

## ۱ مقدمه و هدف پروژه

**مسئله مورد بررسی:** یافتن کوتاه‌ترین مسیر جاده‌ای (بر حسب کیلومتر) در یک گراف وزن‌دار که گره‌های آن مراکز استان‌ها و یال‌های آن، فاصله‌های جاده‌ای بین مراکز همسایه هستند. هدف، مقایسه کارایی و بهینگی (Optimality) این الگوریتم‌ها از نظر طول مسیر یافت شده، تعداد گره‌های بازدیدی و زمان اجرا است.

## ۲ فرایند پیاده‌سازی

### ۲.۱ بارگذاری و ساختار داده‌ها

در ابتدا، داده‌های مورد نیاز شامل فاصله‌های جاده‌ای، همسایگی‌ها و فاصله‌های خط مستقیم بین ۳۱ مرکز استان از فایل‌های اکسل بارگذاری شدند.

**ساختار گراف:** گراف مراکز استان‌ها به صورت یک دیکشنری مجاورت (Adjacency Dictionary) پیاده‌سازی شد، که در آن هر کلید نماینده یک شهر و مقدار آن یک دیکشنری دیگر شامل همسایگان و وزن یال (فاصله بر حسب کیلومتر) است.

### ۲.۲ الگوریتم‌ها و دلیل انتخاب

نحوه پیاده‌سازی	دلیل انتخاب	الگوریتم
استفاده از deque (صف) برای حفظ ترتیب سطح‌به‌سطح جستجو.	به عنوان یک معیار پایه برای جستجوی ناآگاهانه و یافتن کوتاه‌ترین مسیر بر حسب تعداد گام‌ها.	BFS (جستجوی سطح-اول)
استفاده از list به عنوان پشته (Stack).	به عنوان نمونه‌ای دیگر از جستجوی ناآگاهانه، که مسیر را به صورت عمیق دنبال می‌کند و حافظه کمتری مصرف می‌کند.	DFS (جستجوی عمق-اول)
استفاده از heapq (صف اولویت‌دار) بر اساس تابع ارزیابی $f(n)=g(n)+h(n)$ .	به عنوان الگوریتم جستجوی آگاهانه و بهینه، مناسب برای گراف‌های وزن‌دار.	A*

Weighted A*	برای بررسی تأثیر افزایش وزن هیوریستیک (w) در جهت افزایش سرعت جستجو و Trade-off بین سرعت و بهینگی.	مشابه A*، اما با تابع ارزیابی $f(n) = g(n) + w \cdot h(n)$ که در آن $w > 1$
-------------	---	--

## ۲,۳ فرضیات (Heuristic)

- تابع **Heuristic (h(n))**: فاصله خط مستقیم بین گره فعلی و گره هدف از فایل ProvinceCentersStraightLineDistances.xlsx استخراج شد.
- **Admissibility و Consistency**: فاصله خط مستقیم همیشه کمترین مسافت ممکن بین دو نقطه است و بنابراین، هرگز بزرگتر از فاصله واقعی جاده‌ای (هزینه واقعی از n تا هدف،  $h^*(n)$  نیست. این امر تضمین می‌کند که هیوریستیک **Admissible و Consistent** است، در نتیجه، الگوریتم A\* بهینه‌ترین مسیر را پیدا خواهد کرد.

## ۳ نتایج و تجزیه و تحلیل

برای تحلیل نهایی، دو مسیر با ویژگی‌های متفاوت (یک مسیر کوتاه و یک مسیر بلند) انتخاب و الگوریتم‌ها مقایسه شدند:

### ۳,۱,۱ مقایسه نتایج برای مسیر تبریز - بندرعباس

الگوریتم	مسیر یافت شده	طول مسیر (km)	گره‌های بازدیدی	زمان اجرا (s)
BFS	تبریز → زنجان → قزوین → کرمان → بندرعباس	2058	14	0.000067
DFS	(بسیار طولانی و غیربهینه)	5391	15	0.000058
A*	تبریز → زنجان → همدان → خرم‌آباد → شهرکرد → یاسوج → شیراز → بندرعباس	2007	24	0.023055
Weighted A* (w=1.5)	تبریز → زنجان → قزوین → کرمان → بندرعباس	2058	7	0.010721

تجزیه و تحلیل:

$A^*$  بهترین طول مسیر (۲۰۰۷ کیلومتر) را یافت، که بهینگی آن را تأیید می‌کند ۷۷۷۷.

- **DFS** یک مسیر بسیار غیربهینه را پیدا کرد، که نشان می‌دهد در گراف‌های وزن‌دار، **DFS** فاقد بهینگی است.
- **Weighted  $A^*$  ( $w=1.5$ )** با بازدید از تنها ۷ گره، بیشترین صرفه‌جویی در فضای جستجو را داشت و بسیار سریع عمل کرد، اما مسیری با طول ۲۰۵۸ کیلومتر) غیربهینه نسبت به ( $A^*$  یافت. این نشان دهنده Trade-off بین سرعت و کیفیت مسیر است.

۳,۲ مقایسه نتایج برای مسیر تهران - اصفهان

الگوریتم	مسیر یافت شده	طول مسیر (km)	گره‌های بازدیدی	زمان اجرا (s)
BFS	تهران → اراک → اصفهان	562	2	0.000026
DFS	(بسیار طولانی و غیربهینه)	4835	27	0.00013
$A^*$	تهران → قم → اصفهان	421	4	0.003117
Weighted $A^*$ ( $w=1.5$ )	تهران → قم → اصفهان	421	3	0.003005

تجزیه و تحلیل:

- در این مسیر کوتاه،  $A^*$  و **Weighted  $A^*$**  هر دو مسیر بهینه (۴۲۱ کیلومتر) را یافتند.
- **Weighted  $A^*$  ( $w=1.5$ )** با کمترین گره بازدیدی در میان الگوریتم‌های آگاهانه (۳ گره)، کارایی خود را در سرعت جستجو تأیید کرد.
- **BFS** با کمترین زمان اجرا، مسیری با تنها ۲ یال (تهران  $\rightarrow$  اراک  $\rightarrow$  اصفهان) یافت، اما طول مسیر (۵۶۲ کیلومتر) آن بهینه نبود. این دوباره تأکید می‌کند که BFS کوتاه‌ترین مسیر را بر حسب تعداد گام‌ها می‌یابد، نه لزوماً کوتاه‌ترین مسافت جاده‌ای.

## ۴ جمع‌بندی و نتیجه‌گیری

- **A\*** بهترین عملکرد را در تضمین بهینگی مطلق (یافتن کوتاه‌ترین مسیر جاده‌ای) داشت.
- **Weighted A\*** به دلیل کاهش شدید تعداد گره‌های بازدیدی، به ویژه در مسیرهای طولانی، کارایی بالایی را در سرعت جستجو نشان داد، اما به طور بالقوه ممکن است مسیر غیربهینه را انتخاب کند (مسیر تبریز \_ بندرعباس).
- الگوریتم‌های ناآگاهانه (BFS و DFS) به دلیل نادیده گرفتن وزن یال (فاصله جاده‌ای)، برای مسائل مسیریابی در گراف‌های وزن‌دار مناسب نیستند و مسیرهای غیربهینه یا طولانی پیدا کردند.

## ۵ محدودیت‌ها و مشکلات پروژه

- **محدودیت داده:** اندازه کوچک گراف (۳۱ گره) ممکن است تفاوت‌های کارایی (زمانی) بین الگوریتم‌های آگاهانه و ناآگاهانه را کم‌رنگ کرده باشد. در گراف‌های بزرگتر، انتظار می‌رود مزیت **A\*** و **Weighted A\*** در کاهش گره‌های بازدیدی بسیار پررنگ‌تر باشد.
- **پیچیدگی DFS:** در پیاده‌سازی DFS، به دلیل ماهیت بازگشتی جستجو و وجود حلقه‌ها در گراف (دو طرفه بودن یال‌ها)، کنترل گره‌های **visited** برای جلوگیری از جستجوی بی‌نهایت ضروری بود.