبق مراحل زیر عمل می کند:

1. گره مجاور گره شروع را در لیست باز قرار میدهد و اگر گره ای مانع باشد آنرا از لیست باز حذف میکند.
2. تابع  $f(x)$  را برای گره های داخل لیست باز محاسبه میکنیم و سپس کم هزینه ترین گره را انتخاب میکنیم و گره شروع را در لیست بسته ها قرار میدهیم. تابع  $f(x) = g(x) + h(x)$  است که در آن  $g(x)$  هزینه رسیدن به گره بعدی و  $h(x)$  تخمین فاصله گره بعدی تا گره پایانی است.
3. نود انتخابی را  $n$  در نظر میگیریم، گره های مجاور آنرا بررسی می کنیم اگر در لیست باز ها نبود آنرا به عنوان فرزند  $n$  به لیست اضافه می کنیم، اگر در لیست باز ها بود تابع  $g(n)$  آنرا از مسیر گره  $n$  بررسی میکنم اگر از مسیر قبلی کمتر بود آنرا به عنوان فرزند  $n$  اضافه می کنیم و قبلی را حذف می کنیم و دوباره تابع  $f(n)$  را برای گره هایی که در لیست باز هستند محاسبه میکنیم.
4. مرحله سوم را تا اضافه شدن نقطه هدف به لیست بسته ها ادامه می دهیم، سپس از نقطه هدف شروع میکنیم با دنبال کردن نقاط پدر به نقطه شروع رسیدن تا یک مسیر از نقطه شروع تا پایان پیدا کنیم.
5. بعد از اضافه شدن گره هدف کل گره های داخل لیست بسته توسط توابع  $P(x, y)$  و  $W(x, y)$  چک میشوند تا گره هایی که شرایط را برآورده نمی کنند را حذف کنند.
6. چک میکنیم که آیا تعداد گره ها در لیست بسته در مرحله ۵ کاهش می یابد یا خیر اگر کاهش یافته بودن به مرحله ۵ ادامه میدهیم در غیر این صورت از حلقه خارج شده و گره های باقی مانده را در لیست نهایی ذخیره می کنیم.
7. گره های باقی مانده را بهم وصل می کنیم تا یک مسیر جدید بدست بیاید و همزمان مسیر چرخش را هموار می کنیم.

### 3-2- اهداف روش Geometric A-Star Algorithm

در الگوریتم سنتی A-star در محیط های پیچیده و یا بزرگ این الگوریتم مسیر های دندانه ای و نامنظم تولید میکند، روش هندسی الگوریتم A\* با ترکیب مزایای الگوریتم A-Star و الگوریتم درونیابی، آن را از طریق قوانین هندسی بهینه می کند. که باعث میشود AGV ها مسیر های بهینه و راحتتری را بدست بیاورند.

### 4-2- چالش های Geometric A-Star Algorithm

از جمله چالش هایی که این روش با آن روبرو است مدل سازی محیط هایی است که این الگوریتم باید روی آنها اجرا شود، این الگوریتم یک الگوریتم پیدا کردن مسیر با استفاده از گراف است در نتیجه برای اجرای آن باید بتوانیم از یک تصویر یا ... یک گراف از راه هایی که AGV میتواند از آنها عبور کند شبیه سازی کنیم برای مثال در یک بندر اجسام و موانع زیادی مانند کانتینر ها وجود دارد که باعث می شود الگوریتم جستجوی مسیر نتواند بطور مستقیم اعمال شود.

## فصل سوم الگوریتم A-star

### ۱-۳- الگوریتم A-star چیست؟

الگوریتم A-star یک الگوریتم مسیریابی است برای پیدا کردن کوتاه ترین مسیر در یک گراف وزن دار است که بر مبنای الگوریتم دایجسترا<sup>۱</sup> ساخته شده است و این به این معنی است که یک الگوریتم حریصانه<sup>۲</sup> به حساب می آید. از این الگوریتم در برنامه های مسیریابی، رباتیک، هوش مصنوعی، بازی های رایانه ای و ... استفاده می شود چرا که علاوه بر پاسخ درست و قابل اعتماد دارای سرعت خوبی نیز هست. همچنین این الگوریتم از یک تابع اکتشافی<sup>۳</sup> استفاده می کند که به ما تخمینی از فاصله فعلی ما تا نقطه هدف را ارائه می دهد.

الگوریتم A\* به این صورت عمل می کند که در هر مرحله یک تابع  $f(n)$  را بررسی می کند که هزینه جابجایی ما از نقطه فعلی تا نقطه  $n$  است، این تابع از مجموع  $g(n)$  و  $h(n)$  تشکیل شده است.  $g(n)$  هزینه (فاصله) مشخص از نقطه فعلی تا نقطه  $n$  را به ما برمی گرداند و  $h(n)$  تخمین فاصله نقطه  $n$  تا نقطه هدف را به ما برمی گرداند که همان تابع اکتشافی است که پیشتر به آن اشاره کردیم. الگوریتم A\* در هر مرحله با محاسبه تابع  $f(n)$  برای گره های در لیست باز کم هزینه ترین گره را انتخاب می کند و به آن می رود.

### ۲-۳- نحوه عملکرد الگوریتم A-star

پیش از هرچیزی ما دو لیست باز و بسته را تعریف می کنیم که لیست باز شامل گره هایی است که هنوز به آنها نرفته ایم و لیست بسته لیست گره هایی است که آنها را مشاهده کرده ایم.

- لیست باز را ایجاد می کنیم.
- گره شروع را در لیست باز قرار می دهیم. لیست بسته را ایجاد می کنیم.

---

<sup>۱</sup>Dijkstra Algorithm

<sup>۲</sup>Greedy Algorithm

<sup>۳</sup>Heuristic function

- مراحل زیر را تا زمانی که لیست باز خالی نشده است ادامه می‌دهیم.
  1. از بین گره‌های موجود در لیست باز گره با کمترین  $f$  را پیدا کنید و اسم آنرا  $q$  بگذارید.
  2.  $Q$  را از لیست باز خارج کنید.
  3.  $q$  را به نوبت چک می‌کنیم که در کدام یک از دسته‌های زیر قرار دارند
    - i. اگر نقطه هدف بود، دیگر جستجو نمی‌کنیم.
    - ii. اگر هدف نبود توابع  $g, h$  را برای آن محاسبه می‌کنیم.
    - iii. اگر این گره در لیست باز بود و مقدار  $f$  آن کمتر از  $f$  محاسبه شده جدید بود از آن رد می‌شویم. اگر بیشتر بود پدر آن را عوض کرده و  $q$  را بجایش قرار می‌دهیم.
    - iv. اگر این گره در لیست بسته بود و مقدار  $f$  آن کمتر از  $f$  محاسبه شده جدید بود از آن رد می‌شویم. اگر بیشتر بود پدر آن را عوض کرده و  $q$  را بجایش قرار می‌دهیم همچنین آنرا دوباره به لیست باز اضافه می‌کنیم.
- 4.  $q$  را در لیست بسته قرار می‌دهیم.

### ۳-۳- تابع اکتشافی

- محاسبه تابع  $g$  کار پیچیده‌ای نیست اما چگونه تابع  $h$  را محاسبه کنیم؟  
ما برای اینکار میتوانیم از دو روش عمل کنیم
- (a) محاسبه دقیق مقدار  $h$  که کار زمان‌بری است و شاید کار الگوریتم ما را سخت هم بکند.
- (b) استفاده از تکنیک‌های مختلف برای بدست آوردن تخمینی از مقدار  $h$ .
- در این گزارش به چند تکنیک برای تخمین زدن فاصله نقاط تا نقطه هدف می‌پردازیم.

1. فاصله منهتن<sup>1</sup>

این روش مجموع مقادیر اختلاف مختصات نقطه فعلی تا نقطه پایانی است که فرمول آن به صورت زیر است:

$$H(n) = (\text{current\_node.x} - \text{goal\_node.x}) + (\text{current\_node.y} - \text{goal\_node.y})$$

از این روش بیشتر در موقعیت هایی استفاده می‌کنیم که فقط در چهار جهت اصلی می‌توانیم حرکت کنیم.

2. فاصله مورب<sup>2</sup>

این چیزی بیش از بزرگترین مقدار مطلق تفاوت بین مختصات x و y سلول فعلی و سلول هدف نیست که فرمول آن به صورت زیر است:

D برابر سائیز هر گره است و  $D_2$  قطر فاصله دو گره است.

$$D_x = (\text{current\_node.x} - \text{goal\_node.x})$$

$$D_y = (\text{current\_node.y} - \text{goal\_node.y})$$

$$H(n) = D * (D_x + D_y) + (D_2 - 2 * D) * \min(D_x, D_y)$$

از این روش بیشتر برای موقعیت هایی استفاده می‌شود که توان حرکت در جهت را داریم.

---

<sup>1</sup> Manhattan Distance

<sup>1</sup> Diagonal Distance

3. فاصله اقلیدسی<sup>1</sup>

فاصله قطری بین دو نقطه فعلی و پایان است که توسط فرمول زیر بدست می‌آید:

$$H(n) = \text{sqrt}((\text{current\_node}.x - \text{goal\_node}.x)^2 + (\text{current\_node}.y - \text{goal\_node}.y)^2)$$

---

<sup>2</sup> Euclidean Distance

## فصل چارم جمع بندي و نتیجه‌گیری



## جمع‌بندی و نتیجه‌گیری

این گزارش خلاصه‌ای از مقاله Geometric A-Star Algorithm است که در آن روشی برای بهینه‌سازی الگوریتم A-star ارائه شده است، که با ترکیب مزایای الگوریتم A-star و الگوریتم درون‌یابی به یک روش بهینه برای پیدا کردن کوتاه‌ترین مسیر با کمترین هزینه منجر شده است. همچنین با توضیح الگوریتم سنتی A-star سعی در ارائه دید بهتر به خواننده این گزارش دارد.

## منابع و مراجع

- [1] M. Gang Tang; M. Congqi Tang; M. Christophe Claramunt; “Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment”

### **Abstract**

This report is a summary of a method to improve the A-star routing algorithm, which provides a geometric method of the A-star algorithm for self-guided vehicles, which increases the speed and quality of the traditional algorithm.

#### **Key Words:**

Automated Guided Vehicle (AGV)

Path planning

A-star Algorithm



**Amirkabir University of Technology  
(Tehran Polytechnic)**

**Department of Mathematics and Computer Science**

**Second report**

# **Algorithms for path planning and graph traversal**

**By  
Arman Salehi**

**Supervisor  
Dr. Ghatei**

**March 2024**